

Maximilian Reisner

Interdiffusion and Phase Formation at Iron-Tungsten Interfaces

IPP 2017-09
November 2017



Abschlussarbeit im Masterstudiengang Physik

Interdiffusion and Phase Formation at Iron-Tungsten Interfaces

Maximilian Reisner

02.11.2017

Erstgutachter (Themensteller): Prof. Dr. Ulrich Stroth

Zweitgutachter:

Betreuer: Dr. Martin Oberkofler

Contents

1. Introduction	1
2. Mathematical description of diffusion	5
2.1. Basic diffusion equations	5
2.2. The Boltzmann-Matano method for assessment of $D(c)$	6
2.3. Numerical solutions	8
2.3.1. Constant diffusion coefficient D	8
2.3.2. Concentration dependent diffusion coefficient $D(c)$	11
2.3.3. Reactive diffusion simulations	16
3. Analysis techniques	25
3.1. Scanning electron microscopy	25
3.2. Rutherford backscattering spectrometry	26
3.3. Sputter XPS depth profiling	30
4. Sample preparation	31
4.1. Polishing and recrystallization	31
4.2. Deposition of tungsten	33
4.3. Annealing	35
5. Assessment of interdiffusion and phase growth	37
5.1. Effective $D(C)$ from Rutherford backscattering spectrometry	37
5.1.1. Measured RBS spectra	37
5.1.2. Obtaining concentration profiles	39
5.1.3. The concentration dependence of the diffusion coefficient	43
5.1.4. The temperature dependence of the diffusion coefficient	47
5.2. Interdiffusion and phase growth studied by focussed ion beam cross sectioning	51
5.2.1. The composition of the intermediate phase	52
5.2.2. The growth rate of the intermediate phase	58
5.2.3. Grain boundary diffusion	62
6. Summary	65

7. Conclusion	67
8. Acknowledgements	69
A. Attachments	71
A.1. Discretization of the Differential equations	71
A.2. Algorithms used to solve the linear equations	74
A.3. Source code for simulations with constant D	77
A.4. Source code for simulations with concentration dependent $D(c)$	82
A.5. Source code for simulations with reactive diffusion	88
A.6. Raw Data for resulting $D_0(c)$	105

Abstract

Low-activation steels are possible candidates for wall materials in a future nuclear fusion power plant. Through a process called preferential sputtering, an enriched tungsten layer is expected to develop on these steels, lessening erosion and thus increasing their lifetime and reducing contamination of the fusion plasma. However, the process of preferential sputtering may be countered by interdiffusion of tungsten and iron. Therefore, the interdiffusion coefficient between iron and tungsten is assessed in this thesis.

This is done by annealing a binary W-Fe system and analyzing the resulting concentration profiles by means of Rutherford backscattering spectrometry (RBS) and focussed ion beam cross sectioning. From the depth profiles measured by means of RBS, effective interdiffusion coefficients are obtained via the Boltzmann-Matano method and through forward simulations. Between temperatures of 900 and 1100 K, the functional dependence of this effective interdiffusion coefficient on concentration and temperature can be described by

$$D(c, T) = D_0(c) \exp\left(-\frac{2.7 \text{ eV}}{k_B T}\right),$$

with $D_0(c)$ decreasing from $(1.2 \pm 0.7) \cdot 10^{-5} \frac{\text{m}^2}{\text{s}}$ in the W tracer limit to $(3.5 \pm 0.8) \cdot 10^{-7} \frac{\text{m}^2}{\text{s}}$ at a W concentration of 70 at.%. This result is consistent with the pre-existing literature values at very low concentrations of tungsten.

For temperatures above 1000 K, the formation of an intermediate phase (most likely Fe_2W) was observed by means of focussed ion beam cross sectioning and X-ray photoelectron spectroscopy. The growth rate of this phase at various temperatures has been investigated and the interdiffusion coefficient inside this phase was determined to be $(12 \pm 2) \cdot 10^{-19} \frac{\text{m}^2}{\text{s}}$ at 1050 K and $(42 \pm 7) \cdot 10^{-19} \frac{\text{m}^2}{\text{s}}$ at 1100 K. A simulation code to study interdiffusion in the presence of phase formation systems has been developed.

1. Introduction

Problems like global warming and finding suitable repositories for the nuclear waste from nuclear fission power plants have led humanity to the challenge of finding new ways of satisfying our ever increasing energy demand. Alternatives like renewable energy and hydroelectric power plants seem like an obvious choice. However, these are accompanied by other problems like taking up a lot of space, having to be built at specific spots to function as efficiently as possible, and being unreliable due to their strong dependence on time of day and weather. A better replacement for nuclear fission power plants, and power plants that burn fossil fuels, would be a power plant that is able to provide a reliable energy output of about 1 GW, similar to today's conventional power plants, that at the same time doesn't produce greenhouse gasses or radioactive waste, that would be needed to be stored for millennia. A possible candidate for such a power plant would be a nuclear fusion power plant.^[4]

A nuclear fusion power plant produces energy through fusing light elements, typically isotopes of hydrogen, into heavier elements, typically helium – similar to our sun. The currently most developed reactor solution, the TOKAMAK^[22], requires the hydrogen isotopes to be continuously kept in the state of a plasma with temperatures in the range of 10^8 K. Only at those temperatures can the positively charged hydrogen nuclei overcome the repulsive Coulomb barrier and come close enough to one another to undergo a fusion reaction.

Since no material exists that could withstand such high temperatures, the plasma needs to be contained by a magnetic field. Some particles, however, will always escape and hit the wall of the fusion reactor. Naturally, this wall should be made from materials that can withstand such a particle bombardment for as long as possible, and even more importantly release as little contaminants as possible into the plasma. Another constraint on such a wall material is, that it needs to have very little retention and permeation of hydrogen isotopes, in particular the radioactive and expensive tritium; for safety reasons as well as for ensuring a cost-efficient operation.

One of the candidate materials for being used in some recessed areas of the main chamber wall for the proposed first nuclear fusion power station DEMO, is a low-activation steel called EUROFER^[7], which consists predominantly of the elements iron and chromium with also some low percentage of tungsten. While steel is a good choice of material for building big constructions, it is not well suited to withstand the heat and particle bombardment expected on the inside of a fusion reactor.

A material that is more suitable, is pure tungsten, due to its high melting point as well as high atomic mass and enthalpy of sublimation, which result in a very low sputter yield.^[17] Ideally, one would coat the complete inside of the vessel of a nuclear fusion reactor with tungsten. This, however is technically difficult because the coating tends to break off of the underlying steel. Through a process called preferential sputtering, EUROFER could in principle have the capability to create its own protective tungsten layer^[16]: When bombarded with high-energy particles, the iron on the surface of EUROFER is expected to be much more likely to be sputtered away than the tungsten, due to its higher sputter yield. After some fluence, all the iron in the surface would be gone, leaving a thin layer of tungsten behind which would provide the desired protection, i.e. a reduced sputtering yield (see figure 1.1).

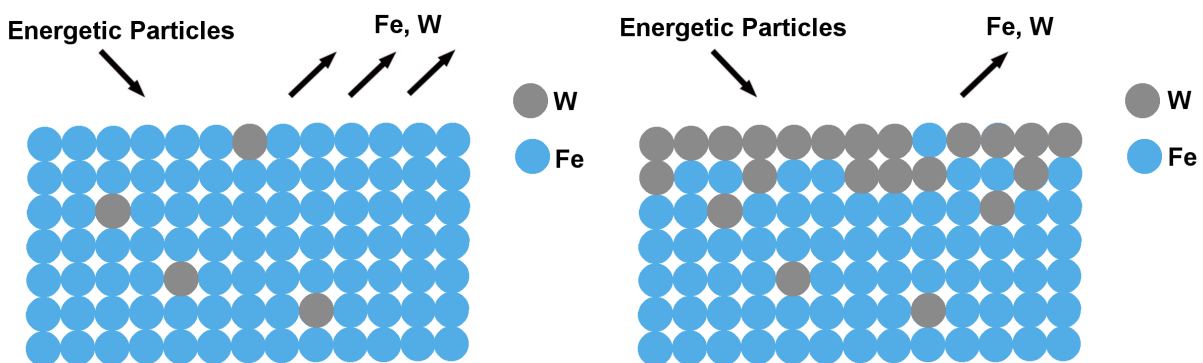


Figure 1.1.: Simplified depiction of EUROFER as a binary system of iron and tungsten. Through the principle of preferential sputtering, a protective layer of tungsten may be formed after a short amount of time, providing protection from the constant particle bombardment of the plasma.

One problem that might counteract this idealized process, is diffusion. Given the temperatures EUROFER is expected reach when bombarded with high energy particles, the desired gradient in concentration is likely to lead to a particle flux trying to get the system back to an equilibrium. Thus, tungsten from the surface would diffuse into the bulk material while iron from the bulk would diffuse to the surface, where it would ultimately be sputtered away again (see figure 1.2).

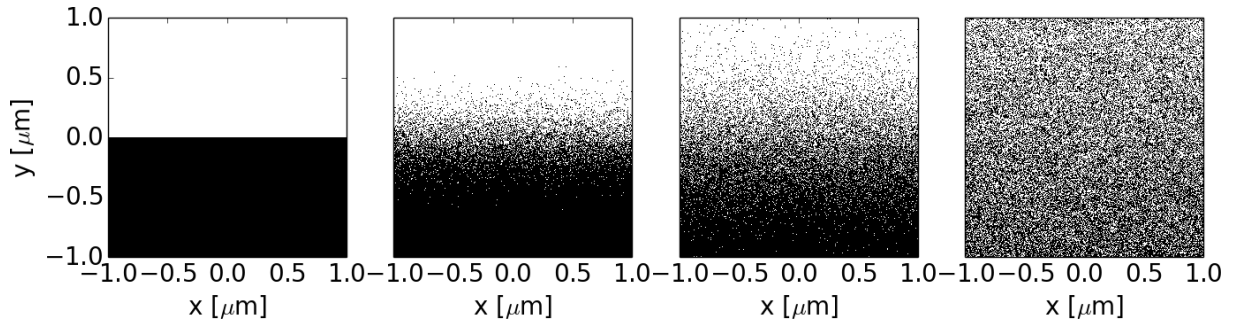


Figure 1.2.: Simulations of a general interdiffusion process. Beginning with a layered binary system, material from the bottom layer is brought back to the surface.

To figure out whether the expected diffusion is strong enough to negate the effects of preferential sputtering, simulations using the program SDTrimSP are proposed. In first simulations, tracer-diffusion coefficients have been used.^[19] For more accurate results, however, the concentration dependent interdiffusion coefficient between iron and tungsten needs to be known. First measurements with the aim of determining this interdiffusion coefficient have previously been conducted in the course of a bachelor's thesis.^[20] In this work we expand on these first measurements and create numerical tools that describe the observations and which may aid future SDTrimSP simulations.

To this end, tungsten has been sputter-deposited onto iron substrates and annealed at various temperatures and for different times. The resulting concentration depth profiles are then measured both with Rutherford-backscattering-spectrometry (RBS) and by viewing cross-sections, that were cut with focussed ion beams (FIB), in a Scanning electron microscope (SEM). Interdiffusion coefficients are determined from the RBS measurements by means of the Boltzmann-Matano method and by comparing the measured concentration profiles with concentration profiles obtained through simulations. The rate of phase growth observed in the cross sections is quantified and the interdiffusion coefficient in the intermediate phase is determined from this.

2. Mathematical description of diffusion

In this chapter, the mathematical description of various diffusive processes and their numerical implementation is presented and it is explained how diffusion coefficients can be extracted from measured data. The SDTrimSP-simulations, that these measurements are done for, require an one-dimensional diffusion coefficient. Therefore, lateral variations e.g. due to potential dependencies on grain orientation, are not taken into account and are averaged over. The equations in the following sections are consequently also one dimensional.

2.1. Basic diffusion equations

Whenever there is a local gradient in concentration and an absence of external forces, there is a flux of particles trying to get the system into an equilibrium. This can be described using Ficks first law^[10]:

$$J = -D(c) \frac{\partial n}{\partial x}, \quad (2.1.1)$$

where J is the particle flux, i.e. the number of particles that flow through a certain area within a certain time, n the number density as a function of space (depth) and D the diffusion coefficient, a proportionality factor relating those two measurands. In general, D can depend on the concentrations c of the interdiffusing elements.

Using the continuity equation $\frac{\partial n}{\partial t} = -\frac{\partial J}{\partial x}$ and 2.1.1, one can derive Ficks second law

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} \left(D(c) \frac{\partial c}{\partial x} \right), \quad (2.1.2)$$

which is generally the equation used to determine diffusion coefficients from measured depth profiles. Here, we have replaced the number density n with the relative concentration $c = \frac{n}{\rho}$, which we can do under the assumption that the total density ρ of our sample does not depend on the composition.

Generally, the diffusion coefficient is not only dependent on the concentration, but also on the temperature. The temperature dependence can very often be described by an Arrhenius behaviour

$$D(c, T) = D_0(c) \exp\left(-\frac{\Delta H}{k_B T}\right), \quad (2.1.3)$$

where ΔH is the activation enthalpy for interdiffusion and k_B the Boltzmann constant.

In this thesis, two different methods of determining the diffusion coefficient will be applied. The first one, described in section 2.2, is the Boltzmann-Matano method^[10], that is based on inverting equation 2.1.2. The second method consists in simulating diffusion processes and adjusting the input $D(c)$ to match the output depth profile to the measured one. In the subsections 2.3.1 and 2.3.2, the simulations will be based on equation 2.1.2. This method will be expanded in subsection 2.3.3 to systems with formation of intermediate phases.

2.2. The Boltzmann-Matano method for assessment of $D(c)$

Starting from Fick's second law 2.1.2, the substitution $\lambda(c) = \frac{x}{\sqrt{t}}$ is used, resulting in

$$-\frac{1}{2}\lambda \frac{dc}{d\lambda} = \frac{d}{d\lambda} \left(D(c) \frac{dc}{d\lambda} \right). \quad (2.2.1)$$

Integrating over λ yields

$$-\frac{1}{2} \int_{c^-}^c \lambda d\tilde{c} = D(\tilde{c}) \frac{d\tilde{c}}{d\lambda} \Big|_c - D(\tilde{c}) \frac{d\tilde{c}}{d\lambda} \Big|_{c^-}. \quad (2.2.2)$$

Here, c^- is a starting concentration defined by the boundary conditions.

Now, with the assumption that the concentration does not change with respect to x at the borders, $\frac{d\tilde{c}}{d\lambda} \Big|_{c^-} = \frac{d\tilde{c}}{d\lambda} \Big|_{c^+} = 0$, the Boltzmann-Matano equation 2.2.3 can be derived, by resubstituting λ and solving for D

$$D(c) = -\frac{1}{2t} \frac{1}{(d\tilde{c}/dx)_c} \int_{c^-}^c x d\tilde{c}. \quad (2.2.3)$$

It should be noted that, due to the condition that the derivatives $\frac{dc}{dx}$ must be 0 for $c = c^-$ and $c = c^+$, the denominator in the Boltzmann-Matano equation becomes 0 at the boundaries. To prevent the equation from diverging, the integral $\int_{c^-}^c x d\tilde{c}$ must also be 0 for $c = c^-$ and $c = c^+$. For c^- , this condition is automatically fulfilled; for c^+ one needs to choose the spatial coordinate system accordingly, which can be seen in figure 2.1. The new origin is called the Matano interface.

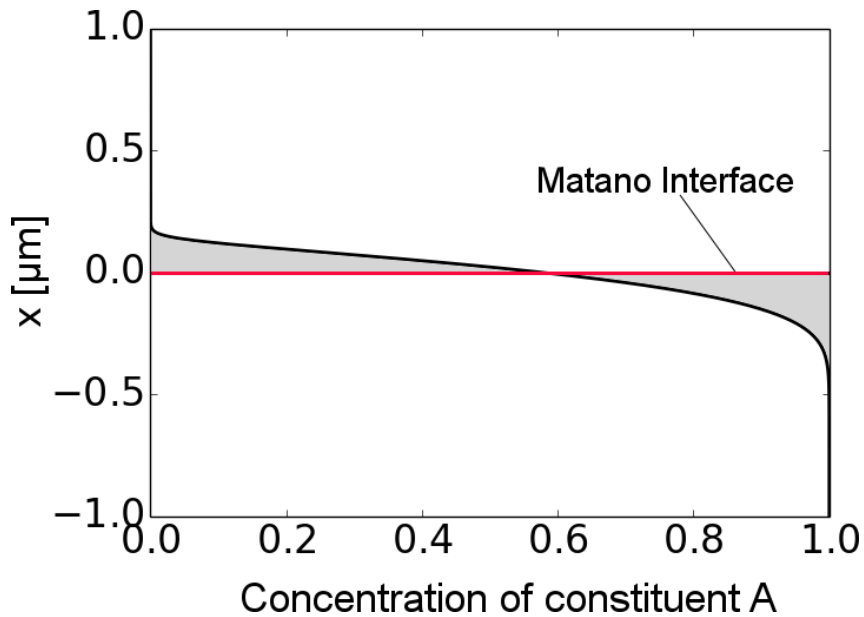


Figure 2.1.: Example of a depth profile after some interdiffusion (plotted as x over c) to illustrate the concept of the Matano interface. The original binary sample (before interdiffusion) consisted of a layer of a material B on top of a layer of a material A.

Even when choosing the coordinate system accordingly, to prevent the equation from diverging, one still divides very small numbers by each other at the extremes of c . As a result, for very high and very low concentrations, the accuracy of the Boltzmann-Matano method is not very good. Additionally, due to the fact, that the Boltzmann-Matano method requires the derivative of the concentration profile, it is not very exact if there are only few data points coarsely resolving the concentration profile.

Therefore, we require another way of determining the diffusion-coefficient. Such a way would be to simulate a diffusion process and adjusting the input $D(c)$ until the output concentration-profile matches the one that was measured. In the following section, the numerical implementation of interdiffusion simulations for three different cases (constant D , $D(c)$ and reactive-diffusion) will be detailed and the validity of these simulations will be tested.

2.3. Numerical solutions

This section is split into three parts. In section 2.3.1, diffusion with a constant diffusion coefficient will be simulated. This simple case is useful for understanding the underlying principles. The accuracy of these simulations can be tested by comparing them to an analytical solution that exists in this case, allowing to easily benchmark the numerical implementation. In section 2.3.2, simulations using a concentration dependent diffusion coefficient are presented and then cross-checked with results using the Boltzmann-Matano method. In section 2.3.3, the simulations take into account the formation of phases in the material, as opposed to considering the two constituents as being completely miscible at all compositions.

The simulations in the course of this work are done using Python 2.7.3.. The actual source codes can be viewed in appendices A.3,A.4 and A.5.

At the base of all our simulations of diffusion is Fick's second law 2.1.2, which will in the following be discretized, using a finite difference method.

2.3.1. Constant diffusion coefficient D

To better explain how the simulations used are implemented, and to verify their validity, we first simplify Fick's second law, by setting the diffusion coefficient constant, $D(c) = D$:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} \quad (2.3.1)$$

This is discretized by replacing the derivatives with difference quotients, using the Crank-Nicholson method^[3], which is numerically stable under any conditions and second-order accurate^[6]:

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \frac{D}{2} \left(\frac{c_{i+1}^{n+1} - 2c_i^{n+1} + c_{i-1}^{n+1}}{\Delta x^2} + \frac{c_{i+1}^n - 2c_i^n + c_{i-1}^n}{\Delta x^2} \right). \quad (2.3.2)$$

Here, time is discretized in N_T steps of size Δt , denoted by n , and space is discretized in N_L steps of size Δx , denoted by i . A more detailed description of how equation 2.3.1 is discretized and how to arrive at equation 2.3.2, can be found in appendix A.1.

For $\frac{\partial^2}{\partial x^2}c_0$ and $\frac{\partial^2}{\partial x^2}c_{N_L}$, the non-existent indices $i = -1$ and $i = N_L + 1$ would be required. Equation 2.3.2 is therefore only valid at the inner grid points from $i = 1$ to $i = N_L - 1$. For the concentrations at the grid points $i = 0$ and $i = N_L$, boundary conditions need to be set. In the course of this thesis, Dirichlet boundary conditions are used^[6]; Therefore c_0 and c_{N_L} are set to be fixed constants.

For each given time step n , the current concentration profile is assumed to be known, starting from an initial concentration profile. With the equations 2.3.2, the concentration profile for $n+1$ is then calculated subsequently for all following time-steps.

To do this, one needs to solve the equations 2.3.2 for the concentration profile at time step $n+1$. In matrix form, this system of equations looks like the following:

$$\left(\frac{1}{\Delta t} \mathbb{1} - \frac{D}{2\Delta x^2} M \right) \vec{c}^{n+1} = \left(\frac{1}{\Delta t} \mathbb{1} + \frac{D}{2\Delta x^2} M \right) \vec{c}^n + \frac{D}{\Delta x^2} \vec{c}_{BC} \quad (2.3.3)$$

This is now simply a linear equation of the form $A\vec{c}^{n+1} = \vec{b}$, that needs to be solved for \vec{c}^{n+1} . For a more detailed description, including the explicit form of the matrix M , see appendix A.1.

Solving equation 2.3.3 could, of course, be achieved by inverting the matrix A , but that is computationally expensive. Since the matrices we are dealing with in this case are always tridiagonal, we can solve the linear equation faster using the Thomas-Algorithm. A precise description on how this algorithm works can be found in appendix A.2. This algorithm only requires $O(N)$ steps, while inverting a matrix via gaussian elimination, for instance, would require $O(N^3)$ steps^[6].

In the following, we benchmark our simulation code against an analytic solution of the diffusion equation.

If the initial conditions are simple step-functions of the form

$$c(x, t = 0) = \begin{cases} c_{max}, & \text{if } x < 0 \\ c_{min}, & \text{if } x \geq 0 \end{cases},$$

our simplified diffusion equation

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

has the analytical solution

$$c(x, t) = \bar{c} + \frac{\Delta c}{2} \operatorname{erf} \left(\frac{x}{2\sqrt{Dt}} \right), \quad (2.3.4)$$

where $\bar{c} = \frac{c_{min} + c_{max}}{2}$ is the average concentration, and $\Delta c = \frac{c_{min} - c_{max}}{2}$ is the concentration 'amplitude'.^[3] With this it is possible to test the simulations, by comparing them to the analytical solution.

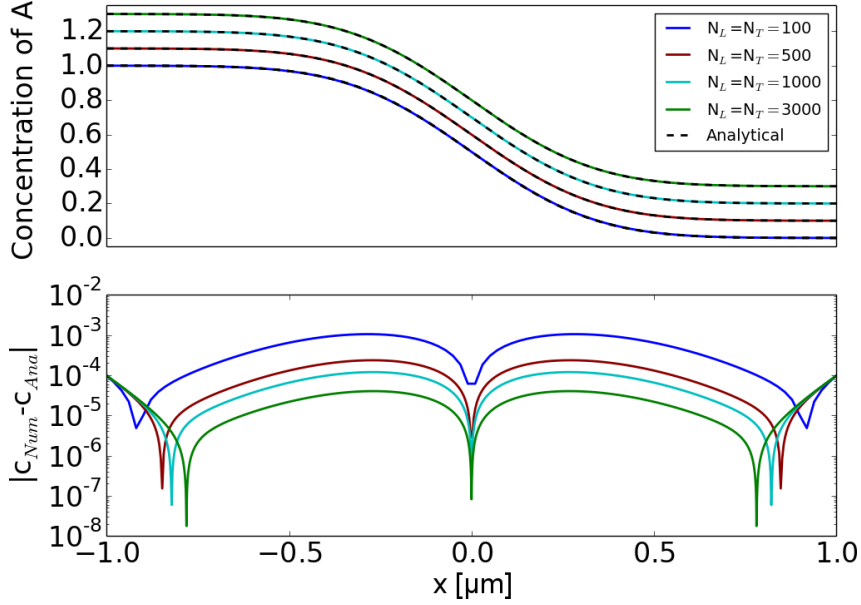


Figure 2.2.: Upper graph: Calculated and analytical concentration profiles after interdiffusion. The concentration profiles are stacked by adding multiples of 0.1. The diffusion coefficient used was $D = 10^{-19} \frac{m^2}{s}$, the diffusion time used was $T = 1$ h. The discretization was varied, from $N_L = N_T = 100$ to $N_L = N_T = 3000$. Lower graph: Absolute difference between the analytical and the numerically calculated solution.

As is shown in figure 2.2, when comparing the analytical solution to the numerically calculated one, one finds the error to be dependent on Δt and Δx . For a finer discretization, the error is reduced, but even for a very coarse discretization, the error is smaller than 1×10^{-3} , which is a sufficiently high accuracy for our purposes.

Since we can now be confident, that our simulations yield satisfactory results, we can move on to simulations with concentration dependent diffusion coefficients, for which analytical solutions in general don't exist.

2.3.2. Concentration dependent diffusion coefficient $D(c)$

In general, the interdiffusion coefficient can depend on the concentrations of the inter-diffusing elements. In that case, the simplification made in subsection 2.3.1 cannot be applied. Therefore, equation 2.1.2, Fick's second law

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} \left(D(c(x)) \frac{\partial c}{\partial x} \right)$$

has to be discretized, again using the Crank-Nicholson method

$$\frac{c_i^{n+1} - c_i^n}{\Delta t} = \frac{D_{i+1}^n \cdot \left(\frac{\partial c}{\partial x}\right)_{i+1}^{n+1} - D_{i-1}^n \cdot \left(\frac{\partial c}{\partial x}\right)_{i-1}^{n+1}}{8\Delta x^2} + \frac{D_{i+1}^n \cdot \left(\frac{\partial c}{\partial x}\right)_{i+1}^n - D_{i-1}^n \cdot \left(\frac{\partial c}{\partial x}\right)_{i-1}^n}{8\Delta x^2}. \quad (2.3.5)$$

Like in the previous subsection, equation 2.3.5 is only applicable to the indices $i=1$ through $i=N_L-1$. As Boundary Conditions, we have set c_0 and c_{L+1} to be constants, and the derivatives $\frac{\partial}{\partial x}c_0$ and $\frac{\partial}{\partial x}c_{L+1}$ to be 0. A more detailed description of the discretization can again be found in appendix A.1.

Note that the concentration dependent diffusion coefficient is taken at the time step n , for both terms of the Crank-Nicholson method. This is because the diffusion coefficient matrix itself depends on the concentration profile, generally in a nonlinear way. Solving this in an implicit way, like the Crank-Nicholson method does, would be very complicated and resource intensive. The error introduced this way can be kept at an acceptable level by choosing a sufficiently small Δt .

Then, like before in the case of a constant D , one needs to solve the system of equations 2.3.5 for the concentration values c_i^{n+1} . In matrix form, the system of equations can be written as:

$$\left(\frac{1}{\Delta t} \mathbb{1} - \frac{1}{8\Delta x^2} MD^n M \right) \vec{c}^{n+1} = \left(\frac{1}{\Delta t} \mathbb{1} + \frac{1}{8\Delta x^2} MD^n M \right) \vec{c}^n + \frac{1}{4\Delta x^2} MD^n \cdot \vec{c}_{BC} \quad (2.3.6)$$

(For a more detailed description of this equation, be referred to appendix A.1.)

Again, the problem reduces to solving a linear equation of the form $A\vec{c}^{n+1} = b$. In this case, the Thomas algorithm cannot be used; instead a different method is applied. A detailed description can be found in appendix A.2.

Our method of obtaining a solution to the diffusion equation, the Crank-Nicholson method is known to converge fast and to be unconditionally stable. However, it is also known to be prone to oscillations on top of the solution.^{[3][6]} In the previous section, these oscillations could be dampened by choosing Δx and Δt such, that the stability condition $D \cdot \frac{\Delta t}{\Delta x^2} < 0.5$ is fulfilled. However, when taking into account a concentration dependent diffusion coefficient, we seem to always get such oscillations, an example of which can be seen in figure 2.3.

To be able to use the results from these simulations better – fitting them to measured data or further evaluating the results with the Boltzmann-Matano method, for example – a binomially weighted running average is applied

$$\overline{c}_i^n = \frac{1}{4}(c_{i-1}^n + 2c_i^n + c_{i+1}^n), \quad (2.3.7)$$

resulting in smooth curves which can also be seen in figure 2.3. The two boundaries remain the constants they were set to during the calculation.

