

Improving qualitative and quantitative SEM EDS analysis with open-source data processing

BRYNJAR MORKA MÆHLUM

Nanotechnology student

The Nanotechnology Specialization Project, TFY4520

Department of Physics
Norwegian University of Science and Technology
Norway
December 19, 2022

Abstract

Energy dispersive spectroscopy (EDS) is a widely used analytical technique for identifying the elemental composition of materials. The accuracy of EDS analysis can be affected by various factors, including instrumental parameters and setup, sample preparation, and the various steps involved in the analysis process. In this study, the aim was to improve the accuracy of EDS analysis through the development of new or improved data processing techniques. The approach involved both qualitative and quantitative analysis. The results demonstrated some potential improvement of the EDS analysis, including improved calibration of the used setup, an idea for new background modelling, and Jupyter Notebooks to improve the transparency of the analysis. While the work represents a negligible contribution to the field of EDS analysis, it has the potential to help an analyst to improve their understanding of the analysis process.

In this work, a scanning electron microscope (SEM) was used to acquire EDS data, which was analyzed in the software from the detector manufacturer (AZtec), with the open-source Python package HyperSpy, and with custom Python code, using Jupyter Notebooks, SciPy, NumPy, and Plotly. One of the results is a suggested improvement to calibration of the setup used, an Oxford Instruments Xmax 80 mm² detector on a FEI SEM APREO, located at the NanoLab at Norwegian University of Science and Technology (NTNU). The new calibration is based on peak and background modelling done with Python. The difference between the current and suggested calibration is to increase the dispersion from 10 eV/channel to 10.03 eV/channel, and increase the zero-offset from 20 channels to 21.1 channels. The suggested calibration is shown to have a root-mean-square deviation of 1/3 of the current calibration, on the peaks acquired in this study.

Some of the code used in this study is attached in the appendix, and all code is available on GitHub. Two repositories are available, one for all the code produced in this study, and one distilled version with selected parts of the code. The repositories include the data used, and are intended to be used for people who want to understand parts of the steps in the analysis process better. The code can do interactive plotting, peak and background modelling, calibration of the energy scale, and simple quantitative analysis. Links to the repositories are:

<https://github.com/brynjarmorka/eds-analysis>

<https://github.com/brynjarmorka/eds-analysis-final>

Sammendrag

Energidispersivrøntgenspektroskopi (EDS) er en mye brukt analysemetode for å identifisere den elementære sammensetningen av materialer. Nøyaktigheten til EDS-analyser kan påvirkes av ulike faktorer, inkludert instrumentparametere og -oppsett, prøveforberedelse og ulike trinn i analyseprosessen. I denne studien var målet å forbedre EDS-analysen gjennom utvikling av nye eller forbedrede databehandlingsteknikker. Tilnærmingen involverte både kvalitativ og kvantitativ analyse. Resultatene viser muligheter for å forbedre EDS-analysen, inkludert forbedret kalibrering av det brukte oppsettet, en idé om ny bakgrunnsmodellering og Jupyter Notebooks for å øke transparensen i analysen. Selv om arbeidet utgjør et neglisjerbart bidrag til feltet EDS-analyse, har arbeidet potensialet til å øke forståelsen av analyseprosessen for brukere.

I denne studien ble et skaneelektronmikroskop (SEM) brukt til å anskaffe EDS-data, som ble analysert i programvaren AZtec fra detektorprodusenten, med den åpne kildekoden HyperSpy og tilpasset Python-kode, med hjelp fra Jupyter Notebooks, SciPy, NumPy og Plotly. Et av resultatene er et forslag om å forbedre kalibreringen av oppsettet som ble brukt, en Oxford Instruments Xmax 80 mm² detektor på en FEI SEM APREO, som er på NanoLab ved Norges teknisk-naturvitenskapelige universitet (NTNU). Den nye kalibreringen baserer seg på røntgentopp- og bakgrunnsmodellering gjort med Python. Forskjellen mellom den gjeldende og foreslalte kalibreringen er å øke dispersjonen fra 10 eV/kanal til 10,03 eV/kanal, og øke nullpunktjusteringen fra 20 kanaler til 21,1 kanaler. Den foreslalte kalibreringen viser seg å ha et gjennomsnittlig kvadratisk avvik på 1/3 av den gjeldende kalibreringen, på de røntgentoppene som ble analysert i denne studien.

Noen av koden som ble brukt i denne studien er vedlagt i appendiks, og all koden er tilgjengelig på GitHub. Koden kan plotte spekter, modellere karakteristiske røntgenstråletopper og bakgrunnsstråling, kalibrere energiskalaen og gjøre enkel kvantitativ analyse. To reservoar (repositories) er tilgjengelige, én for all koden som ble produsert i denne studien, og én destillert versjon med utvalgte deler av koden. Reservoarene inneholder også datagrunnlaget, og de er tilgjengelig primært for at en bruker skal kunne lettere forstå deler av EDS-analysen. Reservoarene er tilgjengelig på:

<https://github.com/brynjarmorka/eds-analysis>

<https://github.com/brynjarmorka/eds-analysis-final>

Preface and acknowledgements

This is my specialization project done in the 9th semester of the master's program in Nanotechnology at NTNU. During my studies I have tried to focus on the development of my skills in programming, and this project is a continuation of that. I appreciate the opportunity to combine programming and nanotechnology in this project, and I look forward to continuing the work in my master's thesis.

For guidance and support during the project, I would like to thank my supervisor, Professor *Antonius T.J. van Helvoort*, at the TEM Gemini Center, Department of Physics at NTNU. His enthusiasm has been an inspiration, and our weekly meeting have been a great success.

The SEM EDS data used in this project is from NTNU NanoLab. Thanks to the Norwegian Research Council for funding the NorFab and NanoLab, and to the NanoLab staff for providing an excellent working environment and for their help with the data acquisition.

I am grateful to my teachers at Forskerlinja at Vestby videregående skole, who inspired me to study nanotechnology and taught me how to write scientific reports. I am even more grateful to Timini and Kulli8, my time at Nanotechnology would probably have been shorter without you. I want to thank my family for the support and holidays home in Hvitsten. The COVID-time would have sucked way more if I had not been busy as the leader of Timini, joined Organisasjonskollegiet, and taken the year of telemark skiing and climbing in Innsbruck. I want to thank my friends for all the fun we have had, the good food we have made and the adventures we have shared.

Last but not least, I wanna thank me.

A handwritten signature in black ink, appearing to read "Brynjar Morka Mæhlum".

Brynjar Morka Mæhlum

Trondheim, 17.12.2022

Abbreviations

AZ	AZtec
BSE	Backscattered Electrons
CL	Cliff-Lorimer
EDS	Electron Dispersive X-ray Spectroscopy
EDSX	Electron Dispersive X-ray Spectroscopy
EDX	Electron Dispersive X-ray Spectroscopy
GUI	Graphical User Interface
HS	HyperSpy
RMSD	Root Mean Square Deviation
SE	Secondary Electrons
SEM	Scanning Electron Microscope
TEM	Transmission Electron Microscope
XRB	The X-ray Data Booklet

Contents

1	Introduction	1
2	Theory	4
2.1	Theoretical view on characteristic X-rays	4
2.1.1	Formation of characteristic X-rays	4
2.1.2	Naming convention	6
2.1.3	Energy and intensity	8
2.2	Empirical view on characteristic X-rays	13
2.2.1	From lines to peaks	13
2.2.2	Intensity	13
2.3	Data processing	15
2.3.1	HyperSpy	15
2.3.2	Cliff-Lorimer and the k-factors	15
2.3.3	Calibration of the spectrum	16
2.4	Detection system	16
2.4.1	EDS detector	16
3	Method	17
3.1	Materials and specimen	17
3.2	The microscope and the detector	18
3.3	Data processing	18
3.3.1	AZtec and HyperSpy	19
3.3.2	Self-made data processing	19
4	Results	21
4.1	Qualitative analysis of the spectra	21

4.1.1	Initial plot	21
4.1.2	Calibration	25
4.1.3	Spectra from the studied materials	29
4.2	Quantitative results	34
5	Discussion	38
5.1	General results from the spectra	38
5.1.1	Peak intensities	38
5.1.2	The background	41
5.1.3	The strays and artifacts	42
5.2	Quantification in AZtec	44
5.3	Analysis steps in HyperSpy	45
5.3.1	Loading the data and specifying the elements	45
5.3.2	Removing the background linearly	47
5.3.3	Quantification after linear background removal	48
5.3.4	Removing the background with model fitting	48
5.3.5	Quantification after model fitting	50
5.3.6	Calibrating the spectrum with the HyperSpy model	50
5.4	Peak and background modelling	50
5.5	Calibration	52
5.6	Analysis failure	54
6	Conclusion and future work	56
6.1	Main findings	56
6.2	Future work	58
Appendices		62
A	HyperSpy quantification	63
B	Peak and background modelling	66

Chapter I

Introduction

The main goal of this project is to improve EDS analysis. EDS analysis is a compositional imaging technique with high spatial resolution, which is used for the elemental characterization of materials at the microscopic scale. EDS is a powerful technique used in many fields, for example materials science, geology, biology and environmental science. The technique is used to identify the chemical composition of samples, study microstructures and phases, and understand the properties of materials. In general, EDS analysis involves four steps. The first step is data acquisition, where the data consists of a list of X-ray energies and the corresponding counts of X-rays detected at each energy. The second step is calibration of the energy scale. The third step is qualitative analysis, where the elements in the sample are identified, as well as artifacts in the data. The fourth step is quantitative analysis, where the concentration of each element in the sample is calculated.

Improving EDS analysis would benefit many fields, and there are many ways to improve EDS analysis. The improvement could be both qualitative and quantitative. The accuracy of quantitative EDS, which is how accurate the concentration of each element is, is limited by a number of factors, and the accuracy is typically in the range of 5-10% [1,2]. One way to improve EDS is to make the analysis more transparent, which would allow users to understand the analysis better and adjust it to their needs. One example of this is when a user adjusts the peak model to fit the data better, it is likely that they can improve the accuracy of the analysis. A second option is to improve the input parameters of the analysis by control checking the instrument with a known sample, and also alter the analysis steps to enhance the results. If the instrument is not calibrated correctly, the results will probably be worse than they could be. A third way is to make the quantitative analysis more accurate, which would improve EDS analysis.

Most of the time used in this project was spent on trying to understand some methods of data processing in EDS analysis. The different methods were applied on the same data sets to see how they influence the qualitative analysis. Towards the end the Cliff-Lorimer quantification in HyperSpy was applied on some data to see how data processing

influence the quantitative analysis. Increasing the transparency was done by doing the analysis in open-source Jupyter Notebooks. With these Notebooks, which are available on GitHub¹², an analyst can easily see what is happening in the different steps of the analysis. The Notebooks also allow the user to adjust the analysis to their needs, for example by changing the model of the peaks and background. In this project, the main focus have been trying to improve the transparency of the analysis. A main problem statement was formulated, and it was broken down into five sub-problems.

Main problem statement. How do different spectroscopy data processing influence the qualitative analysis, and how do they affect the quantitative Cliff-Lorimer analysis in HyperSpy?

The first subproblem is about the analysis in AZtec. AZtec is the software provided by Oxford Instruments for analyzing EDS data. AZtec is a black box, which means that the code is hidden, and the user does not know what is happening inside the software. The analysis was done both on quantification and to see what the software identifies. It was expected that AZtec would have functionality for some data processing, while also lacking some functionality. Since it is a black box, the user has few options to change the analysis, and verifying the results is difficult.

Sub-problem 1. How accurate is the quantification in AZtec, and what does it identify?

The second subproblem is about the data processing in HyperSpy. HyperSpy is an open-source Python package for analyzing EDS data, and it is used in this project. Since HyperSpy is open-source, the code is available, and the user can see what is happening inside the software. The Python package is developed by a community of scientists, where the starting point was TEM data analysis. HyperSpy have functionality for EDS analysis, but is somewhat limited on SEM EDS analysis. However, many of the methods implemented for TEM data analysis is equally applicable for SEM data analysis.

Sub-problem 2. What are done with the data at the different steps in the analysis when using HyperSpy?

After analyzing the data with AZtec and HyperSpy, the third subproblem is dealing with code developed in this project for analyzing the data. The analysis is done in Python with Jupyter Notebooks, and are dealing with making a model of the peaks and background. Making a model of the spectrum is an important step in the EDS analysis.

¹<https://github.com/brynjarmorka/eds-analysis>

²<https://github.com/brynjarmorka/eds-analysis-final>

Sub-problem 3. How can the peaks and the background be modelled?

The fourth subproblem is about the calibration of the energy scale, where methods from both HyperSpy and the self-developed code are compared to the calibration in AZtec. Calibration is a crucial step in the EDS analysis, and it is important to have a good calibration to get accurate results.

Sub-problem 4. How good is the calibration in AZtec and HyperSpy?

The fifth and last subproblem is about the analysis failing. During the project some choices were made, and some choices were not optimal. The subproblem is both about when the analysis fails, and trying to identify why it fails. The identification of such fails is important to improve the analysis. Sometimes the analysis fails because of the data, and sometimes it fails because of the user making poor choices, and sometimes it fails because of bugs in the software.

Sub-problem 5. When does the analysis fail, both in AZtec and HyperSpy?

This report is structured as follows. The pre chapters contain the abstract in English and Norwegian, the preface and acknowledgements, and the abbreviations. This introduction chapter contains the motivation for the project, the main problem statement, and the sub-problems. The theory chapter contains the physics of X-rays and empirical adjustments in the analysis, a section on data processing, and a section about the hardware in an EDS setup. The method chapter explains how the data was collected. The result chapter contains qualitative and quantitative results. The discussion chapter follow the structure of the sub-problems, and discuss both the methods and the results of the analysis consecutively. The conclusion chapter summarizes the report with an answer to the main problem statement, and provides ideas for further work. The appendix contains a part of the code used in the analysis, which is also available on GitHub³.

³<https://github.com/brynjarmorka?tab=repositories>

Chapter 2

Theory

Energy dispersive X-ray spectroscopy (EDS) is a technique for analyzing the elemental composition of a sample with a spatial resolution, used in SEM and TEM. The technique is based on excitation of core shell electrons, which are bound to the atom with different strengths in different elements, and thus the electron relaxation results in very specific photon energies. EDS can be used to determine both the qualitative and quantitative composition of a sample. This chapter will cover the theoretical formation of characteristic X-rays, the empirical adjustments done due to creation and detection issues, explain quantitative calculations, and briefly explain the EDS system.

2.1 Theoretical view on characteristic X-rays

This section is primarily based on Hollas [3, Ch. 8.2] and Goldstein [1, Ch. 4.2]. It covers the theoretical physics behind creation of characteristic X-rays.

2.1.1 Formation of characteristic X-rays

The formation of characteristic X-rays is an inelastic quantum mechanical scattering process in two steps. In the following four equations the subscripts are referring to specific electrons in order to distinguish between them, which is also used in [Figure 2.1](#). The figure shows the formation of characteristic X-rays with the atom as in the Bohr model, where ionization and relaxation is divided into two steps. The three relaxations showed produce three characteristic X-rays with different energies. The naming of the X-rays are covered in [Section 2.1.2](#), and illustrated in [Figure 2.2](#). Only the relevant electrons are shown in the figure. In the ionization step, the electron from the beam ionize the atom which ejects an electron from the core orbital. In the relaxation step an electron from a higher orbital relaxes to the hole left by the ejected electron, and the figure shows the three possible relaxations. The first step is described in [Equation \(2.1\)](#), where electron e_1^- from the incident electron beam eject electron e_2^- from the core orbital of atom A [3, Eq. (8.12)].

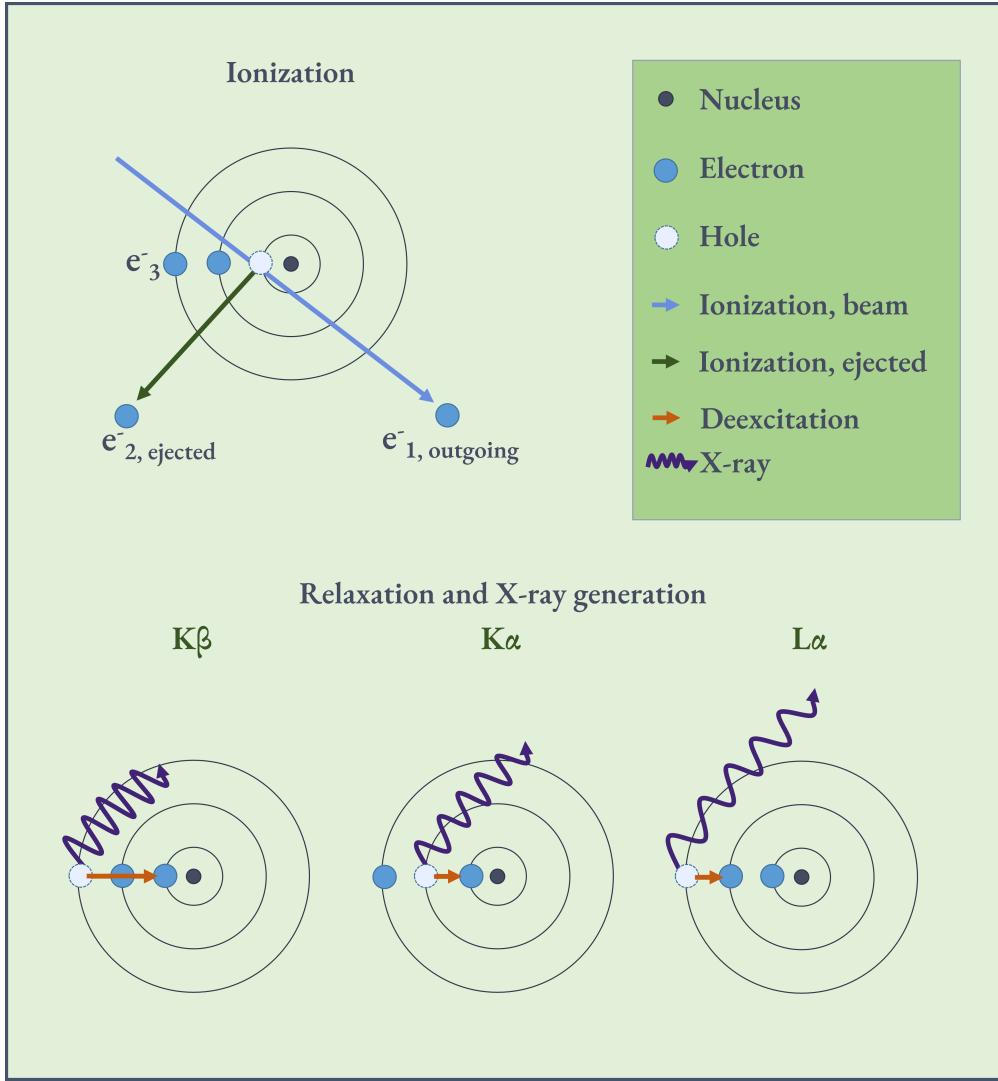


Figure 2.1: The formation of characteristic X-rays with the atom as in the Bohr model, where ionization and relaxation is divided into two steps. The three relaxations showed produce three characteristic X-rays with different energies. Increased energy is illustrated as X-rays with shorter wavelengths.

The incident electron from the beam loose energy to both breaking the binding energy of the core orbital and to the kinetic energy of the ejected electron. The energy is given by [Equation \(2.2\)](#), and illustrated as the difference in energy level between the orbitals in [Figure 2.2](#). The user can control the incident electron energy $E_{1, \text{incident}}$ by the acceleration voltage V_{acc} of the electron gun. The energy of the characteristic X-ray is dependent on the binding energy of the core orbital, $E_{2, \text{core shell, binding}}$. In EDS $E_{1, \text{outgoing}}$ and $E_{2, \text{kinetic}}$ serve no purpose [[Eq. \(4.1\)](#)].

$$E_{1, \text{incident}} = E_{1, \text{outgoing}} + E_{2, \text{core shell, binding}} + E_{2, \text{kinetic}} \quad (2.2)$$

In the second step electron e_3^- from a higher energy orbital relaxes to the hole in the core orbital of atom A, and the difference in energy is emitted as a photon with a specific energy $h\nu$ called the characteristic X-ray [3, Eq. (8.12)].



The energy of the characteristic X-ray is the difference in energy between the ionized orbital and the orbital filling the hole, shown in [Equation \(2.4\)](#). The equation specifies the energy of the X-ray as $h\nu$, but [Section 2.2](#) explains why users of EDS just use the energy directly, usually measured in eV or keV [1, Eq. (4.2b)].

$$h\nu_{\text{X-ray}} = E_{2, \text{core shell, binding}} - E_{3, \text{outer shell, binding}} \quad (2.4)$$

In the second step in [Equation \(2.3\)](#) it is also a probability that the relaxation energy is used to eject and give kinetic energy to another electron from a higher energy orbital. This process results in two ejected electrons, both the ionized electron from the core orbital and a second ejected electron from a higher energy orbital. The second ejected electron is called an Auger electron. Auger electrons are used for surface studies, because they can penetrate around 2 nm solid material and thus does not escape from inside the sample. The X-rays are emitted in all directions and penetrate typically 4000 nm, and are the signal in EDS. The ratio between the characteristic X-ray photons and Auger electrons are known as the fluorescent (quantum) yield, ω .

$$\omega = \frac{\text{X-ray photons}}{\text{Auger electrons}} \quad (2.5)$$

The fluorescent yield is heavily dependent on the experimental setup and can be approximated, which is covered as one of the empirical factors in [Section 2.2](#).

2.1.2 Naming convention

The transition lines are grouped and named semi-systematic, based on the orbital the vacancy is in, and the orbital the electron is relaxed from. The naming convention is semi-systematic because it is the original empirical system published in Nature by the Swedish physicist Siegbahn in 1916 [4], when they did not have the knowledge we have today. The International Union of Pure and Applied Chemistry made a more systematic naming convention for X-ray lines which is supposed to be the official one [1, Ch. 4.2.4]. However, the Siegbahn notation is used in the X-ray Data Booklet, in HyperSpy, and by the TEM group at NTNU, and thus is used in this thesis. [Figure 2.2](#) shows some orbitals and the

corresponding X-ray lines with the Siegbahn notation. These lines are the ones which are identified in EDS analysis in this project.

