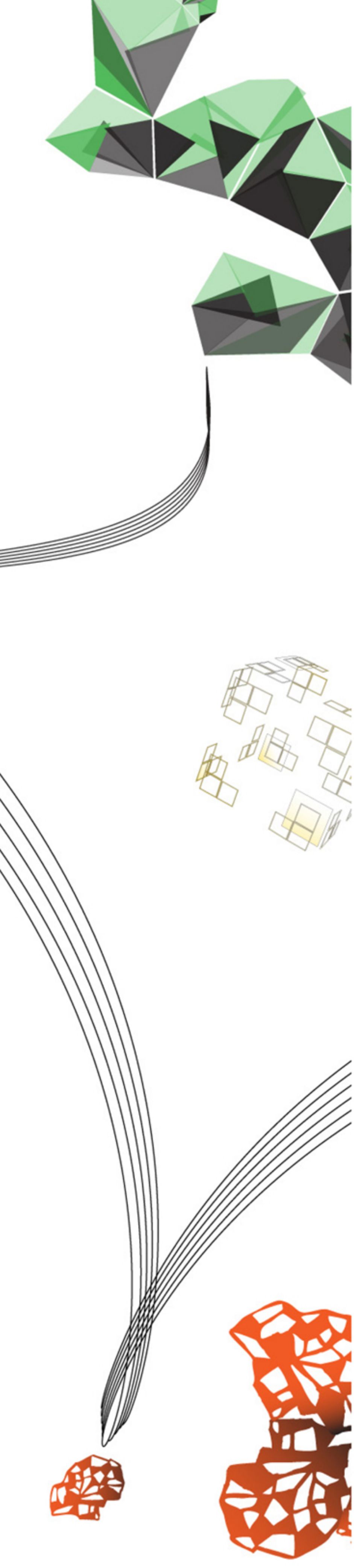


Master Thesis

Extracting the reaction kinetics of the formation of CIGS from *in-situ* X-ray diffraction measurements

Jeroen de Cloet

Faculty of Electrical Engineering, Mahtematics & Computer Science (EEMCS)
Mathematics of Computational Science

Supervisors:
dr. ing. J. Emmelkamp (TNO/Solliance)
prof. dr. ir. J.J.W. van der Vegt (University of Twente)

Assessment committee:
dr. ing. J. Emmelkamp (TNO/Solliance)
prof. dr. ir. J.J.W. van der Vegt (University of Twente)
dr. J.W. Polderman (University of Twente)

28 June 2017

UNIVERSITY OF TWENTE. TNO innovation for life

Abstract

During the formation of the Copper Indium Gallium Selenium (CIGS) layer in thin film solar cells, many chemical reactions occur. To gain more insight in these reactions *in-situ* X-ray diffraction (XRD) measurements are done at TNO/Solliance. We developed a tool to analyze the diffraction patterns that result from the XRD measurements. The diffraction patterns contain certain intensities at different angles, ranging from 20° to 60°. Our analysis works in two steps: decomposition of the diffraction pattern into reflections of individual components and the extraction of the reaction rates and kinetics of several reactions, i.e. the frequency factor and the activation energy in the Arrhenius equation. The analysis in the first step consists of some preliminary data analysis and the decomposition of the diffraction pattern. In the preliminary data analysis we apply a noise filter, find a background approximation, correct for the effects of the thermal expansion and the z-displacement. Finally, we locate the peaks in the diffraction pattern and use this to identify the different components in each measurement. For the decomposition, the Rietveld refinement method is applied to find a good fit of the different components in the diffraction pattern. This fitting is a least squares method where we optimize the parameters step by step with nonlinear optimization methods (multivariable linear least squares method, brute-force method, interior point algorithm, OptQuest NLP algorithm, genetic algorithm). After the diffraction pattern has been analyzed, we use the integral of the fitted components as a measure for the quantity of each component. These integrals are indexed as surface concentrations by using the fact that the total amount of copper, indium and gallium should be constant throughout the measurements series. For the absorption of selenium we use an approximation. With the change in surface concentration we can calculate the reaction rates at different times/temperatures by solving a linear program. The constants in the Arrhenius equation are then fitted with a nonlinear least squares method by comparing the reaction rates at different temperatures and volumetric concentrations.

Keywords: CIGS recrystallization reactions, *in-situ* X-ray diffraction, Rietveld refinement, nonlinear optimization

Contents

List of symbols

Alphabet			**Capitals**			**Greek alphabet**			**Calligraphy letters**		
a	Lattice parameter	[Å]	A	Absorption factor	[−]	α	Lattice parameter	[°]	$\mathcal{A}$	$\mathrm{Re}(F)$	[−]
b	Lattice parameter	[Å]	B	Temperature factor variable	$\left[\text{Å}^2\right]$	β	Lattice parameter	[°]	$\mathcal{B}$	$\mathrm{Im}(F)$	[−]
c	Lattice parameter	[Å]	C	Concentration	[mol m^{-3}]	γ	Lattice parameter	[°]	$\mathcal{C}$	Chebyshev function	[−]
d	Distance between planes	[Å]	$\widetilde{C}$	Adjusted concentration	[mol m^{-3}]	δ	Base shift	[°]	$\mathcal{F}$	Additional variable	[−]
f	(non)linear function	[−]	$D^{‘}$	Difference in concentration	[mol m^{-2}]	ϵ	Bound	[−]	$\mathcal{M}$	Big-M parameter	[−]
f_q	Scattering factor	[−]	E	Ridge	[−]	ζ	FWHM parameter	[−]	$\mathcal{P}$	Preferred orientation factor	[−]
g	(non)linear constraints	[−]	Ea	Activation energy	[J mol^{-1}]	η	FWHM parameter	[−]	$\mathcal{R}$	Gas constant	[J mol^{-1} K^{-1}]
h	Miller index	[−]	F	Structure factor	[−]	θ	Bragg angle	[°]	$\mathcal{T}$	Temperature factor	[−]
i	Miller index	[−]	G	Preferred orientation parameter	[−]	κ_q	Background parameters	$\left[\circ^{-q}\right]$			
j	Reflection index	[−]	H_G	FWHM Gaussian	[−]	λ	Wavelength	[Å]	**Multiple letters**		
k_0	Frequency factor	[mol^{-1} m^3 s^{-1}]	H_L	FWHM Lorentzian	[−]	μ	Penalty factor	[−]	$Diff$	Difference observed and background intensity	[−]
k	Miller index	[−]	I^{bg}	Background intensities	[−]	ν	Site occupancy number	[−]	Int	Integral values	[−]
l	Miller index	[−]	I^{calc}	Calculated intensities	[−]	ξ	Peak shape function	[−]	ITM	Integral-To-Mass	[mol m^{-2}]
l_i	Leaves	[−]	I^{obs}	Observed intensities	[−]	ρ	Optimization parameter	[−]	LP	Lorentz polarization factor	[−]
m	Multiplicity factor	[−]	L	Lorentz factor	[−]	τ	Thermal expansion rate	$\left[K^{-1}\right]$	$Prod$	Stoichiometric coefficients products	[−]
n	Number of measurements	[−]	M	Background polynomial order	[−]	ϕ	Wavelet	[−]	$Reac$	Stoichiometric coefficients reactants	[−]
n_i	Reaction order	[−]	N	Number of phases	[−]	χ	Angular difference for z-shift	[°]	WT	Wavelet transform	[−]
p	Plane	[−]	P	Polarization factor	[−]	ω	Angle between planes	[°]			
q	Index	[−]	Q	Indexing matrix	[−]	Γ	Gamma function	[−]			
r	Radius diffractometer	$[m]$	R	Reaction rate	[mol m^{-3} s^{-1}]	Δ	Trust region	[−]			
s	$\sin(\theta)/\lambda$	[Å^{-1}]	$\widetilde{R}$	Chance in surface conc. due to reactions	[mol m^{-2}]	Θ	Chebyshev variable	[−]			
s_i	Slack variable	[−]	$\widehat{R}$	Corrected reaction rate	[mol m^{-3} s^{-1}]	Φ	Phase	[−]			
t_{step}	Time between two measurements	$[s]$	R_t	Total residual factor	[%]	Ω	Minimization function	[−]			
u_q	Thermal expansion coefficients	[Å K^{-q}]	S	Scaling factor	[−]						
v	Degrees of freedom	[−]	$\widetilde{S}$	Adjusted scaling factor	[−]						
w	Weights	[−]	T	Temperature	[K]						
x	Reciprocal position atom	[−]	U	FWHM parameter	[−]						
y	Reciprocal position atom	[−]	V	FWHM parameter	[−]						
z	Reciprocal position atom	[−]	W	FWHM parameter	[−]						
			X	FWHM parameter	[−]						
			Y	FWHM parameter	[−]						

List of components and their crystal properties

Phase	Crystal structure	Space group	Lattice parameters	Reference
Cu	Cubic	F m -3 m	$a = 3.5819\text{Å}$	[1]
In	Tetragonal	I 4/m m m	$a = 3.2542\text{Å}, c = 4.9542\text{Å}$	[2]
Mo	Cubic	I m -3 m	$a = 3.1468\text{Å}$	[3]
$Cu_{16}In_9$	Hexagonal	P 6_3/m m c	$a = 4.269\text{Å}, c = 5.239\text{Å}$	[4]
$Cu_{11}In_9$	Monoclinic	C 2/m	$a = 12.814\text{Å}, b = 4.354\text{Å}, c = 7.353\text{Å}, \beta = 54.49°$	[5]
$CuIn_2$	Tetragonal	I 4/m c m	$a = 6.645\text{Å}, c = 5.376\text{Å}$	[6]
Cu_9Ga_4	Cubic	P -4 3 m	$a = 8.747\text{Å}$	[7]
Cu_7Ga_4	Cubic	P -4 3 m	$a = 8.7373\text{Å}$	[8]
Cu_3Ga_2	Cubic	P -4 3 m	$a = 8.6949\text{Å}$	[8]
In_4Se_3	Orthorhombic	P 1	$a = 4.1789\text{Å}, b = 12.5711\text{Å}, c = 15.5937\text{Å}$	[9]
$\gamma - \text{InSe}$	Hexagonal	R 3 m	$a = 4.000\text{Å}, c = 24.950\text{Å}$	[10]
In_6Se_7	Monoclinic	P 2_1/m	$a = 9.433\text{Å}, b = 4.064\text{Å}, c = 17.663\text{Å}, \beta = 100.92°$	[11]
$\beta - In_2Se_3$	Hexagonal	R -3 m	$a = 4.050\text{Å}, c = 29.410\text{Å}$	[12]
$\gamma - In_2Se_3$	Hexagonal	P 6_1	$a = 7.110\text{Å}, c = 19.340\text{Å}$	[13]
$CuSe_2$	Orthorhombic	P n n m	$a = 5.103\text{Å}, b = 6.292\text{Å}, c = 3.812\text{Å}$	[14]
CuSe	Hexagonal	P 6_3/m m c	$a = 3.952\text{Å}, b = 3.952\text{Å}, c = 17.244\text{Å}$	[15]
$Cu_{2-x}Se$	Cubic	F -4 3 m	$a = 5.729\text{Å}$	[16]
GaSe	Hexagonal	P -6 m 2	$a = 3.743\text{Å}, c = 15.919\text{Å}$	[17]
Ga_2Se_3	Monoclinic	C c	$a = 6.660\text{Å}, c = 11.650\text{Å}, \gamma = 108.12°$	[18]
$MoSe_2$	Hexagonal	P 6_3/m m c	$a = 3.288\text{Å}, c = 12.900\text{Å}$	[19]
$\alpha - CuInSe_2$	Tetragonal	I -4 2 d	$a = 5.773\text{Å}, c = 11.550\text{Å}$	[20]
$\beta - CuInSe_2$	Cubic	F -4 3 m	$a = 5.7817\text{Å}$	[21]
$\alpha - CuGaSe_2$	Tetragonal	I -4 2 d	$a = 5.614\text{Å}, c = 11.022\text{Å}$	[22]

Preface

With this thesis I conclude my graduation project and thereby my master thesis. Together with my internship I spend the last nine months at TNO/Solliance in Eindhoven. The two (related) projects I did were a challenging, interesting and fun to work on. Using mathematics to solve a physical model of a innovative project, such as solar cells, is a nice application. Although such projects are never completely finished, I am satisfied with the work I have done and the tool that I created. There are a few people I would like to thank for helping and supporting me.

First of all, I thank dr. ing. Jurjen Emmelkamp, my supervisor at TNO, for all the help he gave me on this project. He always had time for my questions and he supported me the whole time. He was convinced I could solve any problem although it was never supposed to be easy. His expertise and reviews really improved this work. Furthermore, we had some interesting discussions about the different views of mathematics and physics.

Next, I express my gratitude to prof. dr. ir. J.J.W. van der Vegt, my supervisor at the University of Twente. He could always make some time for a friendly chat or to help me with my project or other questions. Furthermore, I thank him for the corrections and comments to help improve my thesis. As a teacher he inspired me to get into this field of mathematics and I enjoyed his courses.

Finally, I thank all my family and friends for all the fun and support throughout my study period. My parents were always interested in my project and gave me the freedom to whatever I do. I thank my fellow board members, Kevin, Anke, Sander and Stef, for the best year of my studies. I thank Erwin for the many beers, discussions and projects we did together during our masters. Also, I thanks my colleges for the nice lunch discussions and the games of Hearts.

1 Introduction

At TNO/Solliance one type of thin film solar cells that is developed is the Copper Indium Gallium Selenium (CIGS) solar cell. CIGS solar cells have a soda-lime glass as carrier in our case. Molybdenum is deposited on this glass and serves as back contact of the solar cell. On top of the molybdenum layer copper, indium and gallium are deposited by sputtering or electrodeposition at room temperature. This stack is then put into an oven with selenium vapor (selenization process). The selenium is absorbed by the stack and reacts to form CIS and CGS, which can interdiffuse to form CIGS. Additionally, cadmium sulfide (CdS), which acts as an n-type semiconductor, and a transparent conductive oxide (TCO) are added on top of the CIGS layer to complete the cell.

Previously we developed a simulation model for the recrystallization process of the CIGS layer during the selenization process [23]. In this model over eighty chemical reactions between copper, indium, gallium and selenium are incorporated. Also, the diffusion of components through the CIGS stack and the heat transfer in the oven are taken into account. The diffusion is modeled with Fick's law and the chemical reactions are based on the phase diagrams of the basic components. To calculate the reaction speeds of the individual reactions an Arrhenius equation is used:

$$R = k_0 e^{-\frac{E_a}{\mathcal{R}T}} C_1^{n_1} C_2^{n_2} \ldots . \tag{1}$$

Here, R is the reaction rate [mol m^{-3} s^{-1}], k_0 is the frequency factor [mol$^{1-\sum_i n_i}$ m$^{-3+3\sum_i n_i}$ s^{-1}], E_a is the activation energy [J mol^{-1}], $\mathcal{R}$ is the gas constant [J mol^{-1} K^{-1}], T is the absolute temperature [K], C_i is the volumetric concentration of the i-th reactant [mol m^{-3}] and n_i is the corresponding reaction order [$-$]. The frequency factor and the activation energy are unknown constants for almost all reactions. Some educated guesses were made for these constants based on our knowledge and experience on the recrystallization process.

The goal of this work is to create a tool to extract the reaction kinetics of the individual reactions from *in-situ* X-ray diffraction measurements. This tool analyzes the diffraction pattern and deduces the concentration of every component over time. Relating the changes in concentration with the different reactions that can take place, we can calculate the reaction speed for each reaction and thus with the Arrhenius equation find the reaction kinetics. A similar approach has previously been used by F. Hergert [24] to identify the different chemical reactions that take place during the formation of CIGS. To be able to determine which reactions take place, he evaluated the diffraction patterns with the Rietveld refinement method and created figures of how the phases (relatively) change as function of time. Based on these figures he concluded which reactions take place. We intend to take this a step further and explicitly calculate the reaction rates and deduce the kinetics.

In this work we start with an overview of the principles of X-ray diffraction (Chapter 2). Here Bragg's law, powder diffraction and thin film diffraction measurements will be discussed. We will shown one *in-situ* X-ray diffraction experiment done at TNO/Solliance of a stack of copper, gallium and indium that is heated and then cooled down again (so without selenium) to illustrate the problem with the analysis. Also, we show a measurement that we created by using the simulated concentrations from the earlier described model [23]. A basic understanding of crystal structures, unit cells, crystallographic planes and their symmetry is needed for the concept of X-ray diffraction. Therefore, it is advised to read Appendix A first in case these principles are unknown. After Chapter 2, we will explain how our tool can extract the reaction kinetics. Our program works in two steps: the first step is the analysis of the diffraction pattern and the second step is the quantification of the different components and the extraction of the reaction kinetics. The first step starts with a preliminary data analysis to incorporate several factors that are not directly related to the reflections of the crystallographic planes. Then we use the Rietveld method to calculate the reflections of the individual components and adjust parameters to match the diffraction measurement. In the second step we use the fact that the amount of copper, indium and gallium is constant over time, while selenium is absorbed by the stack, to quantify the components in the fitted diffraction pattern. Next, we use the change in concentration over time to find the reaction rates and thus the reaction kinetics.

Step 1: Analysis of the diffraction pattern
To analyze the diffraction pattern, we first perform a preliminary data analysis (Chapter 3). This consists of noise filtering, background approximation, peak detection, corrections for the thermal expansion and the z-displacement and the shift due to stresses. Whenever measurements are done, there is noise in the signal. We will present two types of filtering methods, averaging and low pass filter, to reduce the noise level in our measurements. There is also a background signal in our measurements. This can be a significant part and cannot be ignored. We will approximate the background signal with a (Chebyshev) polynomial. The peaks in a diffraction pattern are the result of reflection of crystallographic planes in solids. Different components have peaks at different locations and hence we can identify the components by the locations of their peaks. Some peaks can overlap or may be buried in the noise level. Thus an algorithm is needed that automatically detects peaks, also when peaks partially overlap. We will use an algorithm that is based on the wavelet transform as J. Gregoire [25] did, but with some improvements (replacing the user defined parameters by signal dependent parameters). Since the section about the peak detection is quite extensive, we decided to make it into a separate chapter (Chapter 4). The selenization takes place in an oven. The changing temperature has a great influence on the crystal structure of some components and thus on the position of the peaks. Therefore, we adjust the lattice parameters before trying to fit the whole diffraction pattern. Due to the temperature gradient in the setup, the position of the sample might be shifted a little. This effect we call the z-displacement or z-shift. Since the molybdenum peak is very stationary in our temperature range (300–900 K), we use this peak as a reference to calculate the z-shift. Then we correct all the diffraction angles with the trigonometric relations. Finally, the stresses in the precursor stack cause the unit cells to deform. This results in a shift of the location of the peaks in the diffraction pattern. We calculate this shift for each component to match the measured positions with the theoretical positions.

The second part of the first analysis step is the actual matching of the measured diffraction pattern with a calculated diffraction pattern. The Rietveld refinement method is used to create a function that matches the diffraction pattern for each component (Chapter 5). This is a least-squares method that utilizes the theoretical positions and intensities of the individual components and by refining some parameters it tries to create a good fit. We do not optimize all the different parameters in the Rietveld method at once, but one or two at a time. This produces better results, since it is easier to find several minima in multiple subspaces than one specific minimum in a large space. Another advantage is that the least squares problem is in some cases linear. Then there is a (quick) direct solution to the problem. When the problem is nonlinear, we resort to more involved algorithms, such as the interior-point algorithm, the OptQuest NLP algorithm and the genetic algorithm. All the different algorithms we use will be discussed in Chapter 6. Then for each set of parameters we will analyze the optimality and speed of the different algorithms and choose the best performing one for our program (Chapter 7).

Step 2: Quantification and reactions
Once we have found a suitable fit to the diffraction pattern, we can start with the quantification (Chapter 8). The integral over the diffraction angle is a measure for the amount of the individual component. Then we can index the integrated intensities to represent surface concentrations or surface densities. However, the indexing is not straightforward. If the integrated intensity of one component is twice that of another component, then this does not mean that the quantity is also twice as much. We calculate the indexing parameters by using the total amount of copper, indium, gallium and selenium. With the changes in the surface concentration, we can calculate the change in surface concentration caused by each reaction. The this can be converted to reaction rates by dividing by the time between two measurements and the height of the stack. Of course, not all reactions occur at the same time, so we only solve for the reactions that can occur according to the phase diagrams. This involves solving a positive linear system of equations. Finally, we use the volumetric concentration, temperatures and the calculated reactions rates to fit the parameters in the Arrhenius equation (20). This is also a nonlinear least squares minimization problem, which will be solved with a OptQuest NLP algorithm to find the global minimum. For the fitted parameters, the confidence intervals are also calculated.

2 Principles of X-ray diffraction

When a propagating X-ray wave encounters some regularly spaced particles, where the wavelength is in the same order of magnitude as the spacing between the particles, then diffraction occurs [26]. The resulting interference pattern is dependent on the phases of the diffracted waves. X-rays are electromagnetic waves with high energies and short wavelengths, in the order of the atomic spacing for solids. When an X-ray beam hits a solid material, then some of the waves are scattered in all directions by the electrons of the atoms that make up the solid. Not all diffracted waves appear, however, in the interference pattern due to destructive interference (the resultant of out of phase waves). Bragg's law gives a condition for the intensity of the diffracted beam related to the angle between the diffracted beam and the solid.

2.1 Bragg's law

Assume we have a crystal with parallel planes A and B as shown in Figure 1. The spacing between the planes is given by d_{hkl}, where h, k and l are the Miller indices of the plane. A beam of monochromatic coherent X-rays with wavelength λ hits these planes under an angle θ. Two waves are scattered by the atoms P and Q and they leave the crystal under the same angle θ.

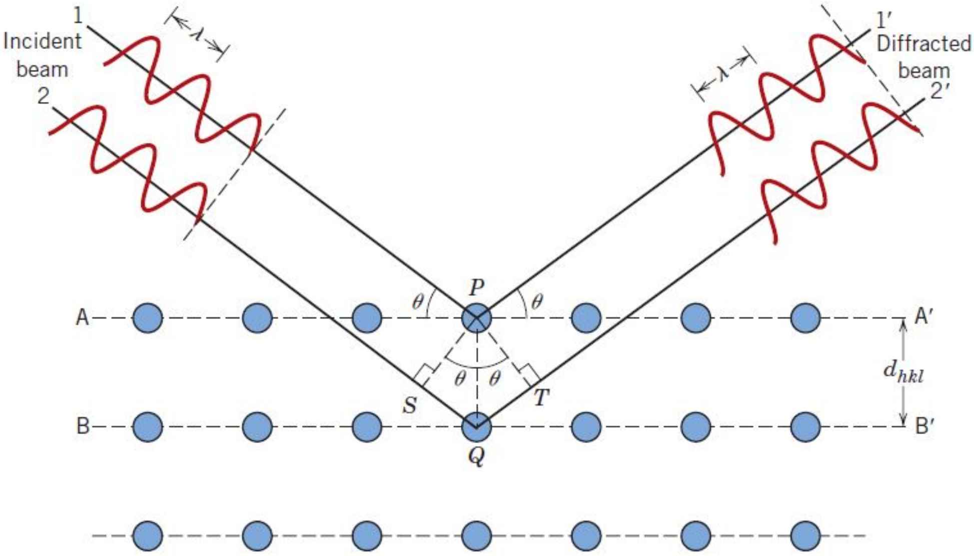


Figure 1: Diffraction of X-ray waves by crystallographic planes. [26]

Constructive interference occurs when the diffracted waves have the same phase. Thus the path difference between waves $1 + 1'$ and $2 + 2'$ should be equal to exactly one (or more) wavelengths. The path difference is $|SQ| + |QT|$, which is equal to $2d_{hkl}\sin(\theta)$. This results in Bragg's law:

$$n\lambda = 2d_{hkl}\sin(\theta), \qquad \text{for } n \in \mathbb{N}^+. \tag{2}$$

Bragg's law relates the diffraction angle to the inter planar spacing d_{hkl}. Then the equation, which relates the distance between planes with the lattice parameters a, b and c and the Miller indices h, k and l, can be used to calculate the orientation of the plane:

$$d_{hkl} = \left(\frac{k^2}{a^2} + \frac{h^2}{b^2} + \frac{l^2}{c^2}\right)^{-1/2}. \tag{3}$$

This equation holds for orthorhombic unit cells ($a \neq b \neq c, \alpha = \beta = \gamma = 90°$). For other unit cells, similar equations can be derived (Equations (22) and (23a)–(23f) in Appendix A.3). Bragg's law only indicates if diffraction occurs for the atoms at the corners of the unit cells. However, atoms at other positions in a unit cell also scatter waves, e.g. in the center (body-centered) or at the center of the faces (face centered). This can result in out of phase waves. For instance, for diffraction in a body-centered cubic crystal structure $h + k + l$ should be an even number and in a face-centered cubic (FCC) crystal structure h, k and l should all be even or odd. Thus Bragg's law is a necessary, but not sufficient condition for diffraction. The crystal structure, therefore, has a large influence on the reflected planes.

2.2 Diffraction pattern

There are different techniques to study the properties of a compound with X-ray diffraction. A common technique is powder diffraction. Here some material is turned into powder and then for each angle the intensity of the diffraction angle is measured. A powder is stress-free and all crystals are randomly orientated. This is useful when you are interested in the structure of a crystal or to identify the different components in a material. We have thin films, which have stresses and a preferred orientation in which a crystal grows. This has a large influence on the resulting diffraction pattern and it is harder to analyze, since the positions of Bragg peaks can be shifted due to stresses and the number of distinct reflections is reduced, because some planes are favored. To be able to find the reaction kinetics, *in-situ* measurements are done. With this method, the diffraction intensities are continuously measured during the selenization process and are separated into different fixed time intervals. This gives a number of measurements over time that we can study. First, we will explain powder diffraction as it is the optimal way for diffraction measurements. Then, we will discuss what changes for thin film measurements and some of the extra difficulties.

2.2.1 Powder diffraction

By using powdered crystals, all reflections of the crystallographic planes should be visible in the diffraction pattern, since it is assumed that the crystals in a powder are randomly orientated. To measure the intensities of the diffracted beam a diffractometer is used, shown in Figure 2.

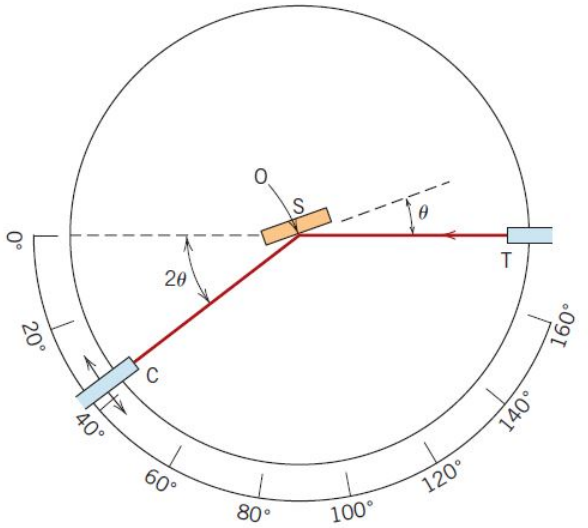


Figure 2: Diffractometer with sample (S), X-ray source (T) and detector (C). [26]

The X-ray source T and detector C move to measure a wide spectrum of angles. For each (discrete) diffraction angle 2θ the number of reflections is measured (in counts). This results in a diffraction pattern. Then the intensities can be checked with a theoretical reconstruction of the diffraction pattern. The relative intensities of the reflections should be similar, since at every angle the reflection of all planes are measured due to the randomness in the powder. Some theoretical powder diffraction patterns with the planes corresponding to their Bragg peaks are shown in Figure 3. These diffraction patterns were calculated with the Rietveld method for different components. The width and the absolute height of the Bragg peaks are not representative for real powder diffraction patterns, but they are just randomly chosen. The relative intensity of the individual peaks is in some way representative. This depends significantly on the setup of the diffractometer.

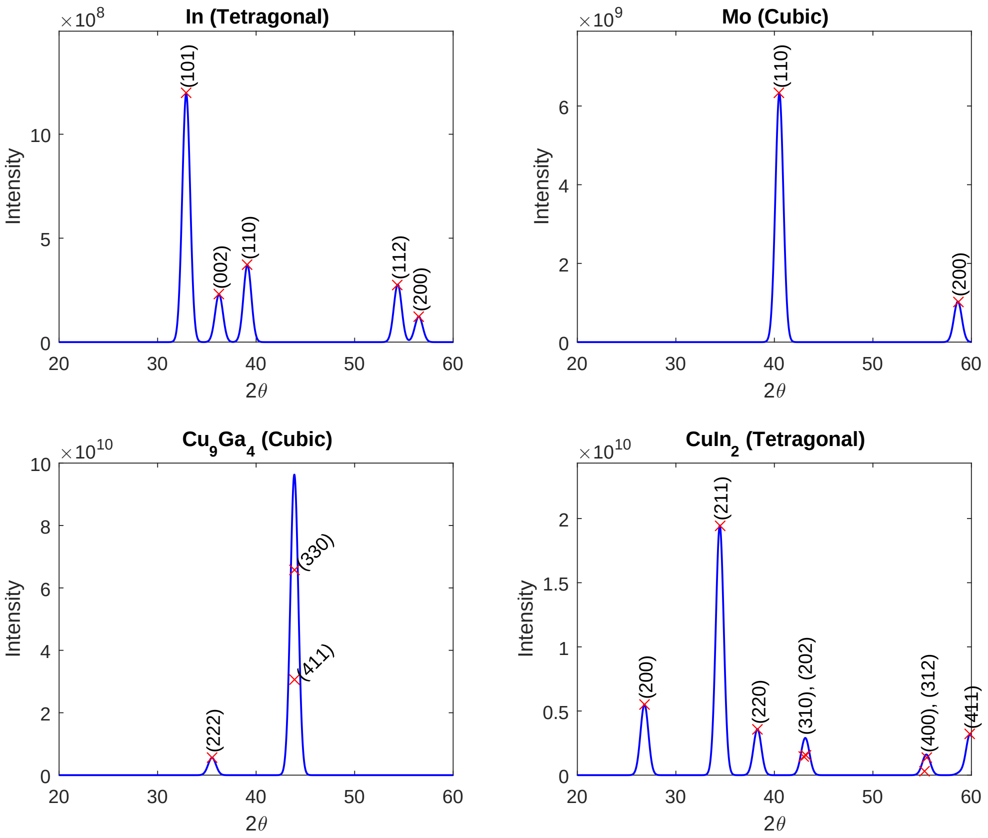


Figure 3: Theoretical powder diffraction patterns for $\mathrm{In}, \mathrm{Mo}, \mathrm{Cu_9Ga_4}$ and $\mathrm{CuIn_2}$.

Figure 3 shows that a single component has multiple peaks and that there is a difference in intensity of the reflections amongst others. These four components and more are present during the formation of CIGS. During the analysis, we will have to match the measured profile with these individual diffraction patterns and determine which measured reflection belongs to which component and crystallographic plane.

2.2.2 Thin film recrystallization

Thin film measurements are more difficult to analyze, because not all reflections of crystallographic planes might be visible in the diffraction pattern. Crystals can grow in a specific orientation that results in a preferred plane (orientation). Therefore, one particular plane is much more common than another plane, which is not the case in a powder. This affects the relative intensities of the reflections. When there is a very strong preferred orientation, then it might be possible that only at one specific angle the reflection of a crystal is visible and that all the other reflections have disappeared. During the selenization process the sample is put into an oven. This means that there will be stresses due to differences in thermal expansion of the different materials. The stress causes a slight change in the unit cell parameters, depending on the direction of the stress. A change in the lattice parameters causes the diffraction angles of the reflections to shift, because the distance between two planes changes. There is also one factor that actually helps us to analyze the *in-situ* measurements: the total amount of material stays the same (except for selenium, which is absorbed by the precursor stack). We assume that the integral over the reflections per component has a linear relationship with the amount of component. Thus, the total value of the integral has to be conserved throughout the measurements. These factors should all be taken into account when doing the analysis on the diffraction pattern.

In one *in-situ* thin film measurement done at TNO/Solliance the precursor stack of copper, indium and gallium on a molybdenum back contact was heated in an oven and then cooled down again. Although there was no selenium added, still much is happening. Every twenty seconds data is collected for a range of 2θ between 20° and 60°. The first measurement is shown in Figure 4.

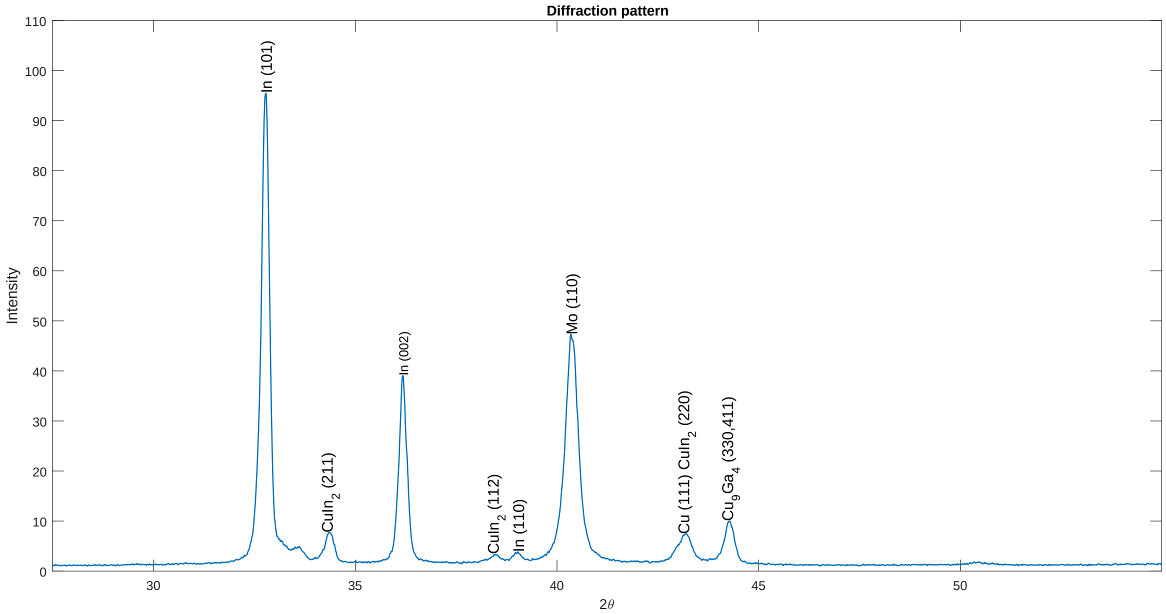


Figure 4: Analyzed diffraction pattern of the first measurement of the series shown in Figure 5.

The entire measurement consists of 91 of these figures. Figure 5 shows the intensity in different colors for the 2θ range for all measurements.

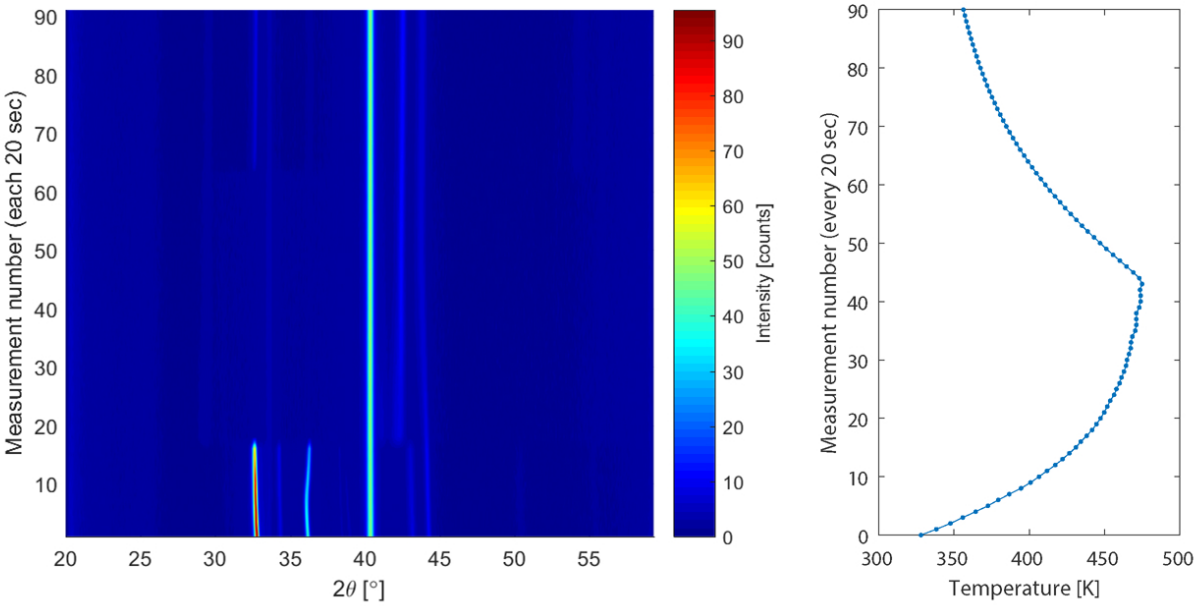


Figure 5: XRD measurements of the heating and cooling down of the precursor stack with corresponding temperatures.

In Figure 5 the melting point of indium (430 K) is clearly visible in the XRD measurements. At the 17th measurement the temperature in the oven is approximately 434 K. Since liquid indium has no crystalline structure, it does not have any reflections and it seems to disappear. Also the effects of thermal expansion are clearly visible. The positions of the reflections shift to different angles as a function of temperature. The diffraction pattern in Figure 4 shows that most components have multiple peaks at different positions. Comparing the measured diffraction pattern to the theoretical powder diffraction patterns already tells us a lot. For instance, indium only has three peaks in the measurement, but five in the powder diffraction pattern (see Figure 3). Therefore, indium must have a (strong) preferred orientation. Furthermore, the peaks from Cu(111) and $CuIn_2(202)$ overlap. Recognizing this difference is another difficulty in the analysis of the diffraction pattern.

Unfortunately, no measurements with selenium have yet been done at TNO/Solliance. To be able to test our tool for a case with selenium, we create an XRD measurement with the simulation model from our previous work [23]. This model simulates the formation of CIGS from a precursor stack that absorbs selenium. One of the outputs is the surface concentration of the components over time. Since the integral in a diffraction pattern represents the amount of that component, we linked the surface concentrations to integrals. Next, we can create a diffraction pattern by setting some random values for the different parameters, e.g. peak shape, preferred orientations factor, etc.. The resulting series of XRD measurements is shown in Figure 6. The maximum temperature is 850 K and we did not add any background signal or noise.

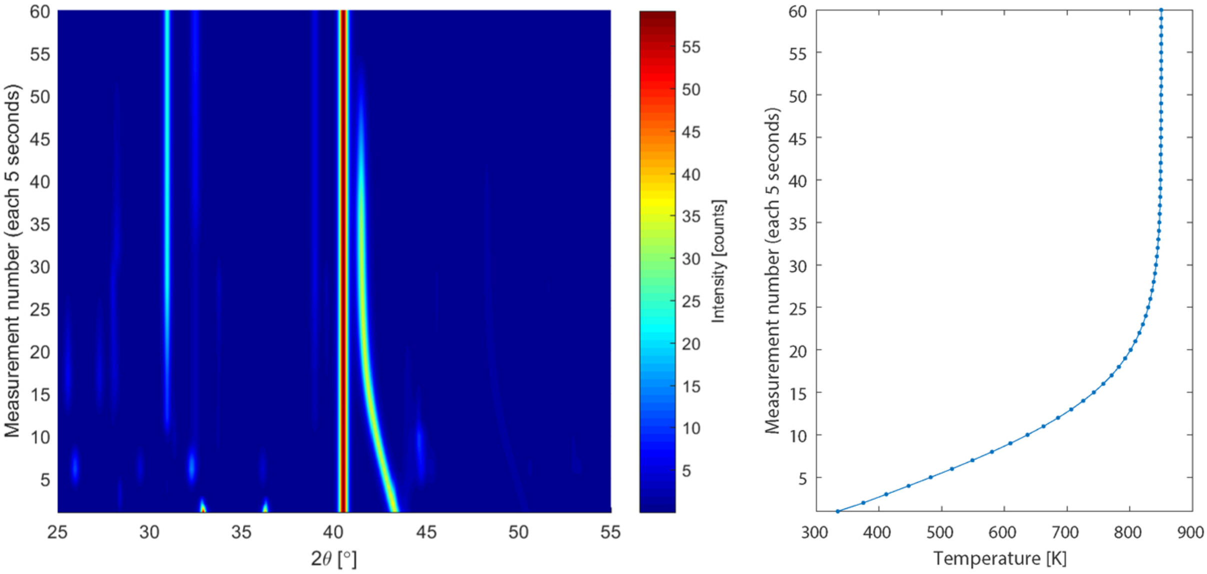


Figure 6: XRD measurements with selenium, created with simulation model, with corresponding temperatures.

When the analysis is done correctly, we should get the same concentrations as we used to create this measurement. In the diffraction pattern in Figure 6 many different peaks can be seen. In a real XRD measurement with selenium, this might not be the case since not all intermediate phases crystallize during the formation of CIGS or their reactions might be too fast to be able to detect the component.

We will analyze each measurement separately. For each measurement we first perform a preliminary data analysis and then fit some parameters in the Rietveld method to decompose the diffraction pattern into reflections of individual components.

3 Preliminary data analysis

The first step in the analysis of the diffraction pattern consists of the application of several corrections. These corrections involve a noise filter, the background approximation, a correction for the expansion of the unit cell due to the increase in temperature, corrections for the incorrect position of the sample and a base shift of the location of the peaks due to stresses in the precursor stack. Also, we use a peak detection algorithm to find the locations of the peaks in the diffraction pattern. This is an extensive subject and thus we put it in a separate chapter.

3.1 Noise filter

Each measurement contains some level of noise. Noise is the addition of random fluctuations to the signal. This is different from the background intensity as it is not random (which will be discussed in the next section). The diffraction pattern consists of low frequencies (slow changing), while the noise contains all frequencies (slow and fast changing). Thus in the frequency domain, we can separate the high frequencies of the noise from the low frequencies of the signal. There are several ways to get reduce noise by filtering the high level frequencies. However, these filters should never be applied before doing the Rietveld refinement [27]. Smoothing causes peak broadening and a decrease resolution. It can also result in the disappearance of small peaks. Therefore, smoothing should only be applied before 'side'-processes, such as approximating the background intensity and finding the peak locations (with caution). We will discuss two smoothing operators: averaging and a low pass filter.

The easiest smoothing operator to reduce the noise level is averaging. This operator takes the weighted average of the current value and its neighbors. This also reduces the absolute intensity of the peaks, so we have to be careful with this. Let u be the input signal (measured values) and y be the output signal (filtered valued). The averaging operator is:

$$y_i = \frac{1}{4}\left(u_{i-1} + 2u_i + u_{i+1}\right),$$

or we can also take the neighbors of the neighbors into account:

$$y_i = \frac{1}{16}\left(u_{i-2} + 4u_{i-1} + 6u_i + 4u_{i+1} + u_{i+2}\right).$$

Here, the subscripts represent discrete angles of the measured diffraction angles.