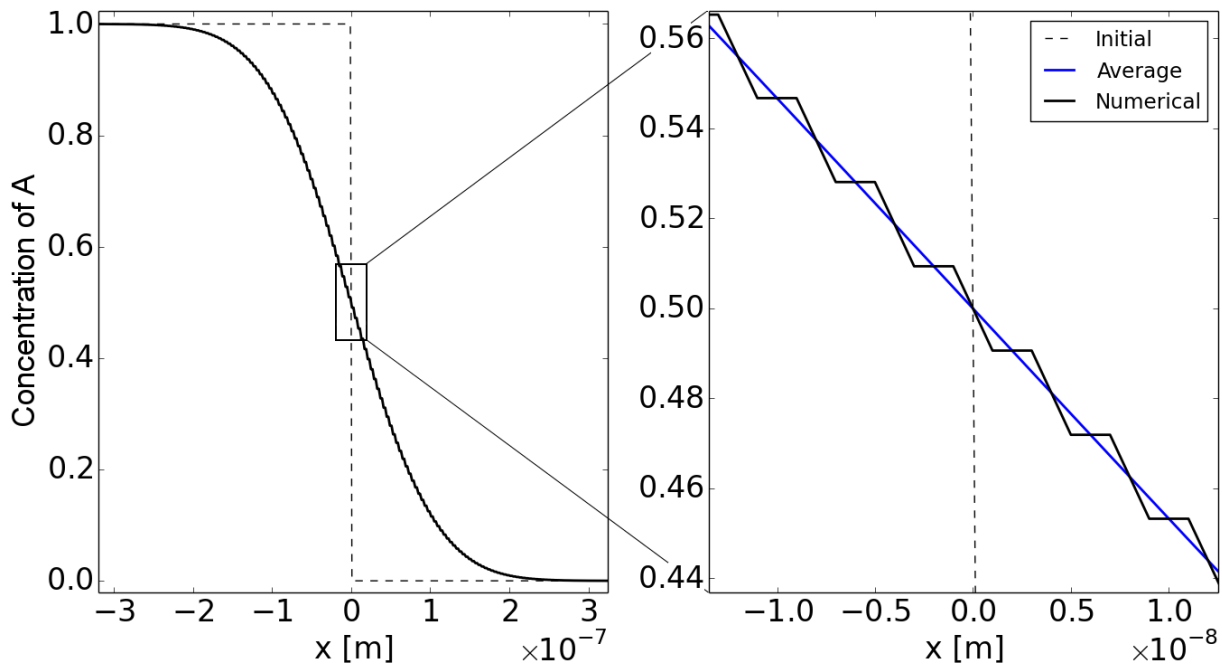


Figure 2.3.: Example of a depth profile resulting from the numerical solution of the diffusion equation with a concentration dependent interdiffusion coefficient. The initial depth profile is a step function centered at $x=0$. The right graph is a zoom-in to better visualize the occurring numerical oscillations and the result of the binomially weighted running average.

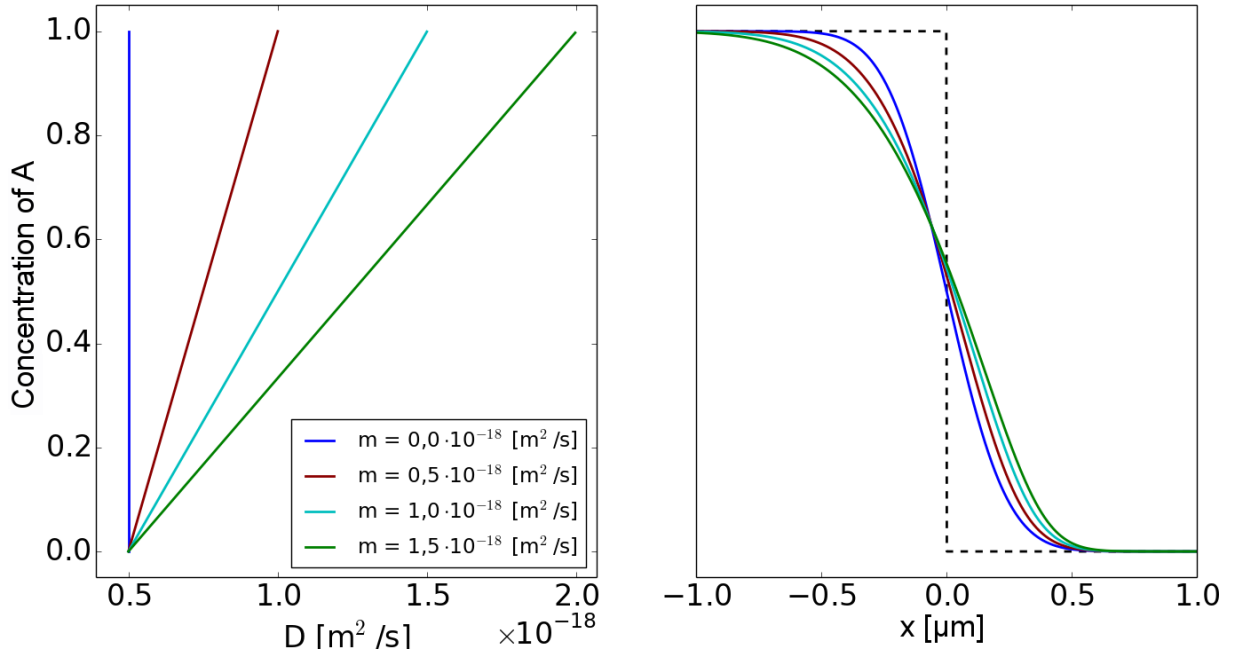
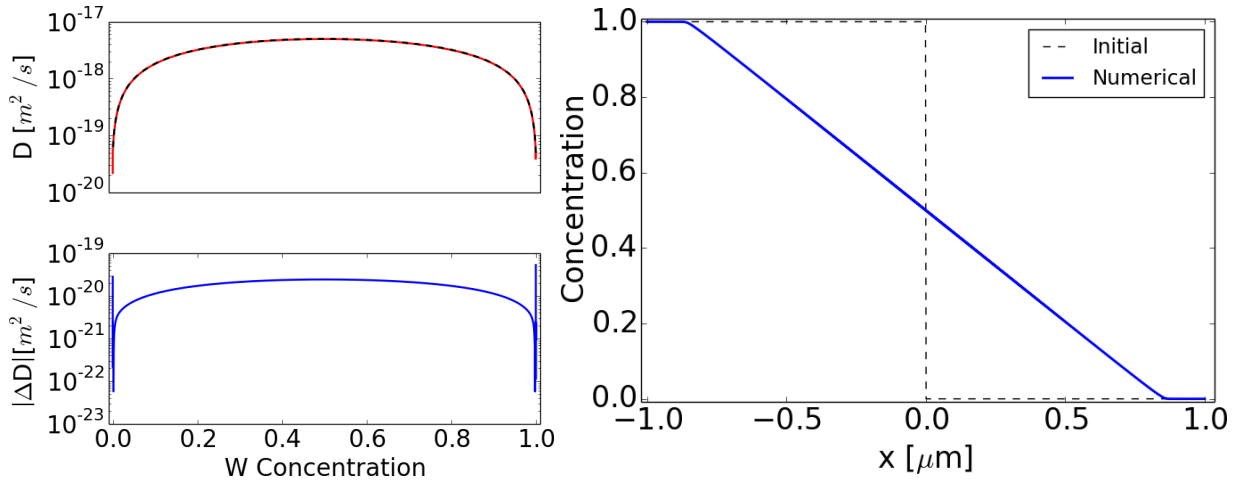


Figure 2.4.: Left graph: Different concentration dependent Diffusion coefficients of the form $0.5 \cdot 10^{-18} + m \cdot c$ m²/s, with m ranging between 0 and $1.5 \cdot 10^{-18}$ m²/s. The concentration was chosen to be on the y-axis, so a direct comparison with the right plot would be easier. Right graph: The resulting concentration profiles according to the forward simulations using a concentration dependent $D(c)$. Also drawn in is the initial concentration profile; a step-function.

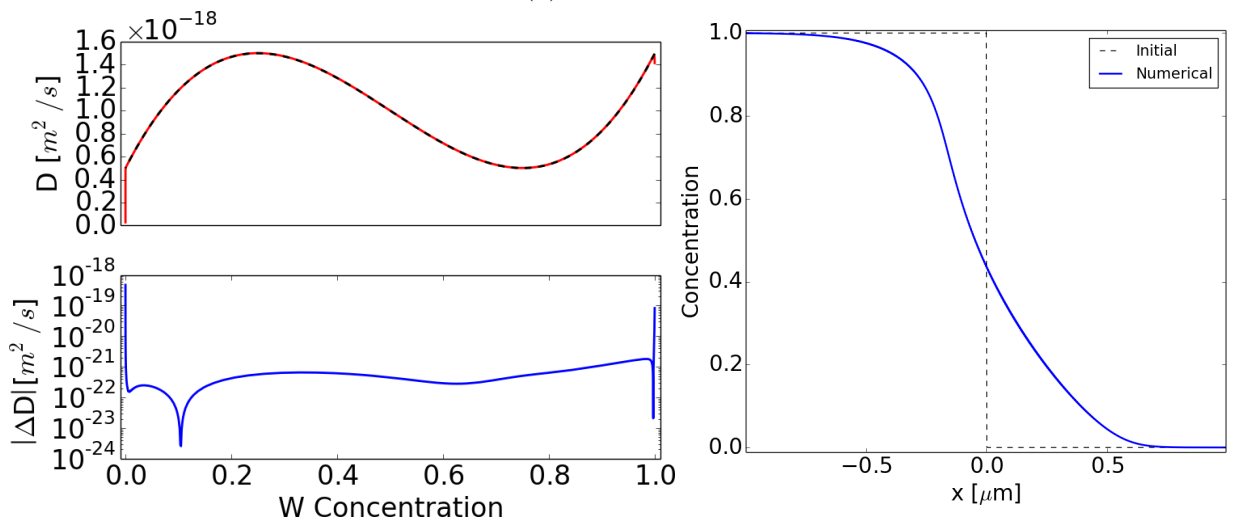
Illustrated in figure 2.4 are some results from simulations with concentration dependent diffusion coefficients showing the effects of bigger or smaller gradients in the diffusion coefficient. It can be seen that a bigger gradient leads to a more asymmetrical concentration profile, with the part where the diffusion coefficient is higher, being more 'smeared out'.

Diffusion with a concentration dependent diffusion coefficient generally does not have an analytical solution one could use to test these simulations. However, we can cross-check the simulations with the results from the Boltzmann-Matano method.

For this, we perform simulations for several different concentration dependencies of the diffusion coefficients and then apply the Boltzmann-Matano method on the resulting concentration profile. Figure 2.5 shows the diffusion coefficients that have been used for these tests. In the upper example in figure 2.5a, the given concentration dependence of the diffusion coefficient is an inverted parabola. The resulting concentration profile after some diffusion time (shown in figure 2.5a) is linear. The example in figure 2.5b shows that the method works well also for relatively complicated dependencies of the diffusion coefficient on concentration, in this case with two extrema at intermediate concentrations.



(a) Example a



(b) Example b

Figure 2.5.: Two examples of applying the Boltzmann-Matano method to the results of forward simulations with a concentration dependent $D(c)$. For example a, the diffusion coefficient used was a parabola ranging over two orders of magnitude: $D(c) = (0.5 - 198c(c - 1))10^{-19} \frac{m^2}{s}$ (upper left plot, black dashed curve). In example b, the diffusion coefficient used was a polynomial function of order three: $D(c) = (16c^3 - 24c^2 + 9c + 0.5)10^{-18} \frac{m^2}{s}$ (lower left plot, black dashed curve). The resulting concentration profiles are depicted on the right side, along with the initial conditions. The diffusion coefficients obtained from these concentration profiles are depicted in red, overlaid with the initially chosen one. Finally, the absolute differences between the initial diffusion coefficients and the results from the Boltzmann-Matano method are shown below the $D(C)$ -graphs. For both simulations, time and space were divided into 1000 grid points, respectively. The simulated diffusion time was 10 hours.

The diffusion coefficients retrieved by applying the Boltzmann-Matano method to the resulting depth profiles are overlaid to the initially chosen diffusion coefficients. Also shown in the figure is the absolute difference between between the initially chosen diffusion coefficients and the results from the Boltzmann-Matano method. The deviation lies several orders of magnitude below the actual value of the diffusion coefficients, except for extreme concentration values near 0 or 1. This, however, can be attributed to the Boltzmann-Matano method which does not work properly in those cases for the reasons described in section 2.2.

Other than for the extreme concentrations, the diffusion coefficients from the Boltzmann-Matano method are in good agreement with the diffusion coefficients that were used in the corresponding simulations. We can therefore conclude that our simulations, taking a concentration dependent $D(c)$ into account, work as expected.

2.3.3. Reactive diffusion simulations

Describing the diffusion the way we have done in the previous section, with a concentration dependent diffusion coefficient, is strictly speaking only possible in concentration ranges where both materials homogeneously mix. In section 5.1 this description is nevertheless applied to the FeW systems also for cases where we observe phase formation. The resulting $D(C)$ must in this case be interpreted as an 'effective' interdiffusion coefficient. A proper description of interdiffusion in the presence of phase formation is given in this section: Whenever a multi-component system features miscibility gaps or the components form phases, the correct way of describing the diffusion processes is a reactive diffusion approach.^[5]

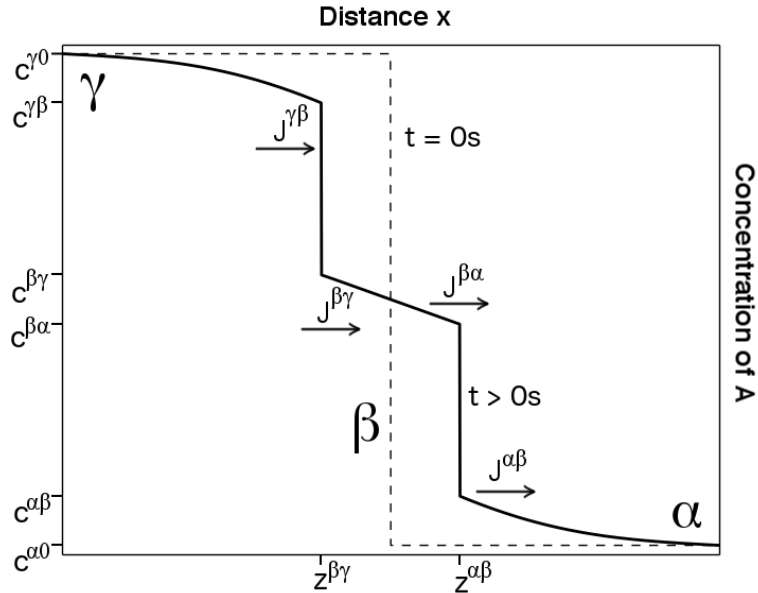


Figure 2.6.: Typical concentration profile of a constituent A in a two-component system when taking reactive diffusion into account. α and γ denote two pure materials of a binary system, β an intermediate phase that forms between those two. $z^{\alpha \beta}$ and $z^{\beta \gamma}$ denote the interface positions between those phases, the movement of which is determined by the diffusion within the phases. $c^{\alpha 0}$, $c^{\alpha \beta}$, etc. denote the solubility ranges of each phase. These are often available from phase diagrams. $J^{\alpha \beta}$, $J^{\beta \alpha}$, $J^{\beta \gamma}$ and $J^{\gamma \beta}$ are the fluxes right at the interfaces. They determine the rate at which the interfaces move and the phases grow.

We assume that the phase formation at the interface is not the rate-limiting process. In this case, the propagation of the interface is completely dependent on the diffusion within the individual phases and we can simulate the diffusion within the individual phases as described in the previous sections. In figure 2.6, an example of this can be seen for the case of a two-component system: The concentration of the constituent A is plotted versus depth. α , β and γ represent three phases, in each of which diffusion is simulated as described in the previous sections.

As boundary conditions, we set the concentrations at the borders of each individual phase as constant. These concentrations are the limits of the solubility, that can be extracted from a phase diagram. Concentration ratios outside of these limits don't exist in equilibrium (miscibility gaps).

For the α -phase, these concentrations are $c^{\alpha 0}$ and $c^{\alpha\beta}$, for the β -phase $c^{\beta\alpha}$ and $c^{\beta\gamma}$ and for the γ -phase $c^{\gamma\beta}$ and $c^{\gamma 0}$.

The difference between $J^{\alpha\beta}$ and $J^{\beta\alpha}$, the fluxes in the α - and β -phases, right at their phase interface, is the net amount of atoms of constituent A that accrue at the α - β -interface at any given time. These particles increase the concentration in the α -phase above the solubility range of this phase. Therefore, they take part in a phase transition to the β -phase. Thus, in a certain volume in the α -phase, near the interface, the concentration of constituent A increases from $c^{\alpha\beta}$ to $c^{\beta\alpha}$ and this volume is changed to be part of the β -phase now. Therefore, the interface $z^{\alpha\beta}$ between those phases moves by $dz^{\alpha\beta}$, meaning the β -phase has gained $dz^{\alpha\beta}c^{\beta\alpha}\rho$ particles per unit area, and the α -phase has lost $dz^{\alpha\beta}c^{\alpha\beta}\rho$ particles per unit area. As always, we neglect any changes in the particle density ρ due to composition changes. With conservation of the number of particles, we get the equation $(J^{\beta\alpha} - J^{\alpha\beta}) \cdot dt = dz^{\alpha\beta}(c^{\beta\alpha} - c^{\alpha\beta})\rho$ from which we can derive an equation describing the movement of the phase interface

$$\frac{dz^{\alpha\beta}}{dt} = \frac{1}{\rho} \cdot \frac{J^{\beta\alpha} - J^{\alpha\beta}}{c^{\beta\alpha} - c^{\alpha\beta}}. \quad (2.3.8)$$

Analogous considerations lead to an equivalent equation for the β - γ -interface

$$\frac{dz^{\beta\gamma}}{dt} = \frac{1}{\rho} \cdot \frac{J^{\gamma\beta} - J^{\beta\gamma}}{c^{\gamma\beta} - c^{\beta\gamma}}. \quad (2.3.9)$$

The Flux generally is given as

$$J = -D \cdot \rho \cdot \frac{\partial c}{\partial x}, \quad (2.3.10)$$

but it should be noted that the concentration profile at the phase interfaces is not generally differentiable. The fluxes in equations 2.3.8 and 2.3.9 therefore contain left- and right-sided derivatives.

They are discretized as

$$J^{\alpha\beta} = -D_\alpha \cdot \rho \cdot \frac{c_{\alpha,1} - c_{\alpha,0}}{\Delta x}, \quad (2.3.11)$$

$$J^{\beta\alpha} = -D_\beta \cdot \rho \cdot \frac{c_{\beta,N_\beta} - c_{\beta,N_\beta-1}}{\Delta x}, \quad (2.3.12)$$

$$J^{\beta\gamma} = -D_\beta \cdot \rho \cdot \frac{c_{\beta,1} - c_{\beta,0}}{\Delta x}, \quad (2.3.13)$$

and

$$J^{\gamma\beta} = -D_\gamma \cdot \rho \cdot \frac{c_{\gamma,N_\gamma} - c_{\gamma,N_\gamma-1}}{\Delta x}. \quad (2.3.14)$$

Here D_α , D_β and D_γ are the diffusion coefficients in the respective phases and c_α , c_β and c_γ are the concentration vectors in the individual phases, each running between the indexes 0 and N_α , N_β or N_γ respectively.

As in the previous sections, the simulations calculate the progress during each time step Δt , over which we iterate until the final time step $T = N_T \cdot \Delta t$. Typically, during one time step, the interfaces $z^{\alpha\beta}$ and $z^{\beta\gamma}$ move significantly less than the spatial discretization can resolve. Therefore, the actual positions of the interfaces do not immediately change each time step. Instead, the movements $\Delta z^{\alpha\beta}$ and $\Delta z^{\beta\gamma}$ are stored internally and are accumulated over the time steps. When an interface is supposed to have moved a distance $> \Delta x$, i.e. more than a spatial grid point, its position is updated.

One point deserves further commentary: In our interdiffusion experiments, there are initially only two phases: the two pure materials. It is therefore necessary to specify under what circumstances a third (or further) phase is formed.

In the code implemented in this thesis, the problem was solved as follows:

Initially, $z^{\alpha\beta}$, the border between the α - and β -phase, and $z^{\beta\gamma}$, the border between the β - and γ -phase are set to be identical. The indices indicating the position of these interfaces, $i^{\alpha\beta}$ and $i^{\beta\gamma}$ for the right side of the interface, and $i^{\beta\alpha}$ and $i^{\gamma\beta}$, for the left side are consequently identical as well. If there is no intermediate phase yet, $J^{\beta\alpha}$ and $J^{\beta\gamma}$, the fluxes 'inside' this nonexistent phase are set to be zero in the equations 2.3.8 and 2.3.9. Without a flux pushing them apart, the interfaces at $z^{\alpha\beta}$ and $z^{\beta\gamma}$ are pushed towards (and past) each other, creating an overlapping section belonging both to the α - and the γ -phase. When this happens, the indices are swapped, resulting in a section, which belongs neither to the α - nor the γ -phase. This is now our newly formed β -phase, which may grow according to equations 2.3.8 and 2.3.9 (see figure 2.7). It is necessary to differ between a symmetric case and an asymmetric case. The symmetric case occurs when the intermediate phase grows at the same rate from both sides, i.e. from the α -phase and from the γ -phase. This case is depicted in the upper part of figure 2.7. After the first movement of indices, the width of the overlapping region is $2 \cdot \Delta x$. An asymmetric example is shown in the bottom part of figure 2.7 for the case where the growth of the β -phase occurs at a higher rate in the original domain of the α -phase.

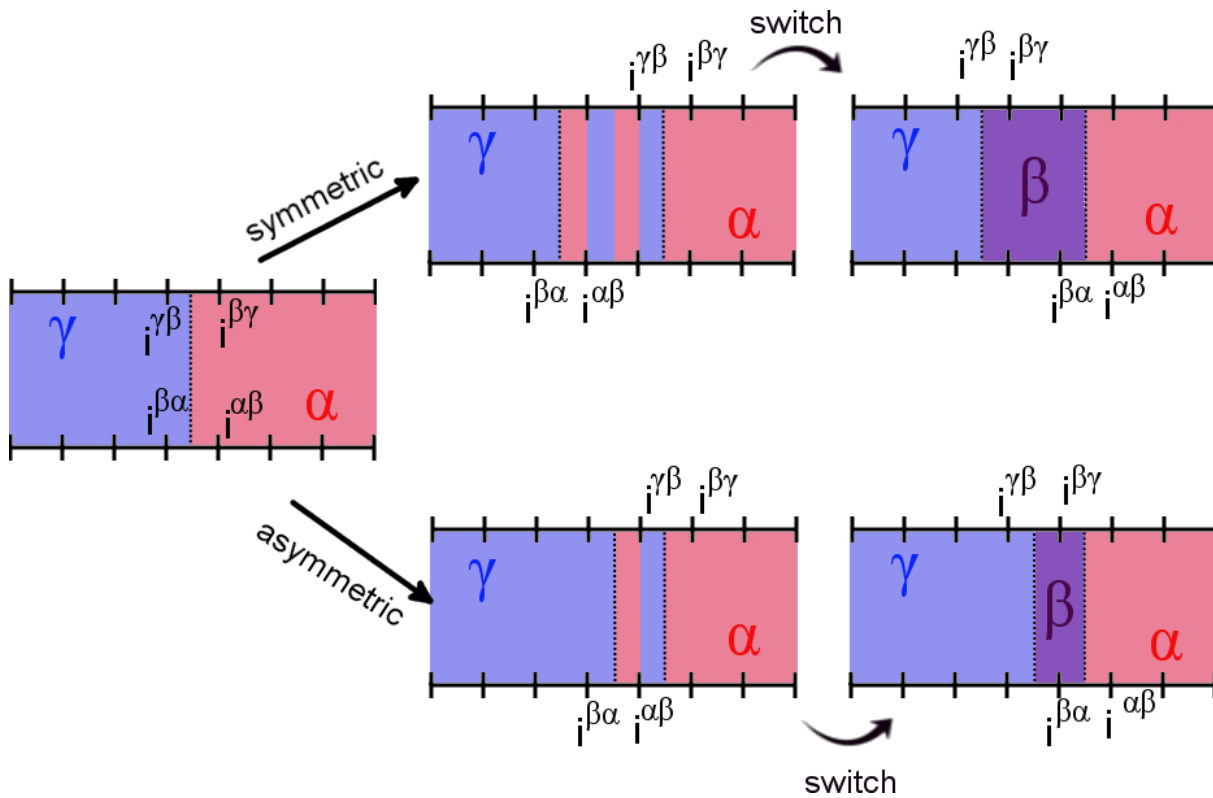


Figure 2.7.: Schematic description of how a new phase is formed. Without a intermediate phase, the interfaces are pushed past each other, resulting the phases to overlap. In that case, the interfaces are swapped, resulting in an empty space between them. This empty space becomes the new intermediate phase. Top: Symmetric case. Bottom: Example of asymmetric case.

If the solubility range in each phase is sufficiently narrow, the diffusion coefficients within the individual phases can be assumed to be constant. In that case there is an analytical solution for the reactive diffusion system, if we start from an initial concentration profile in the form of a step function, like the dashed line in the upper part of figure 2.8. We can use this solution to benchmark our simulation of reactive diffusion.^[5]

In this analytical solution, the concentration-profiles in the three different phases are given by

$$c^\alpha = c^{\alpha 0} + \frac{c^{\alpha\beta} - c^{\alpha 0}}{1 - \operatorname{erf}(K^{\alpha\beta})} \left(1 - \operatorname{erf} \left(\frac{x}{\sqrt{4D^\alpha t}} \right) \right),$$

$$c^\beta = \frac{c^{\beta\gamma}}{\operatorname{erf}(K^{\beta\alpha}) - \operatorname{erf}(K^{\beta\gamma})} \left(\operatorname{erf}(K^{\beta\alpha}) - \operatorname{erf} \left(\frac{x}{\sqrt{4D^\beta t}} \right) \right) - \frac{c^{\beta\alpha}}{\operatorname{erf}(K^{\beta\alpha}) - \operatorname{erf}(K^{\beta\gamma})} \left(\operatorname{erf} \left(K^{\beta\gamma} \right) - \operatorname{erf} \left(\frac{x}{\sqrt{4D^\beta t}} \right) \right) \quad (2.3.15)$$

and

$$c^\gamma = c^{\gamma 0} + \frac{c^{\gamma\beta} - c^{\gamma 0}}{1 - \operatorname{erf}(K^{\gamma\beta})} \left(1 + \operatorname{erf} \left(\frac{x}{\sqrt{4D^\gamma t}} \right) \right).$$

The positions of the phase interfaces are given by

$$z^{\alpha\beta} = K^{\alpha\beta} \sqrt{4D^\alpha t} = K^{\beta\alpha} \sqrt{4D^\beta t}, \quad (2.3.16)$$

and

$$z^{\beta\gamma} = K^{\beta\gamma} \sqrt{4D^\beta t} = K^{\gamma\beta} \sqrt{4D^\gamma t}. \quad (2.3.17)$$

$K^{\alpha\beta}$ and $K^{\beta\gamma}$ are proportionality-factors that are indirectly determined through the equations

$$c^{\beta\alpha} - c^{\alpha\beta} = \frac{c^{\alpha 0} - c^{\alpha\beta}}{K^{\alpha\beta}\sqrt{\pi}(1 - \operatorname{erf}(K^{\alpha\beta}))} \exp(-(K^{\alpha\beta})^2) + \frac{c^{\beta\gamma} - c^{\beta\alpha}}{K^{\beta\alpha}\sqrt{\pi}(\operatorname{erf}(K^{\beta\alpha}) - \operatorname{erf}(K^{\beta\gamma}))} \exp(-(K^{\beta\alpha})^2), \quad (2.3.18)$$

$$c^{\gamma\beta} - c^{\beta\gamma} = \frac{c^{\beta\alpha} - c^{\beta\gamma}}{K^{\beta\gamma}\sqrt{\pi}(\operatorname{erf}(K^{\beta\alpha}) - \operatorname{erf}(K^{\gamma\beta}))} \exp(-(K^{\beta\gamma})^2) + \frac{c^{\gamma 0} - c^{\gamma\beta}}{K^{\gamma\beta}\sqrt{\pi}(1 + \operatorname{erf}(K^{\gamma\beta}))} \exp(-(K^{\gamma\beta})^2), \quad (2.3.19)$$

$$K^{\alpha\beta} = K^{\beta\alpha}\sqrt{D^\beta/D^\alpha}, \quad (2.3.20)$$

and

$$K^{\gamma\beta} = K^{\beta\gamma}\sqrt{D^\beta/D^\gamma}. \quad (2.3.21)$$

As can be seen in figure 2.8, the difference between the analytical and numerical solution is of order of magnitude between 10^{-5} and 10^{-3} . Due to the discretization, however, the positions of the interfaces sometimes differ by one Δx , resulting in a high deviation at that particular point. As long as the spatial discretization is sufficiently fine, this does not pose a problem.

By subtracting equation 2.3.16 from equation 2.3.17, we can describe the growth of the width w of the β -phase with respect to time

$$w = z^{\alpha\beta} - z^{\beta\gamma} = \sqrt{4D^\beta(K^{\beta\alpha} - K^{\beta\gamma})^2 t} = \sqrt{Kt}. \quad (2.3.22)$$

Here, the constant K can be viewed as an 'effective' diffusion coefficient. According to equation 2.3.22, the thickness w of an intermediate phase follows a square root behaviour with respect to time. This can be reproduced with our simulations, as can be seen in figure 2.9.

By determining the factor K with a fit to a measured time series of the thickness w , the diffusion coefficient D_β in the intermediate phase can be determined with equation 2.3.22.