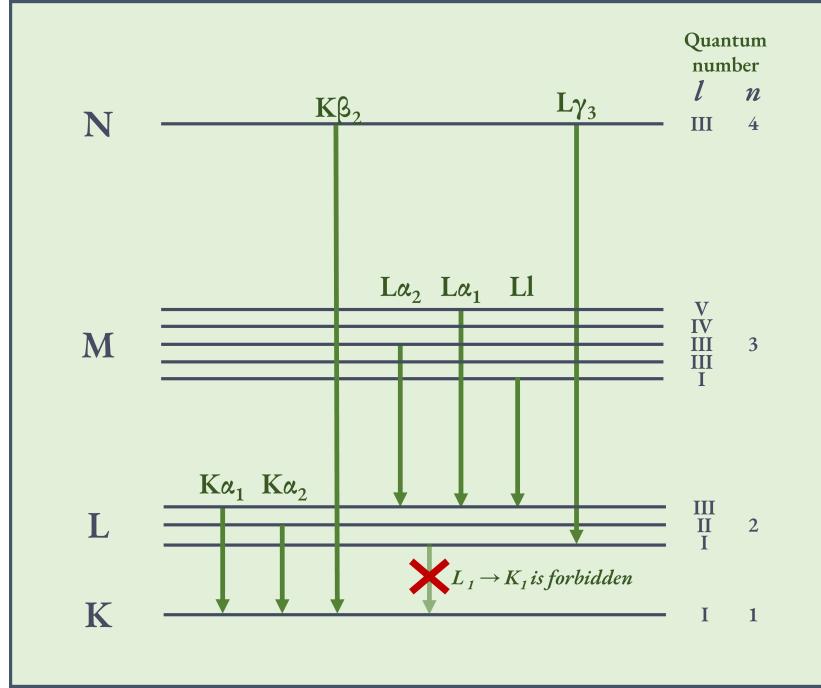


Figure 2.2: Schematic of the electron relaxations relevant for this thesis, with their naming convention. The lines are the orbitals in the different shells, and the arrows are the relaxations. Quantum number n and l are shown on the right side of the figure. The figure also show one forbidden line. The distance between the orbitals are increasing with the Z number. When the energy of two lines are to close that they are unresolvable, they are referred to as one line, e.g. $K\alpha_1$ and $K\alpha_2$ are together forming $K\alpha$. For Ga, $K\alpha = 9.25 \text{ keV}$, $K\beta = 10.26 \text{ keV}$ and $L\alpha = 1.01 \text{ keV}$.

The X-rays are first named by which shell in the Bohr model the vacancy is in, i.e. the principal quantum number n of the vacancy orbital. Relaxations to the innermost shell $n = 1$ is named K-transitions, relaxations to $n = 2$ is L-transitions, relaxations to $n = 3$ is M-transitions.

The X-rays are further grouped with Greek letters in families. Orbitals close in energy are usually in the same group, which means that electrons in the same shell usually are in the same family. This naming is non-systematic, but tends to follow a pattern where the transitions labeled α are the lower energy transitions corresponding to the $n + 1$ orbitals, and the transitions labeled β are the higher energy transitions corresponding to the $n + 2$ orbitals. For example, $L \rightarrow K$ are α -transitions.

In addition, the lines in an X-ray group are labeled with subscript numbers which generally start with the highest intensity. This is a splitting of the lines due to different energy levels of the orbitals in the same shell. The different energy levels are due to the spin-orbit coupling, which is the interaction between the electron spin and the orbital

angular momentum. The spin-orbit coupling increase with increased Z , which separates the lines more and more. The splitting of the α family to α_1 and α_2 are usually first resolvable in EDS for elements heavier than tin with $Z = 50$ [3, Ch. 8.2.2.3].

Putting these three naming conventions together, we name the transition $L_3 \rightarrow K_1$ as $K\alpha_1$, and $L_2 \rightarrow K_1$ as $K\alpha_2$, with more examples in Figure 2.2. The transition $L_1 \rightarrow K_1$ has $\Delta l = 0$ and is thus forbidden by the selection rules, see Equation (2.6). In gallium the $K\alpha_1 = 9251.74$ eV and $K\alpha_2 = 9224.82$ eV [5] have an energy difference of $\Delta E = 26.92$ eV, which is too low to be resolved in EDS, as shown in Figure 4.4.

2.1.3 Energy and intensity

The energy of the characteristic X-ray depends on which orbital the vacancy is in, which orbital the electron is relaxed from, and the amount of protons in the core of atom A. Higher Z means higher energy of the characteristic X-ray line, because the energy difference between the relaxation orbital and the ionized orbital is larger. Atoms with higher Z have more possible transitions, because they have more electrons and orbitals.

The selection rules, which govern the allowed transitions for the formation of characteristic X-rays, are based on the Pauli exclusion principle and the spin-orbit coupling. See Figure 2.3 for illustration of the quantum number n and l , which are relevant for the selection rules. Quantum number j is the total angular momentum, which is the sum of the orbital angular momentum l and the spin angular momentum s . The selection rules in Equation (2.6) is for the electron which relaxes to the vacancy in the core orbital of atom A [3, Sec. 8.2.2.2].

$$\Delta n \geq 1; \quad \Delta l = \pm 1; \quad \Delta j = 0, \pm 1 \quad (2.6)$$

Even though heavy atoms like gold have more than 30 possible transitions, only a few are detectable in EDS. Lines which are undetectable have low abundance, are too close to other lines, or are forbidden by the selection rules [1, Ch. 4.2.3]. Lines which are detectable have an intensity dependent on the amount of the element in the sample, because a higher amount gives more counts in the detector. The counts in EDS the number of X-ray photons detected in a specific energy range. The energy range is typically around 10 eV. In theory, the ratio of the atomic concentration between two elements are proportional to the ratio of the corresponding lines from the elements. However, there are many factors which affect the intensity of the lines, which are covered in Section 2.2. A theoretical spectrum with just lines are shown in Figure 2.4. The lines are not spread, and have different energy and intensity. Another illustration of a theoretical spectrum, but with a background radiation, is shown in Figure 2.5. The background radiation is generated exponentially, but at lower energies the background radiation is absorbed and thus get the illustrated shape. A real spectrum is shown in Figure 2.6. In this figure the lines are spread, and there is a background present which is low. The shape of a real background is shown in Figure 2.7. Both Figures 2.6 and 2.7 have counts on the y-axis.

Theoretical values of characteristic X-ray lines are available in the X-ray Data Booklet [5] and in HyperSpy [7]. These

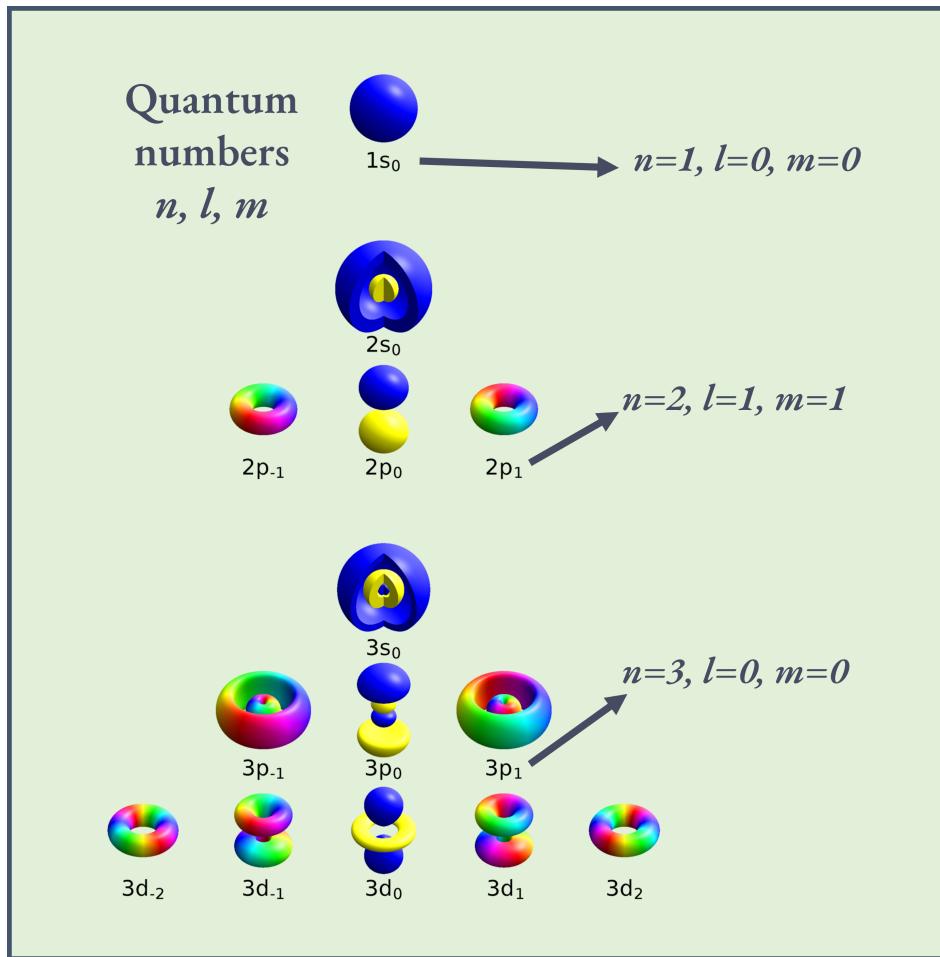


Figure 2.3: The quantum numbers n , l , and m in a hydrogen-like atom. The principal quantum number is shown as the block with values $n = 1, 2, 3$. The Azimuthal quantum number is the rows as $l = s, p, d$, where their numbers are 0, 1 and 2. The magnetic quantum number is the columns as $m = -2, -1, 0, 1, 2$. The spin quantum number, s , is not geometrically independent. The total angular momentum quantum number $j = l + s$, and since s is geometrically independent it is not possible to visualize j here. The figure is an altered version of a figure from the article "Quantum number" on Wikipedia, made by Geek3 - Own work, Created with hydrogen 1.1, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=67681892>.

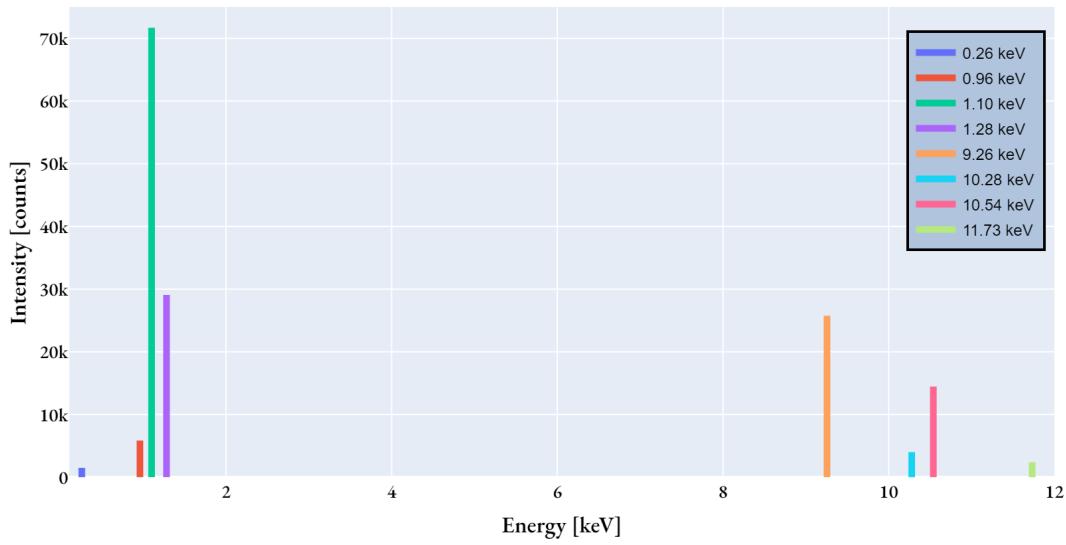


Figure 2.4: Theoretical lines in a spectrum. The lines have different energy and intensity.

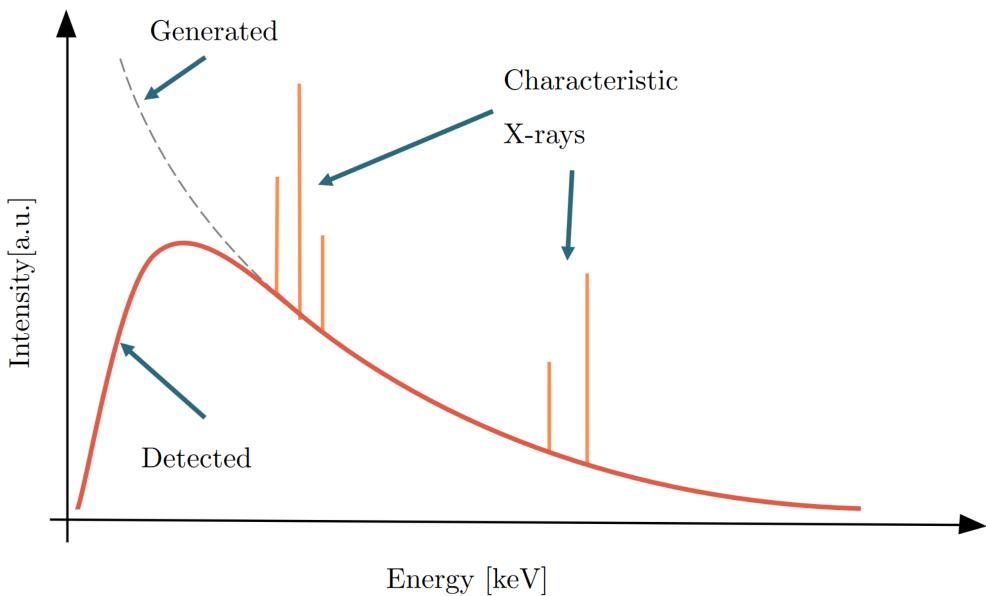


Figure 2.5: An illustration of a theoretical spectrum with characteristic X-rays and background radiation. The background radiation is being generated exponentially, but lower energies the X-rays are absorbed, and thus the detected background get the illustrated shape. The figure is borrowed from Skomedal [6, Fig. 2.8].

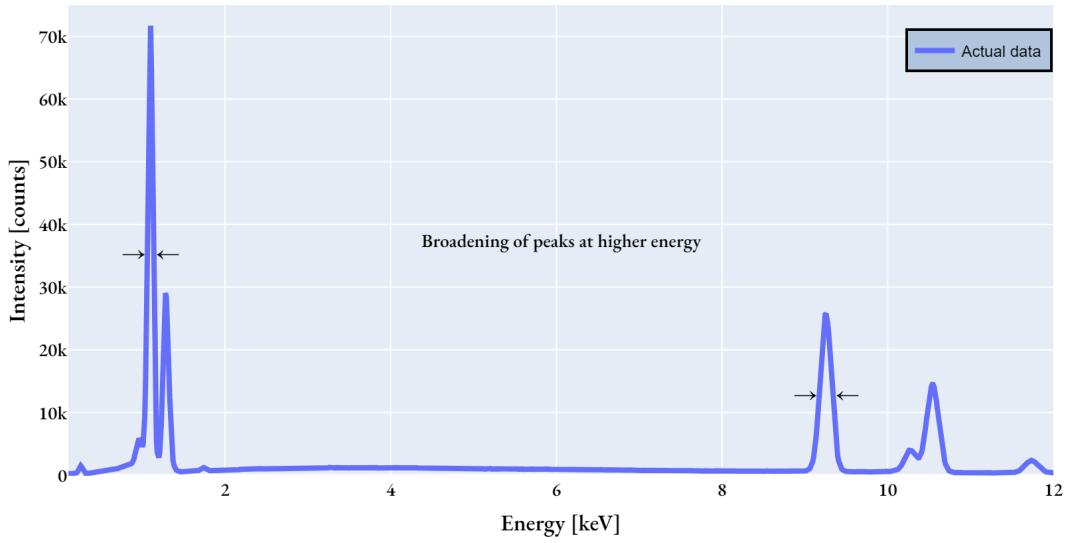


Figure 2.6: Actual lines, or peaks, in a spectrum. The lines have different energy and intensity, and they have a spread. The peaks have Gaussian shapes, as the formation of characteristic X-rays is a probability process. This is a real spectrum with a background, but it is low and do not show. A typical background is plotted in [Figure 2.7](#).

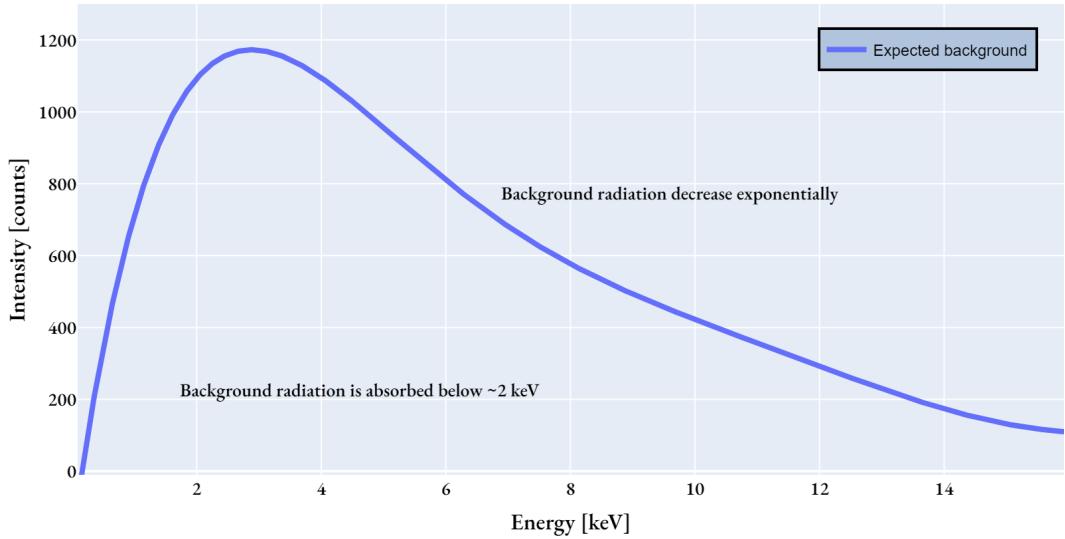


Figure 2.7: Expected background in a spectrum. This is a fit of a real background spectrum, but the exact numbers are not important. The background radiation is being generated exponentially, but at lower energies it is being absorbed. The figure is meant to illustrate the shape of the background.

include the energy and the weight of the lines. The lines for Ga and Mo are listed in Table 2.1 and Table 2.2. There are minor differences in the energies and weights from the XRB and HS. One bigger difference between the two tables is that HyperSpy have a line for Mo $L\gamma_3$, which is not in the X-ray Data Booklet. Another difference is that the X-ray Data Booklet splits the $K\alpha$ line into two lines, while HyperSpy have one line for the $K\alpha$ line. In addition, the XRB list Mo $L\alpha$ as two lines and Mo $K\beta$ as three lines. These additional lines are very close in energy and only one of the lines have a higher weight, so these line separations are excluded from the tables.

Table 2.1: Ga lines from the HyperSpy library [7] and X-ray Data Booklet (XRB) [5, Table 1.2 and 1.3]. HyperSpy operates with a single line for $K\alpha$, while the XRB lists two lines. The weights are relative to the strongest line in each shell for each element.

Line	XRB, energy [keV]	XRB, weight	HS, energy [keV]	HS, weight
$K\alpha$	9.25174	1.0	9.2517	1.0
$K\alpha_2$	9.22482	0.51		
$K\beta_1$	10.2642	0.66	10.2642	0.13
$L\alpha_1$	1.09792	1.11	1.098	1.0
$L\beta_1$	1.1248	0.66	1.1249	0.17
Ll	0.957	0.07	0.9573	0.05

Table 2.2: Mo lines from the HyperSpy library [7] and X-ray Data Booklet (XRB) [5, Table 1.2 and 1.3]. HyperSpy lists a single line for $K\alpha$, while the XRB lists two lines. HyperSpy include a line at 2.83 keV for Mo $L\gamma_3$, which is not listed in the XRB. HyperSpy also include a line at 2.47 keV for Mo $L\beta_3$, which is not listed in the XRB.

Line	XRB, energy [keV]	XRB, weight	HS energy [keV]	HS, weight
$L\beta_1$	2.3948	0.53	2.3948	0.33
$K\beta$	19.608	0.15	19.6072	0.15
$K\alpha$	17.4793	1.0	17.4793	1.0
$K\alpha_2$	17.3743	0.52		
$L\beta_2$	2.5183	0.05	2.5184	0.05
$L\alpha$	2.2932	1.0	2.2932	1.0
Ll	2.0157	0.05	2.0156	0.04
$L\beta_3$			2.4732	0.06
$L\gamma_3$			2.8307	0.01
$L\gamma_1$	2.6235	0.03	2.6233	0.01

2.2 Empirical view on characteristic X-rays

On top of the theoretical physics there are many experimental processes affecting both the creation and the detection of characteristic X-rays. The scientific community uses an empirically influenced approach in EDS analysis. This approach includes empirical equations to deal with atomic number effects, absorption in the sample, secondary scattering or fluorescence, the imperfect beam, scattering in the chamber, several cross sections and the imperfect detector. There are many processes that can and do happen which complicates the physics. The easy view is just to analyze the signal and do not worry too much about the physics behind it. This is the empirical way.

2.2.1 From lines to peaks

A striking difference between the theoretical and the empirical view is that the lines are not lines, but peaks with a varying width. This is illustrated in [Figure 2.4](#) and [Figure 2.6](#). The peaks in the EDS spectrums have a Gaussian shape, which is a result of the signal being a statistical distribution of the electronically detected X-rays. [Figure 2.6](#) shows that lower energy lines are narrower than higher energy lines. The electronics in the detector are broadening the lines, and each detector has its own broadening function [8].