The second smoothing operator is a low pass filter. As the name suggests, it filters the high frequencies and lets the low frequencies pass. This is done by applying a transfer function h to the input signal. A general filter is a convolution of u and h:

$$y(\theta) = \int h(\theta - \tau)u(\tau)\,\mathrm{d}\tau,$$

where θ and τ are arbitrary variables without a unit. However, when applying this filter, θ corresponds here to the diffraction angle. For discrete values, the integral is a summation and we can easily choose h such that we have the averaging operators that we defined earlier. However, we want to improve the filter and hence use the continuous version for now. A convolution is equivalent to a multiplication in the frequency domain. Thus, by using the Fourier transform (hats) of the signal, we have that:

$$\hat{y}(\omega) = \hat{h}(\omega)\hat{u}(\omega),$$

where the variable ω is the frequency [π rad sample^{-1}]. The 'sample' represents the unit that we work with, so in our case it is the diffraction angle [°]. Now, the transfer function can be defined such that $\hat{h}(\omega) = 1$ for low values of ω and 0 for high values of ω. That would be a low pass filter, since $\hat{y}$ would have all the same values as $\hat{u}$ for all the low values of ω. Similarly, a high pass filter could be created by the converse of the transfer function. Also, a band gap filter is made by setting the transfer function to one for a specific range of frequencies and zero elsewhere. Examples of the transfer function of these three filters are shown in Figure 7.

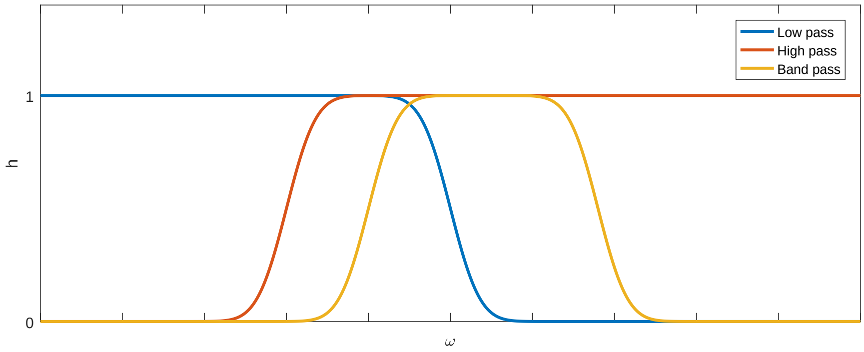


Figure 7: Transfer functions of a low pass, high pass and band pass filter.

We will need a low pass filter. There is a whole class of functions that are low pass filters. They all have a cut-off frequency ω_c. This is the frequency that is the border between the on and off part of the filter. We will use the Butterworth filter with a cut-off frequency of 0.15 π rad sample^{-1} (on a scale from zero to one). The transfer function of the Butterworth filter is:

$$\hat{h}(\omega) = \frac{1}{\prod_{k=1}^{n}(\omega - \omega_k)/\omega_c},$$

where ω_k is the k-th pole of the so-called Butterworth polynomial. The exact definition of this polynomial is not necessary for the discussion here, but it is important to note that there is an order n in the Butterworth polynomial that defines the sharpness of the cut-off. The magnitude of the Butterworth filter, which is a measure for the effectivity of the filter, is shown for different orders n in Figure 8. In our tool we use $n = 3$.

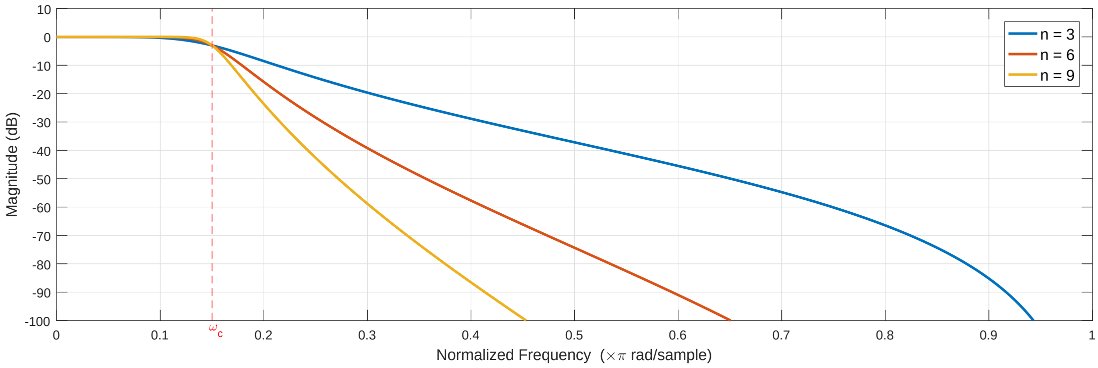


Figure 8: Magnitude of the Butterworth filter for different orders of n with normalized cuf-off frequency 0.15 π rad sample^{-1}.

3.2 Background signal

The background signal consists of many effects that cause scattering in all directions, e.g., inelastic scattering, scattering from air and the sample holder, multiple reflections, etc. It is impossible to find relations to account for all these factors and thus we approximate the background by a general function.

There are many choices for this general function. Here we will discuss two functions: (regular) polynomials and Chebyshev polynomials. Let $I_i^{bg} \equiv I^{bg}(2\theta_i)$ be the background function we want to approximate. The first approximation is an M-th order polynomial:

$$I_i^{bg} = \sum_{q=0}^{M} \kappa_q \cdot (2\theta_i)^q,$$

where κ_q $\left[\circ^{-q}\right]$ are the parameters that have to be optimized given a diffraction pattern. The second approximation is a M-th order Chebyshev polynomial:

$$I_i^{bg} = \sum_{q=0}^{M} \kappa_q \cdot \mathcal{C}_q(\Theta_i),$$

where $\mathcal{C}$ is the Chebyshev function that satisfies the recurrence relation $\mathcal{C}_{q+1}(\Theta_i) = 2\Theta_i \cdot \mathcal{C}_q(\Theta_i) - \mathcal{C}_{q-1}(\Theta_i)$, with $\mathcal{C}_0 = 1$ and $\mathcal{C}_1 = \Theta_i$ and Θ_i is given by the following function:

$$\Theta_i = \frac{2(2\theta_i - 2\theta_{\min})}{2\theta_{\max} - 2\theta_{\min}} - 1.$$

For the optimization of the parameters a simple least squares method does not work, since the peaks would dominate. Thus we have to get rid of the contribution of the peaks to be able to fit the background. One possibility is to apply weights to each value. The minimization problem becomes:

$$\min_{\kappa_q} \sum_i w_i (I_i^{obs} - I_i^{bg})^2,$$

where w_i are the weights at each angle $2\theta_i$. For the weights we give angles with a high intensity (peaks) low weights and angles with low intensities (noise) high weights. The logical choice is to use the inverse measured intensities as weights (or higher orders):

$$w_i = \frac{1}{I_i^{obs}}, \frac{1}{(I_i^{obs})^2}, \ldots.$$

Since the noise is highly oscillatory, it is possible that neighboring points in the noise region get a very different weight (this effect increases for higher orders). To avoid this, we first smoothen the function by using a low pass filter (see Section 3.1). Then the minimization problem can be solved. Another possibility is to remove the contribution of the peaks. With our peak detection algorithm (see Section 4) we can find the peaks in the diffraction pattern. Then we take an arbitrary range around the located peaks and remove those values. This results in a new (smaller) set of observed values $\tilde{I}_i^{obs}$ that can be used to find an approximation for the background. The minimization problem becomes:

$$\min_{\kappa_q} \sum_i (\tilde{I}_i^{obs} - I_i^{bg})^2.$$

Both minimization problems are linear in their parameters and thus can be solved easily by a multivariable linear least squares method (see Section 6.1).

3.3 Thermal expansion

When a metal is heated, it expands. This expansion is caused by the increase in kinetic energy of the atoms. The atoms will move around their position in the crystal lattice more than at a lower temperature and thus increase the average distance between the atoms. In other words, the unit cell lengths increase. A change in unit cell lengths causes the distance between two planes to increase and hence shift the position of the Bragg peak of the corresponding plane. This effect is clearly visible in the example we gave earlier (Figure 5 in Section 2.2.2). The expansion rate is different for each material. Furthermore, the expansion rate is not necessarily the same in every direction. For instance, in a cubic unit cell all lengths are the same and the material will increase by the same amount in every direction (isotropic). However, for a tetragonal unit cell, the base and the height have different lengths and can increase with different rates (anisotropic). As most of the components that we use have a non-cubic unit cell, we assume from now on that the thermal expansion is anisotropic.

The relation between the temperature change and the increase in unit cell lengths a (or b or c) can be assumed to be linear. This relation is given by the formula (4):

$$\frac{\mathrm{d}a}{\mathrm{d}T} = \tau_a a(T), \tag{4}$$

where τ_a is the thermal expansion rate [K^{-1}] in the direction of a [Å]. For small temperature changes (few degrees), the thermal expansion rate can be assumed to be constant. However, we will work a temperature ranges of a few hundreds of degrees, thus we will require the thermal expansion rate also to be a function of temperature. Solving Equation (4) with the thermal expansion rate as function of temperature gives:

$$a(T) = a_0 e^{\int_{T_0}^{T} \tau_a(\bar{T})\,\mathrm{d}\bar{T}},$$

where a_0 and T_0 are the initial unit cell length and temperature, respectively, and $\bar{T}$ is a dummy variable.

For some components, their thermal expansion rate has been measured. For instance, the thermal expansion rate of copper is estimated by $4.93 \cdot 10^{-5} + 4.99 \cdot 10^{-8} T$ for 300 K$\leq T \leq$ 1100K [28]. For indium, which has a tetragonal structure, we will use the direct formula for the expansion of the lattice parameters from [29]:

$$\begin{aligned} a =& u_0 - u_1(T - T_0) + u_2(T - T_0)^2 - u_3(T - T_0)^3 + u_4(T - T_0)^4, \\ c =& u_0 + u_1(T - T_0) - u_2(T - T_0)^2 + u_3(T - T_0)^3 + u_4(T - T_0)^4, \end{aligned}$$

with the coefficients:

	u_0 [Å]	u_1 $[10^{-4}\text{ÅK}^{-1}]$	u_2 $[10^{-6}\text{ÅK}^{-2}]$	u_3 $[10^{-8}\text{ÅK}^{-3}]$	u_4 $[10^{-10}\text{ÅK}^{-4}]$
a	3.249	0.584	3.345	1.191	0
c	4.946	3.546	7.495	5.622	-1.770

and $T_0 = 273.15K$. These formulas hold for $293K \leq T \leq 430K$ (melting point of indium). When we compare the positions of the Bragg peaks of indium using these formulas to correct the locations in the measurements, we see that this approximation is quite good. The locations for the three dominant peaks for different temperatures (which correspond to a measurement) are shown in Figure 9.

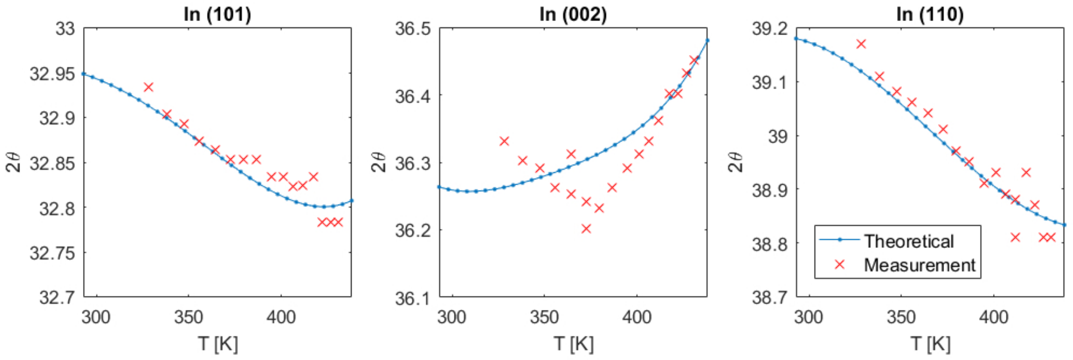


Figure 9: Change in location of the three dominant peaks of indium due to thermal expansion.

Only the reflection of the (002) plane shows a somewhat different behavior. The lattice parameters might have to be adjusted for those measurements where the difference is too large (and results in incorrect fitting). For the other components, the thermal expansion rate is unknown and thus the lattice might also have to be optimized. How the lattice is optimized will be explained later on in Section 7.4.

3.4 z-displacement

Due to temperature changes in the oven, the setup holder of the sample might be displaced by a small fraction. Although this displacement is small, the effects are noticeable in the measurements. We call this effect the z-displacement or z-shift. The sample is then a bit too high or too low and this causes the measured intensities to shift. This shift is nonlinear and has to be corrected for this. The z-shift effect is shown in Figure 10.

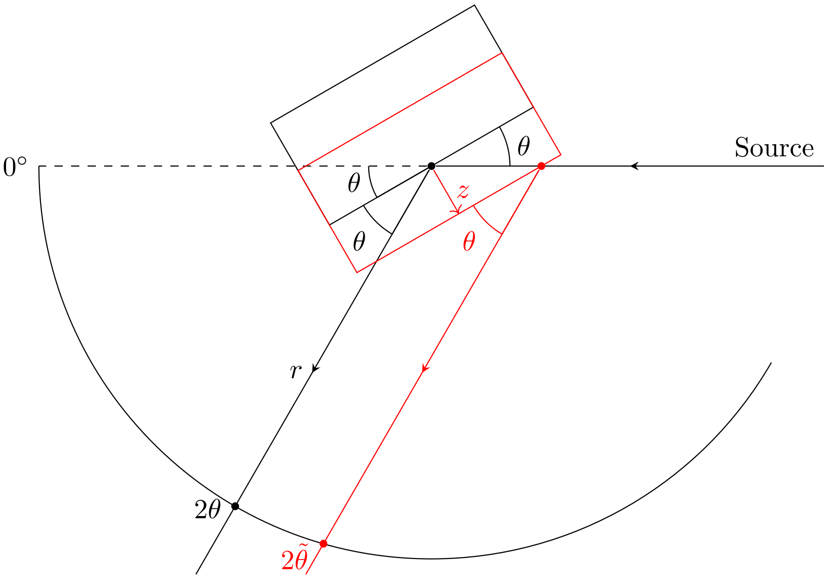


Figure 10: Effect of z-shift on the measured angles 2θ.

In Figure 10 the black sample is at the correct position and the red sample is at the shifted position. The scaling in this figure is not correct, so the effect is exaggerated. The value of the z-shift cannot be measured, but the (110) molybdenum peak acts as a marker in our measurements. The molybdenum peak is very consistent, since it is not influenced by temperature changes. By comparing the theoretical value of the molybdenum peak (40.51°) with the measured value, we can calculate the difference. This difference can be traced back to a certain z-shift. The z-shift can then be used to calculate the difference for all other angles.
Let 2θ be the theoretical value ("correct" value) and $2\tilde{\theta}$ be the measured value ("shifted" value). Furthermore, let r be the radius of the diffractometer and χ be the angular difference between 2θ and $2\tilde{\theta}$. Thus we have that: $2\tilde{\theta} = 2\theta + \chi$. To be able to find some trigonometric relations, we add some guide lines to Figure 10, resulting in Figure 11.

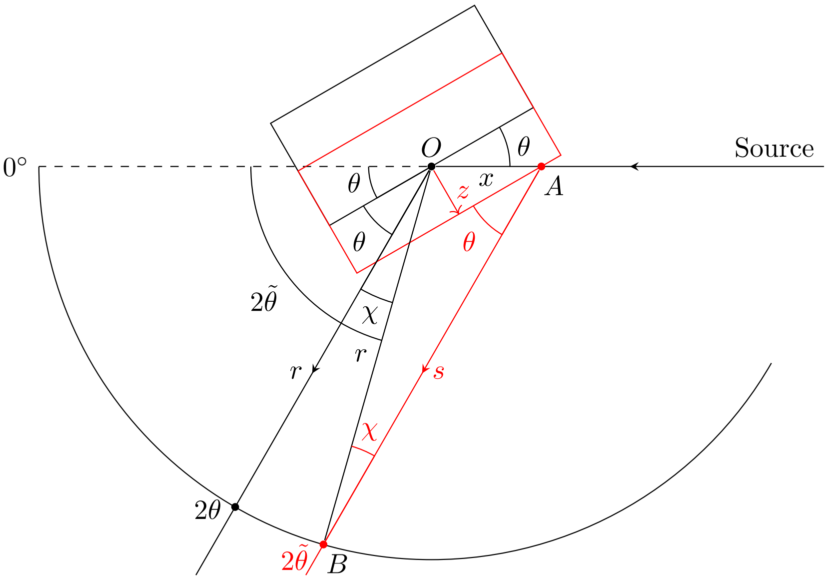


Figure 11: Effect of z-shift on the measured angles 2θ with extra guide lines.

Using the law of sines in the triangle OAB gives us:

$$\frac{x}{\sin(\chi)} = \frac{r}{\sin(2\theta)} = \frac{s}{\sin(\pi - 2\tilde{\theta})}.$$

Furthermore, we know that $\sin(\theta) = \dfrac{z}{x}$. Combining both formulas results in:

$$z = x\sin(\theta) = r\frac{\sin(\chi)\sin(\theta)}{\sin(2\theta)}.$$

We can calculate this since r, χ and θ are known for molybdenum. Now we want to calculate for the entire 2θ range with z known and χ unknown. We use the first equality from the law of sines to find:

$$\begin{aligned}
\frac{x}{\sin(\chi)} &= \frac{r}{\sin(2\theta)},\\
\frac{z}{\sin(\chi)\sin(\theta)} &= \frac{r}{\sin(2\theta)},\\
z\sin(2\theta) &= r\sin(2\tilde{\theta} - 2\theta)\sin(\theta),\\
2z\cos(\theta)\sin(\theta) &= r\sin(2\tilde{\theta} - 2\theta)\sin(\theta),\\
\frac{2z}{r} &= \frac{\sin(2\tilde{\theta} - 2\theta)}{\cos(\theta)}.
\end{aligned}$$

The sine in the numerator can be expanded with the sum-rule. However, this does not help us; this equation cannot be solved explicitly for θ. Therefore, it has to be solved numerically. We will use the Newton-Raphson method for this. Define the following functions:

$$\begin{aligned}
f(\theta) &= \frac{\sin(2\tilde{\theta} - 2\theta)}{\cos(\theta)} - \frac{2z}{r},\\
f'(\theta) &= \frac{-2\cos(2\tilde{\theta} - 2\theta) + \tan(\theta)\sin(2\tilde{\theta} - 2\theta)}{\cos(\theta)}.
\end{aligned}$$

Then solve $f(\theta) = 0$ by the iterative scheme:

$$\theta_{n+1} = \theta_n - \frac{f(\theta_n)}{f'(\theta_n)}.$$

The stopping criterium is when $|\theta_{n+1} - \theta_n|$ is small enough (in the order of machine precision). As initial condition θ_0 we take $\tilde{\theta}$ as we assume that this value is close to the correct value θ, since the z-shift is small.

Applying this method on the first measurement, we obatin that the position of the measured molybdenum peak is 40.35° and the theoretical value is 40.51°, thus $\chi_{\text{Mo}} = -0.16°$. Using $r = 164.7989$ mm, this results in a z-shift of $-0.24(53)$ mm. Next, solving χ for all angles with the Newton-Raphson method gives us the correction for the z-shift. This correction for each angle is shown in Figure 12. Thus to get the corrected diffraction angles, we simply use $2\theta = 2\tilde{\theta} + \chi$.

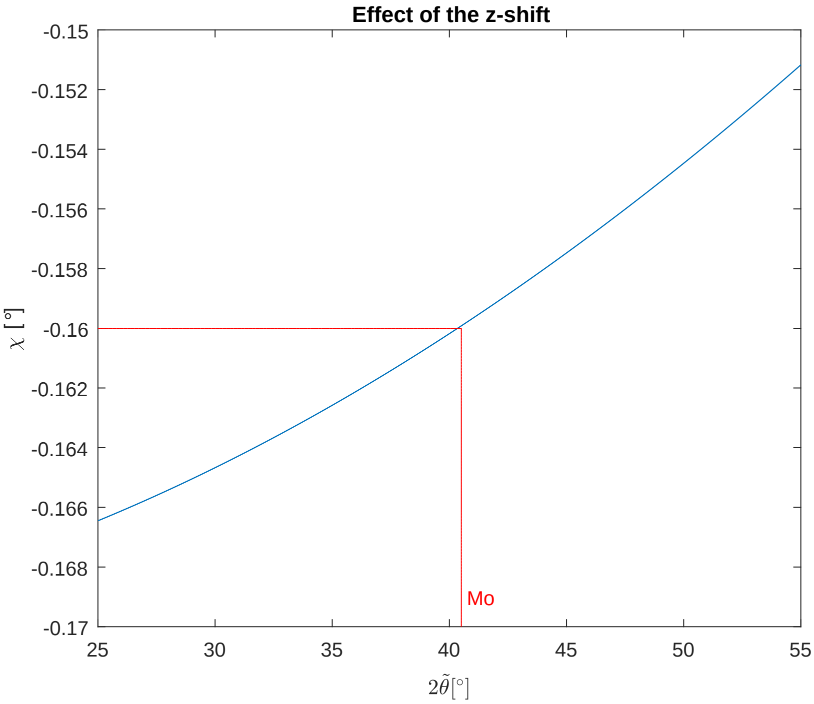


Figure 12: The effect of the z-shift on all measured angles.

3.5 Stresses

Due to the difference in thermal expansion rates of the different components, some components expand more than others. This results in stresses in the precursor stack. A stress causes the lattice parameters to change. It can either shrink, expand or deform, depending on the expansion of the neighboring component. During the recrystallization reactions, the crystal structure can also change and therefore also the amount of stress. This makes it hard to incorporate the stress in the model.

We will therefore not explicitly calculate the stresses, but we will approximate the effect of the stress. This effect is a change in the lattice parameters. This leads to a shift of the peak locations in the diffraction pattern. Therefore, we will define a base shift δ for each component to account for the stress factor. To calculate this, we will compare the locations of the peaks with the theoretical positions.

3.6 Summary of the preliminary data analysis

In this chapter we presented several corrections that have to be applied before decomposing the diffraction pattern. The noise filter will be applied before approximating the diffraction pattern and before using the peak detection algorithm. The filter we use is a low pass filter. To approximate the background intensity we use a Chebyshev polynomial and the weighted least squares method with weights the inverse intensities. For the correction of the unit cell lengths due to the effects of temperature changes, we use some approximations we found in literature. When this is not sufficient, we will optimize the lattice parameters later on in the model. The position of the molybdenum peak is used as reference for the incorrect z-position of the holder. A correction for this is applied to the measured diffraction angle. Finally, we introduced a base shift parameter to correct for the effects of stresses in the precursor stack.

4 Peak detection

Visually peaks are easily detected. However, we do not want to manually select the peaks, but we rather have an algorithm that can detect the peaks. To find all local maxima in a signal (XRD measurement) is not sufficient, since the noise contains many of the detected peaks. Even if we find some bound for the noise level and accept all the local maxima above that bound, it would still not be sufficient. Overlapping peaks are not uncommon and these will not be detected as local maxima. Therefore our algorithm also needs to detect these overlapping peaks. Our algorithm is based on that of Gregoire et al. [25], but with some slight improvements. The algorithm is based on the wavelet transform, which we will explain first.

4.1 Wavelet transform

A function $f(x)$ in a Hilbert space (complete inner product space) can be represented by a linear combination of the (orthonormal) basis functions $\phi_k(x)$:

$$f(x) = \sum_{k=1}^{\infty} c_k \phi_k(x),$$

where the coefficients c_k are given by the projections (inner product) of the function onto the basis functions:

$$c_k = (f(x), \phi_k(x)) = \int_{-\infty}^{\infty} \overline{\phi_k(x)} f(x) \, \mathrm{d}x,$$

where an overbar means the complex conjugate. Since there is no unique set of basis functions for f, different basis functions can be chosen such that the coefficients represent specific aspects of f. For instance, frequencies of a signal can be analyzed with the Fourier transform, which uses $e^{2\pi i k x}$ as a basis function. We are interested in the locations of the peaks in our XRD signal. Peaks are characterized by their height and width. For the detection of peaks the height is not very important as long as it is greater than the noise level (otherwise we cannot identify it as a peak). Thus we are looking for a basis function where we can vary the width and the position of the function. This family of functions is called wavelets:

$$\phi_{a,b}(x) = \frac{1}{\sqrt{a}} \phi \left(\frac{x-b}{a} \right).$$

Here a is the scaling parameter (mainly in the x-direction, thus the width), b the shift parameter and ϕ (without subscripts) is called the mother wavelet. The wavelet transform T is then defined as:

$$WT(a,b) = \int_{-\infty}^{\infty} \overline{\phi_{a,b}(x)} f(x) \, \mathrm{d}x. \tag{5}$$

There are many types of mother wavelets. We need a wavelet that is insensitive to noise (rapid periodical changes) and can detect peaks. As Gregoire et al. [25] suggests, the Mexican hat wavelet is a good choice. This is the negative (normalized) second derivative of a Gaussian function:

$$\phi(x) = \frac{2}{\sqrt{3}\pi^{1/4}} \left(1 - x^2\right) e^{-\frac{1}{2}x^2}.$$

This wavelet gets its name from the shape of this function, shown in Figure 13.

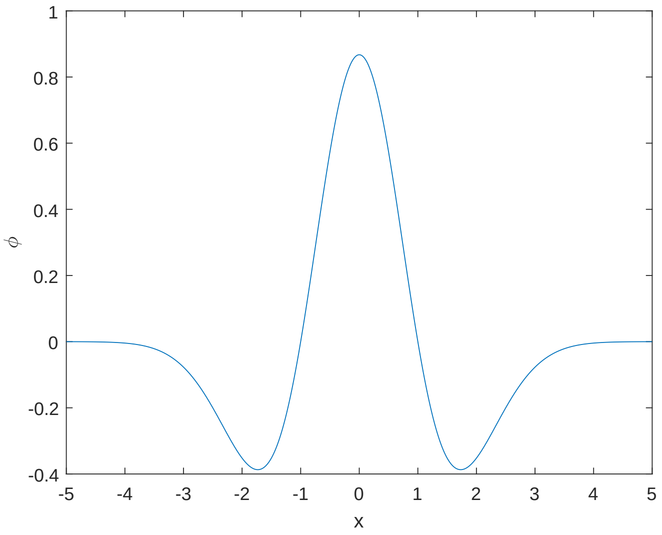


Figure 13: Mexican hat wavelet.

The value of the wavelet transform with the Mexican hat wavelet for a uniform noise level would be close to zero as the positive and negative parts in the integral would cancel each other. Meanwhile at the position of a peak the value of the wavelet transform would become larger than that of the noise level.

4.2 Algorithm

As an example, we use the XRD from one of the measurements performed at TNO/Solliance. The diffraction pattern with the recognized peaks at the first measured time is shown in Figure 14.

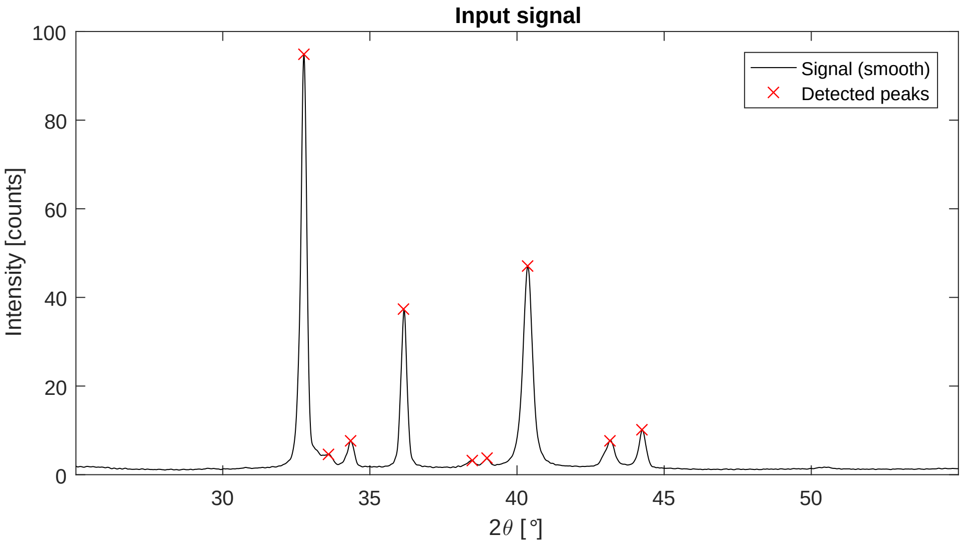


Figure 14: Smoothened signal with the detected peaks.

The wavelet transform does not directly give us the desired peaks. Before and after we use the wavelet transform, some additional processing has to be done. The first thing we do is to approximate the background signal and use this to find a bound for the background intensity ϵ in the diffraction pattern. Any peak that we find must have a higher intensity than this bound. Next, we will apply a smoothing filter to the diffraction pattern to suppress the noise. For this filtered pattern we calculate the wavelet transform for a set of scaling factors and find a bound for the noise in the wavelet transform for each scale (see Figure 15).

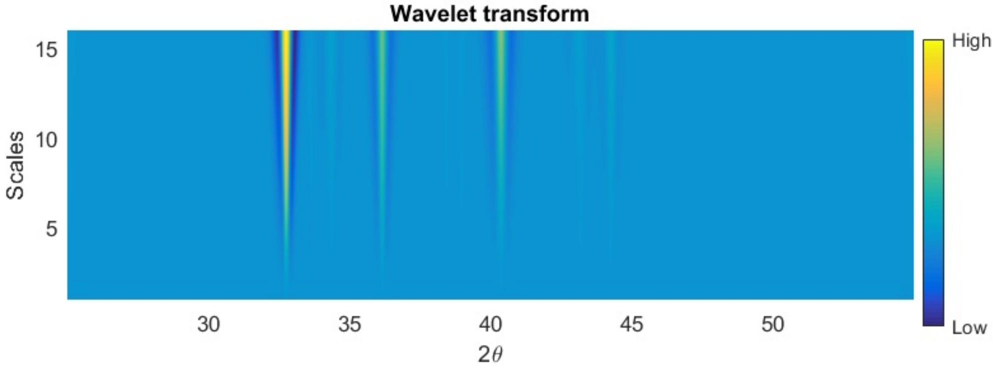


Figure 15: Wavelet transform on different scales.

After that we search in the wavelet transform for local maxima which are larger than the bound for the background intensity ϵ. This gives us a set of local maxima for each scale (variations in a). These nodes will be connected between different scales to form ridges (see Figure 16). Finally we use a criterium to check if a ridge is at the position of a peak.

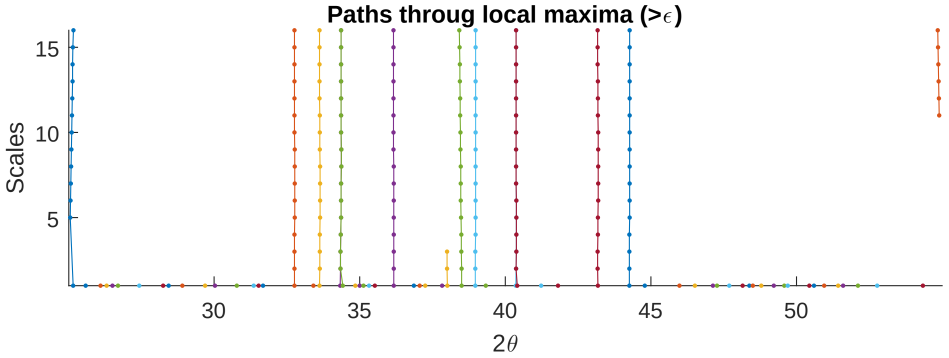


Figure 16: Ridges of local maxima.

4.2.1 Background signal bound

To find a bound for the points in the background signal in the diffraction pattern, we first solve a minimization problem for the approximation of the background intensity. This gives us a Chebyshev polynomial that goes through the non-peak points. However, we need an upper bound for the background signal, not the average values. First, we will show the resulting bound (see Figure 17) and then we explain how to systematically get this.

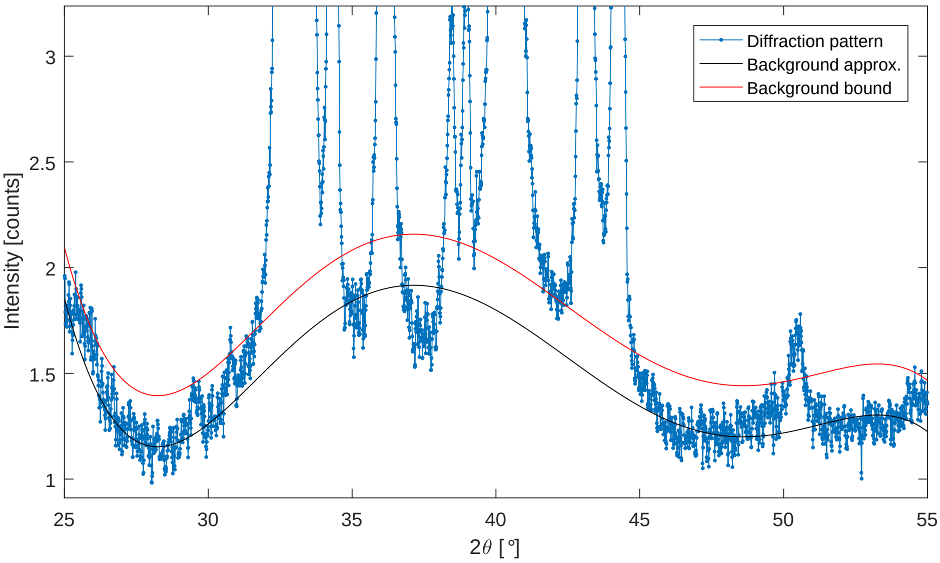


Figure 17: The background approximation and its bound for the first measurement.

We would like to use the largest difference between the background polynomial and the background signal as our bound. However, we have not yet distinguished the points that belong to the background signal from the points belonging to the peaks (that is precisely the problem). Let us define the difference as:

$$Diff_i = |I_i^{obs} - I_i^{bg}|.$$

Upon sorting this vector from low to high values, we can get a general idea what values belong to the background and what values belong to the peaks (see Figure 18).

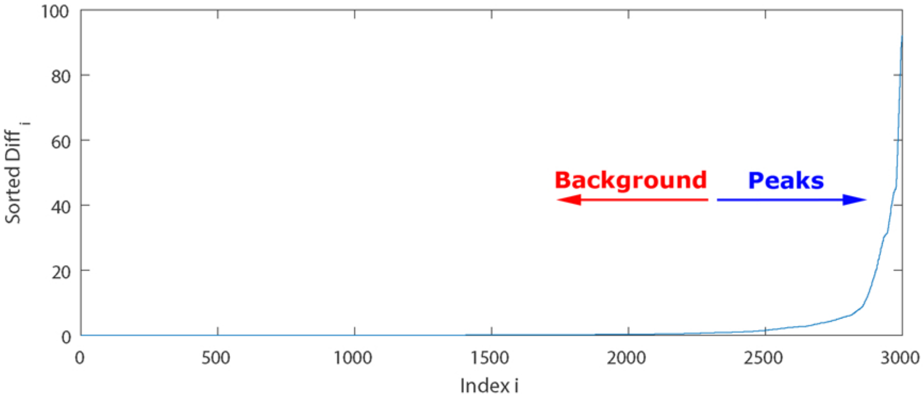


Figure 18: Sorted absolute difference between the background approximation and the intensities in the diffraction pattern.

By looking at this figure, we assume that more than half of the measured points belong to the background region. We predict that approximately 60-70% of the points belong to the background, depending on how many peaks there are in the diffraction pattern. When there are more peaks, the number of points in the background signal are less. The goal is to find an index that separates the background points from the peak points. To find this index, we take the median intensity of the diffraction pattern (so not in $Diff$) and multiply that by a constant value. This constant value has to be chosen such that the resulting index is between 60-70% of the total number of points, depending on the amount of peaks. For this constant, we will use the value 1.15 that we found by analyzing multiple measurements. Then we look op the value of $Diff$ at this index and this becomes our bound ϵ.
For the bound of the wavelet transforms we use the same index, but now apply it to the sorted difference of the wavelet transform and its background approximation. This background approximation is zero everywhere, because the Mexican hat wavelet is an even function. The bound for the wavelet transform on a scale a we call ϵ_{WT_a}.

4.2.2 Continuous wavelet transform

To compute the wavelet transform we will use the `cwt` build-in function of Matlab. This function computes the continuous wavelet transform with a convolution. Starting from the wavelet transform (5) and setting $\tilde{x} = \frac{x}{a}$ and $\hat{b} = \frac{b}{a}$, the equation can be rewritten as a convolution:

$$\begin{aligned} WT(a,b) &= \int_{-\infty}^{\infty} \overline{\phi_{a,b}(x)} f(x)\, \mathrm{d}x, \\ &= \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \overline{\phi\left(\frac{x-b}{a}\right)} f(x)\, \mathrm{d}x, \\ &= \sqrt{a} \int_{-\infty}^{\infty} \overline{\phi(\tilde{x}-\hat{b})} f(\tilde{x})\, \mathrm{d}\tilde{x}, \\ &= \sqrt{a} \int_{-\infty}^{\infty} \overline{\phi(\hat{b}-\tilde{x})} f(\tilde{x})\, \mathrm{d}\tilde{x}, \\ &= \sqrt{a} \left(f * \bar{\phi}\right)(\hat{b}). \end{aligned}$$

Here we used the fact that the Mexican hat wavelet is an even function, namely $\phi(-x) = \phi(x)$. A convolution can be computed easily using the Fast Fourier Transform (FFT). z.

4.2.3 Local maxima ridges

A local maximum is defined as a function value that is larger than its two neighbors (in one dimension). This can be mathematically represented as:

$$f(x_i) \text{ is a local maximum if } (\operatorname{sgn}(f(x_{i-1}) - f(x_i)) - \operatorname{sgn}(f(x_i) - f(x_{i+1}))) < 0,$$

where sgn() is the sign of a value $(-1, 0, +1)$. We compute locations of the local maxima for the signal (LM_S) and the wavelet transform (LM_{WT}) and keep the local maxima that are above the background signal bound ϵ. The points in LM_S and LM_{WT} should be close to each other, since a peak in the signal results in a peak in the wavelet transform. In case there are points in L_T that are not near any point in LM_S, then we remove those points from LM_{WT}.

To be able to detect peaks we group the local maxima on different scales together in ridges. To keep track of all the ridges we use a tree structure in MATLAB, provided by J. Tinevez [30], where each ridge is a branch of the tree. We start on the largest scale and work back towards the smallest scale, because on the largest scale the peaks are stretched the most and thus only the largest peaks appear. For each smaller scale we add the local maxima on that scale to the tree. To do this, we keep track of the leaves of the tree (end points of each branch/ridge) and add each point that is within a range δ from a leaf to that branch. If there is only one point that has to be added, then this point becomes the new leaf of that branch. When there are multiple points that have to be added, then for each such point a new subbranch is initiated. If there are points that are not near any of the leaves, then a new branch is started. This process is visually represented in Figure 19. Here E_i are ridges and l_i are the leaves (end points).

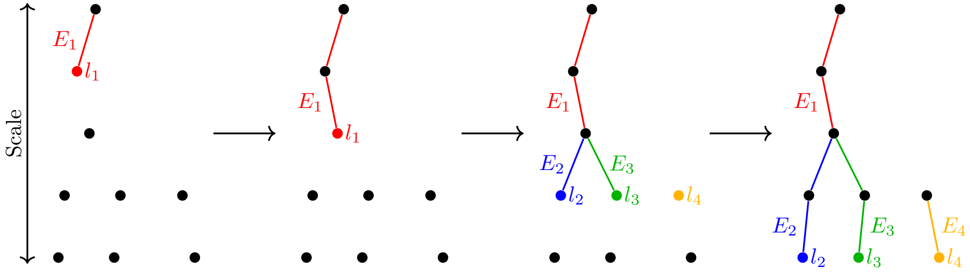


Figure 19: Iterations in the creation of the tree structure.

For each ridge we store the scales, locations and the value of the wavelet transforms of the points. A pseudo– code for this process is given in Appendix D.1.

4.2.4 Criteria for peaks

After we have connected all points to form a tree. The leaves of the tree (local maxima on the lowest scale of the wavelet transform) are all possible locations for peaks. To determine if a leaf represents a location of a peak, we will look at the following two values:

- the sum of the values of the wavelet transform of points in the branch of the leaf.
- the sum of the values of the wavelet transform of points in the roots of the branch.

We will illustrate this with an example. As example we take the last tree from Figure 19, which we represented again in Figure 20.

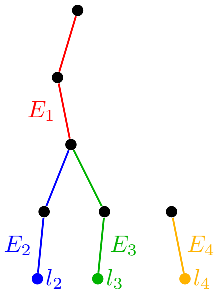


Figure 20: The last tree from Figure 19.

The possible locations for trees are l_2, l_3 and l_4. These leaves represent a value of the diffraction angle 2θ at the lowest scale ($a = 1$). The ridges E_i are sets of points $\{v_1, v_2, \dots\}$, that represent a location and a scale, $v_1 = \{a_1, b_1\}$. When we talk about the sum of the values of the wavelet transform of a branch, this is $WT(E_i) = \sum_{v \in E_i} WT(v)$. A root of a branch we will represent by the $\prec$ symbol. Thus in Figure 20, we have $E_1 \prec E_2$ and $E_1 \prec E_3$. Furthermore, for the set of all roots until the bottom is reached (in our case the largest scale; our tree is upside down), we use the notation: $E_j \prec\!\prec E_i$. The second value describe above becomes: $WT(\prec E_i) = \sum_{E_j \prec\prec E_i} WT(E_j)$. Since the Bragg peaks have a high value of the wavelet transform at the largest scale and the noise has a low value, we can distinguish them. This means that for a peak $WT(E_i) + WT(\prec E_i) \gg 0$ and for noise $WT(E_i) + WT(\prec E_i) \approx 0$. Thus we can set a bound ϵ_{Peak} to determine if a leaf presents a peak:

$$\begin{aligned} l_i \text{ is a peak} &\iff WT(E_i) + WT(\prec E_i) \geq \epsilon_{Peak}, \\ l_i \text{ is a not peak} &\iff WT(E_i) + WT(\prec E_i) < \epsilon_{Peak}. \end{aligned}$$

The only thing left to do is to set an appropriate bound ϵ_{Peak}. Gregoire et al. [25] used some user-defined bounds, but we will use a bound that is dependent on the diffraction pattern. For this bound we take the average of the measured intensities: $\epsilon_{Peak} = mean(I^{obs})$. This bound is a little bit higher than the background intensity and thus always (much) greater than zero. This means that it will filter out the false peaks from the noise. For the smaller peaks in the diffraction pattern, we assume that the wavelet transform on the largest scale has a greater value than the average of the measured intensities. As the intensity of a small peak is approximately equal to the average of all the measured intensities and the maximum scale we use is 16, this is a reasonable assumption.