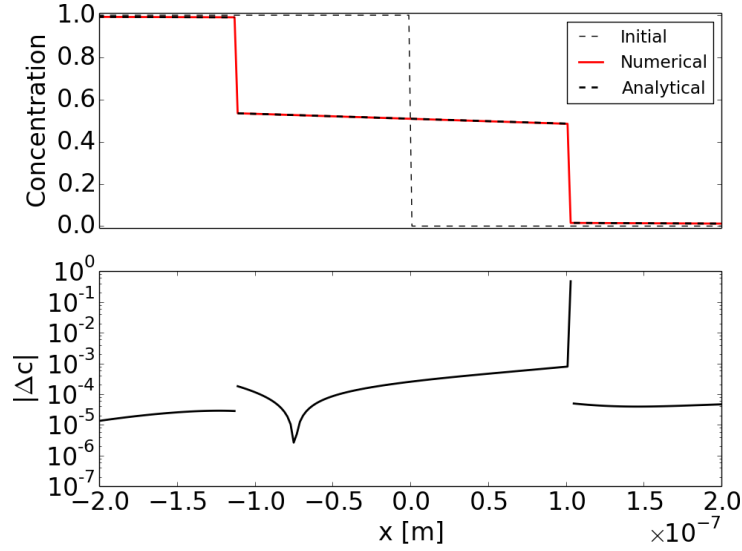


Figure 2.8.: Top Picture: Comparison between the analytically and the numerically calculated concentration profile. The solubility ranges were set to be $c_{\alpha\beta} = 1.5\%$, $c_{\beta\alpha} = 48.5\%$, $c_{\beta\gamma} = 53.5\%$ and $c_{\gamma\beta} = 99\%$. The diffusion coefficients in the alpha and gamma phases were set to $D_\alpha = D_\gamma = 5 \cdot 10^{-19} \frac{m^2}{s}$, and in the beta phase to $D_\beta = 7 \cdot 10^{-19} \frac{m^2}{s}$. The diffusion time was 24 hours, the calculation space was $\pm 5 \mu m$. The number of spatial grid points was 5000, and the number of temporal grid points 7201. Bottom Picture: Absolute difference between the analytically and numerically calculated concentration profile.

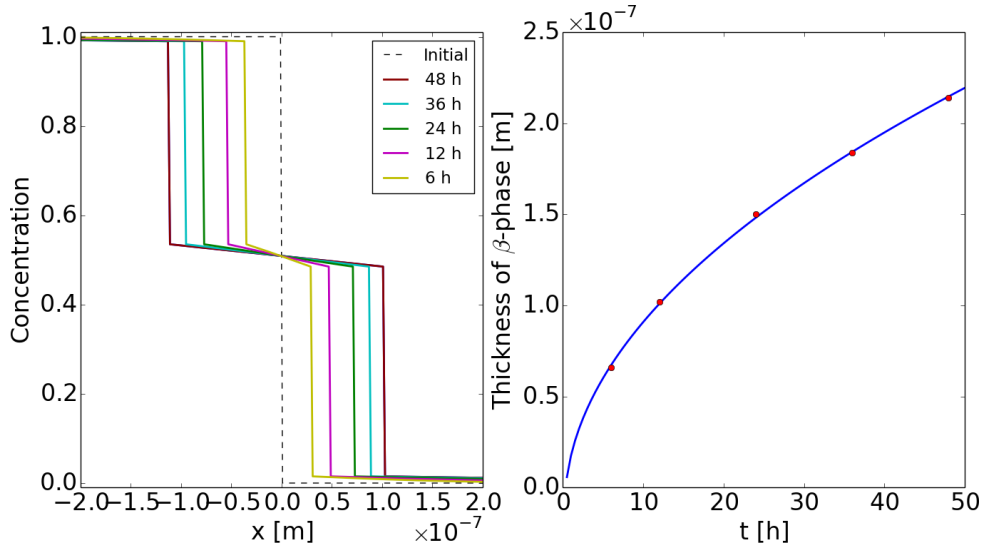


Figure 2.9.: Left graph: Simulated concentration profiles at different time points. Right graph: Width of the intermediate phase vs time (red dots) and square root fit to the data (blue line).

Now, the code has been shown to reproduce the analytical solution for the simple case of constant diffusion coefficients in the different phases and only one intermediate phase. It is, however, possible to expand upon these simulations, to include more phases or a concentration dependent diffusion coefficient, by modifying the way new phases are created or adapting the code for interdiffusion in each phase the way it was shown in subsection 2.3.2. While adapting the code to a concentration dependent diffusion coefficient is very simple, adding more phases is a bit trickier, but can still be achieved. A time series of a simulation from a simple first implementation of such a multiple-phases-code is shown in figure 2.10. It can nicely be observed in this example how the phase with the broader solubility range grows more quickly. This dependence is in concordance with the study of Kajihara et al..^[5]

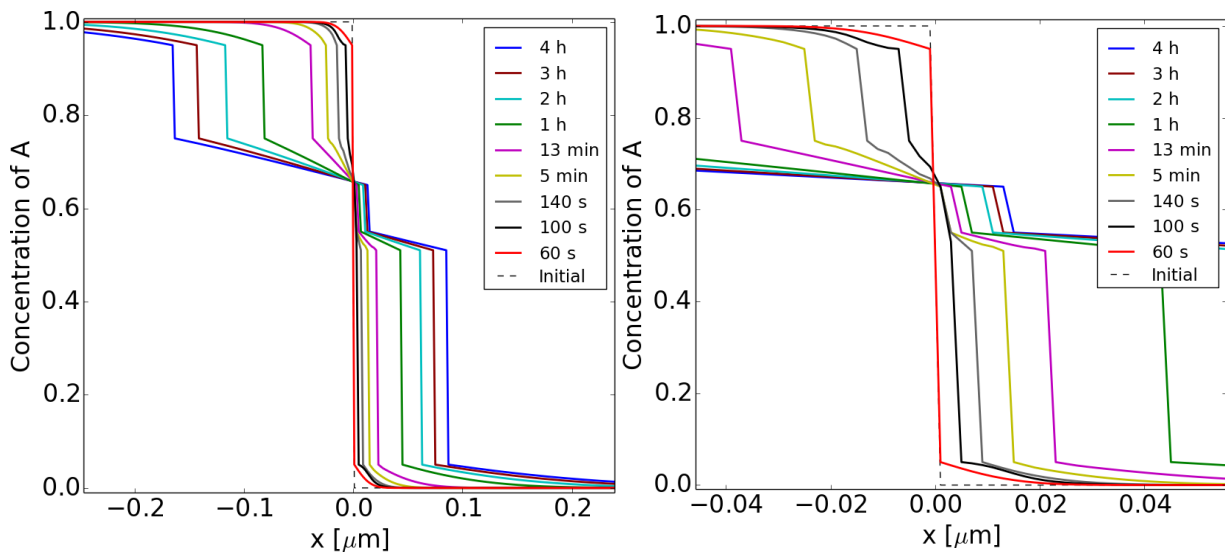


Figure 2.10.: Simulation with reactive diffusion code modified to create two intermediate phases. Left: Overview; Right: Zoom in to illustrate the creation of the intermediate phases. The solubility ranges were chosen to be 5% and 95% for the pure materials; The intermediate phases lie between 51 – 55% and 65 – 75%. The diffusion coefficient in the pure materials were both set to be $1 \times 10^{-18} \frac{m^2}{s}$; the diffusion coefficients in the intermediate phases were set to be $3 \times 10^{-18} \frac{m^2}{s}$.

3. Analysis techniques

After establishing how diffusion coefficients can be determined, we will now briefly explain the experimental techniques, that were used in the course of this thesis.

3.1. Scanning electron microscopy

With scanning electron microscopy (SEM), a focussed electron beam scans the surface of the material to be investigated. These electrons interact with the material and can be deflected, create new, secondary electrons and induce X-ray emission. By detecting the emitted electrons and radiation, information about the material can be obtained.^[13]

Backscattered electrons (BSE) primarily give information about the composition of the material; the higher the atomic number of the targeted material, the higher the yield of backscattered electrons and thus the brighter the detected signal. An example of a BSE-image can be seen in figure 5.24.

Secondary electrons (SE) are electrons freed by the energy deposited to the material by the primary electron beam. They scatter isotropically and can therefore give information about the topography of the material; from an elevated part of the surface, more secondary electrons can leave the material than from a flat surface. Since there is always a fraction of secondary electrons that are excited by backscattered electrons, and the SE-emission is dependent on the material's work function, SE images also contain information about the material's composition. An example of a SE-image can be seen in figure 5.18.

The characteristic peaks of emitted X-ray radiation can give information about the elemental composition of the targeted material. This method is called energy-dispersive X-ray emission spectroscopy (EDX). An example of an EDX-image can be seen in figure 5.16.

When targeting the material with a focussed ion beam (FIB), atoms can be ejected (sputtered) from the material. In our case we use Ga^+ . With this method we can cut into the material to obtain cross sections of the iron-tungsten interface. To avoid rounding

off the edge of the cut from a not perfectly focussed ion beam, a layer of platinum is deposited onto the material, before the cut is conducted. An image of such a FIB cut, along with the deposited platinum layer can be seen in figure 5.13.

3.2. Rutherford backscattering spectrometry

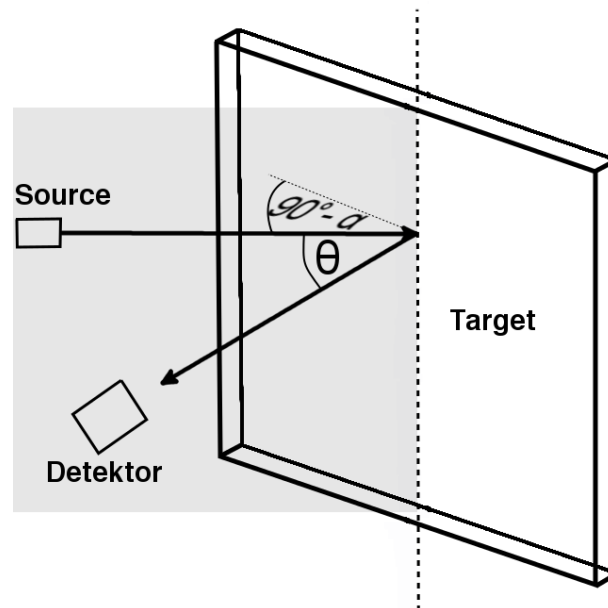


Figure 3.1.: Schematic of the setup of a IBA experiment using the Cornell geometry. Ion beams are created in a source and scattered off a target sample, that is rotated by an angle α . The ions that were scattered under an angle θ , in our case $\theta = 165^\circ$, are detected with a solid state detector.

RBS is based on the analysis of the energy of projectiles, that were backscattered from a target material (see figure 3.1). At the energies of several MeV, that were used in the course of this thesis, the scattering process of an incoming ion off atoms in the target can be viewed as a simple two body collision (see figure 3.2), since the binding energy stored in the lattice of the target material can be neglected and the de Broglie wavelength of the incoming particles is much smaller than the lattice constant, meaning that diffraction can be neglected as well.^[13]

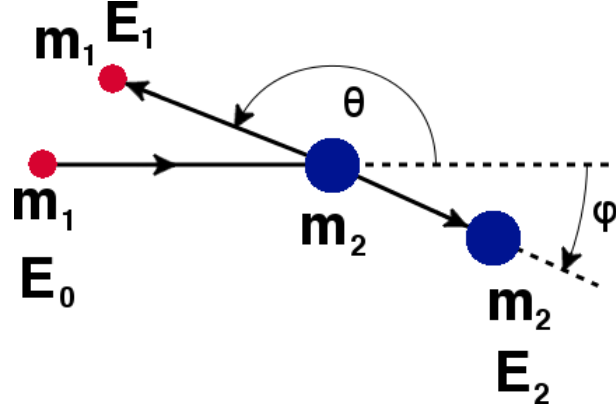


Figure 3.2.: Schematic of a two body collision. The incoming particle of mass m_1 and energy E_0 hits the resting target of mass m_2 . After the collision, the incoming particle is deflected by an angle of θ with the energy E_1 , while the target particle is recoiled at an angle φ .

Therefore, for the scattering process, we only need to consider conservation of energy

$$\frac{m_1 v_0^2}{2} = E_0 = E_1 + E_2 = \frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2}, \quad (3.2.1)$$

and momentum

$$m_1 v_0 = m_1 v_1 \cos \theta + m_2 v_2 \cos \varphi, \quad (3.2.2)$$

$$m_1 v_1 \sin \theta = m_2 v_2 \sin \varphi. \quad (3.2.3)$$

From 3.2.1, 3.2.2 and 3.2.3 the ratio between the energies of the incoming and the backscattered particles can be calculated to be

$$\frac{E_1}{E_0} = \left(\frac{m_1 \cos \theta \pm \sqrt{m_2^2 - m_1^2 \sin^2 \theta}}{m_1 + m_2} \right)^2. \quad (3.2.4)$$

From equation 3.2.4, one can see, that the mass of the target atoms can be determined by measuring the energy of backscattered ions under a given angle θ .

The areal density of atoms in the target material can be deduced from the fraction of particles scattered under a certain angle θ via

$$\Delta Q = \frac{d\sigma}{d\Omega} \Delta\Omega \cdot N_a \cdot Q. \quad (3.2.5)$$

Here, Q is the number of incoming particles, ΔQ the number of backscattered particles, $\Delta\Omega$ the total solid angle covered by the detector and N_a the areal number density of particles in said target sample. $\frac{d\sigma}{d\Omega}$ is the differential cross section, in our case Rutherford's cross section

$$\frac{d\sigma}{d\Omega} = \left(\frac{Z_1 Z_2 e^2}{2E} \right)^2 \frac{\left(\cos \theta + \sqrt{1 - \left(\frac{m_1}{m_2} \right)^2 \sin^2 \theta} \right)^2}{\sin^4 \theta \sqrt{1 - \left(\frac{m_1}{m_2} \right)^2 \sin^2 \theta}}, \quad (3.2.6)$$

since nuclei are scattering off each other due to their Coulomb potential.

These equations only apply for very thin samples or nuclei right at the surface of the target material. At thicker targets, particles can penetrate deeper in the material before they are backscattered. In that case, they lose energy by interacting with the free electrons in the material and because of small angle elastic scattering on the nuclei. For higher energies, the energy loss can be described by the Bethe-Bloch-Formula

$$-\frac{dE}{dx} = \frac{4\pi}{m_e c^2} \frac{n Z^2}{\beta^2} \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \left(\ln \frac{2m_e c^2 \beta^2}{I(1-\beta^2)} - \beta^2 \right), \quad (3.2.7)$$

where $\beta = \frac{v}{c}$, c is the speed of light, e is the electron charge and I the mean excitation potential of the atoms, typically $16 \text{ eV} \cdot Z^{0.9}$.^[15] With this, it is possible to gain information about the depth profile of the target material.

In figure 3.3, RBS spectra of typical samples used in this thesis can be seen. Initially the samples always consist of a thin tungsten layer on top of an iron substrate. Upon annealing, the materials diffuse into each other and the concentration depth profiles smear out at the interface. The right plateau in the two graphs of figure 3.3 is the signal coming from the tungsten. The rightmost edge corresponds to ions scattered directly from the surface. The bulk of the plateau stems from ions that penetrated into the W layer before the scattering event. Ions backscattered from W at the interface contribute to the signal at the left edge of the W plateau. The same applies to the Fe edge and plateau. When interdiffusion has occurred, some tungsten can be found deeper into the iron layer and some iron has moved closer to the surface. This leads to a small tungsten signal appearing at lower energies in the measured spectrum, and a small iron signal appearing at higher energies. As a result, the spectrum appears more 'smeared out'.

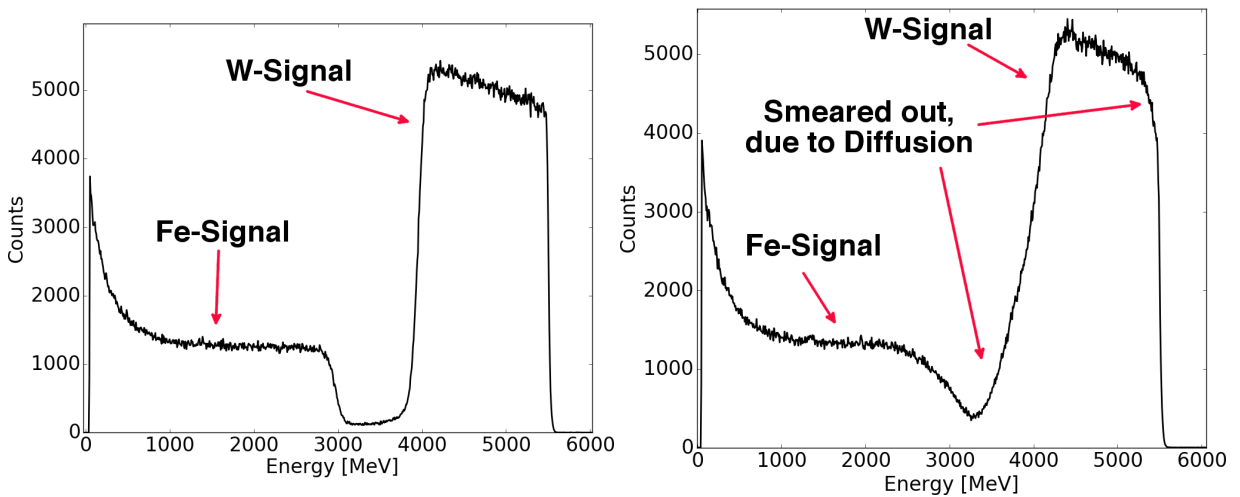


Figure 3.3.: Typical spectra that were measured with Rutherford backscattering. The left figure shows the spectrum of a reference sample that was not annealed, the right figure shows the spectrum of a sample annealed for 36 hours at 1050 K. The spectra were measured using 6 MeV $^4\text{He}^{2+}$ ions under a 45° angle.

3.3. Sputter XPS depth profiling

X-ray photoelectron spectroscopy (XPS) is based on the reverse principle of EDX. A monochromatic X-ray beam, in our case 1486 eV Al-K α radiation, targets a material and excites electrons via the photoelectric effect. Subsequently emitted secondary electrons are then analyzed in a hemispherical electron energy analyzer. From their spectrum, it is possible to determine the material composition of the target: From the positions of resonance peaks, the atomic species, the target is made out of, can be distinguished. From the relative height of the peaks, the concentration of the different atomic species can be inferred.^[13] Since the mean free path of the excited electrons is only a few nanometers, no depth profile can be obtained. To solve this problem, sputter XPS is done: After taking a XPS-spectrum, the surface layer of the material is sputtered away with ions, in our case Ar ions with an energy of 10 keV. After that, another XPS-spectrum of the surface is acquired, followed again by sputtering away another layer of the material. These two processes can be alternated, until a satisfactory depth is reached. It should be noted, that not all atoms are sputtered away by the ion beam at the same rate. Preferential sputtering and forward implantation modify the initial concentration depth profile. This needs to be taken into account upon interpretation of the resulting XPS signal intensities.

4. Sample preparation

In this chapter, the sample preparation steps before their analysis, are summarized.

4.1. Polishing and recrystallization

We start with iron substrates of high purity, to avoid influencing the interdiffusion with contaminants in the substrate. Two kinds of materials have been used: Some of the samples have dimensions of $10 \times 15 \times 1$ mm and are 99.5 wt% pure (Goodfellow FE000410). According to the manufacturer, typical contaminants include:

Mn	Si	C	P	S
3000 ppmw	1000 ppmw	< 800 ppmw	< 400 ppmw	< 500 ppmw

Other samples have dimensions of $8 \times 8 \times 1$ mm and are 99.99 + wt% pure (Goodfellow FE000411). In this case, measured contaminants include

Al	B	Cr	Co	Cu	Mn
1.5 ppmw	1.1 ppmw	2.4 ppmw	6.7 ppmw	0.9 ppmw	0.64 ppmw
Mo	Ni	Ti	W	Zn	Sn
0.22 ppmw	1.5 ppmw	0.79 ppmw	0.27 ppmw	0.35 ppmw	0.3 ppmw

These two different substrate grades will from now on be referred to as FE99.5 (Goodfellow FE000410) and FE99.99 (Goodfellow FE000411).

After cleaning the substrates in an ultrasonic bath for 30 minutes, they are mechanically ground with 1200-grit abrasive paper for 40 s, 2500-grit abrasive paper for 20 s and 4000-grit abrasive paper for 40 s. After that, they are polished for 3 min with a Kulzer Touch LAM 2TS8 cloth covered with 'Bio Diamant Liquid Red'. Finally, they are polished with a nap cloth covered with the Silicon-Oxide suspension 'OP-U' for 30 s and then again 30 s with water, to wash away the OP-U suspension. This way, R_a roughness values of approximately 10 nm are reached.

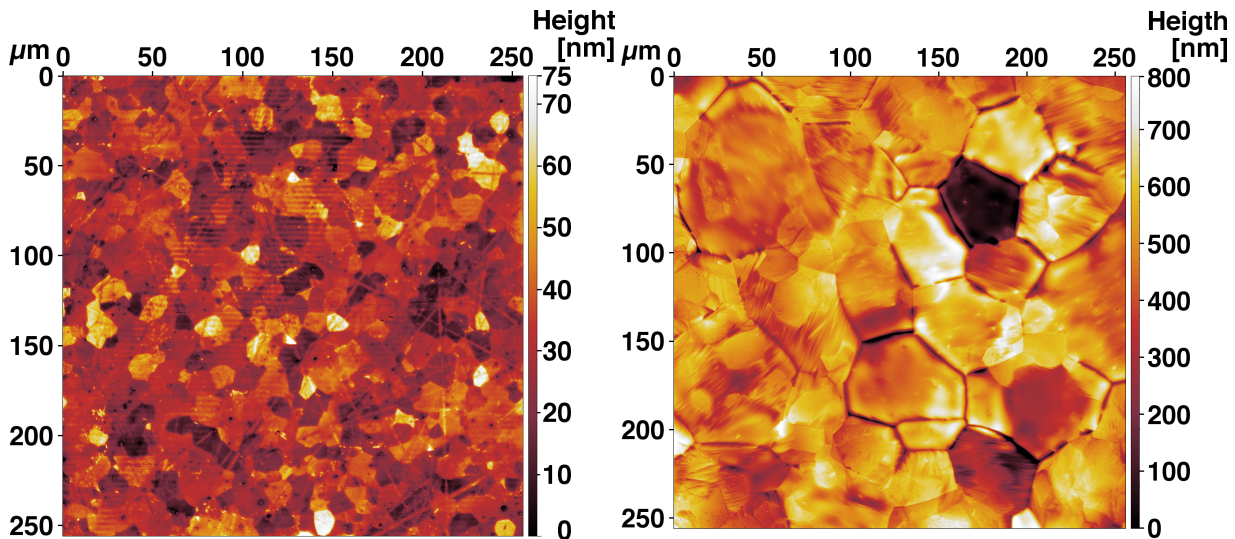


Figure 4.1.: Confocal laser scanning microscope images of a typical iron substrate (FE99.5) after being polished (left), and after being annealed for 96 h (2×48 h) at 1200 K (right). The average grain size grew from approximately $10 \mu\text{m}$ to $50 \mu\text{m}$.

After cleaning them again in an ultrasonic bath, they were heated in a molybdenum box at 1200 K for 12 h (FE99.5) and 96 h (FE99.99) in two sessions of 48 h. Between those two sessions, the substrates were rearranged in the oven to minimize the effects of a potential temperature gradient. The annealing was done to recrystallize the smear layer induced by the polishing and to make the iron grains grow bigger. Both an amorphous smear layer and grain boundaries with their high density of lattice defects can influence the diffusion process; for a well defined experiment, these effects need to be kept as small as possible. In figure 4.1, images can be seen, showing the roughness and grain size before and after annealing of the less pure samples, taken with a confocal laser scanning microscope. A confocal laser scanning microscope moves a focussed laser beam over the surface of the sample. Through an aperture, only light that is reflected from the plane that is in focus is detected, enabling the user to determine the exact distance to the surface, and from this the topography of the surface. From this, a 3D image of the surface can be reconstructed.^[2] The grains have grown from approximately $10 \mu\text{m}$ to $50 \mu\text{m}$ on some of the samples. On the more pure samples (FE99.99), much bigger grains, of a size of several hundred μm have formed, as can be seen from the light microscope images in figure 4.2.

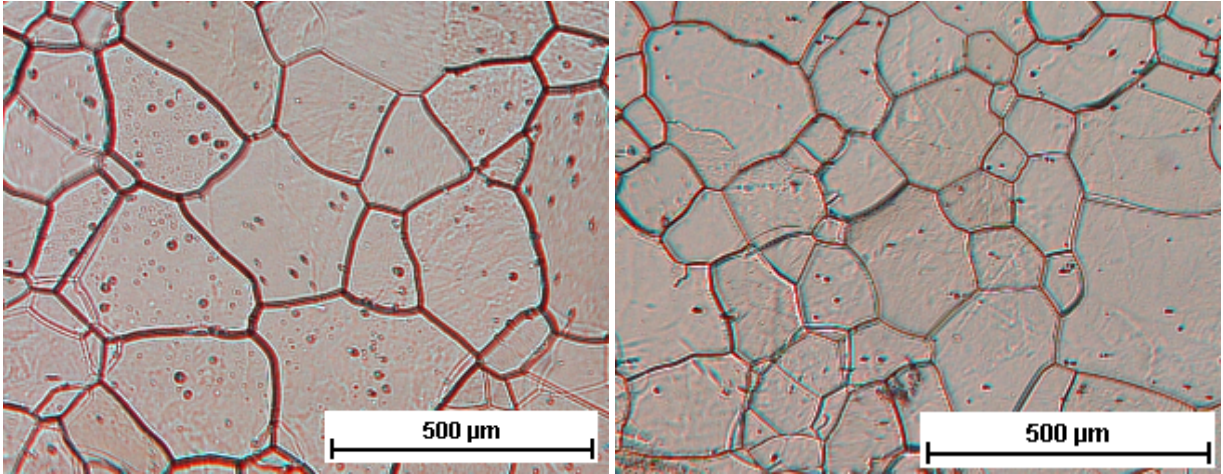


Figure 4.2.: Light microscope images of iron substrates (FE99.99) after being annealed for 96 h (2×48 h) at 1200 K (right). The size of these grains grew to several hundred μm .

4.2. Deposition of tungsten

After the recrystallization, a $1\ \mu\text{m}$ thick tungsten layer was deposited on top of those substrates. This was done by using the magnetron sputter device 'DENTON Vacuum, LLC Discovery-18 Sputtering System' in DC mode.

This device works as follows: An argon plasma is created by a glow discharge induced by a 500 V voltage. Ar ions from that plasma then impact on a target made out of tungsten and sputter away tungsten atoms. This also creates secondary electrons that maintain the plasma. By using a magnetic field, created by permanent magnets, the plasma is made sure to have the highest density in the vicinity of the sputtering target. Ejected tungsten atoms are deposited onto the iron substrates (see figure 4.3).^[21]

Before each actual deposition run, the surfaces of the Fe substrates are sputter-cleaned by applying an 98 W RF potential to the substrate holder, of an voltage between 500 and 600 V. This step is necessary for removal of surface oxides. However, it needs to be limited in time due to potential development of surface roughness. To figure out the optimal etching time, a test series was performed.

Four samples were etched for 0 min, 1 min, 20 min and 40 min and then coated with a 16 nm layer of tungsten, to make sure the iron does not immediately oxidize again after being taken out of the vacuum. These are then looked at via Sputter-XPS. To determine the oxygen content, we looked at the O1s line, between 424 and 538 eV. For each sample and each measured spectrum, the background was removed, by subtracting a linear fit through

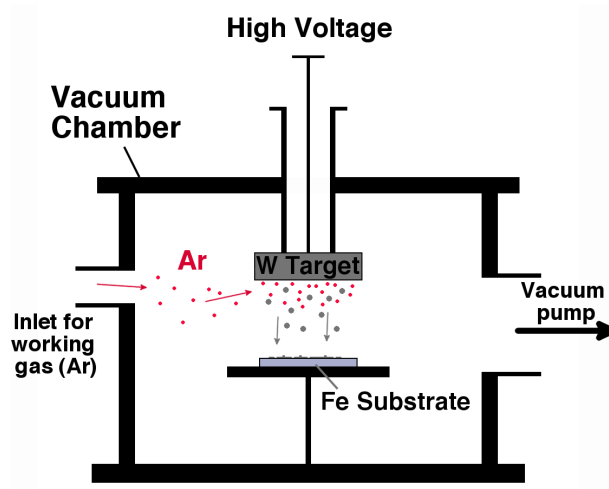


Figure 4.3.: Schematic of a sputter deposition process.

the areas away from the actual O1s peak. Then, each spectrum was integrated over and plotted against the fluence of argon ions that were used to to sputter away previous layers, up to the point of measuring the spectrum (figure 4.4). From this measurement series, we can be sure, that 20 min of pre-sputtering is sufficient to remove all the oxygen from the surface of the iron substrate. This time was therefore chosen for the preparation of the substrates.