As the peaks are broadened, every peak in the spectrum can be fitted with a Gaussian function. A Gaussian peak is a curve defined by the following equation:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (2.7)$$

In the equation μ is the center of the peak, σ is the standard deviation, and x is the energy. When doing peak fitting, the first term is treated as the amplitude and is a parameter which is fitted, i.e. $1/\sigma\sqrt{2\pi}$ is swapped with a parameter A . The three fitted parameters are μ (center of the peak), σ (standard deviation) and A (amplitude). x is the energy, i.e. the x-axis in the EDS spectrum.

The width of the peak is a measure of the broadening of the line, and is usually given as the FWHM in EDS analysis. The FWHM is connected to the standard deviation of the Gaussian distribution, which is given by [Equation \(2.7\)](#). The FWHM can be calculated from the standard deviation, σ , with:

$$\text{FWHM} = \sigma 2\sqrt{2 \ln(2)} \quad (2.8)$$

2.2.2 Intensity

The intensity, or weight, of a line is dependent on multiple empirical and physical factors. This subsection will briefly cover fluorescent yield, critical ionization energy, and empirical weights in HyperSpy. In practice, the weight of the lines are included as a part of the k-factors or k-ratios, which are presented in [Section 2.3.2](#). Only the strongest lines are

listed in the X-ray Data Booklet, where the theoretical values of the characteristic X-ray lines are listed. The list include: $K\alpha_1$, $K\alpha_2$, $K\beta_1$, $K\beta_2$, $L\alpha_1$, $L\alpha_2$, $L\beta_1$, $L\beta_2$, $L\gamma_1$, $M\alpha_1$ [5]. HyperSpy have a library of the X-ray lines of the elements too. The X-ray lines which were identified in this project are illustrated in Figure 2.2.

The fluorescent yield ω is non-linearly dependent on Z . Figure (4.3) in Goldstein [1] is copied in Figure 2.8, which shows that the fluorescent yield for the first 90 elements is mostly increasing with Z . For K- and L-shell fluorescent yield, ω is strictly increasing. For M-shell fluorescent yield, ω is strictly increasing till around $Z = 80$, and then it starts to decrease. The figure also show that for the same element $\omega_K > \omega_L > \omega_M$. When dealing with thin samples, the fluorescent yield can be approximated by an empirical formula based on tin. The formula is given in Equation (2.9), where $a = 10^6$ for K-shell. The two key takeaways from this formula is that ω is dependent on Z , but kinda similar for close elements like Ga $Z = 31$ and As $Z = 33$.

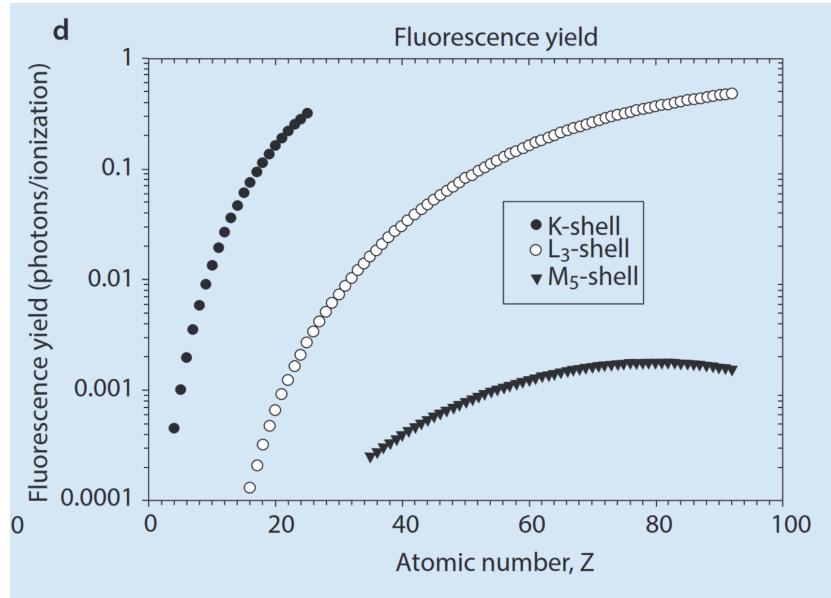


Figure 2.8: Fluorescent yield for K-, L- and M-shell, as a function of atomic number Z . The figure is copied from Goldstein [1, Fig. 4.3 (d)].

$$\omega = \frac{Z^4}{a + Z^4} \quad (2.9)$$

The critical ionization energy, E_C , is the energy the incident beam need to ionize a core electron in an atom. If the energy of the incident beam is lower than E_C , the core electron is not ionized and no peak can be detected. The critical ionization energy is dependent on the atomic number, Z , of the atom. Higher Z means higher E_C , because the core electrons are bound stronger to the nucleus. When the incident beam has an energy higher than E_C and continues to increase, the amount of ionization is not constant and not linear. The amount of ionization with varying energy above E_C is dependent on the ionization cross section and the overvoltage, which again are dependent on what shell

the ionization happens in. The solution to this is empirically estimations and using k-factors where all factors like this is either cancelled out or included in the k-factor correction.

HyperSpy have an integrated list of the characteristic X-rays, with both the energy and the weight of the lines. Goldstein uses different intensity weights for isolated atoms, thin foils and bulk samples [1, Ch. 4.2.6], and all are dependent on the atomic number and ionized shell. One could venture down the rabbit hole of finding the theoretical weights for different lines, but that will not be done in this thesis.

2.3 Data processing

This section on data processing covers what HyperSpy is, quantification with Cliff-Lorimer and calibration of a spectrum.

2.3.1 HyperSpy

HyperSpy is a Python package for processing and analyzing signals and multidimensional data. It is open-source and free to use, and is developed by a community of scientists and engineers. Updates and discussion about the package is done on GitHub at <https://github.com/hyperSpy/hyperSpy>. The documentation for the package is available at <https://hyperSpy.org/hyperSpy-doc/current/index.html>. The origin of HyperSpy is in the field of TEM, but it has been expanded to include other fields such as SEM.

Currently, HyperSpy have implemented functions for quantification of TEM EDS data, but not for SEM EDS data. The functions and methods for treating SEM EDS data is still limited, while the methods for TEM EDS data is more developed.

2.3.2 Cliff-Lorimer and the k-factors

One of the included quantification methods in HyperSpy is Cliff-Lorimer [9]. The method is based on the thin-foil criteria, where it is assumed that all absorption and fluorescence can be ignored. The CL method is based on the assumption that the intensity of the characteristic X-rays from two elements in the sample is proportional to the concentration of the elements. Each element has a correction factor for the intensity emitted from its lines, which is called the k-factor. The concentrations, intensities and k-factors are related by the following equation:

$$\frac{C_a}{C_b} = \frac{k_a I_a}{k_b I_b} = k_{ab} \frac{I_a}{I_b} \quad (2.10)$$

If the same method is to be applied for a sample where the thin-foil criteria is not valid, a correction matrix must be applied. The correction matrix in SEM EDS is correcting for the atomic number effects (Z), the absorption in the bulk sample (A) and the fluorescence (F). This is called the ZAF correction matrix [10] [1, Sec. 19.10].

2.3.3 Calibration of the spectrum

Calibration of a spectrum is finding the energy per channel of the spectrum and the zero-offset. The calibration process is a general process, and is not specific to SEM EDS. In EDS the calibration can be calculated with two known peaks. E_1 and E_2 is the energy of the two characteristic X-rays, and c_1 and c_2 is the channel number of the two characteristic X-rays.

$$\text{Dispersion} = \frac{E_2 - E_1}{C_2 - C_1} \quad (2.11)$$

With the dispersion, the zero-offset can be calculated with the following equation

$$\text{Zero-offset in channels} = C_1 - E_1 \cdot \text{Dispersion} \quad (2.12)$$

2.4 Detection system

This section covers briefly the EDS detector. It is expected that the reader knows the working principles of a SEM.

2.4.1 EDS detector

An EDS system consist of a detector inside an electron microscope and a computer controlling the system. The detector is a silicon drift detector (SDD) that is sensitive to X-rays. The incident electron beam hits the sample where X-rays are generated and emitted. When the X-rays hit the detector, the electrons in the silicon drift detector are excited and the electrons drift to the anode. This is registered as a signal, and the amount of electrons is proportional to the intensity of the X-rays. When a signal is being registered, the detector cannot receive any more signals, and this is called the dead time. The computer controls the detector and shuts the detector off and on during the acquisition of the spectrum. Even though it looks like a spectrum is acquired simultaneously for all the energies in the spectrum, the detector can only register one photon at a time. The registered photons are registered in channels, which are a small energy range.

The EDS system is described nicely in Skomedal [6], and more in detail in Goldstein [7]. A nice table with relevant parameters for the EDS system is found in "Getting Started with NIST* DTSA-II" by Ritchie [8].

Chapter 3

Method

The data needed in this project was SEM EDS spectra. The spectra were acquired on a SEM APREO at NTNU NanoLab. The spectra were from a sample containing different sections with different elements to get different spectra to work with. The relevant material for data extraction is available on the GitHub repository for this project. The data is also available on the GitHub repository for this project, under the folder `data/2022-09-06_EDS-Apreo`. This method chapter describes the material used, the microscope and detector, and then the data processing. The data processing starts with the analysis in AZtec and HyperSpy, and then presents the self-made post data treatment.

3.1 Materials and specimen

EDS data was collected on one specimen with sections containing different materials. A $300 \mu\text{m}$ 2" Si wafer was cut in half and mounted with Cu tape on an FIB stub. On the Si wafer a smaller piece of a GaAs wafer was mounted with Cu tape. A Mo disk was mounted to the Si wafer with Ag paint. In the middle of the Mo disk there was a hole where a TEM grid was mounted. The TEM grid had GaAs nanowires on it, and the grid was made of Mo with C film. Growth of the GaAs nanowires is described in [6, Ch. 3]. A schematic of the sample is shown in [Figure 3.1](#).

Six areas on the specimen was sampled, and their respective spectra are named: GaAs, Mo, (GaAs) NW, Si, Cu and Al. GaAs is a piece of a GaAs wafer, where the Ga and As are 1:1. Mo is a Mo disk. (GaAs) NW is a Cu TEM 300 mesh grid with holey C film, where GaAs nanowires were grown by Au droplets, sometimes referred to as NW and sometimes as GaAs NW. Si is the Si wafer. Cu is the Cu-tape. Al is the Al FIB stub.

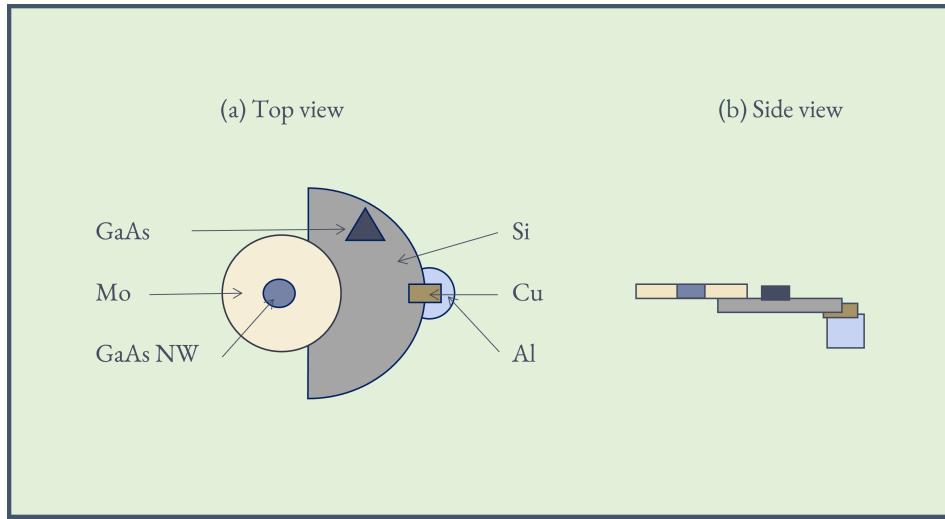


Figure 3.1: The sample used for the EDS data collection in the SEM Apreo. (a) is top view and (b) is side view. GaAs is a piece of a GaAs wafer. Mo is a Mo disk. GaAs NW is the TEM grid of Mo with C film and GaAs nanowires. Si is the Si wafer. Cu is the Cu tape. Al is the Al FIB stub.

3.2 The microscope and the detector

The data was collected with the SEM Apreo from FEI with an Oxford EDX detector at NTNU NanoLab. The detector is an EDX Oxford Xmax 80 mm² detector, with reported energy resolution of 127 eV [12]. The acceleration voltage, V_{acc} , can be set from 0.2 to 30 kV. In this project the V_{acc} was set to 5, 10, 15, and 30 kV. The beam current, I_{beam} , has a maximum value of 400 nA. In this project the I_{beam} was set to 0.2, 0.4, 0.8, and 1.6 nA. The SEM has an adjustable working distance. The SEM is equipped with a BSE and a SE detector.

The data collection was done on 5, 10, 15, and 30 kV. The beam current was 0.2, 0.4, 0.8, or 1.6 nA, depending on the dead time of the detector, trying to get the dead time around 30%. All samples were collected on 2048 channels ranging from 0 to 20 keV. Working distance was 10 mm. Processing time was set to 5 and each spectrum was collected with up to 2 minutes time live. SE and BSE imaging was used for locating areas of interest, and SE images were saved for each area. Table 4.1 in the result chapter have the settings used for the different spectra with some results.

3.3 Data processing

This data processing section is divided into two parts. First it is explained what was done in AZtec and HyperSpy. Secondly it is explained how the data was processed in Python.

3.3.1 AZtec and HyperSpy

The data from AZtec was extracted as described in Appendix A of Lundeby's master thesis [13]. Thanks to Mari Skomedal [6] and Martin Lundeby [13] who made manuals for extracting data from AZtec. Both qualitative and quantitative results were acquired. The quantitative results were acquired after selecting the elements in the sample and noted as atomic percent. The k-factors calculated theoretically by AZtec was noted down, as they are needed for the quantitative analysis in HyperSpy.

The data was analyzed in HyperSpy, both qualitative and quantitative. The qualitative analysis was done by loading the spectra and plotting with lines marking the theoretical peak center and empirical weight. The quantitative analysis was done on the GaAs bulk spectra with the Cliff-Lorimer analysis. All the quantitative analysis was done with the k-factors from AZtec. The intensity of a peak was calculated in two ways. The first way was to use the area under the peaks in the raw spectra, where the background was removed linearly. The second way was to make a fitted model of the background and the peaks as Gaussian curves, and use the area under the peaks in the model. The quantification was done with three type of calibrations. The different calibrations used was a model fit calibration in HyperSpy, a self-made calibration and the calibration from AZtec.

3.3.2 Self-made data processing

This subsection describes the data processing done by the author. In this project the data was analyzed with new code, which is available on the GitHub profile of the author, <https://github.com/brynjarmorka/>. The end goal of this code is to improve understanding of the analysis process, and show how some steps can be done in different ways, e.g. calibration, peak finding and background estimation. The code is written in Python [14] and uses Jupyter lab 3.4.5 [15]. HyperSpy 1.7.1 [7] was used for the analysis. NumPy 1.22. [16] is used for calculations, SciPy 1.9.0 [17] for fitting and peak finding, and Plotly 5.10.0 [18] for plotting. All packages are installed with Mamba on Python 3.8.13. The repository "eds-analysis"¹ contains the code developed throughout the semester, and the repository "eds-analysis-final"² contains a selection of the code. The code selected for the second repository is the code which is intended to enhance a users understanding of the analysis steps. This subsection describes the normalization, the Gaussian fitting and peak finding, the calibration and finding the area under the peaks.

Since the total amount of counts in a spectrum differs with V_{acc} , I_{beam} , DT, etc., the spectra had to be normalized to be able to compare them. Initially the spectra were normalized to the highest peak in each spectrum, $\text{Intensity}_{\text{relative to max}} = \text{Counts}_{\text{raw}} / \text{Counts}_{\text{max}}$. For the SciPy function `curve_fit` to work, the spectra had to be normalized and the raw channel numbers used as x-axis.

The fitting of a model to the spectra was done progressively more and more advanced. All the peaks was fitted as Gaussian curves, and the fitting itself was done with the SciPy function `curve_fit`. The first model fitting was just

¹<https://github.com/brynjarmorka/eds-analysis/>

²<https://github.com/brynjarmorka/eds-analysis-final>

making a Gaussian at two specified peaks. The second model was fitting a Gaussian at all the peaks in the spectrum, but still with a user input of the peak positions. The third model was fitting a Gaussian at all the peaks in the spectrum, but with the peak positions found by the SciPy function `find_peaks`. The problem with these models was that one of the peaks was usually moved to compensate for the background. Thus, the forth model was made, where the background was fitted as a sixth order polynomial. The background was fitted after removing the peaks, and then the peaks were fitted on top of the background. The fifth and final model was fitting the peaks and background in one go. The background was fitted as an n-order polynomial, and different orders were tested.

A third repository, `spectroscopy-channel-calibration`³, was made specifically for calibration of spectra, which was used in the course "TFY4255 - Materials Physics" at NTNU, October 2022. The model used in this calibration is the first fitting model, and thus is not as advanced as the model used in the final code. The calibration is done with a spectrum of known elements where the user inputs the energy of the peaks. The user further specifies the channel value of two peaks. The energy of the peaks are available through HyperSpy, or can be set manually from e.g. the X-ray Data Booklet. The code makes a Gaussian fit to the two peaks, to find the true peak center. A plot of the spectrum and the fit are shown, and the user can decide if the fit is good enough. The code then calculates the dispersion, and the zero-offset. In the end the code plots the spectrum with the calibration, and the user can decide if the calibration is good enough. The same calibration principle is implemented in "eds-analysis-final", but without the plots and the user interaction. The final calibration is using the fifth model with both background and peaks in one fit. However, since the calibration only needs the distance between two peaks, using the first model could be sufficient.

A function finding the area under the peaks was made as a simple first step in a quantitative analysis. This simple analysis was used to quantify if different calibrations gave different results. Implementing the Cliff-Lorimer method for quantification was done. The author started to look at implementation of a factorless quantification method, but did not have time to dive deep enough into that in this project.

³<https://github.com/brynjarmorka/spectroscopy-channel-calibration>

Chapter 4

Results

The results are presented in this chapter as plots and tables, and explained in the text. First qualitative then quantitative results are presented. All the spectra taken were qualitatively analyzed. Only the GaAs bulk spectra was quantitatively analyzed. The qualitative analysis starts with the calibration of the spectra, and then the spectra from the six sampled areas are presented. The quantitative analysis shows analysis done in AZtec, then quantify Ga and As with the CL ratio, and finally use the CL method in HyperSpy.

4.1 Qualitative analysis of the spectra

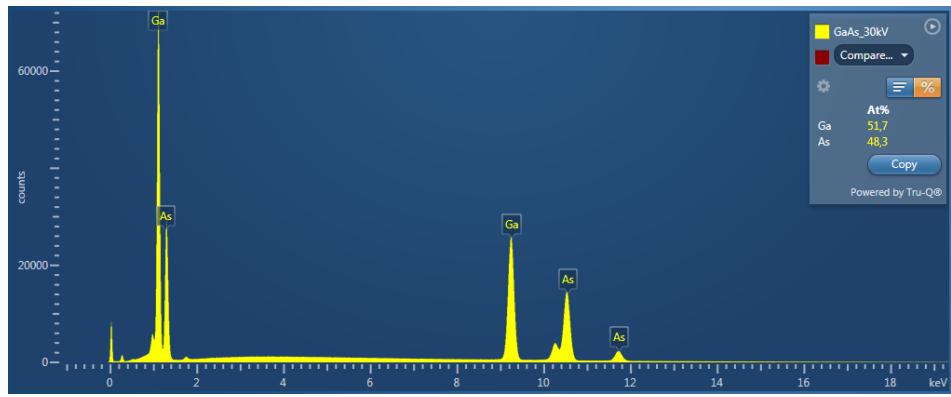
This qualitative section presents first the initial plots of the spectra done in AZtec and with HyperSpy, then the different calibrations and how well they match the theoretical values, and finally the spectra from the studied materials. The spectra are presented as plots and the calibrations are presented in tables, with their settings and counts in [Table 4.1](#).

4.1.1 Initial plot

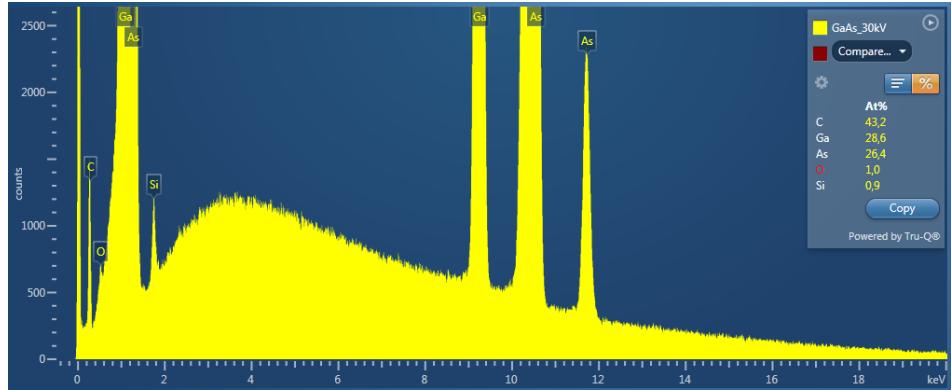
The initial plots are [Figure 4.1](#) and [Figure 4.2](#), showing the GaAs 30 kV spectrum. In [Section 4.1.3](#) all the spectra are presented. The initial two plots use the calibration from AZtec. The first figure is two screenshots from the AZtec software, and the second figure is a plot from HyperSpy. AZtec give the atomic percent of the elements in these plots. [Figure 4.1](#) from AZtec have three subplots. In (a) Ga and As are annotated. In (b) Ga, As, Si, O and C are annotated, and the y-axis is scaled to show the smaller peaks. (c) has a logarithmic scale on the y-axis, which is another way to show the smaller peaks. The HyperSpy plot in [Figure 4.2](#) utilize the Python package Matplotlib for plotting, and it has the theoretical line values annotated. The height of the line show an estimate of the weight of the peak. In the plot it is clear that the calibration is somewhat off, since the line positions does not match the center of the peaks. In the next section, the spectrum is calibrated using SciPy, and plotted with Plotly to more clearly visualize the spectra details.