5 Rietveld refinement

After the preliminary data analysis, we can start with the decomposition of the diffraction pattern. We use the Rietveld refinement method in this study. This method approximates the reflections of different components by using the crystal structure. To approximate the diffraction pattern Rietveld [31] used a function that accounts for most of the physical properties of the crystal and the X-rays. Then a (nonlinear) least squares optimization method can be used to fit the parameters in the approximation. Let I_i^{obs} be the measured intensity at diffraction angle $2\theta_i$ and let I_i^{calc} be the calculated profile at diffraction angle $2\theta_i$. Then the goal is to minimize:

$$\min \sum_i (I_i^{obs} - I_i^{calc} - I_i^{bg})^2, \tag{6}$$

where I_i^{bg} is the background intensity. The approximation for the diffraction pattern is given by the following function:

$$I_i^{calc} = \sum_\Phi S_\Phi \sum_{j=1}^{N_\Phi} m_j |F_j|^2 LP_j \mathcal{P}_j \xi_{\Phi ij} A_j, \tag{7}$$

where Φ represents a phase and N_Φ is the total amount of reflections (planes) for that phase and j represents a single plane. The different factors are:

S_Φ	:	Scaling factor of phase Φ	LP_j	:	Lorentz polarization factor
m_j	:	Multiplicity of the reflection j	$\xi_{\Phi ij}$	:	Peak shape function
F_j	:	Structure factor	$\mathcal{P}_j$	:	Preferred orientation factor

The structure factor F is the most significant for the intensity. This factor accounts for the scattering of the X-rays by the atoms in the crystal structure for each plane. The scale factor is just a general scaling factor that also includes some scaling constants of other factors. The multiplicity factor is the number of planes with the same diffraction angle. When several planes have the same diffraction angle, only the reflection of one plane is calculated and then multiplied with the number of planes with the same diffraction angle. The Lorentz and polarization factor are factors that account for the geometry of the diffraction and the partial polarization of the scattered X-rays. The preferred orientation factor can change the relative intensities of the reflections, whenever a preferred plane is present, otherwise this factor is just one. The shape of the Bragg peaks is represented by the peak shape function $\xi_{\Phi ij}$. This function gives the intensity for each plane j of phase Φ at the angle $2\theta_i$. The asymmetry factor is used to introduce asymmetry in the peak shapes. All these factors will now be discussed in more detail.

5.1 Multiplicity factor

The multiplicity factor of a peak is the number of planes with the same distance between them. Since the distance between the planes directly corresponds with the diffraction angle, multiple planes can have the same diffraction angle. This causes the intensity to multiply with the same amount. Looking at the equations for the distance between planes (Equations (23a–23f) in Appendix A.3), we can find out which planes have the same distance between them. For instance, the equation for the distance between planes in a tetragonal crystal is:

$$\frac{1}{d^2} = \frac{h^2 + k^2}{a^2} + \frac{l^2}{c^2}$$

Here we can immediately see that negative Miller indices result in the same distance as their corresponding positive values. We can also interchange the h and k values. With this equation we can check for every plane how many similar planes there are. For a tetragonal crystal, there are seven families of planes that are similar. These families and their multiplicity are given in Table 1.

Table 1: Families of planes and their multiplicity factor in a tetragonal crystal

Multiplicity	Similar planes
16	$hkl, \bar{h}kl, h\bar{k}l, hk\bar{l}, \bar{h}\bar{k}l, \bar{h}k\bar{l}, h\bar{k}\bar{l}, \bar{h}\bar{k}\bar{l}$ $khl, \bar{k}hl, k\bar{h}l, kh\bar{l}, \bar{k}\bar{h}l, \bar{k}h\bar{l}, k\bar{h}\bar{l}, \bar{k}\bar{h}\bar{l}$
8	$hhl, \bar{h}hl, h\bar{h}l, \bar{h}\bar{h}l, hh\bar{l}, \bar{h}h\bar{l}, h\bar{h}\bar{l}, \bar{h}\bar{h}\bar{l}$
8	$h0l, \bar{h}0l, h0\bar{l}, \bar{h}0\bar{l}, 0kl, 0\bar{k}l, 0k\bar{l}, 0\bar{k}\bar{l}$
8	$hk0, \bar{h}k0, h\bar{k}0, \bar{h}\bar{k}0, kh0, \bar{k}h0, k\bar{h}0, \bar{k}\bar{h}0$
4	$hh0, \bar{h}h0, h\bar{h}0, \bar{h}\bar{h}0$
4	$h00, \bar{h}00, 0k0, 0\bar{k}0$
2	$00l, 00\bar{l}$

Upon checking for all planes to which family they belong, we can calculate their multiplicity. Next, we take one representative of each family to create a set of planes that we use for the Rietveld refinement. The set of planes for a cubic structure has only a few planes, while the set of planes for a monoclinic structure has many planes. Having a large set of planes makes the algorithms computationally more expensive since the reflections of all planes (in the set) have to be computed. The tables for the families of planes and their multiplicity for all crystal structures are given in Appendix B.1.

5.2 Structure factor

The structure factor accounts for reflections due to the scattering of X-rays by the atoms in the crystal structure. Therefore, the crystal structure has a large influence on the structure factor. The positions of the atoms in the unit cell determine the magnitude of the reflection of each plane. Furthermore, the kind of atom (copper, indium, gallium, selenium or molybdenum) affects the scattering rate of the X-rays. Also, the temperature influences the vibration of the atoms around their position. All these effects are taken into account in the structure factor [27]. The equation for the structure factor is:

$$F_j = \sum_{q=1}^{N_{atoms}} \nu_q \mathcal{T}_q(s) f_q(s) e^{2\pi \underline{i}(hx_q + ky_q + lz_q)}, \tag{8}$$

where $\underline{i}$ is the imaginary unit and the sum is taken over all the atoms in the unit cell. The different factors we explained earlier are:

$$\begin{aligned} s &: \ \sin\theta_k/\lambda \ [\mathrm{m}^{-1}], \\ \nu_q &: \ \text{Site occupation number } [-], \\ \mathcal{T}_q &: \ \text{Temperature factor } [-], \\ f_q &: \ \text{Scattering factor } [-], \\ h,k,l &: \ \text{Miller indices } [-], \\ x_q, y_q, z_q &: \ \text{Fractional coordinates of the } r\text{th atom } [-], \end{aligned}$$

where $\lambda = 1.54060$Å is the wavelength that is used for the measurements (Cu Kα). The individual factors will be explained later. Let us now focus on the last part of the structure factor. This part implies that the structure factor is an complex value. However, the structure factor will be used in absolute value and squared (as in (7)). This results in a real value, since we know that:

$$|F_j|^2 = (\mathrm{Re}(F_j))^2 + (\mathrm{Im}(F_j))^2.$$

By applying Euler's formula, we can rewrite $|F_j|^2$ as:

$$|F_j|^2 = \mathcal{A}_j^2 + \mathcal{B}_j^2,$$

where

$$\mathcal{A}_j = \sum_{q=1}^{N_{atoms}} \nu_q \mathcal{T}_q(s) f_q(s) \cos(2\pi(hx_q + ky_q + lz_q)),$$

$$\mathcal{B}_j = \sum_{q=1}^{N_{atoms}} \nu_q \mathcal{T}_q(s) f_q(s) \sin(2\pi(hx_q + ky_q + lz_q)).$$

Due to symmetry in the crystal structures, some of the sines and cosines may cancel each other out. For each space group, the sin/cos factors are given in Tables A1.4.3.1–A1.4.3.7 in the International Table for Crystallography, Volume B [32]. We will use these tables to calculate this part of the structure factor. The other three parts of the structure factor (site occupation number, scattering factor and temperature factor) will be discussed now.

5.2.1 Site occupation number

The asymmetric part of the unit cell is repeated via symmetry operations. The site occupation number is defined as the fraction of occupied symmetrical positions. If all of the symmetrical positions of one asymmetric atom are filled, then the site occupation number is equal to one. If not all positions are filled, then the number is less than one. This happens when one position can host atoms of different components. For instance in the crystal structure of Cu_9Ga_4, one position has 0.75 copper and 0.25 gallium. This means that in one out of four unit cells this position is occupied by a gallium atom and the other three times copper is present at that position.

5.2.2 Scattering factor

For X-rays, the magnitude of scattering is dependent on the number of electrons of an atom [27]. More precisely, the scattering factor is dependent on the radial distribution of the electrons around the nucleus. Also, it is a function of the Bragg angle. This is a complicated function, so for computational purposes we use an approximation:

$$f_q(s) = \sum_{k=1}^{4} a_{q,k} e^{-b_{q,k}s^2} + c_q \quad \text{with } s = \frac{\sin(\theta_j)}{\lambda} \quad [\mathrm{m}^{-1}],$$

where $a_{q,k}$ $[-]$,$b_{q,k}$ $[\mathrm{m}^2]$ and c_q $[-]$ are constants dependent on the atom type. These constants are presented in Table 6.1.1.4 in the International Table for Crystallography, Volume C [33]. This function is valid for $0 < s < 2.0\text{Å}^{-1}$. Since we use a wavelength $\lambda = 1.54060\text{Å}$ and a maximum angle of $2\theta = 60°$, this condition is always satisfied.

5.2.3 Temperature factor

Atoms continuously vibrate around their equilibrium position due to temperature. Therefore, this factor is also called the atomic displacement factor. To account for the oscillations, the temperature factor is incorporated in the structure factor, since the vibrations are different for each atom and each position. The simplest form to account for the displacements is by the following function:

$$\mathcal{T}_q(s) = e^{-B_q s^2}, \quad \text{with } s = \frac{\sin(\theta_j)}{\lambda},$$

where B_q is the displacement parameter $[\mathrm{m}^2]$ of the qth atom. For the intermetallic compounds we work with, B typically ranges from 0.5Å^2 to 1Å^2 [27]. For (in)organic compounds, B is usually considerably higher ($1\text{Å}^2 - 10\text{Å}^2$). In Figure 21 can be seen that the effect of the temperature factor is more noticeable for high Bragg angles. However, we work with small angles and the displacement factor is also small, thus the effect of the temperature factor is negligible (approximately equal to one) in our case.

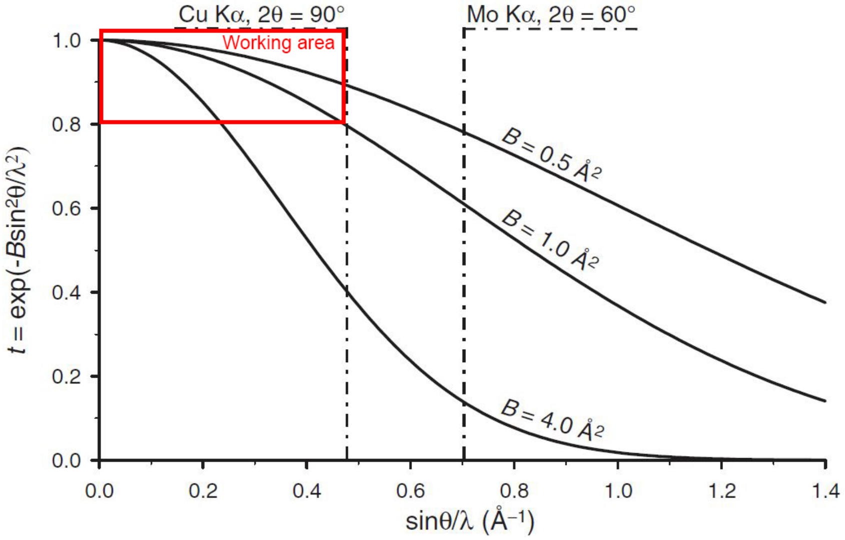


Figure 21: Temperature factor as function of $s = \sin(\theta)/\lambda$ for different values of B : $0.5, 1.0$ and 4.0Å^2. The two vertical dashed lines correspond to the two upper bounds of the Bragg angles when using Cu Kα and Mo Kα radiation. [27]

5.3 Lorentz polarization factor

The Lorentz polarization factor consists of two parts: the Lorentz factor and the polarization factor. Both are a function of the diffraction angle θ. The Lorentz factor accounts for two aspects of the geometrical effects. The first one is due to the finite size of the reciprocal lattice points and the second one is due to the variable radii of the Debye rings. A wider explanation of these effects is beyond the scope of this work. For this, the reader is referred to Section 8.6.4 of [27]. The results of both effects is given by the following function:

$$L = \frac{1}{\cos(\theta)\sin^2(\theta)}.$$

Here any scaling factors are omitted, because they can be moved to the scaling factor in (7).
The polarization factor is a result from the partial polarization of the X-rays after scattering. The partially polarized beam can be decomposed into a parallel and perpendicular part with respect to the goniometric axis. The diffracted intensity is proportional to the square of the amplitude of both components. The squared amplitudes are 1 for the parallel part and $\cos^2(2\theta)$ for the perpendicular part. Together, they form the polarization factor:

$$P = \frac{1+\cos^2(2\theta)}{2},$$

A monochromator can be used to remove any unwanted wavelengths, such that only one wavelength remains. When a monochromator is used, this gives additional polarization. The effect on the polarization factor is as follows:

$$P = \frac{1-K+K\cos^2(2\theta_{mono})\cos^2(2\theta)}{2},$$

where $2\theta_{mono}$ is the Bragg angle of the reflection of the monochromator and K is the fractional polarization of the beam [27].
Combining the Lorentz and polarization factor gives:

$$LP_j = \begin{cases} \dfrac{1+\cos^2(2\theta_j)}{2\sin^2(\theta_j)\cos(\theta_j)} & \text{No monochromator,} \\[2ex] \dfrac{1-K+K\cos^2(2\theta_{mono})\cos^2(2\theta_j)}{2\sin^2(\theta_j)\cos(\theta_j)} & \text{With monochromator.} \end{cases}$$

A representation of the effect of the Lorentz polarization factor without a monochromator is shown in Figure 22. We will only use this factor at small angles ($20° \leq 2\theta \leq 60°$), but for completeness, the entire 2θ range is represented.

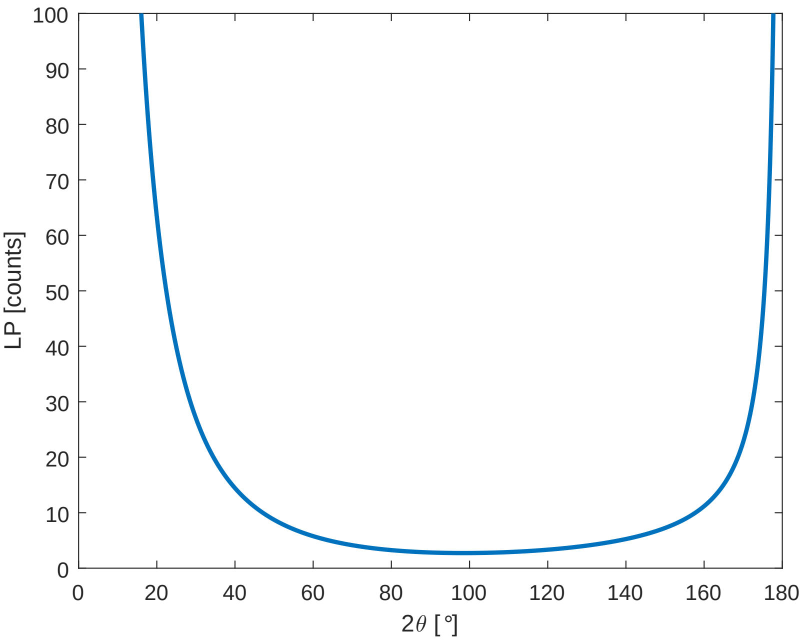


Figure 22: Lorentz polarization factor without a monochromator as function of the diffraction angle 2θ.

5.4 Peak shape function

The shape of the Bragg peaks can differ in width and height for each component. Furthermore, multiple peaks of the same component can also differ in shape. One of the parameters to describe the shape of a peak is the Full Width Half Maximum (FWHM), which we will denote with H. This is a measure for the width of the peaks at half of the maximal height. As we said earlier, this need not be a constant value. Usually, it is a function of θ. This function is different for other peak shapes. Originally, Rietveld assumed the peaks to be Gaussian functions [31]:

$$Gaussian : \xi_{ij} = \frac{\sqrt{4\ln(2)}}{H_{G,j}} \frac{1}{\sqrt{\pi}} e^{-\frac{4\ln(2)}{H_{G,j}^2}(2\theta_i - 2\theta_j)^2},$$

where $H_{G,j}$ is the FWHM of the jth reflection, $2\theta_i$ is the measured angle and $2\theta_j$ is the angle of the jth reflection. The FWHM is given by the function:

$$H_{G,j} = \sqrt{U\tan^2(\theta_j) + V\tan(\theta_j) + W}, \qquad \text{with } U, V \text{ and } W \text{ parameters to be optimized.}$$

A Gaussian was good as first approximation, but the tails were too narrow. Therefore in later approximations, a Lorentzian was added and the combination of a Gaussian and a Lorentzian was used, called the Pseudo-Voigt function. The function of a Lorentzian is:

$$Lorentzian : \xi_{ij} = \frac{2}{\pi H_{L,j}} \cdot \left(1 + \frac{4}{H_{L,k}^2}(2\theta_i - 2\theta_j)^2\right)^{-1},$$

where

$$H_{L,j} = \frac{X}{\cos(\theta_j)} + Y\tan(\theta_j), \qquad \text{with } X \text{ and } Y \text{ parameters to be optimized.}$$

The Pseudo-Voigt function is then a linear combination of the two shapes:

$$\xi_{ij} = PV_j = \eta \cdot Lorentzian + (1-\eta) \cdot Gaussian, \qquad \text{with } \eta \in [0,1].$$

Here, η is a function of the angle θ and is defined as:

$$\eta_j = \eta_0 + \eta_2 2\theta_j + \eta_2 (2\theta_j)^2, \qquad \text{with } \eta_1, \eta_2 \text{ and } \eta_3 \text{ parameters to be optimized.}$$

The Pseudo-Voigt function has a total of eight parameters that have to be optimized for each component.

Another possibility to approximate the peak shapes is the Pearson-VII function [27]. This is similar to the Lorentzian, but now the exponent varies:

$$\xi_{ij} = PVII_j = \frac{\Gamma(\zeta)}{\Gamma(\zeta - 0.5)} \frac{\sqrt{4(2^{1/\zeta} - 1)}}{H_{G,j}} \frac{1}{\sqrt{\pi}} \left(1 + \frac{4(2^{1/\zeta} - 1)}{H_{G,j}^2}(2\theta_i - 2\theta_j)^2\right)^{-\zeta}.$$

The gamma function is needed to normalize the function, i.e. $\int PVII\,\mathrm{d}\theta = 1$. For $\zeta = 1$, this is equal to the Lorentzian, for $\zeta \approx 10$ this is similar to a Gaussian. The coefficient ζ also depends on θ and is given by the function:

$$\zeta_j = \zeta_0 + \zeta_2(2\theta_j)^{-1} + \zeta_2(2\theta_j)^{-2}, \qquad \text{with } \zeta_1, \zeta_2 \text{ and } \zeta_3 \text{ parameters to be optimized.}$$

The Pearson-VII function has a total of six variables. All four peak shapes are shown in Figure 23 as an illustration of their relations between width and height. Later, we will determine which peak shape function is best to use when we analyze the diffraction patterns (in Section 7.5).

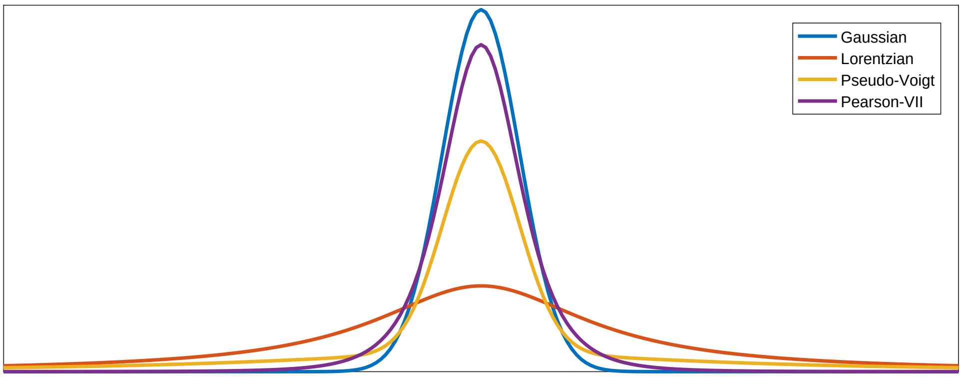


Figure 23: Different peak shapes for the approximation of a Bragg peak.

5.5 Preferred orientation

When a crystal grows, it might have a preferred growing direction (plane), because this direction is energetically more favorable. In a powder diffraction pattern, this effect is absent due to the random orientation of the particles. However, we do not have a powder and thus the effect of a preferred orientation is important. The mutual ratio between the peaks changes due to the preferred orientation of one plane, thereby some peaks might disappear in the observed diffraction pattern, while others become stronger.

To account for the effect of the preferred orientation of one plane on the intensity of the reflections, we need the angle between two planes. For example, if the plane (500) is the preferred plane, then the intensity of the (600) plane will also increase, since they are not so different (small angle/parallel). However, reflections of planes that are perpendicular to (500) will diminish. Let ω_j be the angle between a preferred plane $p_1 = (h_1k_1l_1)$ and another plane $p_j = (h_jk_jl_j)$. Since the angle between two planes is equal to the angle between their normal vectors, the angle ω can be calculated by:

$$\cos(\omega_j) = \frac{\vec{p}_1}{\|\vec{p}_1\|} \bullet \frac{\vec{p}_j}{\|\vec{p}_j\|}.$$

However, the Miller notation for the planes is with respect to the lattice parameters. Therefore, the Miller indices should not be used, but the real lengths and angles. The relations for the angle between planes for all different unit cells is given in [47].

Now that we have the angle between the planes, we need a function that translates these angles to a factor. Originally, Rietveld [31] used a Gaussian to represent the effect of the preferred orientation with a free parameter G that has to be optimized for each diffraction pattern:

$$\mathcal{P}_j = e^{-G\omega_j^2}.$$

Later, a different (better) function has been suggested by W. Dollase [34]:

$$\mathcal{P}_j = \left(G^2 \cos^2(\omega_j) + \frac{\sin^2(\omega_j)}{G} \right)^{-\frac{3}{2}} .$$

When G is smaller than one, this means that the plane is preferred and when G is larger than one, the plane is avoided. The above formulas describe the effect of the preferred plane p_1 on another plane. A Bragg peak usually consists of reflections of multiple planes. The effect of one preferred plane with all the planes that make up one Bragg peak has to be calculated in order to determine how much the intensity of a peak will increase or decrease. To find all the planes that make up a Bragg peak, we use the tables that were introduced in the description of the multiplicity factor (also see Appendix B.1).

A component can have multiple preferred planes, as indium has for instance. An example of the preferred orientations in indium is shown in Figure 24. The two preferred planes are (101) and (002), with factors G_1 and G_2 respectively. The peak shapes are Gaussians, but we simply used them to represent a peak (instead of just crosses). Their shape is not important here.

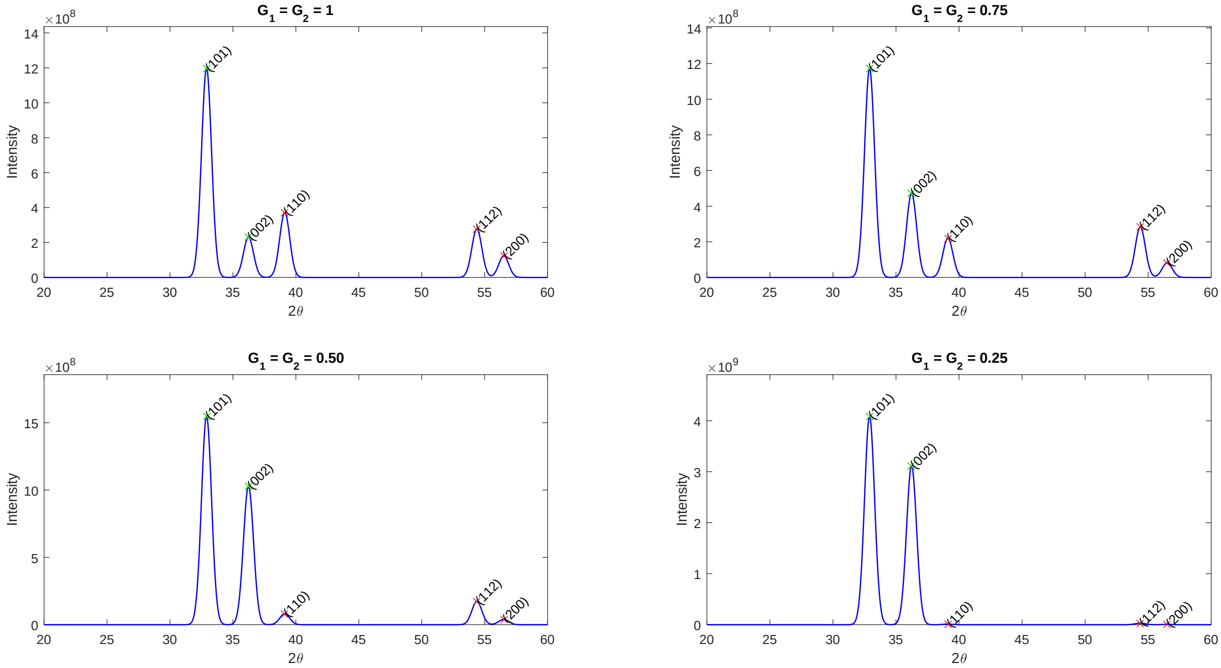


Figure 24: Relative intensities of the theoretical diffraction pattern of indium with preferred planes (101) and (002) (green crosses) for different values of G_1 and G_2.

6 Methods for optimization

In the Rietveld method, there are many parameters that can be varied. For each diffraction pattern, these parameters have to be fitted. To find a closest approximation to the measurements, we defined a least squares minimization problem. Depending on which variable we want to fit, we can use different methods to solve the minimization problem. For instance, the optimization of the background intensity is a linear least squares problem. This can be solved exactly. However, in the Rietveld refinement most variables have a nonlinear dependency. For those parameters, we have to use another algorithm. Our goal is to find a solution for the parameters ρ in the weighted least squares optimization problem (9):

$$\begin{aligned} &\min_{\rho} \Omega = \sum_{i=1}^{n} w_i \left(I_i^{obs} - f(2\theta_i, \rho - I^{bg}) \right)^2 \\ &\text{s.t.}\, g(2\theta_i, \rho) \leq 0, \end{aligned} \tag{9}$$

where w_i are the weights, f is the (non)linear function (in ρ), representing I^{calc} in 6 (Section 5), and g are the (non)linear constraints. The methods we will discuss are:

- Exact (global) solution to the linear unconstrained least squares problem,
- Brute-force method, a nonlinear global optimization method,
- Interior-point algorithm, a nonlinear constraint local optimization algorithm,
- OptQuest NLP algorithm, a nonlinear constraint global optimization algorithm,
- Genetic algorithm, a nonlinear constraint global optimization algorithm.

We always aim to find the global optimum. This can be quite challenging, since there might be many local minima present. Due to the working principles of each algorithm, they might get stuck at a local minimum. However, sometimes there is only one minimum in the region of interest and then we can use a local optimization method. Therefore, in the analysis of a measurement we have to compare the results of the algorithms and choose the one that performs the best (lowest value of Ω and fastest convergence). In this chapter, many additional variables are defined. The meaning of these variables are restricted to each section only and have no meaning in the other parts of this work.

6.1 Multivariable linear least squares

Assume that in (9) f is a linear function of ρ and there are no constraints g, then the minimization problem can be solved exactly for ρ. Let ρ be a vector with all the parameters used for optimization. Since f is linear, we can write it as a linear combination of these variables:

$$f(2\theta_i, \rho) = X_{i,1}\rho_1 + X_{i,2}\rho_2 + \cdots + X_{i,M}\rho_M,$$

where X is a matrix dependent on $2\theta_i$. Thus, let us write $f = X\rho$, where $X \in \mathbb{R}^{n\times M}$ and $\rho \in \mathbb{R}^{M\times 1}$ (with $M < n$). Furthermore, let $\mathbf{w} \in \mathbb{R}^{n\times n}$ be a diagonal matrix containing the weights on the diagonal and let $\mathbb{I}^{obs} \in \mathbb{R}^{n\times 1}$ be the vector with the observed intensity. We rewrite the least squares minimization problem (9) as:

$$\begin{aligned} \Omega &= \sum_{i=1}^{n} w_i \left(I_i^{obs} - f(2\theta_i, \rho) \right)^2, \\ &= \|\mathbf{w}^{\frac{1}{2}}(\mathbf{I}^{obs} - X\rho)\|^2, \\ &= (\mathbf{I}^{obs} - X\rho)^{\mathrm{T}}(\mathbf{w}^{\frac{1}{2}})^{\mathrm{T}}\mathbf{w}^{\frac{1}{2}}(\mathbf{I}^{obs} - X\rho), \\ &= (\mathbf{I}^{obs})^{\mathrm{T}}\mathbf{w}\mathbf{I}^{obs} - 2(\mathbf{I}^{obs})^{\mathrm{T}}\mathbf{w}X\rho + \rho^{\mathrm{T}}(X^{\mathrm{T}}\mathbf{w}X)\rho. \end{aligned}$$

This is a second order polynomial function in ρ. To get the minimum, we set the derivative with respect to ρ to zero:

$$\begin{aligned} &\frac{\partial \Omega}{\partial \rho} = -2X^{\mathrm{T}}\mathbf{w}\mathbf{I}^{obs} + 2(X^{\mathrm{T}}\mathbf{w}X)\rho = 0, \\ &\Rightarrow \rho = (X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}}\mathbf{w}\mathbf{I}^{obs}. \end{aligned}$$

Now, we have an expression for an optimal value of ρ, but is this really a minimum (and not a maximum)? And is the matrix $X^{\mathrm{T}}\mathbf{w}X$ invertible?

As for the invertibility of the matrix $X^{\mathrm{T}}\mathbf{w}X$, the columns of $\mathbf{w}^{\frac{1}{2}}X$ have to be linear independent. Since $\mathbf{w}$ is a diagonal matrix with all (positive) scalar values, this does not change the linear dependency of the whole matrix. As for the columns of X, we can always find a number of columns that are linear independent. For this submatrix we can redefine the parameters ρ as a linear combination of the old parameters. By doing so, we created a new system where the columns of X are really linear independent and thus the matrix $X^{\mathrm{T}}\mathbf{w}X$ is invertible and we have an optimal solution for ρ.

To show that $\rho = (X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}}\mathbf{w}\mathbf{I}^{obs}$ is really a minimum, we define a different $\hat{\rho}$ that is the global minimum and show that these two values are equal to each other. To reduce the writing, we introduce another matrix: $E = \mathbf{I}^{obs} - X\rho$. Then we have a global minimum that satisfies:

$$\begin{aligned}
\Omega &= (\mathbf{I}^{obs} - X\hat{\rho})^{\mathrm{T}}\mathbf{w}(\mathbf{I}^{obs} - X\hat{\rho}),\\
&= (\mathbf{I}^{obs} - X\rho + X\rho - X\hat{\rho})^{\mathrm{T}}\mathbf{w}(\mathbf{I}^{obs} - X\rho + X\rho - X\hat{\rho}),\\
&= (E + X(\rho - \hat{\rho}))^{\mathrm{T}}\mathbf{w}(E + X(\rho - \hat{\rho})),\\
&= E^{\mathrm{T}}\mathbf{w}E + 2E^{\mathrm{T}}\mathbf{w}X(\rho - \hat{\rho}) + (X(\rho - \hat{\rho}))^{\mathrm{T}}\mathbf{w}(X(\rho - \hat{\rho})),\\
&= E^{\mathrm{T}}\mathbf{w}E + \|\mathbf{w}^{\frac{1}{2}}X(\rho - \hat{\rho})\|^2.
\end{aligned}$$

Here, we used that $E^{\mathrm{T}}\mathbf{w}X = 0$, which follows from substituting the definitions of E and ρ:

$$\begin{aligned}
E^{\mathrm{T}}\mathbf{w}X &= (\mathbf{I}^{obs} - X\rho)^{\mathrm{T}}\mathbf{w}X,\\
&= (\mathbf{I}^{obs} - X(X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}}\mathbf{w}\mathbf{I}^{obs})^{\mathrm{T}}\mathbf{w}X,\\
&= (\mathbf{I}^{obs})^{\mathrm{T}}(I - X(X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}}\mathbf{w})^{\mathrm{T}}\mathbf{w}X,\\
&= (\mathbf{I}^{obs})^{\mathrm{T}}(I - \mathbf{w}X(X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}})\mathbf{w}X,\\
&= (\mathbf{I}^{obs})^{\mathrm{T}}(\mathbf{w}X - \mathbf{w}X(X^{\mathrm{T}}\mathbf{w}X)^{-1}X^{\mathrm{T}}\mathbf{w}X),\\
&= 0.
\end{aligned}$$

Returning to the equation for the global minimum; $E^{\mathrm{T}}\mathbf{w}E$ is the global minimum when the term with the norm is equal to zero. The only possibility for the norm term to be equal to zero is when $\rho = \hat{\rho}$. Thus, the optimum we found earlier is truly a global minimum.

6.2 Brute-force

The brute-force method is a simple algorithm that we created for small nonlinear unconstrained problems. Say we want to minimize a general nonlinear function $f(x)$ (representing Ω in (9)) for $x_{\min} \leq x \leq x_{\max}$. If the function is easy to evaluate (computationally inexpensive) and slowly changing (no large derivatives), then we could use the brute-force method.

We first create a set of points x_i with $i = 1, \ldots, m$ that cover the bounds of the parameters $x \in [x_{\min}, x_{\max}]$. Next, the function is evaluated at all these points (hence the name) and we take the minimum: $\min_i f(x_i)$. This would be inefficient for a large m, thus we take an m such that all aspects of the function are represented, i.e. a linear interpolation of the function values at the selected points would be a good approximation to the function. Assume the minimum is found at the $\hat{\imath}$-th position. Then, we repeat the procedure with $x \in [x_{\hat{\imath}-1}, x_{\hat{\imath}+1}]$ to get a more accurate answer. As the interval is located around the minimum, we assume that the shape of the function in this interval is a parabola (or cone). Therefore, we can use a very small value of m, like 10. This whole procedure is done n times. Thus we have two parameters in this algorithm, m and n. In the remainder of this work, we will denote this algorithm as BF^n_m.

The number of function evaluations grows exponentially with the number of parameters in f and linear in the number of points m. When we use for the second and further iterations $m = 10$, the total number of function evaluations is $m^{\dim(x)} + 10(n-1)$. Since m (for the first iteration) is generally a number between 25 and 100, this algorithm is efficient when there is one or at most two parameters in f. Otherwise, the number of function evaluations in the first iterations is too large and other algorithms are almost surely faster. Thus, we only created the algorithm for one parameter (BF^n_m) or two parameters ($\mathrm{BF2}^n_m$). The codes are shown in Appendix D.2.1 and D.2.2 respectively.

6.3 Interior-Point

The Interior-point algorithm, as we call it, is a combination of Sequential Quadratic Programming (SQP) methods and interior-point methods. In MATLAB, this is one of the algorithms in the `fmincon` function. SQP methods are very efficient for solving medium-size nonlinear programs and interior-point methods are useful for large linear programs [35]. The combination of the two works for large nonlinear programs. The problem that we consider is:

$$\begin{aligned} &\min_x f(x) \\ &\text{s.t. } g(x) \leq 0, \end{aligned} \tag{10}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ and nonlinear constraints $g : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions.
Starting with the interior-point approach, we define the barrier problem:

$$\begin{aligned} &\min_x f(x) - \mu \sum_{i=1}^{m} \ln(s_i) \\ &\text{s.t. } g(x) + s = 0, \end{aligned} \tag{11}$$

where $\mu > 0$ and the slack variables s_i are assumed to be greater than zero. The goal is to find an approximate solution to (11) for fixed μ. Then the procedure is repeated for decreasing values of μ to finally find an approximation to the original problem (10).
To find an approximation to (11), we first define the Lagrangian of (11):

$$\mathcal{L}(x, s, \lambda) = f(x) - \mu \sum_{i=1}^{m} \ln(s_i) + \lambda^T (g(x) + s),$$

where $\lambda \in \mathbb{R}^m$ are the Lagrangian multipliers. The first order optimality conditions (first derivative of the Lagrangian equals zero) for the Barrier problem are:

$$\begin{aligned} \nabla_x \mathcal{L} &= \nabla f(x) + A(x)\lambda = 0, \\ \nabla_s \mathcal{L} &= \mu S^{-1} e + \lambda = 0, \end{aligned}$$

where $A(x) = (\nabla g_1(x), \ldots, \nabla g_m(x))$, $e = (1, \ldots, 1)^T$ and $S = \texttt{diag}(s_1, \ldots, s_m)$. Solving the first order optimality condition gives us the Lagrangian multipliers λ. These we will use later in the SQP method. Furthermore, we redefine our variables to be represented by only one symbol:

$$\begin{aligned} z &= (x, s)\,, \\ \phi(z) &= f(x) - \mu \sum_{i=1}^{m} \ln(s_i), \\ c(z) &= g(x) + s, \end{aligned}$$

then the barrier problem (11) becomes:

$$\begin{aligned} &\min_z \phi(z) \\ &\text{s.t. } c(z) = 0. \end{aligned} \tag{12}$$

Now we use an SQP method to solve (12). This means that we solve z iteratively, i.e. $z_{j+1} = z_j + d$ with a small update step $d = (d_x, d_s)$. This d is the solution of the quadratic program:

$$\begin{aligned} &\min_d \nabla \phi(z)^T d + \frac{1}{2} d^T W d \\ &\text{s.t. } \hat{A}(z)^T d + c(z) = 0, \end{aligned} \tag{13}$$

where W is the Hessian of the Lagrangian: $W = \nabla^2_{zz} \mathcal{L} = \begin{pmatrix} \nabla^2_{xx} \mathcal{L} & 0 \\ 0 & \mu S^{-2} \end{pmatrix}$ and $\hat{A}$ is the Jacobian of $c(z)$: $\hat{A}(z)^T = \begin{pmatrix} A(x)^T & I \end{pmatrix}$. Next, we define a trust region: $\|(d_x, S^{-1} d_s)\| \leq \Delta$, where $\Delta > 0$

is changed every iteration and the Euclidean norm is used. The slack step is multiplied with S^{-1}, because this is shown to be more effective [36]. This trust regions does not necessarily guarantee that $s + d_s$ remains positive. Therefore, an extra constraint is used to keep the slack variables positive:

$$s + d_s \geq (1 - \tau)s,$$

with the parameter $\tau \in (0, 1)$ and close to 1. Then the quadratic problem (13) becomes:

$$\begin{aligned} \min_d \, & \nabla\phi(z)^T d + \frac{1}{2} d^T W d \\ \text{s.t.} \, & \hat{A}(z)^T d + c(z) = 0 \\ & \|(d_x, S^{-1} d_s)\| \leq \Delta \\ & d_s \geq -\tau s, \end{aligned} \tag{14}$$

In general, it is hard to find an exact solution d to (14). It is very well possible that when d satisfies the constrains, it does not lie within the trust region. There are several ways to adjust the constraints such that an approximate solution can be found. As in [35], first, a v is found that lies well within the trust region, then solve d such that the constraints are still satisfied. To calculate v, the trust region is made smaller by introduction of parameter $0 < \xi < 1$ and replacing Δ by $\xi\Delta$. Then the problem is to find v such that the constraint is minimized in this tightened trust region:

$$\begin{aligned} \min_v \, & \|\hat{A}(z)^T v + c(z)\| \\ \text{s.t.} \, & \|(v_x, S^{-1} v_s)\| \leq \xi\Delta \\ & v_s \geq -\xi\tau s. \end{aligned}$$

Now d can be approximated by solving the modified version of (14):

$$\begin{aligned} \min_d \, & \nabla\phi(z)^T d + \frac{1}{2} d^T W d \\ \text{s.t.} \, & \hat{A}(z)^T d = \hat{A}(z)^T v \\ & \|(d_x, S^{-1} d_s)\| \leq \Delta \\ & d_s \geq -\tau s. \end{aligned}$$

The constraints are now always satisfied, since $d = v$ is a feasible solution.
Finally, we need to decide whether or not we accept the calculated step d. Therefore, a merit function is defined:

$$\psi(z, \nu) = \phi(z) + \nu\|c(z)\|, \tag{15}$$

where $\nu > 0$ is the penalty parameter. This merit function is defined in such a way that when ν is greater than a certain threshold value, then a Karush-Kuhn-Tucker (KKT) point of the barrier problem (12) is a stationary point of the merit function ψ. When such a KKT point exists, then it is a minimizer of the barrier problem [37]. A point $\bar{z}$ is a KKT point of problem (12), if there exists multipliers $\bar{y}_i \geq 0$ with $i = 1, \ldots, m$ such that:

$$\nabla\phi(\bar{z}) + \sum_{i=1}^{m} \bar{y}_i \nabla c_i(\bar{z}) = 0.$$

Upon setting $\nu = \max_i(\bar{y}_i)$, this can be rewritten as $\nabla\phi(\bar{z}) + \nu\|\nabla c(\bar{z})\| = 0$. This is exactly the derivative of the merit function set equal to zero, meaning that $\bar{z}$ is a stationary point. When the step d gives a sufficient reduction in the merit function (15), then it is accepted, otherwise it is discarded. This decision is based on comparing the reduction with a predicted reduction of the merit function. The precise definition of the predicted reduction is beyond the scope of this work, but it can be found in Chapter 2 of [35]. The iteration is completed by updating the trust region radius Δ. When d was accepted, Δ is halved, otherwise Δ is increased by a constant factor.
Next, we will give a short summary how the algorithm works in pseudo-code.