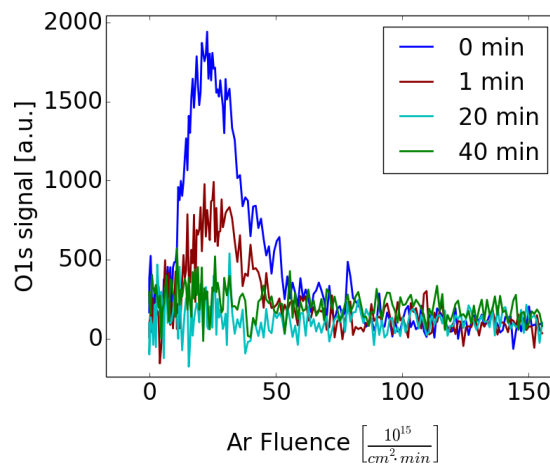


Figure 4.4.: Sum over the Spectra around the O1s line, between binding energies of 424 and 538 eV plotted against the argon fluence after which the spectrum was measured. It can be seen, that after 1 min etching there is still oxygen in the sample. After 20 min there is barely any signal above the background and the signal cannot be reduced by etching even longer.

4.3. Annealing

Finally, we anneal our newly created W/Fe-system, to induce diffusion. A typical annealing process is depicted in figure 4.5. At first, the temperature is held at 200°C to minimize sample surface contamination by potential outgassing. After that, the temperature is raised at a rate of $60 \frac{K}{min}$ and then held for various times. The cooling happens naturally after turning off the oven; after 4 minutes, the temperature is already below 700 K. Compared to the annealing times, the heating-up and cooling down periods can consequently be neglected.

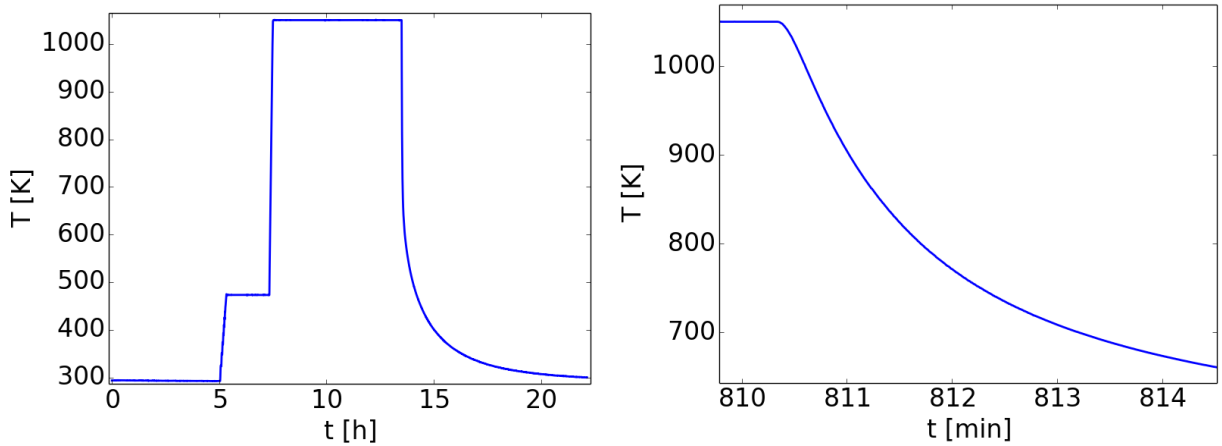


Figure 4.5.: Left Graph: Typical temperature curve of an annealing process done in the course of this thesis. After five hours of pumping to reach a good vacuum and 2 hours at 200°C to minimize sample surface contamination by potential outgassing, the temperature is held at 1050 K for six hours; Right Graph: Zoom in on the cooldown period.

In table 4.1, an overview of all the time/temperature combinations for the samples created is given, as well as an overview of the measurement techniques applied on them. Furthermore is it indicated, when the substrate consists out of the less pure FE99.5 instead of the purer FE99.99. On the samples annealed for 12 hours at 1000 K and for 36 hours at 900 K, no SEM was performed, since samples annealed at the same temperatures, but for longer times, already don't show phase formation.

T [K] \ t [h]	6	12	24	36	48	51	60
900	RBS SEM			RBS			RBS SEM
950	RBS SEM						
1000	RBS SEM	RBS	RBS SEM	RBS SEM			SEM
1050	RBS SEM	SEM	RBS SEM	RBS SEM	RBS SEM	RBS SEM	
1100	SEM	SEM	RBS SEM		RBS SEM		
1150	SEM						

Table 4.1.: Overview of all the times and temperatures, samples were heated to induce diffusion, and with what methods they were looked into. Cells that are (partially) highlighted gray indicate substrates made out of the less pure iron FE99.5, cells (partially) highlighted in blue indicate samples where the formation of an intermediate phase was observed. Cells highlighted in green indicate an incompletely formed intermediate phase.

5. Assessment of interdiffusion and phase growth

As shown in chapter 2, it is necessary to measure the concentration profile with respect to the depth into the sample, across the tungsten-iron interface where diffusion has occurred. To achieve this, two methods have been applied. The first one being ion beam analysis (described in section 3.2) where helium ions are accelerated onto the target and the energy spectrum of the backscattered ions is measured. From this, one can determine the composition of the targeted material. The second method is to cut into the material with a fast ion beam (FIB) and then investigate the cross-section with an electron microscope (described in section 3.1). Both these methods and the experimental results they yielded will be described in the following sections.

5.1. Effective $D(C)$ from Rutherford backscattering spectrometry

5.1.1. Measured RBS spectra

RBS measurements were conducted at the tandem accelerator at the IPP Garching. In figure 5.1, the influence on the measured spectra, of annealing the samples at various temperatures, can be seen on an exemplary series of samples, that were heated between 900 and 1050 K, each for 6 hours. For each of these samples, two spectra were measured at two different positions, to get a feeling for the homogeneity of the sample and for the reproducibility of the measurement. For 1000 K and 1050 K, diffusion has visibly happened, which can be seen by the higher signal intensity in the section between the tungsten and the iron peak and also, because the left edge of the tungsten peak is a bit more smeared out. The spectra of the samples that were heated at 900 K and 950 K cannot be distinguished from the spectrum of the reference sample, which means there has been less diffusion than we can resolve with RBS. These two samples are therefore left out of the subsequent evaluations.

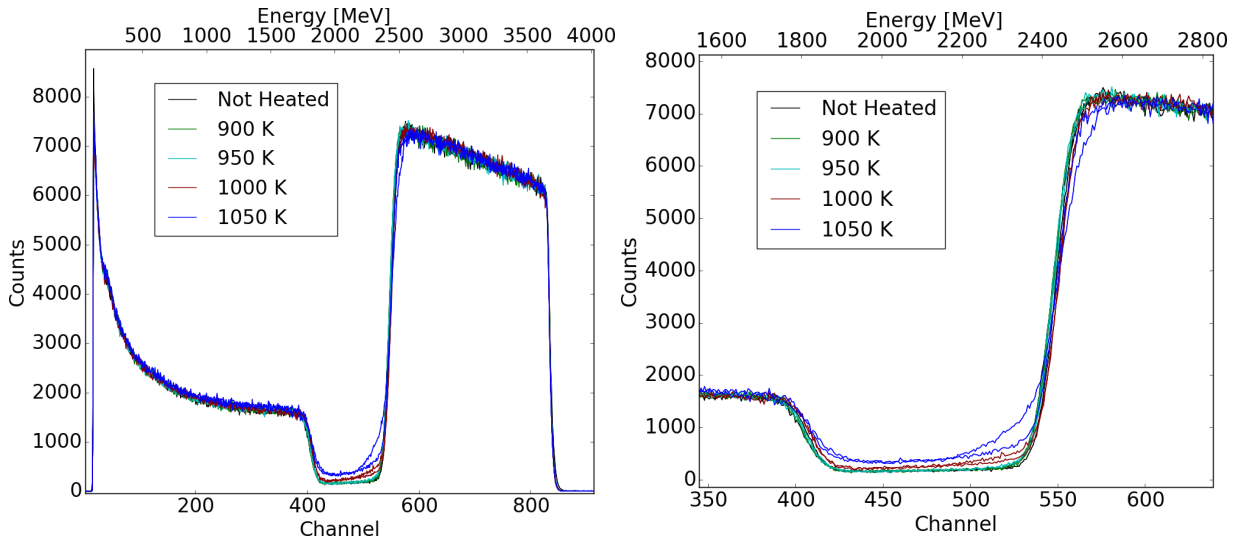


Figure 5.1.: Example for the temperature dependence of the measured spectra. All samples had FE99.99 as substrate and all but the unheated reference sample were heated for 6 hours. Measurements were taken with 4 MeV He^+ -Ions under a 0° angle.

Especially for higher temperatures, it becomes also apparent that there is a difference between the individual spots on a sample, which may be an indicator that the diffusion coefficient is dependent on the lattice orientation of the individual grains. As can be seen in figure 4.2, some grains have grown to a size of several hundred μm , meaning our ion beam, with an diameter of about one mm, does not average over many of them.

In figure 5.2, the influence of annealing the samples for different times can be seen on exemplary time series. The measured samples were each heated at 1050 K (left figure) and at 1100 K (right figure), for varying times, except for the reference sample, which was not heated at all. Generally, the trend goes into the direction of a more smeared out spectrum for longer diffusion times, with the sample heated for 36 hours at 1050 K being an outlier. For some samples, especially the ones that were heated for longer times and at higher temperatures, the right edge of the tungsten peak is smeared out too. This is due to iron atoms moving to the surface by means of grain boundary diffusion. This will be addressed in more detail in section 5.2.

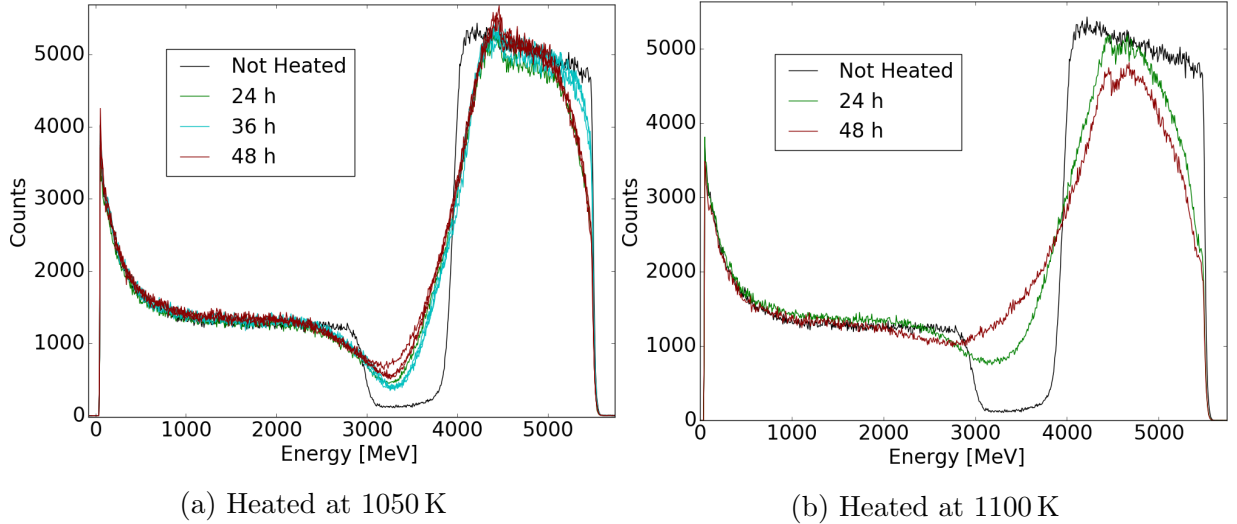


Figure 5.2.: Examples to illustrate the time dependence of the measured spectra. Measurements were taken with 6 MeV ${}^4\text{He}^{2+}$ -ions under a 45° angle. All iron substrates were the more pure FE99.99.

5.1.2. Obtaining concentration profiles

To get the concentration profiles from the measured spectra, we use the program SimNRA^[9], which simulates a RBS spectrum based on input values like the energy and ion species of the ion beam, the geometrical setup of the experiment and the composition of the target. Since we know the geometry of the experiment and the characteristics of the beam, we only need to determine the composition of the target. This is done with SimNRA by splitting up the target in several homogenous layers of various thicknesses and iteratively do simulations with varying layers, until the simulated result matches the measured spectrum.

On samples that were annealed for only short times (6 hours), an unresolvable mismatch between the measured data, and the expected spectrum, according to the program SimNRA, became apparent. Even the spectrum of the reference sample, that was only a homogenous layer of tungsten on top of a homogenous substrate of iron, did not match, as can be seen in figure 5.3. The reason for that might be because SimNRA underestimates multiple scattering.

Multiple scattering means that the projectile is not simply scattered from one of the target atoms, but instead hits several target atoms consecutively and is each time deflected by a small angle and each time loses some energy, before it finally leaves the target. When such a multiply scattered particle is detected, it has much lower energy, then it would

have, had it only undergone a single scattering event. This leads to an increased signal at lower energies. With heavier and denser elements, like tungsten, multiple scattering tends to happen more often, which can result in a relatively high background at low energies. For later measurements, a higher energy was chosen, to reduce multiple scattering, at the expense of some depth resolution.

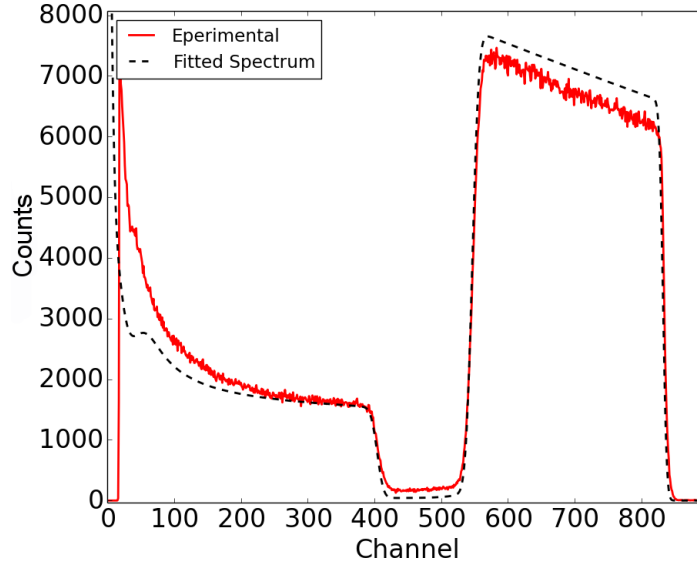


Figure 5.3.: The measured spectrum of the unheated reference sample consisting of a $1\mu\text{m}$ thick tungsten layer on top of an iron substrate, compared to the spectrum SimNRA would expect with the experimental setup that was used, which is a $4\text{ MeV } ^4\text{He}^+$ beam hitting at an 0° angle.

Since, in the case of these short diffusion times, only little diffusion has occurred and the spectrum barely changed. To be able to fit these spectra as precisely as possible, the difference between the measured reference spectrum, and the spectrum calculated with SimNRA was subtracted from all the spectra of this low temperature series, thus lowering the background between the tungsten and the iron peak.

For the measurements of samples that were annealed for longer times, this method was not applicable, because the spectra of strongly interdiffused samples look qualitatively different from the unheated samples and consequently also the multiple scattering background changes. The subtraction of the difference between the reference sample and the simulation could therefore not be justified. This may have an influence on the resulting diffusion coefficients, but when comparing the diffusion coefficients resulting from the subtracted and unsubtracted spectra, the results seem to match within the error bars.

In figure 5.4, examples for fits to the measured spectra can be seen. The left figure shows the fit to a spectrum, that was corrected to account for multiple scattering. The right figure shows the spectrum of a highly interdiffused sample; it can be seen that correcting for multiple scattering by simply subtracting the difference between the measured and the simulated spectrum of the reference sample cannot be justified on the highly interdiffused sample, since the spectrum is much too smeared out and looks qualitatively different from the reference spectrum.

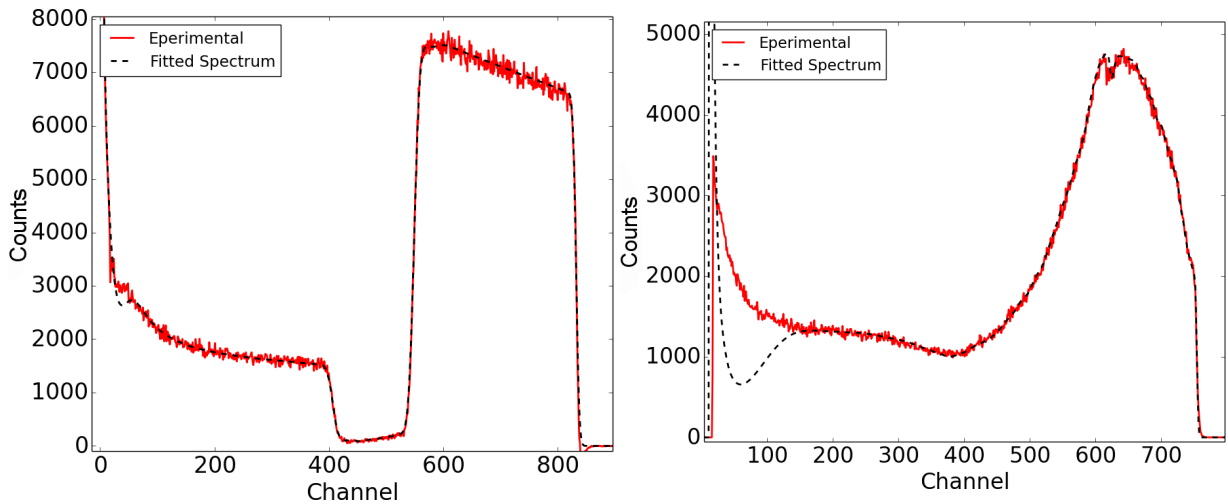


Figure 5.4.: Fits to Spectrum. Left: Fit to corrected spectrum with little interdiffusion. Sample heated at 1000 K for 6 hours; Right: Fit to uncorrected spectrum with strong interdiffusion. Sample heated at 1100 K for 48 hours.

In figure 5.5 the resulting concentration profiles from the SimNRA fits can be seen. For each sample, several measurements were done on different spots of the sample, to identify outliers due to measurement errors. The fitted concentration profiles were then averaged over and the standard deviation calculated.

For many of the samples heated at higher temperatures and for longer times, the tungsten concentration is lowered at the surface and never reaches 100%. This is because of the aforementioned grain boundary diffusion which leads to iron accumulating at the surface. As already stated, this will be addressed in section 5.2.3.

It should be noted, that some of these concentration profiles seem to be a bit inconsistent. The most apparent one being the measurement of samples heated at 1000 K for 24 hours, which there were two of. One of them seems to have diffused much more than the other sample, even though there were both annealed at the same time. It is not clear what happened there, but the concentration profile looks more similar to the ones from samples heated at higher temperatures. This might indicate a temperature gradient in the oven.

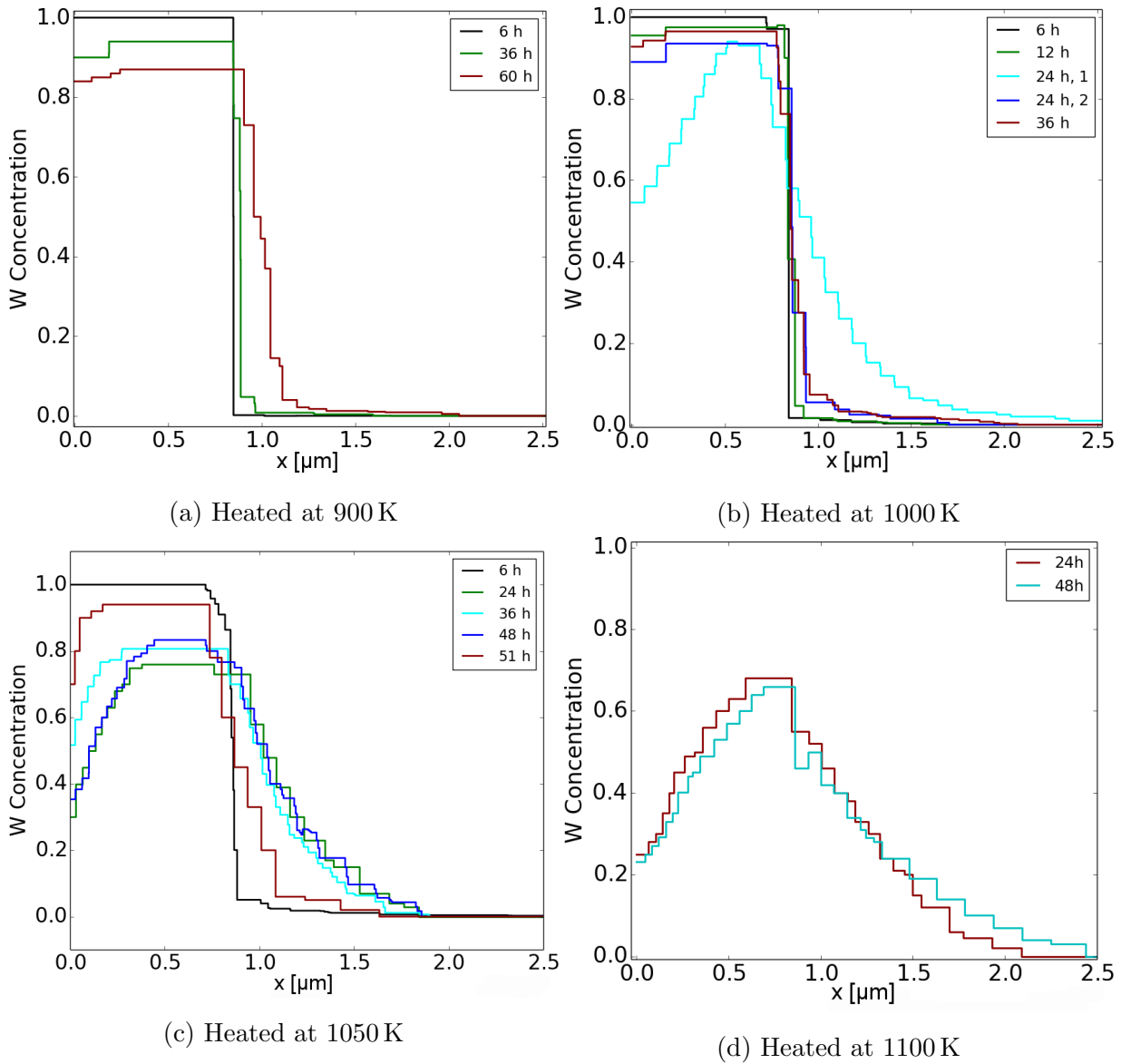


Figure 5.5.: Concentration profiles resulting from the fits to the measured RBS spectra. For samples, where several measurements were done at different positions, the average is depicted.

Similarly, the samples heated at 1050 K seem inconsistent, with longer times not implying stronger diffusion. Especially the sample heated for 51 hours seems to have diffused a lot less. This could be related to the fact that this sample was the only one, of the samples RBS measurements were done on, that had a substrate made out of FE99.5 instead of FE99.99.

5.1.3. The concentration dependence of the diffusion coefficient

With these concentration profiles, we can now determine the diffusion coefficients.

Procedure and results using forward simulations

At first, we determine the diffusion coefficient by fitting the concentration profiles from forward simulations with concentration dependent diffusion coefficients, to the measured concentration profiles. Examples of this can be seen in figure 5.6. Many of the concentration profiles do not reach 100% tungsten concentration anywhere and have even decreasing values towards the surface. This is due to the aforementioned grain boundary diffusion, which transports material to the surface, where it starts to diffuse into the W layer from the top. This does of course tamper with the assessment of the profile at the interface, where it is of interest to us. In these cases, the fit was done only to the concentration range, that seems unaffected by the surface diffusion. As a result, in these cases it was not possible to determine a diffusion coefficient for higher concentration values.

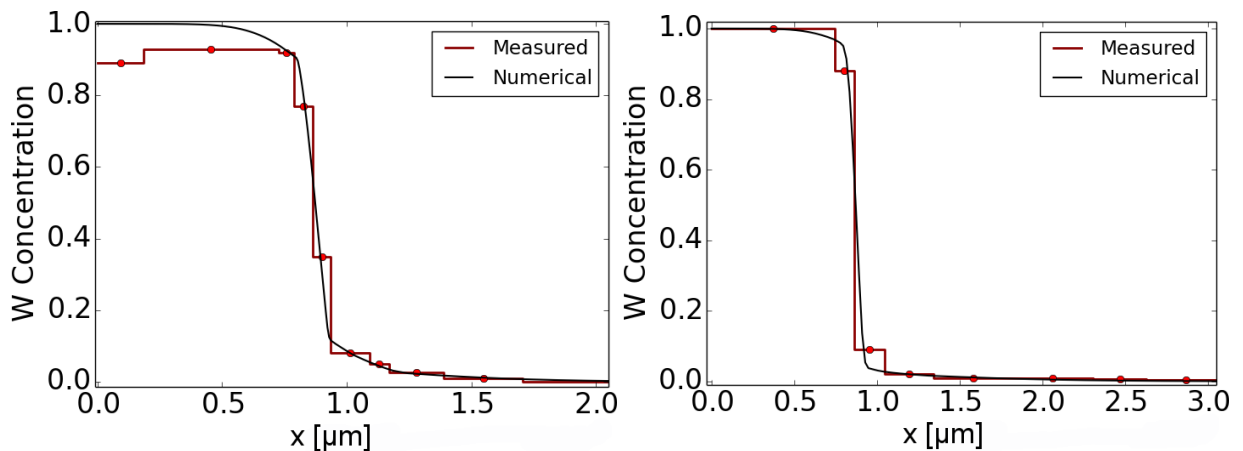


Figure 5.6.: Examples for fits using the simulation code with a concentration dependent $D(c)$. In the left picture, the concentration profile was from a sample heated for 24 hours at 1000 K, in the right picture from a sample heated for 6 hours at 1050 K.

In figure 5.7, the resulting diffusion coefficients are shown for the individual measurements. For all samples, the iron substrates were FE99.99, except for the one that was annealed for 51 hours at 1050 K, where FE99.5 was used. Each graph collects the data for a given temperature. The average for each temperature is also shown in addition. The results for high concentrations, where diffusion coefficients could not be determined with confidence,

were drawn in in dashed lines. The results for the sample heated at 900 K for 6 hours are shown in figure 5.7a, but were not considered in the calculation of the average, since, as stated earlier, the measured spectrum could not be distinguished from the reference sample, and therefore no reasonable fit to the obtained concentration profile could be done. This rationale seems justified by the fact that the resulting diffusion coefficients from measurements on this sample clearly deviate from the results for the other samples heated at this temperature.

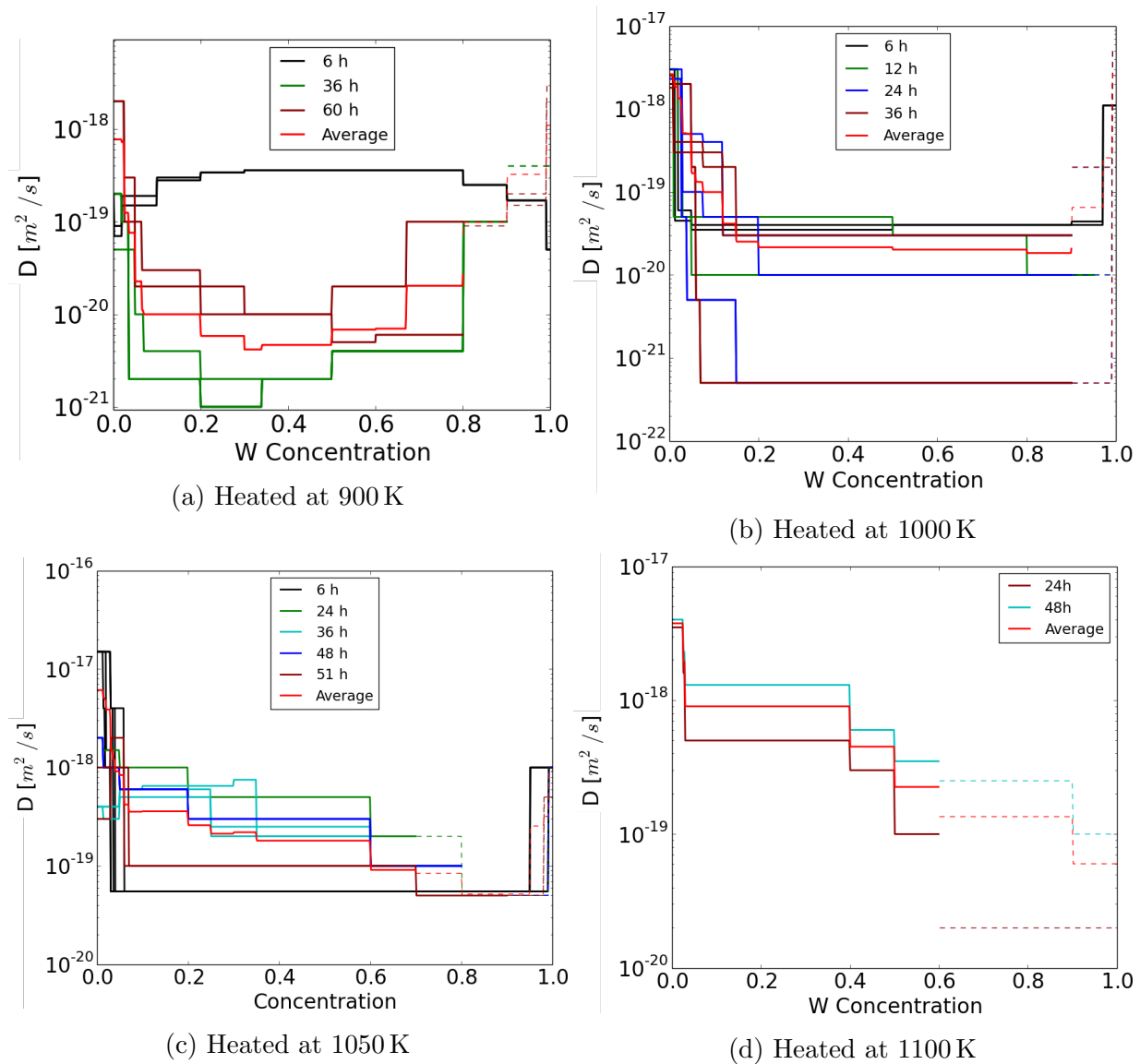


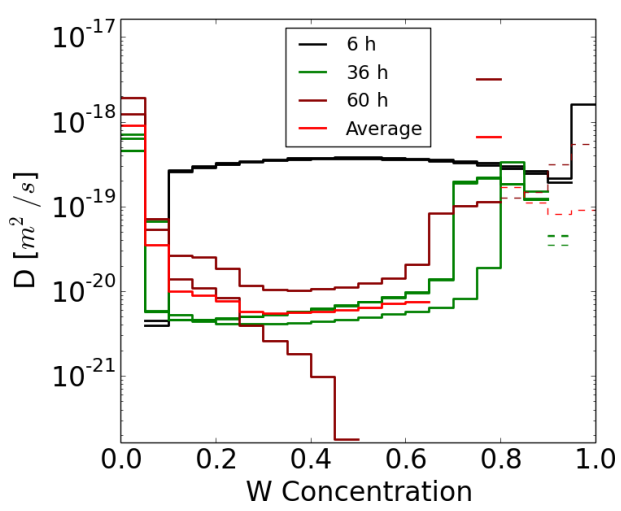
Figure 5.7.: Concentration dependent diffusion coefficient determined from fitting with forward simulations taking a concentration dependent $D(c)$ into account.

