



TECHNISCHE  
UNIVERSITÄT  
WIEN

Vienna University of Technology

## DIPLOMA THESIS

# Diffusion Simulation in Metals with Moving Phase Boundaries

INSTITUTE OF APPLIED PHYSICS  
INSTITUTE OF MATERIAL SCIENCE AND TECHNOLOGY

Supervisors:

PETER MOHN

ERNST KOZESCHNIK

Author:

MANUEL OSCHGAN

MATRICULATION NUMBER: 0526133

LEYSTRASSE 26/1/37 - 1200 WIEN

## Abstract

Moving phase boundaries have been researched for a long time by different fields of science and many interesting processes exist that involve moving interfaces, like the melting of ice, phase transformations in metals or tumor growth. To gain an overview of the field, some important formulations of moving boundary problems are described and analytic solutions presented, where possible. Three basic types of numerical approaches are emphasized and several different methods, belonging to each type, are reviewed. The model of moving interfaces is implemented in the microstructural simulation-software MatCalc. A variable-spaced-grid approach, similar to the Murray Landis method, is used to track the interface movement. In order to compute the interface velocity, the local equilibrium hypothesis is applied. Reducing deviations from the mass balance requires inclusion of the Murray Landis correction term into the diffusion equation. Heat treatments are introduced to simulate phase transformations of iron-carbon alloys and at the end, a solidification example and a peritectic reaction, involving austenite and ferrite, are shown.

## Zusammenfassung

Bewegte Phasengrenzen werden seit geraumer Zeit von unterschiedlichen Disziplinen der Wissenschaft untersucht und viele interessante Prozesse beinhalten solche Grenzflächen, wie zum Beispiel das Schmelzen von Eis, Phasenumwandlungen in Metallen oder Tumor Wachstum. Um einen Überblick über das Gebiet zu erhalten, werden einige wichtige Formulierungen solcher Probleme vorgestellt und analytische Lösungen präsentiert, wo diese möglich sind. Es werden drei grundlegende Typen von numerischen Lösungsverfahren hervorgehoben und einige unterschiedliche Methoden besprochen, die zu diesen Typen gehören. Die Mikrostruktur Simulations-Software MatCalc wird um ein Modell zur Beschreibung von bewegten Phasengrenzen erweitert. Dieses Modell mit variabler Gitter-Breite, ähnlich zur Murray Landis Methode, wird verwendet um die Interface Bewegung zu beschreiben. Die Geschwindigkeit der Phasengrenze stammt aus einem lokalen Gleichgewichts-Ansatz. Eine Reduktion der Abweichungen von der Massenbilanz wird durch die Verwendung der Murray Landis Korrektur sichergestellt. Um die Simulation von Phasentransformationen in Eisen-Kohlenstoff Legierungen zu ermöglichen, werden Wärmebehandlungen verwendet. Den Abschluss stellt die Simulation eines Erstarrungs-Beispiels und einer peritektischen Reaktion dar.

# Contents

1	Introduction	1
2	Objectives	2
3	State of the art	3
3.1	Diffusion Theory . . . . .	3
3.1.1	Microscopic diffusion . . . . .	3
3.1.2	Macroscopic diffusion . . . . .	4
3.1.3	Joining both worlds . . . . .	5
3.1.4	Multi-component diffusion . . . . .	5
3.2	Moving boundary problems . . . . .	6
3.2.1	Classical Stefan Problem . . . . .	7
3.2.2	Analytic solutions . . . . .	13
3.2.3	Front tracking methods . . . . .	18
3.2.4	Front fixing methods . . . . .	26
3.2.5	Fixed domain methods . . . . .	32
4	Simulation and Results	36
4.1	Diffusion calculation in MatCalc . . . . .	36
4.1.1	Implementation . . . . .	37
4.1.2	Explicit and trapezoidal methods . . . . .	39
4.1.3	Boundary conditions . . . . .	40
4.1.4	Comparison with analytic solutions . . . . .	41
4.1.5	Darken experiment . . . . .	44
4.2	Moving interfaces . . . . .	48
4.2.1	Variable grid distribution . . . . .	50
4.2.2	Interface checks and cell spacing . . . . .	52
4.2.3	Time step limit . . . . .	58
4.2.4	Composition treatment . . . . .	61
4.2.5	Interface velocity . . . . .	65
4.2.6	Murray Landis correction . . . . .	67
4.2.7	Heat treatments . . . . .	68
4.2.8	Examples . . . . .	69

---

4.3	Phase transformations involving moving interfaces . . . . .	77
4.3.1	Ferritic solidification in iron-carbon alloys . . . . .	81
4.3.2	Peritectic reaction involving ferrite and austenite . . . . .	85
5	Discussion	90
6	Summary	91
	Bibliography	92
	List of Figures	96
	List of Tables	99

# 1 Introduction

In several different fields of science, researchers have dealt with moving phase boundaries, being it engineers, who want to address practical problems or mathematicians, who develop models or numerical analysts, who find suitable algorithms where analytic solutions are not possible. Some examples for practical problems are melting of ice, oxygen diffusion, solid-liquid transitions in rewritable recording and also many biological processes involve moving interfaces, like wound healing, tumor growth or transport mechanisms in brain tissue. Although the practical part of the thesis is mainly focused on diffusion problems, due to the similarity to heat conduction problems, some of the theoretical models are explained using the heat equation.

The theoretical part of this thesis starts with a short introduction in diffusion theory, where the microscopic and the macroscopic view are compared and a link between diffusion coefficients and atomistic parameters is established.

Afterwards, the classical Stefan problem is presented, a case of heat conduction with a moving phase boundary between a water and an ice phase. Besides some enhancements of the model like convection, a mixed region of the two phases and the extension to higher dimensions, some analytic solutions for simple problems, like oxygen growth on wafers, are addressed. An important type of analytic solutions are the so called ‘Similarity solutions’, with one of the first of this type, which is also presented in this chapter, has been derived by Franz Neumann. The subsequent three sections present efforts to numerically solve the moving boundary problem, distinguished by the way of computing the moving boundary. Front tracking methods trace the interface explicitly, either by using a fixed grid with approximations near the interface or by deforming the grid to represent the exact position. Worth mentioning are also the finite element methods, which are briefly described in this section. Approaches that are mainly based on variable transformations are the so called front fixing methods. They retain the interface at a constant position, but still require a closure condition at this place. The aim of body fitted coordinates is to simplify the computations after transformation to a rectangular grid. In fixed domain methods, the interface is not calculated directly, instead a function that holds on the whole domain is utilized to describe the problem. Especially for difficult-to-track interfaces, these are the techniques of choice. The chapter ‘Simulation and Results’ introduces the diffusion algorithms of MatCalc and describes the implementation of moving-boundary handling. Several computed examples are also presented at the end.

## 2 Objectives

The goal of this thesis is the development and implementation of an algorithm for solving moving boundary problems into the simulation module of the software MatCalc. Existing code for diffusion simulation should be advanced and numerical errors due to approximations in the model must be kept in an acceptable range. Flexibility should be given for further implementation of different models for interface-velocity calculation and a simple model for two component interfaces is to be included. Proper simulation of a basic interface-movement example applying this velocity model must be carried out. To be able to simulate phase transformations in metals, the heat treatment functionality should be used in conjunction with the interface calculation to solve a solidification example. Additionally, an example of a peritectic reaction should be demonstrated, to correctly represent the expected experimental behavior.

## 3 State of the art

Relevant theories that describe diffusion processes and moving boundary problems are presented in this chapter. The reader should get an overview of the interesting field and learn basics that are needed to understand the following chapter, which contains the details about the practical implementation of moving boundary problems in the software MatCalc.

### 3.1 Diffusion Theory

This section gives a short introduction in diffusion theory, where the different views of microscopic and macroscopic diffusion processes are explained. Although the laws of the macroscopic processes have been observed phenomenologically, it is possible to relate them to the atomistic behavior of a diffusing substance. Furthermore, a multi-component version of the diffusion equation is derived.

#### 3.1.1 Microscopic diffusion

On a microscopic scale, the predominant factors for diffusion in solids are either the vacancy exchange mechanism or interstitial diffusion using lattice gaps. For substitutional atoms of the size of matrix atoms, there is also a theoretical possibility of direct exchange, but its probability is magnitudes lower. At high temperatures, substitutional diffusion is the dominant factor, because the vacancy density in metals rises with increasing temperature and more atoms get pulled out of their bulk lattice positions to move to the surface. Since most interstitial lattice positions are usually empty and the activation energy for interstitial diffusion is much lower than for vacancy exchange, interstitial diffusion is much faster. Also, the density of interstitial vacancies is not temperature dependent, whereas substitutional diffusion stalls at lower temperatures. Since the simulation of atomistic systems with at least millions of atoms interacting is computationally too expensive, the problem has to be viewed at a macroscopic scale also.

### 3.1.2 Macroscopic diffusion

The basis for macroscopic diffusion models was laid by the german physiologist Adolf Fick, who investigated diffusive mixing of salt and water. According to the phenomenological law he observed, diffusion is driven by the composition gradient of the diffusing particles. Through this gradient, a diffusive flux occurs from regions with high particle density to regions with low particle density, which is formulated as

$$j = -D \frac{\partial c}{\partial x} \quad (3.1)$$

for one dimension, with  $x$  being the spacial coordinate,  $j$  the flux and  $c$  the concentration gradient. The proportionality constant  $D$  is named diffusion coefficient. Combining the flux with the continuity equation

$$\frac{\partial c}{\partial t} + \partial_x j = 0, \quad (3.2)$$

which represents the law of mass conservation in a closed system, results in Fick's second law of diffusion

$$\frac{\partial c}{\partial t} - D \frac{\partial^2 c}{\partial x^2} = 0. \quad (3.3)$$

It is a second order partial differential equation for which analytic solutions of simple problems exist, but it must be solved numerically for more complex problems. The diffusion coefficient is considered independent of space and composition in this formulation.

Writing equations (3.1) to (3.3) in three dimensions using index notation leads to

$$\frac{\partial c(\vec{x}, t)}{\partial t} + \partial_i j_i(\vec{x}, t) = 0 \quad (3.4)$$

$$j_i = -D \frac{\partial c}{\partial x_i} \quad (3.5)$$

$$\frac{\partial c}{\partial t} - D \Delta c = 0. \quad (3.6)$$

Again, in equation (3.6) the diffusion coefficient  $D$  is assumed to be constant, therefore, it can be written in front of the differentials forming the Laplace operator  $\Delta$ . This, however, is only true for isotropic media. For anisotropic media, a symmetric diffusion matrix is used instead of the constant coefficient.



### 3.1.3 Joining both worlds

Linking both the macroscopic and the microscopic view, it is possible to relate the previously phenomenological diffusion coefficient  $D$  with atomistic properties like the jump frequency of atoms. This is done by comparing the atomistic root mean square displacement, which is based on the jump frequency, with the same displacement being derived from the phenomenological model, which includes the diffusion coefficient. As a result, the Einstein equation

$$D = \frac{1}{2d} \Gamma_D \lambda^2 \quad (3.7)$$

is obtained, with the dimensionality  $d$  and the atomic distance  $\lambda$ . The jump frequency

$$\Gamma_D = \nu Z \exp\left(-\frac{Q_D}{k_B T}\right) \quad (3.8)$$

contains the ratio of the activation energy  $Q_D$  and the thermal energy  $k_B T$ , the number of nearest neighbors  $Z$  and the vibration frequency  $\nu$ . The apparent contradiction of the time reversibility at the microscopic scale and the irreversibility at the macroscopic scale can be understood by looking at theories of statistical thermodynamics. Once a many particle system evolves from a non-equilibrium state to an equilibrium state, it gets very unlikely that the initial state is reached once again. This is expressed through the second law of thermodynamics, according to which a closed system's entropy can not decrease with time. Rudolf Clausius regarded the entropy as the dissipative work done by the atoms or molecules during such a transformation. Also it is important, that at the initial stage, the different particles are thought to be uncorrelated, but during the transformation they become correlated through their interaction, which is another formulation for rising entropy.

### 3.1.4 Multi-component diffusion

In chemically inhomogeneous systems, the diffusive force

$$\vec{F} = -\nabla \mu(\vec{r}) \quad (3.9)$$

can be expressed in terms of the chemical potential  $\mu$ , with the definition

$$\mu = \mu_0 + RT \ln a. \quad (3.10)$$

$R$  is the gas constant,  $T$  the temperature,  $a$  the thermodynamic activity and  $\mu_0$  the standard chemical potential (usually at  $a = 1$ ).

The flux  $\vec{j} = c\vec{u}$  is proportional to the drift velocity  $\vec{u}$  with factor  $c$ , the local concentration.  $\vec{u} = B\vec{F}$  relates the force and the velocity, with the mobility  $B$  being either a

scalar or a tensor for non-isotropic systems. Combining these equations, a generalized flux

$$\vec{j} = -cB\nabla\mu(\vec{r}) \quad (3.11)$$

is obtained, which is driven by the gradient of the chemical potential. This is important for multi-component systems, where the potentials of the different substances may differ.

In multi-component diffusion, the chemical potential of one substance is determined by the compositions of the other substances, which in turn are location dependent. For convenience, the flux can be expressed as gradient of the compositions, instead of the chemical potential. The gradients of the chemical potentials then become part of the diffusion matrix, which is especially useful for the kind of numerical calculations done in the software MatCalc. Rewriting (3.11) in terms of the composition yields

$$j_{il} = -c_l B_l \frac{\partial \mu(X_k)}{\partial r_i} = -c_l B_l \sum_{k=1}^n \frac{\partial \mu}{\partial X_k} \frac{\partial X_k}{\partial r_i} = -X_l B_l \sum_{k=1}^n \frac{\partial \mu}{\partial X_k} \frac{\partial c_k}{\partial r_i}. \quad (3.12)$$

The indices  $l$  and  $k$  refer to the elements and  $i$  is the vector index and the composition in mole fraction is denoted by  $X$ . This leads to diffusion coefficients

$$D_{lk} = X_l B_l \frac{\partial \mu}{\partial X_k}, \quad (3.13)$$

which can be used to rewrite the flux  $\vec{j}_l = -D_{lk}\nabla c_k$ , where the summation over the element indices is carried out implicitly. Inserting the flux into the continuity equation

$$\frac{\partial c_l}{\partial t} + \nabla \cdot \vec{j}_l = 0 \quad (3.14)$$

yields the multi-component version of Fick's second law

$$\frac{\partial c_l}{\partial t} - \sum_{k=1}^n D_{lk} \Delta c_k = 0, \quad (3.15)$$

where the summation is written explicitly for clarification. It shows, that for solving one component's diffusion equation, all other components concentrations must be taken into account. For further details of diffusion models in metals see [JRK<sup>+</sup>07].

## 3.2 Moving boundary problems

Boundary problems have been studied for a long time from scientists, both to describe processes in nature and technology. The distinction between free and moving boundary

problems is the time dependence of the latter ones. Although in both cases the exact position of an interface is not known a priori, for free boundary problems the interface is stationary, whereas for moving boundary problems, it can migrate with time. Examples of free boundary problems are seepage through dams and porous flow between different interfaces. Having a sharp boundary implies, that one of the porous media is saturated. Moving boundary problems, on the other hand, arise frequently in phase transformations of different materials. Mathematically, both problems are often described using differential equations, which have to satisfy certain conditions at the domain boundary. There exist several analytic solutions for simplified problems, but more sophisticated examples require numerical methods. John Crank gives a very broad overview of both diffusion and heat conduction problems involving moving interfaces. His book ‘Free and Moving Boundary Problems’ [Cra87] gives a review of the complete field, being mostly focused on heat conduction problems. ‘The Mathematics of Diffusion’ [Cra75], contains moving boundary problems in chapter 13.

### 3.2.1 Classical Stefan Problem

A simple case of a moving boundary problem was formulated by J. Stefan in 1889 [Ste89]. It deals with heat conduction and interface movement during formation of ice in polar seas. The interface is considered to be infinitesimal thin for Stefan problems, which requires that the real interface thickness is negligible. Although being a heat conduction problem, its solvability is very similar to simple diffusion problems in solids. The sheet of ice, initially at the melting temperature of zero, will produce a region of water that moves into the sheet, and both phases are separated by a thin interface. In the whole ice block, the temperature is fixed at the initial value, whereas in the water region, it drops, from a boundary value of  $U_0$ , to the ice temperature. The heat conduction equation

$$c\rho \frac{\partial u}{\partial t} = K \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < s(t), \quad t > 0, \quad (3.16)$$

is used to calculate the thermal evolution in the water region, which is bounded by the interface at position  $s(t)$ . Variables  $c, \rho, u, K$  are the specific heat, the density, the temperature and the thermal conductivity respectively. Using

$$u = U_0, \quad x = 0, \quad t > 0, \quad (3.17)$$

as boundary condition and

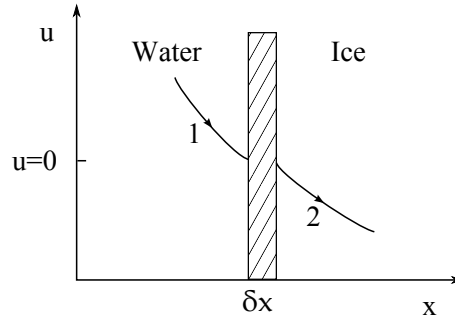
$$u = 0, \quad x > 0, \quad t = 0, \quad (3.18)$$

$$s(0) = 0 \quad (3.19)$$

as initial conditions, still the two conditions

$$\left. \begin{aligned} u &= 0, \\ -K \frac{\partial u}{\partial x} &= L \rho \frac{ds}{dt}, \end{aligned} \right\} \quad x = s(t), \quad t > 0, \quad (3.20)$$

are needed at the position of the interface, with  $L$  being the latent heat which is consumed during the phase transformation.



**Figure 3.1:** Stefan condition for two phase problem

In an extended case, the heat flow can also occur in the ice phase, when it is not at the melting temperature initially. A separate heat conduction problem has to be solved for both parts. Again, equation (3.16) is used, but with different parameter values for each phase. To connect both problems at the phase boundary, the modified Stefan conditions are

$$\left. \begin{aligned} u_1 &= u_2 = 0, \\ K_2 \frac{\partial u_2}{\partial x} - K_1 \frac{\partial u_1}{\partial x} &= L \rho \frac{ds}{dt}, \end{aligned} \right\} \quad x = s(t), \quad (3.21)$$

with the assumptions, that ice and water have the same density and the volume change during melting can be ignored. The problem can be generalized to multiple phases, by adding one further diffusion equation for each phase and connecting the phases again with the Stefan condition. Figure 3.1 shows the transformation process of a  $\delta x$  thick layer of ice to water, which requires the heat  $L \rho \delta x$  to be supplied. The amount of heat that enters the shaded region is labeled with index one, both in the figure and in equation (3.21). Balance between that term, and the one that enters the ice region, is assured by the latent heat term on the right side of the equation.

Often, the equations are written in a dimensionless form, what leads, by introduction

of the variable transformation

$$X = \frac{x}{l}, \quad T = \frac{K}{c\rho} \frac{t}{l^2}, \quad U = \frac{u}{U_0}, \quad S = \frac{s}{l}, \quad (3.22)$$

with  $l$  being an arbitrarily chosen standard length, to a simplified form of the heat equation

$$\frac{\partial U}{\partial T} = \frac{\partial^2 U}{\partial X^2}, \quad 0 < X < S(t), \quad T > 0, \quad (3.23)$$

the modified boundary and initial conditions

$$\begin{aligned} U &= 1, & X &= 0, & T &> 0, \\ U &= 0, & X &> 0, & T &= 0, \\ S(0) &= 0, \end{aligned} \quad (3.24)$$

and the closure at the interface

$$\left. \begin{aligned} U &= 0, \\ \frac{\partial U}{\partial X} &= \lambda \frac{dS}{dT}, \end{aligned} \right\} \quad X = S(T), \quad T > 0. \quad (3.25)$$

With equation (3.25), a dimensionless parameter  $\lambda = \frac{L}{c u_0}$  has been introduced, whose inverse is called the Stefan number. If the units on both sides of an equation do not seem to fit together, such a transformation was performed.

### Convection and density differences

For the derivation of the Stefan condition, which implies energy conservation across the interface, the assumption is made that both phases are incompressible and at rest. In a more general case, also the mass and momentum have to be conserved, with the mass conservation taken into account by the equation

$$\varrho_1 \frac{dS}{dT} = \varrho_2 \left( \frac{dS}{dT} - v \right), \quad X = S(T) \quad (3.26)$$

for two phases, of different densities  $\varrho_1$  and  $\varrho_2$ , where  $v$  denotes the velocity of the liquid phase. To incorporate the convection, a term proportional to the temperature gradient is added to the heat equation, resulting in

$$\frac{\partial U}{\partial T} = \frac{\partial^2 U}{\partial X^2} - v \frac{\partial U}{\partial X}. \quad (3.27)$$

Density variations are not the only reason that can cause the convection term in (3.27). For higher dimensional cases, this term and the problem itself become much more complicated.

### ‘Mushy’ region

Some scientists tried to describe a mixture between liquid and solid phases, called a ‘mushy’ region. Mathematically, the addition of a heating term  $h(T, x, t)$  to the right side of equation (3.16), will cause a non-sharp interface, consisting of a finite region at the melting temperature. A sophisticated semi-empirical description was developed by Solomon et al. in 1982 [SWA82], based upon microscopic studies of Thomas and Westwater [TW63]. They distinguish three zones divided by the interfaces  $X(t)$  and  $Y(t)$ , with the first being the solid/mushy interface, and the second the mushy/liquid interface. Both the liquid and the mushy region are assumed to be at the melting temperature  $T_m$ , only the solid region has a variable temperature. Solomon et al. incorporate a fixed fraction of the latent heat  $\theta L$  into the mushy region and relate the width of the region with the temperature gradient in the solid phase, at the interface, by

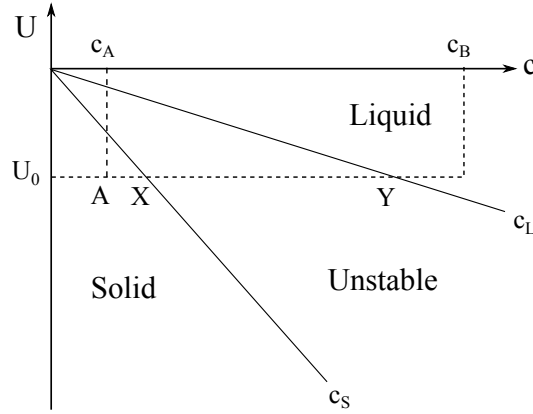
$$[Y(t) - X(t)] \left. \frac{\partial u}{\partial x} \right|_{X(t)} = \text{constant}. \quad (3.28)$$

This leads to an extended version of the Stefan condition, namely

$$K \left. \frac{\partial u}{\partial x} \right|_{X(t)} = \varrho L \left[ \theta \frac{dX}{dt} + (1 - \theta) \frac{dY}{dt} \right]. \quad (3.29)$$

### Coupled diffusion and heat conduction

A typical situation, where simultaneous diffusion and heat conduction processes occur, is an alloy solidification problem. The reason for this is that, in alloys, the melting temperature depends on the local composition of one or more secondary substances, called the ‘impurities’, in a primary substance, which ensembles the majority of the alloy. The simple eutectic diagram in figure 3.2 will be used to describe the situation of one alloying element. Suppose the concentration of the impurity in the alloy is  $c_A$  at some arbitrary point in the sample. If this point is at the temperature  $U_0$ , then this corresponds to the point  $A$  in the figure. Increasing the concentration to the value  $c_B$  at the same temperature, leads to point  $B$  in the liquid region. Between the solidus and the liquidus line, there is an unstable region where no equilibrium between the pure substance and the alloying element is possible. However, both phases can exist there, at the compositions  $c_L$  and  $c_S$ . Therefore, the melting temperature will be defined by the impurity concentration and at a phase boundary, there will be a concentration jump of  $c_L - c_S$ . Describing such a problem requires the coupling of the



**Figure 3.2:** Eutectic phase diagram of an alloy

diffusion and heat conduction equation and two further conditions which specify the interface velocity. It will be presented in a non-dimensional form, what is the reason why for more than one phase, the time transformation  $T = \frac{K_0}{c_0 \rho} \frac{t}{l^2}$  is applied. The variables  $\kappa_i = \frac{K_i}{K_0} \frac{c_0}{c_i}$  and  $\gamma_i = \frac{K_i}{\lambda K_0}$  are introduced as coefficients of the temperature gradients, and each phase temperature is normalized to the arbitrary temperature  $u_0$ . Additionally, also the normalized diffusion coefficients  $\mathcal{D}_i = \frac{D_i}{D_0}$  are, combined with the equations

$$\frac{\partial U_1}{\partial T} = \kappa_1 \frac{\partial^2 U_1}{\partial X^2}, \quad \frac{\partial C_1}{\partial T} = \mathcal{D}_1 \frac{\partial^2 C_1}{\partial X^2}, \quad 0 < X < S(t), \quad (3.30)$$

$$\frac{\partial U_2}{\partial T} = \kappa_2 \frac{\partial^2 U_2}{\partial X^2}, \quad \frac{\partial C_2}{\partial T} = \mathcal{D}_2 \frac{\partial^2 C_2}{\partial X^2}, \quad S(T) < X < 1, \quad (3.31)$$

$$\left. \begin{aligned} \gamma_2 \frac{\partial U_2}{\partial X} - \gamma_1 \frac{\partial U_1}{\partial X} &= \frac{dS}{dT}, \\ \mathcal{D}_2 \frac{\partial C_2}{\partial X} - \mathcal{D}_1 \frac{\partial C_1}{\partial X} &= (C_1 - C_2) \frac{dS}{dT}, \\ U_1 = U_2 = g, \quad C_1 &= c_S(g), \quad C_2 = c_L(g), \end{aligned} \right\} \quad X = S(T), \quad (3.32)$$

needed to describe the whole problem. Variables with the lower index zero denote normalization values of the corresponding physical properties. The Stefan condition for the diffusion part of the problem is quite similar to the heat conduction condition, besides, that the composition at the interface is also included on the right side.

### Two space dimensions

To extend the heat equation to another space dimension, the second derivative will be replaced by the Laplace operator  $\Delta$ , but the heat conductivity  $K$  will still be considered uniform for a whole phase. The modified equation reads

$$\rho c_i \frac{\partial u_i}{\partial t} = K_i \Delta u_i, \quad \vec{x} \in D_i, \quad i = 1, 2, \quad 0 < t < E, \quad (3.33)$$

the conditions for the sample boundaries change to

$$\frac{\partial u_i}{\partial n} = g_i(\vec{x}, t), \quad \vec{x} \in D_i, \quad 0 < t < E, \quad (3.34)$$

$$u_1(\vec{x}, t) = u_2(\vec{x}, t) = u_m, \quad (3.35)$$

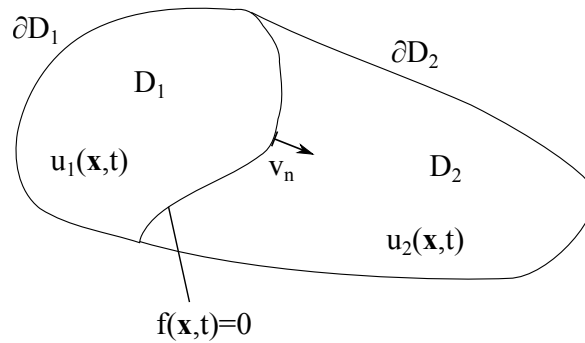
the modified Stefan condition is

$$K_2 \frac{\partial u_2}{\partial n} - K_1 \frac{\partial u_1}{\partial n} = -\rho L v_n, \quad (3.36)$$

and the conditions for the interface itself become

$$\left. \begin{aligned} u(\vec{x}, 0) &= u_0(\vec{x}), \\ f(\vec{x}, 0) &= f_0(\vec{x}), \end{aligned} \right\} \quad t = 0. \quad (3.37)$$

The equations above contain major changes compared to one dimensional models.  $D_i$  denotes the domain, in which the equations are evaluated and  $i$  represents the phase index. Some new variables like the end time  $E$ , the position vector  $\vec{x}$ , the melting temperature  $u_m$  and the temperature jump at the boundary  $g_i$  are introduced. Primary, the interface is now described by an implicit function of parameters  $\vec{x}$  and  $t$ . Also, the derivatives point into the normal direction  $n$  and the interface velocity  $v_n$  is orthogonal to the curve described by  $f(\vec{x}, t) = 0$ . This situation is illustrated in figure 3.3. By using,



**Figure 3.3:** Moving boundary problem in two dimensions



as shown below, the definition of the normal vector, the normal temperature-derivation, the interface velocity

$$\mathbf{n} = \nabla f / |\nabla f| = \nabla u_i / |\nabla u_i|, \quad (3.38)$$

$$\frac{\partial u_i}{\partial n} = \nabla u_i \cdot \mathbf{n} = (\nabla u_i \cdot \nabla f) / |\nabla f|, \quad (3.39)$$

$$v_n = \mathbf{v} \cdot \mathbf{n} = (\mathbf{v} \cdot \nabla f) / |\nabla f|, \quad (3.40)$$

and the total differential of the implicit function  $df(x, y, t) = 0$ , it is possible to obtain

$$|\nabla f| v_n = -\frac{\partial f}{\partial t}, \quad (3.41)$$

and insert it, combined with (3.39) into (3.36), resulting in a version of the Stefan condition

$$K_1 \nabla u_1 \cdot \nabla f - K_2 \nabla u_2 \cdot \nabla f = -\rho L \frac{\partial f}{\partial t}, \quad (3.42)$$

which contains derivatives of the implicit function.

### 3.2.2 Analytic solutions

This chapter will discuss some approximate analytic solutions for boundary value problems, which are mostly of the diffusion type. Although these solutions can not reproduce complex problems with enough detail, they can be used to solve cases where their assumptions hold, what often gives a great practical benefit. Steady state solutions simplify the problem in a way, that the concentration profile is assumed to be time independent, as it would be after enough time on a stationary interface. Based on the calculated concentration field, the interface motion is derived from the fluxes of atoms on the interface.