Interior Point Algorithm

1. Initialize by selecting a $\mu > 0$, $z_0 = (x_0, s_0)$ and Lagrange multipliers λ.
2. If the problem (10) has been solved accurately enough, stop, else, continue.
3. Approximately solve the barrier problem (12) by performing the following steps:
   (a) Set the initial trust region radius $\Delta > 0$, the tightening parameter $\xi \in (0, 1)$ and the penalty parameter $\nu > 0$
   (b) If the barrier problem (12) is solved accurately enough, go to 4.
   (c) Calculate $v = (v_x, v_s)$ by solving the first subproblem:
   $$\begin{aligned} \min_{v} \ & \|\hat{A}(z)^T v + c(z)\| \\ \text{s.t.} \ & \|(v_x, S^{-1} v_s)\| \leq \xi\Delta \\ & v_s \geq -\xi\tau s. \end{aligned}$$
   (d) Calculate the step $d = (d_x, d_x)$ by solving the second subproblem:
   $$\begin{aligned} \min_{d} \ & \nabla\phi(z)^T d + \frac{1}{2} d^T W d \\ \text{s.t.} \ & \hat{A}(z)^T d = \hat{A}(z)^T v \\ & \|(d_x, S^{-1} d_s)\| \leq \Delta \\ & d_s \geq -\tau s. \end{aligned}$$
   (e) When the step d gives a sufficient reduction of the merit function (15), set $z_{j+1} = z_j + d$, compute the new Lagrangian multipliers and go to 3(b). Otherwise, decrease Δ and go to 3(c).
4. Decrease the barrier parameter μ and return to 2.

The detailed descriptions how both subproblems are solved are omitted here, since this is beyond the scope of this work. The idea for solving both problems is that the solutions have a certain specific form, e.g. the steepest descent direction of the steps. Then, this step has to live up to some predictions to be accepted. The exact definitions can be found in [35] and [36].

6.4 OptQeust NLP

The OptQuest NonLinear Program (OQNLP) algorithm is an algorithm that is intended to find global minima of smooth (un)constrained mixed integer nonlinear programs [38], thus in particular our least squares minimization problem (9). It generates a set of trial points with the Scatter Search algorithm. These candidate points are filtered to form a smaller subset that are the initial points for a local solver, for instance the Interior-Point algorithm. In MATLAB, this algorithm is embedded in the `GlobalSearch` function.

The idea of a multi-start algorithm is to find the global minimum by using a set of multiple points and starting a local gradient-based solver from all of these points. Each local minimum has a basin of attraction. This is a subset of the entire space that consist of all the points that converge to the same local minimum. The probability that there is a starting point that converges to the global solution approaches one as the number of starting points goes to infinity. However, this is very inefficient because each local minimum is found many times. Thus, the goal is to find the smallest set of starting points such that we still find the global solution. There are several procedures for filtering the large set of starting points, such as the Multi-Level Single Linkage (MLSL) procedure [39,40] and the Random Linkage algorithm (RL) [41]. Both use the objective function value of the starting points as criteria for filtering starting points. Furthermore in MLSL, points, that are within a critical distance of each other or withing a critical distance of a local minimum, are also filtered out. In RL, there is a probability function that determines if a local solver is started based on the distance from the current point to the closest points with a better objective function value. Both algorithms can only solve constraint problems with no discrete variables. The Scatter Search algorithm can handle unconstrained problems and also discrete variables.

6.4.1 Scatter Search

The Scatter Seach algorithm works with a set of points that are iteratively updated by deterministic combinations of the previous points. This is different from a genetic algorithm that uses probability based decisions. First, we give the step wise procedure of the Scatter Search algorithm and then a more detailed description of each step. For simplicity, we do not involve discrete variables.

Scatter Search Algorithm

```
    1. Initialize: create a set of three points: one point where all variables equal their lower bounds, one point where they are equal to the upper bounds and one point where they equal the mid-points between their bounds.
    2. Use a diversification generation method to create a diverse set of starting points that satisfy the linear constraints combined with the previous set of points.
    3. Evaluate for every initial point the penalty function $P_{OQ} = \Omega(\rho) + \sum_i w_i \cdot viol(g_i(\rho))$, which is the objective function $\Omega(\rho)$ plus the penalty weights $w_i$ times the maximum percentage violation of the nonlinear constraints $g$. This $P_{OQ}$ is used as quality measure of the points.
While stopping criteria not satisfied
    While some distinct pair of points has not been processed
        4. Select a pair of starting points
        5. Use this pair of points in a solution combination method to create a small set of trial points that satisfy the linear constraints.
        6. Evaluate for each trial point the penalty function $P_{OQ}$.
    End
    7. Update the set of points
    8. If the set of points has changed, return to step 4. Else, restart the procedure by returning to step 2 with the best half points as initial set.
End
```

Step 1 generates the initial set of three points. A fourth point can be added by a user-specified initial point.
Step 2 uses the diversification generation method to generate the remaining points in the initial set. First, this method generates many random points in feasible region of the linear constraints. Then, random points are selected and added to the initial set that are farthest from its neighbor in the initial set. This continues until the initial set has the desired number of points.
Step 3 sorts the points in the set based on their value of the penalty function P_{OQ}.
Step 4 and 5 generates a set of trial solutions by selecting two points and using the solution combination method. This method creates points around the two selected points. Let ρ_1 and ρ_2 be the two chosen points, then five positions $v_1, \ldots, v_5$ are calculated positions are calculated:

$$
\begin{aligned}
d &= (\rho_2 - \rho_1)/2, \\
v_1 &= \rho_1 - d, \\
v_2 &= \rho_1, \\
v_3 &= \rho_1 + d, \\
v_4 &= \rho_2, \\
v_5 &= \rho_2 + d,
\end{aligned}
$$

where d is half of the Euclidean distance between the points. These five positions are used to create four (hyper-)rectangles, shown in Figure 25.

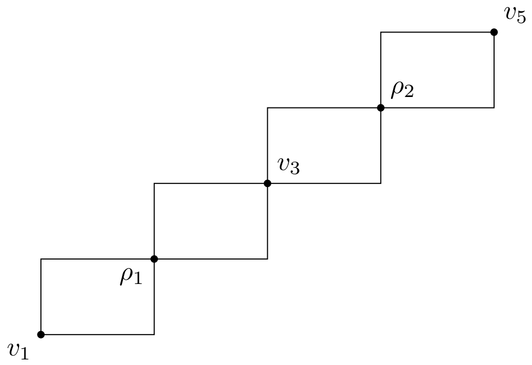


Figure 25: Rectangles for the new trial points in the solution combination method.

Now four random trial points are generated in the four rectangles. These four generated trial points have to satisfy the linear constraints, otherwise they are discarded.
Step 6 evaluates the penalty function P_{OQ} at the trial points.
Step 7 updates the set of points. This is done by replacing the set by the best performing trial points. This is a rigorous update step, since it emphasizes quality over diversity. There is an option to set the diversity and quality equally important. Then, the set of points is split in half. One half is created as described earlier, focusing on quality. The other half focuses on diversity. This is done by checking the distance to any point in the diversity half set. When the minimum distance of this point is larger than that of any point already in the set, then the new point will replace it.
Step 8 makes a decision based on if the set of points is changed or not. When the set is changed, then Steps 4 till 7 are repeated. Otherwise, the best half points are taken from the set and it returns to Step 2. This process is continued until the stopping criteria are met. These criteria might be that the change in quality or diversity of the set of points is too small.

After the set of starting points is generated, a nonlinear program solver can be used to find all the local minima. The global minimum is then the minimum of all the local minima. There are several local solvers, but we will only use the Interior-point algorithm and not further investigate the difference between other local solvers, since we assume that any local solver can find the local minima adequately if their starting point is sufficiently close to the local minimum.

6.5 Genetic algorithm

The genetic algorithm (GA) is based on the biological system of reproduction and evolution. In MATLAB this algorithm is incorporated in the `ga` function. A population (set of points) is evaluated by an objective function and by a 'survival of the fittest' principle the most potent parents are chosen for reproduction (with a certain probability), yielding a set of children. This set of children forms the new generation and this generation should be better than the previous one in terms of the average of their value of the objective function. From this small explanation, a few things become clear about genetic algorithms that distinguish them from any other optimization method. Namely, genetic algorithms work with a set of points instead of just one point and the translation rules are based on probability instead of deterministic rules, e.g. update schemes. Furthermore, genetic algorithms only use the objective function, not any derivatives or other knowledge about the system. In that sense, the system can be viewed as a black box with a few input parameters and an output value. The genetic algorithm then tries to optimize the input parameters to obtain the maximal output.
To explain how a genetic algorithm works, we will use an example to illustrate the different aspects. This section is based on the book of D. Goldberg, *Genetic Algorithms in Search and Optimization & Machine Learning* [42]. Here we will only focus on the key part of genetic algorithms: reproduction, crossover and mutation. These three operators form the basic of genetic algorithms and thus we call it the simple genetic algorithm. However, many additional processes can be added to make the algorithm more robust. This will briefly be discussed in the last paragraph.
As example, let us compute the maximum of the function $f(x) = x^3 - 45x^2 + 600x$ for $x \in [0, 31]$. Upon first sight, this function looks strange, but when visualizing the function, it becomes clear why we chose it (see Figure 26).

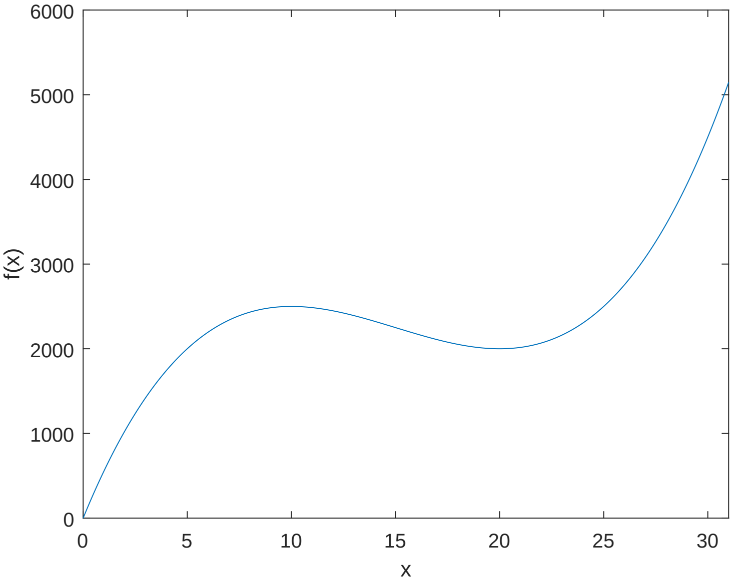


Figure 26: The function to maximize for the genetic algorithm.

This function has two extrema at $x = 10$ and $x = 20$, a local maximum and minimum respectively. Furthermore, the global maximum is at $x = 31$. The goal for the genetic algorithm is to find the global maximum and not the local maximum. There is only one parameter, x. Instead of using the decimal representation, we will use the binary representation. This is a string of length five. The reason for this is that we now have five parameters that are either zero or one instead of one parameters that has 32 options (assuming we work with integers only). This is an advantage since it creates more diversity during the optimization. Let us now create an initial population with four strings (parents) of length five. This is done by a simple coin flip of that either gives a zero or a one. The initial population is:

10100
00011
10001
11011

To create the next generation we will use three operators: reproduction, crossover and mutation. There three operators, all based on probability, determine who is fit enough for mating and how their children look like.

6.5.1 Reproduction

During the reproduction, it will be decided who will produce children. This is based on their fitness value (value of the objective function). The higher the fitness value, the higher chance for reproduction. We calculate the fitness value and their percentage of the total fitness for the initial population. These results are shown in Table 2.

Table 2: The initial population with their fitness value and percentage of the total fitness

#	String	x	$f(x)$	% of total
1	10100	20	2000	23.23
2	00011	3	1422	16.52
3	10001	17	2108	24.49
4	11011	27	3078	35.76
Total			8608	
Average			2152	
Maximum			3078	

We will use the percentages as a change to be selected for mating. Four intervals are created by taking the cumulative values of the percentages, e.g. $[0, 0.2323], [0.2323, 0.3975], [0.3975, 0.6424]$ and $[0.6424, 1]$. Then four random numbers between zero and one are drawn from a uniform distribution. Then, we select four strings for which the random numbers are inside the corresponding intervals. In Figure 27 the intervals are shown and the random numbers that are used to pick four strings, represented by arrows.

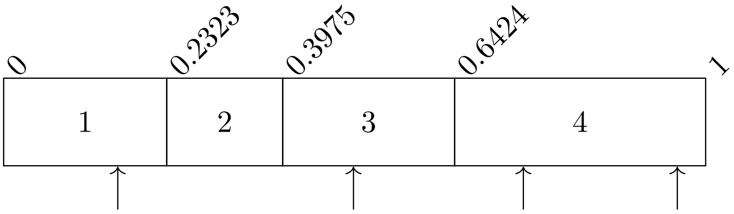


Figure 27: Intervals of the four strings and the random numbers represented by the arrows.

The four strings that are chosen are 1, 3, 4 and 4. As one would expect, strings with a high probability are represented more than strings with a low probability. These four chosen strings will now mate and produce children via the crossover and mutation process.

6.5.2 Crossover and mutation

For mating, we pick two random strings from our chosen set. Let this be the strings 1 and 4, which leaves strings 3 and 4 to mate also. As in nature itself, the children look like their parents. They are not the same as one of their parents, but they have a bit of both. Therefore we cut both strings at a certain position and interchange their tails. There are four possible sites where we can cut the strings. This is again determined randomly. For the first couple we find the random number 2 and for the second couple we get the number 3. This process for the first couple is shown in Figure 28.

$$\begin{array}{ll|l} \#1 & 1\,0 & 1\,0\,0 \\ \#4 & 1\,1 & 0\,1\,1 \end{array} \quad \Rightarrow \quad \begin{array}{l} 1\,0\,0\,1\,1 \\ 1\,1\,1\,0\,0 \end{array}$$

Figure 28: Crossover process between string 1 and 4 with crossover site 2 (measured from the left).

Mutation can change any zero to a one or vice versa in the children. Usually, the chance for mutation to happen is very low, say 0.001. The reason for this is to not completely change the children, but only every once in a while to keep diversity. Without mutation, the population is very likely to be dominated by the fittest individual and it might get stuck at a local maximum. For the second couple it so happens to be that the first child is mutated at the fifth position. The crossover and mutation for the second couple is shown in Figure 29.

$$\begin{array}{ll|l} \#3 & 1\,0\,0 & 0\,1 \\ \#4 & 1\,1\,0 & 1\,1 \end{array} \quad \Rightarrow \quad \begin{array}{l} 1\,0\,0\,1\,\textcircled{1} \Rightarrow 1\,0\,0\,1\,0 \\ 1\,1\,0\,0\,1 \end{array}$$

Figure 29: Crossover process between string 3 and 4 with crossover site 3 and mutation at the fifth position of the first children.

The new generation consists of the four children. The fitness of the children is shown in Table 3. With this new generation we can start again with the reproduction procedure and so create a loop that generates new generations.

Table 3: The fitness value of the new generation.

#	String	x	$f(x)$
1	10011	22	2068
2	11100	28	3472
3	10010	18	2052
4	11010	25	2500
Total			10092
Average			2523
Maximum			3472

We can clearly see that the average and maximum fitness value have increased for the new generation. However, since we used many random variables, so we might be lucky to have a better population. Therefore, we take a look at the long term performance.

6.5.3 Performance

To see the performance of the genetic algorithm on this example, we will use a population with more than four strings, because it is very easy to get stuck at the local maxima with four strings. To get away from the local maxima, a lucky mutation is needed. Then the genetic algorithm is more like a random walk. Therefore, we use a generation of twelve strings (still only integers) and we hope that this is enough to keep diversity in the population. We run the algorithm for twenty generations and the fitness of the best, average and worst performing strings are shown in Figure 30.

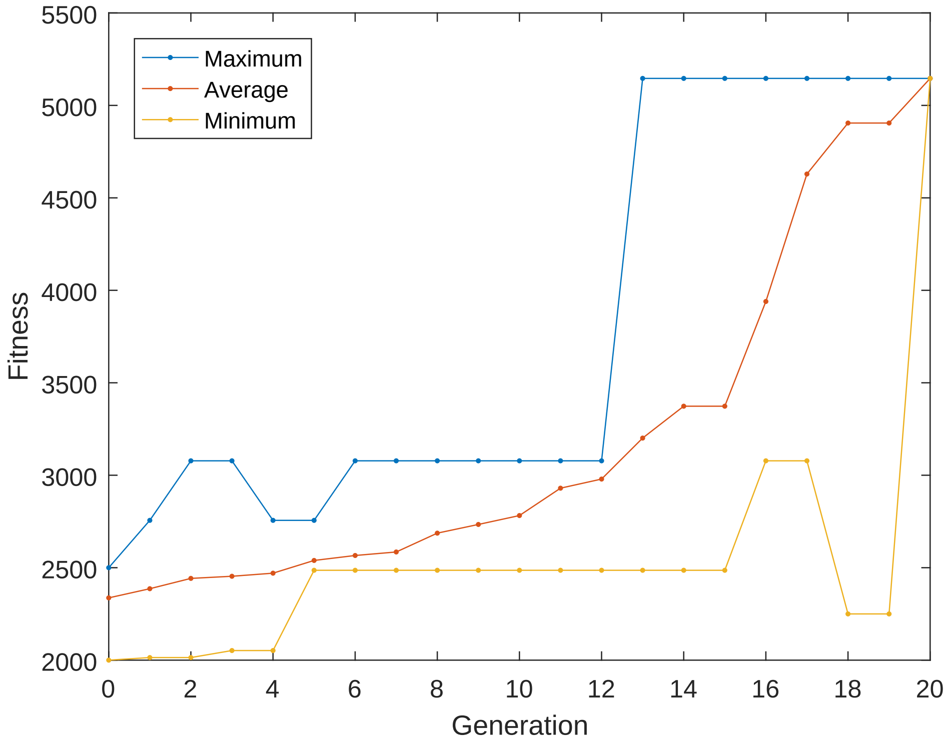


Figure 30: Performance of the (simple) genetic algorithm for twenty generations.

In the beginning, the strings are stuck at the local maximum and the entire population is dominated by that one value (as the average is close to the maximum). At generation 12, the optimal value is found and the whole population starts to adjust to the new optimum. Eventually, the whole population consists of the same strings as the minimum is equal to the maximum.

This behavior is not usual in more difficult problems. The best value converges towards the optimal value, but to actually achieve the optimum is difficult, since it is a random process. However, in the analysis of the XRD patterns, there is not one possibility for each phase, there might be several options that are approximately equally good fits. Thus, it is not necessary to find the exact optimum, a good approximation is enough.

We just claimed that the best string converges to the optimal solution. To prove this, a measure is needed to evaluate every generation. One possible measure is a schemata. This is part of a string that represents a set of strings, for instance *0*11, where the asterisk is the ‘do not care’-symbol. This schemata represents all the strings that have a zero at the second position and a one in the fourth and fifth position. Some schemata are more likely to survive after reproduction than others. For example, the schemata 0***1 is likely to be destroyed by crossover, where 11*** is likely to survive. The defining length is the number of positions between the two outer most fixed positions. For the two examples we gave the defining lengths are 4 and 1 respectively. Schemata with short defining lengths and high fitness are called building blocks. Upon analyzing the reproduction process, it can be shown that the expected amount of building blocks in a successive generation increases at least exponentially (see Chapter 2 in [42]). The idea is that schemata with high fitness are more likely to be used for reproduction and that schemata with short defining lengths are not likely to be destroyed by crossover. Furthermore, mutation occurs at a low frequency and thus has little effect on the important schemata. Concluding that the number of building blocks increase.

6.5.4 Additional operators

Dominance and diploidy

So far we only considered that a genotype has one chromosome (string), called a haploid. However, in nature it is more common to have a diploid. A diploid has two chromosomes that each contain information about the same feature. At first thought it seems unnecessary to keep an excess of information. Furthermore, how should nature decide from which string to take values? This is where dominance comes in. One allele (part of a chromosome) is always chosen above another allele. For example, alleles that decode for brown eyes are dominant over the ones for blue eyes. By using the dominance, the resulting chromosome can be calculated. As the alleles in the genetic algorithm have no physical or biological meaning, the dominance of the alleles is determined by a random process. Furthermore, it is possible that the dominance changes if necessary. The pair of chromosomes acts as a long term memory. By having two strings and using only the dominant parts of both, we keep some alleles shielded against the reproduction process. Alleles that are currently not useful might become useful in the future. This causes diploids to adapt fast if the environment changes. On the other hand, with haploids the reproduction process only keeps the currently highest performing chromosomes and it is harder to adapt in a changing environment.

Duplication and deletion

Duplication and deletion are two operators that effect the influence of mutation. With duplication a gene is duplicated and placed alongside its original on the chromosome. Deletion can remove duplicate genes from a chromosome. If the mutation rate is constant, then by duplicating a gene n times, the probability for that gene to be mutated also increases with a factor n. Conversely, deletion decreases the effective mutation rate. Once mutation occurs, we have to decide if we keep the original gene or the mutated gene. This decision can be made with a dominance-like structure.

Sexes and specialization

Dividing the population in two (or more) groups, such as males and females, can help to create more diversity in the population. Having different groups allows specialization to some extent. This specialization results in a broader range of possibilities upon mating and thus more diversity. For example, consider two professions: hunting and nurturing. Both are essential for survival. Without different groups all individuals should spend an equal amount of time hunting and nurturing. However with males and females, the males can go hunting and the females focus on nurturing. Their offspring would have the best of both.

Professions and speciation

Continuing the idea of sexes and specialization, we can give each individual in the population a job of profession. Thus we will form subgroups in the population. These subgroups each thrive in their own profession, leading to local maxima. To assure that the subgroups continue to exist in the next generations, the selection process during reproduction has to be adjusted. Instead of comparing the fitness of an individual with the fitness of the total group, it will be compared to the fitness of the subgroup and the number of individuals in that subgroup. For example, consider a store with several counters each having a different service rate. When there is a large queue for the fastest counter, it might be smarter to move to a slower counter where the queue is shorter. This is the idea of speciation. The largest subgroup converges to the global maximum, while several other subgroups converge to other local maxima. This might be useful in case there is no clear global maximum, but several local maxima that are all approximately equal.

7 Optimization of the parameters

The Rietveld refinement gives us a function to approximate the diffraction pattern by varying some parameters. So far, we have done the preliminary data analysis (without the base shift) and identified the different parameters in the Rietveld refinement and presented several (non)linear optimization methods. Now, we will determine which optimization method is useful for the optimization of which parameter in the Rietveld method. This will be done by comparing the quality of the fits and the convergence speed of each optimization method. The parameters are optimized in a specific order. This order is based on the same ordering as Young's [43]. A recap of the factors with the number of parameters ('Amount') together with the order in which we optimize them is shown in Table 4.

Table 4: The parameters in the Rietveld method with their order of optimization.

Parameter	Amount	Linear	Order
Scale	1	Yes	1, 2
Base shift	1	No	1
Preferred orientation factor	0-3	No	2
Lattice parameters	1-6	No	3 or not
Peak shape	2-8	No	4

The scale and base shift are the first two parameters that will be optimized. The scaling factor is in the order of 10^{-6} to 10^{-12}, so it is crucial that this is done first. Together with the base shift, which varies from a few hundredths to a few tenths of degrees, the position and the height are good enough such that the correct peaks are linked with the correct components. The next step is the preferred orientation factor to get the correct relative intensities of the peaks of every component. There is a possibility that the lattice parameters that we use are not a good fit, i.e. some peaks of one component are at the correct position, while others are slightly off. If this is the case, then the lattice parameters are optimized and the first two steps are repeated (the scale + base shift and the preferred orientation factor). The last parameters we optimize are those of the peak shape. We presented four different peak shape functions in Section 5.4. Here, we will check which shape gives the best fit. Finally, we have one extra optimization step where all the parameters are varied in a small interval around their optimized values (from the four steps given above).

To see the effect of different optimization steps, we will use two measurements: one of the heating/cooling of the CIG stack and one of the simulated measurement set (see Figures 5 and 6 in Section 2.2.2). We take two measurements with the most components, since this is the most difficult case to optimize. From the heating/cooling measurement series we take the sixteenth measurement (abbreviation: HC_{16}) and from the simulated series we take the seventeenth measurement (abbreviation: Sim_{17}). The diffraction patterns after the preliminary data analysis (without subtraction of the background intensity) are shown in Figures 31 and Figure 32 respectively. Furthermore, we need a measure to qualify the different fits. For this we will use the total residual factor. This we will discuss first and then the optimization of the different parameters will be analyzed.

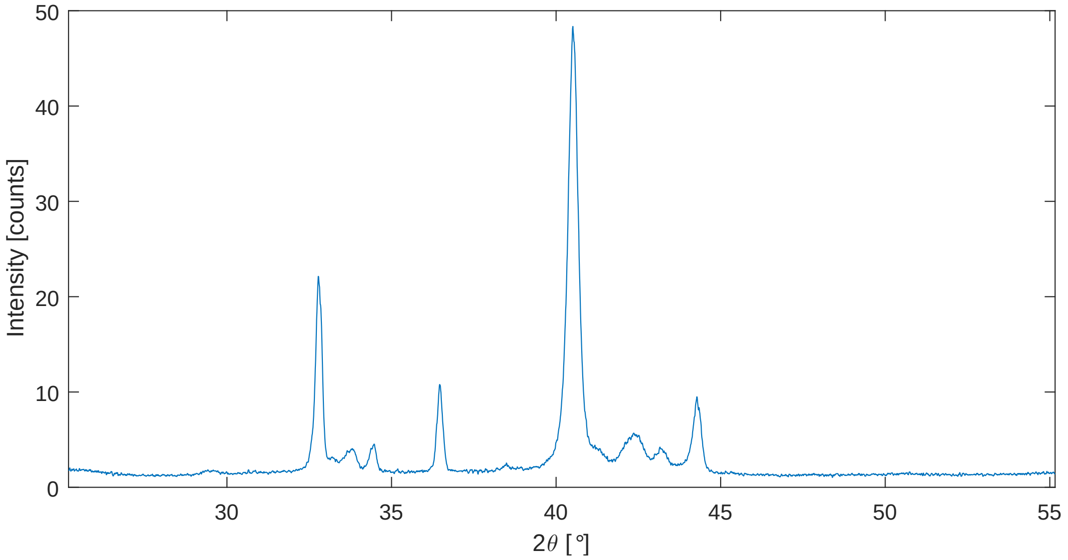


Figure 31: Sixteenth measurement of the heating/cooling series.

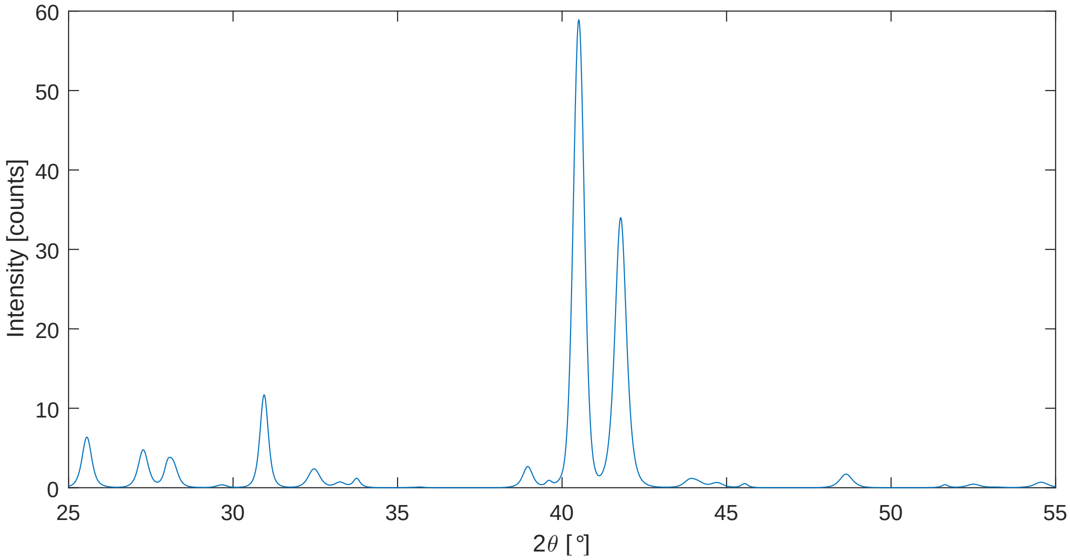


Figure 32: Seventeenth measurement from the simulated series.

7.1 Quality of fit

To compare the measured diffraction pattern with the fitted function, several measures can be used [31,27]. We choose, however, to use only the most general one that compares the entire function with the measurements. Other measures can be used to compare only the profile function or even only the structure factor. These might be useful when one is interested in solving the crystal structure from a diffraction pattern. The total residual factor is given by:

$$R_t = \frac{\sum_i |I_i^{obs} - I_i^{calc} - I_i^{bg}|}{\sum_i I_i^{obs}} \cdot 100\%,$$

where I^{obs} are the measured intensities at diffraction angle $2\theta_i$, I^{calc} is the fitted function via the Rietveld method and I^{bg} is the background intensity. The lower the R_t value is, the closer the fit is to the measured pattern. To give an indication, the total residual factor after each optimization step is shown in Table 5.

Table 5: Total residual factor after each optimization step

Optimization step	HC_{16} R_t [%]	Sim_{17} R_t [%]
Scale and Shift	18.68	41.43
Preferred orientation	14.54	27.75
Lattice Parameters	12.81	27.59
Peak shape	10.50	19.06
All	7.52	4.83

The final residual for the simulated measurement is lower, because we did not add a background signal or noise to it. Therefore, it is theoretically possible to get a perfect fit.

7.2 Scale and Shift

The first and most crucial parameters, as said before, are the scale and shift. The scale is an general scaling factor and the base shift is a result of internal material stresses affecting the crystal lattice distances and thus the XRD spectrum. The base shift may also account for uncorrected thermal expansion effects, since the thermal expansion also causes a shift.
To optimize the scale and shift we use the following function:

$$\Omega(S,\delta) = \sum_i \left(I_i^{obs} - I_i^{calc}(S,\delta) - I_i^{bg} \right)^2 .$$

Here $I_i^{calc}(S,\delta)$ is the resulting function of the Rietveld refinement with only the scale and shift as variables. For the other variables we take arbitrary values, i.e. the preferred orientation factor $G = 1$ and the Pearson-VII peak shape with $[U, V, W, \zeta_1, \zeta_2, \zeta_3] = [-0.01, 0.2, 0.01, 5, 0.5, 0.5]$. Instead of trying to fit the exact scaling factor, which ranges from 10^{-6} to 10^{-12}, we fit $\widetilde{S} = -\log_{10}(S)$, which ranges from 6 to 10. This is easier since you do not have to work with very small numbers and pay attention to the minimum step size in the algorithms. We set the bounds for the shift δ to $[-0.5°, 0.5°]$. Now, we minimize Ω for $\widetilde{S}$ and δ. The results in terms of total residual and computation time are shown in Table 6.

Table 6: Results of different optimization methods for the scale and shift parameters

	HC_{16}		Sim_{17}	
Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
BF2_{50}^{2}	18.68	110.60	41.73	114.14
Interior-point	18.68	5.72	44.89	6.78
OptQuest NLP	18.68	208.45	41.43	212.43
GA	18.68	184.79	41.43	150.13

The residual factor for all four optimization methods is approximately equal, except for the interior-point algorithm for Sim_{17}. The interior-point algorithm is a local search method, therefore, the global optimum is not always found. Thus, despite the fact that the interior-point algorithm is very fast compared to the other algorithms, we will not use it. Instead, we will use the brute-force method, since it finds the global optimum and is the fastest of the other algorithms. The fitted function after optimizing the scale and shift is shown in Figure 33 for HC_{16} and in Figure 34 for Sim_{17}. The reflections of the individual components are shifted in intensity to give a clearer view of the decomposition.

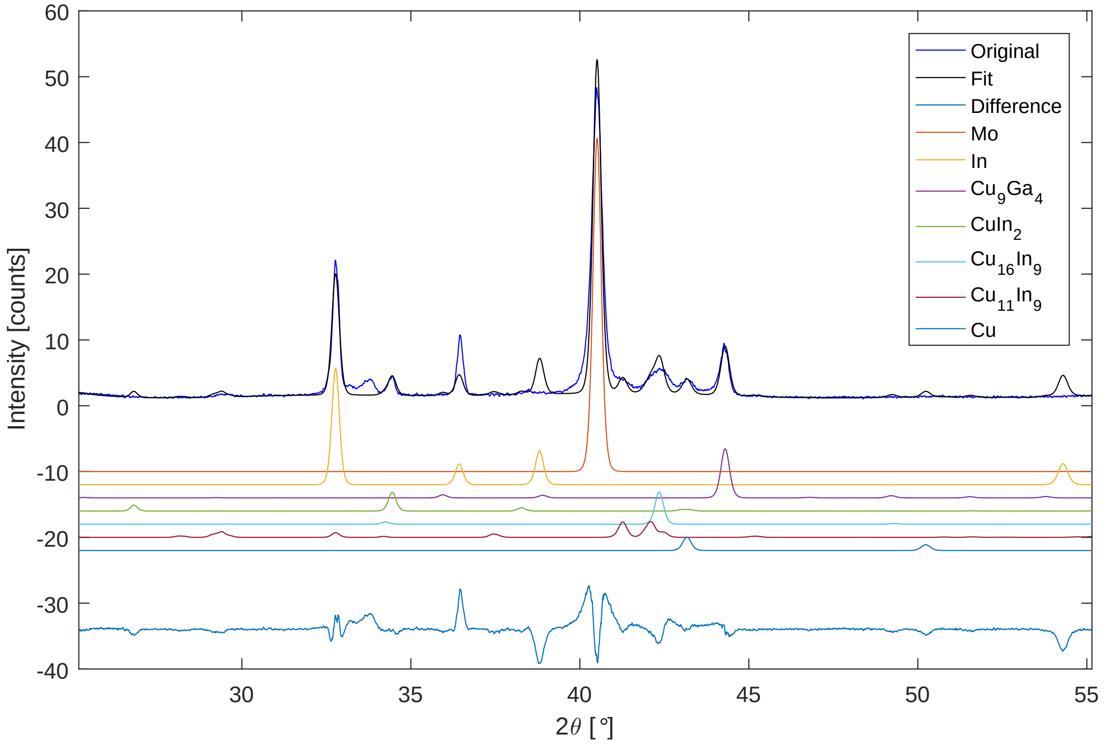


Figure 33: Fitted function after the optimization of the scale and shift parameters for HC_{16} (Brute-force method, $R_t = 18.68\%$).

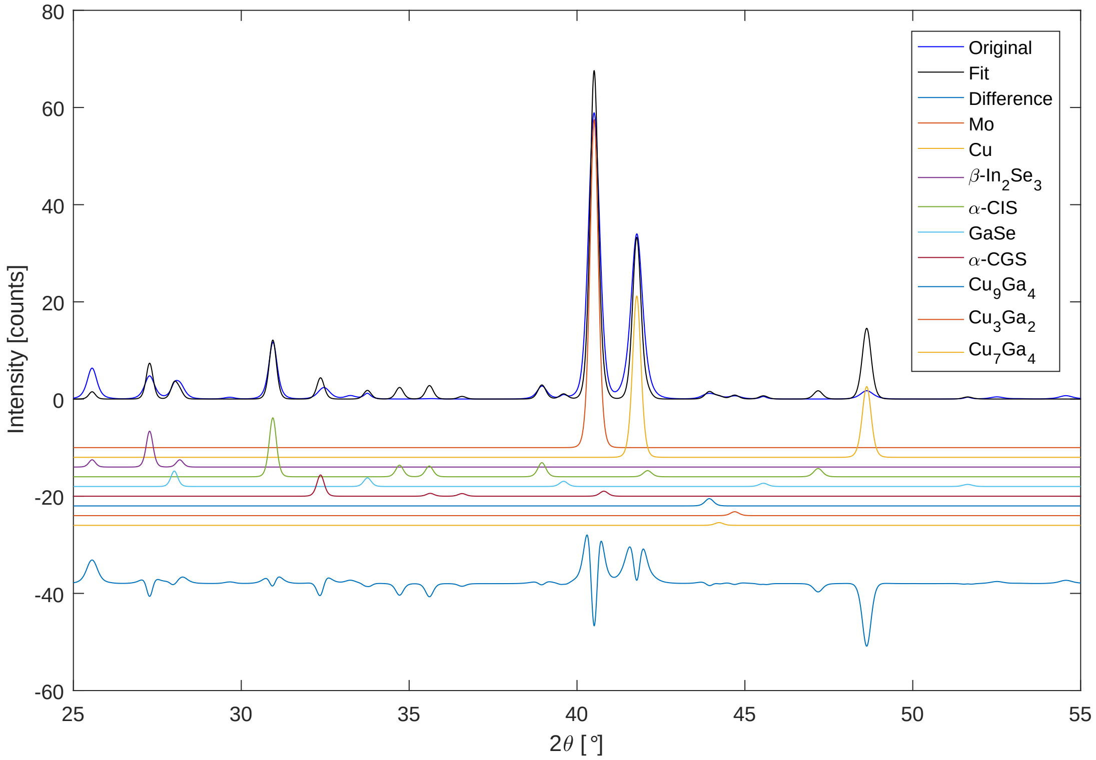


Figure 34: Fitted function after the optimization of the scale and shift parameters for Sim_{17} (Brute-force method, $R_t = 41.73\%$).

7.3 Preferred orientation

The preferred orientation only changes the relative intensities between the different peaks of a single component. Thus to optimize the preferred orientation, we do not need the entire diffraction pattern, only the position of the peaks and their corresponding intensities. This saves computing time as we can work with smaller vectors. Thus, we have to create two vectors: one vector with the specific intensities of one component in the diffraction pattern and the other vector with the theoretical intensities of the same component.

We already optimized the horizontal shift, so the position of the calculated peaks is accurate. Since the diffraction pattern also contains peaks of other components, we only want the intensities at the positions where one component theoretically has its peaks. These positions are easily calculated by Bragg's law for each component. Let us call these positions $\hat{\theta}$. Then, the corresponding intensities in the diffraction pattern are stored in the vector $Int_{\mathrm{DP}} = I^{\mathrm{obs}}(\hat{\theta} + \delta) - I^{\mathrm{bg}}(\hat{\theta} + \delta)$, where the abbreviation 'DP' stands for diffraction pattern and the positions are corrected with the earlier calculated shift. The observed intensities are discrete angles and the shift δ is (usually) an irrational number. In case the observed intensities are not defined at the position $\hat{\theta} + \delta$, we use linear interpolation of the nearby points. To avoid the scaling effect of the preferred orientation factor, we will work with the relative intensities between the peaks, thus we define: $RelInt_{\mathrm{DP}} = Int_{\mathrm{DP}} / \max(Int_{\mathrm{DP}})$.

For the calculated relative intensity, which will depend on the preferred orientation factor, we do not need the entire function of the Rietveld method. We only need the factors that influence the relative intensities, thus the multiplication factor m, the structure factor F, the Lorentz polarization factor LP and the preferred orientation $\mathcal{P}$, with the variable G. The vector becomes: $Int_{RR}(G) = m \cdot |F|^2 \cdot LP \cdot \mathcal{P}(G)$. Here the abbreviation 'RR' stands for Rietveld refinement. Then again, the relative intensity is $RelInt_{\mathrm{RR}}(G) = Int_{\mathrm{RR}}(G) / \max(Int_{\mathrm{RR}}(G))$. Now, we define a least squares optimization problem:

$$\Omega(G) = \sum \left(RelInt_{\mathrm{DP}} - RelInt_{\mathrm{RR}}(G)\right)^2 .$$

Here, G has the same number of variables as there are preferred planes. The bounds we set on G are 0.1 as lower bound and 1 as upper bound. When G is equal to one, this has no effect on the relative intensities. If G would exceed 1, this would mean that the preferred plane is avoided, which would make no sense. Then, the preferred plane should be chosen differently. The preferred planes we use to fit G are shown in Table 7. This selection of planes is based on the planes that give the highest intensity count without a preferred orientation.

Table 7: Preferred planes of the different components in HC_{16} and Sim_{17}

Component	Preferred plane (h,k,(i),l)	Component	Preferred plane (h,k,(i),l)
Cu	(1,1,1)	Cu_7Ga_4	(3,3,0)
In	(1,0,1) and (0,0,2)	Cu_3Ga_2	(3,3,0)
Mo	(1,1,0)	β-In_2Se_3	(0,0,0,9) and (1,0,-1,1)
Cu_9Ga_4	(3,3,0)	GaSe	(0,0,0,4) and (0,0,0,5)
$CuIn_2$	(2,1,1)	α-CIS	(0,0,3) and (0,0,4)
$Cu_{16}In_9$	(1,1,-2,0)	α-CGS	(0,0,3) and (0,0,4)
$Cu_{11}In_9$	(0,2,0) and (4,0,2)		

A full list of preferred planes for each components is presented in Appendix B.2. Now, we optimize the preferred orientation factor with the different optimization methods. The results are shown in Table 8. Note that the computing times in Table 8 and all similar tables that will follow, are the computing times of this specific optimization step, not from all the optimization steps.

Table 8: Results of different optimization methods for the preferred orientation factor

	HC_{16}		Sim_{17}	
Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
$BF1^2_{50}$ or $BF2^2_{50}$	14.58	26.72	27.73	34.03
Interior-point	14.54	1.41	27.75	3.72
OptQuest NLP	14.57	55.96	27.75	99.94
GA	14.59	35.16	27.75	73.65

The interior-point algorithm is the fastest and also performs equally good compared to the global search methods. To make sure we can indeed use a local search method, we look at Ω function of G. As example, we take the optimization of the function for indium in HC_{16}. Indium has two preferred planes, thus there are also two variables G_1 and G_2. The value of Ω is plotted for the range of G_1, G_2 in Figure 35.

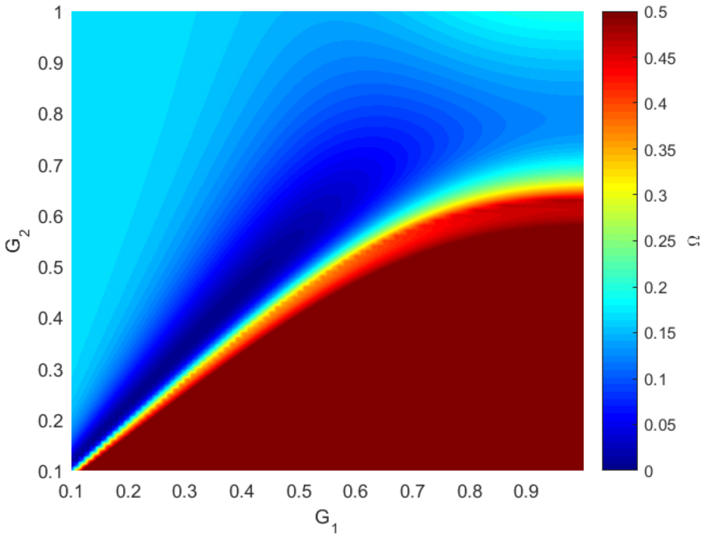


Figure 35: Ω for the preferred orientation parameters G_1 and G_2 for indium in HC_{16}.

Here, the entire red plane in the south east corner is not flat, but monotonically increasing, i.e. the values of Ω are (increasingly) greater than 0.5. Since this is not the region of interest, we thresholded this value and gave the lower values more distinct colors. In Figure 35, we see that there is a single broad area where Ω is minimized. This indicates that the area of attraction is large in this case. Furthermore, we can see that there is only one minimum in this optimization problem. Therefore, we assume it is safe to use the interior-point algorithm.