Table 4.1: The detector settings and registered parameters for the different spectra. Processing time was set to 5 and each spectrum was collected with up to 2 minutes time live. The other detector settings are the same for all the measurements. The table includes the counts of the highest peak and the total counts.

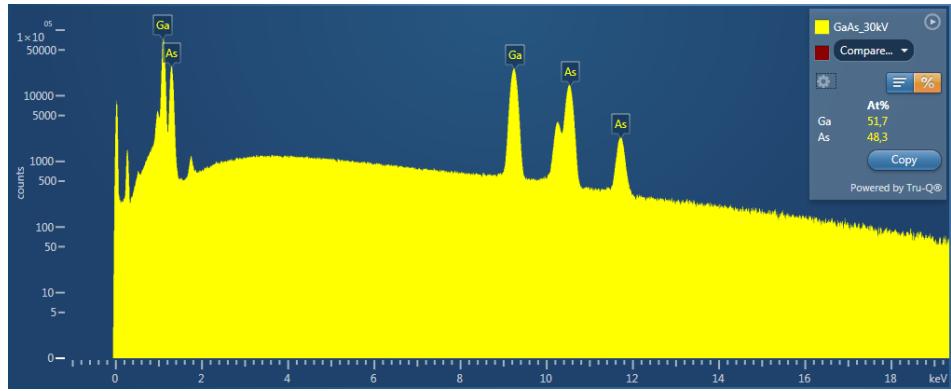
Sample	V _{acc} [kV]	I _{beam} [nA]	Dead time [%]	Live time [s]	Highest peak [k counts]	Total [k counts]
Al	30	0.1	22	120	197	1864
Cu	5	0.2	10	120	18	155
Cu	10	0.2	30	58	5	62
Cu	30	0.4	15	120	94	1060
GaAs	5	1.6	33	44	49	1172
GaAs	10	1.6	65	55	164	3715
GaAs	15	0.8	58	46	112	2653
GaAs	30	0.2	30	120	72	2726
Mo	5	1.6	30	120	72	2738
Mo	10	1.6	66	68	151	4779
Mo	15	0.4	42	67	80	2244
Mo	30	0.2	35	44	45	1211
NW	5	1.6	5	120	25	834
NW	10	1.6	18	120	50	1489
NW	15	1.6	18	120	46	1413
NW	30	1.6	12	120	25	963
Si	5	1.6	27	104	148	1962
Si	10	1.6	65	45	282	3160
Si	15	0.2	25	120	209	2188
Si	30	0.2	40	87	288	2912



(a)



(b)



(c)

Figure 4.1: The spectrum of the GaAs bulk at 30 kV, from AZtec. (a) whole spectrum with Ga and As annotated. (b) cropped to the weaker signals with Ga, As, Si, O and C annotated. (c) whole spectrum with a logarithmic scale on the y-axis, which can make weak signals clearer.

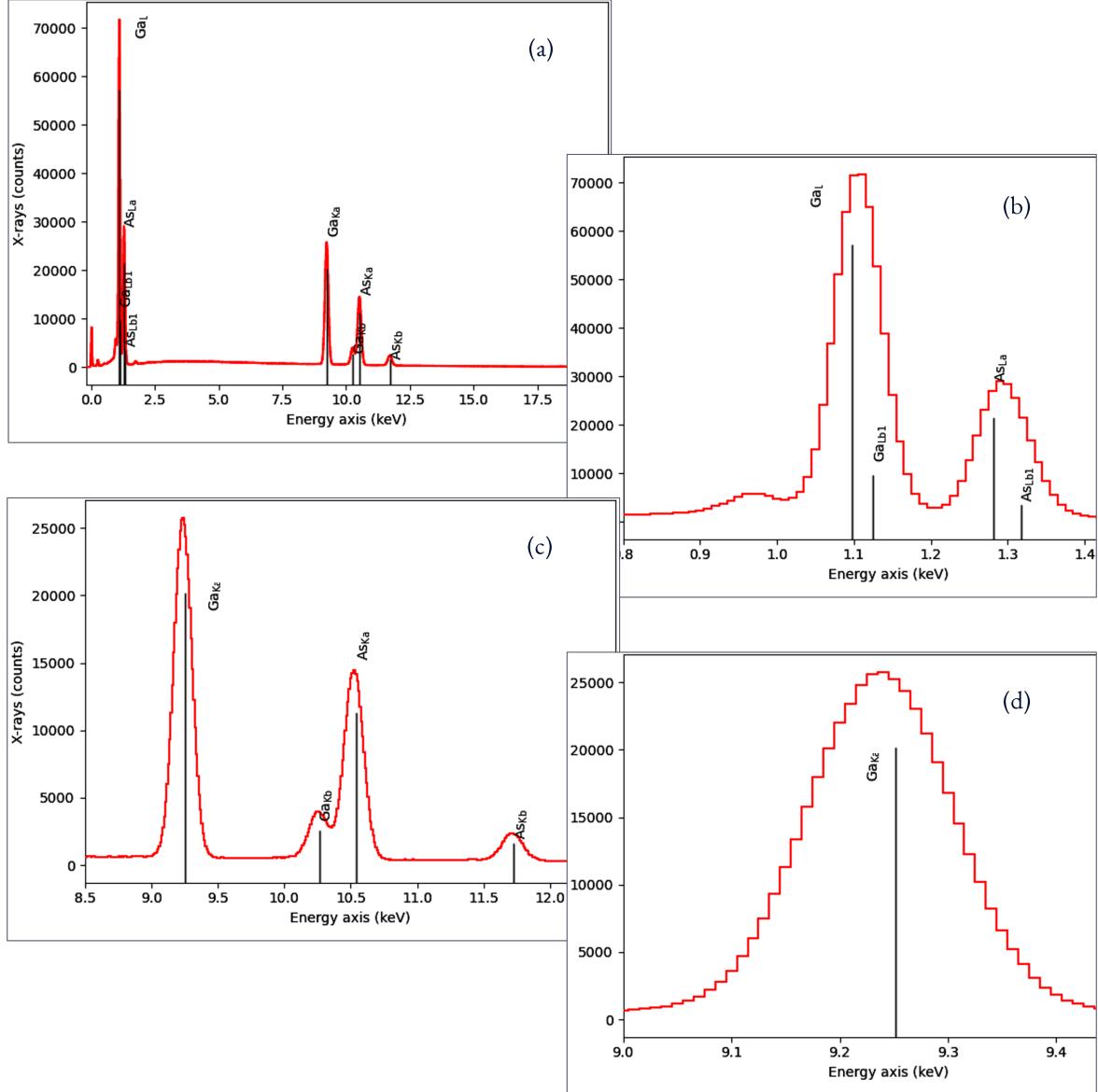


Figure 4.2: The GaAs bulk spectrum taken at 30 kV, plotted with HyperSpy. The vertical lines are the theoretical peak centers.(a) whole spectrum. (b) cropped to the L-peaks. (c) cropped to the K-peaks. (d) Ga $K\alpha$ peak. This plot has the calibration from AZtec, and it is clear that the line position is deviating from the center of the peaks. This plot was made with HyperSpy, which utilize Matplotlib. Fixing the labeling was not trivial, as the functionalities in HyperSpy which makes plotting accessible is hiding away the normal Matplotlib functionalities.

4.1.2 Calibration

Different calibrations were explored. The initial calibration is the one from AZtec, and is the one used in the spectra in [Figure 4.2](#). The AZtec calibration has a left shift for the L-peaks and a right shift for the K-peaks. The second type of calibration is the one given by the model fit in HyperSpy. The third type is from the self-made model fit, using the distance between two high intensity and far apart peaks to calibrate the energy scale. The third type use Ga $L\alpha$ and As $K\alpha$ peaks in the GaAs 30 kV spectrum, and use Mo $L\alpha$ and Mo $K\alpha$ peaks in the Mo 30 kV spectrum.

Values for the four calibrations are given in [Table 4.2](#), i.e. the dispersion and the offset. The deviations are a some eV, and the accuracy of the different calibrations give on specific peaks are given in [Table 4.3](#). Here accuracy is the difference between the theoretical peak position and the fitted peak center in the spectrum, given in eV. The percentage deviation is also given. At the bottom of the table the root-mean-square deviation (RMSD) is given. For almost all the peaks, the deviation is greatest for the AZtec calibration. One exception is the C $K\alpha$ peak, which deviates a less for the AZtec calibration. The difference between the HyperSpy calibration and the self-made calibration on the GaAs is small. In the qualitative section, the effect of the different calibrations on the spectra are explored.

Table 4.2: Different calibration values. The dispersion is calculated with [Equation \(2.11\)](#). The offset is calculated with [Equation \(2.12\)](#). The own calibration was done on Ga $L\alpha$ and As $K\alpha$ from the 30 kV measurement on the GaAs wafer. The HyperSpy calibration was done by making a model and fitting it to the data on the 30 kV GaAs spectrum.

Calibration method	Dispersion,	Zero offset
	[keV/channel]	[channels]
AZtec	0.010000	20.000
HyperSpy	0.010028	21.079
Calibration on Ga $L\alpha$ and As $K\alpha$	0.010030	21.127
Calibration on Mo $L\alpha$ and Mo $K\alpha$	0.010040	21.076

For the third type of calibration, the peaks were located and fitted to Gaussians, and the background was fitted as a polynomial. The fit of the 30 kV GaAs spectrum is shown in [Figure 4.3](#). The fit was done with a background polynomial of order 6 and 12, and both the fit and the background is plotted. Some peaks had too low prominence to be detected by `scipy.signal.find_peaks`, like the Si $K\alpha$ peak. The accuracy of the fit was quantified as the root-mean-square error (RMSE) between the fitted curve and the data, and the RMSE is given in the subplots.

The Ga $K\alpha$ peak is shown in [Figure 4.4](#), with the theoretical peak position from the X-ray Data Booklet [5] (XRB) and HyperSpy, and also where the peak would be with the HS, Mo and AZ calibration. The plot is calibrated with the GaAs spectrum, using Ga $L\alpha$ and As $K\alpha$ peaks. The theoretical lines and other calibration peak centers are marked with dashed lines. In accordance with the results in [Table 4.3](#), the AZtec calibration is most inaccurate. The HyperSpy library with X-ray lines only give one line for the Ga $K\alpha$ peak, while the XRB gives two lines, $K\alpha_1$ and $K\alpha_2$.

Table 4.3: Peak accuracy in the different calibration methods on 30 kV spectra: NW, Mo, Si, Al, Cu. The accuracy here is the deviation from the measured peak to its theoretical value. The percentage deviation is also given. The self-made calibration was done on two 30 kV spectra: GaAs and Mo. The HyperSpy calibration was done on the GaAs spectrum. At the bottom the RMS of the deviation in the column is given.

Peak	Theoretical [keV]	AZtec [eV]	HyperSpy [eV]	GaAs bulk 30 kV [eV]	Mo 30 kV [eV]
As L α	1.2819	12.8, 1.0%	5.6, 0.4%	5.4, 0.4%	7.2, 0.6%
As K α	10.5436	-21.3, -0.2%	-2.7, -0.0%	-1.1, -0.0%	9.8, 0.1%
Ga L α	1.098	11.5, 1.0%	3.8, 0.3%	3.5, 0.3%	5.1, 0.5%
Ga K α	9.2517	-14.2, -0.2%	0.9, 0.0%	2.2, 0.0%	11.9, 0.1%
Cu L α	0.9295	16.4, 1.8%	8.3, 0.9%	8.0, 0.9%	9.4, 1.0%
Cu K α	8.0478	-9.2, -0.1%	2.5, 0.0%	3.7, 0.0%	12.1, 0.2%
Mo K α	17.4793	-56.8, -0.3%	-18.8, -0.1%	-15.8, -0.1%	1.9, 0.0%
Mo L α	2.2932	24.0, 1.0%	19.7, 0.9%	19.7, 0.9%	22.5, 1.0%
Si K α	1.7397	2.9, 0.2%	-3.0, -0.2%	-3.2, -0.2%	-1.0, -0.1%
Al K α	1.4865	3.0, 0.2%	-3.7, -0.2%	-3.9, -0.3%	-1.9, -0.1%
Cu K α	8.0478	-9.3, -0.1%	2.4, 0.0%	3.5, 0.0%	11.9, 0.1%
C K α	0.2774	-8.2, -3.0%	-18.3, -6.6%	-18.7, -6.7%	-17.9, -6.5%
RMS [eV]		14.75	4.62	4.55	9.57

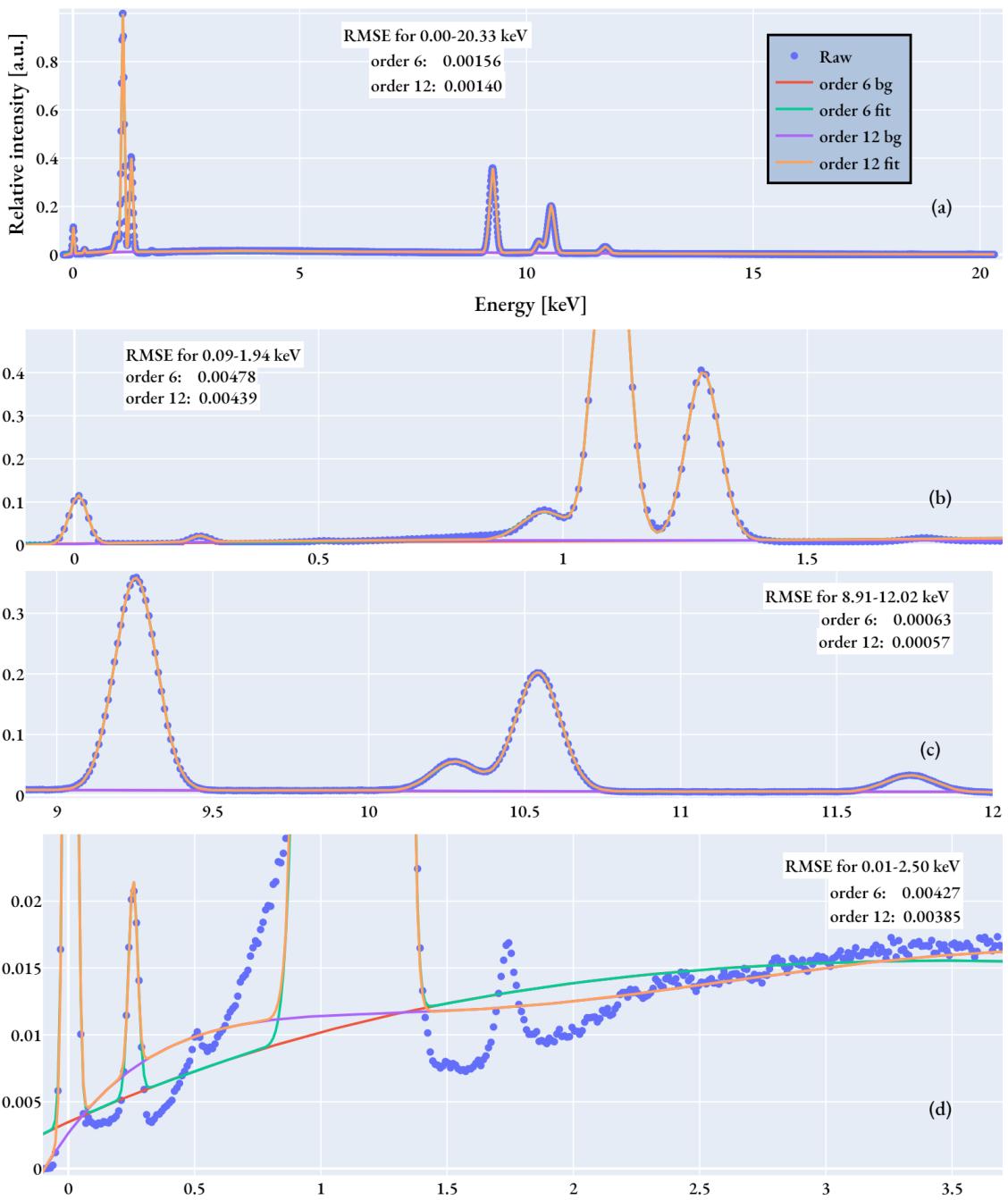


Figure 4.3: The fit of the 30 kV GaAs spectrum. The plots show the raw data (blue markers), the fit (green and orange) and the background (red and purple). The two different fits and backgrounds are with background polynomial of order 6 and 12, respectively. The RMSE for the area of each plot is given on the figures. (a) the whole spectrum. (b) the L-peaks. (c) the K-peaks. (d) the background at the L-peaks.

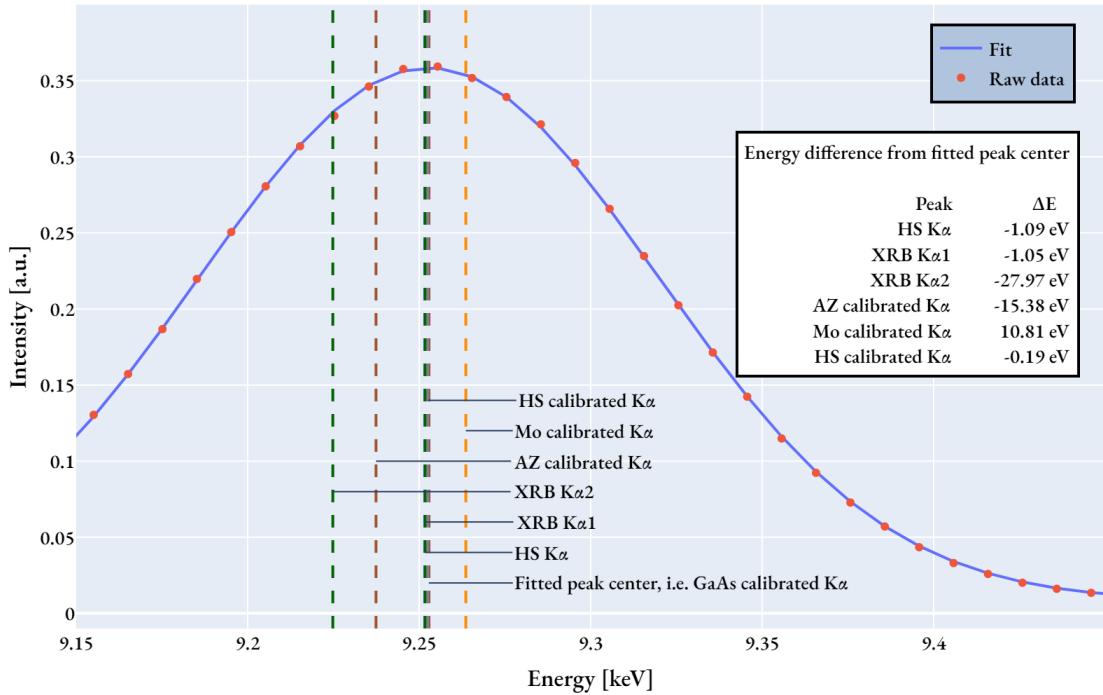


Figure 4.4: The Ga $K\alpha$ peak in the 30 kV GaAs spectrum, calibrated on Ga $L\alpha$ and As $K\alpha$ peaks. The raw data (red markers) and the fit (blue line) are plotted. The vertical dashed lines are the different values for Ga $K\alpha$. The HS, Mo and AZ calibrated $K\alpha$ lines are the shift the peak would have if those calibrations were used. The XRB $K\alpha 1$ and XRB $K\alpha 2$ are the theoretical peak positions from the X-ray Data Booklet. The HS $K\alpha$ is the peak position from HyperSpy. Annotated in the upper right corner are the deviations from the dashed lines to the fitted peak position. Given the energy resolution of 127 eV, the deviations are small.

4.1.3 Spectra from the studied materials

Figure 4.5 to Figure 4.11 shows the spectra for the six different areas of the sample plotted with Plotly. The parameters and settings for all spectra are given in Table 4.1, which does include, accelerating voltage, beam current, dead time, live time, maximum counts registered in one channel and the total counts. The plots are available as an interactive HTML plots on the GitHub repository. The calibration used in these spectra is one with the lowest RMSD, which is the one from the self-made model fit on GaAs. The y-axis is normalized to the highest peak value in each spectrum, i.e. the highest peak is always 1. The spectra are cropped to show the region of interest for the different materials. Peaks which are taller than the y-axis are marked with a gray dashed line. The annotated energy on the plots is the theoretical energy of the peak, and not the fitted peak center. Some black annotation lines are visually off center on the peaks, which is due to the peak deviation previously shown in Table 4.3. The terms peak and signal is used to differentiate between well-defined peaks and signals with low signal-to-noise ratio. The differentiation between peak and signal is not quantified with a threshold in this work. The annotations with an asterisk (*) are the peaks which have not theoretical value, e.g. a sum peak. First the spectra from the pure samples are presented, then the GaAs bulk wafer, and finally the GaAs NW. The order are: Al, Si, Cu, Mo, GaAs bulk, GaAs NW.