#### Oxygen growth on wafers

A nice example of oxygen-layer growth on silicon surface is described in Jackson's 'Kinetic Processes' [Jac04, p. 97]. Such a process was used in the early days of silicon manufacture, to conduct selective doping of wafers. Combined with a photomask applied with photolithography and etching, it is possible to dope only the parts where the photomask still exists, because it can not withstand the annealing temperatures in contrast to the oxide. At room temperature, a thin atomic layer of  $\text{SiO}_2$  will form quickly on the silicon surface. In air or an oxygen atmosphere, at higher temperatures above  $1000^\circ\text{C}$ , the growth process will be some micrometers per hour. The process is still used to isolate active devices on integrated circuits. Also worth to mention is that the growth of the oxide layer happens on the inside, between the silicon and the layer

itself. This is caused by oxygen diffusion through the oxide layer, readily forming new  $\text{SiO}_2$  on the interface, which will keep the interface clean and prevent the dirt from entering the wafer material. The diffusion equation

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} \quad (3.43)$$

reduces, when considering the time independence, to  $\frac{\partial^2 c}{\partial x^2} = 0$ , with the simple solution

$$c = ax + b. \quad (3.44)$$

The constants  $a$  and  $b$  are determined by the boundary conditions  $c = c_s$  at  $x = 0$  and  $c = c_e$  at  $x = l$ , resulting in

$$c = c_s - \frac{c_s - c_e}{l}x, \quad (3.45)$$

for the concentration in the sample, with  $l$  being the width of the silica layer,  $c_s$  being the oxygen concentration on the surface and  $c_e$  being the equilibrium concentration of oxygen in silica. For the boundary condition at the interface,  $c_e$  is taken, because the interface composition is assumed to be equal to the equilibrium concentration at the oxidizing temperature. According to Fick's law, the concentration gradient produces an oxygen flux of

$$j = -D \frac{dc}{dx} = D \frac{c_s - c_e}{l}, \quad (3.46)$$

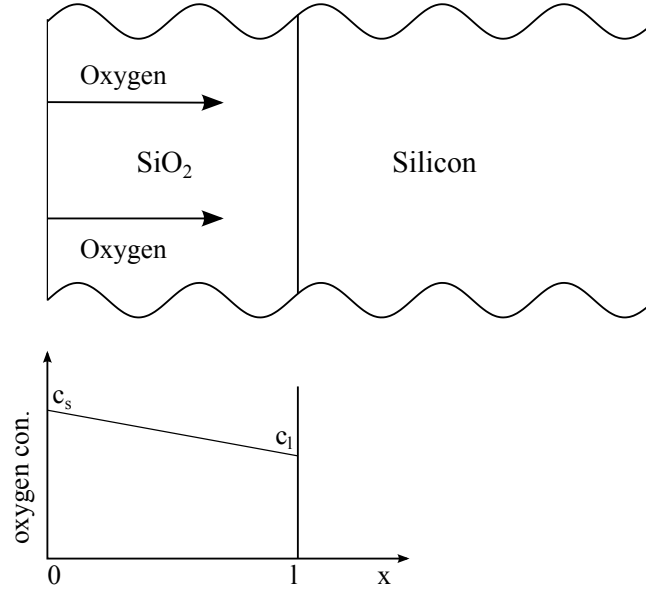
where  $c_s$  and  $c_e$  are excess concentrations of oxygen over the stoichiometric concentration  $c_{oxide}$ . It is then possible to relate the flux from the oxide layer growth  $j = c_{oxide}v$  with the flux from above and obtain a differential equation

$$c_{oxide} \frac{dl}{dt} = D \frac{c_s - c_e}{l}, \quad (3.47)$$

which is solved by

$$l = \left[ 2 \frac{c_s - c_e}{c_{oxide}} Dt + B \right]^{\frac{1}{2}}. \quad (3.48)$$

The parameter  $B$  can be determined by the initial condition. In order to gain the solution in equation (3.48), some simplifications like constant compositions  $c_s$ ,  $c_e$  and a fast oxidation reaction are needed. Also, the diffusion of excess oxygen in the silica layer is ignored. Although the solution has been started on top of a steady state approach to derive an interface movement, the solution is in good agreement with experiments. A diagram of the oxygen diffusion example is shown in figure 3.4.



**Figure 3.4:** Oxygen diffusion into silicon

### Thin layers

In thin layers, the diffusion flux is very high, so instead, the combination of oxygen and silicon limits the whole process. In the Deal-Grove model [DG65], a reaction term proportional to the excess composition is introduced to the equations at the interface. The assumption of an interface composition being identical to the equilibrium concentration is not valid any more. Using a steady state approach, the reaction rate  $k_x(c_i - c_e)$  must be equal to the atom flux to the interface  $D \frac{c_s - c_i}{l}$ , leading to a self regulated system

$$k_s(c_i - c_e) = D \frac{c_s - c_i}{l}, \quad (3.49)$$

where an overflow of diffusion flux will decrease the interface composition, leading to a higher reaction rate, which in turn will decrease the diffusion flux. Rearranging (3.49) and combining with the layer growth rate of (3.47), to eliminate  $c_i$ , yields

$$\frac{dl}{dt} = \frac{k_s(c_s - c_e)}{c_{oxide} \left( \frac{1 + k_s l}{D} \right)}, \quad (3.50)$$

which is solved by

$$l^2 + \frac{2D}{k_s} l = \frac{2D(c_s - c_e)}{c_{oxide}} (t + \tau). \quad (3.51)$$

The constant  $\tau$  can be used to define an initial layer thickness at  $t = 0$  and the coefficients of  $l$  and  $(t + \tau)$  are called  $A$  and  $B$  respectively. On long time scales,  $l$  will grow large, and the linear term can be neglected compared to the quadratic one, what gives the same result as in the previous model. At early times in the reaction limited regime, with a small oxygen layer, the linear term is significant, and the growth rate shows a proportionality to  $B/A$ . The temperature dependence of the constants  $B$  and  $B/A$  is of a Boltzmann form

$$C \exp\left(\frac{-E}{k_B T}\right), \quad (3.52)$$

where the constants  $C$  and  $E$  are to be determined experimentally.

### Neumann type similarity solution

Similarity solutions are functions that solve rather simple moving boundary problems, with the property of having one argument dependent on the square root of time, usually in the form of  $x/\sqrt{t}$  or  $r/\sqrt{t}$  in spherical coordinates. Franz Neumann presented his solution [RW12] to such a problem already in the 1860s. At the interface, there occurs both a jump of concentration, where a finite value of  $c_i$  is assumed, and of the diffusion gradient. The medium, in which the diffusion occurs, is assumed to be semi-infinite. On the surface at  $x = 0$ , a constant composition  $c_1$  is maintained. On the left side of the interface, a diffusion coefficient of  $D_1$  is used whereas on the right side, it is called  $D_2$ . The same also applies for the concentrations  $c_i$ .  $S(t)$  denotes the position of the interface in the sample. Boundary conditions at the interface are

$$\left. \begin{aligned} c_1 &= c_2 = C_i, \\ D_1 \frac{\partial c_1}{\partial x} &= D_2 \frac{\partial c_2}{\partial x}, \end{aligned} \right\} \quad x = S. \quad (3.53)$$

At the beginning of the sample and at infinity, the fixed compositions  $C_1$  and  $C_2$  are applied, as shown below:

$$\begin{aligned} c_1 &= C_1, \quad x = 0, \\ c_2 &= C_2, \quad x = \infty. \end{aligned} \quad (3.54)$$

After combining equations (3.54) and the diffusion equations for both regions

$$\frac{\partial c_1}{\partial t} = D_1 \frac{\partial^2 c_1}{\partial x^2}, \quad 0 < x < S, \quad (3.55)$$

$$\frac{\partial c_2}{\partial t} = D_2 \frac{\partial^2 c_2}{\partial x^2}, \quad x > S, \quad (3.56)$$

one can find particular solutions

$$c_1 = C_1 + A \operatorname{erf} \frac{x}{2\sqrt{D_1 t}}, \quad (3.57)$$

$$c_2 = C_2 + B \operatorname{erfc} \frac{x}{2\sqrt{D_2 t}}, \quad (3.58)$$

where  $\operatorname{erf}$  is the error function, and  $\operatorname{erfc} = 1 - \operatorname{erf}$  is the conjugated error function. Both solutions must satisfy the first relation from (3.53) at the interface, which requires

$$A \operatorname{erf} \frac{S}{2\sqrt{D_1 t}} = C_i - C_1, \quad (3.59)$$

and

$$B \operatorname{erfc} \frac{S}{2\sqrt{D_2 t}} = C_i - C_2. \quad (3.60)$$

This can only be fulfilled, if the interface position is proportional to the square root of time, like

$$S = k\sqrt{t}. \quad (3.61)$$

Differentiating equations (3.57) and (3.58), rephrasing (3.59) and (3.60) to  $A$  and  $B$ , it is possible, by inserting into the second part of (3.53), to obtain the relation

$$\frac{C_i - C_1}{g\left(\frac{k}{2\sqrt{D_1}}\right)} + \frac{C_i - C_2}{f\left(\frac{k}{2\sqrt{D_2}}\right)} = 0, \quad (3.62)$$

which can be solved numerically to gain  $k$  in dependence of the constants  $C_1, C_2, C_i, D_1$  and  $D_2$ . The functions  $f$  and  $g$  are given by

$$g\left(\frac{k}{2\sqrt{D_1}}\right) = \sqrt{\pi} \frac{k}{2\sqrt{D_1}} \exp\left(\frac{k^2}{4D_1}\right) \operatorname{erf} \frac{k}{2\sqrt{D_1}}, \quad (3.63)$$

$$f\left(\frac{k}{2\sqrt{D_2}}\right) = \sqrt{\pi} \frac{k}{2\sqrt{D_2}} \exp\left(\frac{k^2}{4D_2}\right) \operatorname{erfc} \frac{k}{2\sqrt{D_2}}. \quad (3.64)$$

After  $k$  is determined, one can substitute  $k = S/\sqrt{t}$  into (3.59) and (3.60) to calculate values for  $A$  and  $B$ . Using both constants, the composition distribution can be evaluated from (3.57) and (3.58). The condition of a semi-infinite medium is only satisfied at early stages of diffusion into a finite sample.

### Solution for phase change

In this example discussed by Jackson [TJRC53], a crystallization process is described, where the convective mixing in the liquid phase is omitted. The distribution of the

diffusing substance in the liquid phase is assumed to be diffusive only, but on the other hand, diffusion in the solid is neglected. The solid-liquid interface moves with the velocity  $v$  towards the liquid phase. This moving boundary problem is solved through a variable transformation into a coordinate system, which moves with the interface. After a new transformation variable  $z = x - vt$  is defined, the partial derivatives of the diffusion equation (3.43) are replaced through

$$\frac{\partial^2 c}{\partial x^2} = \frac{\partial^2 c}{\partial z^2}, \quad \frac{\partial c}{\partial t} = -v \frac{\partial c}{\partial z}, \quad (3.65)$$

which in turn changes the diffusion equation to an ordinary differential equation

$$D \frac{d^2 c}{dz^2} + v \frac{dc}{dz} = 0, \quad (3.66)$$

being only dependent on the variable  $z$ , with the exact solution

$$c = A \exp\left(-\frac{vz}{D}\right) + B. \quad (3.67)$$

The starting composition far from the interface fixes the constant  $B$  to a composition value of  $c_\infty$ . A conservation condition at the interface

$$v(c_l - c_s)_I = -D \left( \frac{dc}{dz} \right)_I, \quad (3.68)$$

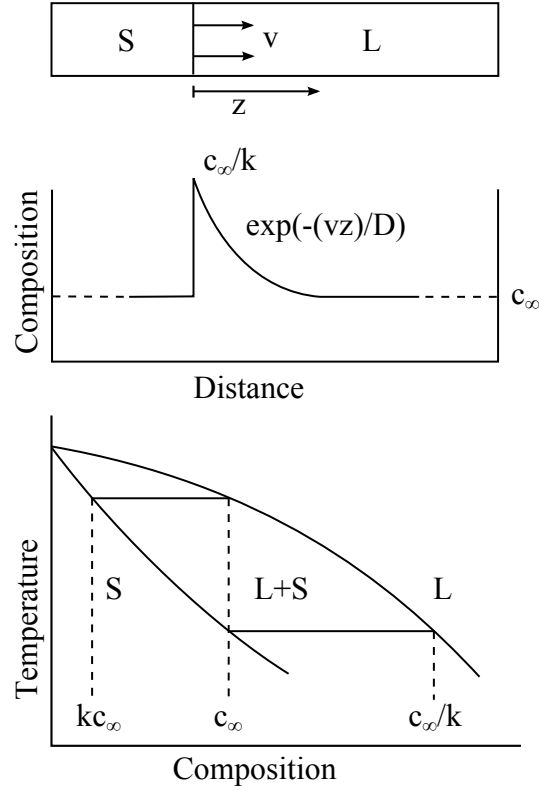
where  $c_l$  and  $c_s$  are the compositions of the solid and liquid phase, respectively, is used to determine the constant  $A$ , by calculating  $dc/dz$  from equation (3.67), to  $A = c_l - c_s$ . Using the definition  $k \equiv c_s/c_l$ , it is possible to rephrase the solution for the composition to

$$c = c_\infty \left( \frac{1}{k} - 1 \right) \exp\left(-\frac{vz}{D}\right) + c_\infty. \quad (3.69)$$

An overview of the situation given in this example can be found in figure 3.5.

### 3.2.3 Front tracking methods

Front tracking methods have the characteristic that, at each time step, the position of the boundary is calculated. The spacial discretisation can either be done using finite difference methods, which are relatively simple to implement, or finite element methods, which are more suitable for non-rectangular bodies in multi-dimensional problems. When a fixed grid is used, the interface will generally be located between two grid points, as shown in figure 3.6. For this reason, derivatives in the neighborhood of the boundary have to be adapted to deal with unequal space intervals and approximate



**Figure 3.5:** Moving boundary problem with diffusion in liquid phase only

the values at the interface. The other method is, to use a deformed grid or transform the variables in such a way that the interface resides always on a grid line or it stays fixed in the transformation domain.

#### Fixed finite difference grid

To incorporate a moving boundary model on a fixed grid into the diffusion equation, a three-point, finite difference formula, of Lagrangian type, can be used in the vicinity of the interface [Cra57]. A function  $f(x)$  can be approximated, by using known values of  $f$  at the fixed points  $a_0$ ,  $a_1$  and  $a_2$  through the sum

$$f(x) = \sum_{j=0}^2 l_j(x) f(a_j), \quad (3.70)$$

with the Lagrange polynomials

$$l_j(x) = \frac{p_2(x)}{(x - a_j)p_2'(a_j)}, \quad p_2(x) = (x - a_0)(x - a_1)(x - a_2), \quad (3.71)$$

where  $p'_2(a_j)$  is the spacial derivative. The first and second derivatives of the function change to

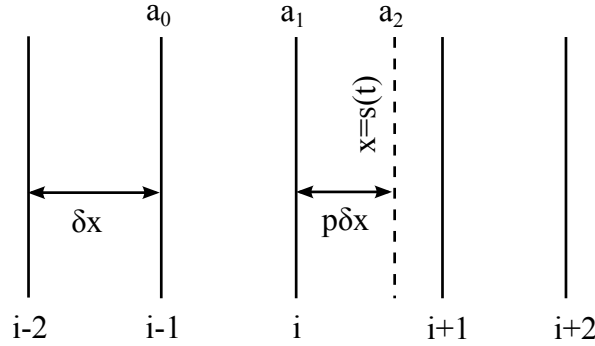
$$\frac{df}{dx} = l'_0(x)f(a_0) + l'_1(x)f(a_1) + l'_2(x)f(a_2), \quad (3.72)$$

$$\frac{1}{2} \frac{d^2f}{dx^2} = \frac{f(a_0)}{(a_0 - a_1)(a_0 - a_2)} + \frac{f(a_1)}{(a_1 - a_2)(a_1 - a_0)} + \frac{f(a_2)}{(a_2 - a_0)(a_2 - a_1)}, \quad (3.73)$$

with the first derivative of a Lagrange polynomial being

$$l'_0(x) = \frac{(x - a_1)(x - a_2)}{(a_0 - a_1)(a_0 - a_2)}. \quad (3.74)$$

The fixed points  $a_0$  to  $a_2$  correspond to the grid lines  $i - 1$ ,  $i$  and the position of



**Figure 3.6:** Approximation of values on a fixed grid

the boundary, whereas for a heat conduction problem, the function  $f$  is replaced by the temperature distribution. Inserting the interpolation into the derivatives, one can obtain

$$\frac{\partial^2 u}{\partial x^2} = \frac{2}{(\delta x)^2} \left( \frac{u_{i-1}}{p+1} - \frac{u_i}{p} + \frac{u_B}{p(p+1)} \right), \quad x = i\delta x \quad (3.75)$$

$$\frac{\partial u}{\partial x} = \frac{1}{\delta x} \left( \frac{p u_{i-1}}{p+1} - \frac{(p+1)u_i}{p} + \frac{(2p+1)u_B}{p(p+1)} \right), \quad x = s(t), \quad (3.76)$$

for positions between  $a_0$  and  $a_1$  and

$$\frac{\partial^2 u}{\partial x^2} = \frac{2}{(\delta x)^2} \left( \frac{u_B}{(1-p)(2-p)} - \frac{u_{i+1}}{1-p} + \frac{u_{i+2}}{2-p} \right), \quad x = (i+1)\delta x \quad (3.77)$$

$$\frac{\partial u}{\partial x} = \frac{1}{\delta x} \left( \frac{(2p-3)u_B}{(1-p)(2-p)} + \frac{(2-p)u_{i+1}}{1-p} - \frac{(1-p)u_{i+2}}{2-p} \right), \quad x = s(t), \quad (3.78)$$



for positions between  $a_1$  and  $a_2$ , where  $u_B$  is the temperature at the interface,  $\delta x$  is the spacing of the fixed grid and  $p$  is a fraction in the interval zero to one, denoting the position of the interface between the grid points  $i$  and  $i + 1$ . Derivatives on other points than  $i$  and  $i + 1$  are calculated using the standard finite difference formula, thus, the temperature at the next time step can be expressed as

$$u_{n,j+1} = u_{n,j} + \frac{\delta t}{(\delta x)^2}(u_{n-1,j} - 2u_{n,j} + u_{n+1,j}), \quad n = 1, 2, \dots, i - 1. \quad (3.79)$$

Using the boundary conditions  $u_{0j} = 1$  and  $u_B = 0$ , a similar equation can be derived for the grid point  $i$ , with the result

$$u_{i,j+1} = u_{i,j} + \frac{2\delta t}{(\delta x)^2} \left( \frac{u_{i-1,j}}{p_j + 1} - \frac{u_{i,j}}{p_j} \right). \quad (3.80)$$

The new fraction  $p$  is formulated, using equation (3.25), as

$$p_{j+1} = p_j - \frac{\delta t}{\lambda(\delta x)^2} \left( \frac{p_j u_{i-1,j}}{p_j + 1} - \frac{(p_j + 1)u_{i,j}}{p_j} \right). \quad (3.81)$$

To simulate a moving boundary problem with this method, the calculation steps are performed in the following order:

1. Calculate new temperatures for all grid points except for the interface point  $i$ .
2. Calculate new temperature for the grid point, next to the interface,  $i$ .
3. Calculate new position fraction  $p$ .
4. Repeat previous steps using the resulting values of the previous round to gather further values and adapt the interface grid point if  $p$  exceeds unity.

### Variable time step

In 1955, Douglas and Gallie [DG55] suggested, to use variable time steps in such a way, that the interface position matches a grid line at each step. Their system of equations

is

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < s(t), \quad t > 0, \quad (3.82)$$

$$\frac{\partial u}{\partial x}(0, t) = -1, \quad t > 0, \quad (3.83)$$

$$\frac{\partial x}{\partial t}(s(t), t) = -\frac{\partial u}{\partial x}, \quad u(s(t), t) = 0, \quad (3.84)$$

$$s(0) = 0. \quad (3.85)$$

They integrate the left part of equation (3.84) and apply the boundary conditions to gain

$$s(t) = t - \int_0^{s(t)} u(x, t) dx. \quad (3.86)$$

Using the differential notation

$$x_i = i\Delta x, \quad t_n = \sum_{k=0}^{n-1} \Delta t_k, \quad f_{i,n} = f(x_i, t_n), \quad (3.87)$$

and the known values  $\Delta t_k$  and  $u_{i,k}$ , for selected  $\Delta t_n^{(0)}$ , a temperature  $u_{i,n+1}^{(0)}$  is chosen as solution for the difference representation of the diffusion equation

$$\frac{u_{i-1,n+1}^{(r)} - 2u_{i,n+1}^{(r)} + u_{i+1,n+1}^{(r)}}{(\Delta x)^2} = \frac{u_{i,n+1}^{(r)} - u_{i,n}^{(r)}}{\Delta t_n^{(r)}}, \quad i = 1, \dots, n, \quad t \geq 0. \quad (3.88)$$

Combined with the boundary condition (3.83) in differential form

$$u_{0,n+1}^{(r)} - u_{1,n+1}^{(r)} = \Delta x, \quad (3.89)$$

the time steps can be improved iteratively as

$$\Delta t_n^{(r+1)} = \left( n + 1 + \sum_{i=1}^n u_{i,n+1}^{(r)} \right) \Delta x - t_n. \quad (3.90)$$

Convergence and stability were established by this implicit scheme of Douglas and Gallie.

### Variable space grid

Murray and Landis [ML59] introduced a variable spaced grid, to keep the interface at the same grid line for all times. With advancing time, the grid intervals are stretched

on one side of the boundary and pinched on the other side. Through the total time derivative of the temperature

$$\frac{du}{dt} = \frac{\partial u}{\partial x} \frac{dx}{dt} + \frac{\partial u}{\partial t}, \quad (3.91)$$

they are able to include a relation for the grid movement into the heat equation (3.23), which, by inserting the total derivative, can be written as

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} \frac{dx}{dt} + \frac{\partial^2 u}{\partial x^2}. \quad (3.92)$$

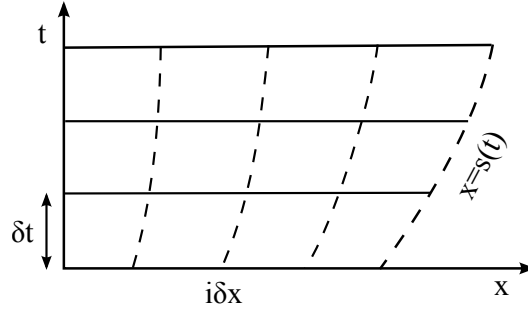
The grid movement can be related to the interface position using

$$\frac{dx_i}{dt} = \frac{x_i}{s(t)} \frac{ds}{dt}, \quad (3.93)$$

and the diffusion equation becomes

$$\frac{\partial u}{\partial t} = \frac{x}{s(t)} \frac{ds}{dt} \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}. \quad (3.94)$$

At each time step, the interface position  $s(t)$  must be updated in the computation of the diffusion equation, for example being updated from the boundary condition (3.25) at the interface. A picture of a Murray-Landis grid deformation caused by a moving interface is shown in figure 3.7.



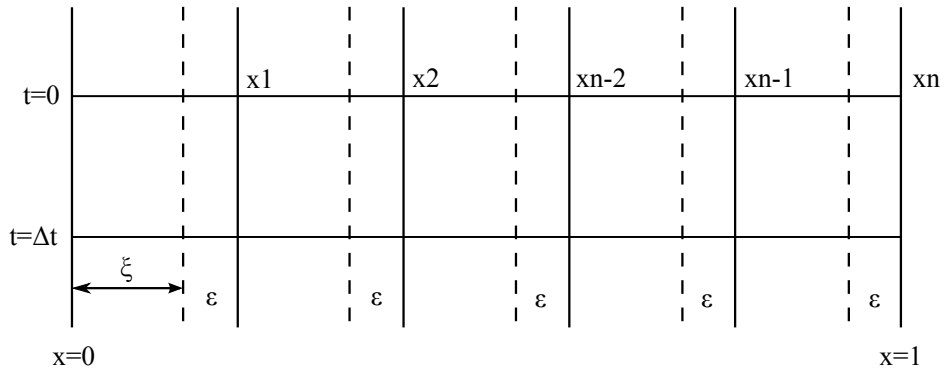
**Figure 3.7:** Deformation of a one dimensional grid according to Murray Landis method

Crank and Gupta [CG72] moved the whole system with the velocity of the moving boundary, which was located at the rightmost side of the sample, being  $x = 1$ . When the interface moves a distance of  $\varepsilon$  to the left, all grid lines are also displaced by the same distance. The result is only one unequal interval at the beginning of the sample at  $x = 0$ . In the first step, an approximation of the temperature next to the interface

is calculated explicitly. Using this value, the new interface position is calculated  $(1 - \varepsilon)$  and the grid is displaced accordingly. At the new grid points, new temperature values are interpolated, either by using cubic splines or polynomials. Since all grid points move with the interface velocity, the enhanced diffusion equation becomes

$$\frac{\partial u}{\partial t} = \frac{ds}{dt} \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2}. \quad (3.95)$$

The deformation of a one dimensional grid using the method presented by Crank and Gupta is shown in figure 3.8.



**Figure 3.8:** Deformation of a one dimensional grid according to Crank Gupta method

### Finite element methods

Finite element methods are most suitable for more dimensional systems and complex geometries, where the elements can be fitted to the moving interface smoothly. In the regions near the interface, the mesh will usually be denser than on other parts of the sample. Formulating a problem in form of a finite element model requires the following steps:

1. Rephrase the differential equation in a variational form as an integral equation.
2. Discretize the sample in a way, that the borders are well fitted by the elements. Such elements can, for example, be triangles (two dimensions) or boxes (three dimensions).
3. Choose an appropriate basis, so that many of the coefficients in the equation vanish.
4. A sparse coefficient matrix is established, which can be solved with LU decomposition for matrices of standard problems, but complex systems need more sophisticated algorithms.

To improve the efficiency of FE methods, there is either the possibility of decreasing the diameter of the elements in the mesh (h-methods), or of using higher order polynomials for the basis functions (p-methods).

A one dimensional heat conduction problem was solved by Bonnerot and Jamet in 1979 [BJ79]. The problem was formulated as

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad (3.96)$$

$$u(x,0) = u_0(x), \quad 0 \leq x \leq s_0, \quad (3.97)$$

$$u(0,t) = g(t), \quad 0 < t \leq T, \quad (3.98)$$

$$u(s(t),t) = 0, \quad 0 < t \leq T, \quad (3.99)$$

$$\frac{ds}{dt} = -c \frac{\partial u}{\partial x}, \quad x = s(t), \quad 0 < t \leq T, \quad (3.100)$$

$$s(0) = s_0, \quad (3.101)$$

where  $s(t)$  denotes the position of the interface, with the initial value  $s_0$  and the initial temperature distribution  $u_0$ . Writing the differential equation in variational form leads to

$$\begin{aligned} & - \int_{t^n}^{t^{n+1}} \int_0^{s(t)} u \frac{\partial \phi}{\partial t} dx dt + \int_{t^n}^{t^{n+1}} \int_0^{s(t)} \frac{\partial u}{\partial x} \frac{\partial \phi}{\partial x} dx dt \\ & + \int_0^{s(t^{n+1})} u(x, t^{n+1}) \phi(x, t^{n+1}) dx - \int_0^{s(t^n)} u(x, t^n) \phi(x, t^n) dx = 0, \end{aligned} \quad (3.102)$$

with the test function  $\phi$  being introduced during the integration. The first two terms occur during derivation through a partial integration and the whole equation relates to a time strip between  $t^n$  and  $t^{n+1}$ . In the integration process the boundary terms are omitted, because the test function vanishes at the boundaries. Unknown values of the temperature (at time step  $t^{n+1}$ ) appear in the first three parts of (3.102), whereas the known values appear in the last term. The time strips are split into biquadratic elements with the defining points  $P_{i+\mu}^{n+\nu}$ , where  $i$  and  $n$  are integers and  $\mu, \nu$  can take the values 0, 1/2 and 1. Such a point  $P_{i+\mu}^{n+\nu}$  is located at position  $x_{i+\mu}^{n+\nu}$  and time  $t^{n+\nu}$ , as shown in figure 3.9. All points located at the interface have the subindex  $I$ . Two additional restrictions for spacial discretization are

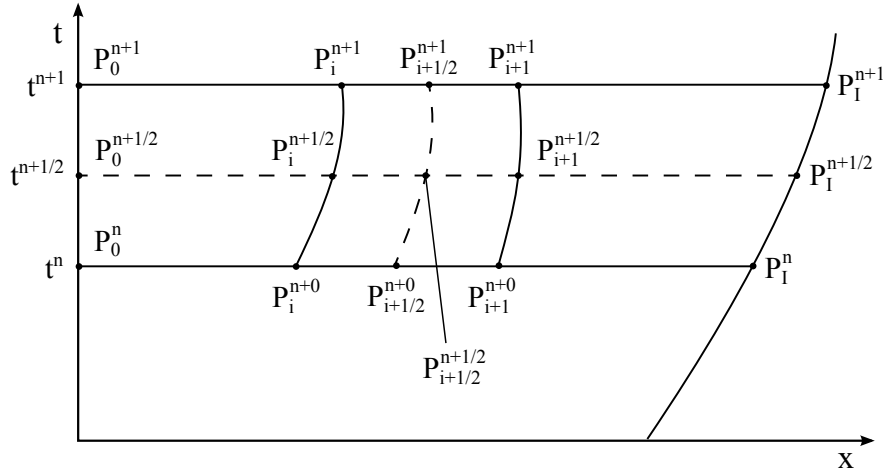
$$x_I^{n+\nu} = s(t^{n+\nu}), \quad x_{i+1/2}^{n+\nu} = \frac{1}{2}(x_i^{n+\nu} + x_{i+1}^{n+\nu}). \quad (3.103)$$

It is possible to vary the position of points and also the number of elements independently for each strip. Bonnerot and James used Simpson quadrature for evaluating

the integrals on each element and obtained a square matrix. They also approximated the spacial gradient of the temperature by using a quadratic interpolation of the three values at the positions  $\mu = 0, 1/2, 1$ . This led them to a discretization of the integral form of (3.100), namely

$$s^{n+\nu} = s^n - c \int_{t^n}^{t^{n+\nu}} (Du)_h(t) dt, \quad (3.104)$$

where  $D$  denotes the differential operator and  $h$  shows that the derivative is discretized. Since the values of the integral term depend on the temperature in the sample, an iterative procedure is needed. The biquadratic finite element strip with a visualization of all variables is shown in figure 3.9.