Since we compare the relative intensities, the scaling might be completely off after the optimization of the preferred orientation factors. So, we do an additional scale optimization afterwards. Since the Rietveld refinement is a linear function in terms of the scaling factor, we can use the exact solution to the least squares problem from Section 6.1. This runs in negligible time (< 0.05 seconds per component). The fitted functions after the optimization of the preferred orientation factor and additional scaling are shown in Figure 36 for HC_{16} and in Figure 37 for Sim_{17}.

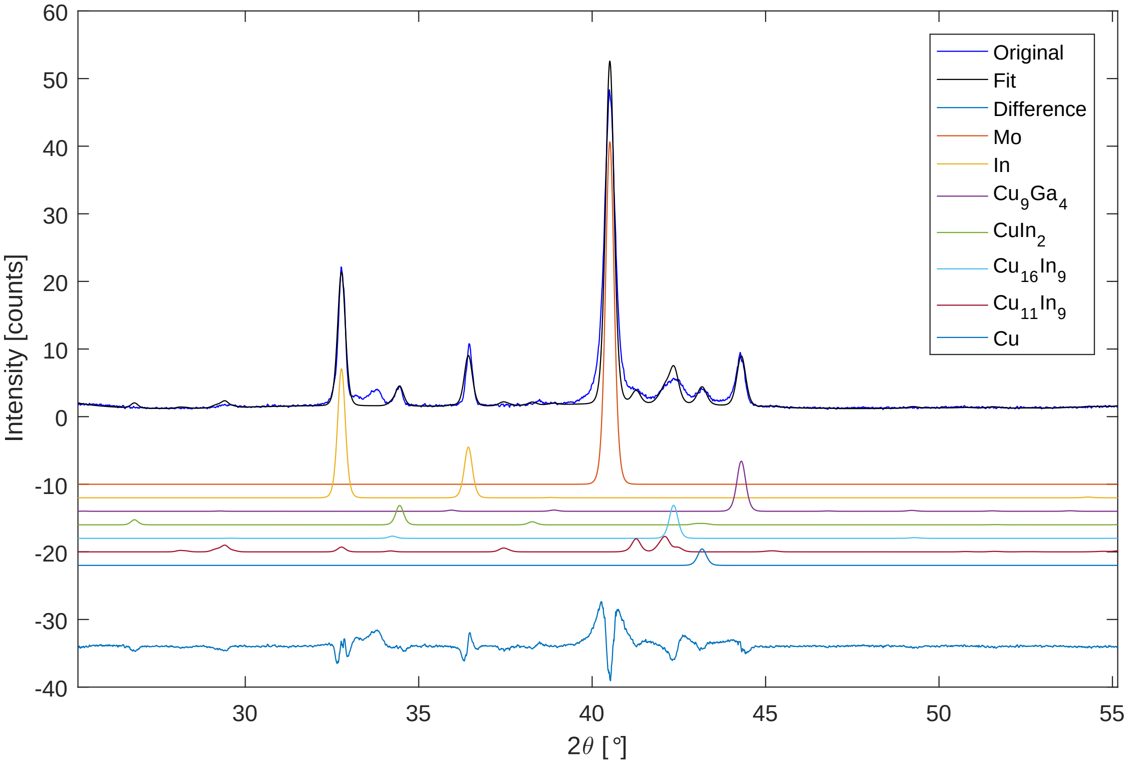


Figure 36: Fitted function after the optimization of the preferred orientation factor and the scale parameter for HC_{16} (Interior-point method, $R_t = 14.54\%$).

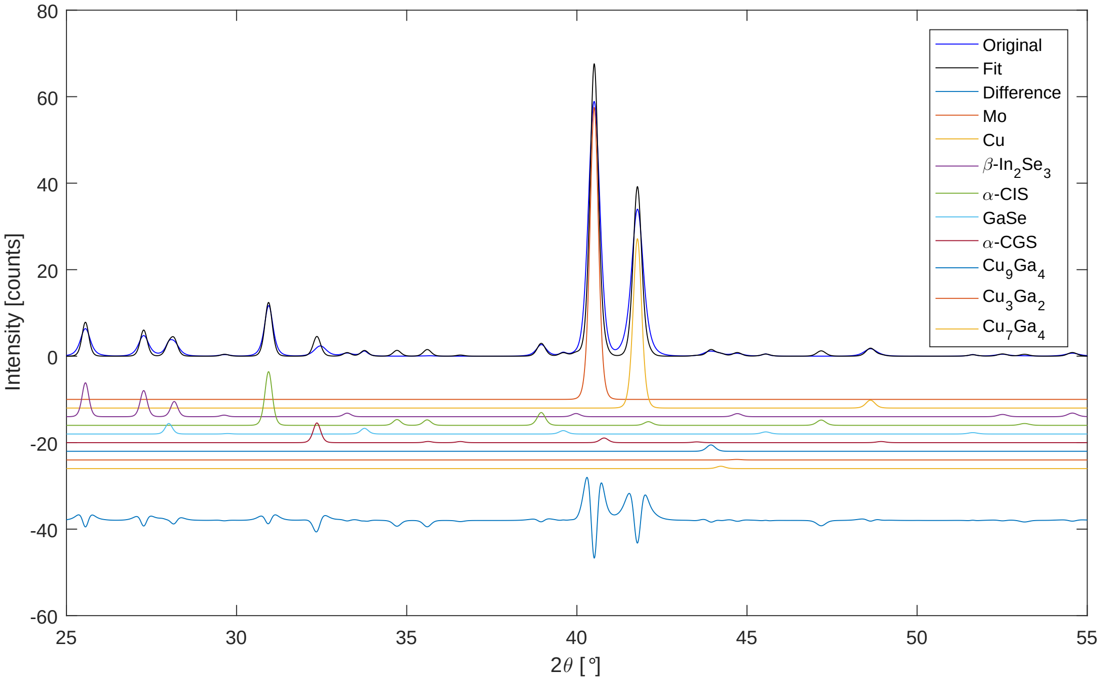


Figure 37: Fitted function after the optimization of the preferred orientation factor and the scale parameter for Sim_{17} (Interior-point method, $R_t = 27.75\%$).

7.4 Lattice parameters

The optimization of the lattice parameters is not always necessary. When a component has multiple peaks and the positions of the calculated peaks match with the positions in the diffraction pattern, then all is fine. However, for components that expand rigorously during heating, like indium, the correction for the thermal expansion we performed in the preliminary data analysis is not always correct. For example, consider the seventh measurement of the heating/cooling of the CIG stack (HC_7). After optimization of the scale, shift and preferred orientation, we obtained the fit that is shown in Figure 38.

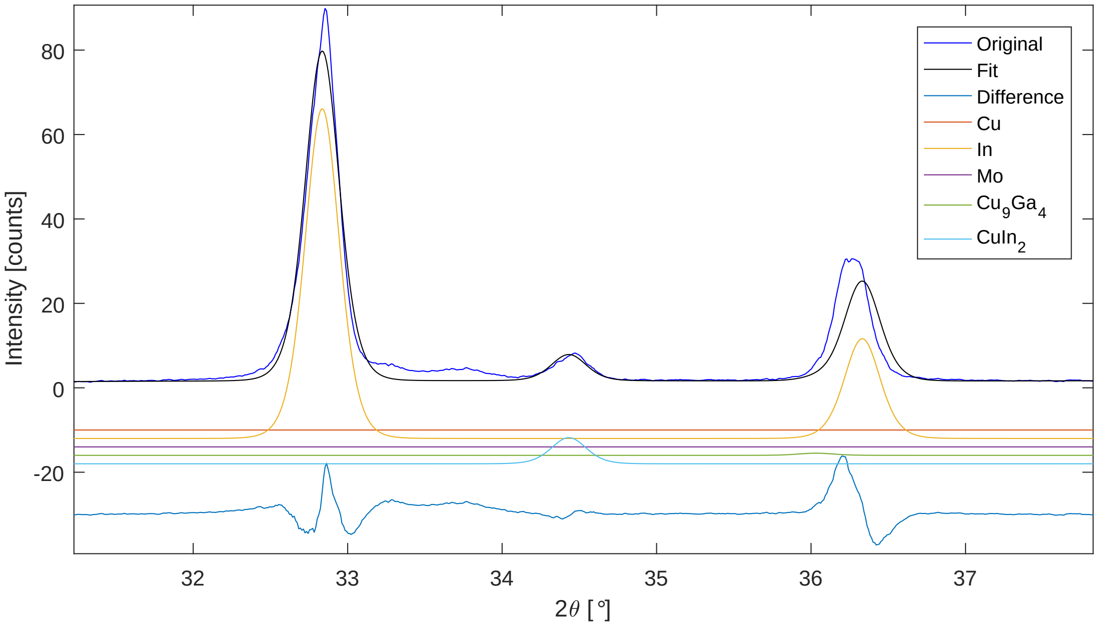


Figure 38: Error due to incorrect lattice parameters for indium.

One of the indium peaks is at the correct position and the other peak is off. This can be corrected by changing the lattice parameters. Thus, we define a least squares optimization problem:

$$\Omega(a,b,c,\alpha,\beta,\gamma) = \sum_i \left(I_i^{obs} - I_i^{calc}(a,b,c,\alpha,\beta,\gamma) - I_i^{bg} \right)^2 .$$

Note that I_i^{calc} represents the Rietveld refinement, which is not directly a function of the lattice parameters. However, the lattice parameters are used to calculate the positions of the peaks. These positions then influence, among others, the Lorentz polarization factor and the structure factor. Furthermore, not all six lattice parameters have to be optimized. Indium has a tetragonal structure, so there are only two varying parameters ($a = b \neq c, \alpha = \beta = \gamma = 90°$). We set a range of 1% in which the parameters can vary. This should be enough to correct the errors, but not give the optimization process the ability to shift a peak to match an entirely different peak. Next, we optimize the lattice parameters for HC_7 with the different optimization methods. The results are shown in Table 9.

Table 9: Results of different optimization methods for the lattice parameters

Method	R_t [%]	Computing time [s]
$BF1^2_{50}$ or $BF2^2_{50}$	13.60	31.36
Interior-point	13.60	1.45
OptQuest NLP	13.60	45.53
GA	13.59	119.31

In this table we see that the interior-point algorithm is the fastest and gives the same results as the other algorithms. Therefore, we will use the interior-point algorithm for the optimization of the lattice parameters. In Figure 38 it can be seen that only a shift of the right indium peak is not enough, the scaling is also slightly incorrect. Thus, we have to re-optimize the preferred orientation factor and afterwards the scaling factor. We set a small range of 0.1 in which the preferred orientation factor can vary and a factor ±1 for the negative logarithm of the scaling factor. The final result of the optimization of the lattice parameters for HC_7 is shown in Figure 39.

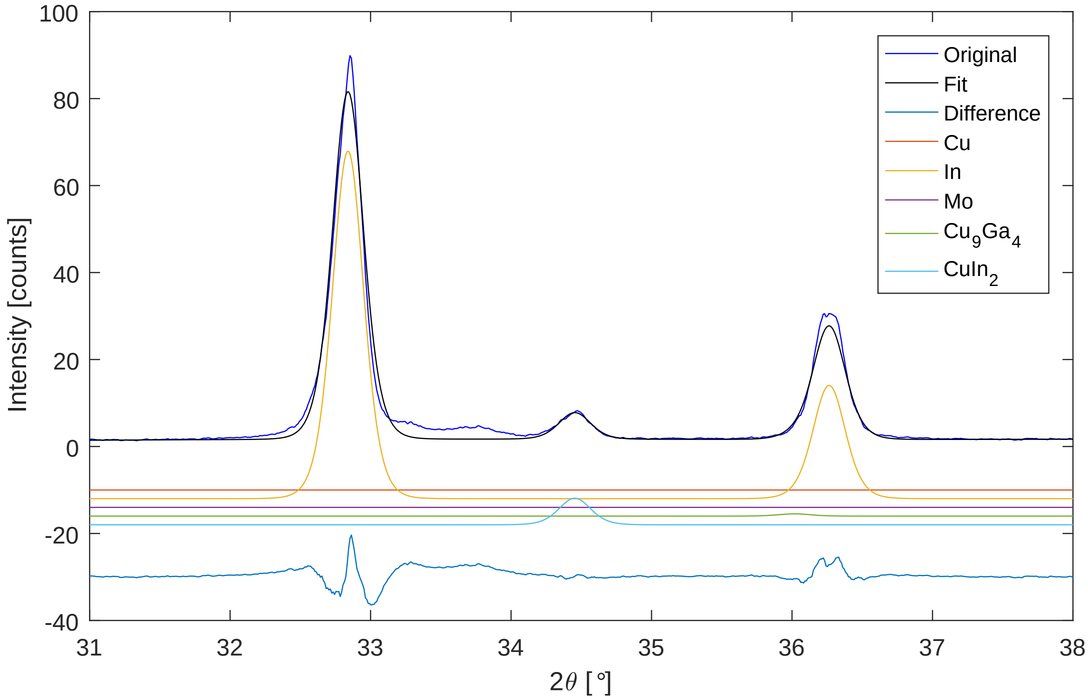


Figure 39: Correction for the error due to incorrect lattice parameters for indium (Interior-point method, $R_t = 13.60\%$).

7.5 Peak shape

For the peak shape we had four different shapes: Gaussian, Lorentzian, Pseudo-Voigt and Pearson-VII. First, we will decide which shape is the best to use based on the residual factor R_t. Next, we analyze the different optimization methods for the peak shape parameters.
For each peak shape function, we define a least squares optimization function:

$$\begin{aligned}
&\text{Gaussian:} && \Omega_1(U,V,W) &&= \sum_i \left(I_i^{obs} - I_i^{calc}(U,V,W) - I_i^{bg}\right)^2, \\
&\text{Lorentzian:} && \Omega_2(X,Y) &&= \sum_i \left(I_i^{obs} - I_i^{calc}(X,Y) - I_i^{bg}\right)^2, \\
&\text{Pseudo-Voigt:} && \Omega_3(U,V,W,X,Y,\eta_1,\eta_2,\eta_3) &&= \sum_i \left(I_i^{obs} - I_i^{calc}(U,V,W,X,Y,\eta_1,\eta_2,\eta_3) - I_i^{bg}\right)^2, \\
&\text{Pearson-VII:} && \Omega_4(U,V,W,\zeta_1,\zeta_2,\zeta_3) &&= \sum_i \left(I_i^{obs} - I_i^{calc}(U,V,W,\zeta_1,\zeta_2,\zeta_3) - I_i^{bg}\right)^2.
\end{aligned}$$

To be able to minimize these functions, there are some constraints on these parameters to keep the function positive and real. These constraints involve the FWHM parameters H_G and H_L, the linear combination parameter η and the exponent ζ (see Section 5.4 for definitions). For the algorithms, we have to rewrite these constraints in the linear system of inequalities. Here, follow the constraints for the different functions and the derivation for the linear system of inequalities.

- For the Gaussian, Pseudo-Voigt and Pearson-VII functions:

$$H_{G,j} = \sqrt{U\tan^2(\theta_j) + V\tan(\theta_j) + W}, \text{ with } H_{G,j} > 0.$$

Here θ_j corresponds to angle of a reflection. Since the FWHM has to be larger than zero, so has to be the argument of the square root:

$$U\tan^2(\theta_j) + V\tan(\theta_j) + W > 0 \quad \forall j.$$

Unfortunately, θ_j is a variable, so this function does not help us. However, we have a certain range for θ_j, namely $10° \leq \theta_j \leq 30°$ (as the range of the diffraction angle is $20° \leq 2\theta_j \leq 60°$). $\tan(\theta_j)$ is always larger than zero for this range and U, V and W can be either negative or positive. Using the fact that $\tan(10°) = \tan(\pi/18)$ and $\tan(30°) = \tan(\pi/6)$, we get the following system of inequalities:

$$\begin{bmatrix} -\tan(\pi/18)^2 & -\tan(\pi/18) & -1 \\ -\tan(\pi/6)^2 & -\tan(\pi/6) & -1 \\ -\tan(\pi/18)^2 & -\tan(\pi/6) & -1 \\ -\tan(\pi/6)^2 & -\tan(\pi/18) & -1 \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} < \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

As we cannot use a strict inequality, we will set it lower or equal than -10^{-3}. Any FWHM smaller than this is not probable.

- For the Lorentzian and Pseudo-Voigt functions:

  $$H_{L,j} = \frac{X}{\cos(\theta_j)} + Y\tan(\theta_j), \text{ with } H_{L,j} > 0.$$

  We use a similar approach as the other FWHM value and create the system of inequalities:

  $$\begin{bmatrix} -1/\cos(\pi/18) & -\tan(\pi/18) \\ -1/\cos(\pi/6) & -\tan(\pi/6) \\ -1/\cos(\pi/18) & -\tan(\pi/6) \\ -1/\cos(\pi/6) & -\tan(\pi/18) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} < \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

  Here, we also replace the strict inequality by lower-or-equal-to -10^{-3}.

- For the Pseudo-Voigt function:

  $$\eta_j = \eta_0 + \eta_2 2\theta_j + \eta_2(2\theta_j)^2, \text{ with } 0 \leq \eta \leq 1.$$

  Since η_j has an lower and upper bound, we rewrite this constraint as two inequalities:

  $$-\eta_0 - \eta_2 2\theta_j - \eta_2(2\theta_j)^2 \leq 0, \qquad \eta_0 + \eta_2 2\theta_j + \eta_2(2\theta_j)^2 \leq 1.$$

  Then, we use the range for $2\theta_j$ and that η_1, η_2 and $\eta_3 \geq 0$ to create the following system of inequalities:

  $$\begin{bmatrix} -1 & -10 & -100 \\ 1 & 30 & 900 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- For the Pearson-VII function:

  $$\zeta_j = \zeta_0 + \zeta_2(2\theta_j)^{-1} + \zeta_2(2\theta_j)^{-2}, \text{ with } \zeta > 0.$$

  Since we have that $\zeta \geq 0.5$ (see Section 5.4), this constraint is automatically satisfied.

Now, we optimize the different peak shape functions with the OptQuest NLP algorithm. We could also have chosen the genetic algorithm,which also works fine. The results are shown in Table 10.

Table 10: Total residual factors and computing times for different peak shapes using the OptQuest NLP algorithm.

	HC_{16}		Sim_{17}	
Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
Gaussian	10.78	132.90	21.65	219.81
Lorentzian	13.83	101.75	34.44	404.18
Pseudo-Voigt	10.45	183.57	21.79	437.77
Pearson-VII	9.91	157.56	21.67	212.09

As the peak shapes in Sim_{17} are created with the Pearson-VII function, we expect that this function is also the best fit. Surprisingly, also a Gaussian and the Pseudo-Voigt function perform well. For HC_{16}, the Pearson-VII function clearly is the best fit. Therefore, we will use the Pearson-VII function as the peak shape function. Next, we compare the different optimization methods. The brute-force method cannot be used, since the Pearson-VII function has six parameters and it would be very slow compared to the other methods (as we argued in Section 6.2). The total residual factor and computing time for the other methods are shown in Table 11.

Table 11: Results of different optimization methods for the peak shape function.

	HC_{16}		Sim_{17}	
Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
Interior-point	10.50	15.42	21.67	47.24
OptQuest NLP	10.50	85.48	21.67	212.09
GA	10.48	605.12	21.67	176.31

Now, all three optimization methods give approximately the same value for R_t. Thus, we are temped to use the interior-point algorithm since it is the fastest. However, after applying this to solve all the simulated diffraction patterns, we noticed that the interior-point algorithm gave suboptimal solutions for several cases. These cases involve peaks that partial overlap. Since we optimize per component, there might be a local optimum where two peaks are fitted as one broad peak. Therefore, we will not use the interior-point algorithm, but the OptQuest NLP algorithm. This gives us more consistent answers. The fitted functions after the optimization of the peak shape are shown in Figure 40 for HC_{16} and in Figure 41 for Sim_{17}.

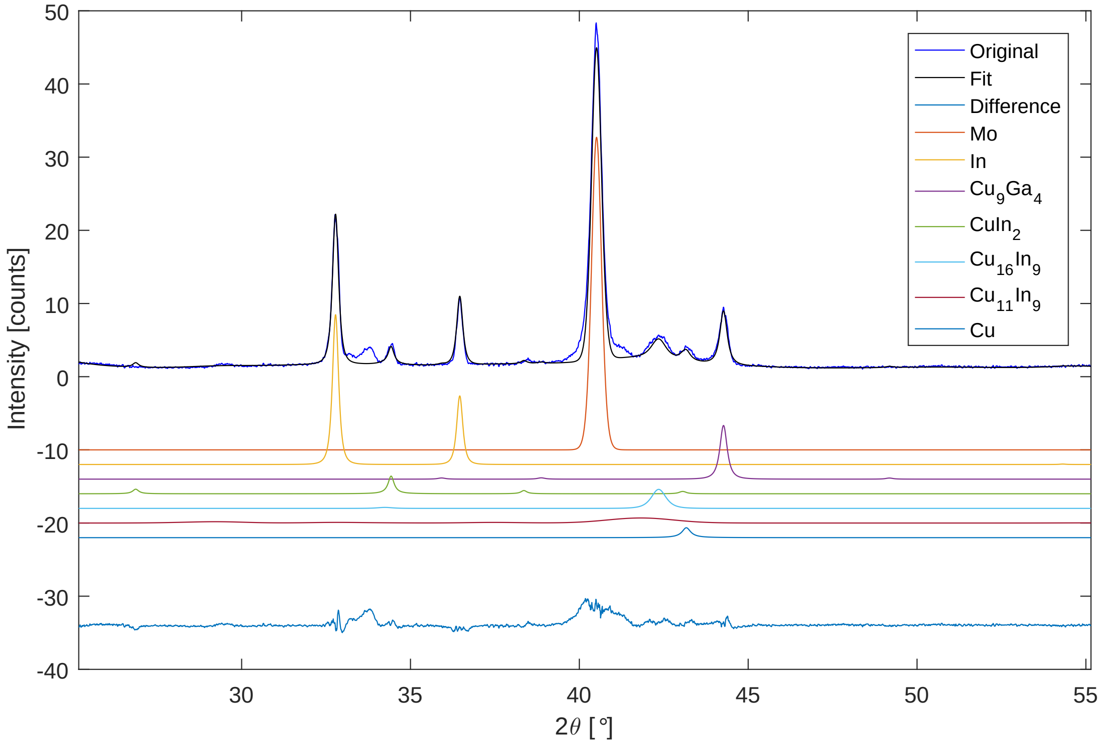


Figure 40: Fitted function after optimizing the peak shape of HC_{16} (OptQuest NLP algorithm, $R_t = 10.50\%$).

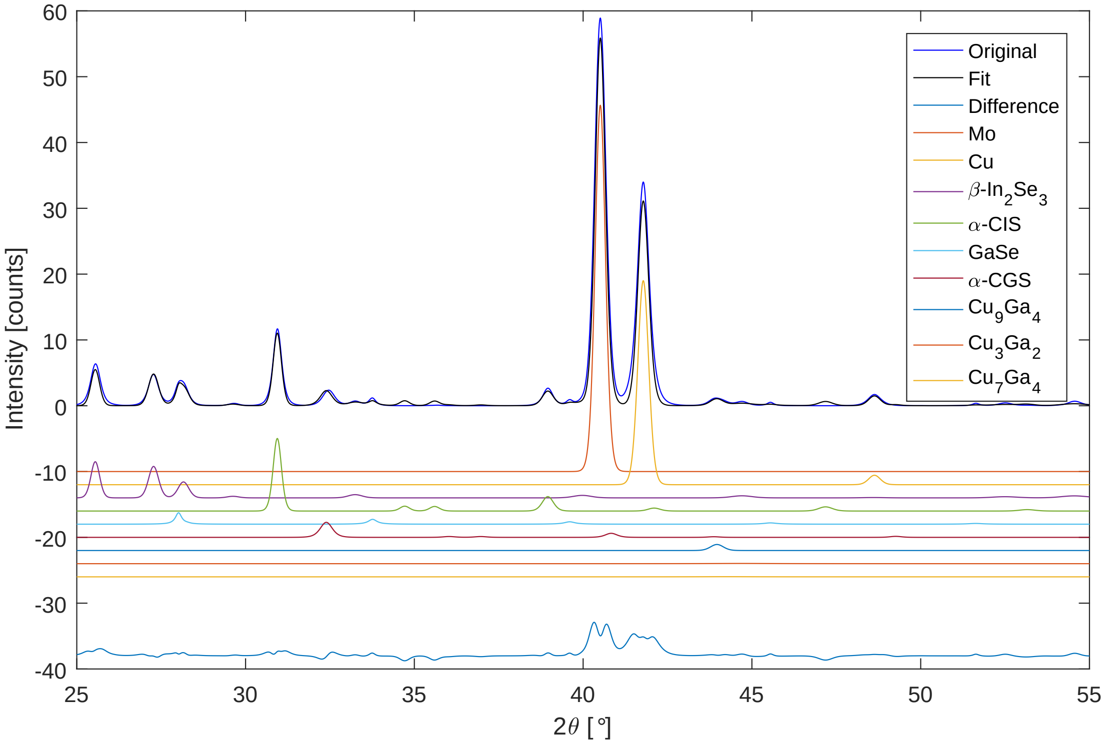


Figure 41: Fitted function after optimizing the peak shape of Sim_{17} (OptQuest NLP algorithm, $R_t = 21.67\%$).

7.6 All parameters

After performing the first four optimization steps, we have an approximation for all the parameters. Next, we perform one final optimization step where all parameters are varied in a small range of their current approximation. This does not include the lattice parameters, because they are not part of the Rietveld refinement. We assume that the lattice parameters are fitted correctly.
We define again a least squares optimization problem, now with all the parameters:

$$\Omega(\widetilde{S}, \delta, G, U, V, W, \zeta_1, \zeta_2, \zeta_3) = \sum_i \left(I_i^{obs} - I_i^{calc}(\widetilde{S}, \delta, G, U, V, W, \zeta_1, \zeta_2, \zeta_3) - I_i^{bg} \right)^2,$$

where $\widetilde{S} = -\log_{10}(S)$ and G can represent multiple variables, depending on the number of preferred planes of a component. We use the same constraints as defined earlier. For the bounds of each parameter we take either a percentage of the current solution, when a parameter is always positive, or an absolute range around the current solution, when the parameters can be both positive and negative. For example, we take the percentage range for $\widetilde{S}$: $[0.75\widetilde{S}, 1.25\widetilde{S}]$ and the absolute range for U: $[U - 0.1, U + 0.1]$. When a percentage range is taken for a negative parameter, the lower bound becomes larger than the upper bound, i.e. $l_b = 0.75 \cdot -1 = -0.75 > 1.25 \cdot -1 = -1.25 = u_b$. We could work around this with a sign function, however, we choose to use an absolute range. The results of the optimization of all parameters for the different methods is shown in Table 12.

Table 12: Results of different optimization methods for all parameters

	HC_{16}		Sim_{17}	
Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
Interior-point	7.53	14.87	4.83	29.10
OptQuest NLP	7.52	471.85	4.83	680.76
GA	7.51	728.25	5.03	1556.7

Since we cannot be sure that the interior-point algorithm always finds the global minimum, we choose to use the OptQuest NLP algorithm for the optimization of all the parameters. The resulting fitted functions are shown in Figure 42 for HC_{16} and in Figure 43 for Sim_{17}.

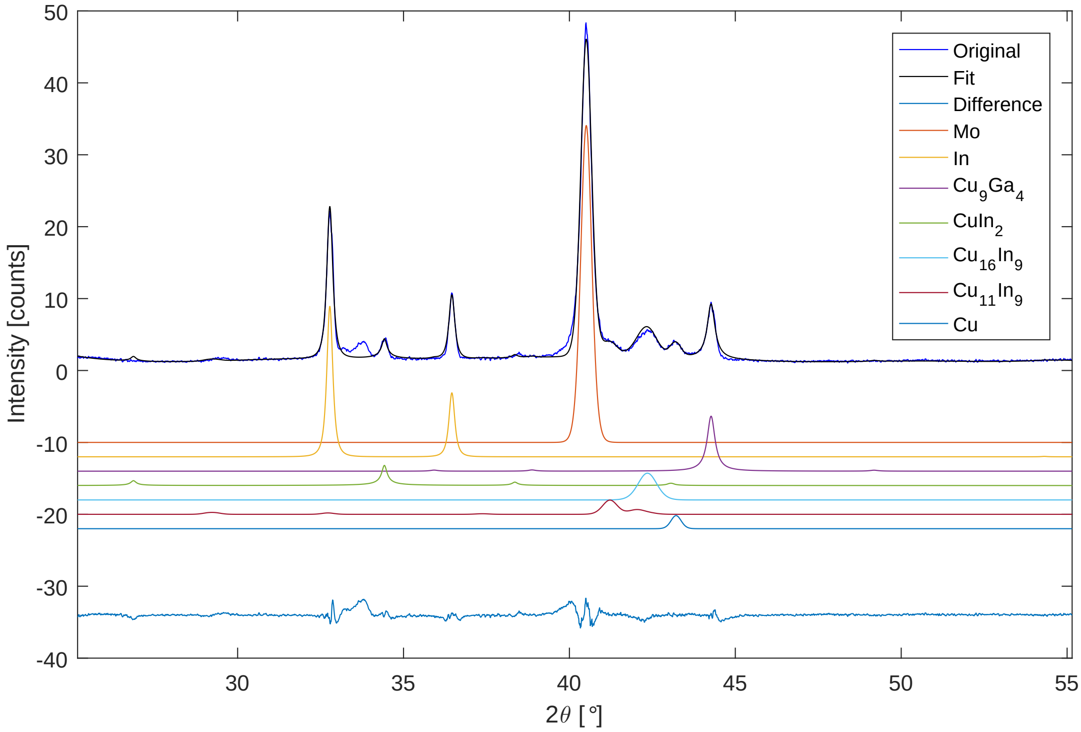


Figure 42: Fitted function after re-optimizing all parameters of HC_{16} (OptQuest NLP algorithm, $R_t = 7.52\%$).

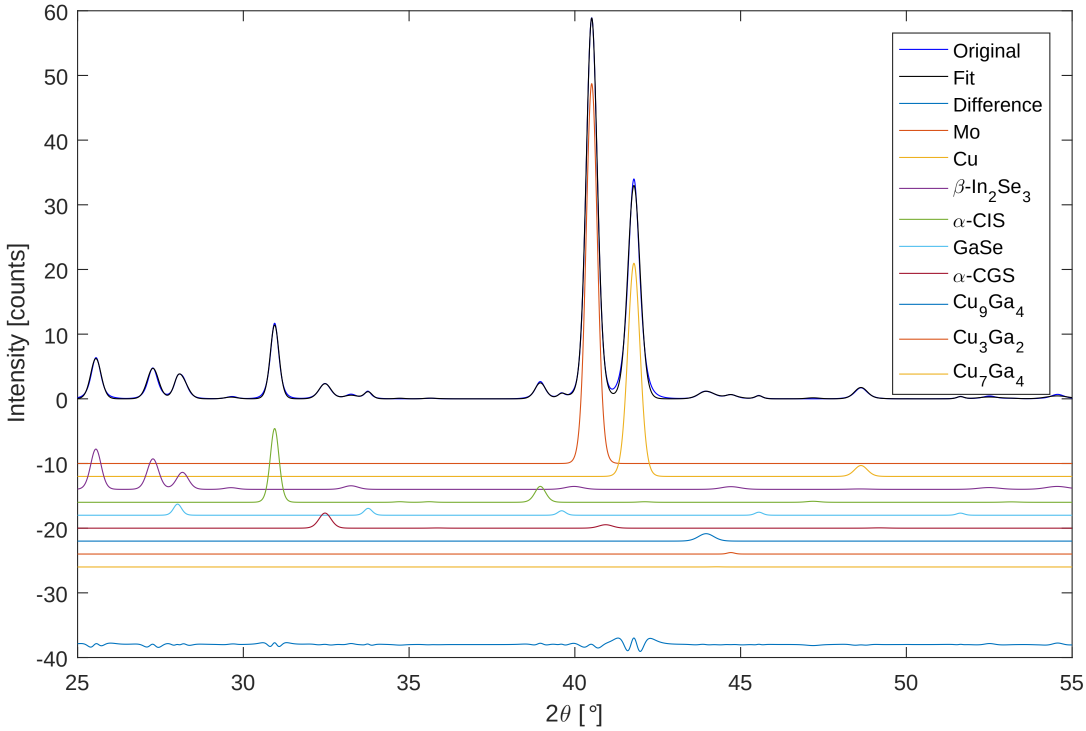


Figure 43: Fitted function after re-optimizing all parameters of Sim_{17} (OptQuest NLP algorithm, $R_t = 4.83\%$).

7.7 Summary of parameter optimization

We have analyzed all optimization steps of the parameters and concluded which optimization method perform the best for each step. A small recap is shown in Table 13.

Table 13: Results of different optimization methods for all parameters

		HC_{16}		Sim_{17}	
Parameter	Method	R_t [%]	Computing time [s]	R_t [%]	Computing time [s]
Scale & shift	Brute-force	18.68	110.60	41.43	150.13
Preferred orientation	Interior-point	14.54	1.41	27.75	3.72
Lattice parameters	Interior-point	12.81	1.68	27.59	2.12
Peak shape	OptQuest NLP	10.50	85.48	19.07	207.82
All parameters	OptQuest NLP	7.52	471.85	4.83	680.76
Total			671.02		1044.55

The total computing time is quite large, but note both measurements have been chosen with the largest number of components. The computation time depends on the number of components, but also on the number of planes each component has. This is clearly apparent in the optimization of the peak shape. For each plane, the shape has to be calculated for the entire range of the diffraction angles. Then, the reflections are summed. We already remove planes after the optimization of the preferred orientations that have a negligible contribution so save computation time. However, the calculation of the peak shape is still the slowest part of the Rietveld method and thus the bottleneck in every function evaluation.

One could also use the interior-point algorithm in the last two optimization steps to decrease the computation time significantly. However, there is a possibility that the answers are sub-optimal.

8 Quantification and reactions

Up till now we have discussed how to analyze the diffraction pattern. The optimization of all the parameters is performed and we obtained an approximation to the diffraction pattern that is decomposed in the different components. The integral of the reflection of each individual component in a diffraction pattern represents the amount of that component. This amount can be a surface concentration [mol m^{-2}], surface density [kg m^{-2}] or something similar. Here, we choose to index the integrals as surface concentrations. Each component has its own transfer rate from integral to concentration. Using the fact that the total amount of copper, indium and gallium should be constant over time, we calculate the transfer rate for all the components by defining a least squares optimization problem, such that the error in the total (constant) amounts is minimized. For the simulated measurement, we know how much selenium is absorbed by the precursor stack at each time (since we tracked it during the simulation), but for real measurements this has to be approximated.

Once we have calculated the surface concentrations for each measurement, we use a filter to smoothen the data to get rid of some small fluctuations. Then, we use the difference between two measurements to calculate the change in surface concentrations that each reaction causes. We have a list of all the reactions that can occur during the formations of CIGS (see Appendix B.3), but not all these reactions can occur at the same time. Based on the surface concentrations and the phase diagrams we can cancel out certain reactions, such that we are left with a smaller set of reactions. Next, we define a positive linear program to match the changes in surface concentration with the reactants and products of the reactions. This gives us a the reactions rates between each set of two measurements.

Finally, we combine the calculated reaction rates with the temperature and volumetric concentrations (surface concentration divided by the height of the stack) to find the frequency factor and the activation energy in the Arrhenius equation (20) (shown in the Introduction).

8.1 Indexing integrals

As said in the introduction of this chapter, the integral of the reflection of each component is a representative for the surface concentration of that component. The integrals of the individual components can be easily calculated from the diffraction patterns. Numerically, integration is the summation over the intensities at each (discrete) diffraction angle. The integrals for the heating/cooling (HC) and simulated (Sim) measurements are shown in Figure 44 and Figure 45, respectively.

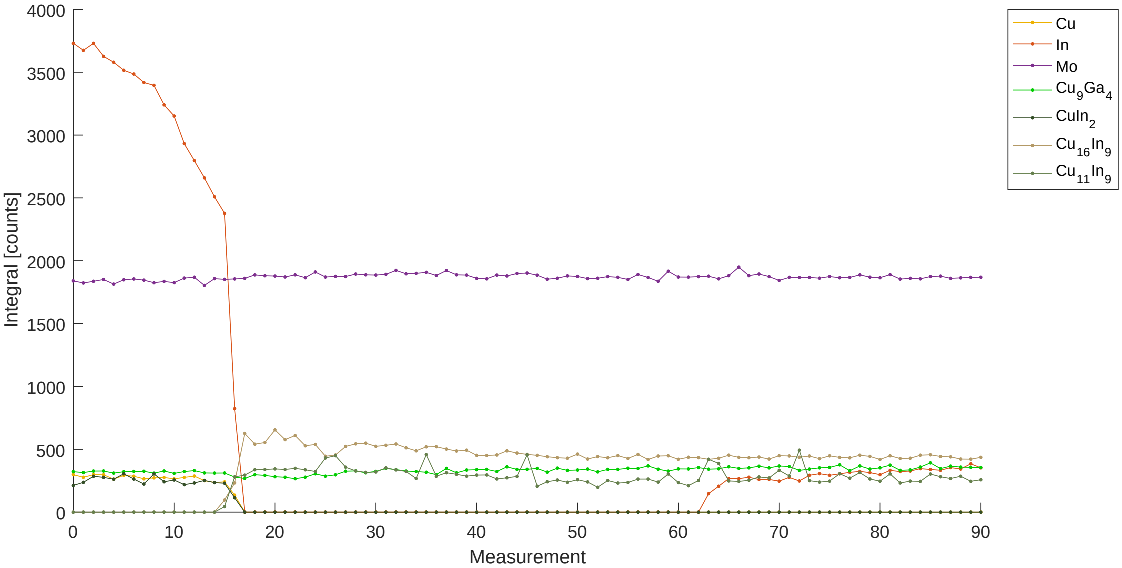


Figure 44: Integrals of the individual components in the HC measurements.

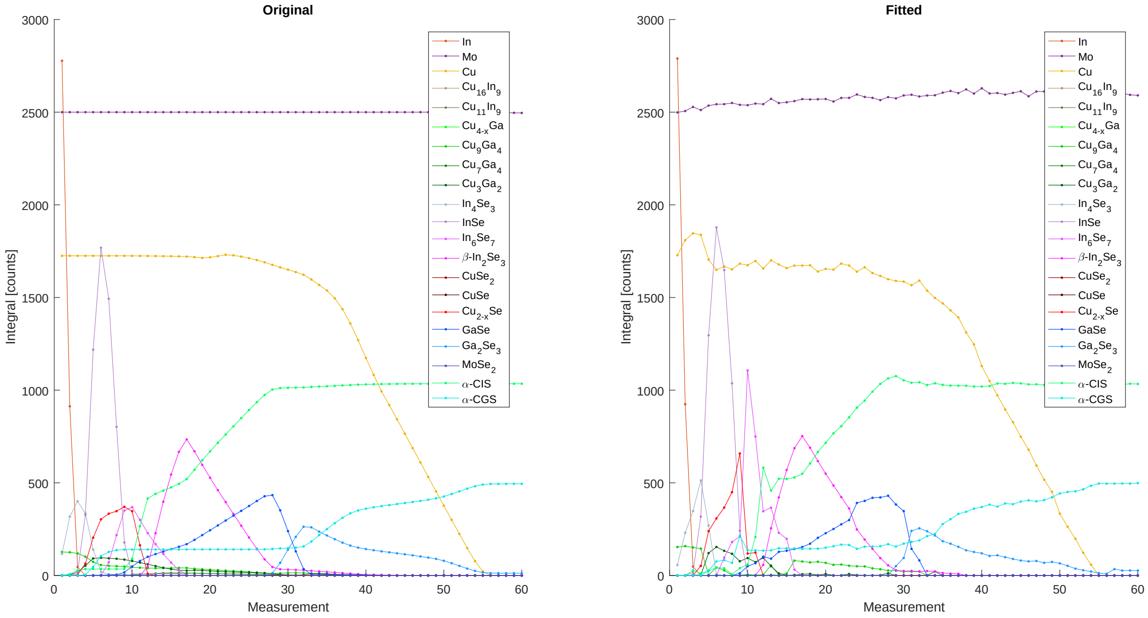


Figure 45: Original integrals (left) and the computed integrals (right) of the individual components in the simulated measurements.

These measurements still have some inconsistencies due to incorrect fits. Therefore, we apply an averaging filter to smoothen it, a similar filter as in Section 3.1. We will treat the indexing of both measurements (heating/cooling and simulated) differently, since the indexing for the heating/cooling measurement cannot be solved uniquely. Until the sixteenth measurement, much indium is disappearing and nothing reappears in return. This is mainly due to the presence of molten indium. This molten indium has no reflections, while it is an extra unknown.

For the indexing we work with transfer rates, which we call Integral-To-Mass (ITM) rates, which differs for each component Φ. Furthermore, let Int_Φ be the vector containing the values of the integrals of a component Φ for all measurements. Then, the surface concentration can be calculated via a simple multiplication: $C'_\Phi = Int_\Phi \cdot ITM_\Phi$. With the molar fractions of each component we can calculate the surface concentrations of the basic components (Cu, In, Ga, Se and Mo). The sum over all the basic components should be equal for all measurements, except for selenium.
To calculate the ITM rates, we have to solve a linear system. We expect that there is no exact solution to this linear system, so we will minimize a least squares problem. First, we set up the linear system. The indexes we will use are:

$$\begin{aligned} \text{Components} : \Phi &= 1, \dots, N \\ \text{Basic components} : i &= 1, \dots, 5 \\ \text{Measurements} : k &= 1, \dots, n. \end{aligned}$$

Then, we define the matrix with all the integrals of each component $\mathbf{Int}$ and the vector of all the ITM rates $\mathbf{ITM}$:

$$\begin{aligned} \mathbf{Int} &= [Int_1, Int_2, \dots, Int_N] \in \mathbb{R}^{n \times N}, \\ \mathbf{ITM} &= [ITM_1, ITM_2, \dots, ITM_N]^T \in \mathbb{R}^{N \times 1}. \end{aligned}$$

Next, we define $MP \in \mathbb{R}^{5 \times N}$ as the mole percentage matrix. This matrix consists of the mole percentages of Cu, In, Ga, Se and Mo in all other components. For example, $Cu_{16}In_9$ had a 16/25 mole percentage of Cu and a 9/25 mole percentage of In. To combine MP with $\mathbf{Int}$, we create a three dimensional matrix $Q \in \mathbb{R}^{5 \times N \times n}$. An element in Q is defined as $Q_{i\Phi k} = MP_{i\Phi} \cdot Int_{k\Phi}$, where i is the index for the basic components, Φ is the index for any component and k is the index for the measurement number. Finally, let $\mathbf{C}'_b \in \mathbb{R}^{5 \times n}$ be the matrix with the surface concentrations of the basic component for each measurement. The surface concentrations $\mathbf{C}'_b$ are calculated by $Q \cdot \mathbf{ITM}$, where the multiplication is along the second dimension and then the matrix is reshaped as $\mathbb{R}^{5\times 1\times n}$ to $\mathbb{R}^{5\times n}$. This surface concentration matrix can be compared with initial surface concentrations $\mathbf{C}_b^{`,\text{init}}$ in a least squares minimization problem (16):

$$\Omega = \sum_{k=1}^{n}\sum_{i=1}^{5}\mathbf{w}\left(\mathbf{C}_b^{`,\text{init}} - \mathbf{C}_b^{'}\right)^2 = \sum_{k=1}^{n}\sum_{i=1}^{5}\mathbf{w}\left(\mathbf{C}_b^{`,\text{init}} - Q\cdot\mathbf{ITM}\right)^2, \tag{16}$$

with the weights $\mathbf{w}_{i,j} = 1/R_{t_j}$ (where R_{t_j} the total residual factor of each measurement). We use these weights, because the total residual factor is a measure for the goodness of the fit. Furthermore, we use a double sum since $\mathbf{C}_b$ is a matrix. Now, we are ready to optimize **ITM** for both measurements.