The Al spectrum from the FIB stub is shown in Figure 4.5. As all the other spectra, this spectrum have a zero peak, a C K α peak at 0.260 keV and a O K α peak at 0.51 keV. Most of the other spectra also have a Si K α peak at 1.74 keV. The tallest peak with a relative intensity of 1 is the combined peak of Al K α and K β at 1.49 keV. The contribution of Al K β to Al K α is 0.013, which is small, but still changes the peak shape. There are some signal at 6.40 keV, which is

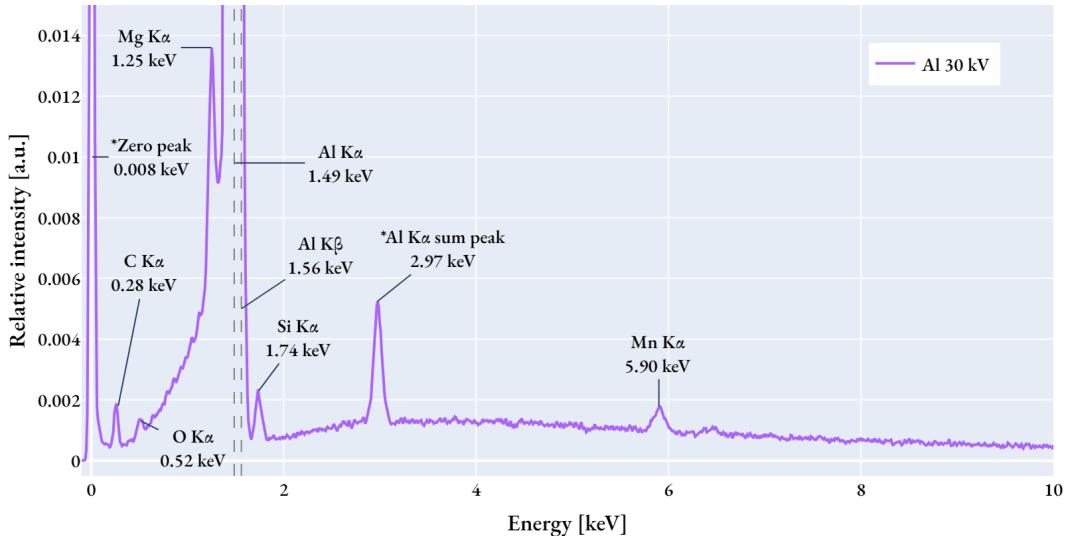


Figure 4.5: The 30 kV spectrum from the Al FIB stub. The peaks are annotated with the theoretical energy. The sum peak and zero peak are annotated with an asterisk, as they do not have a theoretical value. The dashed lines mark Al K α and K β , which together form one peak with relative intensity 1.

where Fe K α would be, but here the signal-to-noise level is low. The signal at 6.40 keV is around 20% higher than the background, where the background have counts from 150-180 and the signal has 210-220 counts. The peaks at 1.25 keV and 5.9 keV indicate that there was Mg and Mn in the sample, either as impurities or as part of the alloy. The last peak in the spectrum is the Al K α sum peak at 2.97 keV. The background is increase rapidly and almost linearly from C K α to Al K α , and drops to 10% after the Al K α peak. The figure also show that the Al K α absorbs part of the background to the right of the peak.

The Si spectra from the pure Si wafer sample area is shown in Figure 4.6. As in the other spectra, this spectrum have a zero peak, a C K α peak at 0.260 keV and a O K α peak at 0.51 keV. In addition, there is an unidentified peak at 0.080 keV. The tallest peak with a relative intensity of 1 is the Si K α peak at 1.73 keV, which have a contribution from the K β peak at 1.84 keV. The relative weight for Si K β to K α is 0.028. As in the Al spectrum, the Si spectra have a sum peak from the tallest peak. The sum peak is at 3.48 keV, which is the sum peak of the Si K α . In all four spectra the background drops significantly after the Si K α peak. The relative drop down is biggest in the 30 kV spectrum. The plot show that the background level is dependent on the acceleration voltage, at least when the spectrum is normalized to the tallest peak.

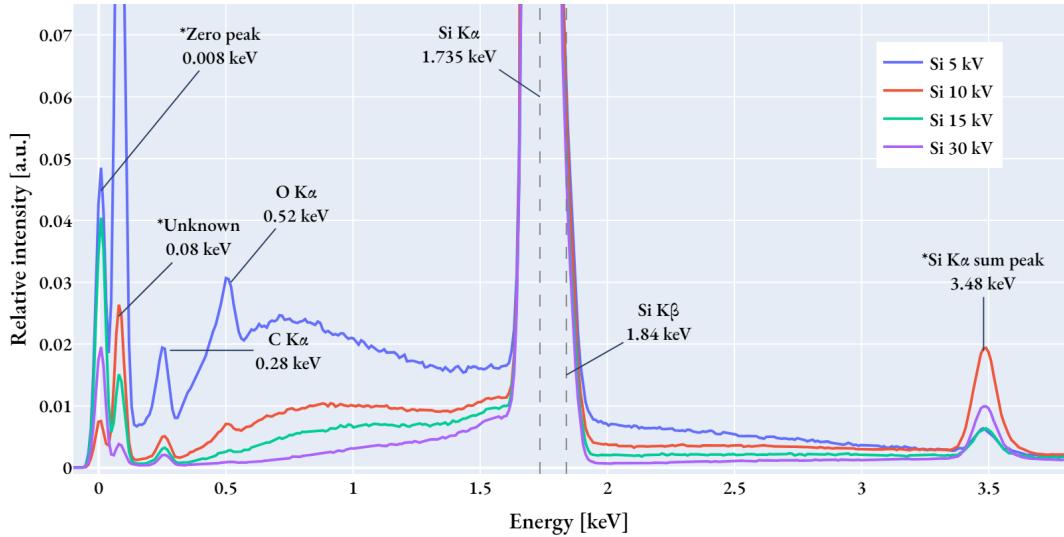


Figure 4.6: The spectra of the Si wafer, taken at 5 (blue), 10 (red), 15 (green), and 30 (purple) kV. The peaks are annotated with the theoretical energy. The sum, unknown and zero peak are annotated with an asterisk, as they do not have a theoretical value. The dashed lines mark Si K α and K β , which together form one peak with relative intensity 1.

The Cu spectra from the Cu-tape is shown in Figure 4.7. These spectra have a zero peak, a C K α , a O K α peak and a small Si K α peak. The tallest peak with a relative intensity of 1 is the C K α peak at 0.26 keV. The Cu K α and Cu K β peaks are only visible at the 30 kV spectrum. The height of the Cu K α peak is 0.017, which means that the C K α peak is more than 55 times taller. None of the spectra have a Cu L α peak, which should have been at 0.93 keV. The 30 kV spectrum have a signal at 2.29 keV, which could be from Mo L α , but no Mo K α signal is visible. In these spectra the background

is highest at the 30 kV spectrum, which is the opposite as for the Si spectra. The peak heights look dependent on the acceleration voltage, but the plot hides the fact that the DT is not equal for all spectra.

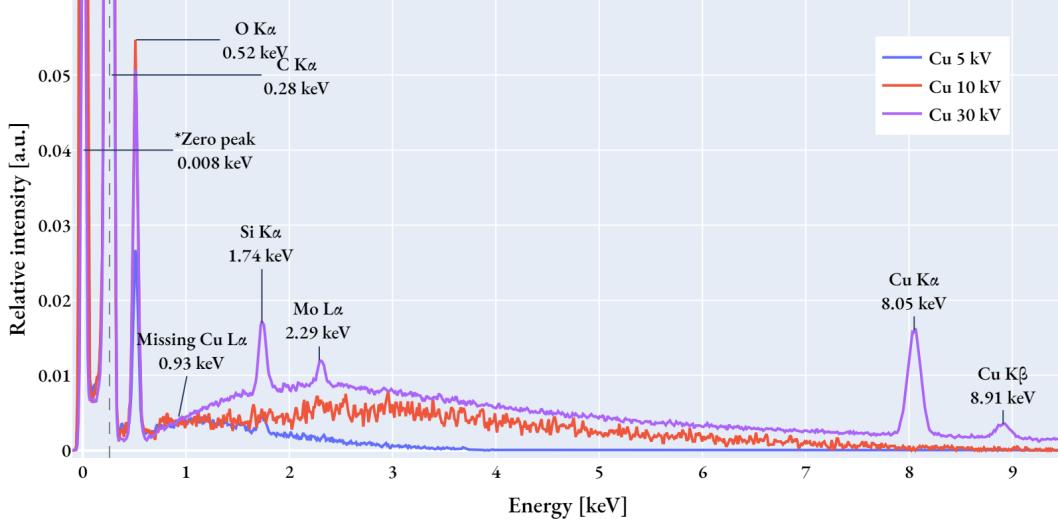


Figure 4.7: The spectra of the Cu-tape, taken at 5 (blue), 10 (red), and 30 (purple) kV. The peaks are annotated with the theoretical energy. The zero peak is annotated with an asterisk, as it does not have a theoretical value. The dashed line mark the C K α peak. The expected, but missing, Cu L α peak is marked.

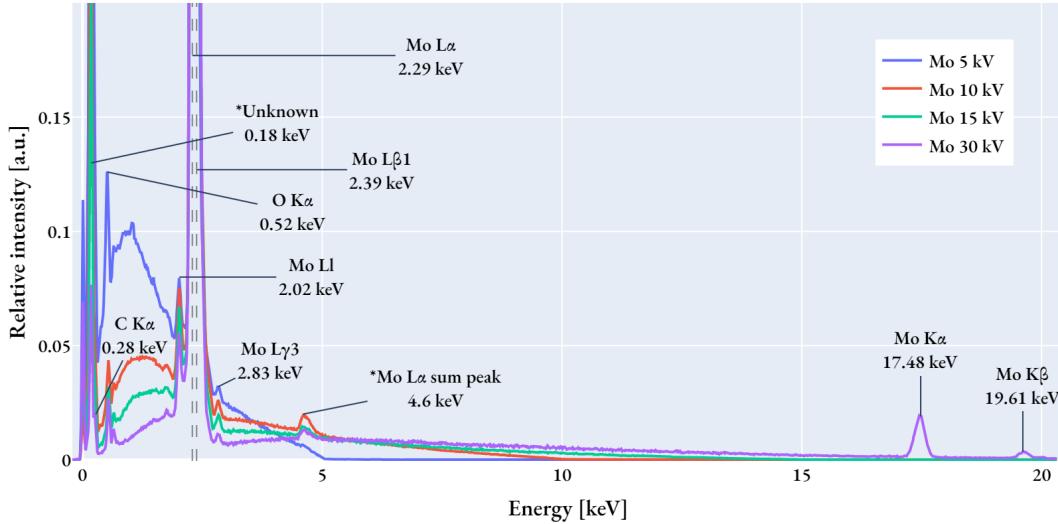


Figure 4.8: The spectra of the Mo disk, taken at 5 (blue), 10 (red), 15 (green), and 30 (purple) kV. The peaks are annotated with the theoretical energy. The sum, unknown and zero are annotated with an asterisk, as they do not have a theoretical value. The dashed lines mark Mo L α and Mo L β ₁. The background of the 5 kV spectrum is higher than the other spectra, which is due to the normalization to the tallest peak.

The Mo spectra from the Mo disk are shown in Figure 4.8. The four spectra have a zero peak, a C K α , a O K α peak. The Si K α give a very small signal, and is visible as the tiny peak before Mo Ll. The background of the 5 kV spectrum is very high, but that is due do its lower signal on the Mo L α , making the 0.18 keV peak the tallest and thus scaling up the background. In other words, the 5 kV spectrum is scaled to another peak, because the Mo L α peak is low. The tallest peak in the 5 kV spectrum is the 0.18 keV peak, which is also visible in the other three spectra. The energy of 0.18 keV only match with the B K α line at 0.1833 keV. The tallest peak in the 10, 15 and 30 kV spectra is the Mo L α peak at 2.29 keV, which is contributed by the Mo L β 1 peak at 2.39 keV. This one peak consists of two overlapping Gaussian, which are barely distinguishable. The HyperSpy database have the energy and weight of two other Mo peaks visible in the spectrum, which are the Mo Ll and Mo L γ 3 peaks at 2.01 keV and 2.83 keV. The X-ray Data Booklet only have the Mo Ll peak, but not the L γ 3 peak. This difference between the listed lines in the X-ray Data Booklet and the lines in the HyperSpy database can be confusing and is part of the discussion in Section 5.5. Only the 30 kV spectra have a signal at the Mo K α and Mo K β peaks at 17.46 and 19.62 keV. The peak at 4.6 keV is the sum peak of Mo L α and Mo L β 1. As in the other spectra, the background drops significantly after the tallest peak.

The GaAs spectra is shown in Figure 4.9 and Figure 4.10. The second figure is the same data with a lower y-axis and wider x-axis interval, to better visualize the background and sum peaks, and also show the K-line sum peaks. Compared to Figure 4.2, the peaks fit better with the theoretical values, which is quantified in Table 4.7. All the spectra have a zero peak, a C K α peak, and an O K α peak. The Ga Ll peak at 0.96 keV is visible in all four spectra. The Si K α peak is small

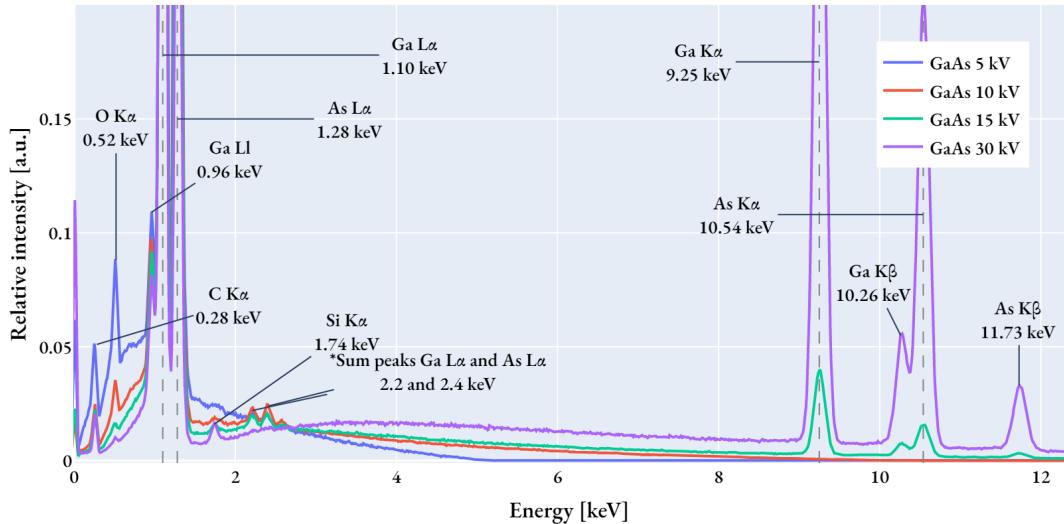


Figure 4.9: The spectra of the GaAs bulk, taken at 5 (blue), 10 (red), 15 (green), and 30 (purple) kV. This is a GaAs wafer, where the ratio of Ga to As is 1:1. The peaks are annotated with the theoretical energy. The sum peak, unknown peak and zero peak are annotated with an asterisk, as they do not have a (known) theoretical value. The dashed lines mark Ga L α , As L α , Ga K α , and As K α , with relative intensities in the 30 kV spectrum at 1, 0.41, 0.36, and 0.20, respectively.

in all the spectra, but strongest in the 30 kV spectrum and weakest in the 5 kV spectrum. The tallest peak in all four spectra is the Ga L α peak at 1.1 keV. The As L α peak has a decreasing relative intensity from 5 to 30 kV. For the first 2 keV the background is relatively highest in the 5 kV spectrum and decreases with increasing voltage. After 2 keV this trend reverses, and the background is highest in the 30 kV spectrum. The K peaks are visible in the 15 and 30 kV spectra. The sum peaks of the L peaks are visible in the 5, 10 and 15 kV spectra, but not in the 30 kV spectrum. The sum peaks of the K peaks are only visible in the 30 kV spectrum.

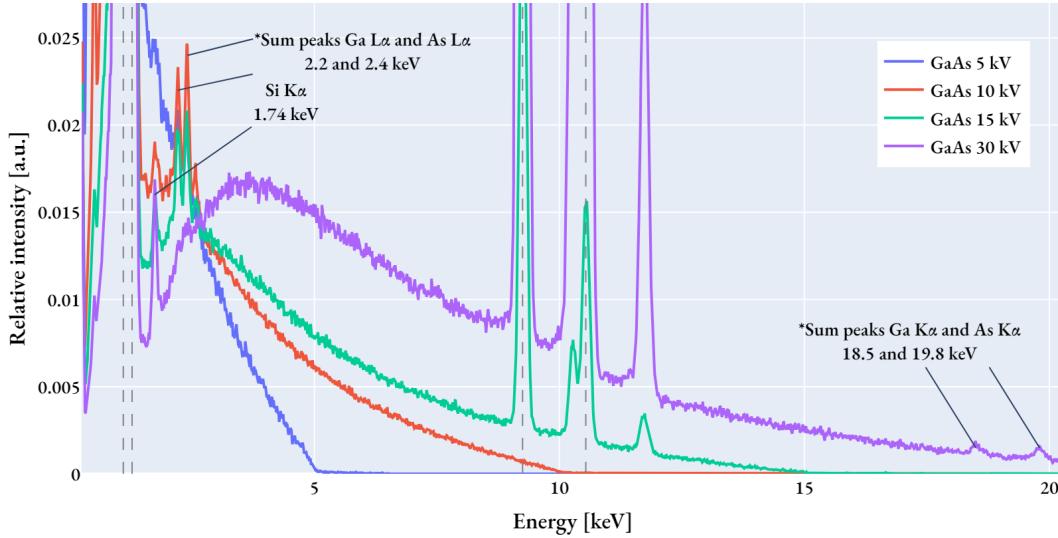


Figure 4.10: A cropped in view of the GaAs bulk spectra in Figure 4.9, but with the whole x-axis to show the K-line sum peaks. This plot is to better visualize the background and sum peaks.

The GaAs nanowire spectra are shown in Figure 4.11. This is the spectra with the most peaks, and contains signal from C, O, Ni, Cu, Ga, As, Si, Mo and Sb. One of the peaks is at 0.389 keV, which could be N K α peak, but it could also be other elements. The Ni signal is both from the L α at 0.85 keV and the K α at 7.49 keV. The Sb signal is from the L α at 3.6 keV, L β_1 at 3.8 keV, and Sb L β_1 at 4.1 keV. The Mo signal is both from the L α at 2.29 keV and the K α at 17.47 keV, but the K peak is very weak and thus not included in the plot. Mo is also giving a small signal at 2.02 keV, which is the Ll line. The Cu signal is both from the L α at 0.92 keV and the K α at 8.05 keV. The tallest peak in all four spectra is the Ga L α peak at 1.1 keV. In the 5 kV spectrum the C K α peak at 0.26 keV is equally high as the Ga L α peak. The As and Ga signals and their ratios are very similar to the GaAs spectrum signal, but the sum peaks are not visible in the NW spectra. The Ga and As L α sum peaks could have been visible, but the sum peak signal coincides with the Mo L α peak at 2.2 keV. The 10, 15 and 30 kV signal have Mo Ll peak at 2.02 keV, between Si K α and Mo L α .

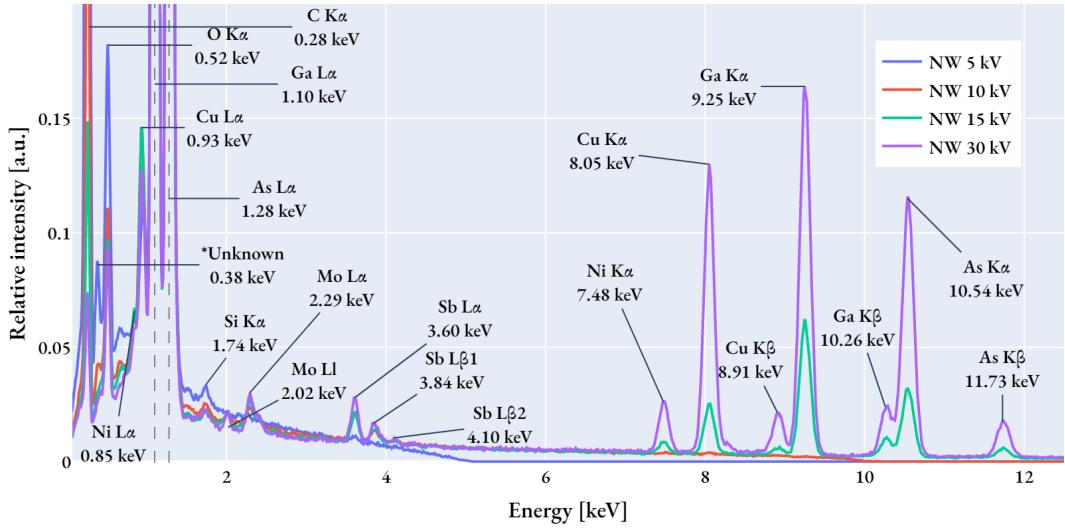


Figure 4.11: The spectra from the nanowire, taken at 5 (blue), 10 (red), 15 (green), and 30 (purple) kV. The peaks are annotated with the theoretical energy. The unknown peak is annotated with an asterisk, as it does not have a known theoretical value. The dashed lines mark Ga L α and As L α , with relative intensities in the 30 kV spectrum at 1 and 0.82.

4.2 Quantitative results

The initial quantification was done on the data from the GaAs wafer in AZtec. The results are presented in Table 4.4. AZtec can treat the data as both TEM and SEM data, and the results were quantified with both methods. The wafer is a 1:1 alloy of gallium and arsenic, so the atomic percent of Ga and As should be 50% and 50%.

Table 4.4: Initial quantification of the GaAs wafer done in AZtec, treating the data as both a SEM and a TEM signal. The ratio in the wafer is 1:1, so the correct ratio is 50% and 50%, because the results are in atomic percent.

V_{acc}	SEM		TEM	
	Ga	As	Ga	As
5 kV	46.26 %	53.74 %	54.22 %	45.78 %
10 kV	48.43 %	51.57 %	58.23 %	41.77 %
15 kV	49.32 %	50.68 %	62.13 %	37.87 %
30 kV	51.74 %	48.26 %	58.11 %	41.89 %

To do the quantitative analysis, HyperSpy needs k-factors from AZtec. The k-factors for Ga and As are given in Table 4.5. These k-factors are from the GaAs bulk wafer, and AZtec have estimated them theoretically.

To better understand the ratios between Ga and As, the areas under the peaks in the spectra were counted. Table 4.6 gives the ratios between the areas under the peaks for 5, 10, 15 and 30 kV. The table compares L α peaks, K α

Table 4.5: K-factors for Ga and As, extracted from AZtec. All the k-factors are theoretically estimated. AZtec provides either the k-factor for the L α or the K α line, and selects automatically based on the energy of the incident electrons.

V_{acc} [kV]	Line	Ga k-factor	As k-factor	$k_{\text{GaAs}} = k_{\text{Ga}}/k_{\text{As}}$
5	L α	1.21	1.086	0.898
10	L α	1.31	1.223	0.934
15	L α	1.331	1.259	0.946
30	K α	4.191	3.268	0.780

peaks, K β peaks and the sum of the peaks. The table also lists the FWHM of the peaks. The ratios are multiplied with the k-factors from AZtec to get the Cliff-Lorimer quantification results as a ratio.