**Figure 3.9:** Finite element discretization of a one dimensional grid according to Bonnerot and James

### 3.2.4 Front fixing methods

In these methods, the interface is kept at a constant position, by applying a coordinate transformation like

$$\xi = x/s(t), \quad (3.105)$$

which fixes the boundary at  $\xi = 1$  throughout the calculation. Using the relations

$$\frac{\partial u}{\partial x} = \frac{1}{s(t)} \frac{\partial u}{\partial \xi}, \quad \frac{\partial^2 u}{\partial x^2} = \frac{1}{[s(t)]^2} \frac{\partial^2 u}{\partial \xi^2}, \quad (3.106)$$

$$\begin{aligned} \left( \frac{\partial u(x,t)}{\partial t} \right)_x &= \left( \frac{\partial u(\xi,t)}{\partial t} \right)_x = \frac{\partial u}{\partial \xi} \frac{\partial \xi}{\partial t} + \left( \frac{\partial u(\xi,t)}{\partial t} \right)_\xi \\ &= -\frac{x}{[s(t)]^2} \frac{ds}{dt} \frac{\partial u}{\partial \xi} + \left( \frac{\partial u(\xi,t)}{\partial t} \right)_\xi, \end{aligned} \quad (3.107)$$

the heat equation can be transformed to

$$\frac{\partial^2 u}{\partial \xi^2} = s^2 \frac{\partial u}{\partial t} - s\xi \frac{ds}{dt} \frac{\partial u}{\partial \xi}, \quad 0 < \xi < 1, \quad t > 0. \quad (3.108)$$

Taking the closure condition at the interface, similar to equation (3.25), a new condition

$$-\frac{1}{s} \frac{\partial u}{\partial \xi} = \frac{ds}{dt}, \quad \xi = 1, \quad t > 0 \quad (3.109)$$

in transformed variables can be obtained, in which the Stefan number is set to one. Landau proposed this method in 1950 [Lan50] and Crank applied a finite difference scheme to it in 1957 [Cra57]. Lotkin improved the accuracy by using variable intervals in  $\xi$  and  $t$  and a divided differences method for evaluation [Lot60].

Elliott [Ell80] wrote about a possibility of discretizing only the spacial variables and integrating the ordinary differential equation over constant  $\xi$  lines, which approximates equation (3.108) by

$$\begin{aligned} \frac{du_j}{dt} &= \frac{1}{s^2} \frac{u_{j-1} - 2u_j + u_{j+1}}{(\Delta\xi)^2} + \xi_j \frac{\dot{s}}{s} \frac{(u_{j+1} - u_{j-1})}{2\Delta\xi}, \\ j &= 0, 1, \dots, N-1, \quad 0 < \xi < 1, \quad t > 0. \end{aligned} \quad (3.110)$$

### Body fitted coordinates

A generalization of the one-dimensional example above is the transformation of a curved-shaped region into a rectangular region, hence the name of the method. Curved lines of the source region are mapped to straight lines of the image region. The same applies to a moving boundary problem, where a time variable region corresponds to a static region in the fixed coordinates. The advantage of having a fixed rectangular working region comes at the price of higher complexity of the transformed equations and the additional work of meshing between the time steps is also not negligible. For the purpose of modeling moving boundary problems, often non-orthogonal transformations

are used, which are easily shown using the example of the Laplace equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0, \quad (3.111)$$

with the general variable  $\phi = \phi(x, y) = \phi(\xi, \eta)$ . The differential relations between the coordinate systems are

$$\xi_x = y_\eta / J, \quad \xi_y = -x_\eta / J, \quad (3.112)$$

$$\eta_x = -y_\xi / J, \quad \eta_y = x_\xi / J, \quad (3.113)$$

with the subindices denoting the derivatives with respect to the corresponding variable and  $J$  being the Jacobian determinant  $x_\xi y_\eta - x_\eta y_\xi \neq 0$ . The differential relations can be obtained by rewriting the differentials

$$dx = x_\xi d\xi + x_\eta d\eta, \quad dy = y_\xi d\xi + y_\eta d\eta, \quad (3.114)$$

in terms of  $d\xi$  and  $d\eta$ , which yields, for example,

$$d\xi = \frac{y_\eta dx - x_\eta dy}{x_\xi y_\eta - x_\eta y_\xi} = \frac{y_\eta dx - x_\eta dy}{J}. \quad (3.115)$$

After transforming the first derivative

$$\frac{\partial \phi}{\partial x} = \phi_\xi \xi_x + \phi_\eta \eta_x = (\phi_\xi y_\eta - \phi_\eta y_\xi) / J \quad (3.116)$$

and the second derivative

$$\begin{aligned} \frac{\partial^2 \phi}{\partial x^2} &= \frac{\partial}{\partial x} \left( \frac{\partial \phi}{\partial x} \right) = \frac{1}{J^2} \left( \frac{\partial}{\partial \xi} y_\eta - \frac{\partial}{\partial \eta} y_\xi \right)^2 \phi \\ &= (\phi_{\xi\xi} y_\eta^2 + \phi_{\eta\eta} y_\xi^2 - 2\phi_{\xi\eta} y_\eta y_\xi) / J^2, \end{aligned} \quad (3.117)$$

the Laplace equation can be written as

$$A\phi_{\xi\xi} + B\phi_{\xi\eta} + C\phi_{\eta\xi} + D\phi_\xi + E\phi_\eta = 0, \quad (3.118)$$



with the variables

$$A = \xi_x^2 + \xi_y^2 = (x_\eta^2 + y_\eta^2)/J^2, \quad (3.119)$$

$$B = 2(\xi_x \eta_x + \xi_y \eta_y) = -2(x_\xi x_\eta + y_\xi y_\eta)/J^2, \quad (3.120)$$

$$C = \eta_x^2 + \eta_y^2 = (x_\xi^2 + y_\xi^2)/J^2, \quad (3.121)$$

$$D = \xi_{xx} + \xi_{yy}, \quad E = \eta_{xx} + \eta_{yy}. \quad (3.122)$$

Additionally, there exists a transformation of the normal derivative of  $\phi$  on a boundary  $y = s(x)$ ,

$$\phi_n = \frac{s' \phi_x - \phi_y}{[1 + (s')^2]^{\frac{1}{2}}} = \frac{1}{J [1 + (s')^2]^{\frac{1}{2}}} \times [\phi_\xi (s' y_\eta + x_\eta) - \phi_\eta (s' y_\xi + x_\xi)], \quad (3.123)$$

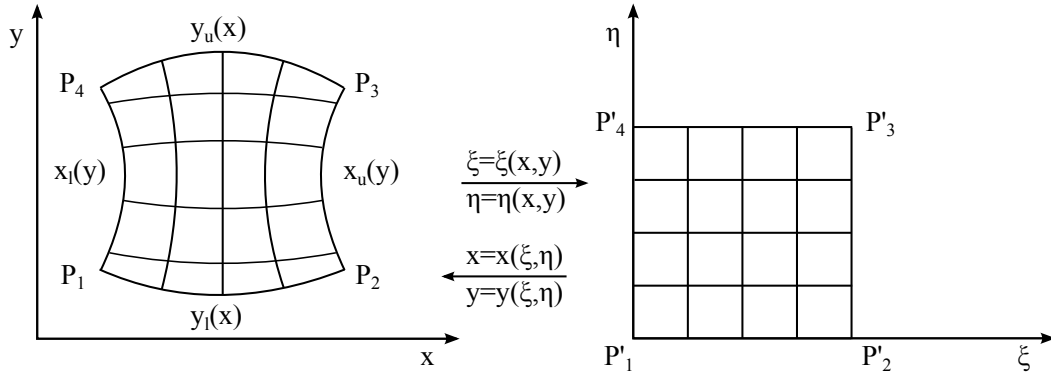
and a similar relation of the time derivative

$$(\phi_t)_{x,y} = (\phi_t)_{\xi,\eta} - \frac{1}{J} (y_\eta \phi_\xi - y_\xi \phi_\eta) (x_t)_{\xi,\eta} - \frac{1}{J} (x_\xi \phi_\eta - x_\eta \phi_\xi) (x_t)_{\xi,\eta}, \quad (3.124)$$

which maps the time derivative at a fixed point in the  $x/y$ -plane to a fixed point in the  $\xi/\eta$ -plane. In 1976, Oberkampf [Obe76] suggested the simple transformation

$$\xi = \frac{x - x_l(y)}{x_u(y) - x_l(y)}, \quad \eta = \frac{y - y_l(x)}{y_u(x) - y_l(x)}, \quad (3.125)$$

for moving interface problems, where the four variables  $x_l, x_u, y_l, y_u$  correspond to the four boundaries of the deformed region, as shown in figure 3.10.



**Figure 3.10:** Example of a grid transformation using body fitted coordinates

Furzeland [Fur77] applied the transformation to a two dimensional problem with one phase, and by using the transformations  $\xi = x$ ,  $\eta = \eta(x, y)$  in conjunction with

equations (3.118) and (3.124), he obtained a form of the two dimensional heat equation

$$u_t = Au_{\xi\xi} + Bu_{\xi\eta} + Cu_{\eta\eta} + D'u_\xi + E'u_\eta, \quad 0 \leq \xi \leq 1, \quad 0 \leq \eta \leq 1, \quad t > 0, \quad (3.126)$$

where the variables  $A$  to  $C$  are defined as before and the new definitions

$$D' = D + (x_\eta y_t - y_\eta x_t)/J, \quad E' = E + (y_\xi x_t - x_\xi y_t)/J \quad (3.127)$$

are introduced. After the application of the proper Oberkampf transformations

$$\xi = x, \quad \eta = y/s(x,t), \quad (3.128)$$

the heat equation is rephrased as

$$u_t = au_{\xi\xi} + bu_{\xi\eta} + cu_{\eta\eta} + du_\xi + eu_\eta, \quad 0 \leq \xi \leq 1, \quad 0 \leq \eta \leq 1, \quad t > 0 \quad (3.129)$$

and the coefficients change to

$$a = 1, \quad b = -2ys_\xi/s^2, \quad c = (1/s)^2 + (b/2)^2, \quad (3.130)$$

$$d = 0, \quad e = (y/s)_{\xi\xi} + y_t/s, \quad (3.131)$$

where the relations  $y_\eta = x$ ,  $y_\xi = ys_\xi/s$ , which follow from the transformation, were used. On the left and right sample boundary, the following derivative condition holds:

$$su_\xi - (ys_\xi u_\eta/s) = 0, \quad \xi = 0,1, \quad (3.132)$$

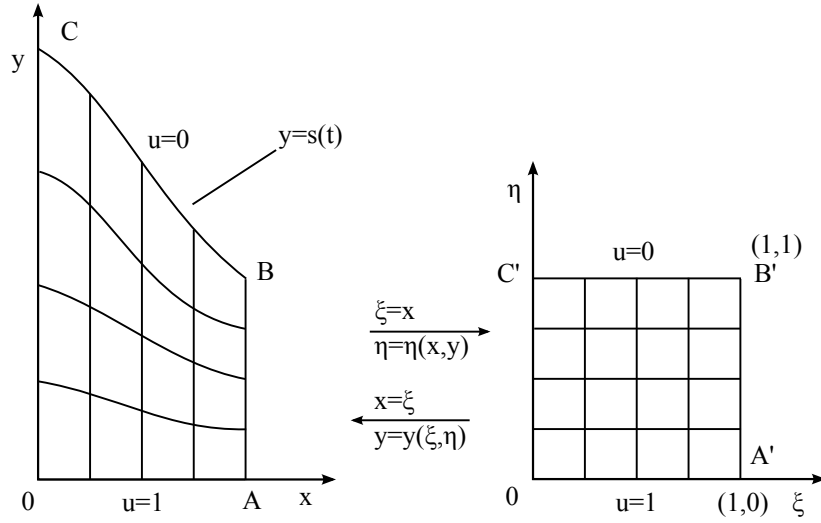
whereas on the moving interface,

$$u = 0, \quad y_t = -(1 + s_\xi^2)u_\eta/s, \quad \eta = 1 \quad (3.133)$$

must be considered. The following steps are essential for the computation:

1. At the beginning with given  $u$  and  $s$ , calculate the values of  $x, y$  for each mesh point of  $\xi, \eta$  (equation (3.128)).
2. Compute the new boundary position using a discretized form of equation (3.133).
3. With the second Oberkampf equation, determine the changes of mesh points caused by the movement.
4. Discretize time and space variables of the heat equation (3.129) and, after calculating a solution, continue with the second step at the next time level.

Figure 3.11 shows the deformed and the static grid for the Furzeland problem.



**Figure 3.11:** Grid transformation with body fitted coordinates as used in the Furze-land example

### Isotherm migration method

The isotherm migration method [DC70] is a case of transformation, where the dependent variable, for example the temperature, is exchanged with one space variable. For a one dimensional heat-transfer problem, this would thus mean a transformation from  $u(x,t)$  to  $x(u,t)$ . Using this method, the movement of a fixed-temperature line can be traced. If the phase boundary is such an isotherm, then this method is suitable, because it tracks their variation. A one-phase problem is described by the following equations:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}; \quad \frac{ds}{dt} = -\lambda \frac{\partial u}{\partial x}, \quad u = 0, \quad x = s(t); \quad (3.134)$$

$$u = 1, \quad x = 0, \quad t > 0; \quad u = 0, \quad x > 0, \quad t = 0. \quad (3.135)$$

The partial derivatives change, after applying the transformation  $u(x,t) \rightarrow x(u,t)$  to

$$\frac{\partial u}{\partial x} = \left( \frac{\partial x}{\partial u} \right)^{-1}, \quad \left( \frac{\partial x}{\partial t} \right)_u = - \left( \frac{\partial u}{\partial t} \right) \left( \frac{\partial x}{\partial u} \right), \quad (3.136)$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x} \left( \frac{\partial x}{\partial u} \right)^{-1} = - \left( \frac{\partial^2 x}{\partial u^2} \right) \left( \frac{\partial x}{\partial u} \right)^{-3}, \quad (3.137)$$

where  $\left(\frac{\partial x}{\partial t}\right)_u$  describes the movement of the isotherm. Inserting the derivatives into the heat equation and into the boundary condition at the interface results in

$$\frac{\partial x}{\partial t} = \left(\frac{\partial x}{\partial u}\right)^{-2} \frac{\partial^2 x}{\partial u^2}, \quad 0 < u < 1, \quad t > 0, \quad (3.138)$$

$$\frac{ds}{dt} = -\lambda \left(\frac{\partial x}{\partial u}\right)^{-1}, \quad u = 0, \quad t > 0, \quad (3.139)$$

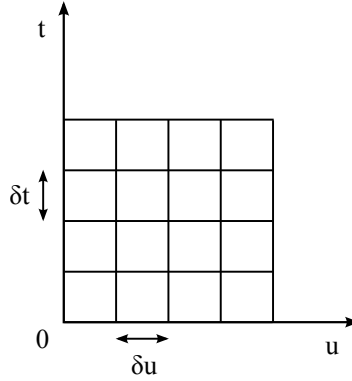
$$x = 0, \quad u = 1, \quad t > 0. \quad (3.140)$$

If the temperature change is too small at  $x = 0, t = 0$ , a starting solution is needed to overcome a possible singularity. Converting equations (3.138) and (3.139) to an explicit finite difference form leads to

$$x_i^{n+1} = x_i^n + 4\delta t \frac{(x_{i+1}^n - 2x_i^n + x_{i-1}^n)}{(x_{i-1}^n - x_{i+1}^n)^2}, \quad (3.141)$$

$$s^{n+1} = s^n - \frac{\lambda \delta t \delta u}{x_0^n - x_1^n}, \quad (3.142)$$

where  $i$ , despite the standard notation of being an index for the spacial discretization, now denotes a point in the isothermal grid that is shown in figure 3.12.



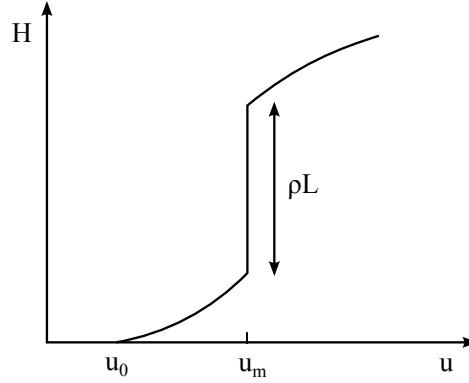
**Figure 3.12:** IMM transformed grid

### 3.2.5 Fixed domain methods

All methods in the previous chapters were front tracking methods and although the front fixing methods transform the boundary to a fixed position in the grid, the closure condition must still be satisfied there. There are cases, where it is impossible to track the interface motion directly, because it may not behave smoothly, especially in two and more space dimensions. Therefore, these methods use the concept of rephrasing

the equations to bind the interface condition to the whole sample. The position of the boundary then follows directly from the solution of the problem. A popular approach to achieve this, is to introduce an enthalpy function (see figure 3.13) that is valid on all areas of the specimen.

Already in 1946, Eyres et al. [EHI<sup>+</sup>46] proposed the use of such a function, which represents the whole heat content including specific and latent heat, for describing this type of problems. To integrate the interface condition in the system, the equations are formulated as follows:



**Figure 3.13:** Enthalpy function in the sample with a jump at the boundary

$$\begin{aligned}
 H(u) &= \int_{u_0}^u \rho(\theta)c(\theta) d\theta, \quad u < u_m, \\
 H(u) &= \int_{u_0}^u \rho(\theta)c(\theta) d\theta + \rho L, \quad u > u_m, \\
 \int_{u_0}^u \rho(\theta)c(\theta) d\theta &\leq H(u) \leq \int_{u_0}^u \rho(\theta)c(\theta) d\theta + \rho L, \quad u = u_m,
 \end{aligned} \tag{3.143}$$

which can be written as a single equation

$$H(u) = \int_{u_0}^u [\rho(\theta)c(\theta) + L\rho(\theta)\delta(\theta - u_m)] d\theta, \tag{3.144}$$

where  $u$  and  $\theta$  denote the temperature,  $\rho$  the density,  $c$  the specific heat,  $u_m$  the melting temperature and  $L$  the latent heat constant. The Dirac delta functional has the purpose, to limit the effect of the latent heat term only to the melting temperature.

As example, the multi dimensional problem

$$\rho_i c_i \frac{\partial u_i}{\partial t} = \nabla(K_i \nabla u_i), \quad i = 1, 2, \quad (3.145)$$

$$u_1 = u_2 = u_m, \quad \left[ K \frac{\partial u}{\partial n} \right]_1^2 = -\rho L v_n, \quad x = s(t), \quad (3.146)$$

with the phase index  $i$  and the interface velocity towards the normal direction  $v_n$ , can be related to the enthalpy by

$$\frac{\partial H}{\partial t} = \rho(u) c(u) \frac{\partial u}{\partial t}, \quad u \geq u_m, \quad (3.147)$$

which is a consequence of the first and second part of (3.143). Please note, that  $c, \rho$  and  $K$  can be functions of  $u, \vec{x}$  and  $t$ . Combining the temperature derivative of the enthalpy (3.147) and the heat equation (3.145) leads to the new description

$$\frac{\partial H(u)}{\partial t} = \nabla(K \nabla u). \quad (3.148)$$

The reason why this equation is ill-posed at the melting temperature is the jump discontinuity of the enthalpy at this point.

A modification, which is valid on the whole region, was proposed by Shamsundar and Sparrow [SS75], where they formulated the energy balance of a finite volume  $V$  and a surface area  $A$ . This led to the integral equation

$$\frac{d}{dt} \int_V H dV + \int_A H \vec{v} \cdot d\vec{A} = \int_A K \nabla u \cdot d\vec{A}, \quad (3.149)$$

which, in its second term denoted by the velocity  $\vec{v}$ , incorporates also the resulting fluid motion caused by density changes or convection. By subtracting the equation

$$\frac{d}{dt} \int_V H_1 dV + \int_A H_1 \vec{v} \cdot d\vec{A} = 0, \quad (3.150)$$

for a one-phase problem being initially at the melting temperature, from the integral equation (3.149), the fluid velocity  $v$  can be eliminated through the following steps:

$$\frac{d}{dt} \int_V (H - H_1) dV + \int_A (H - H_1) \vec{v} \cdot d\vec{A} = \int_A K \nabla u \cdot d\vec{A}, \quad (3.151)$$

$$\frac{d}{dt} \int_V (H - H_1) dV = \int_A K \nabla u \cdot d\vec{A}, \quad (3.152)$$

where from equation (3.151) to (3.152), a vanishing velocity of  $\vec{v} = 0$  is considered in the solid phase, which is possible, because after subtraction of the liquid-enthalpy, only the solid part is left. To summarize, (3.152) describes the enthalpy in a one phase system with variable density, but omits the usage of the fluid velocity. If the heat conductivity  $K$  depends only on the temperature, equation (3.148) can be simplified by a Kirchhoff transformation

$$v = \int^u K(\xi) d\xi, \quad (3.153)$$

to become

$$\frac{\partial H(v)}{\partial t} = \Delta v. \quad (3.154)$$

The equations for the enthalpy can either be solved by using an explicit or implicit method, after applying a space and time discretization. To determine the interface position, one has to inspect the variation of enthalpy and temperature over the whole domain.

## 4 Simulation and Results

For complex diffusion systems, there exist seldom analytic solutions, therefore, numerical algorithms are applied to solve such problems. Since digital computers can not cope with continuous, analog problems, the problem and solution have to be approximated at discrete points. The values of the solution at these points are stored using floating point variables, which are an approximation for real values. To improve the accuracy of the result, a denser set of data points can be used, which will in turn increase the processor and memory demands. A natural border for increasing the number of data points is the number of bits a floating point variable consumes, since not all decimal places of the real value is stored.

### 4.1 Diffusion calculation in MatCalc

MatCalc is able to calculate numerical problems of diffusion problems, but more generally, it is a software package for computer simulation of microstructural processes. MatCalc consists of three modules, which each are suitable for calculation of problems from different fields. The ‘core’ module can perform equilibrium calculations, handle different material properties and compute phase diagrams. The ‘monte’ module is able to simulate precipitation in multi-component, multi-phase environments by using a Monte-Carlo approach. In order to simulate diffusion in solids or heat conduction, the simulation module is appropriate. MatCalc is currently developed at the institute of materials science and technology and available for the most-used operating systems Windows, Linux and MacOS.

The main practical work of the thesis consisted of extending the MatCalc simulation module with an algorithm for simulating moving phase boundaries. Additionally, the explicit and trapezoidal discretization of the diffusion equation were implemented. Parts of the phase boundary algorithm were a modified version of the Murray Landis method, different time step-width limitations, two methods of describing the interface composition and heat treatment handling for phase transformations. Several examples were simulated to either demonstrate the behavior of moving interface in MatCalc, or verify the expected experimental results.

Prior to elaborating on the implementation of moving phase boundaries in MatCalc, the basics of the simulation module are introduced. In this module, a sample is separated into a grid consisting of a finite number of cells. Each of these cells has



its own physical properties like compositions, temperature and size. For the current implementation of MatCalc, a cell property applies to the whole cell. That means, there is no temperature or composition distribution in a cell, the whole property is uniform and can not change until a neighbor cell is reached. In order to solve diffusion problems numerically, the diffusion equation is discretized, both in spacial and time dimensions. The Laplace operator in Fick's second law is replaced by a second-order, finite difference representation and also the time derivation is approximated. Details about numerical derivatives can be found in 'Numerical Recipes: The Art of Scientific Computing' [PTVP07].

#### 4.1.1 Implementation

There are two different ways of parametrizing this operator's time dependence, that lead to either an implicit or explicit solution method, but also a combination of both is possible. A single component, one dimensional system will be used, to show the process of discretization, first in the explicit, then in the implicit form. Applying the implicit method to equation (3.6) leads to

$$\frac{c_i^{t+} - c_i^t}{\Delta t} = D \frac{c_{i+1}^{t+} - 2c_i^{t+} + c_{i-1}^{t+}}{(\Delta x)^2}. \quad (4.1)$$

The upper indices describe the position in time ( $t+$  is an abbreviation for  $t + \Delta t$ ) and the lower indices describe the cell index ( $i + 1$  and  $i - 1$  are the neighbor cells of the current cell), since in this process, space and time are both separated into discrete sections. Another simplification in this notation is that the time and space distances in the denominators are assumed to be equidistant, which is not necessarily the case, but must be taken into account when extending the right side of (4.1).

Regrouping all terms with index  $t+$  on the left side and multiplication with the denominator yields

$$[(\Delta x)^2 + 2D\Delta t]c_i^{t+} - D\Delta t(c_{i+1}^{t+} + c_{i-1}^{t+}) = (\Delta x)^2 c_i^t. \quad (4.2)$$

In contrast to the explicit method, the unknown neighbor compositions of the next time step are used on the right side of (4.1). Through division of equation (4.2) by  $(\Delta x)^2$ , the formulation

$$\left[1 + \frac{2D\Delta t}{(\Delta x)^2}\right] c_i^{t+} - \frac{D\Delta t}{(\Delta x)^2} (c_{i+1}^{t+} + c_{i-1}^{t+}) = c_i^t \quad (4.3)$$

is obtained. Using the abbreviation  $\alpha = \frac{D\Delta t}{(\Delta x)^2}$  and as simplification a one dimensional problem of only three cells, equation (4.3) can be written in the form of the matrix

equation

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 \\ -\alpha & 1+2\alpha & -\alpha \\ 0 & -\alpha & 1+2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_2^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t \\ c_2^t \end{pmatrix}. \quad (4.4)$$

Extension of this problem to  $N$  cells enlargens the matrix to size  $N \times N$  and also the solution vector and the vector on the right side scale up to  $N$  elements. To show what happens in multi dimensional cases, the following grid will be used:

$c_0$	$c_1$	$c_2$	$c_3$
$c_4$	$c_5$	$c_6$	$c_7$
$c_8$	$c_9$	$c_{10}$	$c_{11}$

Inserting the abbreviation  $\gamma = 1 + \sum_N \alpha_N$  into equation (4.4), the matrix extends to:

$$\begin{pmatrix} \gamma & -\alpha & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha & \gamma & -\alpha & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha & \gamma & -\alpha & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha & \gamma & 0 & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 \\ -\alpha & 0 & 0 & 0 & \gamma & -\alpha & 0 & 0 & -\alpha & 0 & 0 & 0 \\ 0 & -\alpha & 0 & 0 & -\alpha & \gamma & -\alpha & 0 & 0 & -\alpha & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 & -\alpha & \gamma & -\alpha & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & -\alpha & 0 & 0 & -\alpha & \gamma & 0 & 0 & 0 & -\alpha \\ 0 & 0 & 0 & 0 & -\alpha & 0 & 0 & 0 & \gamma & -\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & 0 & -\alpha & \gamma & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & 0 & -\alpha & \gamma & -\alpha \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & 0 & -\alpha & \gamma \end{pmatrix} \quad (4.5)$$

The  $2\alpha$  terms in equation (4.4) transform to the sum in  $\gamma$  over all neighbor cells  $N$ , because the values of  $\alpha$  can be different for each cell, for example in case of a variable diffusion coefficient or cell size. For each possible neighbor in the two dimensional grid, one secondary diagonal is added. The multidimensional form of equation (4.3) is

$$\left[ 1 + \sum_N \frac{D\Delta t}{(\Delta x_N)^2} \right] c_i^{t+} - \sum_N \frac{D\Delta t}{(\Delta x_N)^2} c_N^{t+} = c_i^t. \quad (4.6)$$

What happens to the matrix (4.4) is, that it now contains  $N$  secondary diagonals and the  $\alpha$  and  $\gamma$  coefficients can have different values for each cell. The reasons for this are the different number of neighbors per cell, possible different cell spacings caused by a non equidistant grid and also a diffusion coefficient that can vary for each cell. The

extension of equation (4.6) to the multi-component diffusion case is written as

$$\left[ 1 + \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} \right] c_{ij}^{t+} - \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} c_{Nk}^{t+} = c_{ij}^t. \quad (4.7)$$

#### 4.1.2 Explicit and trapezoidal methods

In the explicit version of the equation

$$\frac{c_i^{t+} - c_i^t}{\Delta t} = D \frac{c_{i+1}^t - 2c_i^t + c_{i-1}^t}{(\Delta x)^2}, \quad (4.8)$$

the terms of the Laplace operator are dependent on the current time step. If all compositions at time  $t$  are known, then a straight forward calculation of the composition at  $t+$  is possible. Solving the equation for  $c_i^{t+}$  yields

$$c_i^{t+} = \frac{D\Delta t}{(\Delta x)^2} (c_{i+1}^t - 2c_i^t + c_{i-1}^t) + c_i^t. \quad (4.9)$$

The extension of (4.9) to the multi dimensional case is formulated as

$$c_i^{t+} = D\Delta t \sum_N \frac{c_N^t - c_i^t}{(\Delta x_N)^2} + c_i^t, \quad (4.10)$$

where the sum is over all next neighbors of the current cell, and  $\Delta x_N$  denotes the distance to the neighbor. Further generalization to the multi-component case leads to

$$c_{ij}^{t+} = \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} (c_{Nk}^t - c_{ik}^t) + c_{ij}^t, \quad (4.11)$$

where the composition of the current component  $j$  is determined by all other components  $k$ . Although the explicit solution is easier to calculate than the implicit solution, a major drawback is the inherent instability at larger time steps. To obtain valid results, the Courant-Friedrichs-Levy condition must be obeyed, which sets a limit for the maximum size of time steps (see [CFL67]).