Procedure and results using the Boltzmann-Matano method

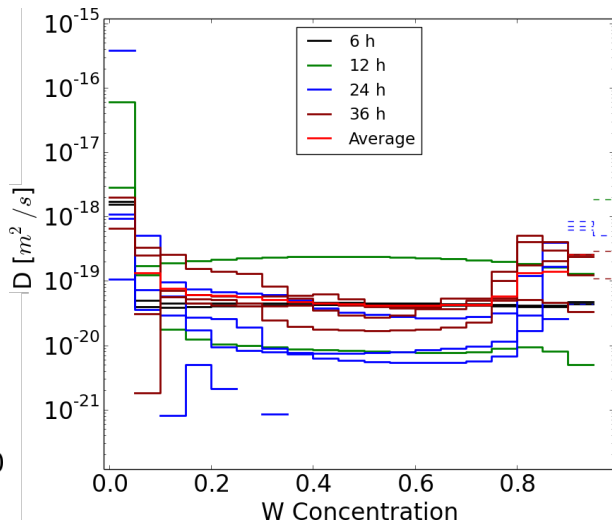
The other method of obtaining the diffusion coefficient, was by applying the Boltzmann-Matano method to the measured concentration profiles. Since the Boltzmann-Matano method requires knowing the derivative of the concentration profile, it cannot be applied to a discrete profile like the one we get from SimNRA. We therefore smoothen the profile by fitting an Akima spline through the mid-points of each discrete step in the concentration profile. Due to some variations, the measured concentration profiles are not strictly monotonous, which in turn causes fluctuations on the resulting diffusion coefficient. To adequately account for the related uncertainties, the resulting $D(c)$ was binned in steps of 5 %.

In figure 5.8, the resulting diffusion coefficients, determined with the Boltzmann-Matano method, are shown, as well as an average for each temperature series. Due to the fluctuations coming from the undulation of the concentration profile, some of the values in the 5 % bins become negative, which obviously could not be displayed on the logarithmic scale. Also, for intermediate concentrations, the resolution was not high enough to fill all of the bins, resulting in some of them having a value of 0, which cannot be displayed on the logarithmic scale either.

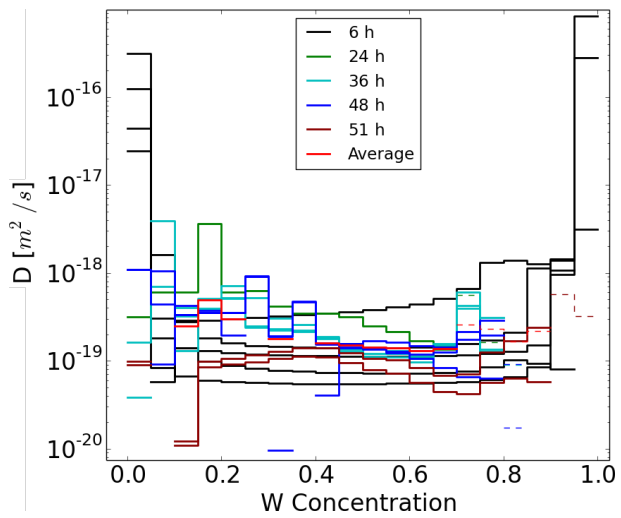
In figure 5.9, the average values for all temperatures, both obtained from the forward simulations with a concentration dependent $D(c)$ and from the Boltzmann-Matano method, are shown. The general trend of the data is a higher diffusion coefficient for very low W concentrations, which seems to decrease at least exponentially towards higher concentrations. The values for very high W concentrations may suggest a increase in the diffusion coefficient, but as explained, this data cannot be relied on. The results of the two different methods are generally in good agreement, only the values at 1000 K differ by a factor of 3. However, the scatter in the data from which these averaged diffusion coefficients are calculated and thus the corresponding standard deviations are very large. With uncertainties often in the range of 100 %, a factor of 3 difference might be considered to be within the error bars. Due to the data obtained with the Boltzmann-Matano method having some outliers because of the fluctuations resulting from the non-monotonous concentration profiles and from the empty bins, the results are now simply used as a means of confirming the results of the simulations, taking a concentration dependent $D(c)$ into account, but were not considered in the further evaluation of this data.



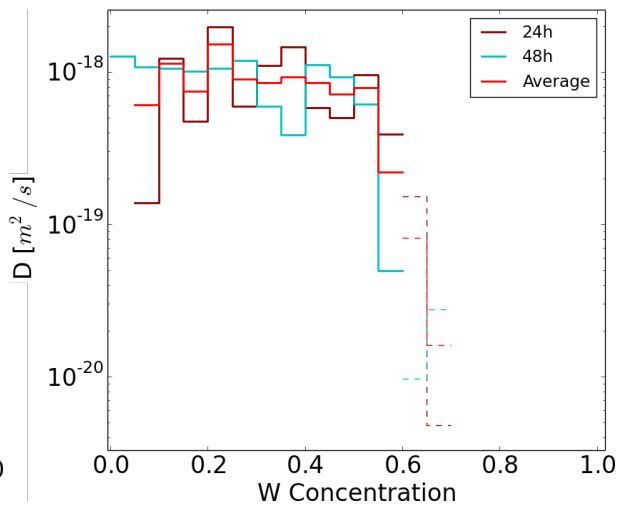
(a) Heated at 900 K



(b) Heated at 1000 K



(c) Heated at 1050 K



(d) Heated at 1100 K

Figure 5.8.: Concentration dependent diffusion coefficient determined with the Boltzmann-Matano method

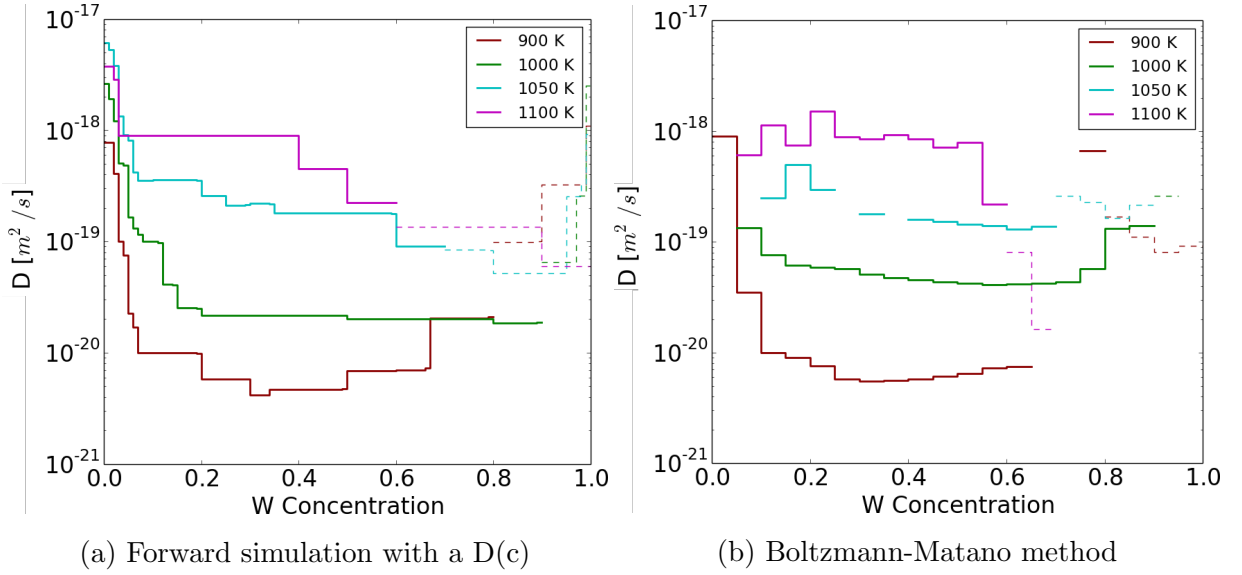


Figure 5.9.: Concentration dependent diffusion coefficient for all measured temperatures, obtained with (a) forward simulations and with (b) the Boltzmann-Matano method. For every temperature, the average over the D(c)-curves that have resulted for various annealing times is depicted.

5.1.4. The temperature dependence of the diffusion coefficient

We now want to determine the temperature dependence of the diffusion coefficient, according to equation 2.1.3. To achieve this, we look at each concentration value from 0% to 70%, in steps of 1%, and plot the Diffusion coefficient versus the reciprocal of the temperature (figure 5.10)

To each of these curves, we then fit the function

$$D_0 \exp\left(-\frac{\Delta H}{k_B T}\right),$$

by minimizing

$$\sum_{i=1}^4 \left(\frac{D_0 \exp\left(-\frac{\Delta H}{k_B T_i}\right) - D_{\text{meas}}(T_i)}{\sigma_{\text{meas}}(T_i)} \right)^2$$

with respect to D_0 , where $T_i = [900 \text{ K}, 1000 \text{ K}, 1050 \text{ K}, 1100 \text{ K}]$, $D_{\text{meas}}(T_i)$ are the averaged diffusion coefficients measured at the different temperatures and $\sigma_{\text{meas}}(T_i)$ are the standard deviations from that average. This fit for D_0 is then repeated for various values

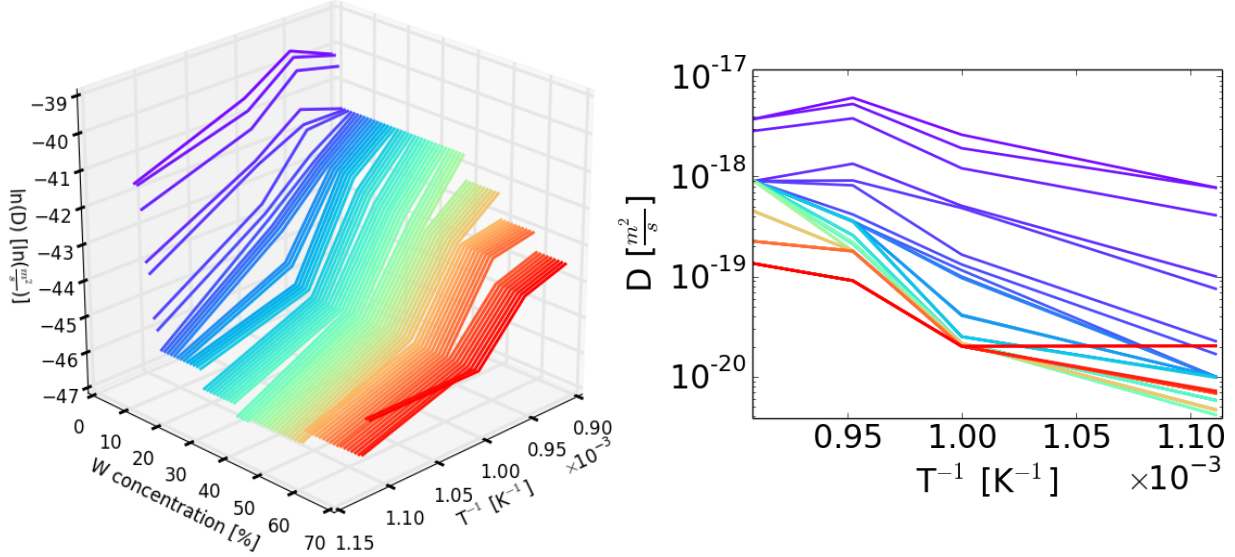


Figure 5.10.: Left: 3D plot depicting the dependence of D on the concentration and the temperature; Right: Logarithmic plot of the diffusion coefficient versus the reciprocal of the temperature, plotted for each concentration value between 0 and 70 %, keeping the same color scheme as in the left plot.

of the enthalpy ΔH . Literature values for ΔH range between 2.2 eV and 3.0 eV^{[1][14][18]}; we therefore iterate the enthalpy between 0.1 eV and 10 eV in steps of 0.1 eV and minimize the sum of all the errors at each concentration value

$$\sum_{c=0}^{0.7} \sum_{i=1}^4 \left(\frac{D_0(c) \exp\left(-\frac{\Delta H}{k_B T_i}\right) - D_{\text{meas}}(c, T_i)}{\sigma_{\text{meas}}(c, T_i)} \right)^2.$$

In the temperature range between 900 and 1100 K, we can describe the diffusion coefficient as

$$D(c, T) = D_0(c) \exp\left(-\frac{2.7 \text{ eV}}{k_B T}\right). \quad (5.1.1)$$

The concentration dependent part $D_0(c)$ can be seen in figure 5.11, along with error estimates from the minimization done to determine the individual values for D_0 . These error estimates are significantly smaller than the error bars calculated for the individual temperature-measurements, which can be seen in figure 5.12. They therefore shouldn't be considered a basis for an estimate of errors for values obtained with formula 5.1.1, but rather be an indicator of how well the calculated diffusion coefficients fit the expected Arrhenius behaviour.

In appendix A.6, the values for $D_0(c)$ can be found in tabular form.

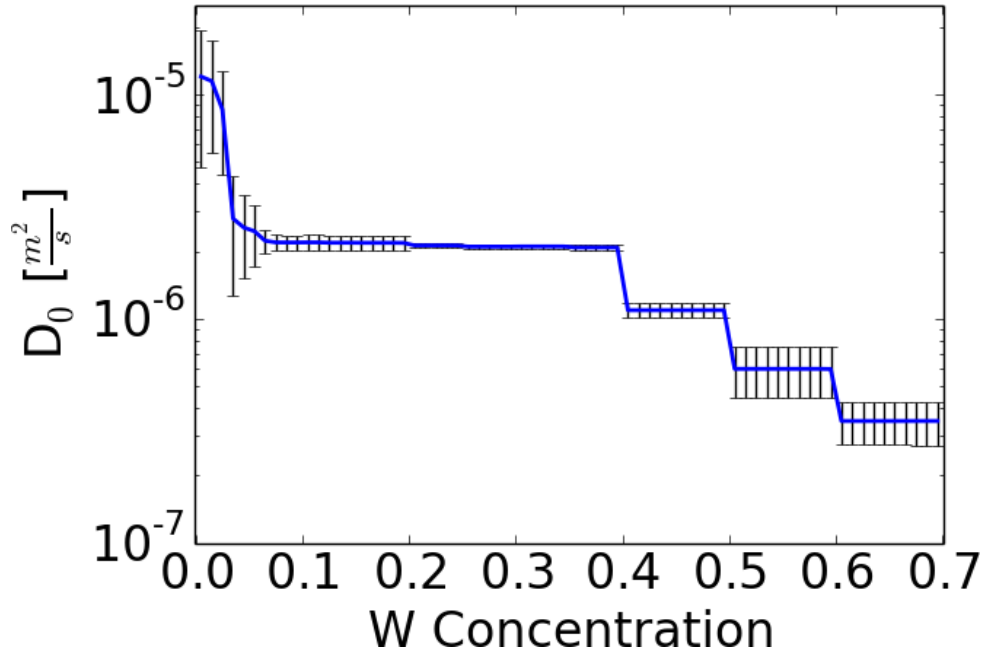
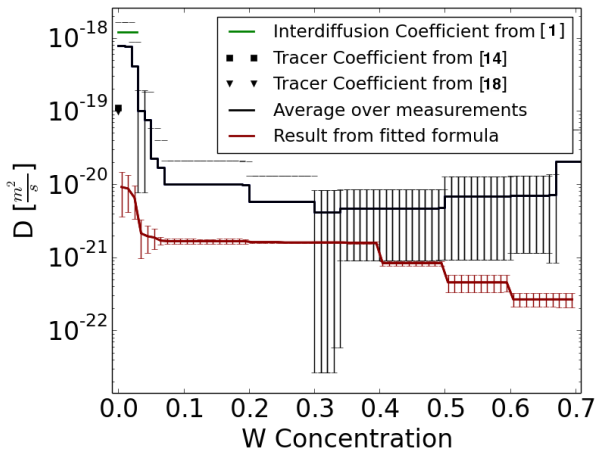
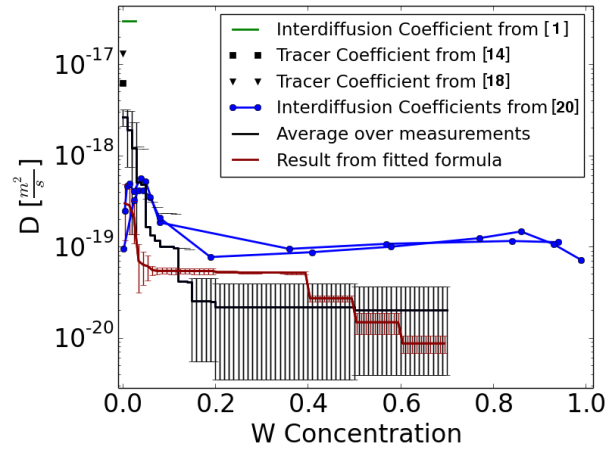


Figure 5.11.: The concentration dependent prefactor D_0 to the temperature dependent diffusion coefficient.

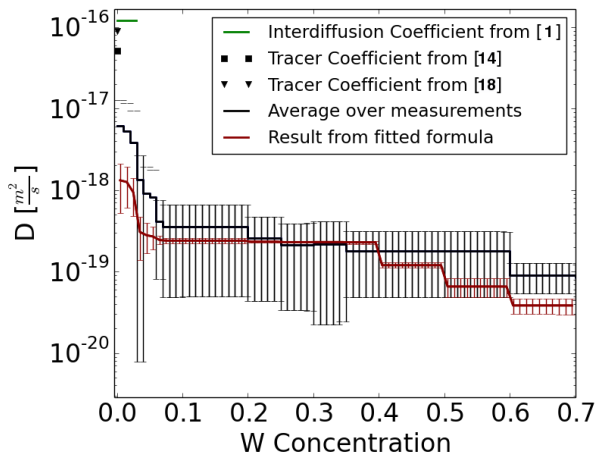
In figure 5.12, the average diffusion coefficients for each temperature, as well as the diffusion coefficients obtained from 5.1.1 are shown, together with preexisting literature values. Our results are in general agreement with the literature values from [1], [14] and [18], considering the steep decrease of $D(c)$ at low W concentrations. The results are lower than the ones from the previous Batchelor's thesis, which had less statistics and used the Boltzmann-Matano method. For the averaged values, the error bars, representing one standard deviation, are relatively big, due to the low statistics of our measurement and the relatively wide range of values obtained from these measurements. Some error bars extend down to negative values and can therefore not be displayed on a logarithmic scale.



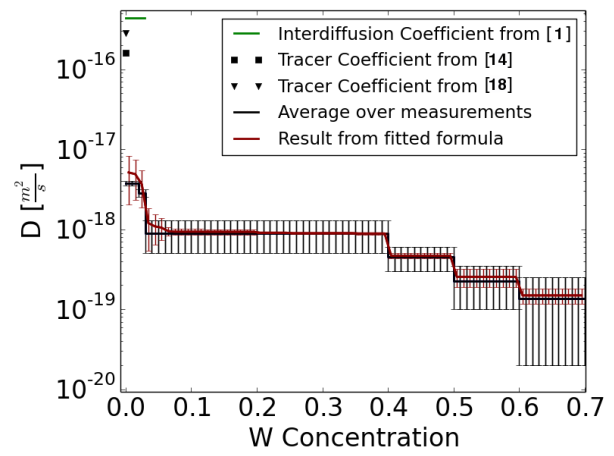
(a) T = 900 K



(b) T = 1000 K



(c) T = 1050 K



(d) T = 1100 K

Figure 5.12.: Concentration dependent diffusion coefficient determined in this thesis, compared to literature values.

5.2. Interdiffusion and phase growth studied by focussed ion beam cross sectioning

In this chapter, we measure concentration profiles by using a Focussed Ion Beam (FIB) to cut into the samples and view their cross sections under a Scanning Electron Microscope (SEM) (See figure 5.13).

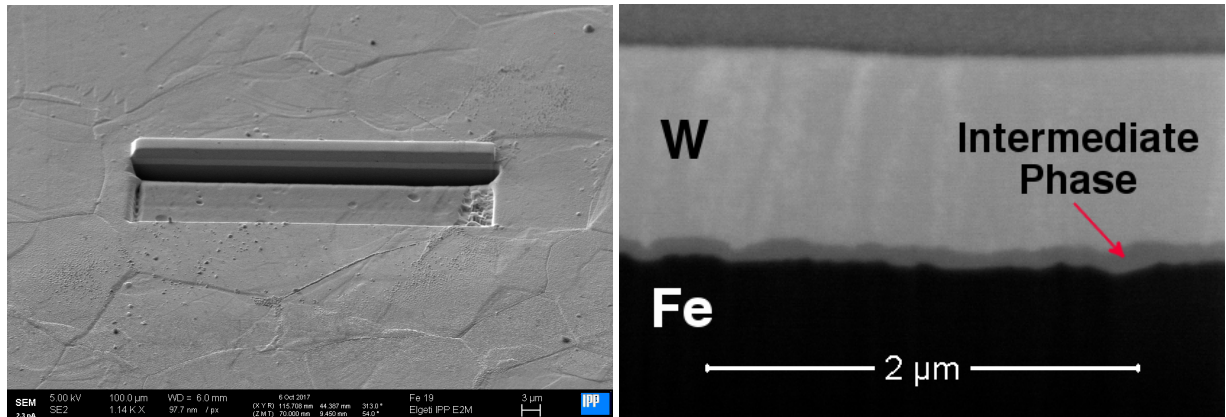


Figure 5.13.: SE Images of a FIB cut on a sample that was heated for 12 h at 1100 K. Left Picture: Overview; Right Picture: Zoom in on the cross section.

For samples that were heated at 1050 K and 1100 K and for more than 6 hours, we observe the formation of an intermediate phase, which can be seen from the different shades of grey in the SE image (figure 5.13 (right)). These shades of grey represent the detected secondary electrons coming from the material. Electrons are more likely to be backscattered by the tungsten than by the iron, due to its higher atomic number. Since a higher number of backscattered electrons then excite secondary electrons, we can differentiate different materials in the SE image, with a brighter shade of grey representing more tungsten.

For samples annealed for only 6 hours we do not observe a phase formation; this could be due to a nucleation process, delaying the initial formation of the phase, or simply because the phase was not yet big enough to be resolved with our methods.

5.2.1. The composition of the intermediate phase

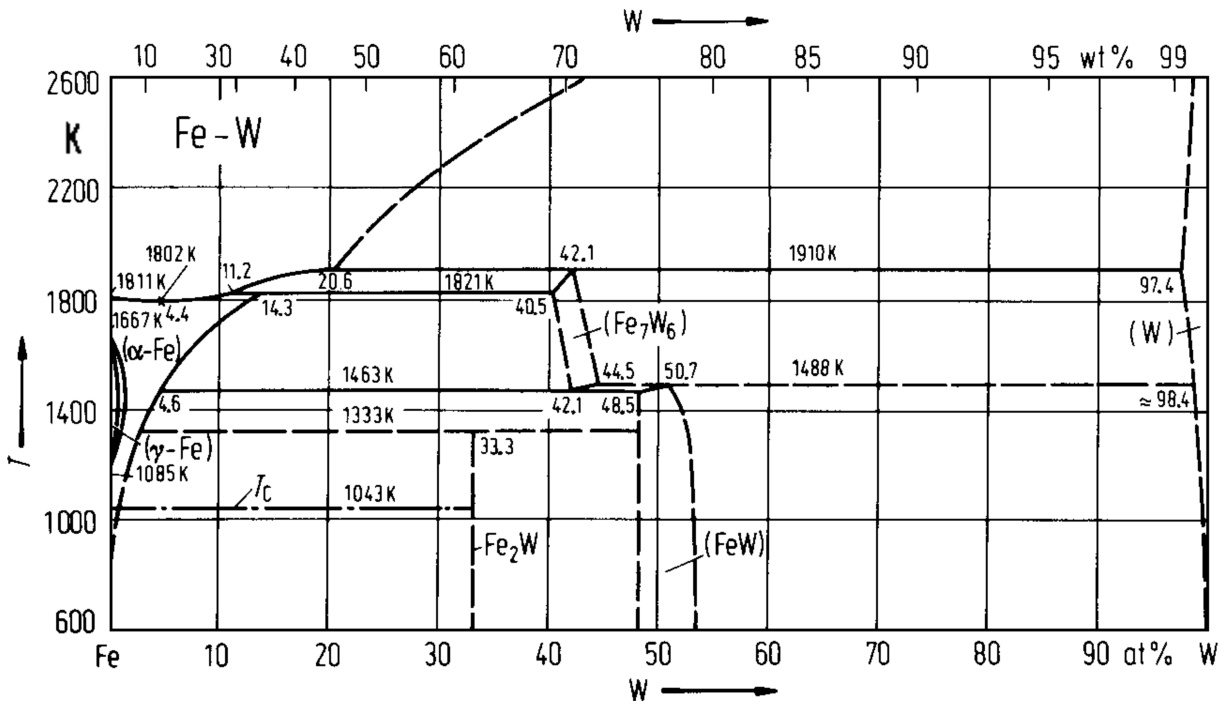


Figure 5.14.: Phase diagram of iron and tungsten. In the temperature-range, we are interested in, there are two iron tungsten phases; FeW and the metastable phase Fe_2W . Picture taken from [8].

The composition of the intermediate phase is of interest to us. According to the iron tungsten phase diagram (figure 5.14), there are two intermediate phases at the relevant temperatures: FeW and Fe_2W . According to the phase diagram Fe_2W is only metastable; it forms as a peritecticum at around 1060 K and then, after several hundreds of hours shifts into the stable FeW. At the timescales relevant for this thesis, however, Fe_2W can be considered stable.^[12]

On some of the FIB-cut cross sections, EDX maps were taken. From their spectra, the Fe L and the W M peaks were identified and the background subtracted from them (figure 5.15). The ratios between these corrected intensity peaks are then compared to standard intensity ratios and a concentration profile is then determined by least squares fitting, with the sum of the Fe and W signal being normalized to 100 %.