One of the adjustments explored was the affect of the calibration on the quantification. Using different the calibrations in [Table 4.2](#) gave different quantification results when using CL in HyperSpy. The results are presented in [Table 4.7](#). The same method was used on all spectra, but the quantification on 10 and 15 kV are obviously wrong.

Table 4.6: Ratios of Ga and As on the GaAs wafer, with the Cliff-Lorimer quantification ratio. $K\beta$ at 15 kV was too low to be detected and is therefore not included in the table. The *ratio* column is *Ga sum* divided by *As sum*. The *CL* column is the CL quantification, done by multiplying the *ratio* with k_{GaAs} , from Table 4.5. The empty cells are for peaks where there was no k-factor or for the sum of the peaks, which have no center and no FWHM.

Peak	CL	Ratio	Ga center [keV]	As center [keV]	Ga FWHM [eV]	As FWHM [eV]	Ga sum [eV]	As sum [eV]
5 kV								
$L\alpha$	1.151	1.282	1.101	1.288	74.010	80.921	75.462	58.844
10 kV								
$L\alpha$	1.349	1.444	1.100	1.287	73.841	80.827	76.222	52.770
15 kV								
$L\alpha$	1.579	1.669	1.100	1.287	73.830	81.137	77.001	46.146
$K\alpha$		2.445	9.253	10.536	155.080	181.951	6.013	2.459
$L\alpha+K\alpha$		1.708				83.014	48.605	
30 kV								
$L\alpha$		2.279	1.098	1.287	72.309	80.849	76.465	33.546
$K\alpha$	1.301	1.678	9.253	10.542	157.799	168.238	58.718	34.994
$K\beta$		1.603	10.276	11.736	171.804	185.034	8.821	5.503
$L\alpha+K\alpha$		1.972				135.184	68.540	
$L\alpha+K\alpha+K\beta$		1.945				144.004	74.042	

Table 4.7: Quantification with different calibration methods. The quantification is done by in HyperSpy with Cliff-Lorimer method on the 30 kV GaAs spectrum. AZ is the AZtec calibration. HS is the HyperSpy calibration. GaAs is the calibration using Ga L α and As K α peak centers. The accuracy of the quantification is the deviation from 50%, because the sampled area is 1:1 GaAs wafer.

Acceleration voltage	Element	Line	AZ calibration	HS calibration	GaAs calibration
5	As	L	44.81 %	44.29 %	44.19 %
5	Ga	L	55.19 %	55.71 %	55.81 %
10	As	L	100.00 %	100.00 %	100.00 %
10	Ga	L	0.00 %	0.00 %	0.00 %
15	As	L	5.23 %	4.39 %	5.87 %
15	Ga	L	94.77 %	95.61 %	94.13 %
30	As	K	56.25 %	57.14 %	59.02 %
30	Ga	K	43.75 %	42.86 %	40.98 %

Chapter 5

Discussion

The discussion is presented in this chapter. Producing code from scratch is both a time-consuming process and a learning process. While developing the code, the author learned a lot about EDS analysis and ideas emerged for how EDS analysis could be improved. This chapter begins with an initial discussion of the qualitative results, and then it connects the sub-problems to the results and work done in the project. The order of the sub-problems is the same as in [Chapter 1](#), which is based on the work process.

5.1 General results from the spectra

This section starts with peak intensities, then the background, and finally some strays observed.

5.1.1 Peak intensities

All the spectra have peaks with high peak-to-background ratio, at least for the main peaks. Good sampling settings give a high signal-to-noise ratio, which is important for analysis. The high peaks in the spectrum are well-defined and fit the Gaussian curves well. These peaks are good enough for qualitative analysis, but there are some smaller signals which have been harder to interpret. A small signal can in some cases be interpreted in a qualitative analysis but is hard to use properly in a quantitative analysis. Interpreting every small signal as radiation from a specific element is not always correct. The small signal of Fe K α at 6.4 keV in the FIB stub spectrum is an example of this, where there is a 20% increase of counts compared to the background. It might be that this signal is from Fe, but it might also just be noise or an artifact. [Table 2.1](#) list the Ga-lines and their weights, and it is clear that some lines have very low weights. In this project it was decided not to use a hard threshold for the signal-to-noise ratio, which can be helpful in some cases. Setting a hard threshold should be based on both literature and empirical data for the specific detector setup. An issue with setting a hard threshold is that it can be too strict, which can lead to missing peaks of elements which are low in

concentration. If the sampling settings were not good enough, some peaks can be missing in the spectra. One example of this is in the NW spectrum: the low As K β peak in the 15 kV spectrum has an overvoltage of 3 keV and is visible, but the Cu K α peak in the 10 kV spectrum with overvoltage of 2 keV is not visible. Multiple data sets on the same area can be used to verify the sampling settings, since missing peaks in one data set can be visible in another data set. In general, the signals in the spectra are best in the 30 kV spectra. In conclusion, it is assumed that the sampling settings were good, at least when it is considered that there are multiple spectra from the same area.

Different elements do give different peak intensities, as expected. The intensity is dependent on the atomic number, which affects both the fluorescent yield and ionization cross section. In the GaAs spectrum, with Ga ($Z = 31$) and As ($Z = 33$), the Ga peaks have both higher max counts and higher sum counts, given in [Table 4.1](#) and [Table 4.6](#). From the theory of fluorescent yield and [Equation \(2.9\)](#), one would expect the As peaks to be more intense than the Ga peaks, but this is not the case. For the K-peaks this could be an effect of the overvoltage, but that would not explain the difference in the L-peaks. Another factor affected by the atomic number is the absorption, which is higher for heavier elements. In the k-factor approach with TEM thin samples the absorption is assumed to be negligible, which is not the case for bulk samples. Thus, some correction for the absorption should be done for bulk samples, which would be Z dependent. In addition, X-rays with lower energies are more absorbed than X-rays with higher energies. The ZAF matrix correction for SEM EDS data would correct for the absorption, but that have not been done in this project. Looking into the ZAF matrix correction would be a good next step for this project. In general, the generation of characteristic X-rays is a complex process, which is why much of the work done in EDS analysis is empirical.

The heights, or intensities, of the peaks are varying a lot. The plots of the spectra in [Figure 4.5](#) to [Figure 4.11](#) are normalized, and the total counts and maximum counts are included in [Table 4.1](#). No direct pattern between the amount of counts and the settings are apparent. Trends observed are that higher counts are obtained by increasing the current or the acceleration voltage, but this also increases the dead time. To figure out a pattern, fewer variables should be changed at the same time. This was done for the nanowire sample, where the I_{beam} and live time was kept constant. Still, a direct pattern is not apparent. As stated in Goldstein, "high counts and stable peak structures are critical for successful peak intensity measurements", especially to identify minor and trace constituents [[1](#), page 318]. The relation between the counts and settings could be further investigated to figure out which settings yield the best spectra in terms of good count statistics.

The main peaks have good count statistics, and the peak with the highest count is below 5 keV in all the spectra. For the spectra with strong K and L peaks from the same element, i.e. the lines of Ga, As, and Mo, the L-peaks are more intense than the K-peaks. From [Figure 2.8](#) it would be expected that the K-peaks would be higher than the L-peaks, but this is not the case. This is an example of the difference between the theoretical and empirical view, where the empirical is influenced by multiple factors like absorption, overvoltage and secondary X-ray generation. In the Mo 30 kV spectrum, the K α peak is only 2% the height of the L α peak. This low intensity of the K α peak is partially due to lower overvoltage, since the peak is above U/2. For the Ga and As the relative intensity of the 30 kV spectra in the bulk

and nanowire is not similar. The K-lines in the bulk sample is around 2 times as intense as the K-lines in the nanowire sample. Both spectra are normalized to the L-line of Ga. It was expected that the nanowire would give lower relative intensity for K-lines, because it is a thin sample where less of the lower level X-rays are absorbed, and thus getting lower relative intensity of higher energy peaks. Using [Table 4.6](#) with the ratios of the K-lines to the L-lines, it is clear that even the ratios of Ga and As lines change in the spectra with different V_{acc} . When applying the theoretically estimated k-factor for the respective spectrum, the ratios get closer to 1, but there is still a spread. This implies that the settings for acquiring spectra is important for the intensity of the peaks, which again is important for the quantitative analysis. Investigation of which settings yield better spectra should improve the accuracy of the quantitative analysis, and can be done in future projects.

Even though most of the spectra appear to have good intensities and good signal-to-noise ratio, there are some strange results. From the results it is clear that the Cu-tape is giving a bad Cu signal and should not be used as a Cu reference. The strongest signal from the Cu-tape is from C, which is probably from the sticky tape material. Regardless, the strangest Cu-tape result is the completely missing signal from Cu $L\alpha$ in the Cu-tape 30 kV spectrum. This spectrum has a clear signal from Cu $K\alpha$ and Cu $K\beta$, but no signal from Cu $L\alpha$ at all. In general the L-peaks are stronger than the K-peaks. In the 30 kV NW spectrum, Mo is giving a low but clear signal at the $L\alpha$ line, and even a small signal at the $L\beta$ and $L\gamma$ lines, but no signal from the Mo $K\alpha$ line. This is not strange as even the K-line signal from pure Mo is quite low, which is probably due to low overvoltage of the high K-peak. The missing Cu $L\alpha$ signal is even stranger when comparing it to the NW spectra, because the Cu signal in the NW spectra all have a clear signal from Cu $L\alpha$ which is around 30% stronger than the Cu $K\alpha$ signal. When the spectra on the Cu-tape sample area was taken, beam made visible damages on the surface, which can be part of the explanation of the missing signal. In the end, the spectra from the Cu-tape are regarded as unreliable. Finding both K-lines and L-lines is an important aspect of qualitative EDS analysis, because a present K-line without an L-line indicate that the identified element is not the correct one. The result with the missing L-line show that, unfortunately, it is possible to acquire spectra where a K-line is present, but the L-line is missing. Further, investigating this could reveal if this is an issue with the Cu-tape, the detector, the settings, or an artifact that can appear in other spectra.

From the spectra shown and [Table 4.6](#) including the FWHM, it is clear that the peaks broaden with higher energy. Increase of the FWHM with energy is expected, as explained in subsection 16.1.1 in Goldstein [1]. From the FWHM the resolution of the detector can be calculated with a conversion factor, since the resolution of EDS is defined as the FWHM of Mn $K\alpha$. This was done by Skomedal, and was considered to be a part of this project too. However, other parts of the project was prioritized.

One of the prioritized parts was to do calibrations, and it was quantified that the AZtec calibration was off, which might be connected to the center of the zero peak. With the most accurate calibration, the zero peak did not have its center at 0 keV. In the spectra the zero peak start before 0 keV and has its maximum at 0.008 keV. Since AZtec is a black box, it is not known how the calibration is done. If AZtec is using the zero peak to calibrate, this could be the reason for less

accurate calibration. Another possibility is that the calibration was done on installation, and that the detector has not been calibrated since then.

A challenge which is unaffected by poor calibration is overlapping peaks. Section 20.3 in Goldstein explains how to deal with overlapping peaks [1]. Dealing with overlapping peaks in model fitting is just modelling two Gaussian curves on top of each other. In the self produced fitting, the Gaussians have three parameters: center, standard deviation and height. Model fitting in HyperSpy is done with fixed centers, which means that the peak centers cannot change. Calibration can affect the result if fitting is done with a model where the peak centers cannot change. If the center of the smaller peak is off, larger peak will dominate the fitting and the smaller peak will get a lower intensity than it should. This problem is not present in the self produced fitting, since the peak centers are free parameters. An example of slightly overlapping peaks is the As K α and Ga K β peaks in the GaAs bulk and NW spectra. An example of severely overlapping peaks is the Mo L α and Mo L β_1 peaks in the Mo spectra. The Mo L α peak has one of the highest error in the peak accuracy in Table 4.3, where the calibrations are compared. The reason for this is that the overlapping Mo L-peaks are identified as just one peak by the automatic peak finder. Thus, the severely overlapping peak is modelled as one single peak which has its center shifted towards the middle of the two peaks. That is why all the calibrations miss the Mo L α peak with 20 eV. This issue was fixed manually when making the Mo L α and Mo K α calibration, and can with more time be fixed in the automatic peak finder.

5.1.2 The background

Another problem in the self produced code that can improve with more time is fitting the background. An idea that emerged towards the end of the project was to model the background as a spline. A spline is a piece wise polynomial function connected in knots. The fit of the background was observed to be bad at lower energies, where the background fit is more crucial for finding the right intensity of a peak. It is more crucial, because the background change more with energy at lower energies. The first iteration of the spline background would be to have the knot at the highest peak, because of the observation that the background decrease to around 50-10% after the highest peak.

All the spectra show the same behavior that the background is higher before the highest peak than after it. In the 30 kV GaAs spectrum the background is at 2200 counts before the overlapping L-line peaks, and falls to 500 counts after the peaks. In the 30 kV Si spectrum the background falls from 2000 to 200 counts after the main peak. The background shape in Si 30 kV is almost linear from 0.6 to 1.6 keV ending at 2000 counts, then it drops to 200 after the peak at 1.6-1.9 keV. After the drop the peak follow the expected shape, illustrated in Figure 2.5 and Figure 2.7. In the 5 and 10 kV Si spectra the background falls from 2500 to 1000 counts after the main peak. The Mo 10 kV spectrum, which is the spectrum with overall most background counts, has a background signal which falls from 5400 to 2900 counts after the main peak. In the nanowire spectra, which in general have the lower background counts, the background falls around 50% after the high L-peaks. The reason for the higher background to the left of the highest peak is that the X-rays formed with energy above the highest peak can be absorbed and re-emitted with the energy of the line in the

highest peak. When the code for the background was written, the effect of the highest peak was not considered, which resulted in visible bad background fits at the lower energies. This is shown in Figure 4.3, where the fit in general is good but misses on the background between 0 and 2.5 keV. The main peak effect on the background is changing both the value and the shape of the background.

The shape of the background is varying between the spectra, and is affected by the peaks and the acceleration voltage. All the 5 kV spectra decrease more or less linearly from 1 to 5 keV. This is due to the overvoltage, which lowers the possible background radiation to zero after 5 keV in the 5 kV spectra. When the acceleration voltage is just 5 kV, the critical ionization energy is not higher than 5 keV. In the GaAs, Si and Mo the relative height of the background decrease with higher acceleration voltage. The background in the Cu spectra increase with higher acceleration voltage. The most similar background signal over different voltages are in the NW spectra. This similarity is due to the fact that thin samples produce less background radiation than bulk samples. In general, backgrounds in the acquired spectra are low, but with different shapes.

5.1.3 The strays and artifacts

In addition to the characteristic peaks and the background, there are also artifacts and strays in the spectra. The background is an artifact which already have been discussed. One of the artifacts that will be discussed below is X-rays generated outside the area of the beam. Understanding the details in a spectrum is key for the qualitative analysis, which is the foundation for the quantitative analysis. Identifying peaks as strays or artifacts can be used to exclude certain elements, as well as making the analyst more certain of what elements are present. Lower peaks are central for the analysis. One example of this is the Al spectra, where two lower peaks match with the energies of Mg and Mn. Asserting what is a stray or artifact and what is a characteristic peak can be challenging, as they can be very similar to the characteristic peaks. An example of this is in the NW spectrum, where the two Mo L-peaks are exactly where the potential coincidence peaks of the L-lines of Ga and As would be. This makes the qualitative analysis harder, because the peak can be mistaken, and the quantitative analysis more uncertain, because the signal in the Mo L-peaks get higher if the coincidence artifact peaks are present. Generally, labeling peaks as strays or artifacts takes time, but it is a crucial part of the analysis.

All the spectra share some common strays and artifacts. All the strays and artifacts explained are described in detail in Goldstein [1]. The bremsstrahlung background radiation is present in all spectra, but with different shapes and intensities. The internal fluorescence peak present in all but the Si spectra, which is the Si K α peak at 1.74 keV coming from the excitation of Si inside the dead layer of the detector. In the Si spectra, the Si K α peak is from the Si wafer. The intensity of the Si K α peak could also be increased by excitations of Si by X-rays from outside the beam, because of the Si wafer. Both O K α and C K α are also present in all spectra, which could be an oxide layer and contamination on the sample. Carbon coating on the sample by the beam was visible on the Cu-tape images, showing that there is carbon some contamination on the sample.

Another artifact present in almost all spectra is sum peaks, or coincidence peaks. No sum peak is visible in the NW spectrum, except for the possibility of the sum peaks of the L-lines of Ga and As overlapping with the Mo L-peaks. There are no sum peaks in the Cu spectra, but these spectra are as explained earlier questionable. Sum peaks are a peak where two X-rays are counted as one, giving a peak at the sum of the energies of the two X-rays. This occurs because the detector can only count one X-ray at a time, and since the counting does take some time, two X-rays can be registered as one. The process is dependent on the counting, which takes a certain amount of time, and this time is the dead time. In other words, a higher dead time will result in more signal in the sum peaks. A good example of this is the GaAs spectra, where the 5 and 30 kV spectra have DT at 33 and 30% and show almost no L-line sum peaks, while the 10 and 15 kV spectra have DT at 65 and 58% and clearly show the L-line sum peaks. The NW spectra have the lowest dead time, which might be why there are no sum peaks in the NW spectra, which fit well with the difference in the GaAs bulk spectra. Another explanation is that the NW spectra is the only one with a thin sample, which could result in less sum peaks. Labeling a sum peak correctly is important, else the signal can be mistaken for a characteristic peak.

One example of important labeling is the Si spectra, where the sum peak is close to the Sb L α peak. Analyzing this peak could be done by looking at the resolution and trying to figure out if the difference of 3.60 - 3.48 keV could be resolved, since the resolution at lower energies gets better. A faster and more reliable solution to this problem is to use the knowledge of the Sb L-peaks, which always come in a series of multiple peaks because of the splitting of the L-lines in Sb, as explained in [Section 2.1.2](#). This splitting effect of the Sb lines are visible in the NW spectra, where the Sb L-peaks are split into three peaks with heights matching the weights in HyperSpy. An even more reliable, but more time-consuming solution, could be to acquire multiple spectra with different DT. Dead time at around 30% was recommended by the supervisor of this project, Antonius T. J. von Helvoort, and is well below the problematic dead time of >60%. The dead time on the nanowire area was closer to 20%, which is what Goldstein [[1](#), page 223] recommends. An issue with the sum peaks when doing quantitative analysis, is that the signal in the sum peaks are supposed to be a part of the signal in the characteristic peaks. When looking at the Al spectrum, with DT at 22%, the sum peak is in the same size range as the other lower peak, which would clutter the quantification of the potential alloy.

The Al spectrum of the FIB stub are labeled with three peaks in the same size order as the Al K α sum peak. These three peaks are from Mg, Si and Mn, which could be present in the FIB stub. An Al alloy with Mg and Si is quite common, according to the supervisor of this project. An Al alloy of Al, Mg, Mn, Si, Fe and Cu is analyzed in an article by Wojciech [[19](#)], and an alloy like that would explain all the peaks and the very low but barely visible signal from Fe K α . Another possible explanation for the Mn peak is that the FIB stub used had some contamination, which could be tested by taking spectra of other FIB stubs. Determining how much of the Si peak is from the alloy and how much is the internal fluorescence peak could be done by taking spectra at different DT.

Another peculiar peak is the Ni K α peak in the NW spectra. The projects' supervisor claim that it is unlikely to be a Ni peak, as this have not been observed in the NW sample earlier. Making the nanowire sample was not a part of this thesis, only analyzing it with SEM EDS. The energy of the peculiar peak match well with the Ni K α energy at 7.50 keV,

but the peak might be an escape peak from Gs K α at 9.24 keV. An escape peak is an X-ray that excites Si in the detector, and thus loose 1.74 keV energy before being detected. However, an issue with this explanation is the peak at 0.85 keV, which would be the Ni L α peak. The peak is not very well-defined, as it overlaps much with the Cu L α and partly Ga L α peaks. Other escape peaks in the rest of the spectra are hard to identify, as the E-1.74 keV of the main peaks usually falls below zero. One exception is the Mo L α peak, where a potential escape peak align with O K α . In conclusion, it is not possible to determine certainly if the peak is an escape peak or a real Ni peak, but it could be a real Ni peak.

Other artifacts seen in the spectra are radiation signals from outside the beam. These signals come both from electrons which scatter in the beam and hit other parts of the sample or the chamber, and from X-rays which are generated and have enough energy to excite and generate new X-rays. The last one is what happens in the detector with the Si K α internal fluorescence peak. The secondary X-rays, i.e. X-rays generated from X-rays, are also emitted from the whole interaction volume. The interaction volume of X-rays is large, since the X-rays are emitted in all directions and do penetrate deep into the sample, as stated in [Section 2.1.1](#). Examples of this type of artifact is the Mo and Cu signal in the NW spectra. The grid is made of Cu, but the nanowire sampled is far from the grid. The Mo signal is from the Mo disk, which also is far from the sampled area. Signals from outside the sampled area can both be confusing and clutter the quantification, for example in the NW spectra where the sampled area is just the nanowire but the spectrum show signals from a larger area.

One last artifact to be discussed is the unknown peaks, which for instance is present in the NW spectrum at 0.38 keV. This energy is close to N K α at 0.392 keV. The Si spectra have an unknown at 0.08 keV, and this peak is very high in the 5 keV spectrum. If it is from a characteristic line, it could be Li K α at 0.054 keV or Be K α at 0.108 keV. Another unknown peak is present in the Mo spectra at 0.18 keV. The Mo unknown match best with B K α at 0.183 keV. Making conclusions about the unknown peaks would be unreliable conclusions, as all the unknown peaks are low in energy, and it is shown in [Table 4.3](#) that the low energy peak accuracy is not very good. In addition, the EDS system is not well-designed for the low energy peaks. This is partly because of the high absorption of the low energy X-rays, making detection difficult. The conclusion for the unknown peaks is that they stay unknown, and could be investigated further by experimenting with e.g. low acceleration voltages.