A mix of the explicit and implicit method is called trapezoidal, with a special case, the Crank-Nicolson method (see [CN96]). The equations (4.8) and (4.1) are merged together and form

$$\frac{c_i^{t+} - c_i^t}{\Delta t} = \gamma_{ti} D \frac{c_{i+1}^{t+} - 2c_i^{t+} + c_{i-1}^{t+}}{(\Delta x)^2} + (1 - \gamma_{ti}) D \frac{c_{i+1}^t - 2c_i^t + c_{i-1}^t}{(\Delta x)^2}. \quad (4.12)$$

When the time integration constant  $\gamma_{ti}$  is one, then (4.12) turns into the implicit case

whereas for the value zero, the explicit case is obtained. Further transformations lead to a relation similar to (4.3):

$$\begin{aligned} & \left[ 1 + \gamma_{ti} \frac{2D\Delta t}{(\Delta x)^2} \right] c_i^{t+} - \gamma_{ti} \frac{D\Delta t}{(\Delta x)^2} (c_{i+1}^{t+} + c_{i-1}^{t+}) = \\ & \left[ 1 - (1 - \gamma_{ti}) \frac{2D\Delta t}{(\Delta x)^2} \right] c_i^t + (1 - \gamma_{ti}) \frac{D\Delta t}{(\Delta x)^2} (c_{i+1}^t + c_{i-1}^t). \end{aligned} \quad (4.13)$$

Now the right side contains also a matrix instead of a vector. The elements of this matrix can easily be calculated, since all  $c^t$  values are known. Below, the extension to the multidimensional and multi-component case is shown, in one single step:

$$\begin{aligned} & \left[ 1 + \gamma_{ti} \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} \right] c_{ij}^{t+} - \gamma_{ti} \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} c_{Nk}^{t+} = \\ & \left[ 1 - (1 - \gamma_{ti}) \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} \right] c_{ij}^t + (1 - \gamma_{ti}) \sum_{k,N} \frac{D_{jk}\Delta t}{(\Delta x_N)^2} c_{Nk}^t. \end{aligned} \quad (4.14)$$

#### 4.1.3 Boundary conditions

Boundary conditions can be applied to the cells, either through fixing the cell composition or the surface composition of a cell. An implicit form representing a four-cell grid, where the second cell has a fixed surface composition, is shown in

$$\begin{pmatrix} 1 + 2\alpha & -\alpha & 0 & 0 \\ -\alpha & 1 + 2\alpha + \alpha_{fs} & -\alpha & 0 \\ 0 & -\alpha & 1 + 2\alpha & -\alpha \\ 0 & 0 & -\alpha & 1 + 2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_2^{t+} \\ c_3^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t + \alpha_{fs} c_{fs} \\ c_2^t \\ c_3^t \end{pmatrix}. \quad (4.15)$$

During creation of the boundary condition, the user has to specify both the fixed surface composition  $c_{fs}$  and the direction which indicates the intended surface. The variable  $\alpha_{fs}$  is similarly defined as  $\alpha$  in equation (4.4), but for the distance  $\Delta x$ , the half neighbor-distance, which equates to the surface distance, is inserted.

The situation changes when using a fixed cell composition as boundary condition. Cells with fixed composition have no corresponding entries in the solution vector and the start vector of an implicit representation. In the computation of the cell matrix elements, the neighbor terms of fixed composition cells are omitted. Instead, these terms, multiplied by the fixed composition, are added to the right side vector. An example of a six cell grid, having the cells 2 and 3 fixed, is shown in (4.16).

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & 0 \\ -\alpha & 1+2\alpha & 0 & 0 \\ 0 & 0 & 1+2\alpha & -\alpha \\ 0 & 0 & -\alpha & 1+2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_4^{t+} \\ c_5^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t + \alpha c_2^t \\ c_4^t + \alpha c_3^t \\ c_5^t \end{pmatrix}. \quad (4.16)$$

The explicit, one-dimensional case, representing a cell with a fixed surface composition, is written as

$$c_i^{t+} = \alpha(c_{i+1}^t - 2c_i^t + c_{i-1}^t) + c_i^t - \alpha_{fs} c_i^t + \alpha_{fs} c_{fs}, \quad (4.17)$$

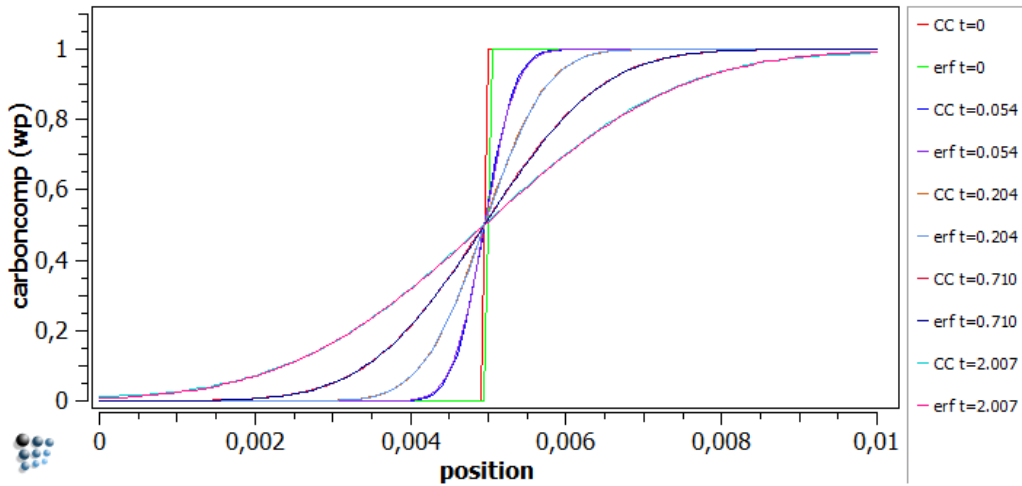
where the terms  $-\alpha_{fs} c_i^t$  and  $\alpha_{fs} c_{fs}$  are added to equation (4.9). Also in the explicit formulation, cells that have a fixed-composition boundary condition attached are completely omitted, both in computation of composition and in the neighbor summation of other cells.

#### 4.1.4 Comparison with analytic solutions

A simulation of a simple diffusion couple is compared with the analytic solution of the problem, an error function. For this purpose a function

$$\operatorname{erf}\left(\underbrace{\frac{x}{2\sqrt{Dt}}}_z\right) = \frac{2}{\sqrt{\pi}} \int_0^z \exp^{-k^2} dk \quad (4.18)$$

was defined in MatCalc. The sample is an austenitic iron bar of 1 cm length, separated into a carbon rich and a carbon depleted side.



**Figure 4.1:** Comparison of calculated solution and error function

In figure 4.1, the results of the simulation are shown at different time steps measured in seconds. During the most of the time, the line-pairs are in perfect agreement, but after a time of two seconds, they do not match any more, because in the analytic solution an infinite sample is assumed. The simulation parameters are shown in table 4.1.

In order to understand the composition notations in this thesis, a short introduction of corresponding units will be given [AA92]. The composition variable  $c_l$  in Fick's laws (see equation (3.15)) denotes the amount of moles of a component  $l$  per unit volume. In literature, the variable  $x$  often represents the non-normalized version of the composition. Other possible units are weight percent or u-fraction, where for the first, the contribution of the component to the total weight is considered. For the unit u-fraction, the mole fraction of one component is normalized to the sum of all substitutional components  $S$  like

$$u_l = \frac{x_l}{\sum_{j \in S} x_j}. \quad (4.19)$$

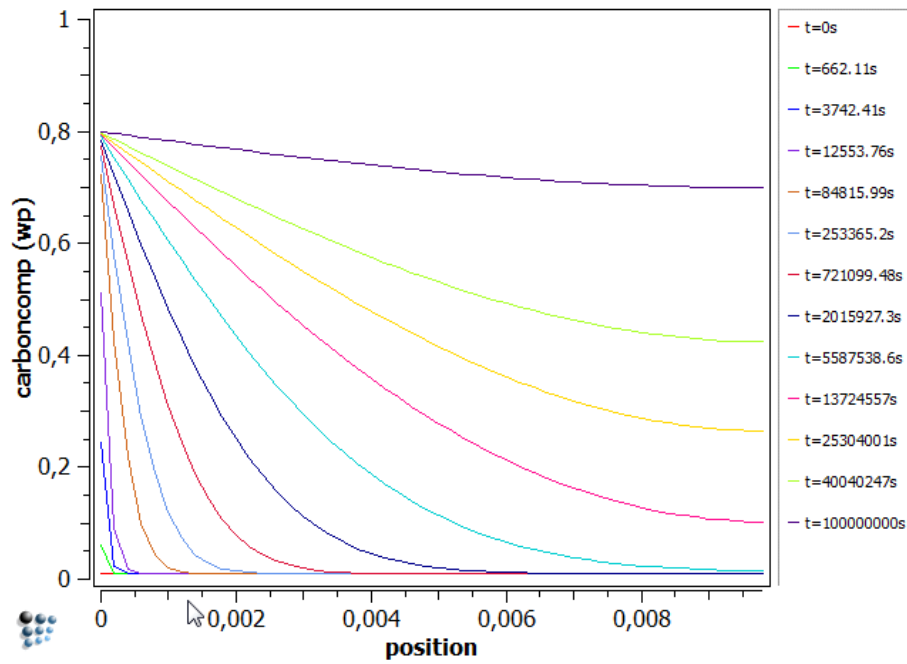
Abbreviations of the units are wp for weight percent and uf for u-fraction.

Sample size:	1 cm
Number of cells:	100
Temperature:	1000 K
Carbon composition	
left side:	1 wp
right side:	$10^{-4}$ wp
Diffusion coefficient:	$10^{-6}$ m <sup>2</sup> /s
Elements:	Fe,C
Phases:	fcc
Geometry:	planar

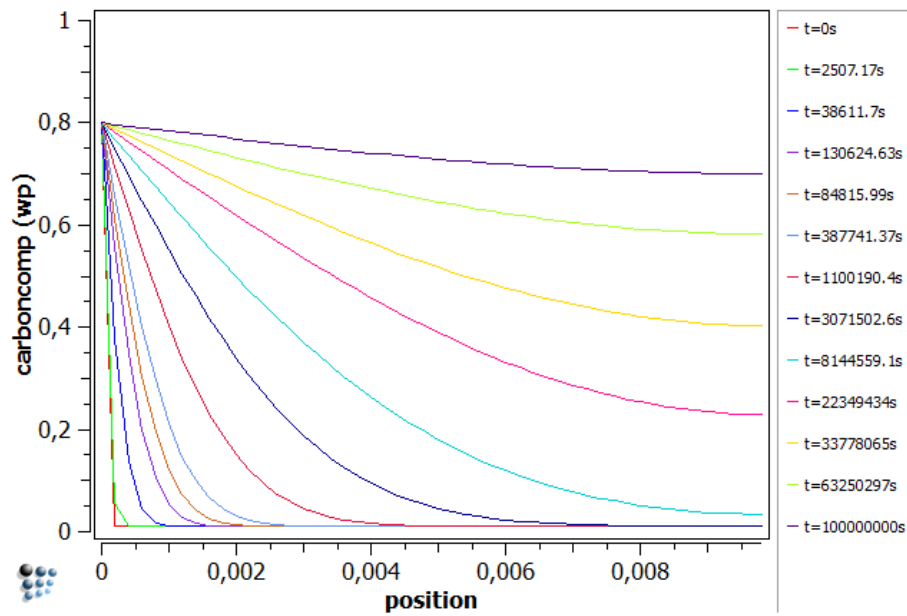
**Table 4.1:** Simulation parameters of diffusion couple example

The second example is a carburization treatment of a metal bar combined with a comparison of having either the cell composition or the surface composition fixed. Carburization is typically used in the steel hardening process.

When looking at figures 4.2 and 4.3, it is obvious that with a fixed cell composition, the composition value in the first cell is fixed right when the simulation starts, whereas in the other case, the composition still has to rise to this level, which takes some time. This is the reason why the carburization in figure 4.2 is slower, and it also seems to be more realistic to set a fixed surface composition, assuming that the cells are not very small. Please check table 4.2 for details on simulation parameters.



**Figure 4.2:** Composition in the sample at different time steps using a fixed surface composition boundary condition



**Figure 4.3:** Composition in the sample at different time steps using a fixed cell composition boundary condition

Sample size:	1 cm
Number of cells:	50
Temperature:	1000 K
Carbon composition	
left side:	0.8 wp
right side:	$10^{-4}$ wp
Diffusion coefficient:	$10^{-12}$ m <sup>2</sup> /s
Elements:	Fe,C
Phases:	fcc
Geometry:	planar

**Table 4.2:** Simulation parameters of carburization example

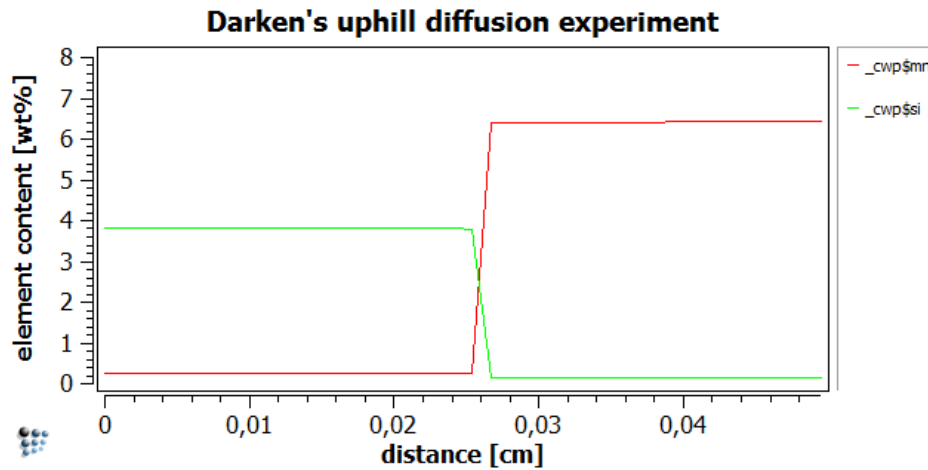
#### 4.1.5 Darken experiment

In his famous paper [Dar49], Darken proved experimentally that in a system of more than two components, it is possible that diffusion can occur against the concentration gradient. Although similar experiments have been observed by Hartley for an acetone-water solution in 1931, Darken was the first to conduct such experiments for metals. He conducted a series of weld-diffusion experiments, where two specimen of nearly the same carbon content, but different element content were welded together and afterwards heat treated in a furnace. The diffusion welding process was performed in a helium atmosphere using a current of 4500 A until a temperature of 1125 °C, measured by an optical pyrometer, was reached. To ensure accurate heat treatment conditions, the furnace had a long zone of temperature uniformity with variations less than 1 °C maximum and a control circuit to prevent fluctuations was used. The atmosphere in the furnace consisted of purified helium gas, to prevent any interference with the carbon content of the samples. Each one of the specimen was kept at 1050 °C for two weeks, cooled down in the furnace to 800 °C and then analyzed after removal. Because of the different chemical potential of carbon in both sides, a carbon redistribution was produced during the treatment. The main reasons for the carbon diffusion in the samples, were the different silicon content in each of the weld parts, for two samples, and also the different manganese content, in one sample. Darken writes that: ‘Silicon decreases and manganese increases the affinity of austenite for carbon’ [Dar49, p. 433]. Also, he found that Molybdenum has a similar, but somewhat greater effect than manganese. The simulation will also prove, what he supposed for samples that are kept in the furnace for a very long time, namely the return of the composition to equilibrium.

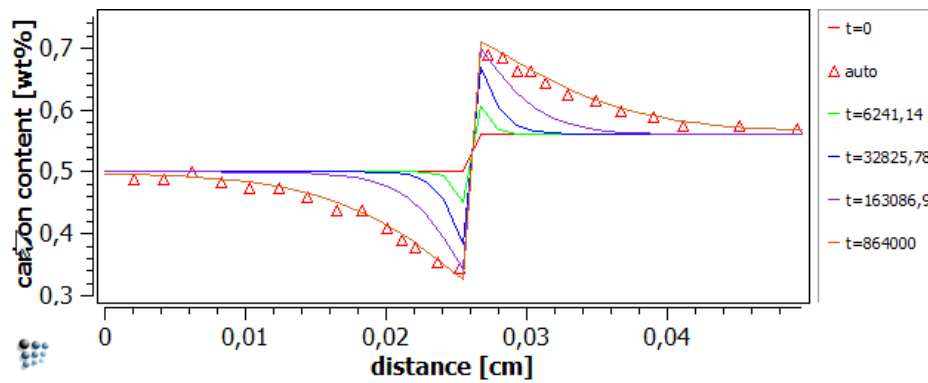
The numerical simulation of the darken experiment was performed, using similar parameters as Darken, as listed in table 4.3. Figure 4.4 shows the initial element



distribution and figure 4.5 the carbon content of the sample at different time steps. The red triangles mark the experimental values observed by Darken, inserted with a table in MatCalc. To separate the process of uphill diffusion from the return to equilibrium, the simulation was halted after 10 days of heating and after analyzing the results, it was started to continue till the end time. The second stage of the simulation process is shown in figure 4.6 and figure 4.7, respectively. All results are in good agreement with both the experimental results of Darken and his predictions.



**Figure 4.4:** Manganese and silicon content in the sample prior to simulation start



**Figure 4.5:** Carbon content in the sample from simulation start to end of uphill stage

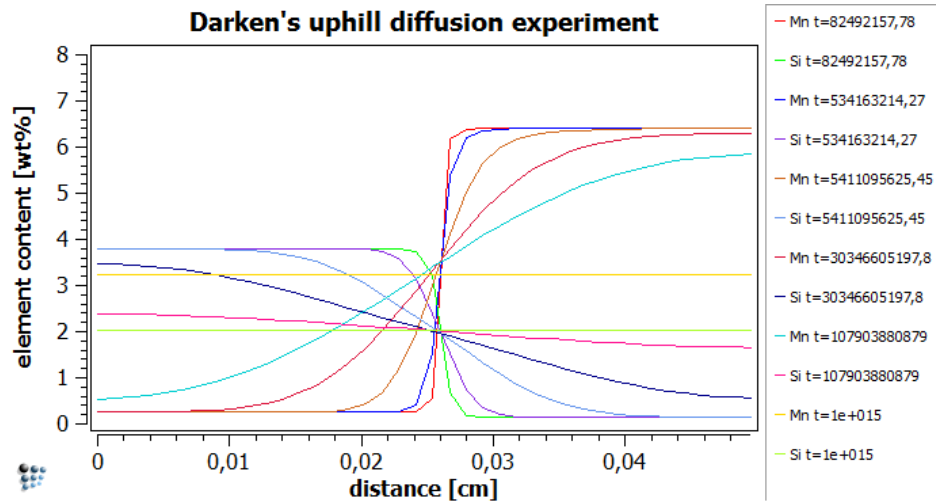


Figure 4.6: Manganese and silicon content beyond 10 days of heat treatment

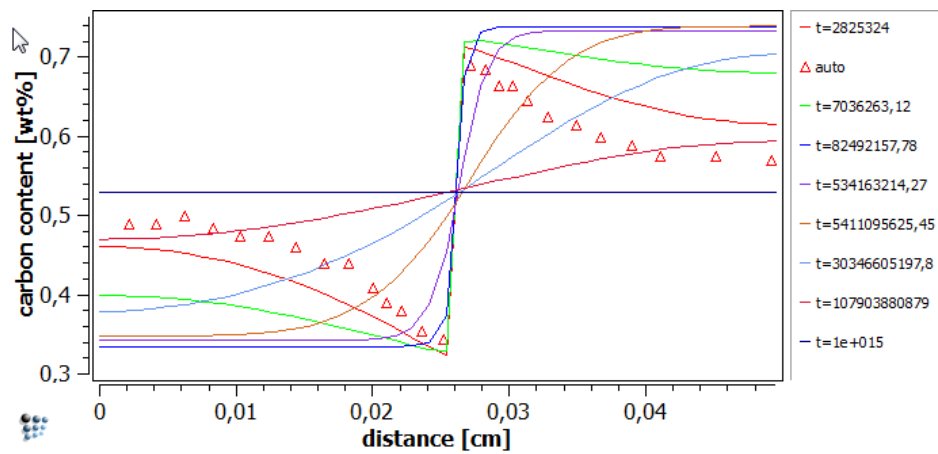


Figure 4.7: Carbon content beyond 10 days of heat treatment

Sample size:	4.9 cm
Number of cells:	40
Temperature:	1050 °C
Carbon composition	
left side:	0.5 wp
right side:	0.56 wp
Silicon composition	
left side:	3.8 wp
right side:	0.14 wp
Manganese composition	
left side:	0.25 wp
right side:	6.45 wp
Diffusion coefficient:	from MatCalc database mc_sample_fe
Elements:	Fe,C,Mn,Si
Phases:	fcc
Geometry:	planar
Time until first stop:	864 000 s (ten days)
Total simulation time:	$10^{15}$ s

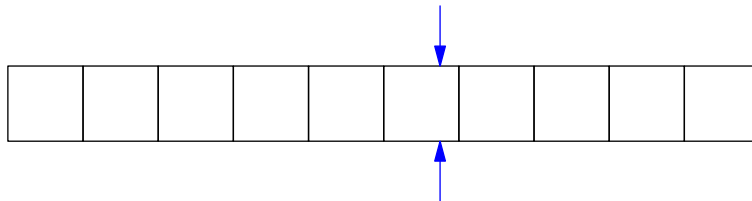
**Table 4.3:** Simulation parameters of Darken example

## 4.2 Moving interfaces

The mathematical description of interface movement during a phase transformation and the modification of the equations describing the heat conduction or diffusion process is usually called Stefan problem. Since the partial differential equations are not valid at an infinitesimally thin interface, the Stefan condition is used to close the gap between the two phases. For diffusional processes, this condition expresses the relation between the interface velocity and the particle-fluxes/compositions, on both sides of the interface.

To be able to simulate moving boundary problems with MatCalc, its simulation module had to be enhanced to support interface objects and their behavior. Several approaches, on how to deal with a moving interface and extend the grid to support such problems, were discussed to be implemented into MatCalc. For simplification and since more dimensional interface problems have not been implemented, only the one-dimensional case will be shown.

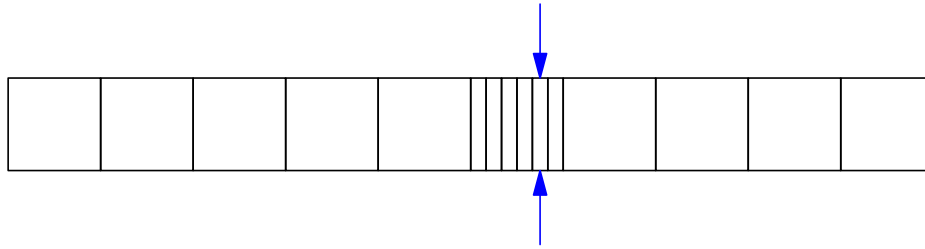
When doing the spacial discretization, it is either possible to use a static grid or a dynamic grid. Using the former, it is difficult to model the position of the interface precisely. A distinction can also be drawn between front tracking and implicit methods, where the first calculate the explicit position of the interface and the second try to estimate the interface position. Front tracking methods include the interface closure condition as part of the problem and some of them, like the Murray Landis method, use a variable spaced grid. One example of a fixed grid approach is to have the interface located on a specific location inside a cell (see figure 4.8), which is then separated into two phases. Each phase's fraction of the cell depends on the position of the interface.



**Figure 4.8:** Interface is located inside a cell

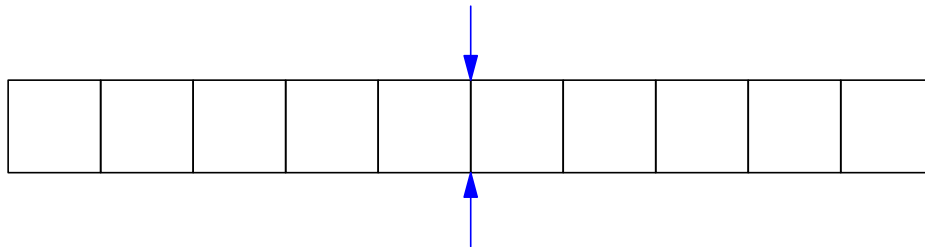
This way of dealing with the problem has the disadvantage that many modifications would have been necessary for implementation into the existing MatCalc diffusion model. To improve calculation in such cases, the interface cell could be separated into further cells (figure 4.9).

The chosen method, which has been implemented in this work and which is a dynamic grid front tracking method similar to the Murray Landis method, keeps the interface between two cells at each time step (figure 4.10). To make this possible, the size of the grid cells is adjusted accordingly, to follow the interface position (figure 4.11). If the interface always stays between the same two cells, the cell size decreases on the side



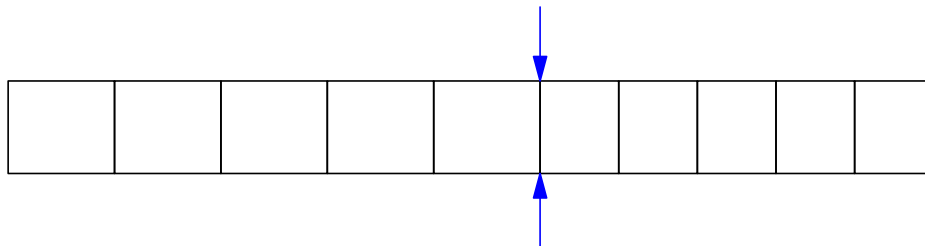
**Figure 4.9:** Region of interface is using smaller cell spacing

the interface moves to and it increases on the other side. This has the drawback that the cell size on one side can get very large and on the other very small. Therefore, it is favorable to swap the interface cell during interface movement.



**Figure 4.10:** Interface is exactly between two cells, no cells displaced

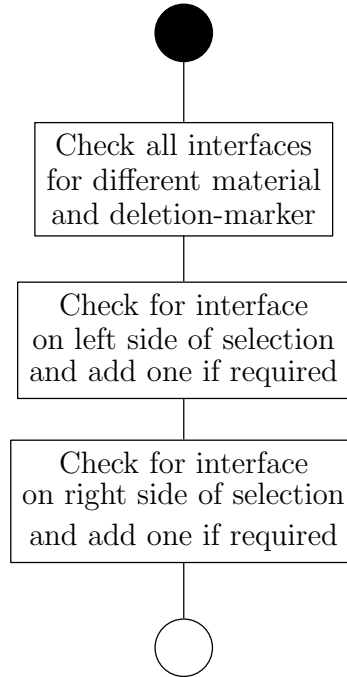
Referring to the definition of MatCalc nomenclature, a node is a point in the grid where two cell edges touch each other. One reason, why the variable spaced grid was introduced is that a parameter to displace grid nodes was already implemented in the software from a previous version. In the current version, an interface is defined as an infinitesimally thin boundary between two cells of the grid and a mushy region is not considered. An example of cell displacement coming from interface movement is shown in figure 4.11.



**Figure 4.11:** Interface is now on an intermediate point, cells are displaced

A mandatory condition for a MatCalc interface is further that the two neighbor cells are assigned different materials. When the user sets two different materials to neighboring cells, an interface is automatically created, its adjacent cells are set and

the x-position, to match the node between these two, is determined. Since it is possible to set the material property of a continuous selection of cells, it can also happen that two interfaces are created after one step. The whole process that is performed, after the material property has been modified, is shown in figure 4.12.



**Figure 4.12:** Flowchart of the "Check interface" process

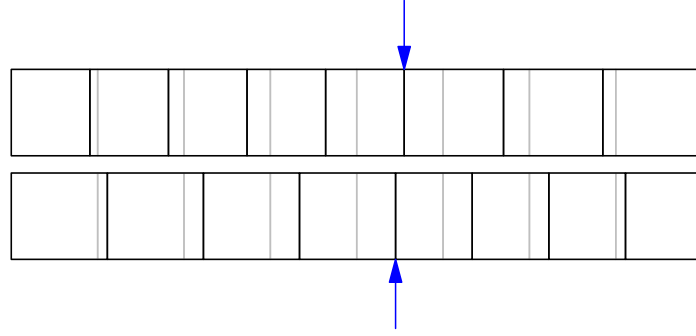
### 4.2.1 Variable grid distribution

When doing a diffusion simulation without moving interfaces, all cells are equally distributed and the displacement of each node is zero. If an interface exists and it has a non-zero velocity, then most probably the nodes will have to be displaced to fit the interface position accordingly. Murray and Landis already proposed in their 1959 paper [ML59], that the variable grid method is the preferred method if the movement of the interface has to be tracked. Their example however, only featured one migrating interface, being always located between the same neighbor cells. In case of the interface moving over a longer distance, the cells on one side grow rather big, and on the other become relatively small, as shown in figure 4.13. The two materials are colored differently to mark the interface. Also, with such a distribution technique, it is hard to simulate the case of an interface escaping the sample. Numerical problems, caused by the tiny cell size, will arise, as soon as the interface reaches one sample-border. To solve this problem, an algorithm was implemented in MatCalc, to support neighbor cell exchange during interface migration. Figure 4.14 shows this process, with the



**Figure 4.13:** Grid displaced in a sub-optimal way

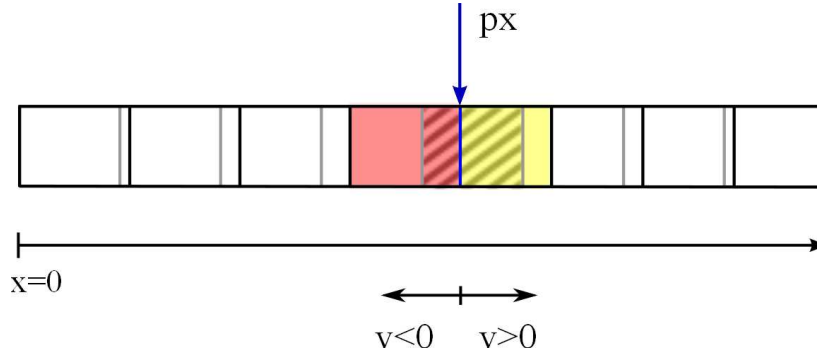
grey cells denoting the non-displaced grid and the black lines the displaced ones. In case of a right-moving interface, the lower cell distribution occurs first. For the



**Figure 4.14:** Cell distribution before and after a neighbor change

purpose of meshing, at each time step, the new position of the interface in the original, non-displaced grid is evaluated and the neighbor cells are adjusted. This also happens, when the user modifies the position of the interface manually, by issuing a MatCalc command.