For the heating/cooling measurement, however, this method would give bad results. There is amorphous indium present that is not detected. Furthermore, it is possible that part of this indium is reacting with Cu_9Ga_4 to form $Cu_9(Ga, In)_4$ or $Cu_{16}(Ga, In)_9$ [44,45]. These two components are not incorporated in our tool yet. To get better results, these two should be added and then the diffraction pattern should be re-analyzed. Even if there was no amorphous indium, this measured set would not be very useful to analyze the reaction kinetics. The time between two measurements is twenty seconds, which is a long time. For measuring the reaction kinetics, smaller time intervals are more useful, because the different phase transactions can be seen more clearly. In this measurement indium melts at 434 K. At that point, it can be seen that the concentrations of copper, indium and $CuIn_2$ decrease, where $Cu_{16}In_9$ and $Cu_{11}In_9$ appear. However, this occurs in two measurements. Two measurements is not enough to be able to determine the reactions kinetics accurately. Therefore, we will not discuss this measurement further.

For the simulated measurement, we do not have any amorphous indium or other amorphous phases. Thus, we can solve the least squares minimization problem (16). We use a genetic algorithm to find the optimal values for **ITM** rates. This finds the optimal values in approximately 15 seconds. Since we used the same technique to create the diffraction pattern from the concentrations in the first place, we can compare the calculated **ITM** rates with the originals. The results from the least squares optimization together with the original rates are shown in Table 14. For some components that are not measured in the diffraction pattern the ITM rate cannot be optimized. The genetic algorithm gives a random value for these components. In Table 14 these values are made transparent. The column ‘Points’ represents the number of data points that are used for the fit. The maximum number of points is sixty, since there are sixty measurements.

Table 14: The calculated and original ITM rates for the simulated measurements

	$ITM\cdot 10^{-6}$ [mol m^{-2}]				$ITM\cdot 10^{-6}$ [mol m^{-2}]		
Component	Computed	Original	Points	Component	Computed	Original	Points
In	8.95	7	3	In_6Se_7	6.64	11	10
Mo	1.96	2	60	β-In_2Se_3	8.71	9	26
Cu	2.72	6	55	$CuSe_2$	16.28	8	0
$Cu_{16}In_9$	21.54	27	0	CuSe	6.06	10	0
$Cu_{11}In_9$	16.08	28	0	$Cu_{2-x}Se$	19.89	13	8
Cu4-xGa	29.22	26	0	GaSe	11.98	15	24
Gu_9Ga_4	24.35	24	32	Ga_2Se_3	13.19	12	33
Cu_7Ga_4	30.00	23	8	$MoSe_2$	13.55	8	0
Cu_3Ga_2	30.00	22	27	α-CIS	25.71	26	58
In_4Se_3	15.23	16	6	α-CGS	24.80	24	56
InSe	13.25	14	7				

The original and computed surface concentrations profiles for the simulated measurement series are shown in Figure 46. The surface concentrations are calculated by $C^{'} = Q\cdot\mathbf{ITM}$ with the optimized ITM rates.

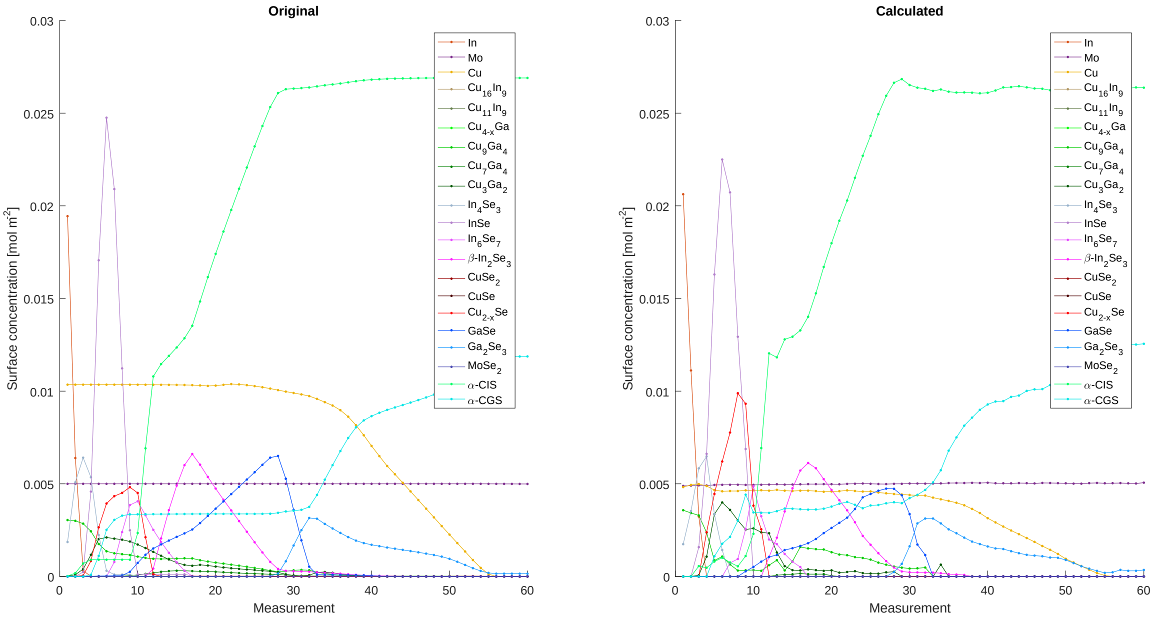


Figure 46: The original surface concentration profile (left) and the calculated surface concentration profile (right) for the simulated measurements.

The largest difference between the original and computed concentrations profiles is the concentration of copper. The computed concentration is about half the original concentration. An explanation for this is that we did not take the holes into account. Instead, we assumed that there are no holes in the components containing copper. This assumption is obviously wrong, but we do not yet have a method to calculate the number of holes. Later, when we calculate the reaction rates, we can approximate the concentration of the holes in each component. Other smaller differences between the original and the computed concentrations involve $Cu_{2-x}Se$, but this is also related to the holes, the surface concentration of Cu_9Ga_4 is too high, which also results in a lower surface concentration of copper, and the overall concentrations of GaSe and Ga_2Se_3 are slightly decreased. The In-Se phases are mostly correct, except for In_6Se_7, which has a slight peak around the tenth measurement. The computed concentrations can now be used to determine the reaction rates.

8.2 Reaction rates

We have calculated the surface concentration of each component for all measurements. With the change in surface concentration, we can find out which reactions took place and in which amounts. The set of reactions that we use is presented in Appendix B.3. First, a set of active reactions is determined by checking the position in the phase diagram of each binary system with the atomic percentages and temperature at each measurement. Next, we define a positive linear system that tries to match the stoichiometric coefficients of the reactions to the calculated surface concentrations. A positive linear system is harder to solve than a normal linear system, since all the reaction 'rates' have to be positive. To solve this, we define a linear program with the condition that the reaction 'rates' are positive. These calculated reaction 'rates' are not real rates, since they have unit [mol m^{-2}]. To convert them to reaction rates, we divide by the time between two measurements and the height of the stack. Also, to get the volumetric concentrations that are used in the Arrhenius equation, we divide the surface concentrations by the height of the stack. After we have calculated the (real) reaction rates [mol m^{-3} s^{-1}], we make one correction. Reactions that use the same reactants slow each other down, since the reactants have to be divided between the reactions. With this correction we determine the reaction rates of the individual reactions from the calculated reaction rates. As working example, we will use the original (exact) surface concentration profile. The results for the computed surface concentrations will be given in the Appendix C.

Let us define the difference in surface concentration $D^{'}$ [mol m^{-2}] of a component Φ between two measurements k and $k+1$ as:

$$D^{'}_{\Phi,k} = C^{'}_{\Phi,k+1} - C^{'}_{\Phi,k},$$

where $C^{'} = \mathbf{Int} \cdot \mathbf{ITM}$ is the matrix containing the surface concentration [mol m^{-2}] of each component at each measurement from the previous section. The difference in surface concentration $D^{'}$ for the exact concentration profile is shown in Figure 47.

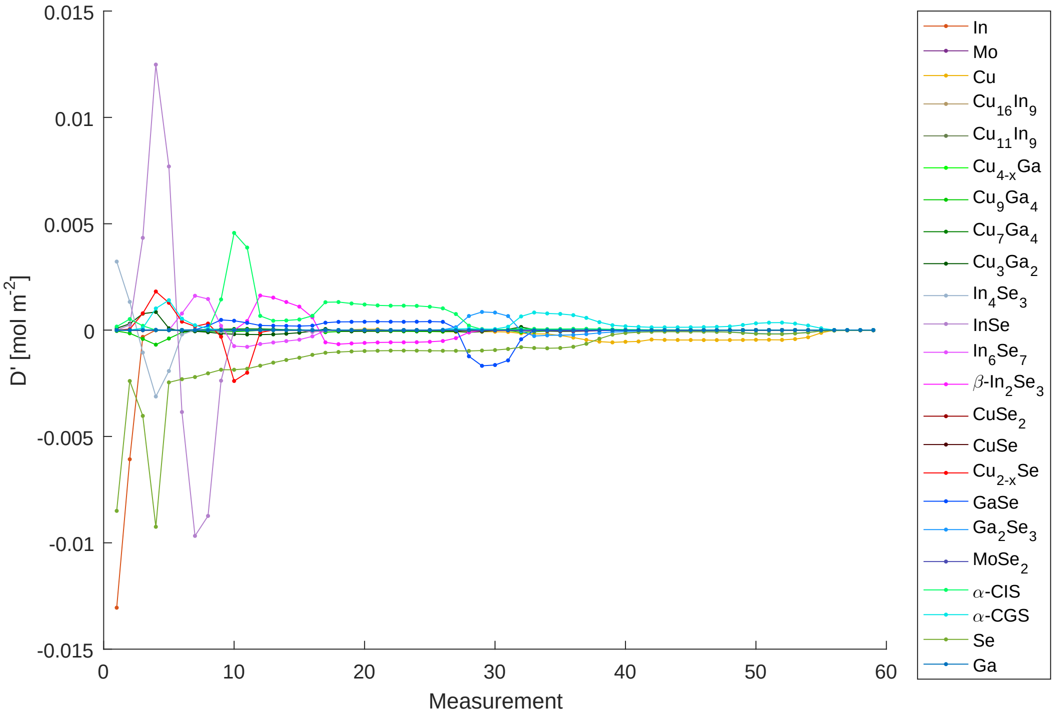


Figure 47: Difference in original surface concentrations for the simulated measurement.

We define the stoichiometric coefficients of the reactants $Reac_{\Phi,l}$ [–] and of the products $Prod_{\Phi,l}$ [–] as matrices that contain for each reaction l the stoichiometric coefficients of each component Φ. Then, we can calculate the change in surface concentration caused by the reactions $\widetilde{R}$ [mol m^{-2}] by solving the positive linear system (17):

$$(-Reac + Prod)\widetilde{R} = D^{'}, \qquad \text{with } \widetilde{R} \geq 0. \tag{17}$$

The reason why we choose for surface concentrations and not volumetric concentrations is that the a change in volumetric concentration can also be caused by the changing height of the stack. This height can change due to the absorption of selenium or the difference in density between the reactants and products. To calculate the actual reaction rate R [mol m^{-2} s^{-1}] we divide the change in surface concentration $\widetilde{R}$ by the time between two measurements t_{step} [s] and the height of the stack h_{stack} [m]: $R = \dfrac{\widetilde{R}}{t_{step} \cdot h_{stack}}$. In the simulated measurements $t_{step} = 5$s. The height of the stack is calculated by using the the molmass and density of each component. The contribution to the height of a single component is calculated by:

$$h^{\Phi}_{stack} = C^{'}_{\Phi} \cdot Molmass_{\Phi} / \rho_{\Phi}.$$

Then, summing over all components Φ gives the total height of the stack. For simplicity, let us write $RP = (-Reac + Prod)$. To solve the positive linear system (17) for $\widetilde{R}$, we cannot simply invert RP and multiply it with both sides for two reasons, because RP is in general not square, therefore not invertible. In order to solve this problem, we will convert the positive linear system to a least squares minimization problem:

$$\begin{aligned} &\min_{\widetilde{R}} \|D^{'} - RP \cdot \widetilde{R}\|_1 \\ &\text{s.t.}\, \widetilde{R} \geq 0. \end{aligned} \tag{18}$$

This is a linear program, but it is not in a standard form yet. For that, we have to get rid of the norm (absolute value) in the objective function. To get rid of the absolute value, we introduce an arbitrary parameter $\mathcal{F}$ (with the same dimensions as $\widetilde{R}$) and rewrite (18) to:

$$\begin{aligned} &\min_{\widetilde{R}} \mathcal{F} \\ &\text{s.t.}\, \mathcal{F} \leq D^{'} - RP \cdot \widetilde{R} \\ &\quad\ \mathcal{F} \geq -(D^{'} - RP \cdot \widetilde{R}) \\ &\quad\ \widetilde{R} \geq 0. \end{aligned}$$

We convert all constraints to lower-equal constraints and move all variables to the left-hand-side:

$$\begin{aligned} &\min_{\widetilde{R},\mathcal{F}} \mathcal{F} \\ &\text{s.t.}\, \mathcal{F} + RP \cdot \widetilde{R} \leq D^{'} \\ &\quad\ -\mathcal{F} - RP \cdot \widetilde{R} \leq D^{'} \\ &\quad\ \widetilde{R} \geq 0. \end{aligned}$$

Finally, we add slack variables to get equality constraint:

$$\begin{aligned} &\min_{\widetilde{R},\mathcal{F},\mathcal{S}_1,\mathcal{S}_2} \quad \mathcal{F} + \mathcal{S}_1 + \mathcal{S}_2 \\ &\text{s.t.}\, \mathcal{F} + RP \cdot \widetilde{R} + \mathcal{S}_1 = D^{'} \\ &\quad\ -\mathcal{F} - RP \cdot \widetilde{R} + \mathcal{S}_2 = D^{'} \\ &\quad\ \widetilde{R} \geq 0, \end{aligned}$$

or in matrix form:

$$\begin{aligned} &\min_{\widetilde{R},\mathcal{F},\mathcal{S}_1,\mathcal{S}_2} \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \widetilde{R} \\ \mathcal{F} \\ \mathcal{S}_1 \\ \mathcal{S}_2 \end{bmatrix} \\ &\text{s.t.} \begin{bmatrix} RP & I & I & 0 \\ RP & -I & 0 & I \end{bmatrix} \begin{bmatrix} \widetilde{R} \\ \mathcal{F} \\ \mathcal{S}_1 \\ \mathcal{S}_2 \end{bmatrix} = \begin{bmatrix} D^{'} \\ D^{'} \end{bmatrix}. \end{aligned} \tag{19}$$

In total, we introduced three new vectors of parameters ($\mathcal{F}, \mathcal{S}_1$ and $\mathcal{S}_2$), so the total number of variables in our problem has quadrupled. For the simulated measurement we use 24 components, thus in total this would give us 96 parameters. However, this is not a big issue since such linear systems can be solved quite fast, for instance, with an interior-point algorithm.

One problem, however, is that the null space of RP is nonempty. For instance, reactions from one In–Se phase to the other and back would give a nonempty null space. Both reactions can have the same reaction rate and the net difference in concentration of each component would be zero. To see this, we take the following reactions (39 and 40 from Appendix B.3):

$$\begin{array}{lll} \widetilde{R}_{39}\text{:} & 1\,\mathrm{In_4Se_3} + 1\,\mathrm{Se} & \longrightarrow & 4\,\mathrm{InSe} \\ \widetilde{R}_{40}\text{:} & 4\,\mathrm{InSe} & \longrightarrow & 1\,\mathrm{In_4Se_3} + 1\,\mathrm{Se} \end{array}$$

When $\widetilde{R}_{39} = \widetilde{R}_{40}$ and we take minus the coefficients of the reactants plus the coefficients of the products of each component, they add up to zero. Therefore, $\widetilde{R}_{39}$ and $\widetilde{R}_{40}$ could be anything as long as they are equal to each other. To solve this problem, we add an extra constraint to set one of the variables to zero when the other is nonzero:

$$\begin{aligned} \widetilde{R}_{39} &\leq \mathcal{M}y \\ \widetilde{R}_{40} &\leq \mathcal{M}(1-y), \end{aligned}$$

where $\mathcal{M}$ is a large number and $y \in \{0,1\}$ is a binary variable. As we introduce a binary variable, the problem becomes a mixed integer linear program. Whenever both of the reactions are in the active set of reactions, we add this constraint to (19). Now, we compute the change in surface concentration due to the reactions $\widetilde{R}$ [mol m^{-2}] at every measurement. These changes are shown in Figure 48. No legend is shown, because it would have approximately eighty entries and it is not possible to give all reactions a distinct color.

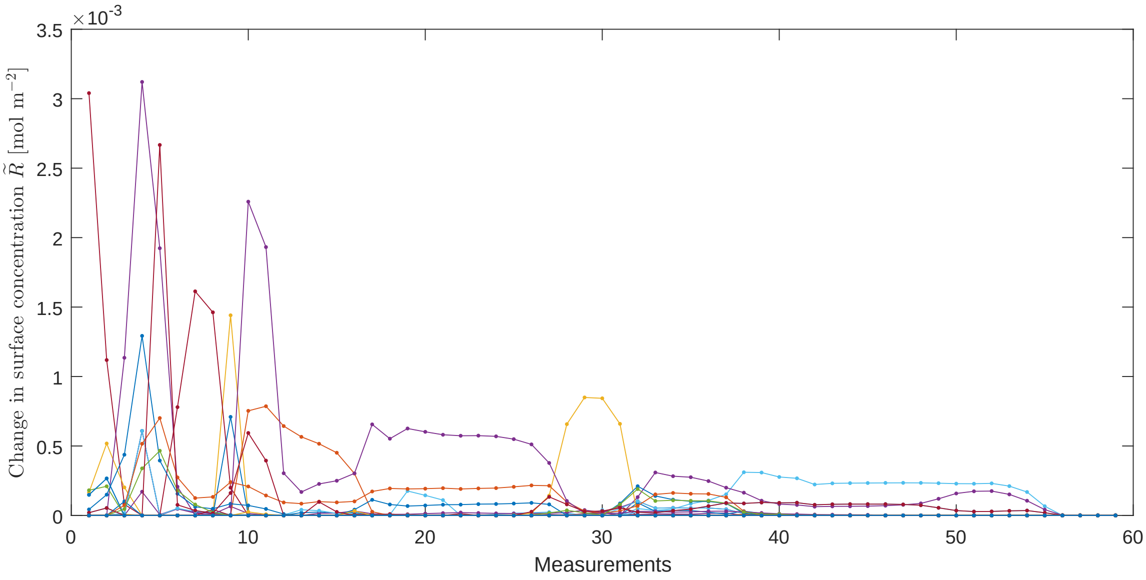


Figure 48: Changes in the exact surface concentration due to reactions of the simulated measurement series.

Even though we used the exact surface concentrations, there is still an error in some components. The absolute and relative errors for each component are shown in Figure 49. They are calculated as follows:

$$\begin{aligned} \text{Absolute error } &= D^{'} - RP \cdot \widetilde{R} \\ \text{Relative error } &= (D^{'} - RP \cdot \widetilde{R})/\|D^{'} - RP \cdot \widetilde{R}\|_1 \cdot 100\%. \end{aligned}$$

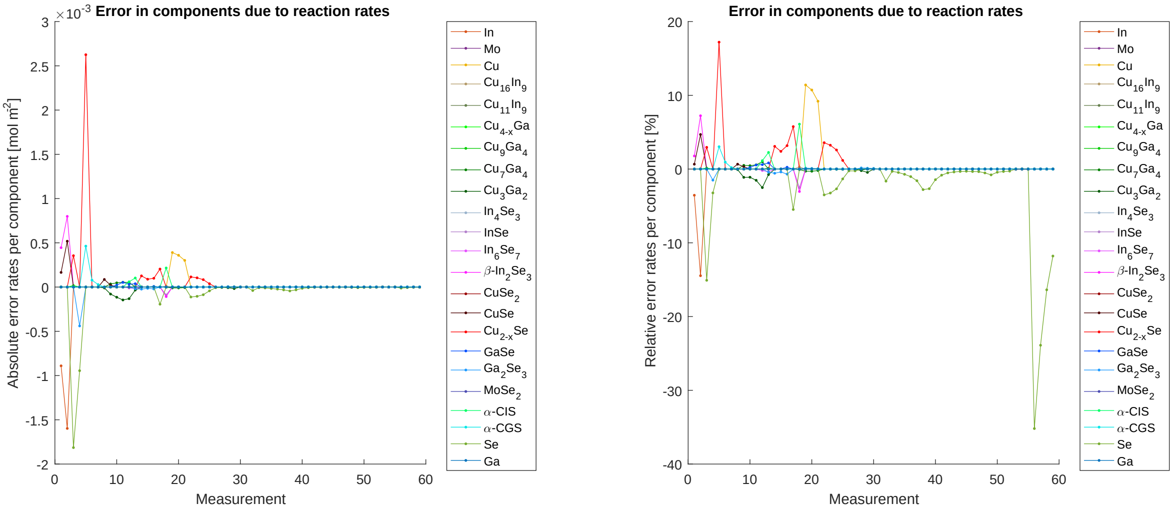


Figure 49: Absolute and relative error rates for the calculated changes in surface concentration $\widetilde{R}$ of the simulated measurement series.

The reason we give both errors is that the relative error for selenium looks quite large. However, the absolute error is very small as there is very little (free) selenium, so this is no problem. The error in selenium in the last 30 measurements is actually correct. During the simulation, there is a very small concentration of free selenium in the CIGS stack. This is due to the solubility of selenium in CIS/CGS. Selenium will still be absorbed by CIGS, even though there is no copper, indium or gallium left. Then, it will react with molybdenum. This is unwanted in solar cells as it decreases the conductivity of the molybdenum layer. The other errors are possibly because we did not incorporate copper holes in the reactions. The concentration of holes can be quite significant and should be added to improve the results.
To get the reaction rates R [mol m^{-3} s^{-1}], we divide $\widetilde{R}$ [mol m^{-2}]by the time step t_{step} [s] and the stack height h_{stack} [m]. The computed reaction rates are shown in Figure 50.

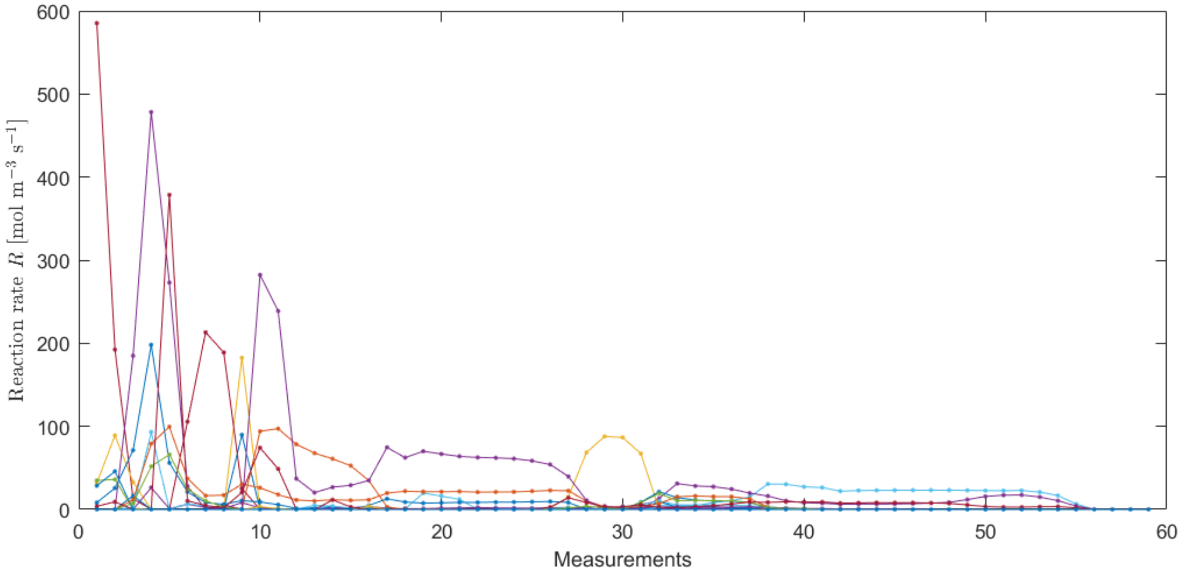


Figure 50: Changes in the exact surface concentration due to reactions of the simulated measurement series.

Correction
At the creation of the simulation model in [23], we argued that reactions that occur at the same time and use (one of) the same components slow each other down. The reason for this is that a certain amount of one component has to be divided between two (or more) reactions. Thus, the actual volumetric concentration that is used in the Arrhenius equation 20 (from the Introduction) is reduced and so the reaction rate is also reduced. The computation of the reduced reaction rates was done by multiplying the original reaction rate with a fraction. This fraction is the original reaction rate divided by the sum of all reaction rates that use one of the same components. The reduced reaction rates R are given by the function:

$$R_j = \frac{\widehat{R}_j^2}{\sum_{J_j} \widehat{R}},$$

where J_j is the set of components that have the same reactants as reaction j. During the simulation, R is calculated from $\widehat{R}$. Now, we have R and we want to calculate the original reaction rates $\widehat{R}$. Since this cannot be done directly, therefore we define $\mathcal{G}(\widehat{R}) = \dfrac{\widehat{R}^2}{\sum_{J_j} \widehat{R}} - R$ and solve $\mathcal{G}(\widehat{R}) = 0$.

The corrected reaction rates $\widehat{R}$ are shown in Figure 51.

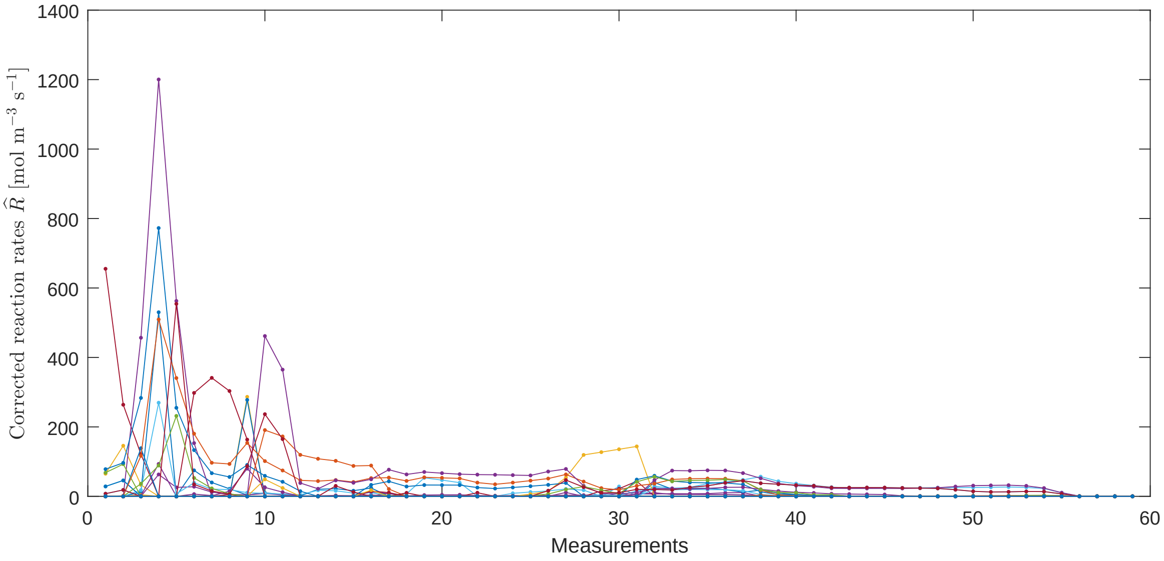


Figure 51: Corrected reaction rates of the simulated measurement series.

Comparing this figure to Figure 50, we see that all the reaction rates have increased (as expected) and the course of the reaction rates is similar. We will use these reaction rates to fit the parameters in the Arrhenius equation.

8.3 Arrhenius parameters

Once we have calculated the reaction rate for every reaction at every measurement, we can fit the Arrhenius parameters. The Arrhenius equation is:

$$\widehat{R} = k_0 e^{-\dfrac{E_a}{\mathcal{R}T}} C_1^{n_1} C_2^{n_2} \dots . \tag{20}$$

Here, $\widehat{R}$ is the corrected reaction rate [mol m^{-3} s^{-1}], k_0 is the frequency factor [mol$^{1-\sum_i n_i}$ m$^{-3+3\sum_i n_i}$ s^{-1}], E_a is the activation energy [J mol^{-1}], $\mathcal{R}$ is the gas constant [J mol^{-1} K^{-1}], T is the absolute temperature [K], C_i is the volumetric concentration of the i-th reactant [mol m^{-3}] and n_i is the corresponding reaction order [$-$]. The temperatures are known and we can calculate the volumetric concentrations by dividing the surface concentrations by the stack height:

$$C = \frac{C^{‘}}{h_{stack}}.$$

The parameters that we want to determine are the frequency factor and the activation energy. Trying to fit this function directly will not work however. There are certain reactions that occur consecutively. The volumetric concentration of an intermediate product can be (close to) zero at two measured times. For example, $Cu_{2-x}Se$ is formed from Cu and Se, but then is immediately used in the formation of CIS or CGS. The volumetric concentration of $Cu_{2-x}Se$ used in the Arrhenius equation for reaction rate of CIS/CGS is not close to zero, but there is a constant supply of $Cu_{2-x}Se$. It is difficult to say what columetric concentration for $Cu_{2-x}Se$ we should use. We will use the maximum amount that is available, meaning that we calculate how much $Cu_{2-x}Se$ can be formed in the reaction of Cu and Se and use these volumetric concentrations in the Arrhenius equation. Meanwhile, we do not change the volumetric concentrations of Cu and Se, since we still need them to fit their Arrhenius parameters. Thus, we add only the products of each reaction to the volumetric concentrations:

$$\widetilde{C} = C + Prod \cdot R \cdot t_{step}.$$

Note that we did not use the corrected reaction rates, because the original reaction rates R are used between two measurements and here we work with volumetric concentrations, thus we do not multiply with h_{stack}. We use the corrected reaction rates $\widehat{R}$ to fit the Arrhenius equation. Now, we define a least squares minimization problem:

$$\Omega(k_0, Ea) = \sum_k \left(\widehat{R}(k) - k_0 e^{-\frac{E_a}{RT(k)}} \widetilde{C}_1^{n_1}(k) \widetilde{C}_2^{n_2}(k)\right)^2, \tag{21}$$

where the sum is over all measurements k where the computed reaction rate is nonzero. There is no point in trying to find the frequency factor and the activation energy if the number of data points, where the reaction rate is nonzero, is too low. The minimum number of data points that we require is set to three. After we optimized both parameters, we can calculate confidence intervals. As this is a nonlinear optimization problem, the Hessian (matrix of the second derivatives) is also needed to estimate the variance. The 95% confidence intervals are calculated as follows [46]:

$$\begin{bmatrix} k_0^* \\ Ea^* \end{bmatrix} \pm t_{0.95,v} \frac{\|Res\|}{\sqrt{v}} \left(\nabla^2 \Omega(k_0^*, Ea^*)\right)^{-1},$$

with the residual $Res = \widehat{R} - k_0 e^{-\frac{E_a}{RT}} \widetilde{C}_1^{n_1} \widetilde{C}_2^{n_2}$, the degrees of freedom v (number of data points minus the number of variables) and the value of the t-distribution $t_{0.95,v}$. The last term represents the inverse of the Hessian at the optimal value (denoted with asterisks). We optimize the frequency factor and the activation energy for the sixteen reactions that have the most data points (ranging from 9 to 59 out of 60). The optimal values, original values (used in the simulation model) and the 95% confidence intervals are shown in Table 15. The actual fits of the Arrhenius equation are shown in Figure 52. Here, the horizontal axis (when held correctly) are the data points. This gives a good view of how the fit matches at each point. In Figure 53, the horizontal axis represents the temperatures of the data points.

In Appendix C the values of the frequency factor and the activation energy are shown in Table 24 for the computed concentrations instead of the exact concentrations. Also, the fits at each point and at the temperatures are shown in Figure 62 and 63, respectively.

Table 15: Calculated frequency factor and activation energy for several reactions using the exact concentrations from the simulated measurements.

	Frequency factor k_0			Activation energy Ea			
Reaction	Calculated	Model	95%-CI	Calculated	Model	95%-CI	Data points
4. 9-x Cu + 4 Ga $\longrightarrow$ 1 Cu_9Ga_4	0.12	500	[-0.945,1.185]	70330	30000	[8029.58,132631]	32
13. 2-x Cu + 1 Se $\longrightarrow$ 1 $Cu_{2-x}Se$	6.20	4000	$[-107.11, 119.5]$	63750	60000	$[-64553.3, 192054]$	37
15. 2 Ga + 3 Se $\longrightarrow$ 1 Ga_2Se_3	$5.28 \cdot 10^{-3}$	1000	$[-7.54 \cdot 10^{-4}, 1.13 \cdot 10^{-2}]$	42771	50000	[38334.8, 47206.3]	42
18. 1 $Cu_{2-x}Se$ + 1 Ga_2Se_3 $\longrightarrow$ 2 α-CGS	$1.21 \cdot 10^{-3}$	5000	$[6.14 \cdot 10^{-4}, 1.80 \cdot 10^{-3}]$	11836	50000	[8711.53, 14960.8]	43
25. 1 Cu_7Ga_4 $\longrightarrow$ 1 Cu_9Ga_4 + 1 Ga	0.31	1000	$[-0.83, 1.45]$	10000	60000	$[-12438.8, 32438.8$	50
27. 2 Cu_3Ga_2 $\longrightarrow$ 1 Cu_7Ga_4 + 1 Ga	0.25	500	$[-0.44606, 0.95446]$	10000	55000	$[-3412.39, 23412.4]$	46
30. Cu_9Ga_4 $\longrightarrow$ 9 Cu + 4 Ga	1.18	1000	$[-428.15, 430.52]$	10000	50000	$[-2.55 \cdot 10^6, 2.57 \cdot 10^6]$	18
39. 1 In_4Se_3 + 1 Se $\longrightarrow$ 4 InSe	1.81	3500	$[-73.057, 109.27]$	50232	60000	[30728.1, 69735.5]	14
42. 6 InSe + 1 Se $\longrightarrow$ 1 In_6Se_7	0.64	4000	$[-1.3404, 2.6134]$	49121	60000	[34485.7, 63756.9]	8
44. 1 In_6Se_7 + 2 Se $\longrightarrow$ 3 β-In_2Se_3	0.20	3000	$[-1.0218, 1.4134]$	44025	50000	[8663.98, 79386.3]	8
50. 1 Cu_9Ga_4 + Se $\longrightarrow$ 1 $Cu_{2-x}Se$ + 2 Cu_3Ga_2	$3.80 \cdot 10^{-3}$	5000	$[1.32 \cdot 10^{-3}, 6.27 \cdot 10^{-3}]$	17869	60000	[15350, 20387.5]	53
51. 1 Cu_3Ga_2 + 4 Se $\longrightarrow$ 2 $Cu_{2-x}Se$ + 2 GaSe	$2.43 \cdot 10^{-3}$	5000	$[6.02 \cdot 10^{-4}, 4.26 \cdot 10^{-3}]$	17588	60000	[14459.9, 20716.2]	59
53. 1 $Cu_{2-x}Se$ + 1 β-In_2Se_3 $\longrightarrow$ 2 α-CIS	$6.22 \cdot 10^{-4}$	4600	$[-1.99 \cdot 10^{-4}, 1.44 \cdot 10^{-3}]$	10000	40000	[2417.46, 17582.5]	46
54. 1 $Cu_{2-x}Se$ + 3 GaSe $\longrightarrow$ 2 α-CGS	$2.46 \cdot 10^{-5}$	5000	$[-2.13 \cdot 10^{-4}, 7.04 \cdot 10^{-4}]$	10000	40000	$[-279.944, 20279.9]$	47
62. 1 α-CIS + 1 $Cu_{2-x}Se$ $\longrightarrow$ 1 α-CIS + 1 Se	$1.99 \cdot 10^4$	4000	$[-1.48 \cdot 10^6, 1.52 \cdot 10^6]$	150000	60000	$[-380367, 680367]$	30
63. 1 α-CGS + 1 $Cu_{2-x}Se$ $\longrightarrow$ 1 α-CGS + 1 Se	$1.54 \cdot 10^{-4}$	4000	$[-4.35 \cdot 10^{-4}, 7.44 \cdot 10^{-4}]$	10000	60000	$[-9661.61, 29661.6]$	45

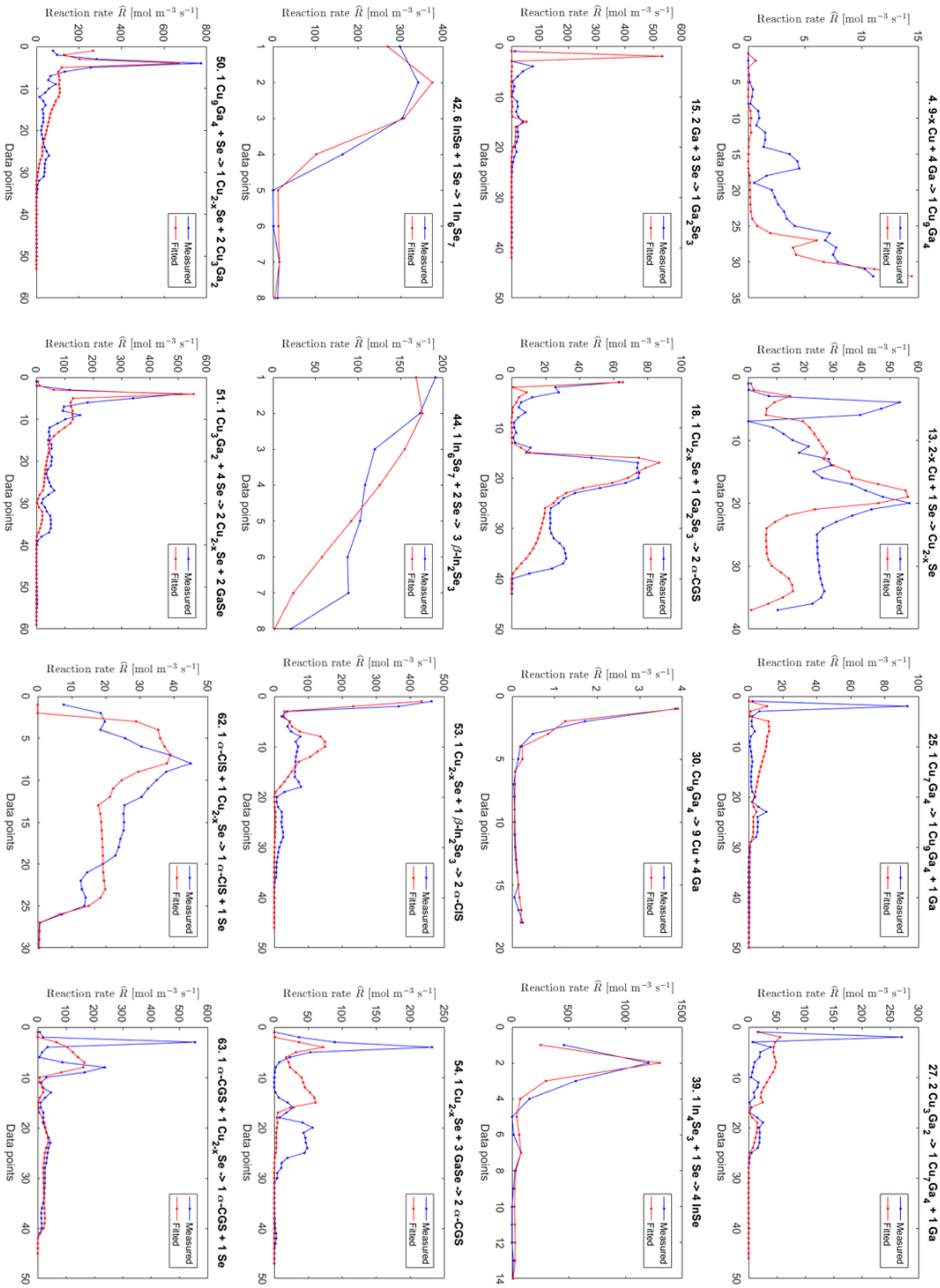


Figure 52: Fitted Arrhenius equations at each data point for a set of reactions.