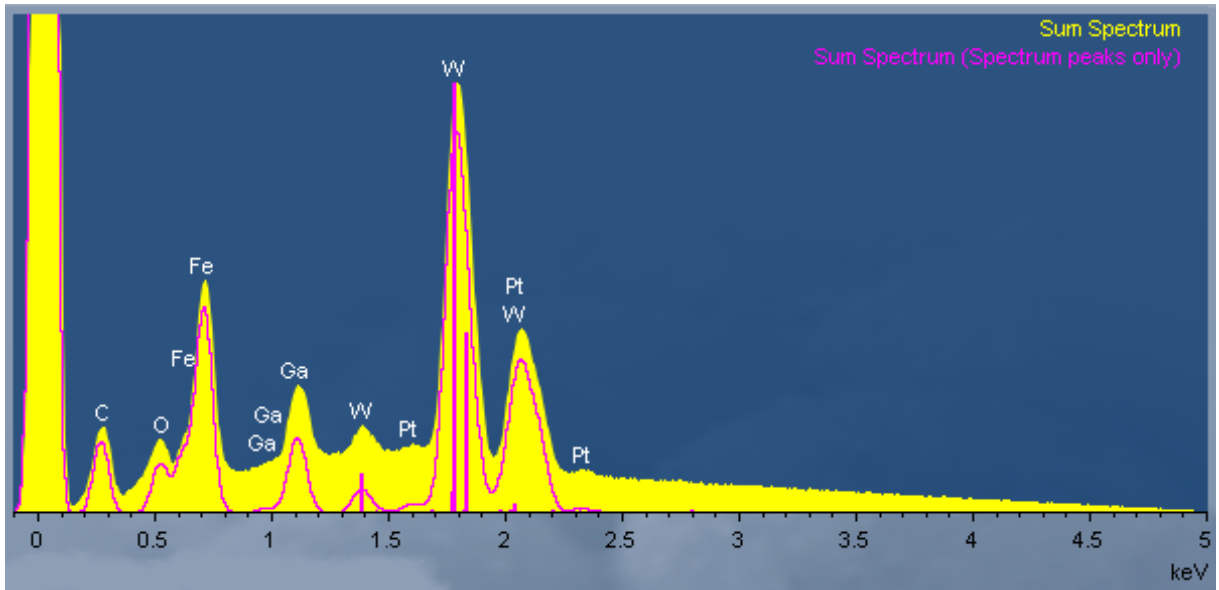


Figure 5.15.: Typical spectrum of an EDX map of a diffused W/Fe system. Drawn in yellow is the measured sum spectrum, in pink the spectrum with subtracted background.

In figure 5.16, two maps of the tungsten concentration at the cross section are shown for two different samples. Displayed in these maps is the tungsten concentration the software of the SEM calculated from the intensity of the W M peak and its ratio compared to the Fe L peak. From top to bottom one can see noisy signal from the deposited Pt layer that was needed for the FIB cut, a layer of varying thickness of a W/Fe composite, the pure W layer, another layer of the same W/Fe composite followed by the pure Fe layer at the bottom.

From these maps, we now want to obtain a depth profile of the concentration, resolving the intermediate phase between the iron and the tungsten layer. To get a better signal to noise ratio, we perform an average over the individual vertical lines. To make sure the intermediate layer does not get averaged away, we select positions at the interfaces where the thickness of the intermediate phase is relatively constant over some distance.

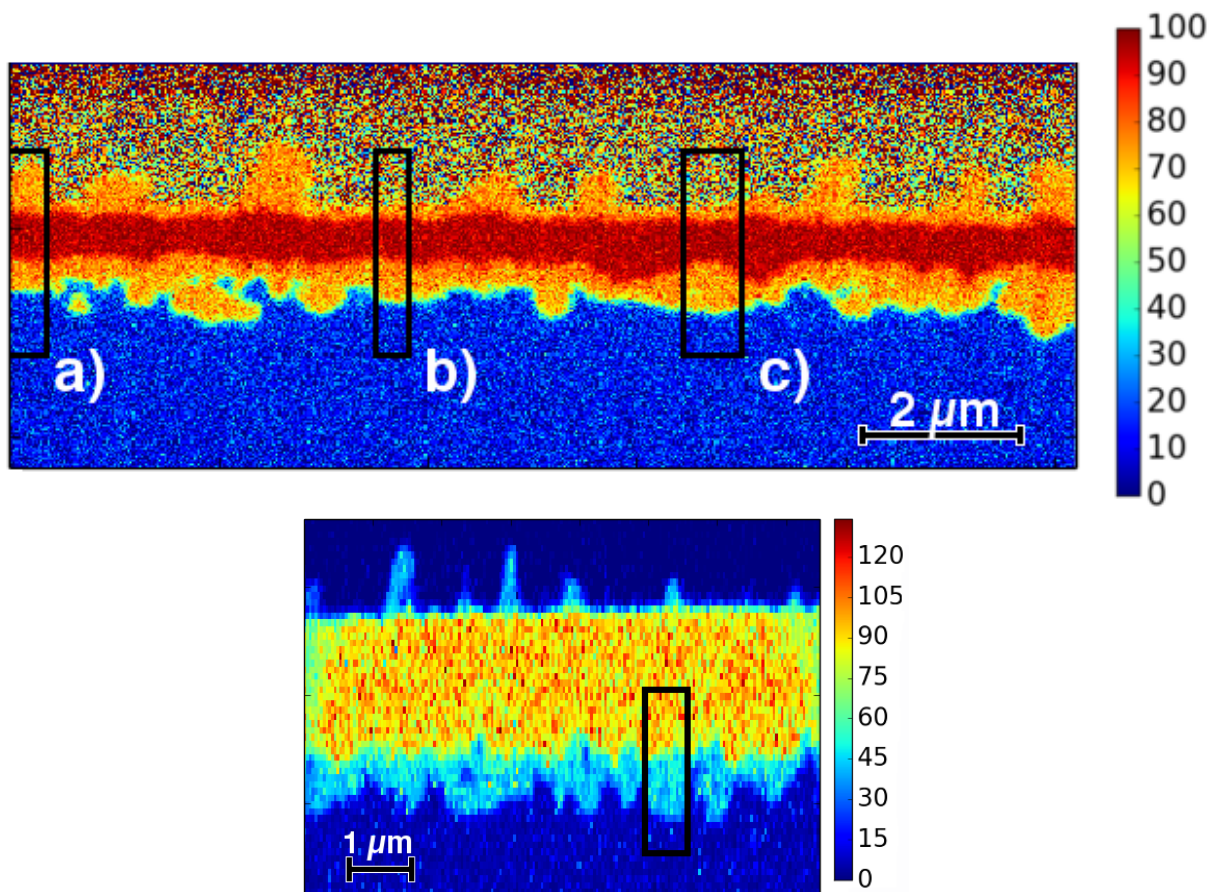


Figure 5.16.: Quantified EDX maps of the signal from the W M-Series, representing the concentration. The upper map is of a sample heated for 48 h at 1100 K, the lower one heated for 36 h at 1050 K. Positions at which the concentration profiles were viewed are also drawn in.

These profiles can be seen in figure 5.17. According to the lower map, the composition of the intermediate layer seems to lie between 40 and 53 %, suggesting FeW; according to the upper map, however, the composition of the intermediate phase seems to lie between 71 and 80 % W concentration suggesting something along the lines of FeW₂, which is not a known phase. This difference can be explained by the fact, that the maps were measured with two different machines using different software for the quantification. It appears, that the quantification of the concentrations from the EDX spectra is not reliable, likely due to the software used under- or overestimating the background under the intensity peaks, leading to an over- or underestimation of the height of certain peaks, resulting in incorrect values for the concentration.

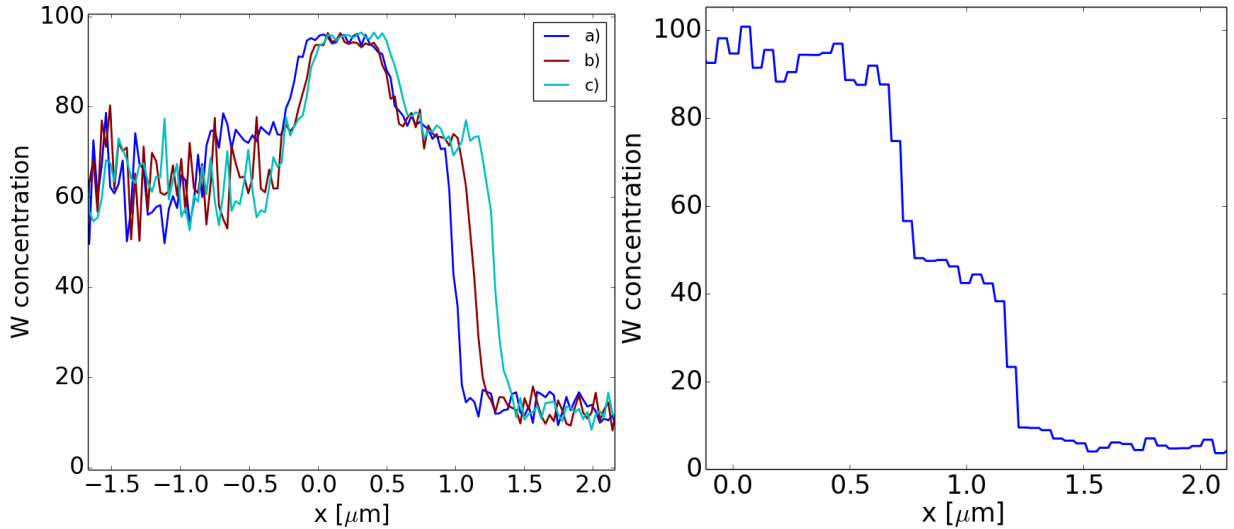


Figure 5.17.: Left: Averaged W concentration profiles at three positions on the FIB cut through a the sample heated for 48 h at 1100 K; Right: Averaged W concentration profile at one position on the FIB cut through a sample heated for 36 h at 1050 K.

Since the results from EDX are not conclusive, the composition was complementarily analyzed by means of sputter-XPS (see further below):

On samples that were heated at higher temperatures, we observe material breaking through the W layer and having migrated to the surface of the substrate, primarily at grain boundaries (see figure 5.18). This becomes especially clear for samples with were annealed for shorter times, where an intermediate phase has settled close to the grain boundary of origin, as can be seen in figure 5.24.

For samples that were heated for longer times, more of the material's surface is covered; in some cases all of it. This material on the surface could already be seen in some of the RBS spectra.

The faster diffusion of material to the surface through damaged grain boundaries should not have any direct influence on bulk diffusion and phase formation at positions away from the grain boundaries. It therefore should not distort the result for the interdiffusion coefficient between iron and tungsten, as long as we exclude those spots from the data. For the RBS analysis, however, this has caused some problems, which have been addressed in section 5.1.

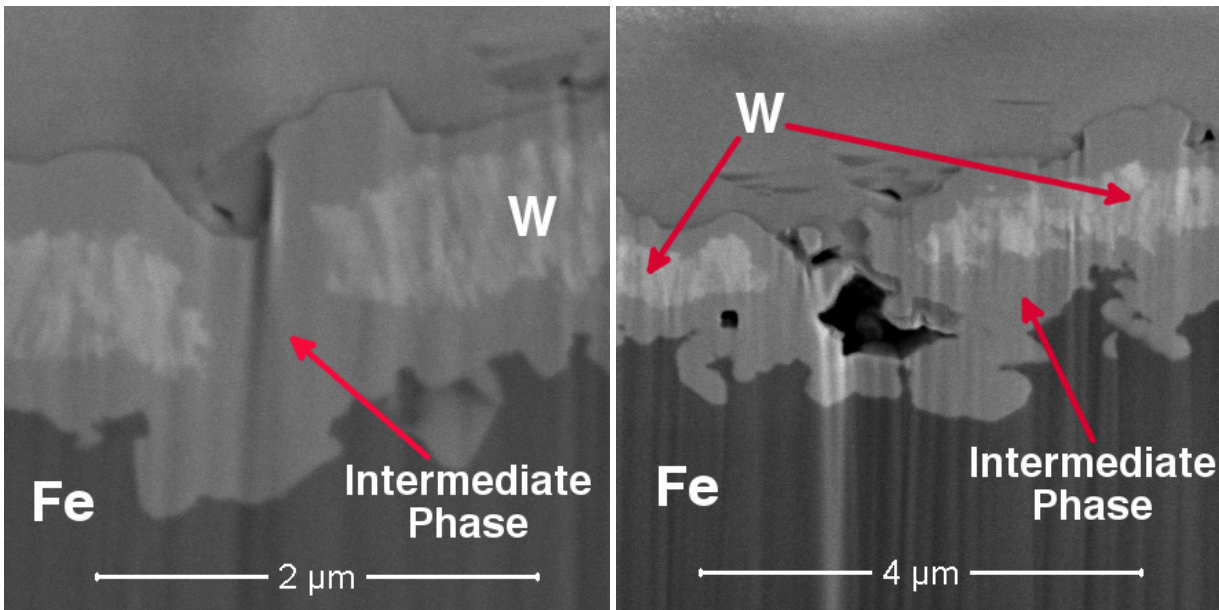


Figure 5.18.: SE images of FIB cuts across rifted grain boundaries where iron has moved to the surface. The phase at the surface, previously determined to be most likely Fe_2W , and the phase between the iron and the tungsten seem to be the same material.

On the other hand, the fact that the intermediate phase, formed at the interface, also appears at the sample surface, can be used to our advantage to determine the composition of the intermediate phase by means of sputter-XPS.

In figure 5.18, the material at the top of the tungsten layer and the intermediate phase below the tungsten layer have the same shade of grey which indicates that they have the same stoichiometry. Likewise, the phases in the EDX maps in figure 5.16 appear to have the same compositions, which can be especially seen from line a) in figure 5.17, where the material at the top is also relatively thick. Since there are only two possible intermediate phases, FeW and Fe_2W , this implies that the material at the top is in fact identical to the intermediate phase.

If the intermediate phase were only between the tungsten layer and the iron substrate, we couldn't investigate it with sputter XPS, since sputtering through the thick tungsten layer would take an unreasonable long amount of time. However, since the same material is also right at the surface, we don't have this problem. In figure 5.19, the results of the sputter XPS measurement are shown.

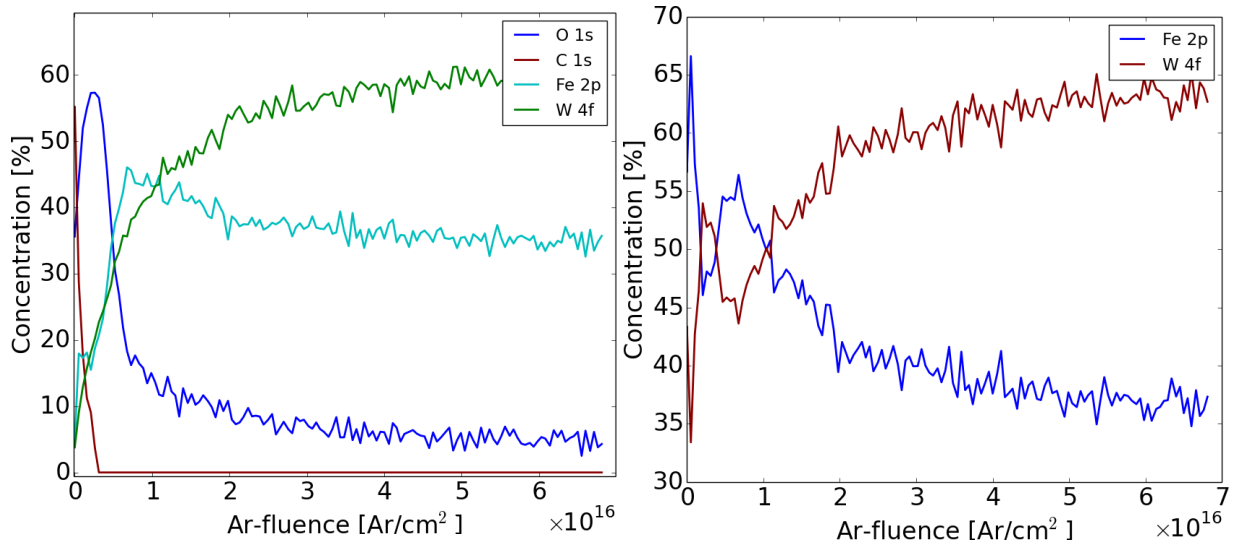


Figure 5.19.: Result of sputter XPS measurement of the surface of a sample that was heated for 48 h at 1100 K. Left: Signal for Fe and W as well as C and O plotted vs the Ar-fluence used to remove the layer until that point. Right: Signal only for Fe and W, normalized to one.

It can be seen, that there is a significant amount of oxygen at the surface of the substrate, which comes from the sample oxidizing after taking it out of the oven. This oxygen leads to a distortion of the data for low fluences, i.e. close to the surface of the material. For higher fluences, an equilibrium has developed, which is determined by the stoichiometry and the relative sputtering yields of Fe and W. Extrapolating the trend from these two signals back to the surface, however, the data seems to suggest a ratio of 2Fe:1W, which would indicate that the intermediate phase formed is Fe_2W . A result we are more inclined to believe than the ones from the EDX measurements.

Still, a possible strong concentration gradient within the XPS-information depth near the target surface leads us to not be completely confident with the identification of Fe_2W . Forward simulations of the sputter-XPS depth profiles, as done in [11], could yield a more trustworthy result. Another possibility would be to employ time-of-flight Rutherford backscattering spectrometry (TOF-RBS), which has a better depth resolution than regular RBS, to determine the stoichiometry of the intermediate phase that fully covers the surface of samples annealed at the highest temperatures for longer times.

5.2.2. The growth rate of the intermediate phase

As was shown in section 2.3.3, we can use the growth of the intermediate phase with respect to time, to determine the diffusion coefficient in the intermediate phase via equation 2.3.22. For this, we first need to determine the average thickness of the intermediate phase. To this end, we first take all the individual SEM pictures taken along each FIB-cut, and stitch them together. Next, we manually mark the intermediate phase (see figure 5.20), using the program Gwyddion, and vertically sum the marked pixels.

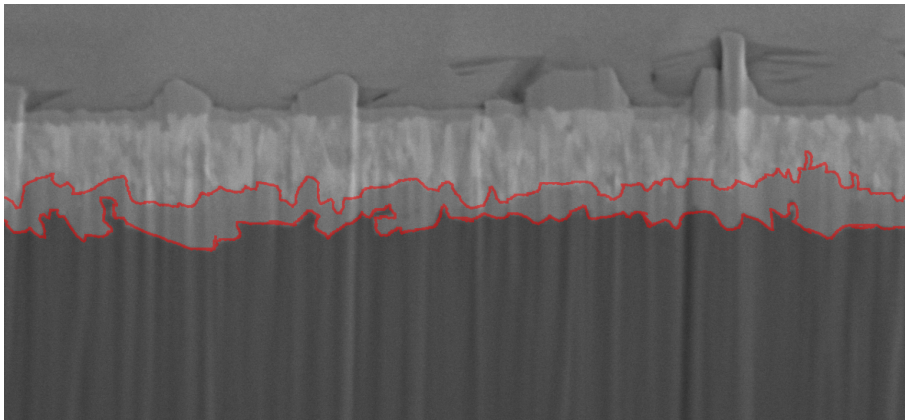


Figure 5.20.: SEM Picture of a sample that was heated for 24 h at 1100 K. The intermediate phase was manually marked, to extract the average width of the intermediate phase.

In certain spots, mostly along grain boundaries, the underlying iron has broken through the tungsten and has migrated to the surface. These spots were not marked and were left out of the calculations of the average thickness of the phase, since there obviously is another process at work besides the interdiffusion we want to study. This process will be discussed briefly in a later section. Now we only have to multiply the result with a proportionality factor to convert the number of pixels into a distance, to get the thickness of the intermediate phase for each horizontal position along the cut. Two examples of this can be seen in figure 5.21.

From these data, we calculate the average and the standard deviation thereupon. For the average, positions where the width was 0 were not taken into account, since this were the spots where no phase was marked, because material has broken through the tungsten, or because we were at the edge of the FIB-cut.

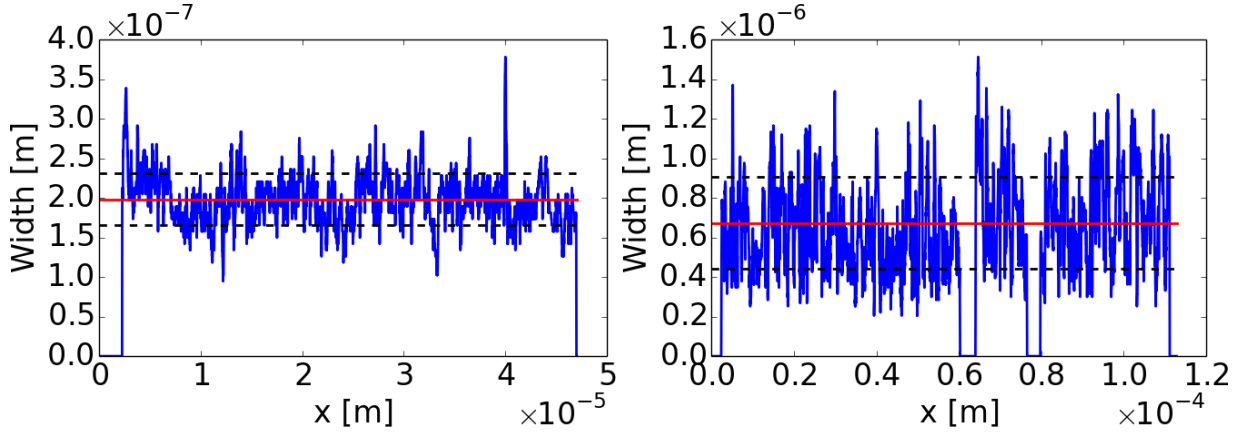


Figure 5.21.: Examples for the width of the intermediate phase as a function of the horizontal position. The sample in the left picture was heated for 12 hours at 1050 K, the sample in the right picture for 48 hours at 1100 K. Indicated in red is the average thickness; the dashed lines represent one standard deviation.

It becomes clear that the intermediate phase in samples, that are heated for longer times and for higher temperatures, are not only thicker, but also have a much higher variation on that thickness, as can be seen in table 5.1.

T [K] \ t [h]	12	24	36	48	51
1050	198 nm ±32 nm	338 nm ±119 nm	351 nm ±127 nm	396 nm ±150 nm	415 nm ±158 nm
1100	222 nm ±49 nm	469 nm ±171 nm		674 nm ±234 nm	

Table 5.1.: The average thicknesses for each sample together with the error estimate of one standard deviation.

In figure 5.22, the average thickness is displayed with respect to time. It can be seen, that it follows the expected square root behaviour, which allows us to apply equation 2.3.22. For the fits through the data points, we allowed for a time offset, since we observe no phase formation within at least the first 6 hours, which could mean a possible nucleation effect.

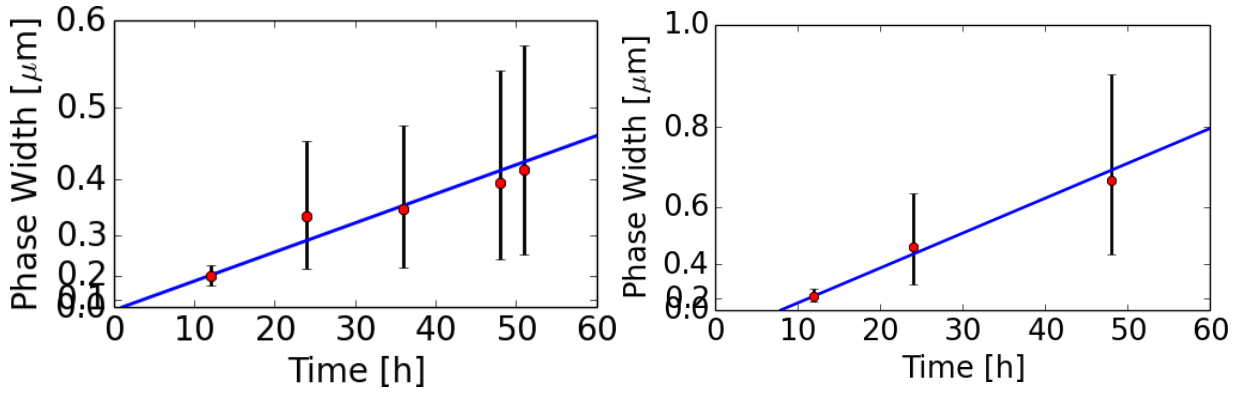


Figure 5.22.: Thickness of the intermediate phase versus the annealing time. In the left plot are the thicknesses of samples heated at 1050 K, in the right plot thicknesses of samples heated at 1100 K. The fitted functions are $60 \left[\frac{nm}{\sqrt{h}} \right] \cdot \sqrt{t - 1.0 [h]}$ for the 1050 K case and $110 \left[\frac{nm}{\sqrt{h}} \right] \cdot \sqrt{t - 7.9 [h]}$ for the 1100 K case. The tick spacing on the y axis increases quadratically, such that the square root dependence is displayed as a straight line.

Further evidence regarding there being a nucleation time before the phase is formed are our measurement series from the samples heated at 1000 K. The samples annealed for 24 hours and for 36 hours do not show a continuous intermediate phase between the tungsten and the iron, only small 'seeds' of this material, as can be seen in figure 5.23 (top). After 60 hours of annealing, a continuous phase was observed, with a thickness of (418 ± 163) nm. So the intermediate layer seems to form around a small nucleus that has to be formed first.

For 900 K, no phase formation was observed after a diffusion time of 60 hours.

From the growth of the phase thickness with respect to time, we can now determine the 'effective' diffusion coefficient K . At 1050 K, this value is $1.0 \times 10^{-18} \frac{m^2}{s}$, at 1100 K $3.4 \times 10^{-18} \frac{m^2}{s}$. Using equation 2.3.22, the interdiffusion coefficient in the intermediate phase can be determined from these values. For this, the values of $K^{\beta\alpha}$ and $K^{\beta\gamma}$ need to be known, which can be calculated via equations 2.3.18 and 2.3.19. These equations require the knowledge of the solubility ranges and the diffusion coefficients in the pure materials. While the solubility ranges $c_{\alpha 0}$ and $c_{\gamma 0}$ can be learned from a tungsten iron phase diagram to be $c_{\alpha 0} = 0.01$ and $c_{\gamma 0} = 0.995$, the solubility range for the intermediate phase cannot be learned from a phase diagram, because Fe_2W is not actually a stable phase and is therefore only drawn in as a thin line at 33.3% tungsten concentration.

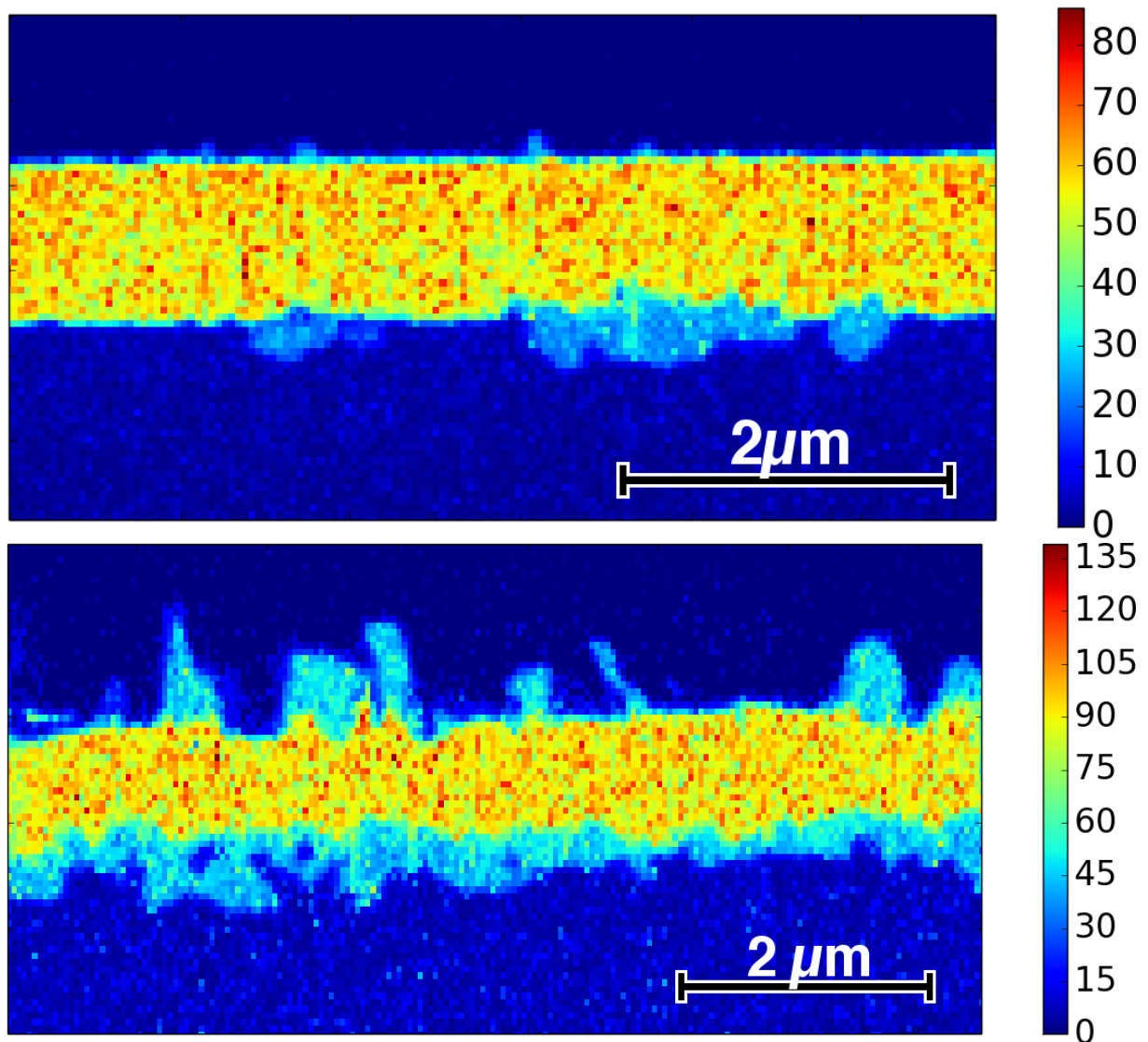


Figure 5.23.: EDX maps of FIB cuts done on samples that were annealed at 1000 K. The upper graph is the map of a sample heated for 24 hours, the lower one of a sample heated for 60 hours.