On the basis of figure 4.15, the basic interface nomenclature of this thesis is introduced. The displaced grid is outlined in black and the grey lines represent the original,



**Figure 4.15:** Important definitions of interface nomenclature

stationary grid. The blue arrow and the line below highlight the interface. Whenever the terms ‘left’, ‘smaller’, ‘right’ or ‘greater’ are used in conjunction with an interface position, they refer to the position arrow below the grid, which points to the right. Smaller positions have a lower  $x$  value. Positive interface velocities cause movements

to the right, whereas negative velocities do the opposite. The left interface neighbor is colored in red, the right neighbor has a yellow coloring and the interface cell is shaded with black bars. The term ‘interface cell’ denotes the cell of the original grid, in which the interface position is located. Figure 4.16 shows the process that finds the interface cell and ensures, that the new interface position lies inside the sample. It also selects the correct interface-neighbors and handles special cases for the sample borders. The term ‘Increasing the right neighbor’ in the diagram means, that the neighbor cell will be the one with index  $i + 1$ , instead of  $i$ . This happens, for example, if the interface is already close to the left boundary, and the calculated neighbor cell index is out of bounds. Please note, that cell indices are also counted from left to right.

Equation

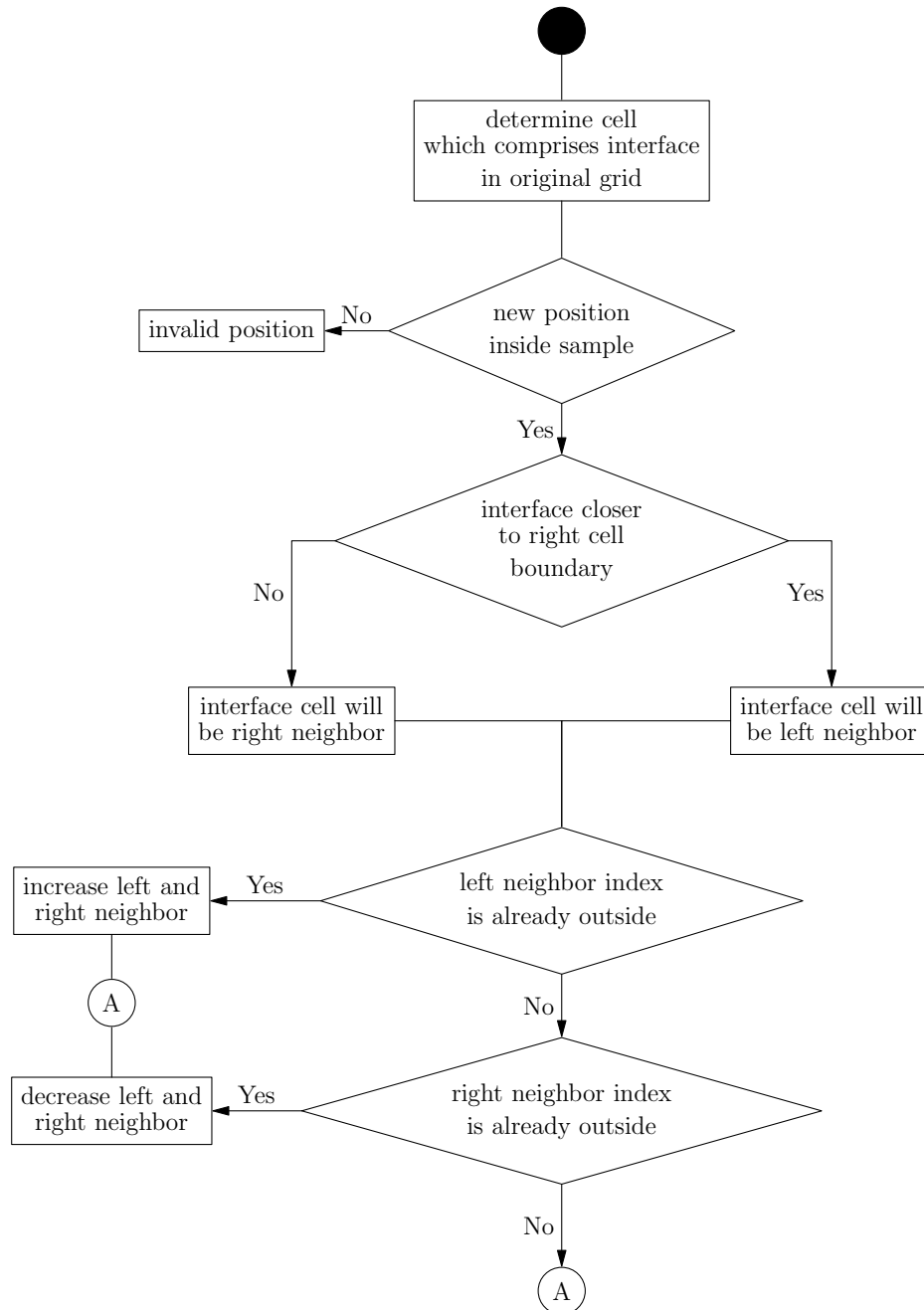
$$itfCell = \frac{itfPos}{sampleSize/nrCells} \quad (4.20)$$

is used to calculate the interface cell, whereby the sample size and the interface position are treated as real values, the others as integer. The position of the interface in this cell, determines the interface neighbors. If the interface is located closer to the right boundary, the interface cell will be chosen as left neighbor (again figure 4.14).

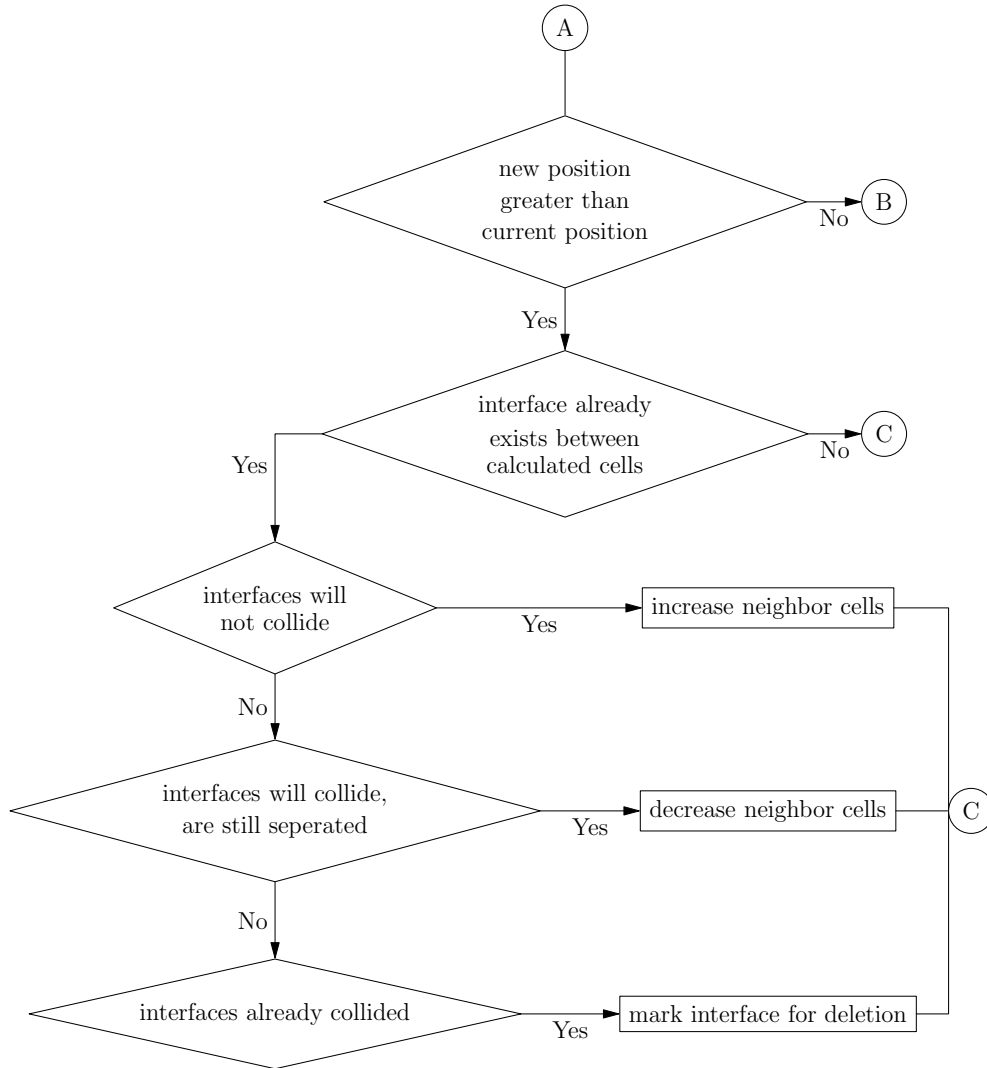
### 4.2.2 Interface checks and cell spacing

Apart from the standard neighbor cell detection for one interface, there are several special cases considering other existing interfaces, which are treated as shown in figure 4.17. Two phase boundaries that reside in the same interface cell, must have different neighbor cells while still being separated. In figure 4.17, the case of a right-moving interface is described. If two interfaces share the same neighbor cell, but move towards each other, the neighbor cells of the right moving interface are increased. Its neighbor cells are decreased, if the interfaces move away from each other. If an interface is overtaken, it is marked for deletion, which implies that the consistency-check will fail on that interface. Nearly the same approach is used for left-moving interfaces, but increasing and decreasing is swapped, the starting-node of figure 4.17 is  $B$  instead of  $A$ , the condition below the start point is changed and instead of node  $B$  an error is returned. After the neighbor cells are determined, some material checks are performed. On the one hand, the interface is only allowed to migrate inside a neighboring phase and on the other hand, the cell materials have to be adapted (see figure 4.18). Additionally, in case of the interface moving beyond the sample borders, the cell composition of the first or last cell is adapted. The interface consistency check automatically removes the interface, when it has either been marked for deletion or the interface resides between two cells consisting of the same material. The next step is to displace the cells on both sides of the interface independently. A displacement region reaches from one neighbor cell to the last cell of a row that shares the same material. In this region, the cell size





**Figure 4.16:** Flowchart of interface cell and neighbor determination

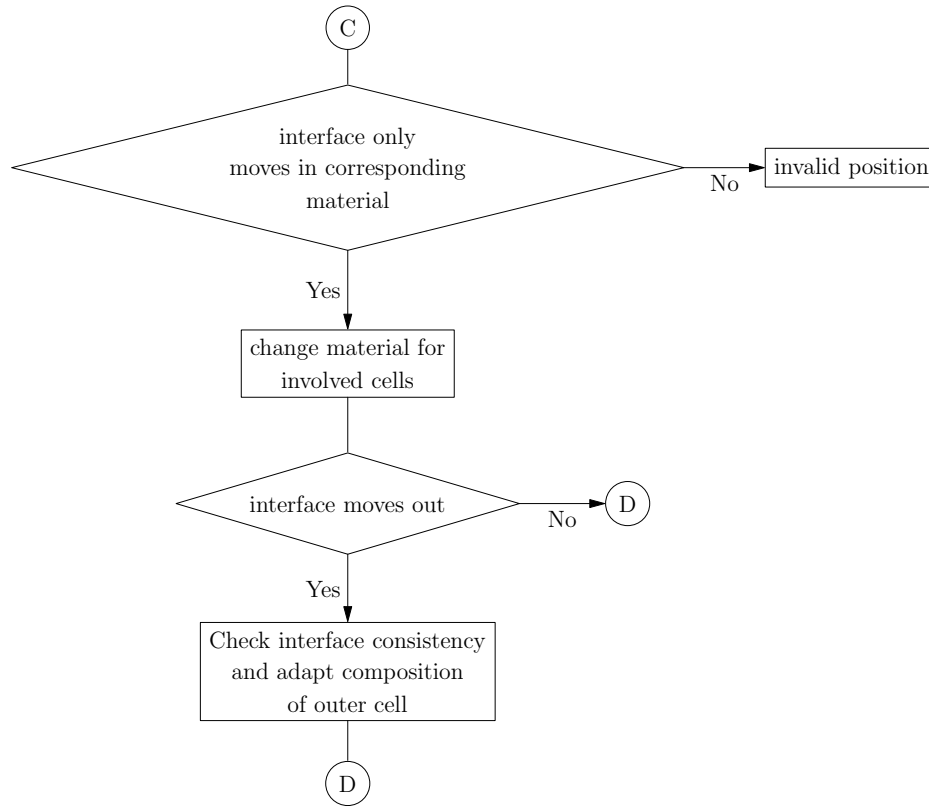


**Figure 4.17:** Flowchart of existing-interface checks

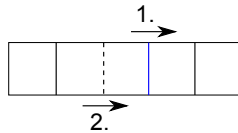
is calculated as follows:

$$cellSize = \frac{regionSize}{nrCellsInRegion}. \quad (4.21)$$

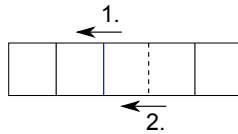
Each cell is displaced, fulfilling this requirement and afterwards, the interface-node is set to the exact interface-position (check figure 4.19). As already mentioned, the neighbor cells change during interface migration. When this happens, also the compositions have to be corrected. In case of a right-moving interface,



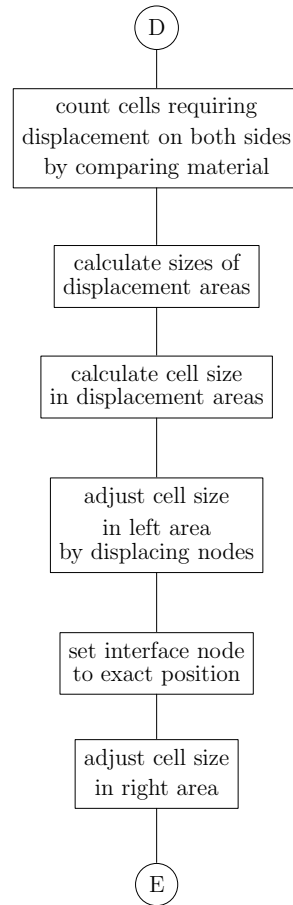
**Figure 4.18:** Flowchart of material correction



the new right neighbor will receive the composition and gradient of the previous, right neighbor and the new, left neighbor will receive the composition of the previous, left neighbor. For a left-moving interface, the process is inverted.



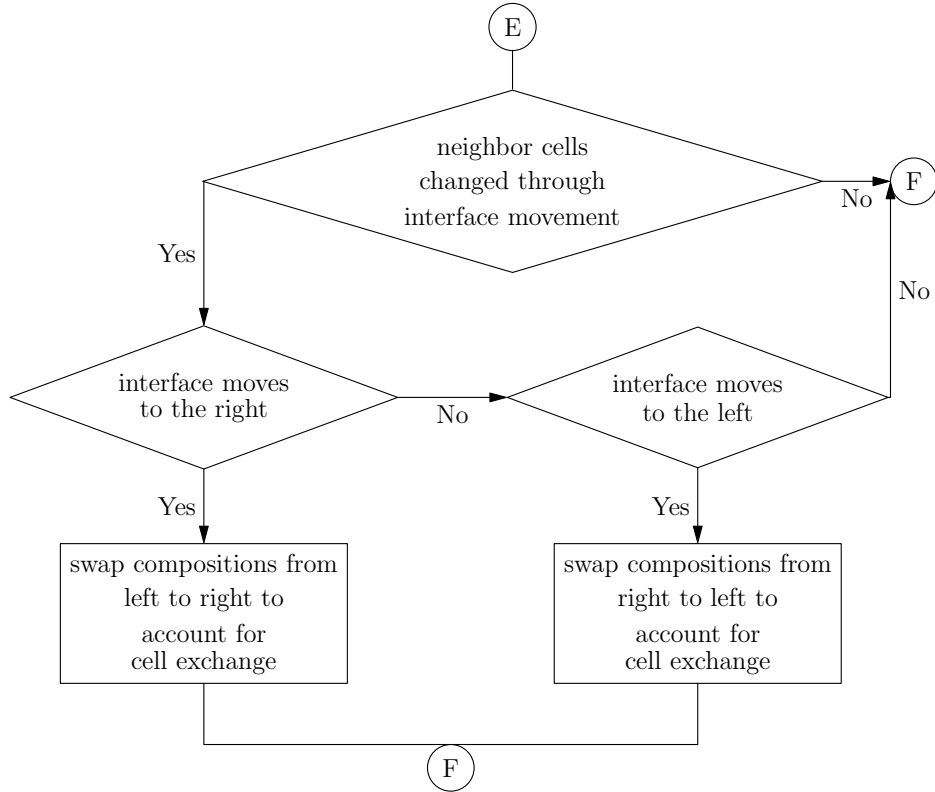
The blue line indicates the current interface position and the dashed line the previous one, both in the non-displaced grid. Figure 4.20 gives an overview of the function containing both processes and the details are shown in the pictures above. After the cells are displaced and, if the interface was not deleted, the consistency-check is performed again.



**Figure 4.19:** Flowchart of cell displacement

### Merging long-range diffusion and interface movement

To familiarize with the behavior of moving boundaries during a simulation, a simple example is studied. The user already defined a grid in MatCalc, set one material to the left half of the sample, another one to the right half and chose a constant interface velocity. After the simulation is started, several procedures are performed during each time step. Figure 4.21 shows a simplified version of this process and additional steps are required to model phase transformations involving moving interfaces. At the beginning, the time-step width is very small with  $10^{-12}$  s, but it increases through multiplication with a factor of ten in every round. The first step is to calculate the diffusion gradients, either using an explicit or implicit method of solving the diffusion equation. Afterwards, the maximum allowed time step is calculated. Diffusion, interface movement and the heat treatments impose limits on the time stepping procedure. If the composition gradient changes too much for one time step, or the current step is greater than the maximum, the interval must be adjusted and the gradients are recalculated. A diffusion



**Figure 4.20:** Flowchart of composition correction

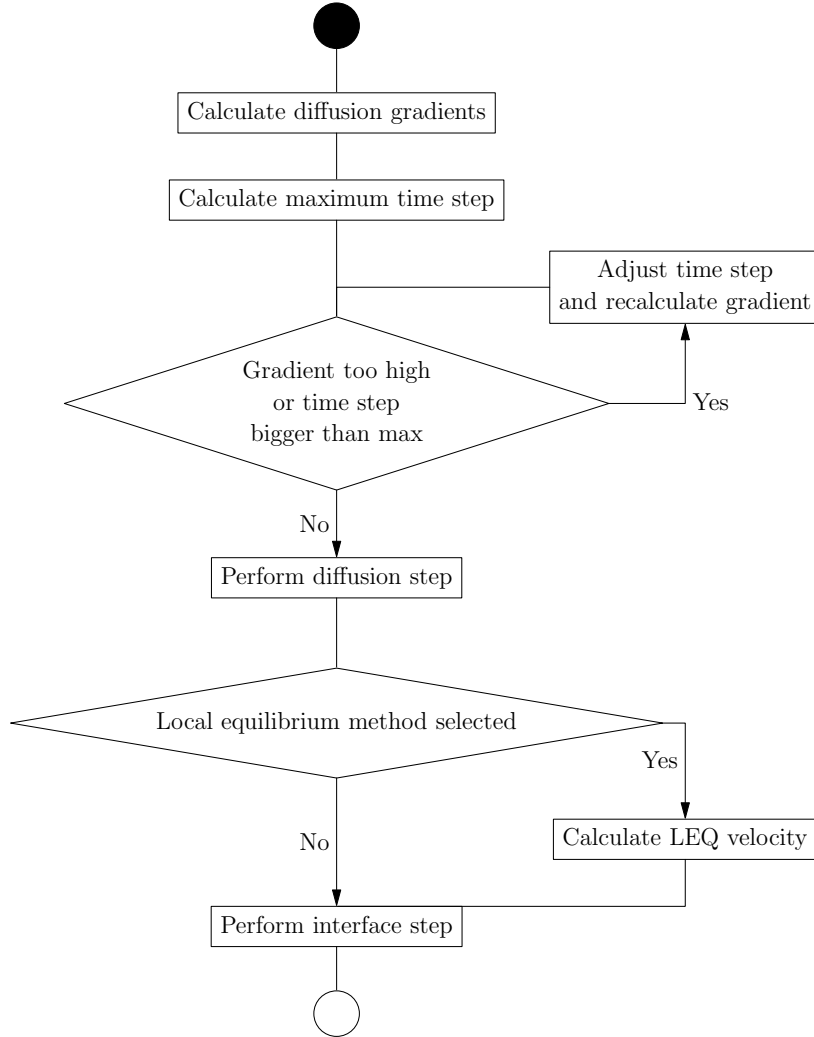
step is carried out using

$$c_{ij}^+ = c_{ij} + \dot{c}_{ij} \Delta t, \quad (4.22)$$

where  $c$  denotes the composition,  $\dot{c}$  the time gradient of composition,  $i$  the cell-,  $j$  the element-indices and  $\Delta t$  the time step. In this example, the part of figure 4.21 involving the local equilibrium velocity is skipped, since a constant velocity is chosen. In the interface step (see figure 4.22), the new position

$$px_i^+ = px_i + vx_i \Delta t \quad (4.23)$$

of each interface is calculated, with  $px_i$ ,  $px_i^+$  being the old, respectively the new position and  $vx_i$  representing the velocity of interface  $i$ . Additionally, this step ensures, that the interface is placed at the outermost position if it approaches the sample border and the cell size becomes too small. Performing this task is only needed when using the local equilibrium (LEQ) method for velocity calculation. Omitting this would constantly decrease the cell size and the interface could never leave the sample. Further discussion of this case follows in section 4.2.3. The last step, which again invokes the process described by figures 4.17 to 4.20, moves the interface to the desired position.



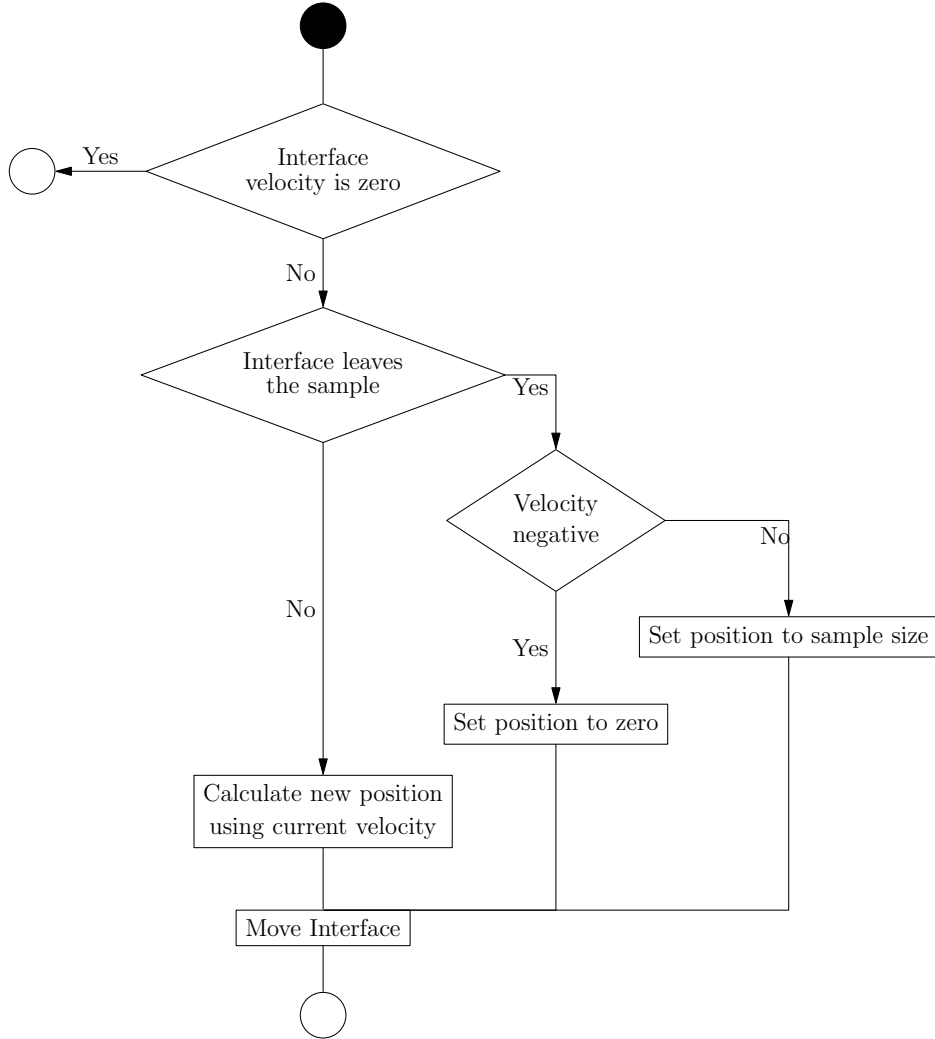
**Figure 4.21:** Flowchart of cell simulation

### 4.2.3 Time step limit

To guarantee numerical stability and integration accuracy during simulation, the size of the time steps must adhere to several limits. The limit check is carried out after the initial calculation of diffusion gradients, as shown in figure 4.21. For diffusion simulation, three different factors are considered during limit calculation. The limit imposed by diffusion is

$$\Delta t_{Diff} = \frac{c_{ij} c_{fact}}{|\dot{c}_{ij}|}, \quad (4.24)$$

with the composition  $c_{ij}$  of cell  $i$ , element  $j$ , the maximum composition change factor  $c_{fact}$ , which is 10 % by default (changeable by the user) and the time gradient of com-



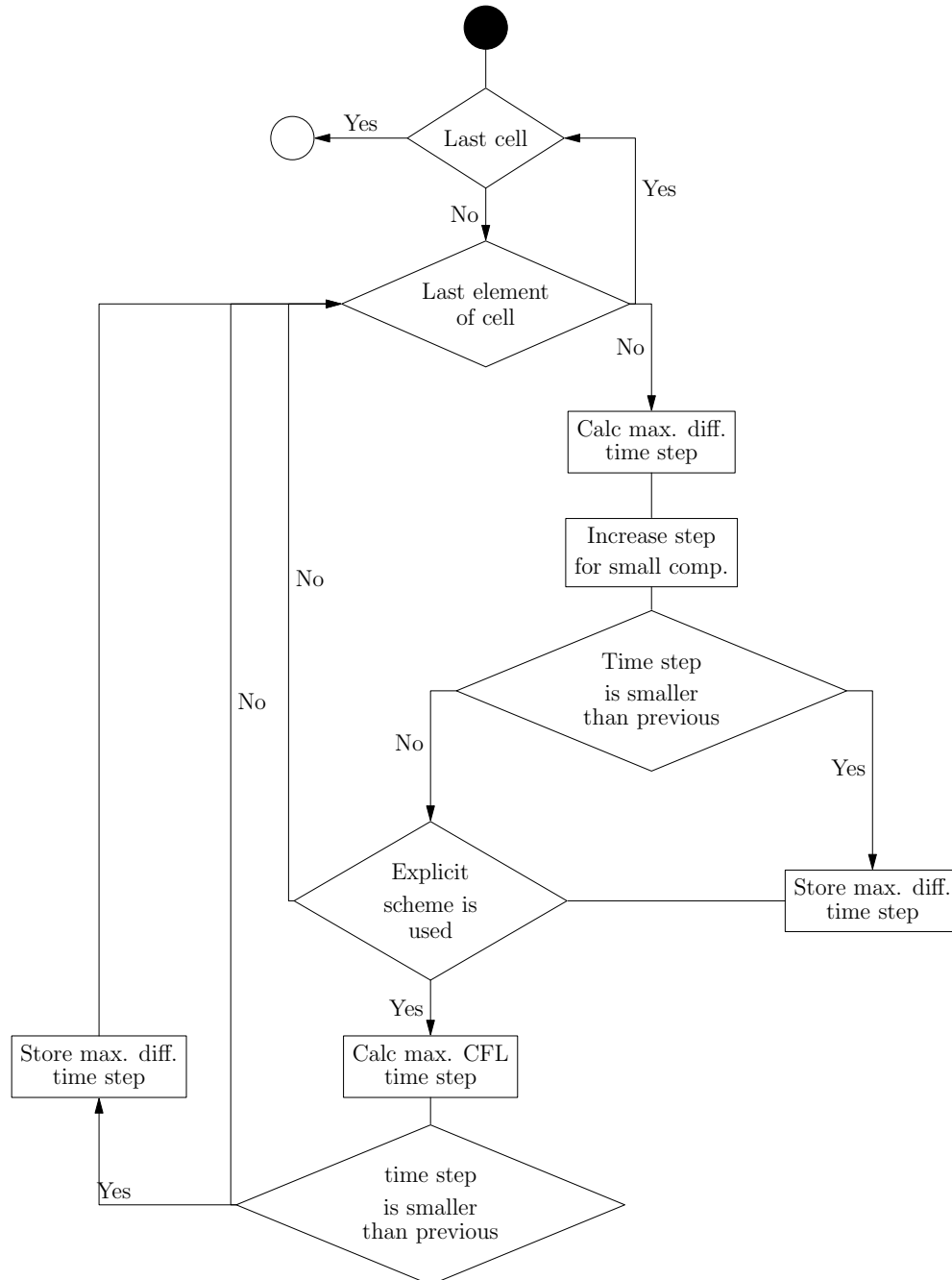
**Figure 4.22:** Flowchart of perform interface step

position  $\dot{c}_{ij}$ . Furthermore, the smaller the composition gets, the higher the maximum time steps will be chosen, to avoid small steps for low composition. This was already integrated in MatCalc, but an addition for the explicit method was implemented in this work, the Courant-Friedrichs-Levy condition (see [CFL67]) of the form

$$\Delta t_{CFL} = \frac{1}{2} \frac{(\Delta x_{min})^2}{|D_{kl}|}. \quad (4.25)$$

$\Delta t_{CFL}$  is the maximum time step that should be made,  $\Delta x_{min}$  the smallest next-neighbor distance and  $D_{ij}$  the diffusion coefficient between element  $k$  and  $l$ . Equation (4.25) is evaluated for each cell and the smallest result is chosen. The CFL condition is

not needed for the implicit version, which is stable regardless of the step size. Figure 4.23 shows the process of calculating the diffusion time limit.