There are other details in the acquired spectra that could be discussed, but time is limited and even though details are interesting they are not always relevant to the main goal of a project. The details included were discussed because they were considered relevant and to show that the data was analyzed in detail.

5.2 Quantification in AZtec

[Sub-problem 1](#) was to do quantification in AZtec. The analysis in AZtec was done on the GaAs bulk data, and are given in [Table 4.4](#). The data was treated as both a SEM and a TEM signal, because of the limitations in HyperSpy, which are discussed in [Section 5.3](#). AZtec has a GUI for analysis, but it is not possible to see what is done with the data at the

different steps.

For all four spectra the quantification was most accurate when treating the data as a SEM signal. This was not unexpected, since the spectra in fact are SEM spectra. It is interesting that the most accurate SEM quantification at 15 kV, while the most inaccurate TEM quantification at 15 kV. In general, the quantification as SEM signals are very accurate, all within 5% of the correct value for Ga and As. The AZtec and HyperSpy quantification on the 5 and 30 kV spectra are very similar, which could imply that they use the same method, i.e. the CL method. However, the quantification on the 10 and 15 kV spectra are very different, because the HyperSpy quantification breaks down. This breakdown is covered more in [Section 5.6](#).

In general the quantification done on the correct type of signal is quite accurate in AZtec, but the problem is that the quantification is done as a black box. Adjusting the parameters of for example the background fitting or the peak finder is not possible, and the user cannot see what is done with the data at the different steps. An issue with this is visible in the initial AZtec plot in [Figure 4.1](#), where the Ga K β peak is not identified. The peak is clearly visible in the unscaled spectrum, but the peak finder in AZtec does not find it. Clicking on the peak opens a window where the user can select what peak this would correspond to, but the only options are a selection of elements where the peak is obviously wrong. Not being able to identify the peak that big should raise concern, especially since the peak is overlapping with the As K α peak and thus would affect the quantification. Since the quantification process is hidden, the user does not know if AZtec deals with the overlapping peaks correctly.

5.3 Analysis steps in HyperSpy

[Sub-problem 2](#) was to figure out what the steps are doing in the qualitative analysis in HyperSpy. HyperSpy have documentation online, and the following subsections explain how the analysis was done in HyperSpy.

Each subsection starts with some code lines, followed by an explanation of what the lines do. All variables inside crocodile need to be set by the user, e.g. `<element_list>` would be set to `['Ga', 'As']` for the GaAs wafer. An example notebook with quantification of the GaAs wafer is attached in the Appendix [Chapter A](#).

5.3.1 Loading the data and specifying the elements

```
s = hs.load(<filepath>, signal="EDS_TEM")  
  
s.set_elements(<element_list>)
```

The first step in the analysis is to load the data as a HyperSpy `signal` type, and specifying the signal as TEM. The `signal` type is a class in HyperSpy that contains the data and the metadata, and it has methods for analysis. The `signal` type must be specified as TEM, because the `signal` type for SEM is very limited and does not have a

method for quantification. When using .emsa files from AZtec, as is done in this project, the metadata contains some relevant and some irrelevant information. The information relevant later in this project is: acceleration voltage, dispersion, zero offset, energy resolution Mn K α , After loading, it is possible to plot the data with `s.plot()`.

Already at this point there is a big problem with using HyperSpy on the acquired data: the analysis methods are not implemented for SEM but for TEM. As explained by Skomedal in her master's thesis [6], the quantification with the Cliff-Lorimer method of TEM EDS data use approximations which are valid for thin samples. The approximation is that the sample is thin enough to ignore absorption and fluorescence, which is a bad approximation to use on bulk SEM EDS data. Still, the results show that it is possible to quantify the elements in the GaAs wafer with the Cliff-Lorimer method and get plausible results, but also that the analysis breaks down occasionally. Implementing quantification from the SEM type signal in HyperSpy have been discussed on the repositories GitHub issues page¹. The topic have been discussed in 2020, 2017 and 2015. People have been working on quantification of SEM EDS data, but the work have not been finished yet.

The quantification done with the CL method in HyperSpy on the three different calibrations in Table 4.7 are all within 10% of the expected composition. Carter and Williams [2, p. 612 and 648] state that the accuracy of EDS is at best \pm 3-5%, but that it could be reduced to around \pm 1.7% with very long acquisition times and a careful analysis. Carter and Williams claim that quantitative errors less than \pm 5-10% takes a lot of time and effort, and compositions with error below 5% should be regarded extra carefully and with suspicion. The best result, i.e. the result closest to 50%, from the HyperSpy CL quantification is strangely from the AZtec calibration with a 6.25% error. It is strange that the AZtec calibration is the best, because that is the calibration which misses most on the line accuracy in Table 4.3. This implies that the calibration is not the most important factor for the quantification, at least not when the calibration is around \pm 1%. The different calibrations are discussed further in Section 5.5. In the 30 kV GaAs sample the signal-to-noise ratio is good, the sample time was long and the peaks are well resolved, which make the input data good and allows for better quantification. The CL method is known to work on SEM data, but normally one would also do the ZAF correction, which is described in chapter 19.10.3 of Goldstein [1]. The ZAF correction adjust the signal for the atomic number effect, the absorption in the bulk of the sample and the X-ray fluorescence of new lines with lower energy than the initial characteristic X-ray [1, Sec. 19.10]. Since Ga and As are number 31 and 33 in the periodic table, it could be that the ZAF corrections are small and would not have a big effect on the results. When doing the quantification it was tested to include O and C, but that changed the results from plausible to completely wrong. The quantification analysis also broke down when using the 10 and 15 kV spectra, but the reason for that was not clear. The breaking down of the analysis is discussed further in Section 5.6.

Results in this project show that the CL method in HyperSpy for TEM EDS data sometimes yield plausible results on SEM EDS data and sometimes break down, thus the method should be used with caution. The goal of this project was not to do or to implement cutting edge quantification of EDS data from a SEM sample, but rather to understand

¹<https://github.com/hyperspy/hyperspy/issues/2332>

the analysis steps and what factors that could affect the results. Thus, using the quantification method from the TEM signal type was regarded as good enough for the principles tested in this project.

5.3.2 Removing the background linearly

```
bw = s1.estimate_background_windows(windows_width=<number>)
```

```
iw = s1.estimate_integration_windows(windows_width=<number>)
```

The next step is to remove the background, which with the above code is done by a linear fit. The background can be removed through model fitting, which is covered in [Section 5.3.4](#). The variable `windows_width` sets how wide the windows are for the background and integration, measured in FWHMs. A good starting value for `windows_width` is 2, but it should be tested by the user with a plot to see if the background will be removed correctly. The estimated windows can be plotted with:

```
s.plot(xray_lines=True, background_windows=bw, integration_windows=iw)
```

Plotting has proven to be a very useful tool for understanding what the different steps in the analysis do. Unfortunately, the implemented plotting in HyperSpy is limited. For example, when plotting a modelled spectrum with Ga and As, the model contains all the peaks in the HyperSpy library and all the peaks are plotted, but it is not possible to see what the different peaks are. In other words, the plot of a model can show all the independent components, but adding a legend is not an option. Another limitation of the plotting is the interactivity when saving the plot. Matplotlib requires the code to run in the background for it to be interactive, while Plotly can save the plot as an interactive HTML file for later inspection. One of the advantages with the plotting in HyperSpy is how accessible it is to plot in the different steps, which the user can use to understand and verify the analysis. The HyperSpy plotting was probably not made with the intention of being used for publication, but rather for quick and easy plotting of the data. If that is the case, the plotting serves its purpose well and has proven to be very useful for this project.

One of the more helpful insights from plotting in this project was that removing the background is not trivial. This is especially true for background signals modelled as a polynomial. Linear background removal works well for some peaks, for example the K-lines of Ga and As, but not for the L-lines. The background around the K-lines are more linear, because of the exponential decay of the background. Around the L-lines the background is affected by the absorption of the Bremsstrahlung lower than 2 keV and the observation of higher background levels below the highest peak in the spectrum. The background removal is discussed further in [Section 5.4](#).

5.3.3 Quantification after linear background removal

```
s_i = s.get_lines_intensity(background_windows=bw, integration_windows=iw)

k_factors = [<k-factor 1>, <k-factor 2>]

quant = s.quantification(s_i, method='CL', factors=k_factors)

print(f'E1: {quant[0].data[0]:.2f} \%, E1: {quant[1].data[0]:.2f} \%)
```

The quantification is done with the four lines of code above, where the last one prints the results. The first line gets the intensity of the peak corresponding to the lines of the specified element. HyperSpy selects automatically which lines to use for quantification. To see which lines are used, the `s_i` variable can be printed. The second line sets the k-factors. The k-factors in this project have been the one from AZtec, which are theoretically estimated. The third line does the quantification, where the method is specified. The method is the Cliff-Lorimer method, described in detail in Mari Skomedal's master thesis [6, Sec. 2.2.3]. HyperSpy has a method for quantification with the zeta factor method. The zeta method requires the value for the beam current, which was not measured in this project.²

The k-factors are essential for the Cliff-Lorimer quantification, and the k-factors listed in Table 4.5 give some insight into how AZtec calculates the k-factors. From the results it is clear that AZtec at least adjust the k-factor for element and for voltage. It could be that factors like the beam current and time live could affect the k-factor, but this was not investigated in this project. The L-line k-factors are start at 1.1 for 5 kV and increase to 1.2 for 15 kV. The As k-factor for the L-line at 5, 10 ad 15 kV is 11%, 7% and 6% greater than for Ga. The K-line k-factor are 3.3 for Ga and 4.2 for As, implying either a high sensitivity for higher voltages or two different models for calculating the k-factors. Another possibility is that the K-lines and L-lines have a separate model for calculating the k-factors. This can be tested by extracting the k-factors for the K-lines from the 15 kV spectrum, as the 15 kV spectrum do have peaks for both the K-lines and L-lines. The k-factors for Ga are in all four cases higher than for As, which makes sense, as all the Ga lines are higher and have more counts than the As lines. Quantification of the ratio between the Ga and As peaks are shown in Table 4.6, where it is clear that the k-factor push the ratio towards 1. The peak ratio times the k-factor for 5 kV is 1.1, and the next closest to 1 is 1.3 for 30 kV. This result implies that the theoretically calculated k-factors could be better for high and low voltages.

5.3.4 Removing the background with model fitting

Another way to remove the background is to fit a model to the data. This step would be done right after loading the data. If the raw data contains a zero peak, as is the case for most Oxford instrument EDS detectors, the zero peak needs to be removed before fitting the model. This requirement can easily cause problems and confusions, as this step is not

²Results from the zeta method can be converted to the cross section method, see the page "EDS Quantification" in the HyperSpy documentation.

very clear in the HyperSpy documentation. The reason for this being not that clear, is probably that different detectors have different zero peaks, and the zero peak is not always at the same place. The zero peak is removed by skipping the first n channels, where n=30 works well with the data from the GaAs wafer. The model fitting is done with the following code:

```
s = s.isig[<zero_peak_last_index>:]

m = s.create_model(auto_background=False)

m.add_polynomial_background(order=12)

m.add_family_lines(<list_of_element_lines>

m.plot()
```

The lines above removes the zero peak, create a model from the `signal` `s`, adds a 12th order polynomial, add the lines of the elements in the `signal`, and plot the model. This model is not fitted, it is just a generated spectrum with the lines of the elements. Eventually, the method `create_model()` can take the `auto_add_lines=True` argument, which will automatically detect the elements in the sample. The model consists of a number of components, which can be accessed with `m.components`. The components are all the Gaussian peaks in the spectrum, in addition to the background as a 12th order polynomial. The Gaussian components are based on the lines that the added elements in `<element_list>` have. If `auto_background` is not set to false, HyperSpy will add a 6th order polynomial. The order of the polynomial can be changed, but it should be tested by the user to see if it is a good fit. 6th order polynomials work, but as shown in [Figure 4.3](#), the 12th order polynomial has a lower RMS error and is thus preferred. Further, the model must be fitted.

```
m.fit()

m.plot()
```

The first line fits the model to the data to the components and the second line plots the model. HyperSpy have an own option for fitting only the background. Since the background is one of the components in `m`, it is fitted with the code line above.

5.3.5 Quantification after model fitting

```
m_i = m.get_lines_intensity()

k_factors = [<k-factor 1>, <k-factor 2>]

quant = s.quantification(s_i, method='CL', factors=k_factors)

print(f'E1: {quant[0].data[0]:.2f} \%, E1: {quant[1].data[0]:.2f} \%)
```

The quantification after model fitting is done in the same way as in [Section 5.3.3](#), but with intensity from the model instead of the signal. When modelling GaAs, the model can add the intensity from both K-lines and L-lines. Since AZtec only gives the k-factors for either the K-lines or the L-lines, the user must remove the lines without k-factors before quantification. As explained in [Section 5.1.1](#), overlapping peaks need to be modelled to get the correct intensities.

5.3.6 Calibrating the spectrum with the HyperSpy model

```
m.calibrate_energy_axis(calibrate='scale')

m.calibrate_energy_axis(calibrate='offset')
```

The two lines above calibrates the spectrum with the HyperSpy model and updates the dispersion and zero offset. The metadata in the `signal`s is updated with the new calibration. Thus, doing the previous step with quantification after model fitting can give a more correct quantification. This is the HyperSpy calibration that is used in [Table 4.7](#). As shown in the table, the HyperSpy calibration gives a better peak accuracy than the calibration from AZtec, and it is about as accurate as the self produced calibration done on the GaAs 30 kV spectrum. This self produced calibration is discussed in [Section 5.5](#), after a discussion of the peak and background modelling in the next section.

5.4 Peak and background modelling

[Sub-problem 3](#) was to find out how the peaks and the background can be modelled. The model was built without HyperSpy, with the idea of making the steps clear for a user. Once the model is fitted to the data, it can be used to calibrate and quantify the peaks and background, which is discussed in [Section 5.5](#). This can be done by comparing the peak ratios, i.e. the relative areas under the peaks, and using these ratios to calculate the Cliff-Lorimer quantification,

as shown in [Table 4.6](#). In the table the ratio is also multiplied with their respective k-factors, to get the Cliff-Lorimer quantification.

The first step in creating a model is to identify the peaks. The peaks are assumed to be Gaussian curves. Other possible shapes are Lorentzian curves, but that has not been a part of this project. The initial way of identifying peaks was that the user manually identified the peaks. Later the peaks were identified `scipy.signal.find_peaks()`, using the argument `prominence=0.01`. The prominence is the height of the peak above the background³. As seen in [Figure 4.3](#), the given prominence identifies many but not all the peaks. Further work on the peak finder is needed to find a good and robust way of identifying the peaks. The figure also shows that the peaks are well-fitted by the model, while the background is fitted fairly well.

Once the peaks have been identified in the data, the next step in creating a model is to fit a Gaussian curve to each peak and a polynomial to the background. In order to do this, the model components need initial guesses for their parameters. For the background, this means providing initial guesses for the coefficients of the polynomial. A good way to do this is to remove the peaks from the data using linear interpolation, and then fit a polynomial to the remaining data. The coefficients of the initial fitted polynomial can then be used as the initial guesses for the background in the model. Each Gaussian is described by its mean, standard deviation, and height, as shown in [Equation \(2.7\)](#). The mean is the peak position, which has its initial guess from the peak finder. The standard deviation is the width of the peak, where $\text{FWHM} = \text{std} * 2 * \sqrt{2 * \ln 2}$ ⁴. The height is the amplitude of the peak. The easiest way to get the initial guesses for the standard deviation and the height is to normalize the spectrum and set both parameters to 1. In the normalization the highest peak was set to 1, and the rest of the peaks were scaled accordingly. With the initial guesses, the whole model is ready to be fitted.

Normalization is a technique that is often used in data analysis to make the fitting process faster and more stable. In the case of this project, it was found that normalizing the data was necessary in order to avoid failure of the SciPy fitting algorithm. This failure is likely due to the fact that `scipy.optimize.curve_fit()` is optimized for certain types or ranges of data, and normalizing the data allows it to handle the data more effectively. It is important to note that although the data is normalized for the purpose of fitting, the quantification of the peaks is based on the ratios between the peaks, which are not affected by the normalization. This means that the normalization does not affect the accuracy of the quantification, but only the speed and stability of the fitting process. However, the perceived spectrum is affected by the normalization, which is important in the qualitative analysis. In this project, different normalization techniques were explored in order to find the one that provided stable results with SciPy, while still allowing the qualitative analysis to be performed.

The normalization that was used in the end was to set the highest peak to 1, and scale the rest of the peaks accordingly. This normalization, with some scaling and cropping of the figure, allowed the peaks to be visible and comparable,

³https://en.wikipedia.org/wiki/Topographic_prominence

⁴FWHM defined at: https://en.wikipedia.org/wiki/Full_width_at_half_maximum

while still allowing the qualitative analysis to be performed. No special features were observed to be lost by using this normalization. Some EDS users and textbooks prefer to use the logarithmic scale on the y-axis instead of normalizing the data. The logarithmic scale is a good way to make the spectrum easier to read, but it does not affect the fitting process. In addition, plotting with a logarithmic scale was tested while doing the qualitative analysis, but it was found that the logarithmic scale provided less detailed information on the smaller peaks and strays than the linear scale. In the initial AZtec plot in [Figure 4.1](#) the spectrum is plotted both with a linear and a logarithmic scale. Using the logarithmic scale is in the end a user preference. Thus, the linear scale with normalization was chosen as the best way to visualize the spectrum for the qualitative analysis, and as the best input for the fitting process.

With the model components ready and the data normalized, the model can be fitted to the data. The `curve_fit()` function uses the Levenberg–Marquardt algorithm to fit the model to the data, and returns the optimal fitted parameters. Fitting both the Gaussians and the background at the same time makes the fitting more stable. One of the first iterations, where the user manually inputted the peaks, the fitting tended to partially fail. The issue was that the fitting only was done on the peaks. To minimize the error in the fitting, one of the Gaussian curves with a low amplitude was moved and got a huge standard deviation, which compensated the background. This was fixed by fitting both the Gaussians and the background at the same time. Doing this made the fitting both better, and it failed less often.

The issue of severely overlapping peaks was encountered when fitting the Mo spectrum. The Mo L α and L β peaks are very close to each other, and even though the fitting algorithm would handle this, the peak finder would not recognize the overlapping peaks as two separate peaks. This problem can be fixed manually, but with the limited time in this project it was categorized as an edge case that would not be handled.

5.5 Calibration

[Sub-problem 4](#) was to figure out how good the calibration is in AZtec and HyperSpy, which lead to making a new calibration in Python. The calibration has been discussed some in [Sections 5.1.1](#), [5.3.1](#) and [5.3.6](#), and is here discussed more in depth.

Calibration is an important step in the analysis, as it ensures that the results obtained from the analysis are accurate and reliable. If the dispersion is a few percent off, the error in the analysis could be significant as the error is accumulates with higher energies. In general, calibration involves the comparison of the measured values with a known reference standard, in order to determine the accuracy and precision of the measurement system. In this project, the reference was two known peaks. Calibration process is important because it allows the user to account for biases or errors in the measurement system, and to correct for them in order to obtain more accurate results. Additionally, calibration can also help to identify any potential problems or issues with the measurement system, allowing the user to take appropriate action to address these issues. A well calibrated system is important in high accuracy measurements.

When working with AZtec, the data is read with a built-in calibration. It might be an option to calibrate the data in

AZtec, but this was not investigated in this project. One thing is sure, and that is that AZtec does not calibrate the spectrum automatically, as all the spectra were extracted with the same calibration. It might be that the calibration of the SEM APREO EDS detector was set e.g. upon installation, and have not been changed since then. Since the AZtec calibration had the worst peak accuracy, it would be interesting to see if changing the calibration of the instrument would change the black box quantification in AZtec. This step was not included in the scope of this project.

A step that was included in the scope of this project was to calibrate the data in HyperSpy. The calibration is described in [Section 5.3.6](#), which gave a dispersion 2.8% higher than the calibration in AZtec and one more channel in the zero offset. The higher dispersion is shifting the whole spectrum to the right, because the energy per channel is higher. The higher zero offset is shifting the spectrum to the left, as more channel values are negative, which does not have a real life interpretation. This left and right shift is making the whole spectrum fit the peaks better, as quantified in [Table 4.3](#). The HS calibration was done on GaAs, and gave very similar results as the self-made calibration.

The self-made calibration was made with two goals in mind: to understand the calibration process better and to verify the calibration in HyperSpy. HyperSpy is open-source software, but reading the source code is not the most efficient way to understand how the calibration is done. The calibration in HyperSpy is done with a model of the spectrum, which is fitted to the data. Thus, the self-made calibration was based on the same principle. A model was made and fitted, and two far apart peaks were used for calibration. Which peaks to use is discussed below. The self-made calibration was done on two spectra, GaAs and Mo. The Mo spectrum gave a dispersion 4.0% higher than the calibration in AZtec, and it was probably too high, based on the poorer peak accuracy. The Ga spectrum gave a dispersion 3.0% higher than the calibration in AZtec, and the peak accuracy was similar to the calibration in HyperSpy. Quantified with the RMSD, the self-made calibration is better than the HS calibration, but too few spectra have been tested to make a final decision of which calibration is better. The HS and self-made calibration perform quite bad at C K α , but this peak is not as useful. Carbon is often an impurity in EDS samples, both present in the chamber and potentially on the sample.

The peaks used for calibration should be far apart and have a good signal-to-noise ratio. The channels between the peaks are interpolated, while the channels outside the peaks are extrapolated. Extrapolated channels can potentially get a larger error, as error in the dispersion accumulates. The peaks need a good signal-to-noise ratio, because the fit needs to be good to find the correct peak position. The problem with the Mo spectrum, where the peaks are further apart, was probably that the Mo K α peak was not good enough to fit, because the signal was low. The value of the Mo K α peak is around 2% of the Mo L α peak, while the lower calibration peak in the GaAs spectrum is 20% of the higher peak. The problem with the Mo spectrum could have been a poor fit of the Mo L α peak, which is overlapping with the Mo K α peak, because the peak finder struggles with this severely overlapping peak. However, the overlapping problem was solved manually for the calibration on the Mo spectrum. If the problem was lower signal-to-noise ratio of Mo K α , the problem could have been solved by using a longer counting time. This could be investigated in the future, to try to optimize the calibration of the detector.