As a first approximation, we choose the solubility range to lie between 28% – 38%, since the EDX maps also yielded a relative variation of $\pm 5\%$ as can be seen in figure 5.17, even though we cannot rely on the absolute values deduced from the EDX measurements. D_{Fe} , the diffusion coefficient in the pure iron phase, can be taken from our RBS measurements; D_W has to be extrapolated to concentrations of 100% tungsten, since we were only able to obtain results for up to 70% tungsten. For this extrapolation, we can either continue the trend of a decreasing D_W or follow the trend of a the interdiffusion coefficient increasing for higher concentrations, that some of the measurements may suggest. Fortunately, the diffusion coefficients in the pure materials don't actually have a very strong influence on the diffusion coefficient in the intermediate phase, and don't significantly change its value,

even when D_W is varied over several orders of magnitude. The resulting interdiffusion coefficient in the Fe_2W phase can be found in table 5.2.

T [K]	$D_{Fe} [10^{-19} \frac{m^2}{s}]$	$D_W [10^{-19} \frac{m^2}{s}]$	$D_{Fe_2W} [10^{-19} \frac{m^2}{s}]$
1050	5 – 20	0.05 – 5	10.5 ± 0.5
1100	20 – 80	0.1 – 10	36 ± 1

Table 5.2.: The interdiffusion coefficient in the Fe_2W phase at 1050 K and 1100 K along with the ranges for D_{Fe} and D_W that went into its calculations. D_{Fe} obtained from our RBS measurements.

For the trace diffusion of tungsten in iron, there exist some literature values, which are a few orders of magnitude higher than the values we measure at very low tungsten concentrations.^{[1][14][18]} Since our diffusion coefficients, determined from RBS-measurements, are only 'effective' diffusion coefficients averaged over the formed phase, it may make more sense to take the literature values for this calculation. For this case, the resulting diffusion coefficient in the Fe_2W phase is about 30 % higher than in the previous calculation. The values can be found in table 5.3.

T [K]	$D_{Fe} [10^{-19} \frac{m^2}{s}]$	$D_W [10^{-19} \frac{m^2}{s}]$	$D_{Fe_2W} [10^{-19} \frac{m^2}{s}]$
1050	500 – 1200	0.05 – 5	13.5 ± 0.5
1100	1600 – 4400	0.1 – 10	46 ± 3

Table 5.3.: The interdiffusion coefficient in the Fe_2W phase at 1050 K and 1100 K along with the ranges for D_{Fe} and D_W that went into its calculations. D_{Fe} set to be in the range of the literature values.

5.2.3. Grain boundary diffusion

As already stated several times in the previous sections, we observe iron atoms to have broken through the tungsten layer and moved to the surface of the W/Fe system, for samples annealed at higher temperatures (see figure 5.18). These breaks are found to have primarily occurred along grain boundaries (see figure 5.24). The iron at the top is found to be most likely in the form of the metastable Fe_2W phase, which is also found between the iron and the tungsten layer.

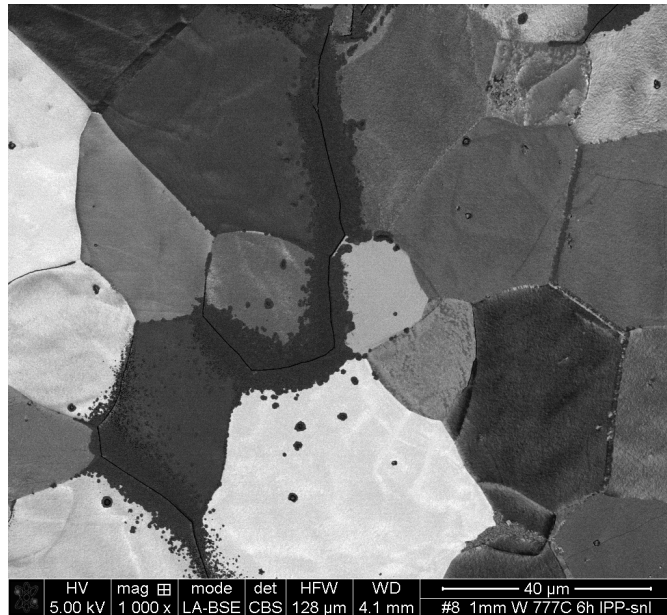


Figure 5.24.: The surface of an iron substrate with one μm tungsten on top, after annealing for 6 hours at 1050 K. Along some grain boundaries, material has flowed out. Picture taken in a scanning electron microscope, using a backscattering electron detector.

A plausible driving mechanism for the breaking of the W layer and the formation of the intermediate phase is that the tungsten layer on top experiences a tensile stress during the annealing, due to the higher thermal expansion of iron. It is reasonable to assume that the anisotropy at the grain boundaries and the stress-induced high density of lattice defects enhance interdiffusion and the formation of the intermediate phase. Thus, a breach is formed, through which iron can diffuse faster (see the interdiffusion coefficients in the intermediate phase compared to the values for tracer diffusion in tables 5.2 and 5.3), leading to excess material at the surface, which can again form an intermediate phase with the tungsten below.

Another explanation could be that the tungsten actually gets ripped apart by the expansion of the underlying iron, leading the iron to be able to diffuse to the surface along open cracks in the W layer, via surface diffusion. The surface-diffused iron inside the crack and on the surface would then interdiffuse with the tungsten, forming the intermediate phase.

6. Summary

In this work, the interdiffusion of iron and tungsten was studied. The underlying motivation is the assessment of the suitability of the low-activation steel EUROFER as a wall material for some (recessed) areas in future nuclear fusion power plants. The experimental strategy was to produce binary layered W/Fe systems, by sputter-depositing a 1 μm thick tungsten layer onto pure iron substrates. These binary systems were then annealed at temperatures ranging between 900 and 1100 K and for times between 6 and 60 hours, to induce interdiffusion.

Concentration profiles perpendicular to the iron-tungsten interface were then measured by means of Rutherford backscattering spectrometry. To determine interdiffusion coefficients from these concentration profiles, a finite difference code was written, that iteratively solves the diffusion equation and can thus simulate interdiffusion with a concentration dependent interdiffusion coefficient $D(c)$. For each annealing condition, the interdiffusion is simulated and the concentration dependence of the diffusion coefficient is varied in the calculations, until the resulting concentration profiles match measured ones. Complementarily, the interdiffusion coefficients were calculated from the measured depth profiles by means of the Boltzmann-Matano method. The applicability of this method is limited by the coarse resolution of the measured concentration profiles. Nevertheless, the order of magnitude and trends, that have resulted from the analysis with forward simulations, are confirmed.

Interdiffusion at temperatures between 900 and 1100 K can be described by an effective coefficient of the form

$$D(c, T) = D_0(c) \exp\left(-\frac{2.7 \text{ eV}}{k_B T}\right),$$

with $D_0(c)$ decreasing from $(1.2 \pm 0.7) \cdot 10^{-5} \frac{\text{m}^2}{\text{s}}$ at 0 % tungsten to $(3.5 \pm 0.8) \cdot 10^{-7} \frac{\text{m}^2}{\text{s}}$ at 70 % tungsten.

The value of 2.7 eV for the activation enthalpy ΔH is in line with the literature for the diffusion of trace concentrations of tungsten in iron, considering the steep increase of

D(c) towards low tungsten concentrations. At intermediate concentrations, our resulting diffusivity values lie below the ones from a previous work, where D(c) was determined at 1000 K using the Boltzmann-Matano method.

SEM images and EDX maps were acquired on cross sections that were milled into the material surfaces with a focussed ion beam. For temperatures above 1000 K, the formation of an intermediate phase was observed. The stoichiometry resulting from XPS measurements is consistent with the known metastable phase Fe₂W. The diffusion coefficient inside that phase was found to be in the range of $(12\pm 2)\cdot 10^{-19}\frac{m^2}{s}$ at 1050 K and $(42\pm 7)\cdot 10^{-19}\frac{m^2}{s}$ at 1100 K, depending on the assumed values for the tracer diffusion coefficients. A finite difference simulation code was developed, that is able to simulate diffusion processes taking into account the formation of such intermediate phases. The power of this simulation tool goes beyond what it has been applied for in this work: It can be used to study diffusion processes involving several intermediate phases, including also concentration dependencies of the diffusion coefficients within each phase.

7. Conclusion

Due to the lateral averaging and the neglect of phase formation, the interdiffusion coefficient that results from the RBS analyses has to be considered an 'effective' $D(c)$. Nevertheless, this effective $D(c)$ could now be implemented in the SDTrimSP code for calculation of sputtering with interdiffusion. A better approach, however, might be to transfer the reactive diffusion code developed in this thesis into the SDTrimSP code and properly take reactive diffusion into account, using the interdiffusion coefficients determined in section 5.2.2.

Regardless of the approach (effective $D(C)$ or reactive diffusion), the interdiffusion coefficients determined in the course of this thesis at intermediate concentrations are orders of magnitude lower than the tracer diffusion coefficients from the existing literature. Those literature values were the basis of first published SDTrimSP simulations performed to determine the effect of interdiffusion on the evolution of sputter yields with fluence. This means, the temperature at which enrichment of tungsten on the surface of EUROFER would be prevented by interdiffusion, would be higher than so far predicted. This increases the potential application of low-activation steels as first wall materials.

8. Acknowledgements

I would like to thank Prof. Dr. Ulrich Stroth for enabling me to write my thesis at the Max-Planck Institute for Plasma Physics.

To my supervisor Dr. Martin Oberkoffer, I would like to express my deep gratitude for his constant advice and support during the completion of this work, and I would like to thank Stefan Elgeti for the countless hours he spent taking SEM images of my samples. For the instructions on multiple devices and their time spent helping me with analyzing my samples, I would like to give my thanks to Gabriele Matern, Till Höschen, Martin Balden and Tiago Silva, as well as Liang Gao and Stefan Kapser. And I would like to thank the members of the E2M group for their warmth and hospitality and their general supportiveness.

Finally, I would like to thank my parents for supporting me throughout my studies.

A. Attachments

A.1. Discretization of the Differential equations

In this appendix, a more in-depth description can be found, on how the differential equations in sections 2.3.1 and 2.3.2 are discretized.

Time is discretized in steps of Δt where n is the index of the time steps, ranging from 0 to N_T , and space is discretized in steps of Δx , where i is the index of the spatial steps, ranging from 0 to N_L (see figure A.1).

To discretize equation 2.3.1, one needs to replace the derivatives with difference quotients. The discretization of the time derivative

$$\frac{\partial c}{\partial t}(x, t) \rightarrow \frac{c_i^{n+1} - c_i^n}{\Delta t}$$

is straight-forward, essentially being the definition of a derivative.

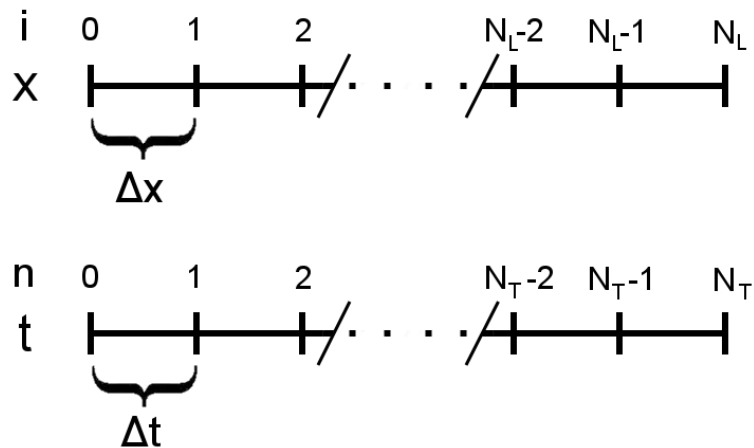


Figure A.1.: Discretization of the space and time intervals in N_L and N_T grid points spaced Δx and Δt apart, respectively.

The second derivative in space is discretized in the following manner: First, one needs to do a Taylor-Expansion at c_{i+1} and c_{i-1} :

$$c_{i+1} = c_i + \Delta x \frac{\partial c_i}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 c_i}{\partial x^2} + \frac{\Delta x^3}{6} \frac{\partial^3 c_i}{\partial x^3} + \frac{\Delta x^4}{24} \frac{\partial^4 c_i}{\partial x^4} + O(\Delta x^5)$$

$$c_{i-1} = c_i - \Delta x \frac{\partial c_i}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 c_i}{\partial x^2} - \frac{\Delta x^3}{6} \frac{\partial^3 c_i}{\partial x^3} + \frac{\Delta x^4}{24} \frac{\partial^4 c_i}{\partial x^4} - O(\Delta x^5)$$

Then, those two equations are added together, so the odd numbered derivatives cancel, and solve for the second derivative of c_i with respect to x , resulting in

$$\frac{\partial^2 c_i}{\partial x^2} = \frac{1}{\Delta x^2} \left(c_{i+1} - 2c_i + c_{i-1} + O(\Delta x^4) \right). \quad (\text{A.1.1})$$

In the following, terms of order Δx^2 and higher are neglected.

In combination with Dirichlet boundary conditions, in matrix form equation A.1.1 looks like the following:

$$\frac{\partial^2}{\partial x^2} \begin{pmatrix} c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_{L-1}^n \\ c_{N_L-1}^n \end{pmatrix} = \frac{1}{\Delta x^2} \begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \ddots & -2 & 1 \\ 0 & 0 & 0 & \dots & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} c_1^n \\ c_2^n \\ c_3^n \\ \vdots \\ c_{L-1}^n \\ c_{N_L-1}^n \end{pmatrix} + \frac{1}{\Delta x^2} \begin{pmatrix} c_0^n \\ 0 \\ 0 \\ \vdots \\ 0 \\ c_{N_L}^n \end{pmatrix} \quad (\text{A.1.2})$$

Or in short notation:

$$\frac{\partial^2}{\partial x^2} \vec{c}^n = \frac{1}{\Delta x^2} M \cdot \vec{c}^n + \frac{1}{\Delta x^2} \vec{c}_{BC}. \quad (\text{A.1.3})$$

We choose to implement the Crank-Nicholson method, which is numerically stable under any conditions and second-order accurate. For this, we must take the average of the discretization of the second derivative in space at the timestep n and the timestep $n + 1$, and set it equal to the discretization of the time derivative, resulting in equation 2.3.2

To discretize equation 2.1.2 we first discretize the right-hand-side of the equation. In matrix-notation, this looks as follows:

$$\frac{\partial}{\partial x} \left(D(\bar{c}^n) \frac{\partial c}{\partial x} \right) \rightarrow \frac{1}{4\Delta x^2} M \cdot D \cdot \left[M \cdot \bar{c}^n + \vec{c}_{BC} \right] \quad (\text{A.1.4})$$

Where

$$M = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & \dots & 0 & 0 \\ 0 & -1 & 0 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \ddots & 0 & 1 \\ 0 & 0 & 0 & \dots & -1 & 0 \end{pmatrix},$$

$$D = \begin{pmatrix} D_1 & 0 & \dots & 0 \\ 0 & D_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & D_{N_L-1} \end{pmatrix},$$

and

$$\vec{c}_{BC} = \begin{pmatrix} -c_0^n \\ 0 \\ 0 \\ \vdots \\ 0 \\ c_{N_L}^n \end{pmatrix}.$$

Applying the Crank-Nicholson method, we end up at equation 2.3.5.

A.2. Algorithms used to solve the linear equations

In this appendix, a more detailed description of the methods used to solve the linear equation systems can be found. For the constant D simulations, the Thomas algorithm was used, for the concentration dependent $D(c)$ simulations a variant thereof.

We start from a linear equation of the form

$$\begin{pmatrix} b_1 & c_1 & 0 & \dots & 0 & 0 \\ a_1 & b_2 & c_2 & \dots & 0 & 0 \\ 0 & a_2 & b_3 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \ddots & b_{N-1} & c_{N-1} \\ 0 & 0 & 0 & \dots & a_{N-1} & b_N \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-1} \\ d_N \end{pmatrix},$$

And bring it into the form

$$\begin{pmatrix} 1 & \tilde{c}_1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \tilde{c}_2 & \dots & 0 & 0 \\ 0 & 0 & 1 & \ddots & \vdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \ddots & 1 & \tilde{c}_{N-1} \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \tilde{d}_3 \\ \vdots \\ \tilde{d}_{N-1} \\ \tilde{d}_N \end{pmatrix}.$$

Where

$$\tilde{c}_i = \begin{cases} \frac{c_1}{b_1} & \text{for } i = 1 \\ \frac{c_i}{b_i - \tilde{c}_{i-1}a_{i-1}} & \text{for } i > 1 \end{cases},$$

and

$$\tilde{d}_i = \begin{cases} \frac{d_1}{b_1} & \text{for } i = 1 \\ \frac{d_i - \tilde{d}_{i-1}a_{i-1}}{b_i - \tilde{c}_{i-1}a_{i-1}} & \text{for } i > 1 \end{cases}.$$

Now, the linear equation can easily be solved by subsequently solving each line, starting from the last line $x_N = \tilde{d}_N$ and working yourself up to the first one.

$$x_i = \begin{cases} \tilde{d}_i & \text{for } i = N \\ \tilde{d}_i - x_{i+1}\tilde{c}_i & \text{for } i < N \end{cases}$$

In the case of the concentration dependent simulations, we do not have a tridiagonal matrix, but instead a matrix of the form:

$$\begin{pmatrix} b_1 & 0 & c_1 & 0 & \dots & 0 & 0 \\ 0 & b_2 & 0 & c_2 & \dots & 0 & 0 \\ a_1 & 0 & b_3 & 0 & \ddots & \vdots & 0 \\ 0 & a_2 & 0 & b_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 & c_{N-2} \\ 0 & 0 & \dots & \ddots & 0 & b_{N-1} & 0 \\ 0 & 0 & 0 & \dots & a_{N-2} & 0 & b_N \end{pmatrix}.$$

Since the Thomas-Algorithm can only be applied to tridiagonal matrices, we need to adapt the algorithm we use a little bit:

We start from a linear equation of the form

$$\begin{pmatrix} b_1 & 0 & c_1 & 0 & \dots & 0 & 0 \\ 0 & b_2 & 0 & c_2 & \dots & 0 & 0 \\ a_1 & 0 & b_3 & 0 & \ddots & \vdots & 0 \\ 0 & a_2 & 0 & b_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 & c_{N-2} \\ 0 & 0 & \dots & \ddots & 0 & b_{N-1} & 0 \\ 0 & 0 & 0 & \dots & a_{N-2} & 0 & b_N \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-1} \\ d_N \end{pmatrix},$$

and bring it into the form

$$\begin{pmatrix} 1 & 0 & \tilde{c}_1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \tilde{c}_2 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 & \tilde{c}_{N-2} \\ 0 & 0 & \dots & \ddots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \tilde{d}_3 \\ \vdots \\ \tilde{d}_{N-1} \\ \tilde{d}_N \end{pmatrix}.$$

Where

$$\tilde{c}_i = \begin{cases} \frac{c_i}{b_i} & \text{for } i = 1 \text{ or } 2 \\ \frac{c_i}{b_i - \tilde{c}_{i-2} a_{i-2}} & \text{for } i > 2, \end{cases}$$

and

$$\tilde{d}_i = \begin{cases} \frac{d_i}{b_i} & \text{for } i = 1 \text{ or } 2 \\ \frac{d_i - \tilde{d}_{i-2} a_{i-2}}{b_i - \tilde{c}_{i-2} a_{i-2}} & \text{for } i > 2. \end{cases}$$

Now, the matrix can again be easily solved by subsequently solving each line, starting from the last line $x_n = \tilde{d}_n$ and working oneself up to the first one.

$$x_i = \begin{cases} \tilde{d}_n & \text{for } i = N \text{ or } N - 1 \\ \tilde{d}_i - x_{i+2} \tilde{c}_i & \text{for } i < N - 1 \end{cases}$$

Like with the Thomas-Algorithm, the number of steps required is of order N .

A.3. Source code for simulations with constant D

```
import numpy as np
from matplotlib import pyplot as py
from scipy.special import erf
import time

StartTime = time.time()

#### Constants ####

D = 10e-19#Diffusion Coefficient

t_max = 100.*3600.#Diffusion Time (in seconds)
l_max = 3.33e-6#Extent of simulated space (in meter)

len_grid = 100#Number of spatial gridpoints
timesteps = 100#Number of temporal gridpoints

#### Coordinates ####
#Create spatial and temporal coordinate system

x = np.linspace(-l_max,l_max,len_grid).tolist()
dx = x[2]-x[1]
t = np.linspace(0,t_max,timesteps).tolist()
dt = t[2]-t[1]

#### Initial Concentration ####
c_init = np.asarray((len_grid/2)*[1]+(len_grid/2)*[0])
```

```

### Analytical Solution ###
def Analytical(x,t):
    return 0.5-0.5*erf(x/(2*np.sqrt(t*D)))

### Thomas-Algorithm ###
#Needed to solve linear system of equations

def ThomasAlgorithm(Matrix,Vector):
    a = Matrix.diagonal(-1)
    b = Matrix.diagonal(0)
    c = Matrix.diagonal(1)
    d = Vector

    cn = []
    dn = []

    cn.append(float(float(c[0])/float(b[0])))
    dn.append(float(float(d[0])/float(b[0])))

    for i in range(1,len(c)):
        c_aux = cn[-1]
        cn.append(c[i]/(b[i]-a[i-1]*c_aux))
        dn.append((d[i]-a[i-1]*dn[-1])/(b[i]-a[i-1]*c_aux))

    dn.append((d[-1]-a[-1]*dn[-1])/(b[-1]-a[-1]*cn[-1]))

    x = []
    x.insert(0,dn[-1])
    for i in range(1,len(dn)):
        x.insert(0,dn[-1-i]-cn[-i]*x[0])

    return x

```

```

#####
### Solve ###
#####

c = c_init.tolist()

#Loop over all time steps
counter = 0
for i in t:
    if i/t_max>counter:#Calculates Progress
        #Remaining Time:
        minutes , seconds = \
            divmod((1-i/t_max)*\
                (time.time()-StartTime)/(i/t_max),60)

        print str(int(100.*i/t_max)) + '_%\t' \
            + str(int(minutes)) + '_min_and_' \
            + str(int(seconds)) + '_s_left'

        counter = counter + 0.01

#Uncomment for live plot
#of calculated concentration profiles
#py.plot(x,c)
#py.plot(x,Analytical(x,i),'k--')
#py.pause(0.0001)
#py.clf()
#py.cla()

```

```

#Create Matrix representation of second derivative
M_d2 = np.diag(np.ones(len(c)-3),1) \
+ np.diag(np.ones(len(c)-3),-1) \
+ np.diag(-2*np.ones(len(c)-2))

#Create Vector representing the boundary conditions
const = np.zeros(len(c)-2)
const[0] = 1#Fixed concentration at left boundary
const[-1] = 0#Fixed concentration at right boundary

#Bring equation into the form M*c = V
#to be solved by Thomas Algorithm

M_d2_CN = 1/dt*np.diag(np.ones(len(c)-2)) \
- 0.5*D*M_d2/(dx**2)

RHS_d2 = np.dot(1/dt*np.diag(np.ones(len(c)-2)) \
+ 0.5*D*M_d2/(dx**2),c[1:-1]) + D*const/(dx**2)

#Solve inner values of c
c = ThomasAlgorithm(M_d2_CN,RHS_d2)

#Add fixed boundary conditions
c.insert(0,1)
c.append(0)

#Information about time required for the simulation
print '100_%'
minutes, seconds = divmod(time.time()-StartTime,60)
print 'The_simulation_actually_took_' \
+ str(int(minutes)) + '_min_and_' \
+ str(int(seconds)) + '_s'

```

```

#Plot the results
fig = py.figure()
ax = fig.add_subplot(211)
py.ylim(-0.05,1.05)
ax.plot(x,c_init,'b',label='Initial')
ax.plot(x,Analytical(x,t_max),'k',label='Analytical')
ax.plot(x,c,'r—',label='Numerical')
ax.xaxis.set_visible(False)
py.ylabel('Concentration of A\n')
py.legend()
py.show()

#Write Results to File
Result = open('Path/Filename.txt','w')
for i in range(len(x)):
    Result.write(str(x[i])+'\t'+str(Analytical(x[i],t_max))\
    +'\t'+str(c[i])+'\t'+str(Difference[i])+'\n')
Result.close()

```


A.4. Source code for simulations with concentration dependent D(c)

```
import numpy as np
from matplotlib import pyplot as py
import time
import re

def nearest(array, value):
    array = np.array(array)
    idxmin = np.argmin(abs(array-value))
    return idxmin

### Constants ###
t_max = 6.*3600.#Diffusion Time (in seconds)
l_max = 10e-6#Extent of simulated space (in meters)

len_grid = 1000#Number of spatial gridpoints
timesteps = 300#Number of temporal gridpoints

### Coordinates ###
#Create spatial and temporal coordinate system

x = np.linspace(-l_max, l_max, len_grid).tolist()
dx = x[2]-x[1]
t = np.linspace(0, t_max, timesteps).tolist()
dt = t[2]-t[1]
```

```

#####
#### Calculations ####
#####

StartTime = time.time()

def D_c(c):#Diffusion Coefficient
    return (0.5 - 198*c*(c-1))*1e-19

    #Instead of an analytical function ,
    #a stepwise function might be better suited
    #for fitting to data

### Initial Concentration ###
c_init = np.asarray((len_grid/2)*[1]+(len_grid/2)*[0])

#Evenly spaced concentration range
#Used later for plotting
c_range = np.linspace(0,1,len_grid)

#####
### PentadiagonalMatrixSolver ###
#####
#Needed to solve linear system of equations

def PentadiagonalMatrixSolver(Matrix,Vector):
    a = Matrix.diagonal(0)
    b = Matrix.diagonal(2)
    c = Matrix.diagonal(-2)
    d = Vector

    bn = []
    dn = []

```

```

bn.append(float(float(b[0])/float(a[0])))
dn.append(float(float(d[0])/float(a[0])))
bn.append(float(float(b[1])/float(a[1])))
dn.append(float(float(d[1])/float(a[1])))

for i in range(2,len(b)):
    b_aux = bn[-2]
    bn.append(b[i]/(a[i]-c[i-2]*b_aux))
    dn.append((d[i]-c[i-2]*dn[-2])/(a[i]-c[i-2]*b_aux))

dn.append((d[-2]-c[-2]*dn[-2])/(a[-2]-c[-2]*bn[-2]))
dn.append((d[-1]-c[-1]*dn[-2])/(a[-1]-c[-1]*bn[-1]))

x = []
x.insert(0,dn[-1])
x.insert(0,dn[-2])
for i in range(2,len(dn)):
    x.insert(0,dn[-1-i]-bn[1-i]*x[1])

return x

#####
### Solve ###
#####
#BC: On both sides derivative == 0, and c = const.

c = c_init.tolist()

#Create Matrix for Solver
M_d1 = np.diag(np.ones(len(c)-3),1) \
+ np.diag(-1*np.ones(len(c)-3),-1)

```

```

counter = 0
#Loop over every Time-Step
for i in t:#Returns progress of the calculation
    if i/t_max>counter:
        #Estimated remaining Time:
        minutes, seconds = divmod((1-i/t_max)\
*(time.time()-StartTime)/(i/t_max),60)

        print str(int(100.*i/t_max)) + '_%\t' \
+ str(int(minutes)) + '_min_and_' \
+ str(int(seconds)) + '_s_left'

        counter = counter + 0.1

#Uncomment for live plot
#of calculated concentration profiles
#py.plot(x,c)
#py.plot(x,c_init)
#py.pause(0.00001)
#py.clf()
#py.cla()

#Translate D(c) to D(x)
D_x = [D_c(jj) for jj in c[1:-1]]

#Combined Matrix M*D*M
M_aux = np.dot(M_d1,np.diag(D_x))
M = np.dot(M_aux,M_d1)

#Boundary Conditions: Fixed Values at the edges
const = np.zeros(len(c)-2)
const[0] = -1#Fixed value at left edge, multiplied by -1
const[-1] = 0#Fixed value at right edge

```

```

#Bring equation into form that can easily be solved
MCN = 1/dt*np.diag(np.ones(len(c)-2)) - M/(8*dx**2)

RHS = np.dot(M/(8*dx**2) \
+ np.diag(np.ones(len(c)-2))/dt, c[1:-1]) \
+ np.dot(M_aux, const)/(4*dx**2)

#Now solve MCN*c = RHS
c = PentadiagonalMatrixSolver(MCN,RHS)
#Add fixed values at the borders
c.insert(0,1)
c.append(0)

c = np.asarray(c)

# Make running (binomially weighed) average to get rid
# of oscillations due to the Crank Nicholson Method
#in combination with sharp initial condition

c_average = [(c[i+1]+2.*c[i]+c[i-1])/4.
for i in range(1,len(c)-1)]
c_average = [1]+c_average+[0]

#Give information about time required for the simulation
print '100_%'
minutes, seconds = divmod(time.time()-StartTime,60)
print 'The_simulation_actually_took_' + str(int(minutes)) \
+ '_min_and_' + str(int(seconds)) + '_s'