**Figure 4.23:** Flowchart of diffusion limited step size determination

The time steps are also limited by interface movement, to prevent interfaces hopping



over cells. As explained in the previous section, if the interface comes close to the sample boundary and the cell size falls below a value of  $10^{-8}$  (measured in the same units as stated in the grid definition), then the time step will be adjusted, to move the interface correctly on the border. This will cause the removal of the interface. For all phase boundaries, the cell size of the neighbor, corresponding to the direction of movement is evaluated. If the size is below the mentioned threshold, which happens only in the vicinity of the border, the step width is set to

$$\Delta t_{Itf} = \frac{sampleSize - pos_{itf}}{|v_{Itf}|} \quad (4.26)$$

for a right moving interface or to

$$\Delta t_{Itf} = \frac{pos_{itf}}{|v_{Itf}|} \quad (4.27)$$

for a left moving one. The general limit for movement

$$\Delta t_{Itf} = \frac{cellSize/10}{|v_{Itf}|}, \quad (4.28)$$

ensures that interfaces move at most a tenth of the cell size per time step. This smoothes their track and improves numerical results. Limits are evaluated for all interfaces and the smallest computed limit is obeyed. Check figure 4.24 for an overview of the interface-limit process.

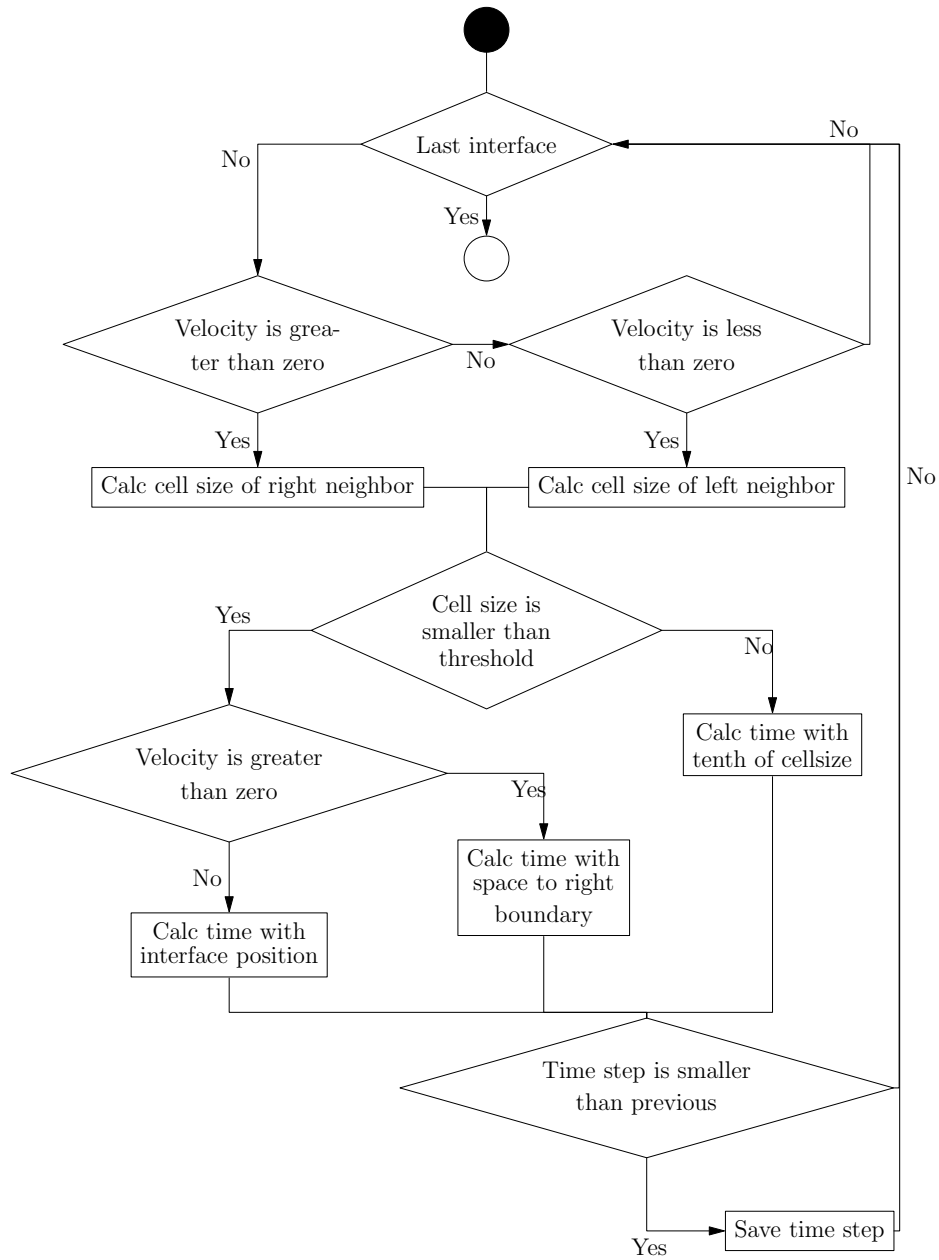
A further limit for time steps is imposed by the heat treatments. The maximum time step is

$$\Delta t_{HT} = \frac{1}{|\dot{T}|}, \quad (4.29)$$

with  $\dot{T}$  being the time gradient of temperature of the heat treatment. This condition will decrease the time step as required to force the temperature change to a value less than 1 K, otherwise phase changes would not be detected accurately. More details about heat treatments are presented in section 4.2.7.

#### 4.2.4 Composition treatment

An important property of an interface is the composition assigned to the two sides of the interface. These represent the coupling conditions in the diffusion Stefan problem. In one dimensions, an interface owns two compositions per element, one for the left and another for the right side, however, also intermediate points improving the description would be possible. The examples presented in this diploma thesis contain interfaces with one fixed element, such as a fixed-carbon interface in an iron sample. Two



**Figure 4.24:** Flowchart of interface limited step size determination

possible ways to achieve a fixed interface composition were implemented in MatCalc and compared. The first method uses fixed boundary conditions for the two sides of an interface, the other method uses virtual interface cells.

### Fixed neighbor cells

Two diffusion boundary conditions are implemented in MatCalc, both a fixed-surface composition and a fixed-cell composition. Furthermore, geometry boundary conditions exist, which define how the outermost cells should be treated in the calculation. Possible transition conditions are: Periodic, symmetric and open to ambient. To implement moving interfaces, the fixed-cell boundary condition was used.

When performing the diffusion gradient calculation, a fixed cell is treated in such a way, that its entry in the start and solution vector of equation (4.4) is skipped. The fixed composition is only considered in the entries of the right side vector of the neighbor cells. Instead of the  $-\alpha$  terms in the matrix, a term  $\alpha c_i$  is added to the right side of each neighbor cell, where  $i$  denotes the index of the fixed-composition cell.

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & 0 \\ -\alpha & 1+2\alpha & 0 & 0 \\ 0 & 0 & 1+2\alpha & -\alpha \\ 0 & 0 & -\alpha & 1+2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_4^{t+} \\ c_5^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t + \alpha c_2 \\ c_4^t + \alpha c_3 \\ c_5^t \end{pmatrix}. \quad (4.30)$$

Equation (4.30) shows a one dimensional example with a six cell grid. Fixing the composition of cells 2 and 3, decreases the rank of the matrix from six to four, whereby the right neighbor term of cell 2 and the left-neighbor term of cell 4 are moved to the right-side vector, multiplied with the fixed composition of cell 2 and 3, respectively. This represents the default case when one interface exists in a sample. The drawback of this method is, that an interface always consists of two cells. Considering the case, where two interfaces with different compositions collide, it has to be assured, that always two thin cells separate them. If the interfaces have the same composition in the shared material, only one cell in between is also possible. Regardless of that, an interface is not infinitesimal small when using the fixed neighbor cells method, since the interface always consists of these two cells, and they can not be used for calculation any more. In the fixed neighbor cells method, a link must be stored between the boundary conditions and the interface. During simulation, when a neighbor cell change happens, the boundary conditions have to be detached from the old and attached to the new neighbor cells. Additionally, if an interface leaves the sample or two interfaces collide, the corresponding boundary conditions must be deleted. This is done in the interface consistency check shown in figure 4.18. If during heat treatment simulation an interface is created, the boundary conditions have to be generated. Further details are discussed in section 4.2.7.

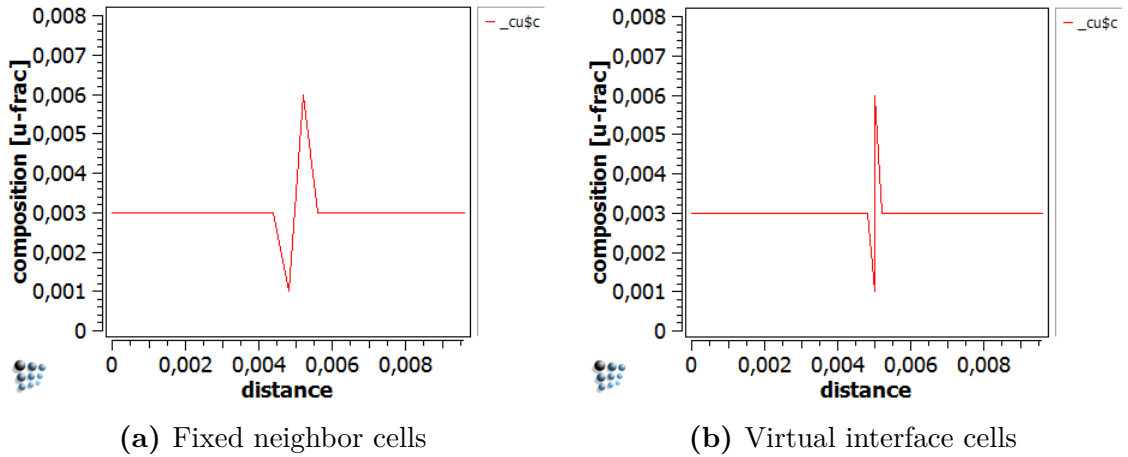
Although this method converges to the virtual interface cells method for a small cell size, the mentioned drawbacks are the reason why the second method was finally implemented in this work. On the other hand, the FNC method required no modification of the graph-plotting routines of MatCalc.

### Virtual interface cells

The virtual interface cells method is a modification of the FNC method. Instead of boundary conditions being managed for each existing interface, two virtual interface cells are used during simulation. These virtual cells behave like fixed cells, placed between the neighbor cells and the interface. To incorporate this feature into the existing diffusion model, the calculation of cell gradients was extended. If a cell is located adjacent to an interface, an interface term is added to the right side of the implicit solution equation, similar to the neighbor terms in the start vector of equation (4.30). The same six-cell system as in the FNC method is now represented by the system of equations as

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & 0 & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+2\alpha & -\alpha & 0 \\ 0 & 0 & 0 & -\alpha & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & 0 & -\alpha & 1+2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_2^{t+} \\ c_3^{t+} \\ c_4^{t+} \\ c_5^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t \\ c_2^t + \alpha_{it} c_{itL} \\ c_3^t + \alpha_{it} c_{itR} \\ c_4^t \\ c_5^t \end{pmatrix}, \quad (4.31)$$

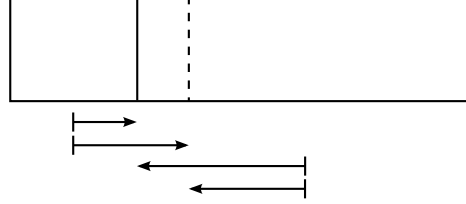
to describe the differences to the VIC model. The rank of the matrix is not reduced this



**Figure 4.25:** Comparison of interface composition handling

time, and the neighbor cells can have a variable composition. Instead of the constant

cell composition, a term  $\alpha_{it} c_{itL}$  or  $\alpha_{it} c_{itR}$  is added to the neighbor cells.  $\alpha_{it} = \frac{D\Delta t}{(\Delta x_{itf})^2}$  is the modified  $\alpha$  of equation (4.4), where instead of taking the distance to the neighbor cell  $\Delta x$ , the distance to the interface  $\Delta x_{itf}$  is used. As simplification, also the half distance to the neighbor cell can be used, however, for a non equidistant grid, this can be incorrect, instead the half cell size should be taken. Figure 4.26 shows the difference between the half distance and the half cell size approach. Two interface neighbor-cells are shown, which have very different spatial extension. If the half cell size is used, the neighbor's  $\Delta x_{itf}$  differ, whereas with the half distance, they are equal. The distance to the interface is correctly represented by the first approach. Distances are always measured from the cell center. Arrows 2 and 4 (counted from the top) belong to the half distance method. The boundary-condition management was omitted with the VIC



**Figure 4.26:** Half distance compared to half cell size

method, but modifications of the graph-plotting routines in MatCalc were required. Previously, when evaluating a profile plot, only the values of each cell were painted. Now, a special case for interfaces is included, where both interface compositions are painted at the interface position. This is also shown in the right picture of figure 4.25.

### 4.2.5 Interface velocity

In order to determine the interface velocity, two possible methods have been implemented. The easier of the two is, to set a fixed velocity, which is then used in equation (4.23). The local equilibrium method, on the other hand, calculates the velocity using the neighbor cell compositions. An equivalent formula has already been used by Murray and Landis [ML59], although they considered a case of heat conduction instead of diffusion. It supports only one element that affects the interface motion and, in the present implementation, is not suitable for multi-component interfaces. A local mass balance equation leads to

$$v_i = \frac{J^a - J^b}{c^a - c^b} \quad (4.32)$$

where  $v_i$  is the interface velocity and  $c^a$  and  $c^b$  are the concentrations on each side of the interface.  $J^a$  and  $J^b$  are the diffusional fluxes pointing in or out of the interface, corresponding to the phases  $a$  or  $b$ , which confine the interface. The fluxes are obtained

from (3.1), resulting in the full expression for the velocity

$$v_i = \frac{-D^a \frac{\partial c^a}{\partial x} + D^b \frac{\partial c^b}{\partial x}}{c^a - c^b}. \quad (4.33)$$

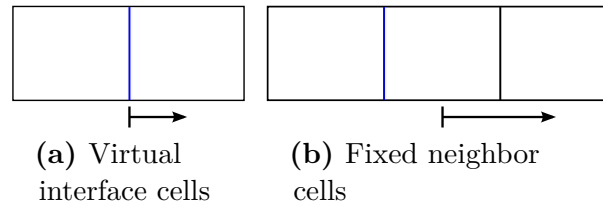
Justified by the differences of the FNC and the VIC method, two possibilities of flux discretization arise.

$$J^a = -D \frac{c_{itL} - c_{nL}}{\Delta x_{VIC}} \quad J^b = -D \frac{c_{nR} - c_{itR}}{\Delta x_{VIC}} \quad (4.34)$$

Equation (4.34) shows the first order discretization for the virtual interface cell method, with omitted phase indices, but one must keep in mind that diffusion coefficients of both fluxes are different at least for different materials. The indices  $nL$  and  $nR$  stand for the left and right neighbor cell of the interface, whereas  $c_{itL}$  and  $c_{itR}$  are the left and right compositions of the interface. The difference to the fixed neighbor cell approach comes from the fact, that the interface composition is expressed through a fixed neighbor composition. Therefore, the fluxes change to

$$J^a = -D \frac{c_{nL} - c_{nL-1}}{\Delta x_{FNC}} \quad J^b = -D \frac{c_{nR+1} - c_{nR}}{\Delta x_{FNC}} \quad (4.35)$$

where the compositions of the neighbor cells are replaced by the compositions of the next neighbor cells  $c_{nL-1}$  and  $c_{nR+1}$ . The interface compositions are substituted by the neighbor compositions. In figure 4.21, the local equilibrium calculation is shown after the diffusion step. It is carried out for each interface, which has this type of velocity calculation enabled. Another important difference is, that the distances  $\Delta x_{FNC}$  and  $\Delta x_{VIC}$  are not equal. The first  $\Delta x$  is the distance from the center of one neighbor cell to the next and the second is the distance from the interface to the center of the neighbor cell. Both cases are shown in figure 4.27, with the interface colored in blue. If the distances are used improperly, huge errors in calculation are created.



**Figure 4.27:** Comparison of distance handling for LEQ method

### 4.2.6 Murray Landis correction

Since the cells in this moving interface model are not stationary, the diffusion equation must be adapted to this case. A simple addition has been introduced by Murray and Landis in [ML59] and similar derivation for diffusion will be presented. The total time derivative of the composition variables on a moving grid is

$$\frac{dc}{dt} = \frac{\partial c}{\partial x} \frac{dx}{dt} + \frac{\partial c}{\partial t}, \quad (4.36)$$

combined with the diffusion equation (3.3) leading to

$$\frac{dc}{dt} = \frac{\partial c}{\partial x} \frac{dx}{dt} + D \frac{\partial^2 c}{\partial x^2}. \quad (4.37)$$

After a time and space discretization, equation (4.37) becomes

$$\frac{c_i^{t+} - c_i^t}{\Delta t} = v_i \frac{c_{i+1}^{t+} - c_{i-1}^{t+}}{2\Delta x} + D \frac{c_{i+1}^{t+} - 2c_i^{t+} + c_{i-1}^{t+}}{(\Delta x)^2}, \quad (4.38)$$

where an implicit time parametrization is used. In contrast to equation (4.32), where  $v_i$  stands for the velocity of the interface, here  $v_i$  denotes the velocity of cell  $i$  introduced by the displacement procedure described in section 4.2.2. Murray and Landis used the simple formula

$$v_i = v \frac{n - j}{n - 1} \quad (4.39)$$

for calculation of the cell velocity from the interface velocity. Since in MatCalc, multiple interfaces are supported, all cell positions of the previous and current time step ( $x_i^t$  and  $x_i^{t+}$ ) are stored and used to calculate the cell velocity, together with the time step width  $\Delta t$  as

$$v_i = \frac{x_i^{t+} - x_i^t}{\Delta t}. \quad (4.40)$$

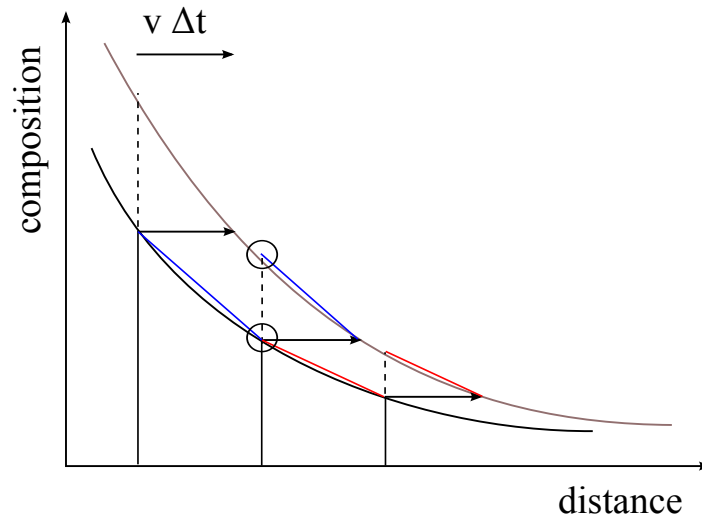
The obtained velocities are then used for calculation in (4.38). The additional term of equation (4.38) was incorporated into the calculation of diffusion gradients, which is shown in figure 4.21. The diffusion matrix of the implicit solution changes to

$$\begin{pmatrix} 1 + 2\alpha & -\alpha + \delta_{ml} & 0 \\ -\alpha + \delta_{ml} & 1 + 2\alpha & -\alpha + \delta_{ml} \\ 0 & -\alpha + \delta_{ml} & 1 + 2\alpha \end{pmatrix} \begin{pmatrix} c_0^{t+} \\ c_1^{t+} \\ c_2^{t+} \end{pmatrix} = \begin{pmatrix} c_0^t \\ c_1^t \\ c_2^t \end{pmatrix}, \quad (4.41)$$

where  $\delta_{ml} = \frac{v_i \Delta t}{2\Delta x}$  is the Murray Landis correction term. Applying this correction is required for reducing the deviations from mass conservation, but still it is not the perfect solution. An improved model has been suggested by Crusius et al. in [CIK<sup>+</sup>92],

with the drawback of higher computational costs.

Since the Murray Landis model uses a displaced grid, the composition values of the cells have to be interpolated, to represent the values at the adjusted positions. The interpolation mechanism is shown in figures 4.28 and 4.29, where the composition profile moves to the right. A translation of the composition profile in one direction is equivalent to a grid translation of the same size in the opposite direction. The new composition values are determined by projecting the slope back from the new position to the old position. For a curved line, this calculated value does not match the exact composition value, only for a straight line, it is precise. The interpolation lines with the same slope are equally colored. In figure 4.28, the top circle shows an example of the resulting deviation.

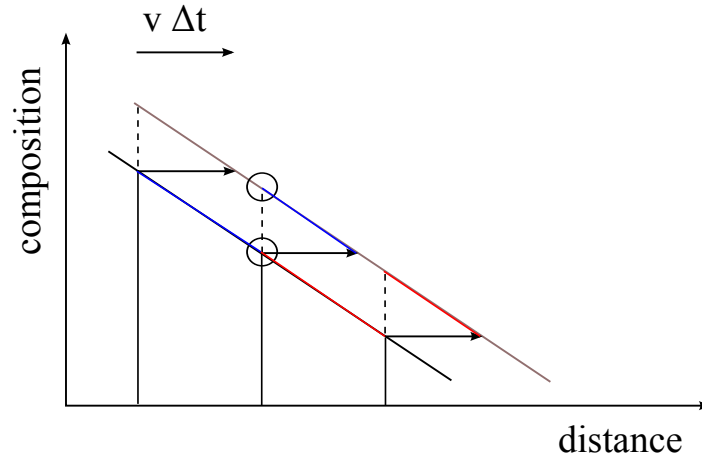


**Figure 4.28:** Interpolation of composition values for the Murray Landis correction - curved-shaped composition

#### 4.2.7 Heat treatments

Usage of heat treatments was already possible in the precipitation kinetics module of MatCalc, although it was not implemented in conjunction with diffusion simulation. A heat treatment consists of an arbitrary number of linear time-temperature segments, which each can have one of three heating modes. Three variable parameters are available, the end temperature, the heat/cooling rate and the duration of the segment. Each heating mode fixes one of these parameters through calculation of the other two, user defined parameters. During the simulation, a heat treatment imposes time limits on the calculation, which is discussed in section 4.2.3. The simulation temperature is changed to the current heat treatment temperature, which, for simplification, is assumed to be uniform in the whole sample. Otherwise, the diffusion and heat conduction





**Figure 4.29:** Interpolation of composition values for the Murray Landis correction - straight composition

equation would have to be solved in parallel. The user can specify a heat treatment and afterwards define, if it should be used in the diffusion simulation. Further details on usage of heat treatments for phase transformations are described in section 4.3. The aim of combining heat treatments with diffusion calculation, is to be able to simulate phase transformations during arbitrary heating or cooling processes.

### 4.2.8 Examples

Three examples were developed to show the implementation of moving interfaces. They do not represent real cases of phase transformations because they will not conserve mass. They only demonstrate the handling of interfaces in the code. The first shows two interfaces colliding with fixed velocities, the second two interfaces leaving the sample and the second an interface movement calculated with the LEQ method.

#### Interface merge

Three regions exist in this artificial system setup, where austenite represents the outer part of the sample and ferrite the core. Two interfaces form and are equipped with different-signed, constant velocities and compositions. Upon simulation start, they move against each other and collide after approximately seven seconds. Afterwards, the sample is uniformly austenitic and the composition returns to equilibrium. The simulation and interface parameters are shown in table 4.4 and table 4.5.

Figure 4.30 shows the two interfaces prior to starting the simulation, at  $t = 0$  s and right before the collision, at  $t = 7.4$  s. In figure 4.31, the composition is shown at four different time steps, to visualize the interface migration. The grid distribution at the

Sample size:	1 cm
Number of cells:	20
Temperature:	1000 °C
Carbon composition	
whole sample:	0.003 uf
Diffusion coefficient:	$10^{-6} \text{ m}^2 \text{ s}^{-1}$
Elements:	Fe,C
Phases:	fcc, bcc
Cell Material:	
fcc:	0..4,14..19
bcc:	5..13
Geometry:	planar
Total simulation time:	$10^8 \text{ s}$

**Table 4.4:** Simulation parameters of interface collision example

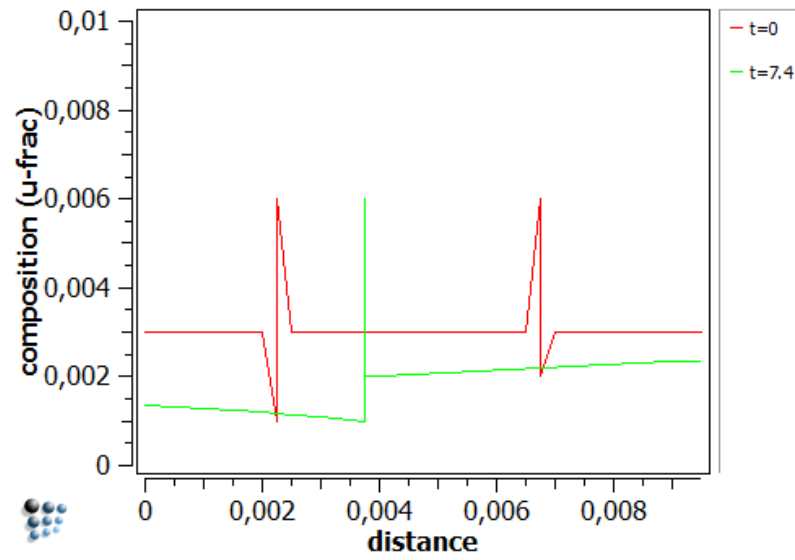
Interface 0:	
Carbon composition	
left:	0.001 uf
right:	0.006 uf
Velocity:	$2 \times 10^{-4} \text{ m s}^{-1}$
Interface 1:	
Carbon composition	
left:	0.006 uf
right:	0.002 uf
Velocity:	$-4 \times 10^{-4} \text{ m s}^{-1}$

**Table 4.5:** Interface parameters of collision example

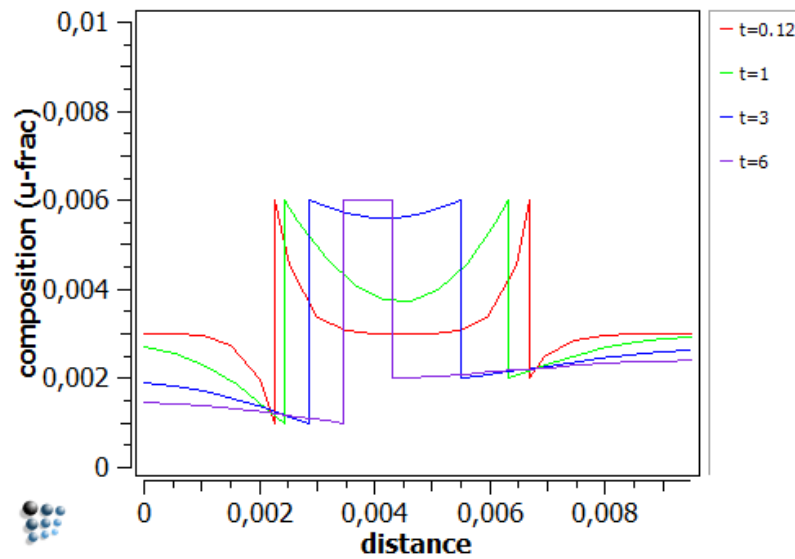
same time steps is shown in figure 4.32, whereby the two additional steps are the first and the last and the time increases from top to bottom. It is important to emphasize that these examples do not conserve mass, and, therefore, the initial and final total composition differ.

#### Interface escaping the sample

Simulation parameters for this example are the the same as in the interface collision example, only the interface parameters differ (see table 4.6). Two interfaces are moving in the opposite direction at different velocities and one after the other leaves the sample. As the cell size becomes smaller at the boundaries, the time steps are delimited by



**Figure 4.30:** Interface merge example: Composition at start and merge



**Figure 4.31:** Interface merge example: Composition during simulation



**Figure 4.32:** Cell distribution for interface merge example

the interface movement. Like in the previous example, figures 4.33 and 4.34 show the composition at different time steps. Figure 4.35 shows the position of the interfaces during simulation. After an interface has left the sample, the reported position will be zero, which is the reason why the green line is dropping to zero at approximately 15 s.