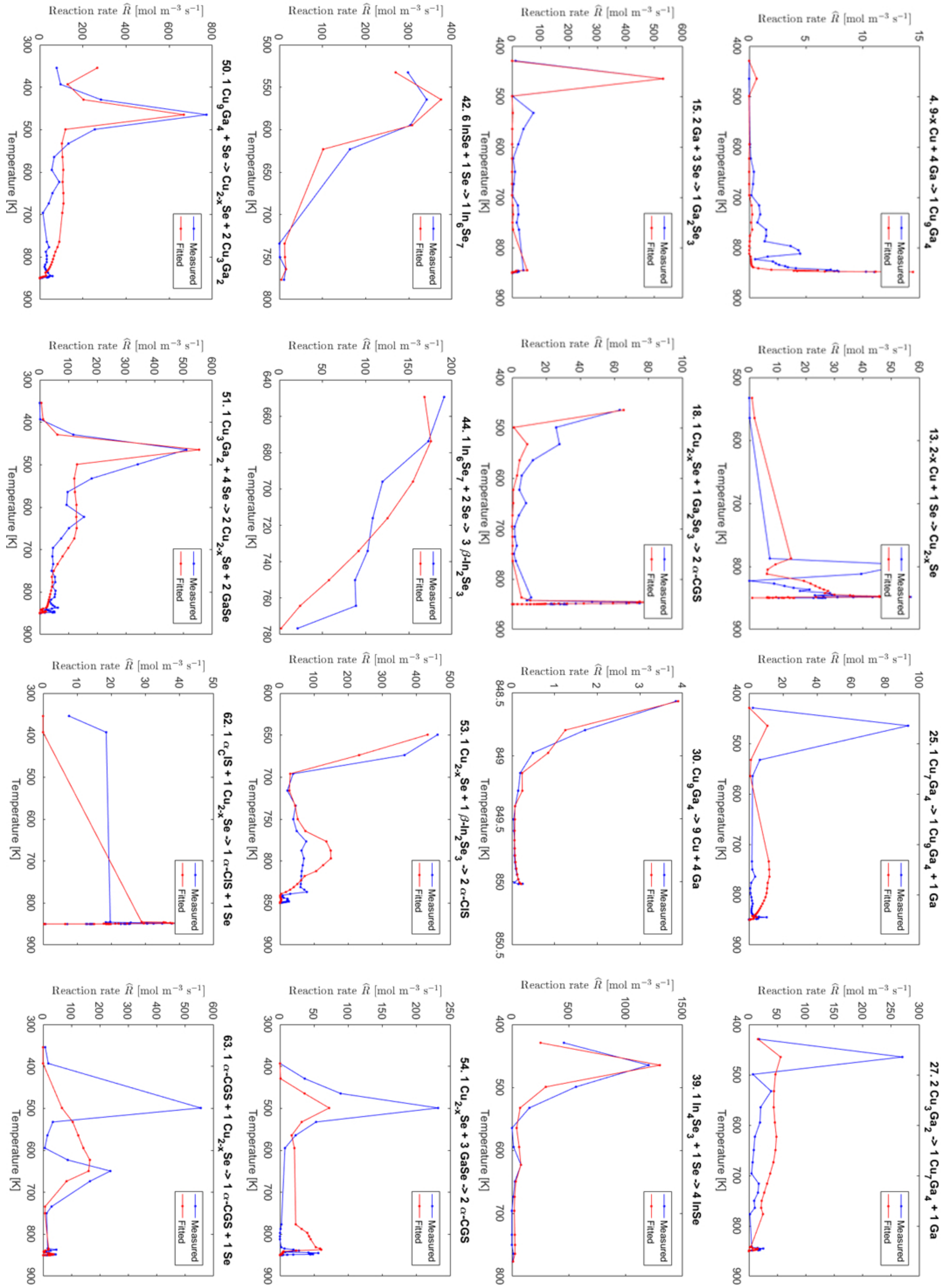


Figure 53: Fitted Arrhenius equations at each temperature point for a set of reactions.

Comparing the optimal values of the frequency factor and the activation energy with the original values, we see that most do not match. The frequency factor is always too low. This might be caused by the conversion from surface to volumetric concentration and from change in surface concentration to reaction rate. Here, we both divided by the height of the stack h_{stack}. In the Arrhenius equation (21), the volumetric concentrations are multiplied, thus the effect of dividing by the stack height is greater than on the left side, where there is only divided by one time the stack height. This would imply that a difference in frequency factor could be seen between reactions with one, two or three reactants. However, this does not clearly follow from our results. There is a slight difference, but this can be a coincidence. Therefore, this reason is unlikely, but we do not have another explanation.

Upon looking at the Figure 52, the shape of the fitted function matches approximately with the measured values. This indicates that the activation energy is adequate. In some cases, e.g. reactions 42, 44 and 53, this is indeed correct. For these reactions in Figure 53, there are points in a wide variety of temperatures. However, this is not always the case. For several reactions, all data points have temperatures that are approximately equal, for instance reaction 13. This makes the temperature dependency very weak. In that case, the activation energy becomes a scaling factor, just as the frequency factor. The frequency factor and the activation energy can both be increased and the fit remains the same. This can be better seen when we visualize Ω for the range of the parameters for the thirteenth reaction (see Figure 54). The computed optimal value is represented with a red cross.

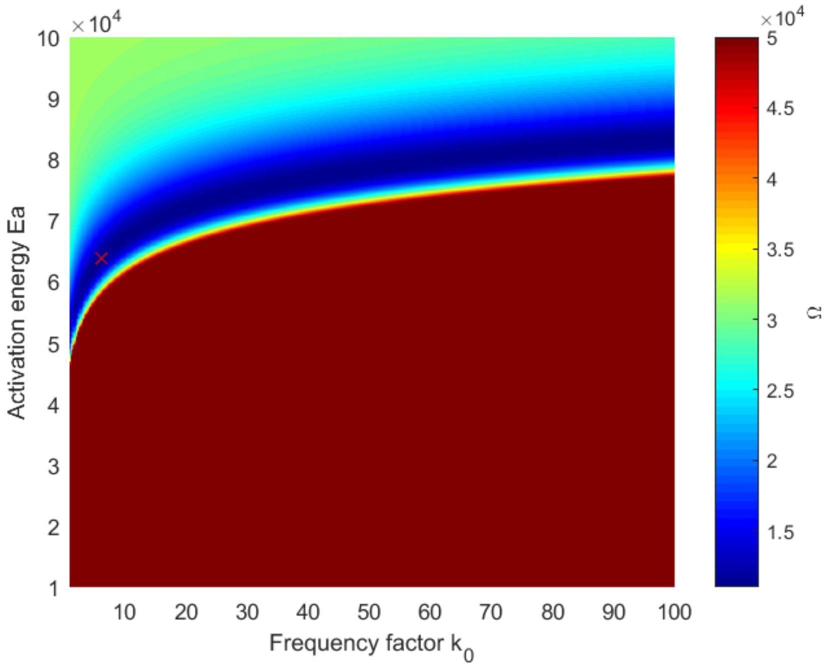


Figure 54: Residual Ω for the total range of the frequency factor and the activation energy for the thirteenth reaction defined in (21).

There is an entire curve where Ω is minimized. However, the values of the model are not on this curve. We took a reaction where this effect is severe, but for the other reaction this effect will also be present. Therefore, to get a better fit for the frequency factor and the activation energy, more data points at different temperatures are needed.

Concluding this chapter, we have constructed a method for determining the frequency factor and the activation energy in the Arrhenius equation. However, with the current data sets, the computed parameters are not consistent. This is mainly due to the fact that the temperature at most data points is constant. This makes the temperature dependency superfluous and both parameters become scaling factors. To resolve this, the measurement should contain points at many different temperatures. For instance, threshold the temperature at critical values for some reactions.

9 Summary and conclusions

For the extraction of the reaction kinetics (Arrhenius parameters) from *in-situ* X-ray diffraction measurements, we developed an analysis tool. To test our tool we used two measurements. The first measurement consists of the heating and cooling of a CIG stack. This measurement was performed at TNO/Solliance. The second measurement is simulated with the model we created in a previous study. In this measurement CIS and CGS are formed from the precursor stack of copper, indium and gallium. The outline of the tool is as follows:

1. Decomposition of the diffraction pattern
    (a) Preliminary data analysis
    (b) Optimizing the parameters in the Rietveld method
2. Calculating the reaction kinetics
    (a) Conversion of integrals to surface concentrations [mol m^{-2}]
    (b) Calculating the reaction rates [mol m^{-3} s^{-1}]
    (c) Fitting the Arrhenius equation

The measured diffraction pattern consists of much more than only the reflection of the crystals. Therefore, we perform a preliminary data analysis to approximate the other factors. We use a low pass noise filter to smoothen the signal and use a Chebyshev polynomial to approximate the background intensity. Next, we locate the peaks with a peak detection algorithm that is based on the wavelet transform of the diffraction pattern. Then, based on the location of the molybdenum peak, we correct for the z-displacement of the holder of the sample, which might be displaced due to temperature gradients in the setup. Molybdenum does not expand upon heating, therefore this is a good reference point. Other components, such as indium, expand upon heating (until indium melt at 434 K). Therefore, we used an approximation to the lattice parameters as function of temperature. Finally, we introduced a base shift of the reflections to account for the stresses in the thin film. However, this parameter will be optimized together with the scale factor of the Rietveld method.

The Rietveld method is a function that calculates the reflection of crystals by using certain properties of the crystal structure. The diffraction angle (position in the diffraction pattern) of the different planes in a crystal is calculated by Bragg's law. The diffraction angle of several planes might be the same due to symmetry in the crystal structure. The multiplicity factor counts the number of planes with the same diffraction angle. The structure factor accounts for the intensity of the reflection of the planes. This factor involves the scattering of X-rays and the occupancy of the sites by atoms in the crystal structure. Usually, the temperature factor (vibrations of atoms due to temperature) is also present in the structure factor, but we neglect this, since it has almost no influence when using metallic compounds. Furthermore, the Lorentz polarization factor accounts for two aspects of the geometrical effects. For the peak shape we use the Pearson-VII function. There is also a general scaling factor. Finally, the preferred orientation factor accounts for the preferred planes in the crystal. Since we work with a thin film, the crystals might grow in one direction. This results in an increased intensity for this particular plane.

In the Rietveld method combined with the stress correction, several parameters can be adjusted to fit the diffraction pattern. These parameters are the base shift (position of all peaks), the lattice parameters (position of an individual peak), scale factor (scaling of all peaks), preferred orientation factor (mutual ratio between peaks) and the peak shape (height, width and base width of all peaks). Almost everything of the reflection of a component can be adjusted. Therefore, we optimize the parameters in several steps. Several algorithms have been examined for the optimization of the parameters. First, the scale and base shift are optimized with the brute-force method to match the measured peaks with the components. Then, the preferred orientation is optimized with the interior point algorithm, followed by the lattice parameters (if needed), which is also done with the interior-point algorithm. The lattice parameters might be off due to the incorrectness of the thermal expansion approximation. Next, the peak shape is optimized with the OptQuest NLP algorithm. Finally, all parameters are varied a little to get an even better result. The calculation of the peak shape function is the bottleneck in the Rietveld function evaluation.

If the function of the Rietveld method adequately approximates the diffraction pattern, we can use the integrated intensities of the individual components as a measure for the surface concentration. The conversion of an integral value to a surface concentration is not trivial. Each component has their own transfer rate. By using the fact that the total amount of copper, indium, gallium and molybdenum stay constant during the measurements and we know how much selenium is absorbed by the stack, we can calculate the individual transfer rates. For the heating/cooling measurement, we could not calculate the surface concentrations as there was amorphous indium present. This gives no reflection and thus there is one extra variable that is always present. This made it impossible to uniquely determine the surface concentrations. For the simulated measurement this could be done, because we used the reverse process to create the diffraction pattern. The retrieval of the transfer rates was satisfactory. The only real difference was in the transfer rate of copper. This was because we did not incorporate the concentration of holes in the analysis.

With the change in surface concentration, we were able to calculate the reaction rates between two measurements. This involved solving a positive linear system that matches the stoichiometric coefficients of the reactants and products with the change in surface concentration for each reaction. This change in concentration can be converted to reaction rates by dividing by the time between two measurements and the height of the stack. Since the reaction rates have to be positive, we cannot find a unique solution. However, this was not be expected since we (again) did account for the holes in the stoichiometric coefficients. The calculated reaction rates for the simulated measurement are adequate. In the first fifteen measurements, many things are happening, therefore, it is difficult to get accurate results. The fitting of the function in the Rietveld method was reasonable in this area, but the reaction rates are also not perfect. The relative errors by comparing the actual change in concentration with the computed change in the reaction rate in this region is maximal 18% for four components (out of 24). Afterwards, we perform a correction for the reactions that happen alongside each other. This causes the reaction rates to decrease as a component has to be divided between two reactions. With this correction, we determine the individual reaction rates.

The corrected reaction rates are finally used to fit the Arrhenius equation. The temperatures at each measurement are known and we calculated the volumetric concentrations by dividing the surface concentration by the height of the stack. Furthermore, we add the products of all reactions to account for the volumetric concentrations of intermediate components. We calculated the frequency factor and the activation energy for sixteen reactions with corresponding confidence intervals. The first thing we noticed is that the frequency factors are too low. Furthermore, at most data points the temperature is almost constant. This makes the temperature dependency very weak. Then, both parameters become scaling factors and they are interchangeable. Therefore, many of the found parameters can be different, but still be optimal. To resolve this problem, the measured data should have a more diverse set of temperatures.

10 Suggestions for further work

So far, the tool to analyze the diffraction patterns has only been tested on two measurement sets; the heating/cooling of a CIG stack and the simulation of the formation of CIGS. The heating/cooling measurement had the problem of amorphous indium and therefore, the concentration profile could not be determined. The simulated measurement was a test to set all the settings of the tool to optimal values (algorithms, constants, etc.). The next step is to use this tool to analyze a real measurement with selenium. Probably, some things have to be adjusted due to the difference between the simulated measurement and a real measurement. These things might involve accuracy of the intermediate answers for different settings. Furthermore, more components have to be added. In this work we already mentioned that $\mathrm{Cu_9(Ga, In)_4}$ and $\mathrm{Cu_{16}(Ga, In)_9}$ should be added, but also extra CIS and CGS phases ($\mathrm{Cu(In, Ga)_3Se_5}$) should be added.

The tool itself can also be improved. The first thing is to enable holes in certain components. This can be done by estimating the concentration of holes in CIS and CGS in the last measurement. Then work backwards with the calculated reaction rates. The holes change the stoichiometric coefficients in the reactions and thus it changes the fits of the reaction rates. Another possibility is to add more factors to the preliminary data analysis or the Rietveld method. The setup with different slits and mirrors (for the X-rays) have an effect on the broadness of the peaks. Incorporating these effects will change the shape of the peaks and thus the integrated intensity. In the Rietveld method, the asymmetry of the peaks can be added. Some peaks are not symmetric due to stresses inside the material.

Another way of solving the least squares minimization problems for the Rietveld parameters is to use a machine learning approach. This kind of approach is often used for pattern recognition and might very well work here. However, a problem with this might be that it needs a large amount of data to train on. This could be solved by generating data with the simulation model. Still some real data is needed to make it work properly.

A Crystal structures

This appendix is a small introduction to the chemistry of crystal structures. It covers the theory of different unit cells that occur in crystals, crystallographic planes and their notation (Miller indices), the distance between parallel planes and the symmetry of planes in a unit cell. The main books that were used to write this chapter are *Fundamentals of Materials Science and Engineering* by W. Callister and D. Rethwisch [26] and *Fundamentals of Powder Diffraction and Structural Characterization of Materials* from V. Pecharsky and P. Zavalij [27].

A.1 Unit cells

In crystalline materials the atoms are ordered in a structural way. There is a basic crystal structure, that is repeated many times. For simple materials, such as pure metals, the crystal structure is easy, since there are not many possibilities to arrange the atoms. For materials with more different elements, the crystal structure is usually more complex. The repeating crystal structure is called the unit cell. The most common unit cells for pure metals are shown Figure 55.

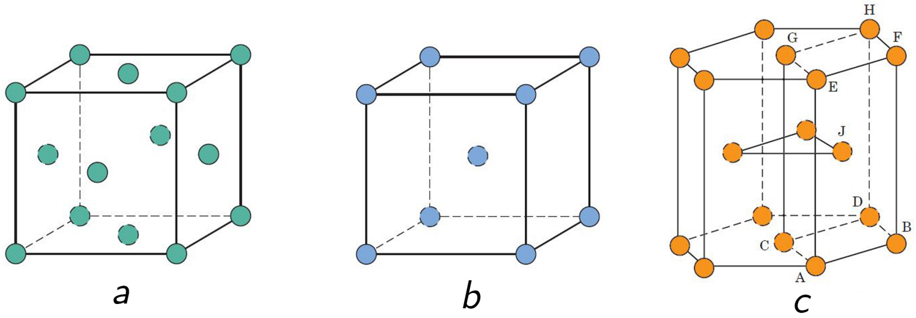


Figure 55: Some unit cells types of crystal structures; **(a)**: Face-Centered Cubic (FCC), **(b)**: Body-Centered Cubic (BCC), **(c)**: Hexagonal Close-Packed (HCP). [26]

The crystal structure has a huge influence on the properties of the material. For instance the atoms in a FCC structure are more closely packed than in a BCC structure, therefore the density of a metal with a FCC structure is higher than that of a metal with a BCC structure. In alloys the metals have different atomic radii. Therefore, they may not fit into a cubic structure, but it may have a tetragonal structure. In general, a unit cell is defined by six parameters: the length, the width and the height of each lattice, a, b, and c respectively, and the angles between the lattices, α, β and γ respectively (see Figure 56).

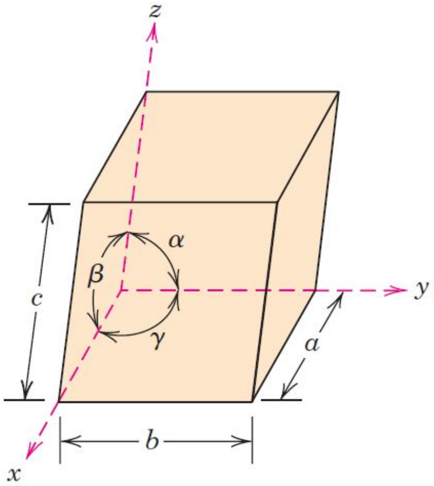


Figure 56: The lattice parameters which define a unit cell. [26]

Upon varying the lattice parameters a different unit cell is created. It is useful to group unit cells with the same geometry together, since they can exhibit the same properties. Some geometries and their names are given in Figure 57. Note that a unit cell is the repeating structure, this tells us nothing about the arrangement of the atoms inside the cell, e.g. FCC and BCC both have a cubic unit cell.

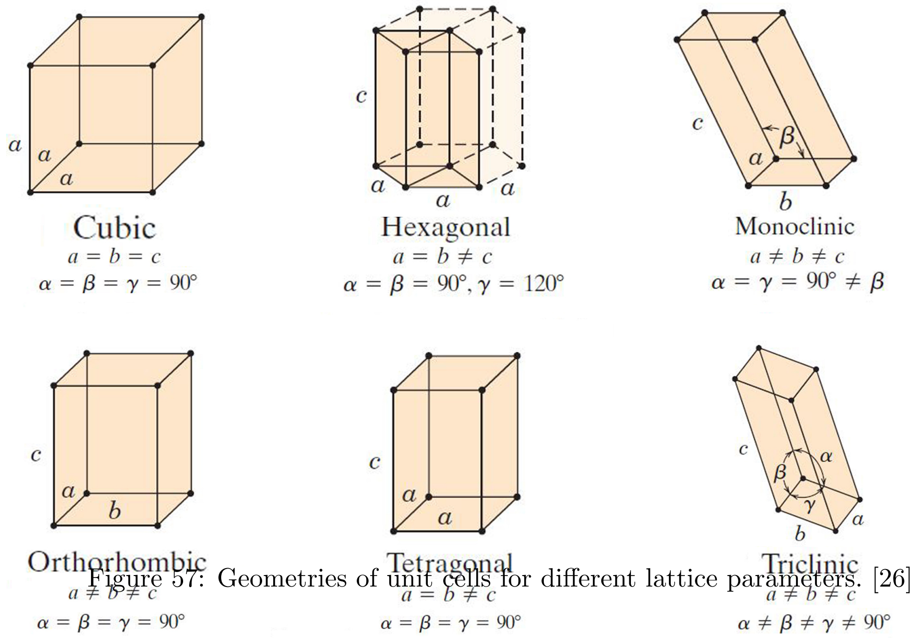
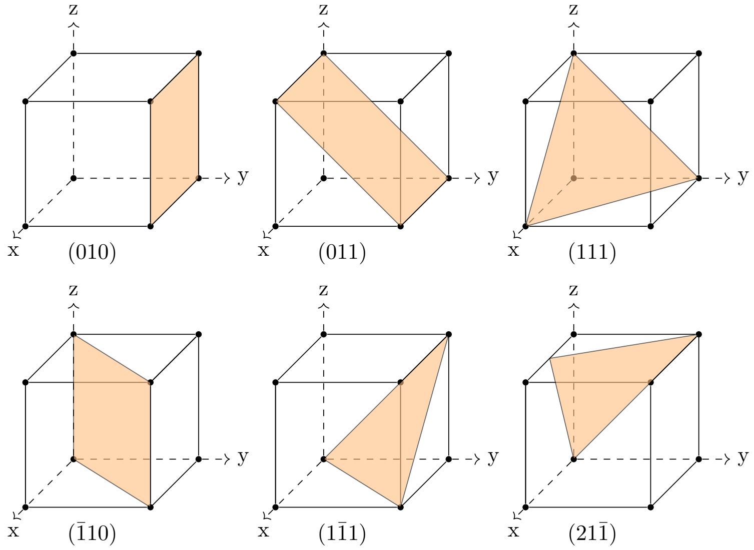


Figure 57: Geometries of unit cells for different lattice parameters. [26]

The positions of atoms in a unit cell are given by coordinates in the xyz-plane relative to the lattice lengths and along the lattices. Thus the coordinates in each direction are always numbers between 0 and 1. For example in a BCC structure the atoms are at the coordinates $000, 100, 010, 110, \frac{1}{2}\frac{1}{2}\frac{1}{2}, 001, 101, 011, 111$.

A.2 Crystallographic planes and Miller indices

Crystallographic planes link atoms to each other in crystal structures. There can be many different planes in a unit cell and some might be similar due to symmetry. The planes are specified by three Miller indices (hkl), with $h, k, l \in \mathbb{Z}$. A plane then passes through the points in the xyz-plane at $a/h, b/k$ and c/l. If one of the indices is zero, then the line (or point in case two of the indices are zero) would be extended infinitely in that direction. Thus the plane will be parallel to the axis whose index is zero. If a index is negative, then the plane passes through a point outside the unit cell. An overbar on an index indicates a minus sign, thus a negative index. Some examples of planes with their Miller indices are shown in Figure 58.



(010) (011) (111)

$(\bar{1}10)$ $(1\bar{1}1)$ $(21\bar{1})$

Figure 58: Examples of crystallographic planes and their corresponding Miller indices.

If the lattice parameters are equal, then some planes are similar due to symmetry. For instance in a pure metal where $a = b = c$, the planes $(100), (010), (001), (00\bar{1}), (0\bar{1}0), (00\bar{1})$ are similar. The number of similar planes in a unit cell is called the multiplicity of a plane. This number will play an important role in the analysis of the diffraction measurements as the distance between these planes is the same. Each crystal structure (unit cell) has a different set of similar plane, because of the different symmetries. Next, we will discuss how to calculate the distance between planes and then the symmetry in crystal structures.

A.3 Distance between planes

The distance between two parallel planes is the length of the vector that is perpendicular to the planes. If the plane has nonnegative Miller indices, then one plane goes through the origin and another plane lies inside the unit cell. Then the distance between the planes is the length of the normal vector, which goes through the origin. This can be easily represented by a vector with three elements (for the three directions). When a plane has negative Miller indices, then the normal vector does not go through the origin. However, we can always redefine our axis such that the normal vector goes through the origin, which we assume in the remainder of this section. Now the length of the normal vector can be calculated with some simple geometry. For example, take an orthorhombic unit cell with a general plane, which intercepts the lattice vectors at $\tilde{a} = \frac{a}{h}, \tilde{b} = \frac{b}{k}$ and $\tilde{c} = \frac{c}{l}$ (see Figure 59).

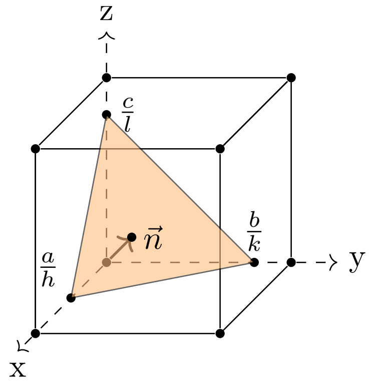


Figure 59: A general plane in an orthorhombic unit cell.

The normal to this plane is given by the cross product of the vectors that make up the plane: $n = (P_2 - P_1) \times (P_3 - P_1)$ with $P_1 = [\tilde{a}, 0, 0], P_2 = [0, \tilde{b}, 0]$ and $P_3 = [0, 0, \tilde{c}]$. The normal vector is: $\vec{n} = (\tilde{b}\tilde{c}, \tilde{a}\tilde{c}, \tilde{a}\tilde{b})$. The formula of the Miller plane is: $\tilde{b}\tilde{c} \cdot x + \tilde{a}\tilde{c} \cdot y + \tilde{a}\tilde{b} \cdot z - \tilde{a}\tilde{b}\tilde{c} = 0$. The minimal length between two planes is the distance between the origin and the intersection of the normal vector (from the origin) and the plane. This vector can be easily calculated by substituting the normal vector with an additional constant in the formula for the Miller plane and then scaling the constant such that the formula is correct. This vector is: $\vec{d} = \frac{\tilde{a}\tilde{b}\tilde{c}}{\|\vec{n}\|}(\tilde{b}\tilde{c}, \tilde{a}\tilde{c}, \tilde{a}\tilde{b})$ with $\|\vec{n}\| = (\tilde{b}\tilde{c})^2 + (\tilde{a}\tilde{c})^2 + (\tilde{a}\tilde{b})^2$. The length of this vector is then the distance between two planes:

$$
\begin{aligned}
\|\vec{d}\| &= \left[\left(\frac{\tilde{a}\tilde{b}\tilde{c}}{\|\vec{n}\|}\tilde{b}\tilde{c}\right)^2 + \left(\frac{\tilde{a}\tilde{b}\tilde{c}}{\|\vec{n}\|}\tilde{a}\tilde{c}\right)^2 + \left(\frac{\tilde{a}\tilde{b}\tilde{c}}{\|\vec{n}\|}\tilde{a}\tilde{b}\right)^2\right]^{1/2}, \\
&= \left[\left(\frac{\tilde{a}\tilde{b}\tilde{c}}{\|\vec{n}\|}\right)^2 \cdot \|\vec{n}\|\right]^{1/2}, \\
&= \left[\frac{\tilde{a}^2\tilde{b}^2\tilde{c}^2}{(\tilde{b}\tilde{c})^2 + (\tilde{a}\tilde{c})^2 + (\tilde{a}\tilde{b})^2}\right]^{1/2}, \\
&= \left[\frac{1}{\frac{1}{\tilde{a}^2} + \frac{1}{\tilde{b}^2} + \frac{1}{\tilde{b}^2}}\right]^{1/2}, \\
&= \left[\frac{k^2}{a^2} + \frac{h^2}{b^2} + \frac{l^2}{c^2}\right]^{-1/2}. \qquad (22)
\end{aligned}
$$

With a similar analysis the distance between planes in different crystal structures can be calculated. This results in the following equations for different crystal structures:

$$\text{Cubic: } \frac{1}{d^2} = \frac{h^2+k^2+l^2}{a^2}, \tag{23a}$$

$$\text{Tetragonal: } \frac{1}{d^2} = \frac{h^2+k^2}{a^2} + \frac{l^2}{c^2}, \tag{23b}$$

$$\text{Hexagonal: } \frac{1}{d^2} = \frac{4}{3}\frac{h^2+hk+k^2}{a^2} + \frac{l^2}{c^2}, \tag{23c}$$

$$\text{Orthorhombic: } \frac{1}{d^2} = \frac{k^2}{a^2} + \frac{h^2}{b^2} + \frac{l^2}{c^2}, \tag{23d}$$

$$\text{Monoclinic: } \frac{1}{d^2} = \frac{h^2}{a^2\sin^2(\beta)} + \frac{k^2}{b^2} + \frac{l^2}{c^2\sin^2(\beta)} + \frac{2hl\cos(\beta)}{ac\sin^2(\beta)}, \tag{23e}$$

$$\begin{aligned}\text{Triclinic: } \frac{1}{d^2} = \Bigg(&\frac{h^2}{a^2}\sin^2(\alpha) + \frac{2kl}{bc}(\cos(\beta)\cos(\gamma) - \cos(\gamma)) + \\ &\frac{k^2}{b^2}\sin^2(\beta) + \frac{2hl}{ac}(\cos(\alpha)\cos(\gamma) - \cos(\beta)) + \\ &\frac{l^2}{c^2}\sin^2(\gamma) + \frac{2hk}{ab}(\cos(\alpha)\cos(\beta) - \cos(\gamma))\Bigg) \\ &/\left(1 - \cos^2(\alpha) - \cos^2(\beta) - \cos^2(\gamma) + 2\cos(\alpha)\cos(\beta)\cos(\gamma)\right).\end{aligned} \tag{23f}$$

The equation for triclinic crystals is the most general and this equation will reduce to the other equations when the metrical conditions of other crystal structures are used. Therefore we will use the triclinic equation for the calculation of the distance between planes.

A.4 Symmetry and space groups

The symmetry in crystal structures is of great importance when analyzing the diffraction pattern, as it influences the location and intensity of the different peaks. Every unit cell has a asymmetric part, which forms the basis of the atoms in a unit cell. Via symmetry operations on the asymmetric part the other atoms in the unit cell can be placed. There are four symmetry operations: rotation, inversion, reflection and translation, which are illustrated in Figure 60.

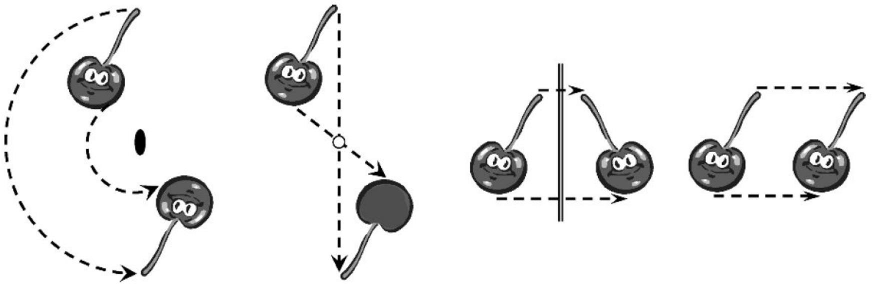


Figure 60: Four symmetry operations. From left to right: rotation, inversion, reflection and translation. [27]

Geometrically these symmetry operations are represented by either a point, vector/line or a plane. The symmetry elements are: rotation axis, center of inversion, mirror plane and translation vector. Some symmetry operations can be combined to form a new symmetry element; rotation and inversion give a roto-inversion axis, rotation and translation give a screw axis and reflection and translation give a glide plane. Thus there are seven symmetry elements in total. All the symmetry elements are represented be a symbol (number or letter). These symbols are given in Table 16.

Table 16: Hermann-Mauguin symbols for symmetry elements.

Symbol	Meaning	Symbol	Meaning
1	360° rotation axis	$\bar{1}$	360° roto-inversion axis [1]
2	180° rotation axis	$\bar{2}$	180° roto-inversion axis [2]
3	120° rotation axis	$\bar{3}$	120° roto-inversion axis
4	90° rotation axis	$\bar{4}$	90° roto-inversion axis
6	60° rotation axis	$\bar{6}$	60° roto-inversion axis
m	Mirror plane	a	glide plane with shift $\frac{1}{2}a$[3]
2_1	Screw axis with shift $\frac{1}{2}$	b	Glide plane with shift $\frac{1}{2}b$[3]
$3_1, 3_2$	Screw axis with shift $\frac{1}{3}, \frac{2}{3}$	c	Glide plane with shift $\frac{1}{2}c$[3]
$4_1, 4_2, 4_3$	Screw axis with shift $\frac{1}{4}, \frac{2}{4} \frac{3}{4}$	n	Glide plane with shift $\frac{1}{2}(a+b)$[3]
$6_1, 6_2, 6_3, 6_4, 6_5$	Screw axis with shift $\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}$	d	Glide plane with shift $\frac{1}{4}(a+b)$[3]

[1] also used for center of inversion.
[2] equal to mirror plane.
[3] where a, b and c are lattice parameters.

Upon combining two different symmetry operations, no new symmetry operations arises. For instance, the combination of a twofold rotation axis with a center of inversion is equal to a mirror plane that is perpendicular to the rotation axis and goes through the center of inversion. Also, the order of the symmetry operations can be interchanged. In symbols the symmetry operation for the example would be:

$$2 \times \bar{1}(\text{ on } 2) = \bar{1}(\text{ on } 2) \times 2 = m(\perp 2 \text{ through } \bar{1}),$$

where × represents the consecutive application of the symmetry operation on an object. Thus the combination of rotation and inversion (which includes reflection) provides a closed set of symmetrical atoms, which are called point groups. These are called point groups, because all the symmetry elements in such a group have at least one point in common.
To be able to characterize crystal structure further, the type of unit cell is important. There are seven types of standard unit cells:

1. Primitive (P), contains no extra atoms, but the corners of the unit cell.

2-4. Base-centered (A, B, C), contains additional atoms at the opposite faces of the A, B or C-plane.

5. Body-centered (I), contains an extra atom at the center of the unit cell.

6. Face-centered (F), contains extra atoms at the centers of all faces of the unit cell.

7. Rhombohedral (R), contains two extra atoms at $\frac{1}{3}\frac{2}{3}\frac{2}{3}$ and $\frac{2}{3}\frac{1}{3}\frac{1}{3}$ (only for trigonal structure).

Examples of some standard unit cells are given in Figure 61.

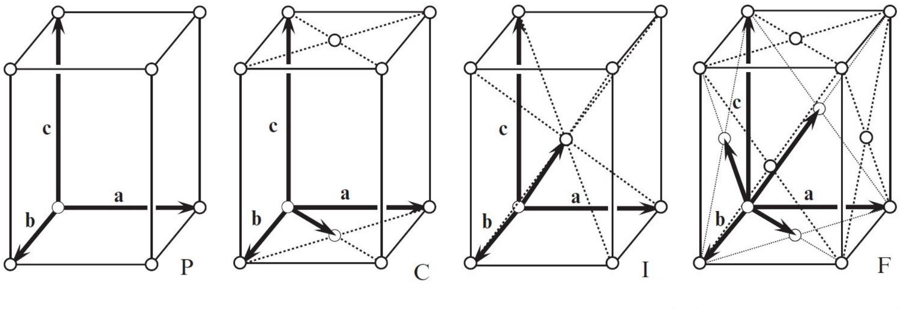


Figure 61: Four standard unit cells. From left to right: primitive (P), base-centered (C), body-centered (I) and face-centered (F). [27]

Combining the symmetry elements from point groups with translation (thus adding screw axis and glide planes) and a given standard unit cell, this results in a space group. Due to the limitations in rotation and translation, there are a total of 230 space groups. Thus the symmetry of a crystal can be described by a space group. A full list of all the space groups and their properties can be found in the International Table for Crystallography Volume A [49]. For more information about symmetry and examples, the reader is referred to the book *Fundamentals of Powder Diffraction and Structural Characterization of Materials* from V. Pecharsky and P. Zavalij [27].

B Tables

B.1 Multiplicity planes

Table 17: Families of planes and their multiplicity factor in a triclinic crystal

Multiplicity	Similar planes
2	$hkl, \bar{h}\bar{k}\bar{l}$

Table 18: Families of planes and their multiplicity factor in a monoclinic crystal, unique b axis

Multiplicity	Similar planes
4	$hkl, \bar{h}k\bar{l}, h\bar{k}l, \bar{h}\bar{k}\bar{l}$
2	$h0l, \bar{h}0\bar{l}$
2	$0k0, 0\bar{k}0$

Table 19: Families of planes and their multiplicity factor in a orthorhombic crystal

Multiplicity	Similar planes
8	$hkl, \bar{h}kl, h\bar{k}l, hk\bar{l}, \bar{h}\bar{k}l, \bar{h}k\bar{l}, h\bar{k}\bar{l}, \bar{h}\bar{k}\bar{l}$
4	$0kl, 0\bar{k}l, 0k\bar{l}, 0\bar{k}\bar{l}$
4	$h0l, \bar{h}0l, h0\bar{l}, \bar{h}0\bar{l}$
4	$hk0, \bar{h}k0, h\bar{k}0, \bar{h}\bar{k}0$
2	$h00, \bar{h}00$
2	$0k0, 0\bar{k}0$
2	$00l, 00\bar{l}$

Table 20: Families of planes and their multiplicity factor in a tetragonal crystal

Multiplicity	Similar planes
16	$hkl, \bar{h}kl, h\bar{k}l, hk\bar{l}, \bar{h}\bar{k}l, \bar{h}k\bar{l}, h\bar{k}\bar{l}, \bar{h}\bar{k}\bar{l}$ $khl, \bar{k}hl, k\bar{h}l, kh\bar{l}, \bar{k}\bar{h}l, \bar{k}h\bar{l}, k\bar{h}\bar{l}, \bar{k}\bar{h}\bar{l}$
8	$hhl, \bar{h}hl, h\bar{h}l, \bar{h}\bar{h}l, hh\bar{l}, \bar{h}h\bar{l}, h\bar{h}\bar{l}, \bar{h}\bar{h}\bar{l}$
8	$h0l, \bar{h}0l, h0\bar{l}, \bar{h}0\bar{l}, 0kl, 0\bar{k}l, 0k\bar{l}, 0\bar{k}\bar{l}$
8	$hk0, \bar{h}k0, h\bar{k}0, \bar{h}\bar{k}0, kh0, \bar{k}h0, k\bar{h}0, \bar{k}\bar{h}0$
4	$hh0, \bar{h}h0, h\bar{h}0, \bar{h}\bar{h}0$
4	$h00, \bar{h}00, 0k0, 0\bar{k}0$
2	$00l, 00\bar{l}$

Table 21: Families of planes and their multiplicity factor in a hexagonal crystal

Multiplicity	Similar planes
24	$hkil, kihl, ihkl, \bar{h}\bar{k}\bar{i}l, \bar{k}\bar{i}\bar{h}l, \bar{i}\bar{h}\bar{k}l, khil, hikl, ikhl, \bar{k}\bar{h}\bar{i}l, \bar{h}\bar{i}\bar{k}l, \bar{i}\bar{k}\bar{h}l$ $hki\bar{l}, kih\bar{l}, ihk\bar{l}, \bar{h}\bar{k}\bar{i}\bar{l}, \bar{k}\bar{i}\bar{h}\bar{l}, \bar{i}\bar{h}\bar{k}\bar{l}, khi\bar{l}, hik\bar{l}, ikh\bar{l}, \bar{k}\bar{h}\bar{i}\bar{l}, \bar{h}\bar{i}\bar{k}\bar{l}, \bar{i}\bar{k}\bar{h}\bar{l}$
12	$hh\bar{2}hl, h\bar{2}hhl, \bar{2}hhhl, \bar{h}\bar{h}2hl, \bar{h}2h\bar{h}l, 2h\bar{h}\bar{h}l, hh\bar{2}h\bar{l}, h\bar{2}hh\bar{l}, \bar{2}hhh\bar{l}, \bar{h}\bar{h}2h\bar{l}, \bar{h}2h\bar{h}\bar{l}, 2h\bar{h}\bar{h}\bar{l}$
12	$0k\bar{k}l, k\bar{k}0l, \bar{k}0kl, 0\bar{k}kl, \bar{k}k0l, k0\bar{k}l, 0k\bar{k}\bar{l}, k\bar{k}0\bar{l}, \bar{k}0k\bar{l}, 0\bar{k}k\bar{l}, \bar{k}k0\bar{l}, k0\bar{k}\bar{l}$
12	$hki0, kih0, ihk0, \bar{h}\bar{k}\bar{i}0, \bar{k}\bar{i}\bar{h}0, \bar{i}\bar{h}\bar{k}0, khi0, hik0, ikh0, \bar{k}\bar{h}\bar{i}0, \bar{h}\bar{i}\bar{k}0, \bar{i}\bar{k}\bar{h}0$
6	$hh\bar{2}h0, h\bar{2}hh0, \bar{2}hhh0, \bar{h}\bar{h}2h0, \bar{h}2h\bar{h}0, 2h\bar{h}\bar{h}0$
6	$0k\bar{k}0, k\bar{k}00, \bar{k}0k0, 0\bar{k}k0, \bar{k}k00, k0\bar{k}0$
2	$000l, 000\bar{l}$

Table 22: Multiplicity factor for planes in a cubic crystal

Multiplicity	Similar planes
48	$hkl, \bar{h}kl, h\bar{k}l, hk\bar{l}, \bar{h}\bar{k}l, \bar{h}k\bar{l}, h\bar{k}\bar{l}, \bar{h}\bar{k}\bar{l}, khl, \bar{k}hl, k\bar{h}l, kh\bar{l}$ $\bar{k}\bar{h}l, \bar{k}h\bar{l}, k\bar{h}\bar{l}, \bar{k}\bar{h}\bar{l}, hlk, \bar{h}lk, h\bar{l}k, hl\bar{k}, \bar{h}\bar{l}k, \bar{h}l\bar{k}, h\bar{l}\bar{k}, \bar{h}\bar{l}\bar{k}$ $lkh, \bar{l}kh, l\bar{k}h, lk\bar{h}, \bar{l}\bar{k}h, \bar{l}k\bar{h}, l\bar{k}\bar{h}, \bar{l}\bar{k}\bar{h}, klh, \bar{k}lh, k\bar{l}h, kl\bar{h}$ $\bar{k}\bar{l}h, \bar{k}l\bar{h}, k\bar{l}\bar{h}, \bar{k}\bar{l}\bar{h}, lhk, \bar{l}hk, l\bar{h}k, lh\bar{k}, \bar{l}\bar{h}k, \bar{l}h\bar{k}, l\bar{h}\bar{k}, \bar{l}\bar{h}\bar{k}$
24	$hhl, \bar{h}hl, h\bar{h}l, \bar{h}\bar{h}l, hh\bar{l}, \bar{h}h\bar{l}, h\bar{h}\bar{l}, \bar{h}\bar{h}\bar{l}, hkk, h\bar{k}k, hk\bar{k}, h\bar{k}\bar{k}$ $\bar{h}kk, \bar{h}\bar{k}k, \bar{h}k\bar{k}, \bar{h}\bar{k}\bar{k}, hkh, \bar{h}kh, hk\bar{h}, \bar{h}k\bar{h}, h\bar{k}h, \bar{h}\bar{k}h, h\bar{k}\bar{h}, \bar{h}\bar{k}\bar{h}$
24	$hk0, \bar{h}k0, h\bar{k}0, \bar{h}\bar{k}0, h0l, \bar{h}0l, h0\bar{l}, \bar{h}0\bar{l}, kh0, \bar{k}h0, k\bar{h}0, \bar{k}\bar{h}0$ $0kl, 0\bar{k}l, 0k\bar{l}, 0\bar{k}\bar{l}, l0h, \bar{l}0h, l0\bar{h}, \bar{l}0\bar{h}, 0lk, 0\bar{l}k, 0l\bar{k}, 0\bar{l}\bar{k}$
12	$h0h, \bar{h}0h, h0\bar{h}, 0kk, 0\bar{k}k, 0k\bar{k}, 0\bar{k}\bar{k}, \bar{h}0\bar{h}, hh0, \bar{h}h0, h\bar{h}0, \bar{h}\bar{h}0$
8	$hhh, \bar{h}hh, h\bar{h}h, hh\bar{h}, \bar{h}\bar{h}h, \bar{h}h\bar{h}, h\bar{h}\bar{h}, \bar{h}\bar{h}\bar{h}$
6	$h00, \bar{h}00, 0k0, 0\bar{k}0, 00l, 00\bar{l}$