```

```

#Plot the results
py.plot(x,c_init , 'k—',linewidth=1,label='Initial ')
py.plot(x,c , 'b',linewidth=0.5)
py.plot(x,c_average , 'r—',linewidth=1.5,label='Numerical ')
py.legend()
py.xlabel('x_[m] ')
py.ylabel('Concentration ')
py.tight_layout()

DDD = [D_c(i) for i in c_range]
py.plot(c_range ,DDD)
py.xlabel('Concentration ')
py.ylabel('D_[$m^2/s$] ')
py.yscale('log ')
py.show()

#Write results to a file
Result = open('Path/Filename.txt', 'w')
for i in range(len(x)):
    Result.write(str(x[i])+ '\t'+str(c_average[i])+ '\n')
Result.close()

```

A.5. Source code for simulations with reactive diffusion

```
import numpy as np
import scipy.special as sp
from scipy.optimize import fsolve
from matplotlib import pyplot as py
import time

StartTime = time.time()

### Constants ###
c_cb = 0.99#    Concentrations at which one phase
c_bc = 0.535#  changes to the other (for BC)
c_ba = 0.485
c_ab = 0.015

D_a = 5e-18#    Diffusion Coefficients
D_b = 6e-18#    in the three different Phases
D_c = 5e-18

t_max = 10*3600.#    in s

dcdt_b = []#    Change in concentration in b-Phase.
#Needs to be defined, because at first there is no b-Phase

c_new_a = []
c_new_b = []
c_new_c = []

len_grid = 1000#    Spacial Resolution.
timesteps = 12000#    Temporal Resolution
```

```

di_ab_aux = 0# Auxiliary Variables, to be used later;
di_bc_aux = 0# determine, when phase boundary moves
dcdt = len_grid*[0]#Auxiliary Variable. Used for Plots
switch = 0#Auxiliary Variable. Used when b-Phase first appears

J_a = 5*[0]# Define Placeholders for Fluxes later
J_b = 5*[0]
J_c = 5*[0]

# Abbreviations
Pi = np.pi
sqrt = np.sqrt
erf = sp.erf

### Coordinates ###

x = np.linspace(-1e-6,1e-6,len_grid).tolist()
dx = x[2]-x[1]
times = np.linspace(0,t_max,timesteps).tolist()
dt = times[2]-times[1]

#Note: if dx and dt are not small enough,
#the simulation will become unstable and may crash.
#Also, dt and dx should be approximately related
#via  $dt = dx**2/(2*D_b)$ .

times = np.arange(0,t_max,dt).tolist()

print 'dx:_' + str(dx) + '_m'
print 'dt:_' + str(dt) + '_s\n'

```



```

#####
### Analytical Solution ###
#####

ca0 = 0
cc0 = 1

def eq(p):#solve eq. 3.27 and 3.28 to determine kbc and kba
    kba, kbc = p

    kab = kba * sqrt(D_b/D_a)
    kcb = kbc * sqrt(D_b/D_c)
    num1 = ca0 - c_ab
    denom1 = kab * sqrt(Pi) * ( 1 - erf(kab) )
    num2 = c_bc - c_ba
    denom2 = kba * sqrt(Pi) * ( erf(kba) - erf(kbc) )
    num3 = c_ba - c_bc
    denom3 = kbc * sqrt(Pi) * ( erf(kba) - erf(kbc) )
    num4 = cc0 - c_cb
    denom4 = kcb * sqrt(Pi) * ( 1 + erf(kcb) )
    eq1 = num1/denom1 * np.exp(-kab**2) + \
    num2/denom2 * np.exp(-kba**2) - c_ba + c_ab
    eq2 = num3/denom3 * np.exp(-kbc**2) + \
    num4/denom4 * np.exp(-kcb**2) - c_cb + c_bc
    return eq1, eq2

kba, kbc = fsolve(eq, (0.2, -0.2))
#Need to make sure that the
#starting values are reasonable

kab = kba*sqrt(D_b/D_a)
kcb = kbc*sqrt(D_b/D_c)

```

```

def c_a(x, t):
    return ca0 + \
        (c_ab-ca0)*(1-erf(x/sqrt(4*D_a*t)))/(1-erf(kab))

def c_b(x, t):
    return ( c_bc*\
        (erf(kba)-erf(x/sqrt(4*D_b*t))))/(erf(kba)-erf(kbc) ) \
        - ( c_ba*\
        (erf(kbc)-erf(x/sqrt(4*D_b*t))))/(erf(kba)-erf(kbc) )

def c_c(x, t):
    return cc0 + \
        (c_cb-cc0)*(1+erf(x/sqrt(4*D_c*t)))/(1+erf(kcb))

def zab(t):
    return kba*sqrt(4*D_b*t)

def zbc(t):
    return kbc*sqrt(4*D_b*t)

def l(t):
    return sqrt(4*D_b*(kba-kbc)**2*t)

def nearest(array, value):
    array = np.array(array)
    idxmin = np.argmin(abs(array-value))
    return idxmin

```

```

#####
#### Thomas-Algorithm ####
#####

def ThomasAlgorithm(Matrix, Vector):
    a = Matrix.diagonal(-1)
    b = Matrix.diagonal(0)
    c = Matrix.diagonal(1)
    d = Vector

    cn = []
    dn = []

    cn.append(float(float(c[0])/float(b[0])))
    dn.append(float(float(d[0])/float(b[0])))

    for i in range(1, len(c)):
        c_aux = cn[-1]
        cn.append(c[i]/(b[i]-a[i-1]*c_aux))
        dn.append((d[i]-a[i-1]*dn[-1])/(b[i]-a[i-1]*c_aux))

    dn.append((d[-1]-a[-1]*dn[-1])/(b[-1]-a[-1]*cn[-1]))

    x = []
    x.insert(0, dn[-1])
    for i in range(1, len(dn)):
        x.insert(0, dn[-1-i]-cn[-i]*x[0])

    return x

```

```

i_cb = len_grid/2 - 1#Initial positions of the phase boundaries
i_bc = i_cb + 1#Identical to the border between a and c phase
i_ba = len_grid/2 - 1
i_ab = i_ba + 1

#Initial Concentration Profile.
c_i = (len_grid/2)*[1] + (len_grid/2)*[0]
c = c_i

#Now Loop over time
counter = 0

for t in times:
    if t/t_max>counter:#Calculates Progress
        #Remaining Time:
        minutes, seconds = divmod((1-t/t_max)*
            (time.time()-StartTime)/(t/t_max),60)
        print str(int(100.*t/t_max)) + '%\t' + \
            str(int(minutes)) + 'min and ' + \
            str(int(seconds)) + 's left'
        counter = counter + 0.01

```

```

## Calculate Diffusion in the three separate Phases ##
##section a:

# Create Matrix for second derivative
M_d2_a = np.diag(np.ones(len(c[i_ab:])-3),1) \
          + np.diag(np.ones(len(c[i_ab:])-3),-1) \
          + np.diag(-2*np.ones(len(c[i_ab:])-2))

#Vector for boundary condition
const_a = np.zeros(len(c[i_ab:])-2)

#Boundary conditions.
const_a[0] = c_ab
const_a[-1] = 0

#Bring everything in the form
#M_d2_a-CN*c_new_a = RHS_d2_a
#according to Crank-Nicholson

M_d2_a_CN = 1/dt*np.diag(np.ones(len(c[i_ab:])-2)) \
- 0.5*D_a*M_d2_a/(dx**2)

RHS_d2_a = np.dot(1/dt*np.diag(np.ones(len(c[i_ab:])-2)) \
+ 0.5*D_a*M_d2_a/(dx**2),c[i_ab+1:-1]) \
+ D_a*const_a/(dx**2)

#Solve Equation with Thomas-Algorithm
c_new_a = ThomasAlgorithm(M_d2_a_CN,RHS_d2_a)

c_new_a.insert(0, c_ab)
c_new_a.append(0)

```

```

##section c:

M_d2_c = np.diag(np.ones(len(c[:i_cb+1])-3),1) \
          + np.diag(np.ones(len(c[:i_cb+1])-3),-1) \
          + np.diag(-2*np.ones(len(c[:i_cb+1])-2))

const_c = np.zeros(len(c[:i_cb+1])-2)

const_c[0] = 1
const_c[-1] = c_cb

M_d2_c_CN = 1/dt*np.diag(np.ones(len(c[:i_cb+1])-2)) \
            - 0.5*D_c*M_d2_c/(dx**2)

RHS_d2_c = \
np.dot(1/dt*np.diag(np.ones(len(c[:i_cb+1])-2)) \
      + 0.5*D_c*M_d2_c/(dx**2),c[1:i_cb+1-1]) \
+ D_c*const_c/(dx**2)

c_new_c = ThomasAlgorithm(M_d2_c_CN,RHS_d2_c)

c_new_c.insert(0,1)
c_new_c.append(c_cb)

```

```

##section b:

if i_ba - i_bc > 2:
#Calculation if there is a b-Phase
    M_d2_b = \
        np.diag(np.ones(len(c[i_bc:i_ba+1])-3),1) \
        + np.diag(np.ones(len(c[i_bc:i_ba+1])-3),-1) \
        + np.diag(-2*np.ones(len(c[i_bc:i_ba+1])-2))

    const_b = np.zeros(len(c[i_bc:i_ba+1])-2)
    const_b[0] = c_bc
    const_b[-1] = c_ba

    M_d2_b_CN = \
        1/dt*np.diag(np.ones(len(c[i_bc:i_ba+1])-2)) \
        - 0.5*D_b*M_d2_b/(dx**2)

    RHS_d2_b = np.dot(1/dt*np.diag(np.ones\
        (len(c[i_bc:i_ba+1])-2)) \
        + 0.5*D_b*M_d2_b/(dx**2),c[i_bc+1:i_ba+1-1]) \
        + D_b*const_b/(dx**2)

    c_new_b = ThomasAlgorithm(M_d2_b_CN, RHS_d2_b)

    c_new_b.insert(0,c_bc)
    c_new_b.append(c_ba)

```

```

elif i_ba-i_bc == 2:
#Newly created b phase that has grown once
    c_new_b = [c_bc ,(c_bc+c_ba)/2. ,c_ba]
elif i_ba-i_bc == 1:
#Newly created b phase, symmetric case
    c_new_b = [c_bc ,c_ba]
elif i_ba == i_bc:
#Newly created b phase, asymmetric case
    c_new_b = [(c_bc+c_ba)/2.]

elif i_ba - i_bc < -1:
    #If a and c overlap, b phase is created.

    i_aux = i_ba#    switch i_cb and i_ba,
    i_ba = i_cb#    so they don't overlap anymore,
    i_cb = i_aux#    and there is space
    i_ab = i_ba + 1#between them
    i_bc = i_cb + 1

    di_aux = di_ab_aux#Swap auxiliary variables
    di_ab_aux = di_bc_aux
    di_bc_aux = di_aux

    c_new_c = c_new_c[:i_cb+1]#cut away overlaped
    c_new_c[-1] = c_cb#area from c_new_a and c_new_c

```



```

        #create b-Phase
        c_new_b = (i_ba-i_bc+1)*[(c_bc+c_ba)/2.]
        if len(c_new_b)>1:
            c_new_b[0] = c_bc
            c_new_b[-1] = c_ba

        c_new_a = c_new_a[-len(c)+len(c_new_c)\
+len(c_new_b):]
        c_new_a[0] = c_ab

        switch = 1
        #Used later to make sure that
        #auxiliary variable is updated correctly.

#### Calculate Flux:####
##J_a

#The only fluxes necessary for simulation
#are J_al and J_ar. The fluxes at the borders.
#Rest only for plotting. Uncomment if needed.

#M_d1_a = np.diag(-1*np.ones(len(c[i_ab:])-3),-1) \
#+ np.diag(np.ones(len(c[i_ab:])-3),1)# Calculate Matrix
#
#                                     for derivative
#const2_a = np.zeros(len(c[i_ab:])-2)#Boundary Condition
#const2_a[-1] = 0
#const2_a[0] = -c_ab

#J_a = -D_a*(np.dot(M_d1_a,c[i_ab+1:-1]) \
#+ const2_a)/(2*dx)#Actually calculate Flux
#J_a = J_a.tolist()
J_al = -(c_new_a[1] - c_new_a[0]) / dx * D_a
J_ar = -(c_new_a[-1] - c_new_a[-2]) / dx * D_a
#J_a.insert(0, J_al)
#J_a.append(J_ar)

```

```

##J_b
#Again, uncomment if needed for plotting, etc.

if i_ba - i_bc > 0:
    #M_d1_b = np.diag(-1*np.ones\
    #(len(c[i_bc:i_ba+1])-3),-1) \
    #+ np.diag(np.ones(len(c[i_bc:i_ba+1])-3),1)
    #const2_b = np.zeros(len(c[i_bc:i_ba+1])-2)
    #const2_b[-1] = c_ba
    #const2_b[0] = -c_bc

    #J_b = -D_b*(np.dot(M_d1_b, c[i_bc+1:i_ba]) \
    #+ const2_b)/(2*dx)
    #J_b = J_b.tolist()
    J_bl = -(c_new_b[1] - c_new_b[0]) / dx * D_b
    J_br = -(c_new_b[-1] - c_new_b[-2]) / dx * D_b
    #J_b.insert(0, J_bl)
    #J_b.append(J_br)

##J_c
#Ditto

#M_d1_c = np.diag(-1*np.ones(len(c[:i_cb+1])-3),-1) \
#+ np.diag(np.ones(len(c[:i_cb+1])-3),1)
#const2_c = np.zeros(len(c[:i_cb+1])-2)
#const2_c[-1] = c_cb
#const2_c[0] = -1

#J_c = -D_c*(np.dot(M_d1_c, c[1:i_cb]) + const2_c)/(2*dx)
#J_c = J_c.tolist()
J_cl = -(c_new_c[1] - c_new_c[0]) / dx * D_c
J_cr = -(c_new_c[-1] - c_new_c[-2]) / dx * D_c
#J_c.insert(0, J_cl)
#J_c.append(J_cr)

```

```

#####
#Calculate new Z: (Position of border between Phases)
#####
#Calculate how much borders should move
if i_ba - i_bc > 0: #If there is a beta phase
    di_ab = dt*(J_br-J_al)/(c_ba-c_ab)
    di_bc = dt*(J_cr-J_bl)/(c_cb-c_bc)
    #Use instead, if Flux-Matrix is uncommented
    #di_ab = dt*(J_b[-1]-J_a[0])/(c_ba-c_ab)
    #di_bc = dt*(J_c[-1]-J_b[0])/(c_cb-c_bc)
elif i_ba-i_bc == 0: #Freshly forming beta phase
    #Flux in this case:
    J_b_lr = -(c_ba - c_bc) / dx * D_b

    di_ab = dt*(J_b_lr-J_al)/(c_ba-c_ab)
    di_bc = dt*(J_cr-J_b_lr)/(c_cb-c_bc)
    #Use instead, if Flux-Matrix is uncommented
    #di_ab = dt*(J_b_lr-J_a[0])/(c_ba-c_ab)
    #di_bc = dt*(J_c[-1]-J_b_lr)/(c_cb-c_bc)
else: #If there is no beta phase
    di_ab = dt*(-J_al)/(c_ba-c_ab)
    di_bc = dt*(J_cr)/(c_cb-c_bc)
    #Use instead, if Flux-Matrix is uncommented
    #di_ab = dt*(-J_a[0])/(c_ba-c_ab)
    #di_bc = dt*(J_c[-1])/(c_cb-c_bc)

```

```

if switch == 0:
    #Add to auxillary variable.
    #If it is bigger then one grid step,
    #z is incremented by one, and
    #auxillary variable is subracted by one.
    di_ab_aux = di_ab_aux + di_ab
    di_bc_aux = di_bc_aux + di_bc
else :
    #If b phase was newly created
    di_ab_aux = di_ab_aux + di_bc
    di_bc_aux = di_bc_aux + di_ab
    switch = 0

#put all together to calculate new c:

c = np.append(c_new_c , c_new_b)
c = np.append(c , c_new_a)

#####
### Plots
#####
#Plots are updated every timestep
py.plot(x , c_i)
py.plot(x , c)

#Border Points
py.plot(x[i_ba] , c[i_ba] , 'ro')
py.plot(x[i_ab] , c[i_ab] , 'ro')
py.plot(x[i_bc] , c[i_bc] , 'ro')
py.plot(x[i_cb] , c[i_cb] , 'ro')

```

```

#Analytical Solution
    opq = 0#Time Headstart
    if t != 0:
        py.plot(x[:nearest(x,zbc(t+opq*dt))],
                c_c(x,t+opq*dt)[:nearest(x,zbc(t+opq*dt))],\
                'k—',lw=2)
        py.plot(x[nearest(x,zbc(t+opq*dt)):\
                nearest(x,zab(t+opq*dt))],\
                c_b(x,t+opq*dt)[nearest(x,zbc(t+opq*dt)):\
                nearest(x,zab(t+opq*dt))], 'k—',lw=2)
        py.plot(x[nearest(x,zab(t+opq*dt)):\
                c_a(x,t+opq*dt)[nearest(x,zab(t+opq*dt))]:],\
                'k—',lw=2)

    py.xlabel('x [m]')
    py.ylabel('Concentration')

#Flux
#    py.plot(x[i_ab:], [k*1e9 for k in J_a])
#    if abs(i_bc-i_ab)>2:
#        py.plot(x[i_bc:i_ba+1], [k*1e9 for k in J_b])
#    py.plot(x[:i_cb+1], [k*1e9 for k in J_c])
py.pause(0.0001)
py.clf()
py.cla()

```

```

#Calculate New Z, Part 2
#Actually move border
if abs(di_ab_aux) > dx:
    i_ab = i_ab + int(di_ab_aux/dx)
    i_ba = i_ba + int(di_ab_aux/dx)
    di_ab_aux = di_ab_aux - np.sign(di_ab_aux)*dx

    #Change concentration to the value
    #of the other side of the border,
    #after the border shifts.
    c[i_ba] = c_ba
    c[i_ab] = c_ab

if abs(di_bc_aux) > dx:
    i_bc = i_bc + int(di_bc_aux/dx)

    i_cb = i_cb + int(di_bc_aux/dx)
    di_bc_aux = di_bc_aux - np.sign(di_bc_aux)*dx

    c[i_bc] = c_bc
    c[i_cb] = c_cb

```

```

print '100_%',
minutes, seconds = divmod(time.time()-StartTime,60)
print 'The_simulation_actually_took_' + str(int(minutes)) + \
'_min_and_' + str(int(seconds)) + '_s'
py.close()
py.plot(x, c_i)
py.plot(x, c)

#py.plot(x[i_ba], c[i_ba], 'ro')#borderpoints specially marked
#py.plot(x[i_ab], c[i_ab], 'ro')
#py.plot(x[i_bc], c[i_bc], 'ro')
#py.plot(x[i_cb], c[i_cb], 'ro')

py.plot(x[:nearest(x, zbc(t))], c_c(x, t)[:nearest(x, zbc(t))],
'k—', lw=3)
py.plot(x[nearest(x, zbc(t)):nearest(x, zab(t))],
c_b(x, t)[nearest(x, zbc(t)):nearest(x, zab(t))], 'k—', lw=3)
py.plot(x[nearest(x, zab(t)):], c_a(x, t)[nearest(x, zab(t)):],
'k—', lw=3)

py.show()

```

A.6. Raw Data for resulting $D_0(c)$

Concentration Range	$D_0(c) \pm \sigma$	Concentration Range	$D_0(c) \pm \sigma$
0.0 - 0.01	$1.20e-05 \pm 7.26e-06$	0.25 - 0.26	$2.10e-06 \pm 4.70e-08$
0.01 - 0.02	$1.15e-05 \pm 5.98e-06$	0.26 - 0.27	$2.10e-06 \pm 4.70e-08$
0.02 - 0.03	$8.56e-06 \pm 4.14e-06$	0.27 - 0.28	$2.10e-06 \pm 4.70e-08$
0.03 - 0.04	$2.80e-06 \pm 1.52e-06$	0.28 - 0.29	$2.10e-06 \pm 4.70e-08$
0.04 - 0.05	$2.55e-06 \pm 1.03e-06$	0.29 - 0.3	$2.10e-06 \pm 4.68e-08$
0.05 - 0.06	$2.46e-06 \pm 7.53e-07$	0.3 - 0.31	$2.10e-06 \pm 4.28e-08$
0.06 - 0.07	$2.23e-06 \pm 2.57e-07$	0.31 - 0.32	$2.10e-06 \pm 4.28e-08$
0.07 - 0.08	$2.19e-06 \pm 1.77e-07$	0.32 - 0.33	$2.10e-06 \pm 4.28e-08$
0.08 - 0.09	$2.19e-06 \pm 1.68e-07$	0.33 - 0.34	$2.10e-06 \pm 4.28e-08$
0.09 - 0.1	$2.19e-06 \pm 1.68e-07$	0.34 - 0.35	$2.10e-06 \pm 4.35e-08$
0.1 - 0.11	$2.19e-06 \pm 1.72e-07$	0.35 - 0.36	$2.08e-06 \pm 7.62e-08$
0.11 - 0.12	$2.19e-06 \pm 1.71e-07$	0.36 - 0.37	$2.08e-06 \pm 7.62e-08$
0.12 - 0.13	$2.18e-06 \pm 1.62e-07$	0.37 - 0.38	$2.08e-06 \pm 7.62e-08$
0.13 - 0.14	$2.18e-06 \pm 1.62e-07$	0.38 - 0.39	$2.08e-06 \pm 7.62e-08$
0.14 - 0.15	$2.18e-06 \pm 1.63e-07$	0.39 - 0.4	$2.08e-06 \pm 7.62e-08$
0.15 - 0.16	$2.18e-06 \pm 1.66e-07$	0.4 - 0.41	$1.09e-06 \pm 8.22e-08$
0.16 - 0.17	$2.18e-06 \pm 1.66e-07$	0.41 - 0.42	$1.09e-06 \pm 8.22e-08$
0.17 - 0.18	$2.18e-06 \pm 1.66e-07$	0.42 - 0.43	$1.09e-06 \pm 8.22e-08$
0.18 - 0.19	$2.18e-06 \pm 1.66e-07$	0.43 - 0.44	$1.09e-06 \pm 8.22e-08$
0.19 - 0.2	$2.18e-06 \pm 1.60e-07$	0.44 - 0.45	$1.09e-06 \pm 8.22e-08$
0.2 - 0.21	$2.12e-06 \pm 5.34e-08$	0.45 - 0.46	$1.09e-06 \pm 8.22e-08$
0.21 - 0.22	$2.12e-06 \pm 5.34e-08$	0.46 - 0.47	$1.09e-06 \pm 8.22e-08$
0.22 - 0.23	$2.12e-06 \pm 5.34e-08$	0.47 - 0.48	$1.09e-06 \pm 8.22e-08$
0.23 - 0.24	$2.12e-06 \pm 5.34e-08$	0.48 - 0.49	$1.09e-06 \pm 8.22e-08$
0.24 - 0.25	$2.12e-06 \pm 5.18e-08$	0.49 - 0.5	$1.09e-06 \pm 8.22e-08$

Table A.1.: Concentration dependent factor $D_0(c)$

Concentration Range	$D_0(c) \pm \sigma$
0.5 - 0.51	$5.98e-07 \pm 1.56e-07$
0.51 - 0.52	$5.98e-07 \pm 1.56e-07$
0.52 - 0.53	$5.98e-07 \pm 1.56e-07$
0.53 - 0.54	$5.98e-07 \pm 1.56e-07$
0.54 - 0.55	$5.98e-07 \pm 1.56e-07$
0.55 - 0.56	$5.98e-07 \pm 1.56e-07$
0.56 - 0.57	$5.98e-07 \pm 1.56e-07$
0.57 - 0.58	$5.98e-07 \pm 1.56e-07$
0.58 - 0.59	$5.98e-07 \pm 1.56e-07$
0.59 - 0.6	$5.97e-07 \pm 1.53e-07$
0.6 - 0.61	$3.50e-07 \pm 7.35e-08$
0.61 - 0.62	$3.50e-07 \pm 7.35e-08$
0.62 - 0.63	$3.50e-07 \pm 7.35e-08$
0.63 - 0.64	$3.50e-07 \pm 7.35e-08$
0.64 - 0.65	$3.50e-07 \pm 7.35e-08$
0.65 - 0.66	$3.50e-07 \pm 7.35e-08$
0.66 - 0.67	$3.50e-07 \pm 7.36e-08$
0.67 - 0.68	$3.50e-07 \pm 7.76e-08$
0.68 - 0.69	$3.50e-07 \pm 7.76e-08$
0.69 - 0.7	$3.50e-07 \pm 7.75e-08$

Table A.2.: Concentration dependent factor $D_0(c)$

Bibliography

- [1] P. J. Alberry and C. W. Haworth. Interdiffusion of Cr, Mo, and W in iron. *Metal Science*, 8(1):407–412, 1974.
- [2] T. R. Corle and G. S. Kino. *Confocal scanning optical microscopy and related imaging systems*. Academic Press, San Diego, 1996.
- [3] J. Crank. *The mathematics of diffusion*. Clarendon Press, Oxford, 2d edition, 1975.
- [4] K. Heinloth. *Die Energiefrage*. Vieweg+Teubner Verlag, Wiesbaden, 2003. DOI: 10.1007/978-3-322-80322-1.
- [5] M. Kajihara. Analysis of kinetics of reactive diffusion in a hypothetical binary system. *Acta Materialia*, 52(5):1193–1200, March 2004.
- [6] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2007. OCLC: ocm86110147.
- [7] R. Lindau, A. Möslang, M. Rieth, M. Klimiankou, E. Materna-Morris, A. Alamo, A.-A. F. Tavassoli, C. Cayron, A.-M. Lancha, P. Fernandez, N. Baluc, R. Schäublin, E. Diegele, G. Filacchioni, J.W. Rensman, B.v.d. Schaaf, E. Lucon, and W. Dietz. Present development status of EUROFER and ODS-EUROFER for application in blanket concepts. *Fusion Engineering and Design*, 75-79:989–996, November 2005.
- [8] O. Madelung, editor. *Dy-Er Fr-Mo*, volume E of *Landolt-Börnstein - Group IV Physical Chemistry*. Springer-Verlag, Berlin/Heidelberg, 1995. DOI: 10.1007/b55397.
- [9] M. Mayer. SIMNRA, a simulation program for the analysis of NRA, RBS and ERDA. AIP, 1999.
- [10] H. Mehrer. *Diffusion in solids fundamentals, methods, materials, diffusion-controlled processes*. Springer, Berlin; New York, 2007. OCLC: 915958622.

- [11] G. Meisl, K. Schmid, O. Encke, T. Höschen, L. Gao, and Ch. Linsmeier. Implantation and erosion of nitrogen in tungsten. *New Journal of Physics*, 16(9):093018, September 2014.
- [12] S. V. Nagender Naidu, editor. *Phase diagrams of binary tungsten alloys*. Number 7 in Monograph series on alloy phase diagrams. Indian Institute of Metals, Calcutta, 1. pr edition, 1991. OCLC: 25862668.
- [13] K. Oura. *Surface science: an introduction ; with 16 tables*. Springer, Berlin, 2003. OCLC: 249200082.
- [14] R. A. Pérez and D. N. Torres. W diffusion in paramagnetic and ferromagnetic α -Fe. *Applied Physics A*, 104(1):329–333, July 2011.
- [15] B. Povh and K. Rith. *Teilchen und Kerne*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [16] K. Sugiyama, J. Roth, V.Kh. Alimov, K. Schmid, M. Balden, S. Elgeti, F. Koch, T. Höschen, M.J. Baldwin, R.P. Doerner, H. Maier, and W. Jacob. Erosion study of FeW binary mixed layer prepared as model system for RAFM steel. *Journal of Nuclear Materials*, 463:272–275, August 2015.
- [17] K. Sugiyama, K. Schmid, and W. Jacob. Sputtering of iron, chromium and tungsten by energetic deuterium ion bombardment. *Nuclear Materials and Energy*, 8:1–7, August 2016.
- [18] S. Takemoto, H. Nitta, Y. Iijima, and Y. Yamazaki. Diffusion of tungsten in α -iron. *Philosophical Magazine*, 87(11):1619–1629, April 2007.
- [19] U. von Toussaint, A. Mutzke, K. Sugiyama, and T. Schwarz-Selinger. Simulation of coupled sputter-diffusion effects. *Physica Scripta*, T167:014023, February 2016.
- [20] A.-S. Walter. Measuring interdiffusion of tungsten and iron with in-situ ion beam analysis. Bachelor’s thesis, Technische Universität München, July 2016.
- [21] K. Wasa, I. Kanno, and H. Kotera, editors. *Handbook of sputter deposition technology: fundamentals and applications for functional thin films, nanomaterials and MEMS*. William Andrew, Waltham, MA, 2nd edition, 2012. OCLC: ocn802326755.
- [22] J. Wesson and D. J. Campbell. *Tokamaks*. Number 149 in Oxford science publications. Oxford University Press, Oxford ; New York, 4th edition, 2011. OCLC: ocn751505944.