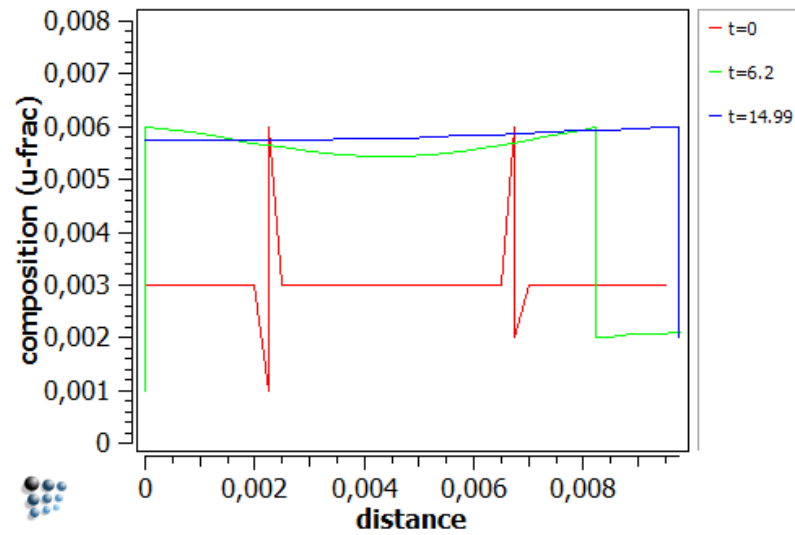
Interface 0:	
Carbon composition	
left:	0.001 uf
right:	0.006 uf
Velocity:	$-4 \times 10^{-4} \text{ m s}^{-1}$
Interface 1:	
Carbon composition	
left:	0.006 uf
right:	0.002 uf
Velocity:	$2 \times 10^{-4} \text{ m s}^{-1}$

**Table 4.6:** Parameters of interface escaping example

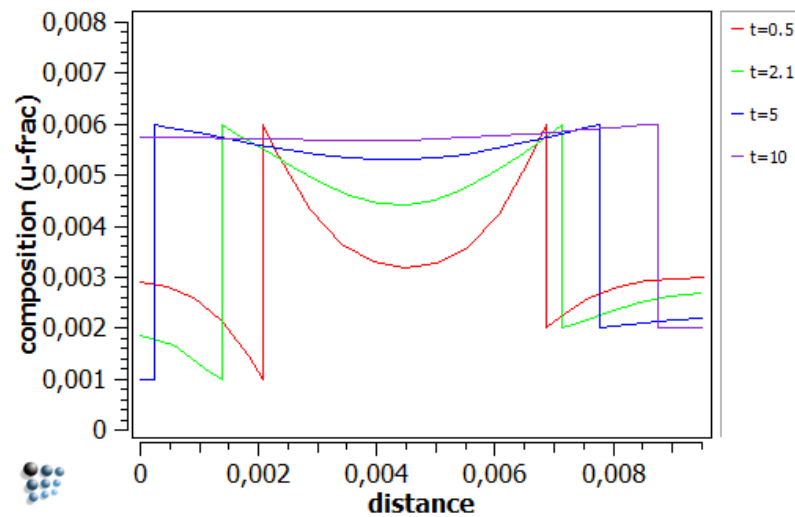
The behavior of time step width is shown in figure 4.36. Although the diffusion limit plays a role in the very beginning, the faster interface slows down the calculation progress the most, during its existence. After it disappears, the second interface is the limiting factor. The two big drops mark the regions, where one interface leaves the sample and the cell size gets very small on one side. Intermediate drops are caused by the increasing and decreasing of cell sizes between the neighbor cell changes. The slower interface has fewer intermediate drops, because it influences the time step limit for a shorter time.

#### Local equilibrium velocity

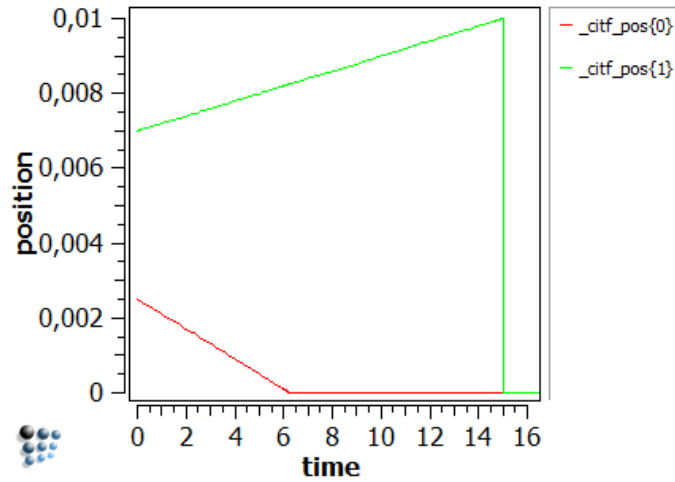
A simple calculation, containing the velocity calculation according to the local equilibrium (LEQ) method, is carried out. The interface starts at the left border of the sample with a predefined carbon composition, and moves through the sample conforming to the composition gradients on each side of the interface (see equation (4.33)). At the



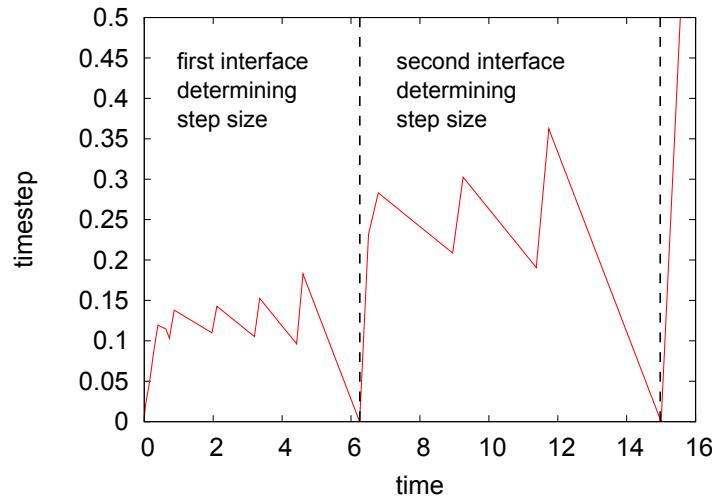
**Figure 4.33:** Interfaces escape the sample: At the beginning and right before each interface leaves



**Figure 4.34:** Interfaces leave the sample: Intermediate time steps



**Figure 4.35:** Interface positions during simulation



**Figure 4.36:** Time steps during simulation

beginning of the simulation, the composition gradients on each side of the interface are very high, thus the interface velocity will decrease with advancing time, when the average phase composition approaches the interface values. The flux on the left side of the interface only plays a role until this side's composition is flattened out, which is already the case at  $t = 3$  s. Afterwards, only the right composition gradient influences the interface velocity. Start and end state of the composition are shown in figure 4.37, the intermediate states in figure 4.38. Both the velocity and the interface position have been plotted over the simulation time (figures 4.40 and 4.39). Their shape is very typical for this kind of example and agrees with the results found in literature.

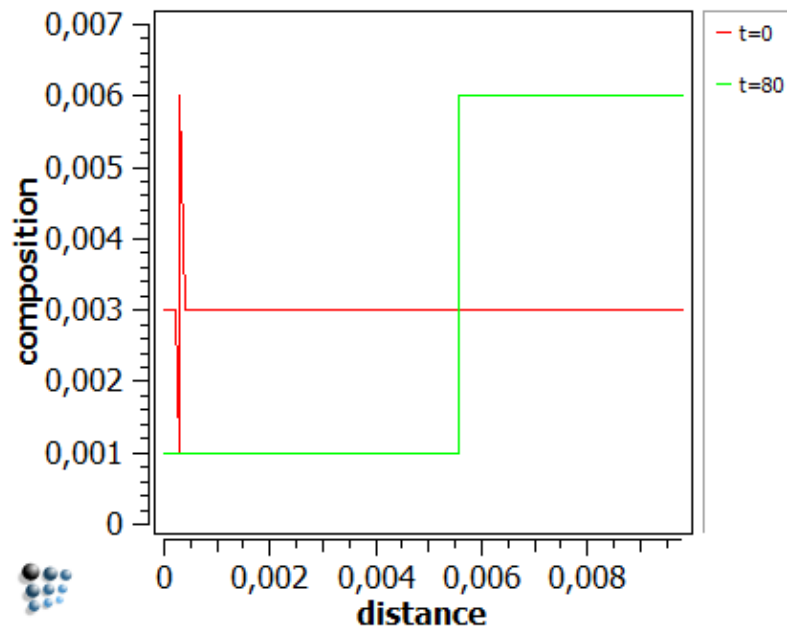


Figure 4.37: LEQ example: Composition at start and end

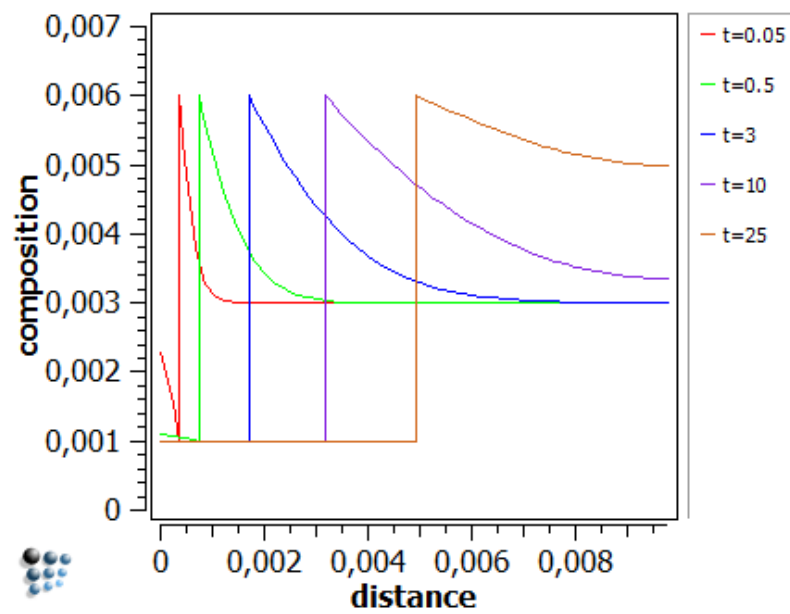


Figure 4.38: LEQ example: Composition at intermediate steps

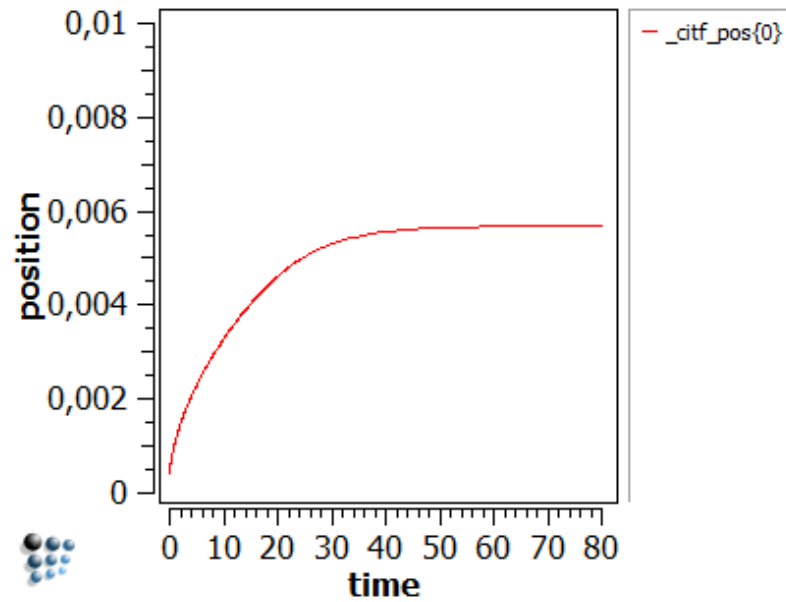


Figure 4.39: LEQ example: Interface position

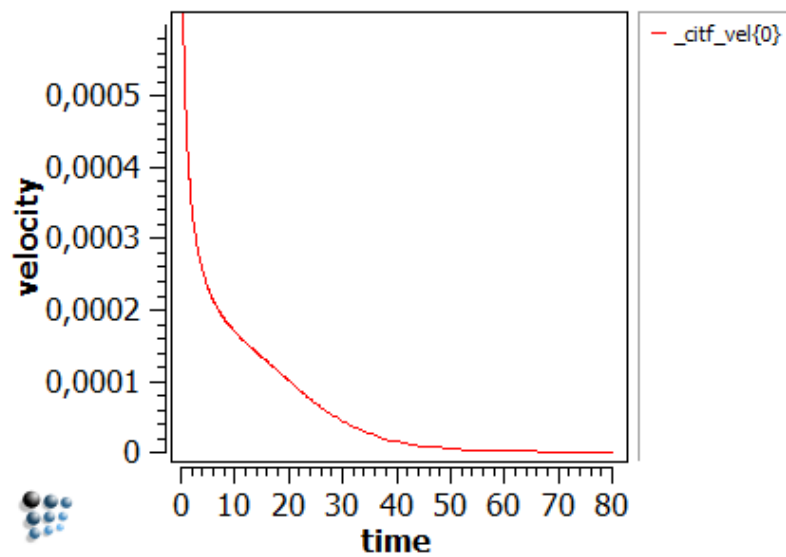


Figure 4.40: LEQ example: Interface velocity



Sample size:	1 cm
Number of cells:	50
Temperature:	800 °C
Carbon composition	
whole sample:	0.003 uf
Diffusion coefficient:	$10^{-6} \text{ m}^2 \text{ s}^{-1}$
Elements:	Fe,C
Phases:	fcc, bcc
Cell Material:	
fcc:	1..49
bcc:	0..1
Geometry:	planar
Total simulation time:	80 s

**Table 4.7:** Simulation parameters of LEQ example

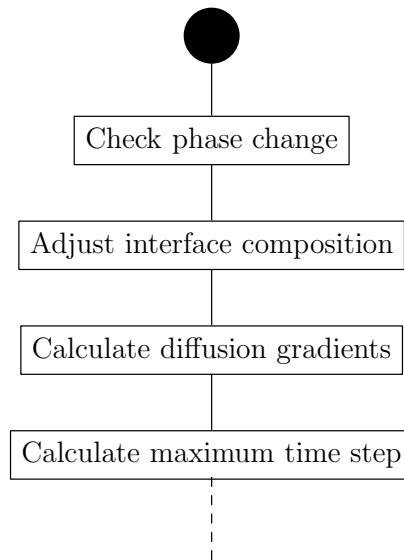
Interface 0:	
Carbon composition	
left:	0.001 uf
right:	0.006 uf
Velocity:	LEQ

**Table 4.8:** Interface parameters of LEQ example

### 4.3 Phase transformations involving moving interfaces

Phase transformations are investigated from scientists over decades. Like the classical phase transformation of water between a solid and a liquid phase, phase transformations in metals are a very interesting field for both researchers and industry, too. Unlike in [ML59], where the temperature of ice or water was the independent variable, in these simulations, the composition of elements in the phases is determined. In contrast to the moving boundary examples in the previous sections, where only artificial interface examples were simulated, now new phases can form and also existing phases can disappear. The basis for this behavior is the phase diagram. MatCalc performs an equilibrium calculation to determine the stable phases at the specific point in the diagram, which represents the current simulation properties. To simulate this type of phase transformation, heat treatments are utilized and the interface and diffusion algorithms are modified. In a simple example, a linear heat treatment is used to simulate a phase transformation in an iron carbon alloy system.

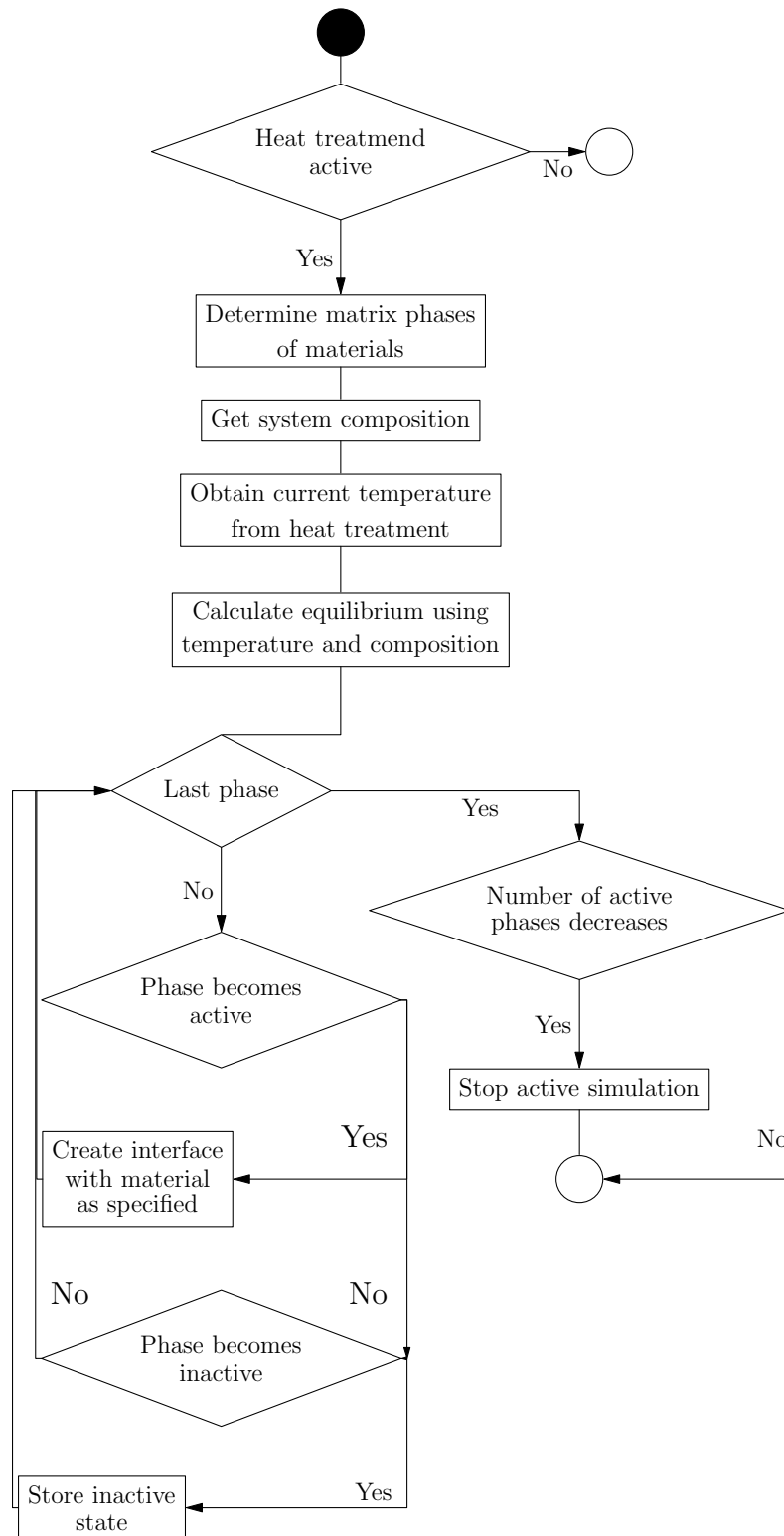
Prior to every diffusion step, the current temperature of the heat treatment is evaluated. At this temperature, an equilibrium calculation is carried out, to check if new phases become stable or existing phases disappear. Since the temperature also influences the equilibrium composition of the phases, the interface composition must be adjusted. The time step limit for heat treatments ensures that no phase formation is skipped. The flowchart in figure 4.21 is extended to describe the simulation of phase transformations, as shown in figure 4.41.



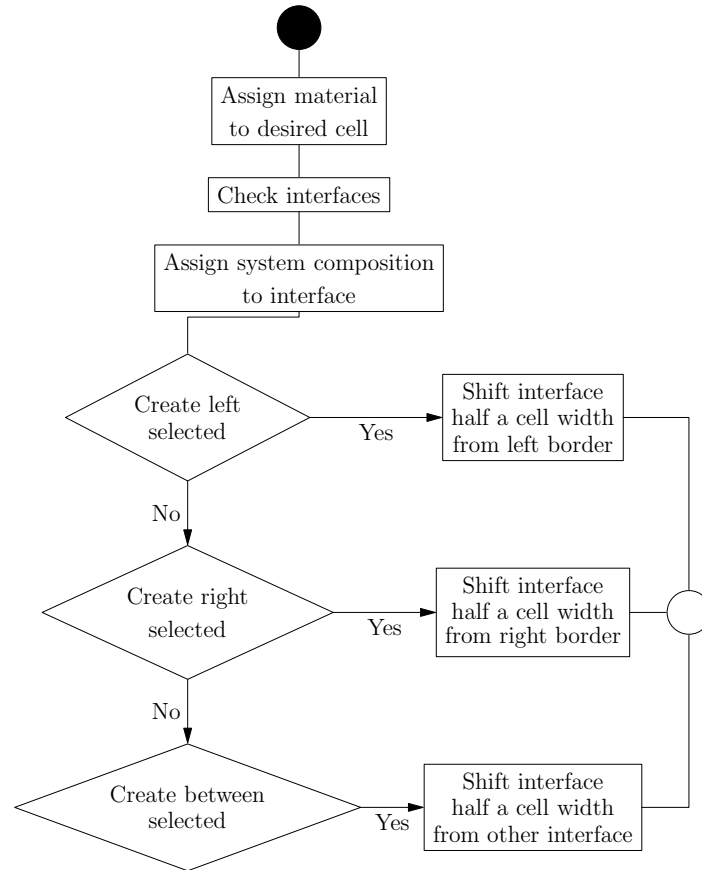
**Figure 4.41:** Extended flowchart of cell simulation

The ‘Check phase change’ process determines the matrix phases in all materials, retrieves the system composition and performs an equilibrium calculation. As parameter for the calculation, the temperature of the heat treatment is used. By comparing the phase status of the current and the previous step, changes in phase activity are recognized. In case of a phase becoming active, an interface has to be created at a predefined position. Before starting the simulation, the user has to specify the position, where a new phase should be created, upon its activation. Possible selections are the left/right sample boundary and the gap between two phases. If a phase becomes active, an interface is created according to the user preferences either on one of the sample boundaries or between two phases.

In the process ‘Create interface with material as specified’, the material related to the activated phase is set to the cell that comprises the new phase. To assure the validity of the interfaces, the process again invokes the ‘Check interface’ function of figure 4.12. In this case, the function creates a new interface object in MatCalc. Apart from temporarily setting the interface composition to the system composition, the interface position is shifted either half of a cell size from the boundary, or half of a cell

**Figure 4.42:** Flowchart of the 'Check phase change' process

size from the previously existing interface to avoid numerical problems arising from a tiny cell size. The process described above, is shown in more detail by figure 4.43.



**Figure 4.43:** Flowchart of the "Create interface with material as specified" process

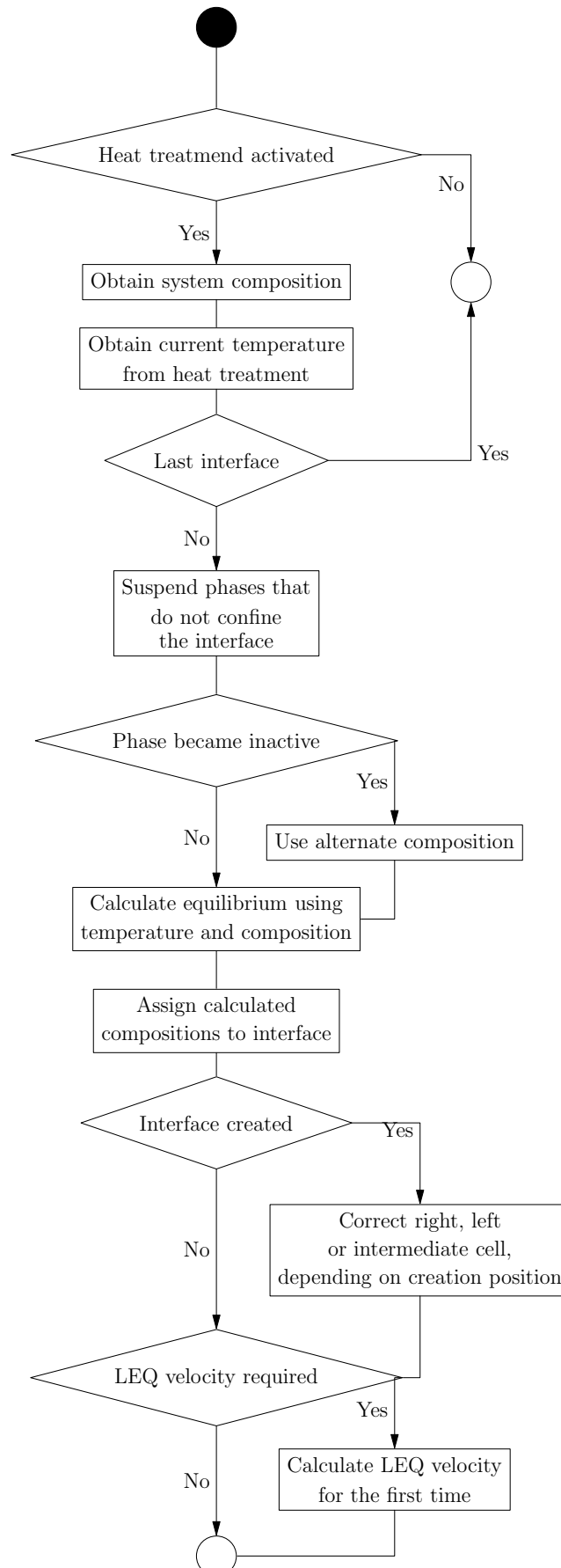
In these phase transformation simulations, the interface compositions have to be adjusted at each step of the temperature evolution. This is performed by the next step, the 'Adjust cell composition' process. The composition values are adapted to represent the corresponding equilibrium compositions in the phase diagram. This means that the interface compositions always follow the boundary lines in the phase diagram. In the algorithm implemented in this work, the appropriate compositions are determined from an equilibrium calculation. For each interface, only the neighboring phases are considered in this calculation. By default, the system composition is assumed as starting condition, but in the special case of the peritectic reaction, an alternative composition is used to be able to correctly reproduce the expected behavior. The alternative composition is not used for every equilibrium calculation, it is only needed when a phase becomes inactive, for example at the peritectic horizontal. If required, the value of the alternative composition must be specified, prior to starting the calculation,

for the desired interface. As soon as a phase gets inactive, the alternate composition will be used, instead of the system composition, in the equilibrium calculations for this specific interface. Furthermore, it is important for this process to know if a phase becomes inactive. As soon as the number of total active phases reduces, it stops the simulation. This behavior is implemented, because the adjustment of equilibrium compositions at the interface requires a two-phase region in the phase diagram. For this reason, the simulation is stopped in the ferritic solidification example, where the process advances from a two phase region to a one phase region.

After the composition of the interface was adjusted, it may be required to correct the neighbor composition, in order to obtain a valid interface velocity. Such a case occurs if an interface has just been created in the previous "Check phase change" step. As example, if an interface is created at the left boundary, the composition of the left neighbor cell must be adapted to the left interface composition value. This will ensure a valid gradient on the left side of the interface, for the velocity calculation. Afterwards, the interface velocity is calculated the first time for the new interface.

#### 4.3.1 Ferritic solidification in iron-carbon alloys

The ferritic solidification of an iron-carbon alloy has been simulated with a linear heat treatment, where the sample cools down from 1550 °C to 1500 °C in fifty seconds. Initially, a low carbon composition of  $9 \times 10^{-4}$  uf (approximately 0.019 35 wp) is assigned to the whole sample, which consists of a homogenous liquid phase. Additionally it is specified that the bcc/liquid interface should form on the left side of the specimen. The system composition and the temperature set the origin for the transformation in the phase diagram (see figure 4.45). After starting the simulation, the temperature drops at 1 °C/s and an equilibrium calculation is performed at each time step to recognize the formation of new phases. At approximately 1536 °C, the bcc phase becomes thermodynamically stable and active and an interface is created. The composition at the interface is determined by the equilibrium calculation and can be extracted from the phase diagram by using the lever rule. Compared to the example of the local equilibrium velocity, in this case, the interface is not slowing down with time. Instead, its velocity increases due to the fact that the interface composition gradients rise during the heat treatment. At 1529 °C, the liquid phase becomes inactive and the simulation is stopped, as intended, when the total number of active phases decreases, which means that the simulation leaves the two-phase equilibrium region. Figure 4.48 shows the histogram for both interface compositions and the interface velocity, whereas figure 4.46 contains the composition distribution in the sample at some characteristic time steps. A comparison of the interface position and the temperature over time can be found in figure 4.47. The parameters of simulation, heat treatments and phase creation are listed in tables 4.9 and 4.10.