B.2 Preferred planes

Table 23: Preferred planes of all used components

Component	Preferred plane (h,k,(i),l)
Cu	(1,1,1)
In	(1,0,1) and (0,0,2)
Mo	(1,1,0)
$Cu_{4-x}Ga$	(0,0,0,2)
Cu_9Ga_4	(3,3,0)
Cu_7Ga_4	(3,3,0)
Cu_3Ga_2	(3,3,0)
Cu_7In_3	(2,3,1) and (2,-3,-1)
$Cu_{16}In_9$	(1,1,-2,0)
$Cu_{11}In_9$	(0,2,0) and (4,0,2)
$CuIn_2$	(2,1,1)
In_4Se_3	(2,0,0) and (0,4,0)
InSe	(0,0,6)
In_6Se_7	(2,1,-3) and (0,2,0)
β-In_2Se_3	(0,0,0,9) and (1,0,-1,1)
γ-In_2Se_3	(0,0,0,6) and (1,1,-2,0)
$CuSe_2$	(1,2,0)
CuSe	(0,0,0,4) and (0,0,0,6)
$Cu_{2-x}Se$	(2,2,0) and (3,1,1)
GaSe	(0,0,0,4) and (0,0,0,5)
Ga_2Se_3	(2,0,0) and (1,-1,3)
$MoSe_2$	(0,0,0,4)
α-CIS	(0,0,3) and (0,0,4)
β-CIS	(1,1,1) and (2,2,0)
α-CGS	(0,0,3) and (0,0,4)

B.3 Reactions

1. $16\,\mathrm{Cu} + 9\,\mathrm{In} \longrightarrow 1\,\mathrm{Cu_{16}In_9}$
2. $11\,\mathrm{Cu} + 9\,\mathrm{In} \longrightarrow 1\,\mathrm{Cu_{11}In_9}$
3. $4\text{-}x\,\mathrm{Cu} + 1\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu}_{4-x}\mathrm{Ga}$
4. $9\text{-}x\,\mathrm{Cu} + 4\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_9Ga_4}$
5. $7\text{-}x\,\mathrm{Cu} + 4\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_7Ga_4}$
6. $3\text{-}x\,\mathrm{Cu} + 2\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_3Ga_2}$
7. $4\,\mathrm{In} + 3\,\mathrm{Se} \longrightarrow 1\,\mathrm{In_4Se_3}$
8. $1\,\mathrm{In} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{InSe}$
9. $6\,\mathrm{In} + 7\,\mathrm{Se} \longrightarrow 1\,\mathrm{In_6Se_7}$
10. $2\,\mathrm{In} + 3\,\mathrm{Se} \longrightarrow 1\,\beta\text{-}\mathrm{In_2Se_3}$
11. $1\,\mathrm{Cu} + 2\,\mathrm{Se} \longrightarrow 1\,\mathrm{CuSe_2}$
12. $1\,\mathrm{Cu} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{CuSe}$
13. $2\text{-}x\,\mathrm{Cu} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{Cu}_{2-x}\mathrm{Se}$
14. $1\,\mathrm{Ga} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{GaSe}$
15. $2\,\mathrm{Ga} + 3\,\mathrm{Se} \longrightarrow 1\,\mathrm{Ga_2Se_3}$
16. $1\,\mathrm{Mo} + 2\,\mathrm{Se} \longrightarrow 1\,\mathrm{MoSe_2}$
17. $1\,\mathrm{CuSe} + 1\,\mathrm{InSe} \longrightarrow 1\,\alpha\text{-}\mathrm{CIS}$
18. $1\,\mathrm{Cu}_{2-x}\mathrm{Se} + 1\,\mathrm{Ga_2Se_3} \longrightarrow 2\,\alpha\text{-}\mathrm{CGS}$
19. $1\,\mathrm{Cu_{16}In_9} + \frac{45}{11}\,\mathrm{In} \longrightarrow \frac{16}{11}\,\mathrm{Cu_{11}In_9}$
20. $1\,\mathrm{Cu_{11}In_9} \longrightarrow \frac{11}{16}\,\mathrm{Cu_{16}In_9} + \frac{45}{11}\,\mathrm{In}$
21. $1\,\mathrm{Cu_{16}In_9} \longrightarrow 16\,\mathrm{Cu} + 9\,\mathrm{In}$
22. $1\,\mathrm{Cu_{11}In_9} \longrightarrow 11\,\mathrm{Cu} + 9\,\mathrm{In}$
23. $1\,\mathrm{Cu_9Ga_4} \longrightarrow 2\,\mathrm{Cu}_{4-x}\mathrm{Ga} + 2\,\mathrm{Ga}$
24. $2\,\mathrm{Cu}_{4-x}\mathrm{Ga} + 2\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_9Ga_4}$
25. $1\,\mathrm{Cu_7Ga_4} \longrightarrow 1\,\mathrm{Cu_9Ga_4} + 1\,\mathrm{Ga}$
26. $1\,\mathrm{Cu_9Ga_4} + 1\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_7Ga_4}$
27. $2\,\mathrm{Cu_3Ga_2} \longrightarrow 1\,\mathrm{Cu_7Ga_4} + 1\,\mathrm{Ga}$
28. $1\,\mathrm{Cu_7Ga_4} + 1\,\mathrm{Ga} \longrightarrow 2\,\mathrm{Cu_3Ga_2}$
29. $\mathrm{Cu}_{4-x}\mathrm{Ga} \longrightarrow 4\,\mathrm{Cu} + \mathrm{Ga}$
30. $\mathrm{Cu_9Ga_4} \longrightarrow 9\,\mathrm{Cu} + 4\,\mathrm{Ga}$
31. $\mathrm{Cu_7Ga_4} \longrightarrow 7\,\mathrm{Cu} + 4\,\mathrm{Ga}$
32. $\mathrm{Cu_3Ga_2} \longrightarrow 3\,\mathrm{Cu} + 2\,\mathrm{Ga}$
33. $1\,\mathrm{CuSe_2} \longrightarrow 1\,\mathrm{CuSe} + 1\,\mathrm{Se}$
34. $1\,\mathrm{CuSe} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{CuSe_2}$
35. $2\,\mathrm{CuSe} \longrightarrow 1\,\mathrm{Cu}_{2-x}\mathrm{Se} + 1\,\mathrm{Se}$
36. $1\,\mathrm{Cu}_{2-x}\mathrm{Se} + 1\,\mathrm{Se} \longrightarrow 2\,\mathrm{CuSe}$
37. $1\,\mathrm{Ga_2Se_3} \longrightarrow 2\,\mathrm{GaSe} + 1\,\mathrm{Se}$
38. $2\,\mathrm{GaSe} + \mathrm{Se} \longrightarrow 1\,\mathrm{Ga_2Se_3}$
39. $1\,\mathrm{In_4Se_3} + 1\,\mathrm{Se} \longrightarrow 4\,\mathrm{InSe}$

40. $4\,\mathrm{InSe} \longrightarrow 1\,\mathrm{In_4Se_3} + 1\,\mathrm{Se}$
41. $1\,\mathrm{In_6Se_7} \longrightarrow 6\,\mathrm{InSe} + 1\,\mathrm{Se}$
42. $6\,\mathrm{InSe} + 1\,\mathrm{Se} \longrightarrow 1\,\mathrm{In_6Se_7}$
43. $3\,\beta\text{-}\mathrm{In_2Se_3} \longrightarrow 1\,\mathrm{In_6Se_7} + 2\,\mathrm{Se}$
44. $1\,\mathrm{In_6Se_7} + 2\,\mathrm{Se} \longrightarrow 3\,\beta\text{-}\mathrm{In_2Se_3}$
45. $1\,\mathrm{In_4Se_3} \longrightarrow 4\,\mathrm{In} + 3\,\mathrm{Se}$
46. $1\,\mathrm{InSe} \longrightarrow 1\,\mathrm{In} + 1\,\mathrm{Se}$
47. $1\,\mathrm{In_6Se_7} \longrightarrow 6\,\mathrm{In} + 7\,\mathrm{Se}$
48. $1\,\beta\text{-}\mathrm{In_2Se_3} \longrightarrow 2\,\mathrm{In} + 3\,\mathrm{Se}$
49. $\mathrm{Cu_{16}In_9} + \mathrm{Se} \longrightarrow \mathrm{Cu_{2-\mathit{x}}Se} + \mathrm{InSe}$
50. $1\,\mathrm{Cu_9Ga_4} + \mathrm{Se} \longrightarrow 1\,\mathrm{Cu_{2-\mathit{x}}Se} + 2\,\mathrm{Cu_3Ga_2}$
51. $1\,\mathrm{Cu_3Ga_2} + 4\,\mathrm{Se} \longrightarrow 2\,\mathrm{Cu_{2-\mathit{x}}Se} + 2\,\mathrm{GaSe}$
52. $1\,\mathrm{Cu_{2-\mathit{x}}Se} + 2\,\mathrm{InSe} + 1\,\mathrm{Se} \longrightarrow 2\,\alpha\text{-}\mathrm{CIS}$
53. $1\,\mathrm{Cu_{2-\mathit{x}}Se} + 1\,\beta\text{-}\mathrm{In_2Se_3} \longrightarrow 2\,\alpha\text{-}\mathrm{CIS}$
54. $1\,\mathrm{Cu_{2-\mathit{x}}Se} + 3\,\mathrm{GaSe} \longrightarrow 2\,\alpha\text{-}\mathrm{CGS}$
55. $1\,\mathrm{Cu_{2-\mathit{x}}Se} + 2\,\mathrm{In} + 3\,\mathrm{Se} \longrightarrow 2\,\alpha\text{-}\mathrm{CIS}$
56. $\mathrm{Cu_{16}In_9} + \mathrm{In} \longrightarrow \mathrm{Cu_{16}In_9} + \mathrm{In}$
57. $\mathrm{Cu_{4-\mathit{x}}Ga} + \mathrm{Ga} \longrightarrow \mathrm{Cu_{4-\mathit{x}}Ga} + \mathrm{Ga}$
58. $\mathrm{Cu_{9-\mathit{x}}Ga_4} + \mathrm{Ga} \longrightarrow \mathrm{Cu_{9-\mathit{x}}Ga_4} + \mathrm{Ga}$
59. $\mathrm{Cu_{7-\mathit{x}}Ga_4} + \mathrm{Ga} \longrightarrow \mathrm{Cu_{7-\mathit{x}}Ga_4} + \mathrm{Ga}$
60. $\mathrm{Cu_{3-\mathit{x}}Ga_2} + \mathrm{Ga} \longrightarrow \mathrm{Cu_{3-\mathit{x}}Ga_2} + \mathrm{Ga}$
61. $\mathrm{Cu_{2-\mathit{x}}Se} + \mathrm{Se} \longrightarrow \mathrm{Cu_{2-\mathit{x}}Se} + \mathrm{Se}$
62. $1\,\alpha\text{-}\mathrm{CIS} + 1\,\mathrm{Cu_{2-\mathit{x}}Se} \longrightarrow 1\,\alpha\text{-}\mathrm{CIS} + 1\,\mathrm{Se}$
63. $1\,\alpha\text{-}\mathrm{CGS} + 1\,\mathrm{Cu_{2-\mathit{x}}Se} \longrightarrow 1\,\alpha\text{-}\mathrm{CGS} + 1\,\mathrm{Se}$
64. $7\,\mathrm{Cu} + 3\,\mathrm{In} \longrightarrow 1\,\mathrm{Cu_7In_3}$
65. $1\,\mathrm{Cu_7In_3} \longrightarrow 7\,\mathrm{Cu} + 3\,\mathrm{In}$
66. $1\,\mathrm{Cu} + 2\,\mathrm{In} \longrightarrow 1\,\mathrm{CuIn_2}$
67. $1\,\mathrm{CuIn_2} \longrightarrow 2\,\mathrm{Cu} + 1\,\mathrm{In}$
68. $2\,\mathrm{Cu_7In_3} + \frac{30}{16}\,\mathrm{In} \longrightarrow \frac{14}{16}\,\mathrm{Cu_{16}In_9}$
69. $1\,\mathrm{Cu_{16}In_9} \longrightarrow \frac{16}{7}\,\mathrm{Cu_7In_3} + \frac{15}{7}\,\mathrm{In}$
70. $1\,\mathrm{Cu} + 2\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_{1-\mathit{x}}Ga_2}$
71. $1\,\mathrm{Cu_{1-\mathit{x}}Ga_2} \longrightarrow 1\,\mathrm{Cu} + 2\,\mathrm{Ga}$
72. $1\,\mathrm{Cu_3Ga_2} + 4\,\mathrm{Ga} \longrightarrow 3\,\mathrm{Cu_{1-\mathit{x}}Ga_2}$
73. $1\,\mathrm{Cu_{1-\mathit{x}}Ga_2} \longrightarrow \frac{1}{3}\,\mathrm{Cu_3Ga_2} + \frac{4}{3}\,\mathrm{Ga}$
74. $1\,\mathrm{Cu_{1-\mathit{x}}Ga_2} + 1\,\mathrm{Ga} \longrightarrow 1\,\mathrm{Cu_{1-\mathit{x}}Ga_2} + 1\,\mathrm{Ga}$
75. $2\,\mathrm{In} + 3\,\mathrm{Se} \longrightarrow 1\,\gamma\text{-}\mathrm{In_2Se_3}$
76. $3\,\gamma\text{-}\mathrm{In_2Se_3} \longrightarrow 1\,\mathrm{In_6Se_7} + 2\,\mathrm{Se}$
77. $1\,\mathrm{In_6Se_7} + 2\,\mathrm{Se} \longrightarrow 3\,\gamma\text{-}\mathrm{In_2Se_3}$
78. $1\,\gamma\text{-}\mathrm{In_2Se_3} \longrightarrow 2\,\mathrm{In} + 3\,\mathrm{Se}$

C Additional results

Table 24: Computed frequency factor and activation energy for several reactions using the computed concentrations from the simulated measurements.

	Frequency factor k_0			Activation energy Ea			
Reaction	Computed	Model	95%-CI	Computed	Model	95%-CI	Data points
4. 9-x Cu + 4 Ga $\longrightarrow$ 1 Cu_9Ga_4	2.66	500	$[-3.1096, 8.4303]$	66038	30000	$[52767.3, 79310.4]$	17
13. 2-x Cu + 1 Se $\longrightarrow$ 1 $Cu_{2-x}Se$	$9.03 \cdot 10^{-3}$	4000	$[-3.50 \cdot 10^{3}, 2.16 \cdot 10^{-2}]$	10000	60000	$[2307.52, 17692.5]$	32
15. 2 Ga + 3 Se $\longrightarrow$ 1 Ga_2Se_3	$7.86 \cdot 10^{-6}$	1000	$[-1.36 \cdot 10^{-4}, 1.52 \cdot 10^{-4}]$	10000	50000	[-116522,136522]	12
18. 1 $Cu_{2-x}Se$ + 1 Ga_2Se_3 $\longrightarrow$ 2 α-CGS	$9.27 \cdot 10^{-4}$	5000	$[-0.34557, 0.34742]$	10000	50000	$[-2.62 \cdot 10^{6}, 2.64 \cdot 10^{6}]$	34
25. 1 Cu_7Ga_4 $\longrightarrow$ 1 Cu_9Ga_4 + 1 Ga	0.93	1000	$[-0.68811, 2.5409]$	10000	60000	$[-666.232, 20666.2]$	17
27. 2 Cu_3Ga_2 $\longrightarrow$ 1 Cu_7Ga_4 + 1 Ga	0.37	500	$[6.11 \cdot 10^{-2}, 0.80]$	10000	55000	$[4019.96, 15980]$	17
39. 1 In_4Se_3 + 1 Se $\longrightarrow$ 4 InSe	25.9	3500	$[-214.91, 732.85]$	62908	60000	$[55613.7, 70201.6]$	5
42. 6 InSe + 1 Se $\longrightarrow$ 1 In_6Se_7	15.9	4000	$[-99.477, 131.22]$	66722	60000	$[31111.5, 102333]$	10
44. 1 In_6Se_7 + 2 Se $\longrightarrow$ 3 β-In_2Se_3	124	3000	$[-241.2, 489.65]$	80486	50000	$[63513.7, 97458.4]$	14
50. 1 Cu_9Ga_4 + Se $\longrightarrow$ 1 $Cu_{2-x}Se$ + 2 Cu_3Ga_2	$3.16 \cdot 10^{-3}$	5000	$[5.19 \cdot 10^{-4}, 5.80 \cdot 10^{-3}]$	16881	60000	$[13424.7, 20337.4]$	30
51. 1 Cu_3Ga_2 + 4 Se $\longrightarrow$ 2 $Cu_{2-x}Se$ + 2 GaSe	$1.17 \cdot 10^{-3}$	5000	$[-1.23 \cdot 10^{-4}, 2.22 \cdot 10^{-3}]$	15852	60000	$[12030.8, 19675]$	38
52. 1 $Cu_{2-x}Se$ + 2 InSe + 1 Se $\longrightarrow$ 2 α-CIS	$1.18 \cdot 10^{-5}$	4600	$[-2.32 \cdot 10^{-4}, 2.56 \cdot 10^{-4}]$	10000	60000	$[-73937.8, 93937.8]$	12
53. 1 $Cu_{2-x}Se$ + 1 β-In_2Se_3 $\longrightarrow$ 2 α-CIS	$1.54 \cdot 10^{-3}$	4600	$[1.85 \cdot 10^{-3}, 4.92 \cdot 10^{3}]$	15931	40000	$[3221.77, 28641.8]$	29
54. 1 $Cu_{2-x}Se$ + 3 GaSe $\longrightarrow$ 2 α-CGS	$2.13 \cdot 10^{-4}$	5000	$[-2.77 \cdot 10^{4}, 7.02 \cdot 10^{-4}]$	10000	40000	$[-1019.69, 21019.7]$	34
62. 1 α-CIS + 1 $Cu_{2-x}Se$ $\longrightarrow$ 1 α-CIS + 1 Se	$4.41 \cdot 10^{-4}$	4000	$[-3.53 \cdot 10^{-2}, 3.62 \cdot 10^{-2}]$	22944	60000	$[-547168, 593057]$	14
63. 1 α-CGS + 1 $Cu_{2-x}Se$ $\longrightarrow$ 1 α-CGS + 1 Se	$7.75 \cdot 10^{-3}$	4000	$[-1.09 \cdot 10^{-2}, 2.64 \cdot 10^{-2}]$	30259	60000	$[15355.1, 40686.5]$	26

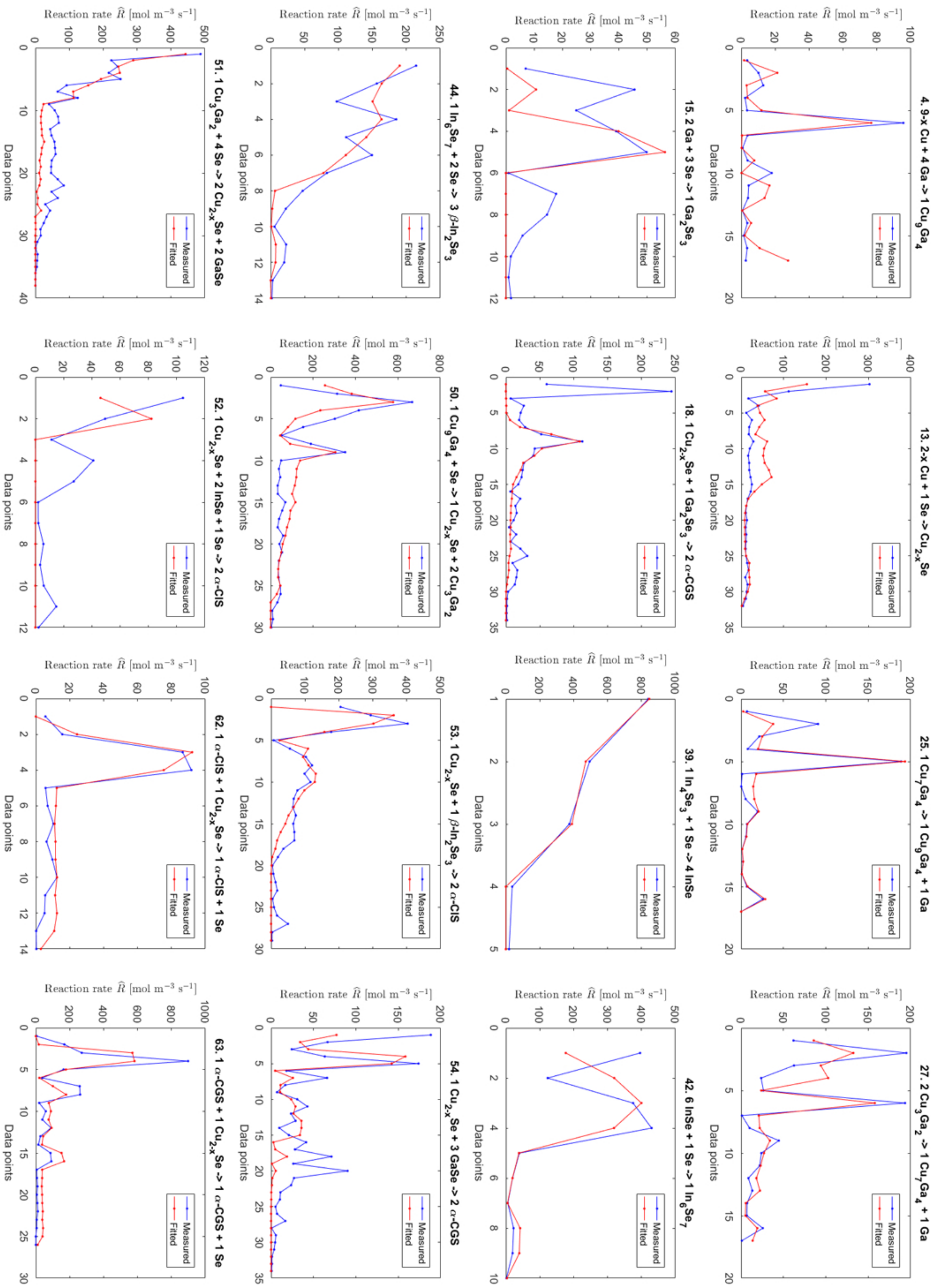


Figure 62: Fitted Arrhenius equations at each data point for a set of reactions from the computed concentrations.

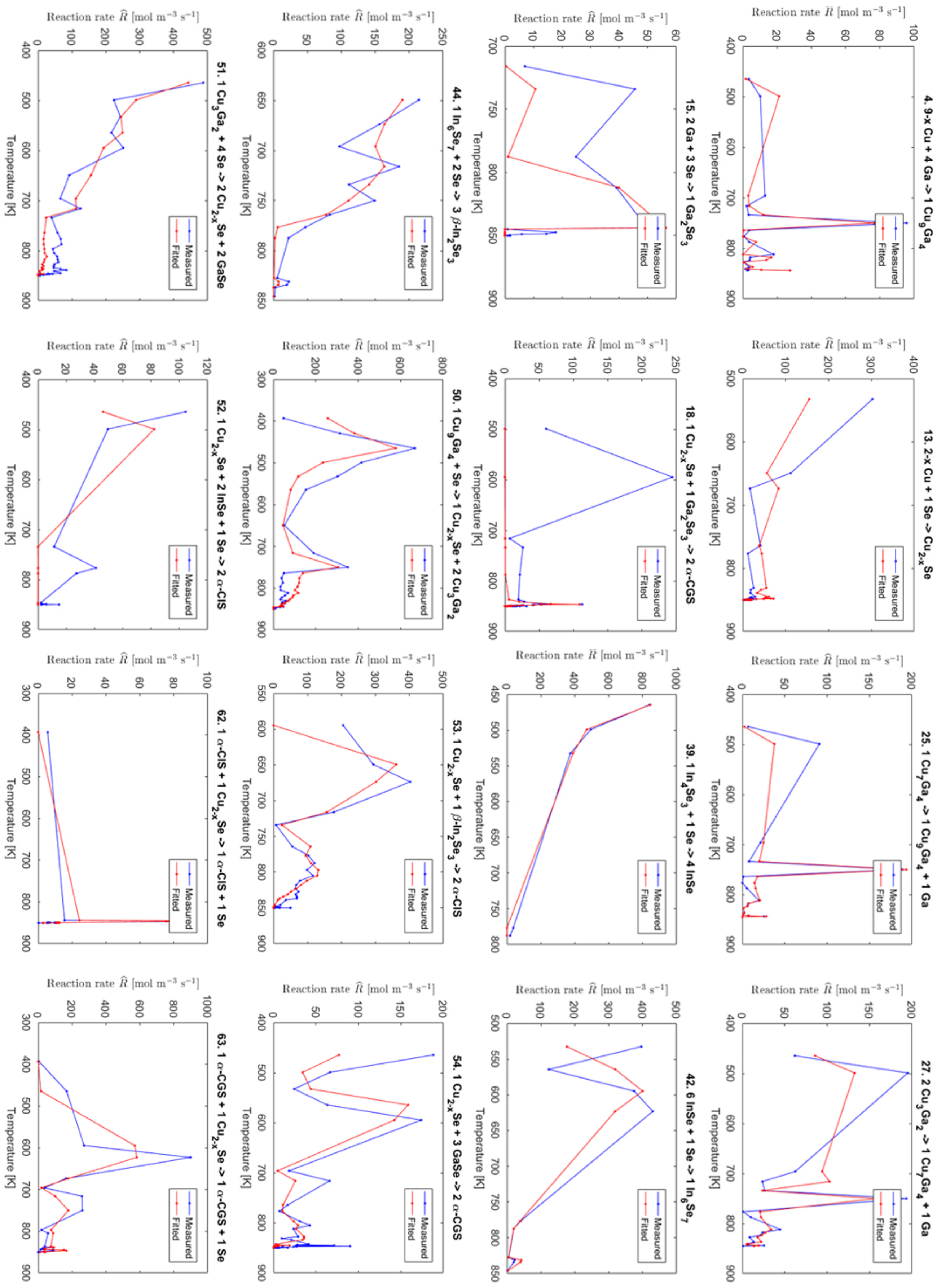


Figure 63: Fitted Arrhenius equations at each temperature point for a set of reactions from the computed concentrations.

D Algorithms

D.1 Tree construction in peak detection

CP : Current Points, set of all the leaves of the tree,
NP : Next Points, set of all the points that have to be added to the tree,
PTA : Points To Add, set of all the points that are closest to $CP(i)$.

Algorithm 1 Tree construction

```
Create an initial set of $CP$ from largest scale.
for All scales $j$ starting at the one but largest scale do
    Get $NP$ for scale $j$.
    $i = 1$;
    while $i \leq$ amount of $CP$ do
        Calculate the distance between $CP$ and all $NP$:
        $\Delta = |NP - CP(i)|$;
        if $\#(\Delta \leq \delta) = 1$ then
            if This point is closer to some other point in $CP$ then
                Do nothing and continue to the next point.
                $i = i + 1$;
            else
                Add point to the tree and update $CP$.
                Remove point form $NP$.
                $i = i + 1$;
            end if
        end if
        if $\#(\Delta \leq \delta) > 1$ then
            Create $PTA$.
            if Some points in $PTA$ are closer to other points in $CP$ then
                Remove point from $PTA$
            end if
            if PTA is empty? then
                Do nothing and continue to the next point.
                $i = i + 1$;
            else
                Add point to the tree and update $CP$.
                Remove point form $NP$.
            end if
        end if
        if $\#(\Delta \leq \delta) < 1$ then
            $i = i + 1$;
        end if
    end while
    if Some points left in $NP$ then
        Create a new branch in the tree and add the points to $CP$.
    end if
end for
```

D.2 Brute force

D.2.1 One variable

```
function [xval,fval] = FindMinBF(fun,left,right,n,m,type)
% Finds minimum of a 1D function 'fun' between the boundaries 'left' and
% 'right'. 'n' is the amount of loops and type is either linear/log spaced

if nargin < 6
    type = 'linear';
end
if nargin < 5
    m = 100;
end

for nn = 1:n
    % Create x
    if strcmp(type,'linear')
        x = linspace(left,right,m);
    elseif strcmp(type,'log')
        x = logspace(log10(left),log10(right),m);
    end

    % Find minimum
    ff = zeros(m,1);
    for i = 1:m
        ff(i) = fun(x(i));
    end
    [Val,Index] = min(ff);
    % Update left and right
    if nn ≠ n
        if Index == 1
            left = x(Index);
            right = x(Index+1);
        elseif Index == m
            left = x(Index−1);
            right = x(Index);
        else
            left = x(Index−1);
            right = x(Index+1);
        end
        m = 10; % for next cycle
    end
end
xval = x(Index);
fval = Val;
```

D.2.2 Two variables

```
function [xval,fval] = FindMinBF2(fun,left,right,n,m,type)
% Finds minimum of a 2D function 'fun' between the boundaries 'left' and
% 'right' (1x2 vectors). 'n' is the amount of loops and type creates either
% linear or log spaced variables.

if nargin < 6
    type = 'linear';
end
if nargin < 5
    m = 100;
end

if length(left) ≠ 2 || length(right) ≠ 2
    error('Left and right bounds should be a 1x2 (or 2x1) vector!');
end

for nn = 1:n
    % Create x1 and x2
    if strcmp(type,'linear')
        x1 = linspace(left(1),right(1),m);
        x2 = linspace(left(2),right(2),m);
    elseif strcmp(type,'log')
        x1 = logspace(log10(left(1)),log10(right(1)),m);
```

```
        x2 = logspace(log10(left(2)),log10(right(2)),m);
    end

    % Find minimum
    ff = zeros(m,m);
    for i = 1:m
        for j = 1:m
            ff(i,j) = fun([x1(i),x2(j)]);
        end
    end
    [Val,Index] = min(ff(:));
    [Row,Col] = ind2sub(size(ff),Index);
    % Update left and right
    if nn ≠ n
        if Row == 1
            if Col == 1
                left(1) = x1(1);
                left(2) = x2(1);
                right(1) = x1(2);
                right(2) = x2(2);
            elseif Col == m
                left(1) = x1(1);
                left(2) = x2(m−1);
                right(1) = x1(2);
                right(2) = x2(m);
            else
                left(1) = x1(1);
                left(2) = x2(Col−1);
                right(1) = x1(2);
                right(2) = x2(Col+1);
            end
        elseif Row == m
            if Col == 1
                left(1) = x1(m−1);
                left(2) = x2(1);
                right(1) = x1(m);
                right(2) = x2(2);
            elseif Col == m
                left(1) = x1(m−1);
                left(2) = x2(m−1);
                right(1) = x1(m);
                right(2) = x2(m);
            else
                left(1) = x1(m−1);
                left(2) = x2(Col−1);
                right(1) = x1(m);
                right(2) = x2(Col+1);
            end
        else
            if Col == 1
                left(1) = x1(Row−1);
                left(2) = x2(1);
                right(1) = x1(Row+1);
                right(2) = x2(2);
            elseif Col == m
                left(1) = x1(Row−1);
                left(2) = x2(m−1);
                right(1) = x1(Row+1);
                right(2) = x2(m);
            else
                left(1) = x1(Row−1);
                left(2) = x2(Col−1);
                right(1) = x1(Row+1);
                right(2) = x2(Col+1);
            end
        end
        m = 10; % for next cycle
    end
end
xval = [x1(Row),x2(Col)];
fval = Val;
```

References

[1] A. Fortes, E. Suard, M. Lemèe-Cailleau, C. Pickard and R. Needs, *Crystal Structure of Ammonia Monohydrate Phase II*, 2009, Journal of the American Chemical Society, Volume 131, Issue 37, p. 1350813515, DOI: 10.1021/ja9052569.

[2] S. Flower and G. Saunders, *The elastic behavior of indium under pressure and with temperature up to the melting point*, 1990, Philosophical Magazine, Part B, Volume 62, Issue 3, p. 311-328, DOI: 10.1080/13642819008208632 .

[3] C. Bernuy-Lopez, M. Allix, C. Bridges, J. Claridge and M. Rosseinsky,$Sr_2MgMoO_{6-\delta}$*: Structure, Phase Stability, and Cation Site Order Control of Reduction*, 2007, Chemistry of Materials, Volume 19, Issue 5, p. 1044-1052, DOI: 10.1021/cm0624116.

[4] F. Laves and H. Wallbaum, *Ueber neue Vertreter des Ni As-Type und ihre kristallchemische Bedeutung*, 1942, Zeitschrift fuer Angewandte Mineralogie, Volume 4, p. 17-46.

[5] T. Rajasekharan and K. Schubert,*Crystal Structure of Cu11In9*, 1981, Zeitschrift für Metallkunde, Volume 72, p. 275-278.

[6] W. Keppner, T. Klas, W. Körner, R. Wesche and G. Schatz, *Compount Formation at Cu-In Thin-Film Interfaces Detected by Perturbed* $\gamma - \gamma$ *Angular Corrections*, 1985, Physical Review Letters, Volume 54, Issue 21, p. 2371-2374, DOI: 10.1103/PhysRevLett.54.2371.

[7] R. Stokhuyzen, J. Brandon, P. Chieh and W. Pearson, *Copper-Gallium,* γCu_9Ga_4, 1974, Acta Crystallography, Volume B30, p. 2910-2911, DOI: 10.1107/S0567740874008478.

[8] O. Tikhomirova, M. Pikunov, I. Marchukova, I. Tochenova and I. Izotova, *An investigation of structural transformation during the solidification of copper-gallium alloys*, 1969, Fiziko-Khimicheskaya Mekhanika Materialov, Volume 5, Issue 4, p. 455-458, DOI: 10.1007/BF00730341.

[9] U. Schwarz, H. Hillebrecht, H. J. Deiseroth and R. Walther, In_4Te_3 *und* In_4Se_3*: Neubestimmung der Kristallstrukturen, druckabhängiges Verhalten und eine Bemerkung zur Nichtexistenz von* In_4S_3, 1995, Zeitschrift für Kristallographie, Volume 210, Issue 5, p. 342-347, DOI: 10.1524/zkri.1995.210.5.342.

[10] P. Gomes da Costa, R. Dandrea, R. Wallis, and M. Balkanski, *First-principles study of the electronic structure of* γ*-InSe and* β*-InSe*, 1978, Physical Review B, Volume 48, p. 14135-14141, DOI: 10.1103/PhysRevB.48.14135.

[11] R. Walther and H. Deiseroth, *Redetermination of the crystal structure of hexaindium heptaselenide,* In_6Se_7, 1995, Zeitschrift für Kristallographie, Volume 210, Issue 5, p. 359, DOI: 10.1524/zkri.1995.210.5.359.

[12] K. Osamura, Y. Murakami and Y. Tomiie, *Crystal structures of* α *and* β*-Indium selenide,* In_2Se_3, 1966, Journal of the Physical Society of Japan, Volume 21, p. 1848-1848, DOI: 10.1143/JPSJ.21.1848.

[13] A. Likforman, D. Carr and R. Hillel, *Structure Cristalline du Séléniure d'Indium* In_2Se_3, 1978, Acta Crystallographica B, Volume 34, p. 1-5 DOI: 10.1107/S0567740878002174.

[14] E. Uhlemann and E. Frank, *Stabilitätsuntersuchungen an Chelaten sterisch behinderter* β*-Diketone*, Zeitschrift für anorganische und allgemeine Chemie, 1965, Volume 340, Issue 5-6, p. 319324, DOI: 10.1002/zaac.19653400510.

[15] V. Milman, *Klockmannite, CuSe: structure, properties and phase stability from ab initio modeling*, Acta Crystallographica B, 2002, Volume 58, p. 437-447, DOI: 10.1107/S0108768102003269.

[16] W. Borchert, *Gitterumwandlungen im System* $Cu_{2-x}Se$, Zeitschrift fur Kristallographie, 1954, Volume 106,p. 5-24, DOI: 10.1524/zkri.1954.106.16.5.

[17] K. Cenzual, L. Gelato, M. Penzo and E. Parthe, *Inorganic structure types with revised space groups.I*, 1991, Acta Crystallographica B, Volume 47, p. 433-439, DOI: 10.1107/S0108768191000903.

[18] G. Ghemard, S. Jaulmes, J. Etienne and J. Flahaut, *Structure de la phase ordonnée du sesquiséléniure de gallium,* Ga_2Se_3, 1983, Acta Crystallographica C, Volume 39, p. 968-971, DOI: 10.1107/S0108270183007039.

[19] P. James and M. Lavik, *The crystal structure of* $MoSe_2$, 1963, Acta Crystallographica, Volume 16, p. 1183, DOI: 10.1107/S0365110X6300311X.

[20] H. Hahn, G. Stoerger, G. Frank, A. Meyer and W. Klingler, *Ueber einige ternaere Chalkogenide mit Chalkopyritstruktur*, 1953, Zeitschrift für Anorganische und Allgemeine Chemie, Volume 271, Issue 3-4, p. 153-170, DOI: 10.1002/zaac.19532710307.

[21] A. Wooten, D. Werder, D. Williams, J. Casson and J. Hollingsworth, *Solution-Liquid-Solid Growth of Ternary Cu-In-Se Semiconductor Nanowires from Multiple- and Single-Source Precursors*, 2009, Journal of the American Chemical Society, Volume 131, Issue 44, p. 16177-16188, DOI: 10.1021/ja905730n.

[22] L. Mandel, R. Tomlinson and M. Hampshire, *Crystal Data for* $CuGaSe_2$, 1977, Journal of Applied Crystallography, Volume 10, p. 130-131, DOI: 10.1107/S0021889877013065.

[23] J. de Cloet, *Modeling the selenization process for the production of CIGS solar cells*, 2016, TNO/Solliance.

[24] F. Hergert, *Chemical formation reactions for* $Cu(In, Ga)Se_2$ *and other chalcopyrite compounts -An in-situ x-ray diffraction study and crystallographic models*, 2007, University of Erlangen-Nürnberg.

[25] J. Gregoire, D. Dale and R. van Dover, *A wavelet transform algorithm for peak detection and application to powder x-ray diffraction data*, 2011, American Institute of Physics, DOI: 10.1063/1.3505103.

[26] W. Callister and D. Rethwisch, *Fundamentals of Material Science and Engineering*, 2013, Wiley, 4th edition, Chapter. 3.

[27] V. Peckarsky and P. Zavalij, *Fundamentals of Powder Diffraction and Structural Characterization of Materials*, 2009, Springer, DOI: 10.1007/978-0-387-09579-0.

[28] Y. Wang, J. Zhang, H. Xu, Z. Lin, L. Daemen, Y. Zhao and L. Wang, *Thermal equation of state of copper studied by high P-T synchrotron x-ray diffraction*, February 2009, Applied Physics Letters, Volume 94, Issue 7, DOI: 10.1063/1.3085997.

[29] D. Rong, C. Zhongjun, C. Quan, F. Jianlong, G. Yu and W. Zhonghua, *Application of Mythen Detector: In-situ XRD Study on The Thermal Expansion Behavior of Metal Indium*, 2016, Science China Physics, Mechanics & Astronomy, Volume 59, Issue 7 DOI: 10.1007/s11433-015-0436-0.

[30] J. Tinevez, *A MATLAB class to represent the tree data structure*, 2015, http://tinevez.github.io/matlab-tree/index.html.

[31] H. Rietveld, *A Profile Refinement Method for Nuclear and Magnetic Structures*, 1969, Journal of Applied Crystallography, Volume 2, p. 65-71, DOI: 10.1107/S0021889869006558.

[32] U. Shmueli et al., *International Table for Crystallography Volume B: Reciprocal Space*, 2005, Springer, DOI: 10.1107/97809553602060000102.

[33] E. Prince et al., *International Table for Crystallography Volume C: Mathematical, Physical and Chemical Tables*, 2005, Springer, DOI: 10.1107/97809553602060000103.

[34] W. Dollase, *Correction of intensities for preferred orientation in powder diffractometry: Application of the March model*, 1986, Journal of Applied Crystallography, Volume 19, p. 267-272, DOI: 10.1107/S0021889886089458.

[35] R. Byrd, J. Gilbert and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, 2000, Mathematical Programming, Volume 89, Issue 1, p. 149185, DOI: 10.1007/PL00011391.

[36] R. Byrd, M. Hribar and J. Nocedal, *An interior point method for large scale nonlinear programming*, 1997, SIAM Journal of Optimization, Volume 9, Issue 4, p. 877-900, DOI: 10.1137/S1052623497325107.

[37] E. de Klerk, C. Roos and T. Terlaky, *Nonlinear Optimization*, 2006, University of Waterloo.

[38] A. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly and R. Marti, *Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization*, 2006, INFORMS Journal of Computing, Volume 19, Issue 3, p. 328-340, DOI: 10.1287/ijoc.1060.0175.

[39] A. Rinnooy Kan and G. Timmer, *Stochastic Global Optimization Methods; part I: Clustering Methods*, 1987, Mathematical Programming, Volume 37 p. 27-56, DOI: 10.1007/BF02592070.

[40] A. Rinnooy Kan and G. Timmer, *Stochastic Global Optimization Methods; part II: Multi Level Methods*, 1987, Mathematical Programming, Volume 37 p. 57-78, DOI: 10.1007/BF02592071.

[41] M. Locatelli, F. Schoen, *Random Linkage: a Family of Acceptance/Rejection Algorithms for Global Optimization*, 1999, Mathematical Programming, Volume 85, Issue 2, p. 379-396, DOI: 10.1007/s101070050062.

[42] D. Goldberg, *Genetic Algorithms in Search Optimization & Machine Learning*, 1989, Addison-Wesley Publishing Company, Inc..

[43] R. Young, *The Rietveld Method*, 1993, Oxford University Press.

[44] C. Muzzilio, C. Campbell and T. Anderson, *CuGaIn thermodynamics: experimental study, modeling, and implications for photovoltaics*, 2016, Journal of Material Science, Volume 51, p. 3362-3379, DOI: 10.1007/s10853-015-9651-3.

[45] M. Purwins, R. Enderle, M. Schmid, P. Berwian, G. Mller, F. Hergert, S. Jost and R. Hock, *Phase relations in the ternary CuGaIn system*, 2007, Thin Solid Films, Volume 515, Issue 15, p. 5895-5898, DOI: 10.1016/j.tsf.2006.12.090.

[46] N. B´orlin, *Nonlinear Optimization: Least Squares Problems and The Gauss-Newton method*, 2007, Umeå University.

[47] T. Duffy, *Crystal geometry functions for XRD*, Princeton University, High-Pressure Mineral Physics and Material Science Laboratory, http://duffy.princeton.edu/sites/default/files/pdfs/links/xtalgeometry.pdf

[48] E. Brandes and G. Brook, *Smithells Metals Reference Book*, 2013, Elsevier, p. 4-16.

[49] T. Hahn et al., *International Table for Crystallography Volume A: Space-Group symmetry*, 2005, Springer, DOI: 10.1107/97809553602060000100.