**Figure 4.44:** Flowchart of the "Adjust interface composition" process

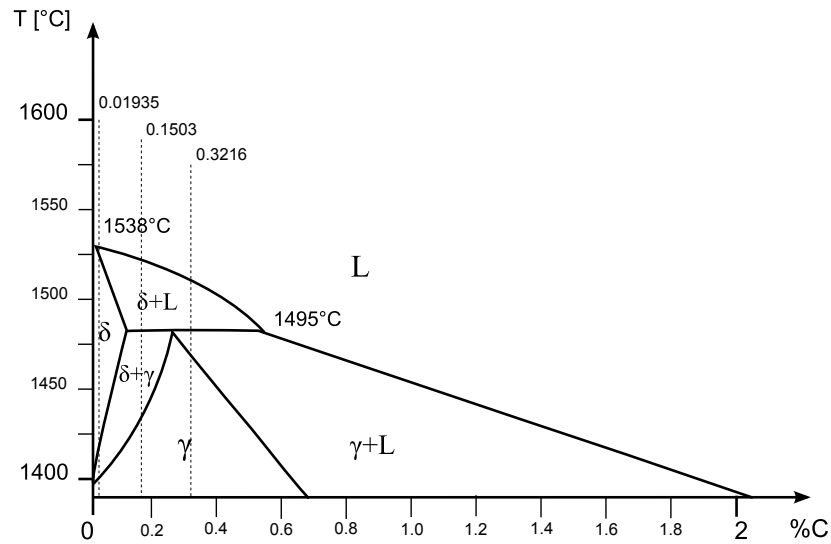


Figure 4.45: Iron carbon phase diagram for solidification

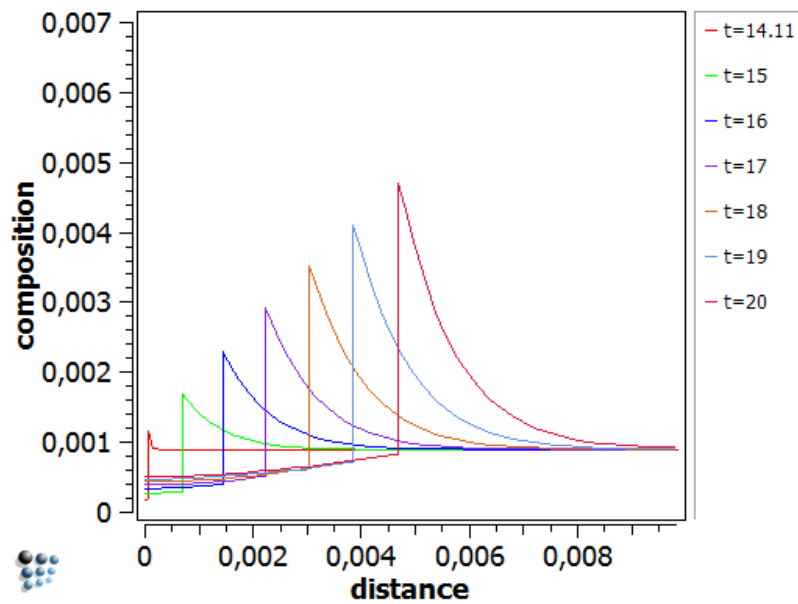
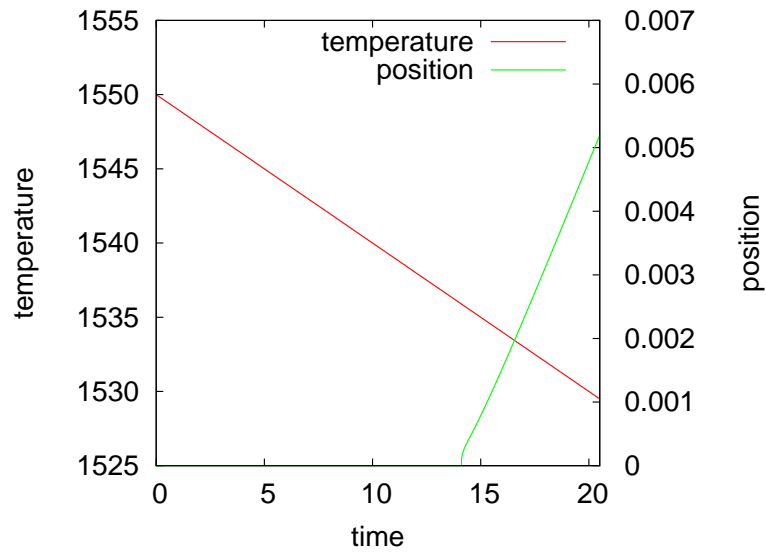
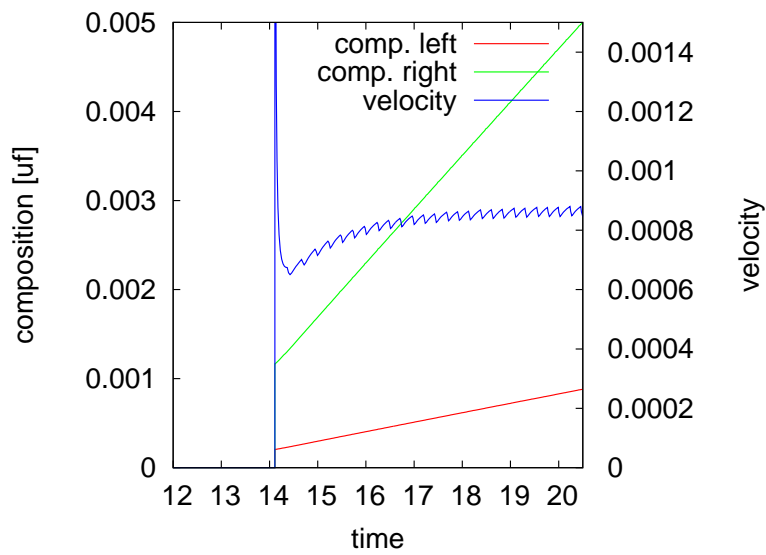


Figure 4.46: Composition for ferritic solidification example



**Figure 4.47:** Temperature and interface position for ferritic solidification example



**Figure 4.48:** Interface composition for ferritic solidification example



Sample size:	1 cm
Number of cells:	50
Temperature:	Heat treatment
Carbon composition	
whole sample:	0.0009 uf
Diffusion coefficient:	$10^{-6} \text{ m}^2 \text{ s}^{-1}$
Elements:	Fe,C
Phases:	bcc, liquid
Cell Material:	
liquid:	0..49
Geometry:	planar
Total simulation time:	$10^3 \text{ s}$

**Table 4.9:** Simulation parameters of ferritic solidification example

Heat treatment:	
1550 °C - 1500 °C:	50 s
Interface creation:	
bcc:	left

**Table 4.10:** Heat treatment and phase-creation parameters of solidification example

### 4.3.2 Peritectic reaction involving ferrite and austenite

The peritectic reaction is simulated in a similar way as the previous example, except that a different system composition and heat treatment are used. A composition of 0.015 uf or 0.3216 wp is already in the range of the peritectic region. Simulation parameters are available in table 4.11 and heat treatment/phase-creation parameters in table 4.12.

Starting from 1530 °C, the bcc phase becomes active at a temperature of 1511 °C, which is reached after approximately seven seconds. While the temperature constantly drops, the composition of the bcc/liquid interface, on both sides, increases until the fcc phase is activated, at 1494 °C. Concurrently, bcc becomes inactive and therefore, MatCalc begins to use the alternate composition. The phase specification defines the creation between the existing phases, which is considered when the second interface forms, at 13.66 s.

Examining the system composition and temperature, the process is now in a region of austenite and liquid. For obvious reasons, this composition can not be utilized to calculate the interface compositions for the bcc/fcc interface. In this example, the alternate composition specifies the value that is used for the equilibrium calculation

of the second interface. Taking the alternate composition of 0.007 uf or 0.1503 wp, it is possible to gain an interface movement that fits to the behavior of a phase getting inactive. The new interface leaves the sample at approximately 17.63 s. With decreasing temperature, also the liquid phase becomes inactive and the simulation stops at 1467 °C. Figures 4.49 and 4.50 show the composition in the sample, prior and after the formation of the second interface, at different characteristic time steps.

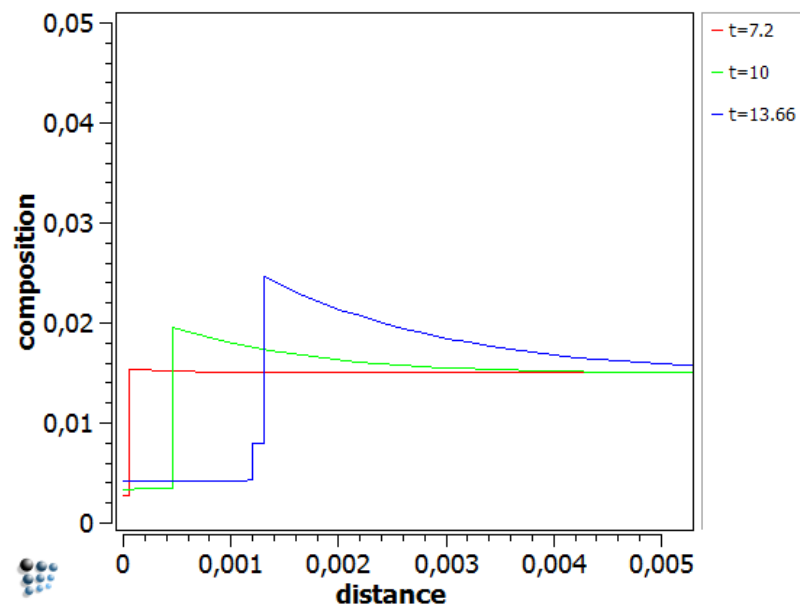
An interesting comparison of the interface positions and the temperature can be found in figure 4.51. The time evolution of the interface compositions, which increase for the first interface and decrease for the second, is shown in figure 4.52. Please note, that the numbers in brackets next to the variable names are used, to specify the interface. In the solidification example this was omitted, because the first interface is used by default. Interface velocities are in the region of some hundreds of micro meters (see figure 4.53).

Sample size:	1 cm
Number of cells:	50
Temperature:	Heat treatment
Carbon composition	
whole sample:	0.015 uf
Diffusion coefficient:	$10^{-6} \text{ m}^2 \text{ s}^{-1}$
Elements:	Fe,C
Phases:	bcc, fcc, liquid
Cell Material:	
liquid:	0..49
Geometry:	planar
Total simulation time:	$10^3 \text{ s}$

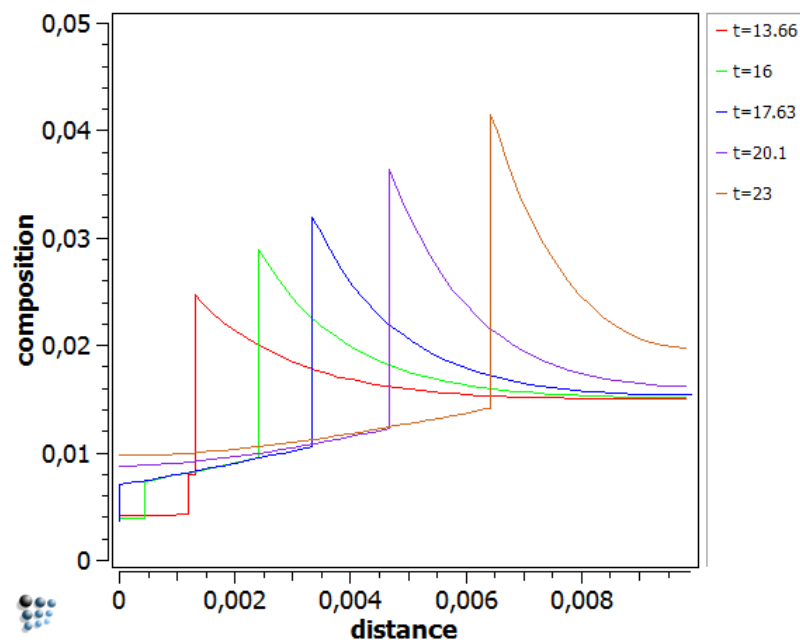
**Table 4.11:** Simulation parameters of peritectic example

Heat treatment:	
1530 °C - 1400 °C:	50 s
Interface creation:	
bcc:	left
fcc:	between
Alternate composition:	
Interface 1:	0.007 uf

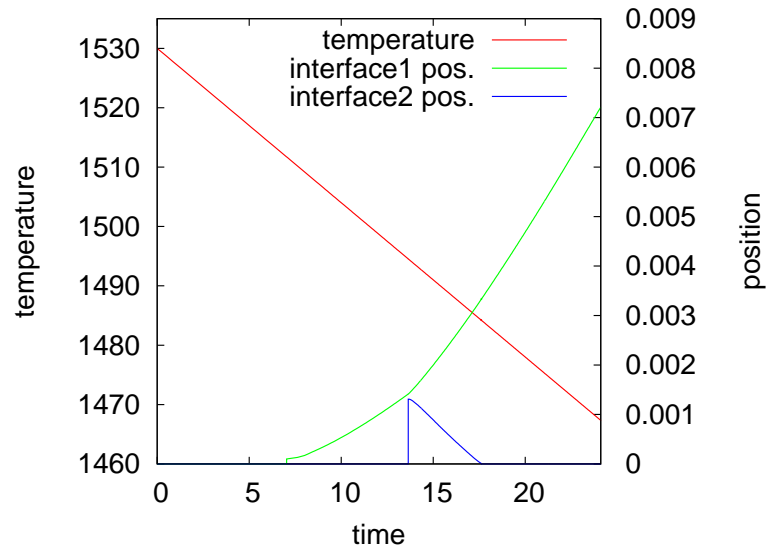
**Table 4.12:** Heat treatment and phase-creation parameters of peritectic example



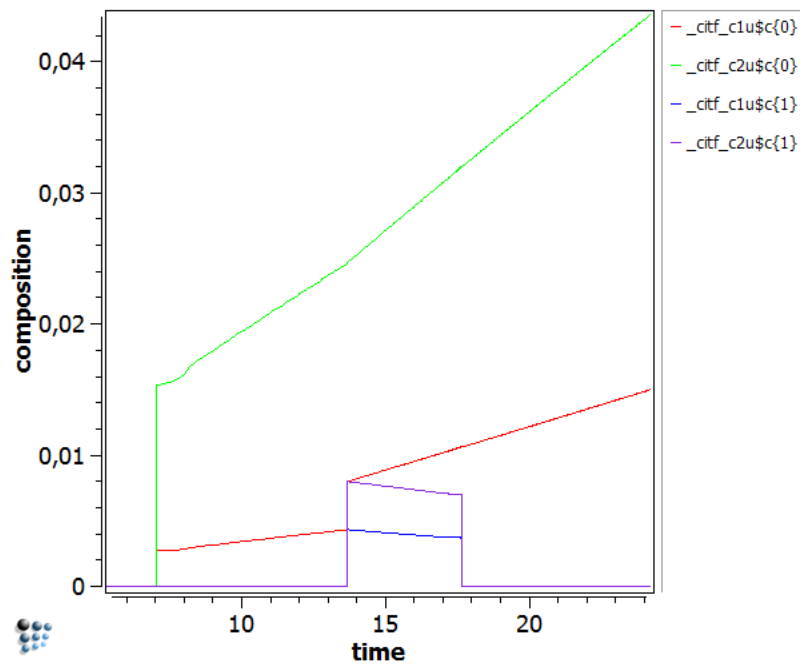
**Figure 4.49:** Peritectic example: Composition in sample prior to second interface formation



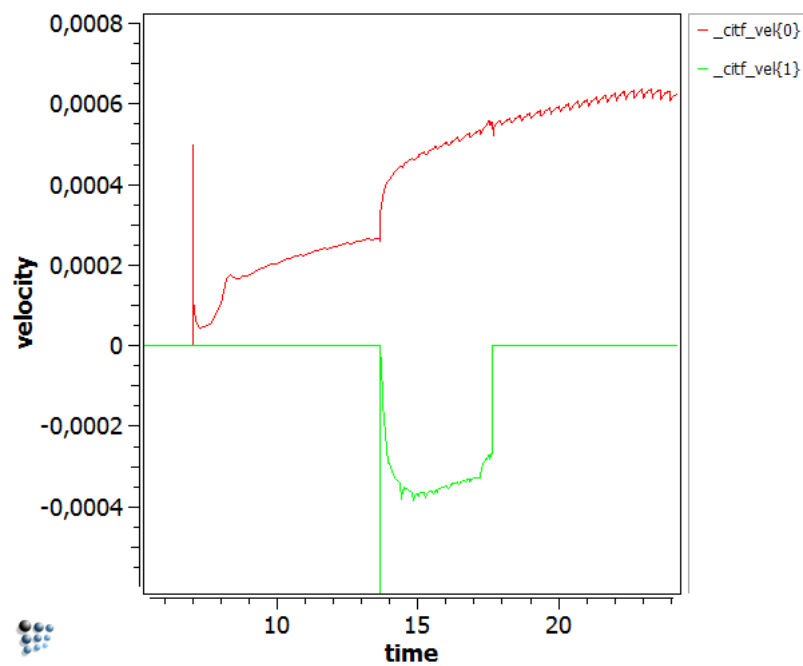
**Figure 4.50:** Peritectic example: Composition after second interface formation



**Figure 4.51:** Peritectic example: Development of interface positions and temperature



**Figure 4.52:** Peritectic example: Development of interface compositions



**Figure 4.53:** Peritectic example: Histogram of interface velocities

## 5 Discussion

The diffusion model of MatCalc was verified by solving a diffusion-couple example and comparing the numeric results to the analytic solution, which were in good agreement. Compared to the Murray Landis method, the model implemented in MatCalc supports multiple interfaces through deriving the cell velocity from the previous and current cell positions instead of the interface velocity. An additional improvement is the cell exchange, required to simulate the movement of interfaces over longer distances and to avoid difficulties that arise when an interface approaches the sample border.

In order to improve the accuracy of the cell exchange, a simple interpolation mechanism could be implemented. In the solidification examples, the initial velocity of an interface could be improved by determining the required cell composition, which causes a vanishing velocity. Instead of using a constant value for the alternate composition, it is advisable to use values along a path running parallel to the boundary line in the phase diagram.

In order to simulate problems of higher complexity, an additional velocity-calculation algorithm is required, which supports multi component systems. However, this is not a trivial task and such models often require approximations to simplify this task. If an extension to multiple space dimension is needed, the finite element method is for sure worth a consideration, although the implementation efforts would be higher. This would make modeling of curved interfaces possible and therefore enhance the scope of the simulation module.

## 6 Summary

In the theoretical part of the present thesis, the macroscopic and the microscopic view of diffusion are explained and a link between them is established through the Einstein equation. Afterwards, several descriptions of moving boundary problems are introduced and analytic problems presented. Numerical solution-methods are needed, because the analytic problems only exist for simple problems. Three major types of numerical methods are shown to get an overview of the scientific research that has been performed in the past.

In the practical part, the diffusion model of the microstructural process simulation software MatCalc is described and the example of a diffusion couple is used to verify its results. Further presented diffusion examples are a carburization process and the famous Darken experiment, where the simulated values are compared to values measured by Darken. In order to model moving phase boundaries, an advanced Murray Landis algorithm of a variable spaced grid, is implemented. A neighbor-cell-exchange mechanism is utilized, to circumvent numerical problems that arise when an interface leaves the sample.

Since the time step width is variable, several limits are obeyed to improve accuracy of calculation. The limits are dependent on composition gradients, interface velocity and the external heat treatment parameters. Two different models of composition treatment are compared, where one uses fixed composition neighbor cells, and the other uses virtual interface cells. The local equilibrium approach calculates the interface velocity using the composition gradients on both interface sides. Some examples showing general interface behavior and the local equilibrium model are demonstrated. A modification of the diffusion equation, for describing moving phase boundaries, is the Murray Landis correction, which is generalized to support multiple interfaces. After the correction is applied, heat treatments are introduced to be able to simulate phase transformations in metals. Additional steps required by the phase transformations are described, which include phase-states observation and interface composition adjustment during heat treatment. In the end, an example of iron-carbon solidification and a peritectic reaction involving ferrite and austenite are presented.

# Bibliography

- [AA92] J. O. Andersson and J. Agren: *Models for numerical treatment of multicomponent diffusion in simple phases*.  
Journal of Applied Physics, 72(4):1350–1355, 1992, ISSN 0021-8979.  
(cited on page 42)
- [BJ79] R. Bonnerot and P. Jamet: *A third order accurate discontinuous finite element method for the one-dimensional Stefan problem*.  
Journal of Computational Physics, 32(2):145–167, 1979, ISSN 0021-9991.  
(cited on page 25)
- [CFL67] R. Courant, K. Friedrichs, and H. Lewy: *On the partial difference equations of mathematical physics*.  
IBM Journal of Research and Development, 11(2):215–234, 1967.  
(cited on pages 39 and 59)
- [CG72] John Crank and R. S. Gupta: *A method for solving moving boundary problems in heat flow using cubic splines or polynomials*.  
IMA Journal of Applied Mathematics, 10(3):296–304, 1972, ISSN 0272-4960.  
(cited on page 23)
- [CIK<sup>+</sup>92] Sabine Crusius, Gerhard Inden, Ursula Knoop, Lars Höglund, and John Ågren: *On the Numerical Treatment of Moving Boundary Problems*.  
Zeitschrift für Metallkunde, 83(9):673–678, 1992.  
(cited on page 67)
- [CN96] John Crank and Phyllis Nicolson: *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type*.  
Advances in Computational Mathematics, 6(1):207–226, 1996.  
(cited on page 39)
- [Cra57] John Crank: *Two methods for the numerical solution of moving-boundary problems in diffusion and heat flow*.  
The Quarterly Journal of Mechanics and Applied Mathematics, 10(2):220, 1957, ISSN 0033-5614.  
(cited on pages 19 and 27)



- [Cra75] John Crank: *The Mathematics of Diffusion*.  
Oxford University Press, USA, 1975, ISBN 0-19-853411-6.  
(cited on page 7)
- [Cra87] John Crank: *Free and Moving Boundary Problems*.  
Oxford University Press, USA, 1987, ISBN 0-19-853357-8.  
(cited on page 7)
- [Dar49] LS Darken: *Diffusion of carbon in austenite with a discontinuity in composition*.  
Trans. AIME, 180:430–438, 1949.  
(cited on page 44)
- [DC70] R. C. Dix and J. Cizek: *The isotherm migration method for transient heat conduction analysis*.  
Heat Transfer, 1, 1970.  
(cited on page 31)
- [DG55] Jim Douglas and T.M. Gallie: *On the numerical integration of a parabolic differential equation subject to a moving boundary condition*.  
Duke Mathematical Journal, 22(4):557–572, 1955.  
(cited on page 21)
- [DG65] B. E. Deal and A. S. Grove: *General relationship for the thermal oxidation of silicon*.  
Journal of Applied Physics, 36(12):3770–3778, 1965, ISSN 0021-8979.  
(cited on page 15)
- [EHI<sup>+</sup>46] N. R. Eyres, D. R. Hartree, J. Ingham, R. Jackson, R. J. Sarjant, and J. B. Wagstaff: *The calculation of variable heat flow in solids*.  
Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, 240(813):1–57, 1946, ISSN 1364-503X.  
(cited on page 33)
- [Ell80] C.M. Elliott: *On a variational inequality formulation of an electrochemical machining moving boundary problem and its approximation by finite element method*.  
Journal of the Institute of Mathematics and its Applications, 25:121–131, 1980.  
(cited on page 27)
- [Fur77] R. M. Furzeland: *The numerical solution of two-dimensional moving boundary problems using curvilinear co-ordinate transformations*.  
Brunel University Mathematics Technical Papers collection, 1977.  
(cited on page 29)

- [Jac04] Kenneth A. Jackson: *Kinetic Processes: Crystal Growth, Diffusion, and Phase Transitions in Materials*.  
Wiley-VCH, 2004, ISBN 3-527-30694-3.  
(cited on page 13)
- [JRK<sup>+</sup>07] Koenraad Janssens, Dierk Raabe, Ernst Kozeschnik, Mark Miodownik, and Britta Nestler: *Computational materials engineering: an introduction to microstructure evolution*.  
Elsevier Academic Press, 2007.  
(cited on page 6)
- [Lan50] H. G. Landau: *Heat conduction in a melting solid*.  
Quarterly of Applied Mathematics, 8(1):81–94, 1950.  
(cited on page 27)
- [Lot60] M. Lotkin: *The calculation of heat flow in melting solids*.  
Quarterly of Applied Mathematics, 18:79–85, 1960.  
(cited on page 27)
- [ML59] William D. Murray and Fred Landis: *Numerical and machine solutions of transient heat-conduction problems involving melting or freezing*.  
J. Heat Transfer. 81C, pages 106–112, 1959.  
(cited on pages 22, 50, 65, 67, and 77)
- [Obe76] W. L. Oberkampf: *Domain mappings for the numerical solution of partial differential equations*.  
International Journal for Numerical Methods in Engineering, 10(1):211–223, 1976, ISSN 1097-0207.  
(cited on page 29)
- [PTVP07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery: *Numerical Recipes: the art of scientific computing*.  
Cambridge University Press, 2007.  
(cited on page 37)
- [RW12] B. Riemann and H. Weber: *Die partiellen Differentialgleichungen der mathematischen Physik*, volume 2.  
F. Vieweg und Sohn, 5th edition, 1912.  
(cited on page 16)
- [SS75] N. Shamsundar and E. M. Sparrow: *Analysis of multidimensional conduction phase change via the enthalpy model*.  
Journal of Heat Transfer, 97:333–340, 1975.  
(cited on page 34)
- [Ste89] Joseph Stefan: *Über einige Probleme der Theorie der Wärmeleitung*.

- Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften Wien,  
98:473–484, 1889.  
(cited on page 7)
- [SWA82] A. D. Solomon, D. G. Wilson, and V. Alexiades: *A mushy zone model with an exact solution*.  
Letters in Heat and Mass Transfer, 9(4):319–324, 1982, ISSN 0094-4548.  
(cited on page 10)
- [TJRC53] W. A. Tiller, K. A. Jackson, J. W. Rutter, and B. Chalmers: *The redistribution of solute atoms during the solidification of metals*.  
Acta metallurgica, 1(4):428–437, 1953, ISSN 0001-6160.  
(cited on page 17)
- [TW63] L. Thomas and J. Westwater: *Microscopic study of solid-liquid interfaces during melting and freezing*.  
Heat transfer-Houston, pages 155–164, 1963.  
(cited on page 10)

# List of Figures

3.1	Stefan condition for two phase problem . . . . .	8
3.2	Eutectic phase diagram of an alloy . . . . .	11
3.3	Moving boundary problem in two dimensions . . . . .	12
3.4	Oxygen diffusion into silicon . . . . .	15
3.5	Moving boundary problem with diffusion in liquid phase only . . . . .	19
3.6	Approximation of values on a fixed grid . . . . .	20
3.7	Deformation of a one dimensional grid according to Murray Landis method	23
3.8	Deformation of a one dimensional grid according to Crank Gupta method	24
3.9	Finite element discretization of a one dimensional grid according to Bonnerot and James . . . . .	26
3.10	Example of a grid transformation using body fitted coordinates . . . . .	29
3.11	Grid transformation with body fitted coordinates as used in the Furze- land example . . . . .	31
3.12	IMM transformed grid . . . . .	32
3.13	Enthalpy function in the sample with a jump at the boundary . . . . .	33
4.1	Comparison of calculated solution and error function . . . . .	41
4.2	Composition in the sample at different time steps using a fixed surface composition boundary condition . . . . .	43
4.3	Composition in the sample at different time steps using a fixed cell composition boundary condition . . . . .	43
4.4	Manganese and silicon content in the sample prior to simulation start .	45
4.5	Carbon content in the sample from simulation start to end of uphill stage	45
4.6	Manganese and silicon content beyond 10 days of heat treatment . . . .	46
4.7	Carbon content beyond 10 days of heat treatment . . . . .	46
4.8	Interface is located inside a cell . . . . .	48
4.9	Region of interface is using smaller cell spacing . . . . .	49
4.10	Interface is exactly between two cells, no cells displaced . . . . .	49
4.11	Interface is now on an intermediate point, cells are displaced . . . . .	49
4.12	Flowchart of the "Check interface" process . . . . .	50
4.13	Grid displaced in a sub-optimal way . . . . .	51
4.14	Cell distribution before and after a neighbor change . . . . .	51
4.15	Important definitions of interface nomenclature . . . . .	51

4.16	Flowchart of interface cell and neighbor determination . . . . .	53
4.17	Flowchart of existing-interface checks . . . . .	54
4.18	Flowchart of material correction . . . . .	55
4.19	Flowchart of cell displacement . . . . .	56
4.20	Flowchart of composition correction . . . . .	57
4.21	Flowchart of cell simulation . . . . .	58
4.22	Flowchart of perform interface step . . . . .	59
4.23	Flowchart of diffusion limited step size determination . . . . .	60
4.24	Flowchart of interface limited step size determination . . . . .	62
4.25	Comparison of interface composition handling . . . . .	64
4.26	Half distance compared to half cell size . . . . .	65
4.27	Comparison of distance handling for LEQ method . . . . .	66
4.28	Interpolation of composition values for the Murray Landis correction - curved-shaped composition . . . . .	68
4.29	Interpolation of composition values for the Murray Landis correction - straight composition . . . . .	69
4.30	Interface merge example: Composition at start and merge . . . . .	71
4.31	Interface merge example: Composition during simulation . . . . .	71
4.32	Cell distribution for interface merge example . . . . .	72
4.33	Interfaces escape the sample: At the beginning and right before each interface leaves . . . . .	73
4.34	Interfaces leave the sample: Intermediate time steps . . . . .	73
4.35	Interface positions during simulation . . . . .	74
4.36	Time steps during simulation . . . . .	74
4.37	LEQ example: Composition at start and end . . . . .	75
4.38	LEQ example: Composition at intermediate steps . . . . .	75
4.39	LEQ example: Interface position . . . . .	76
4.40	LEQ example: Interface velocity . . . . .	76
4.41	Extended flowchart of cell simulation . . . . .	78
4.42	Flowchart of the 'Check phase change' process . . . . .	79
4.43	Flowchart of the "Create interface with material as specified" process .	80
4.44	Flowchart of the "Adjust interface composition" process . . . . .	82
4.45	Iron carbon phase diagram for solidification . . . . .	83
4.46	Composition for ferritic solidification example . . . . .	83
4.47	Temperature and interface position for ferritic solidification example . .	84
4.48	Interface composition for ferritic solidification example . . . . .	84
4.49	Peritectic example: Composition in sample prior to second interface formation . . . . .	87
4.50	Peritectic example: Composition after second interface formation . . .	87
4.51	Peritectic example: Development of interface positions and temperature	88
4.52	Peritectic example: Development of interface compositions . . . . .	88

---

4.53 Peritectic example: Histogram of interface velocities . . . . .	89
--	----

# List of Tables

4.1	Simulation parameters of diffusion couple example . . . . .	42
4.2	Simulation parameters of carburization example . . . . .	44
4.3	Simulation parameters of Darken example . . . . .	47
4.4	Simulation parameters of interface collision example . . . . .	70
4.5	Interface parameters of collision example . . . . .	70
4.6	Parameters of interface escaping example . . . . .	72
4.7	Simulation parameters of LEQ example . . . . .	77
4.8	Interface parameters of LEQ example . . . . .	77
4.9	Simulation parameters of ferritic solidification example . . . . .	85
4.10	Heat treatment and phase-creation parameters of solidification example	85
4.11	Simulation parameters of peritectic example . . . . .	86
4.12	Heat treatment and phase-creation parameters of peritectic example . .	86

# Acknowledgment

I would like to thank Ernst Kozeschnik for his patience and motivation during the different stages of bringing this thesis to life. Furthermore i am grateful to Peter Mohn for his continuing help and support.

Without the caring love and support of my parents, this work would have never been possible. I want to express my special thanks to Bettina Jakopitsch for sharing the past years with me.

Financial support by the Austrian Federal Government (in particular from the Bundesministerium für Verkehr, Innovation und Technologie and the Bundesministerium für Wirtschaft, Familie und Jugend) and the Styrian Provincial Government, represented by Österreichische Forschungsförderungsgesellschaft mbH and by Steirische Wirtschaftsförderungsgesellschaft mbH, within the research activities of the K2 Competence Centre on ‘Integrated Research in Materials, Processing and Product Engineering’, operated by the Materials Center Leoben Forschung GmbH in the framework of the Austrian COMET Competence Centre Programme, is gratefully acknowledged.