One last, but important, aspect is the theoretical value of the peaks. To get the calibration right, the reference peaks need to be at the correct energy. In this work, it was discovered that the theoretical values in the X-ray Data Booklet are slightly deviating from the HyperSpy values. The first thing to note is that the lines listed in the XRB and in HS deviate somewhat, as seen in [Tables 2.1](#) and [2.2](#). Not all the lines in the table are visible in the spectra in [Figures 4.8](#) and [4.9](#). The XRB differentiates between $K\alpha_1$ and $K\alpha_2$, and states that the weight of Ga $K\alpha_2$ is 0.51 relative to Ga $K\alpha_1$. If this was true, the Ga $K\alpha$ peak would be an overlapping peak with a broadening on the right side, but as seen in [Figure 4.4](#), this is not the case. The figure show that the peak fit exactly one Gaussian curve, with its center directly on the $K\alpha$ line from HS, which is the $K\alpha_1$ line in the XRB. The XRB claim that the listed lines are the strongest lines [[5](#), P. 33, Sec. 1.2], but it is missing the Mo $L\gamma_3$ line, which is visible in the Mo spectrum. In general, the weights of the lines listed in the XRB and HS seem to be slightly off from the data collected in this project, because some lines are not visible and some are, but this cannot be explained solely by the weights. It could be that the listed weights would be correct when collecting the data differently or with different materials.

5.6 Analysis failure

[Sub-problem 5](#) was to find out when the analysis fails, both in AZtec and HyperSpy. The time spent on this sub-problem was limited compared to what would have been preferred to get a clearer understanding of both when and why the analysis fails. The observations done are mostly on when the analysis fails, and not why. Why the analysis fail would be a good starting point to improve the EDS analysis.

The fact that the quantification fails when the SEM EDS data is treated as TEM data was expected. As stated earlier, when doing TEM analysis there are some key assumptions which are not valid for SEM data. When analyzing SEM EDS data it is necessary to do the ZAF correction, which was described briefly in [Section 5.3.1](#). Implementing the ZAF matrix correction in HyperSpy for the SEM signal would be a helpful contribution to the HyperSpy project, as this has been mentioned as one key missing feature to do SEM EDS quantification with HyperSpy in the GitHub discussion threads.

One of the strangest results in this project was that the quantification failed at 10 and 15 kV in HyperSpy. The exact same CL quantification code was used on the 5, 10, 15 and 30 kV spectra, but only the 5 and 30 kV spectra gave a reasonable result. The most puzzling thing is that the code gives results on the 5 and 30 kV spectra which looks similar to the TEM quantification results in AZtec, so the code described above in [Section 5.3](#) is not completely wrong. The four spectra look quite similar. One difference is that the 10 and 15 kV spectra have a dead time which was around twice as long as the 5 and 30 kV spectra. The different DT is not a sufficient explanation, it is just an observed difference. The author did not have time to dig deep enough into the source code of HyperSpy to find the reason for the failure. Possible explanations are the selection of lines to use in the quantification, an error in the code or the extracted data, a bug in HyperSpy for certain settings, something with the data treatment as TEM data, or something else. The suggestions are

not really helpful, but could be a starting point for further investigation.

Chapter 6

Conclusion and future work

The conclusion chapter covers the main finding and the limitations of the work done in this project and ideas future work.

6.1 Main findings

This project on analyzing and trying to improve the analysis of EDS data from SEM. It has been a learning process for the author, building knowledge for future projects on data driven analytical work. For the given studied set-up, some improvements have been apparent. The project have dealt with how data processing influence both the qualitative and quantitative analysis of EDS data. The main findings are improvements on calibration of the energy scale, how the background could be modelled better, how the HyperSpy code works and code for direct analysis of EDS data in Python. The limitations were the authors experience with the topic and the available quantification methods in HyperSpy.

The results from the calibrations examined in this project show that the peak deviations are relatively small, compared to the energy resolution of the detector. The reported energy resolution of the detector is 127 eV, while the deviations are from 1-20 eV for the best calibration, and 3-50 eV for the built-in calibration of the AZtec software. Thus, it might be beneficial to recalibrate the energy scale, but it is assumed not to be critical. Calibration in the SEM APREO at NTNU NanoLab can be changed a few percent to improve the peak accuracy, which could be beneficial for the quantitative analysis. The calibration should be done with 30 kV spectra with far apart peaks to get clear peaks and little extrapolation. Calibration was done with Mo L α and Mo K α peaks, but these results were not the best. It was found that using Ga L α and As K α peaks gave the best results, even though the peaks are not as far apart as the Mo peaks. This was concluded to be because of the peaks being more well-defined in the GaAs spectrum. The GaAs calibration was quantified to have three times lower RMS error than the built-in calibration in AZtec. If a recalibration is done,

it would probably be beneficial to take new spectra and calculate the calibration on multiple spectra. When doing the calibration calculation, HyperSpy provides a good tool, but the calculation could also be done with the code produced in this project, which is more transparent and easier to understand.

While working with the calibration and model fitting, it was found that the fit of the background as a polynomial tends to be poor at the lower energies, which is where the highest peaks are located. Modelling the background is a better way to get the peak intensities compared to just using a window around the peaks and raw counts. The fit of the background is partially poor because the intensity of the background is higher before than after the main peaks in the spectrum. The main peak or peaks are the highest peaks in the spectrum, which usually are located around 1-3 keV, and this is the area where the intensity and shape of the background is changing the most. An observed trend is that the background is more linear until the most intense peak, and then after the peak it both drop in intensity and follows the expected background shape.

A section of the discussion is covering what steps in HyperSpy was done, and what these steps do. Working with open-source software allows the user to both understand the steps and alter the analysis to fit the needs of the user. Even though it can take more time than the black-box software, open-source software gives huge advantages which allowing better results. Two of the ultimate goals would be to make the quantification in HyperSpy more accurate and to implement SEM EDS data quantification in HyperSpy. The HyperSpy code needs to be robust and general, which means that the code needs to handle many edge cases. Implementing absorption and fluorescence correction in HyperSpy is beyond the scope of this project.

In addition to working with the open-source software HyperSpy, the EDS data acquired was analyzed directly in Python. The Jupyter notebooks used for the analysis are available on GitHub, and the notebooks are a good way to show how the analysis is done. The steps which are covered in the notebooks are calibration, peak fitting, background subtraction and quantification. Using the notebooks can enhance the understanding of the analysis, because the notebooks are not made to handle edge cases and are not as general as the HyperSpy code. This allows the notebooks to be a better tool for understanding the steps in the analysis, while the HyperSpy code is a better tool for general analysis.

The author had little experience with the topic of EDS analysis before the project, both with the analysis and the data acquisition. Acquiring good EDS spectra does require some experience, which the author did not have before the project. This is probably the source of the poor quality of the Cu-tape spectra acquired. In addition, the author did not have deeper knowledge of EDS data analysis when the spectra were taken, which could have made the acquisition better. Much of the theoretical knowledge of EDS analysis was gained during the project. In other words, much of the time spent on the project was spent on learning the topic. Now that the author has more knowledge of the topic, the master thesis project which will be done during the spring of 2023 will benefit of the knowledge gained in this project.

One of the major limitations, which the author might start to work on during the master thesis project, is the limited quantification methods in HyperSpy for SEM data. The quantification methods in HyperSpy are made for TEM

data, and thus the quantification done in this project is based on some assumptions which are not valid for SEM data. However, many of the steps in the analysis are independent of the data being SEM or TEM, such as calibration, peak fitting, background subtraction. Acquiring SEM data with good signal-to-noise ratio gives a good foundation for the analysis, and with the knowledge gained in this project, the author could be able to improve the EDS analysis in a more general way during the master thesis project.

6.2 Future work

The aim of this project was to improve EDS analysis by using open-source packages. For qualitative analysis, this has been achieved for energy calibration and identifying spectral artifacts. For the qualitative analysis, progress have been made in understanding the fitting of the background with a better model. Ideas for future work based on the findings of this project are separated into three categories: i) improvements for the current code developed, ii) implementations for HyperSpy and iii) general ideas for future work in EDS analysis.

- i) In the discussion chapter there are multiple ideas for improvements of the current code developed in this project. One of the ideas is to model the background as a spline, which would improve the background subtraction and peak fitting. Another idea is to fine tune the peak finder and combine it with the lines in the HyperSpy library, which can improve the peak fitting. The current spectra can be used, but new spectra with e.g. different beam currents on 30 kV should be taken to analyze the fit of the new spectrum model on different data. Data with poorer signal-to-noise ratio should also be taken to analyze the effect of the improved model on these types of spectra, for example by applying denoising routines [20]. Acquiring data with different dead times would also be interesting, to try to see if artifacts like sum peaks can be treated with data processing.
- ii) Adding quantification methods for SEM data to HyperSpy would be a great improvement of the package. Much of the basis for the quantification methods in HyperSpy is available in the TEM signal module, thus the needed new code would mainly be the ZAF matrix correction [1,10]. Another idea is to get a better overview of all the methods available on TEM data, and figure out which ones work on SEM data. Then these methods, for example machine learning tools, could be used on SEM data to improve the analysis in AZtec. In other words, the analysis could be extraction the data from AZtec, applying some methods in HyperSpy and then quantifying the data in AZtec. It would allow the user to use the best of both worlds, the open-source software and the closed-source software.
- ii) Another idea is to enhance the plotting of models in HyperSpy. A simple and effective improvement would be to add a legend with explanations to the plots, which would make the plots more understandable. That would allow the users to see a clearer connection between the spectrum and the listed excitation lines. Since HyperSpy makes models with the excitation lines, it would be easier to find the strays in the spectrum, because they do not get a line in the model. One of the advantages with Plotly over Matplotlib is the possibility to hide plotted elements in the plot with a click, allowing the user to see clearer the details of the spectrum. Making Plotly a possible plotting backend for HyperSpy

would be an improvement of the package, and would allow the user to interact with the plots in a more intuitive way. This change, however, would probably require a lot of work, since the plotting backend is a core part of HyperSpy.

iii) Some general ideas for future work in EDS analysis emerged during the project. One of the ideas is to make an optimized workflow for data acquisition. The data acquired have some flaws, which could be avoided by an optimized workflow. One of these flaws is the poor quality of the Cu-tape spectra, which could be avoided by an optimized workflow. Another idea is to figure out better how AZtec calculates the k-factors, because they are a central part of the Cliff-Lorimer quantification method. This could be done by experimenting with different settings while acquiring data, and then analyzing the effect of the settings on the k-factors.

For the author's master thesis project, the work with improving EDS analysis will continue, and the ideas from this project will be the starting points. Improvements on the current code developed in this project will be done, but the main goal will be to contribute to HyperSpy. Making Jupyter notebooks with examples of how to analyze EDS data is nice and can help a user understand what is happening, but making meaningful contributions to HyperSpy will potentially have a bigger impact.

Bibliography

- [1] J. I. Goldstein, D. E. Newbury, J. R. Michael, N. W. Ritchie, J. H. J. Scott, and D. C. Joy, *Scanning Electron Microscopy and X-Ray Microanalysis*. New York, NY: Springer New York, 2018.
- [2] C. Carter and D. Williams, *Transmission Electron Microscopy: Diffraction, Imaging, and Spectrometry*. Springer International Publishing, 2016.
- [3] J. M. Hollas, *Modern spectroscopy*. Chichester ; Hoboken, NJ: J. Wiley, 4th ed ed., 2004.
- [4] M. Siegbahn, “Relations between the K and L Series of the High-Frequency Spectra,” *Nature*, vol. 96, pp. 676–676, Feb. 1916.
- [5] A. C. Thompson, D. T. Attwood, M. R. Howells, J. B. Kortright, A. L. Robinson, J. H. Underwood, K.-J. Kim, J. Kirz, I. Lindau, P. Pianetta, H. Winick, G. P. Williams, and J. H. Scofield, *X-Ray Data Booklet*. Lawrence Berkeley National Laboratory, University of California, 2nd ed ed., 2004.
- [6] M. S. Skomedal, “Improving quantitative EDS of III-V heterostructure semiconductors in low voltage STEM,” Master’s thesis, Norwegian University of Science and Technology, 2022.
- [7] F. de la Peña, E. Prestat, V. T. Fauske, P. Burdet, J. Lähnemann, P. Jokubauskas, T. Furnival, M. Nord, T. Ostaszewicus, K. E. MacArthur, D. N. Johnstone, M. Sarahan, T. Aarholt, J. Taillon, pquinn dls, V. Migunov, A. Eljarrat, J. Caron, C. Francis, T. Nemoto, T. Poon, S. Mazzucco, actions user, N. Tappy, N. Cautaerts, S. Somnath, T. Slater, M. Walls, F. Winkler, and DENSMerijn, “hyperspy/hyperspy: Release v1.7.1,” June 2022.
- [8] N. Brodusch, K. Zaghib, and R. Gauvin, “Improvement of the energy resolution of energy dispersive spectrometers (eds) using richardson–lucy deconvolution,” *Ultramicroscopy*, vol. 209, p. 112886, 2020.
- [9] G. Cliff and G. W. Lorimer, “The quantitative analysis of thin specimens,” *Journal of Microscopy*, vol. 103, no. 2, pp. 203–207, 1975.
- [10] B. Löddings and L. Reimer, “Energy dispersive x-ray microanalysis on tilted specimens using a modified zaf correction,” *Scanning*, vol. 1, no. 4, pp. 225–229, 1978.

- [11] N. Ritchie, “Getting started with nist* dtsa-ii,” *Microscopy Today*, vol. 19, 01 2011.
- [12] O. Instruments, “X-max 80 - oxford instruments,” 2008.
- [13] D. Lundeby, “Improving the accuracy of TEM-EDX quantification by implementing the zeta-factor method,” Master’s thesis, Norwegian University of Science and Technology, 2019.
- [14] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [15] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, “Jupyter notebooks – a publishing format for reproducible computational workflows,” 2016.
- [16] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, pp. 357–362, Sept. 2020.
- [17] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [18] Plotly, “Plotly technologies inc. - collaborative data science,” 2015. <https://plot.ly>.
- [19] O. Wojciech, Stanisław, E. Kalinowska-Ozgowicz, A. Kowalski, and K. Golombek, “The structure and mechanical properties of al-mg-mn alloys shaped in the process of thermomechanical treatment,” *Journal of Achievements in Materials and Manufacturing Engineering*, vol. 45, pp. 148–156, 01 2011.
- [20] M. Moreira, M. Hillenkamp, G. Divitini, L. H. G. Tizei, C. Ducati, M. A. Cotta, V. Rodrigues, and D. Ugarte, “Improving quantitative eds chemical analysis of alloy nanoparticles by pca denoising: Part i, reducing reconstruction bias,” *Microscopy and Microanalysis*, vol. 28, no. 2, p. 338–349, 2022.

Appendices

Appendix A

HyperSpy quantification

The following notebook is quantification in HyperSpy. The notebook does three quantifications: linear background removal and intensities from the raw spectrum, intensities from a fitted model, and intensities from a model fitted with calibration in HyperSpy.

HyperSpy-quantification

December 19, 2022

1 Quantification in HyperSpy

```
[1]: import hyperspy.api as hs
```

1.1 Quantification with linear background removal

```
[2]: path="data/2022-09-06_EDS-SEM-APREO/"
file = "GaAs_30kV.emsa"
s = hs.load(path+file, signal_type='EDS_TEM')
s.add_elements(['As', 'Ga'])
# s.plot(xray_lines=True)

[3]: bw = s.estimate_background_windows(windows_width=2)
iw = s.estimate_integration_windows(windows_width=2)
# s.plot(xray_lines=True, background_windows=bw, integration_windows=iw)

[4]: s_intensities = s.get_lines_intensity(background_windows=bw,
                                          integration_windows=iw)
k_factors = [4.191, 3.268] # As, Ga
quant = s.quantification(s_intensities, 'CL', factors=k_factors)
print(f'{file} quantification\nAs: {quant[0].data[0]:.2f} %, Ga: {quant[1].data[0]:.2f} %')

#####
##### | 100% Completed | 118.19 ms
GaAs_30kV.emsa quantification
As: 56.95 %, Ga: 43.05 %
```

1.2 Quantification with intensity from model fit

```
[5]: # model fit
m = s.create_model(auto_background=False)
m.add_polynomial_background(order=6)
m.add_family_lines(['As_Ka', 'Ga_Ka'])

m.fit()
'm'
# m.plot(True)
```

```
[5]: 'm'

[6]: m_intensities = m.get_lines_intensity()
for bs in m_intensities:
    print(bs.metadata.Sample.xray_lines)
k_factors = [4.191, 3.268] # As, Ga
m_intensities[0::2] # selects As_Ka and Ga_Ka
quant = s.quantification(m_intensities[0::2], 'CL', factors=k_factors)
print(f'{file} quantification with model fit')
print(f'As: {quant[0].data[0]:.2f} %, Ga: {quant[1].data[0]:.2f} %')

['As_Ka']
['As_La']
['Ga_Ka']
['Ga_La']
##### | 100% Completed | 109.13 ms
GaAs_30kV.emsa quantification with model fit
As: 41.31 %, Ga: 58.69 %
```

1.3 Quantification with intensity from model fit, after calibration

```
[7]: # quanitfication after calibration in HyperSpy

m.calibrate_energy_axis(calibrate='scale')
m.calibrate_energy_axis(calibrate='offset')

m_intensities = m.get_lines_intensity()
for bs in m_intensities:
    print(bs.metadata.Sample.xray_lines)

k_factors = [4.191, 3.268] # As, Ga
m_intensities[0::2] # selects As_Ka and Ga_Ka
quant = s.quantification(m_intensities[0::2], 'CL', factors=k_factors)
print(f'{file} quantification with model fit')
print(f'As: {quant[0].data[0]:.2f} %, Ga: {quant[1].data[0]:.2f} %')

# m.plot(True)

['As_Ka']
['As_La']
['Ga_Ka']
['Ga_La']
##### | 100% Completed | 106.70 ms
GaAs_30kV.emsa quantification with model fit
As: 41.34 %, Ga: 58.66 %
```

Appendix B

Peak and background modelling

This is a notebook for calibration. The calibration is done on the 30 kV spectrum of GaAs bulk.

calibration

December 19, 2022

```
[1]: # Notebook for the custom calibration

[2]: import numpy as np
      import plotly.graph_objects as go
      from scipy.signal import find_peaks
      from scipy.optimize import curve_fit

      from utils.get_emsa_data import get_counts_and_name

[3]: # helper functions important for the calibration

# converts channel to keV
def channel_to_keV(
    value=None, arr=None, dispersion=0.0100283, offset=21.078, use_offset=True
):
    if not use_offset:
        offset = 0
    if value is not None:
        return (value - offset) * dispersion
    if arr is not None:
        return (np.array(arr) - offset) * dispersion
    else:
        raise ValueError("No value or array provided to chennel_to_keV(...)")

# this function makes one gaussian
def gaussian(xaxis, mu, amp, std):
    return amp * np.exp(-((xaxis - mu) ** 2) / (2 * std**2))

# this function makes n gaussians and a m=deg order polynomial on xaxis
def n_gaussians_and_m_order_bg(xaxis, deg, *args):
    poly = args[: deg + 1]
    gauss = args[deg + 1 :]
    n = int(len(gauss) / 3)
    y = np.zeros(len(xaxis))
    for i in range(n):
        y += gaussian(xaxis, gauss[3 * i], gauss[3 * i + 1], gauss[3 * i + 2])
```

1

```

y += np.polyval(poly, xaxis)
return y

# this function removes the background linearly, used to make an initial bg
guess
def remove_linear_bg(y, peaks, pixel_removal=40, offset=10):
    bg = y.copy()
    for p in peaks:
        # if the peak is too close to the edge, just say the bg is 0
        if p < pixel_removal:
            bg[:p] = 0
            continue
        else:
            bg[p - pixel_removal : p + pixel_removal] = np.nan
        # make linear interpolation between nan values
    bg = np.interp(np.arange(len(bg)), np.flatnonzero(~np.isnan(bg)), bg[~np.isnan(bg)])
    bg[: int(offset)] = 0 # first 10 channels or the offset should be 0
    return bg

# This function makes the mu, amp, std and polynomial coefficients
def make_fit(array, x, deg=12, prominence=0.01, pixel_removal=50, offset=20):
    # fits one array to n gaussians and a m order polynomial
    peaks, _ = find_peaks(array, prominence=prominence)
    std = np.ones(len(peaks))
    amp = np.ones(len(peaks))

    # must define the function to cure_fit, since deg is a variable which is
not allowed in the function,
    # because it is not a parameter of the function that curve_fit is trying to
fit
    def fit_func_peaks_and_bg(x, *args):
        poly = args[: deg + 1]
        gauss = args[deg + 1 :]
        n = int(len(gauss) / 3)
        y = np.zeros(len(x))
        for i in range(n):
            y += gaussian(x, gauss[3 * i], gauss[3 * i + 1], gauss[3 * i + 2])
        y += np.polyval(poly, x)
        return y

    # removing the background linearly for an initial guess of the polynomial
    bg = remove_linear_bg(array, peaks, pixel_removal=pixel_removal,
offset=offset)
    # fitting the polynomial

```

```

poly_init = np.polyfit(x, bg, deg)

# fitting
init_vals = list(poly_init)
for i in range(len(peaks)):
    init_vals += [peaks[i], amp[i], std[i]]
fit_vals, covar = curve_fit(fit_func_peaks_and_bg, x, array, p0=init_vals)

return fit_vals, covar

def calculate_calibration(peaks_channel, peaks_keV):
    """
    Calibration of the channel width and offset using two peaks.
    The given channels and keV must correspond.

    Parameters
    -----
    peaks_channel : list
        channel value of peaks, fitted
    peaks_keV : list
        theoretical energy of the two peaks

    Returns
    -----
    float, float
        dispersion, offset
    """

    # figure out the distances between the peaks
    channel_distance = peaks_channel[1] - peaks_channel[0]
    kev_distance = peaks_keV[1] - peaks_keV[0]

    # dispersion = (p1_keV - p0_keV) / (p1_channel - p0_channel)
    dispersion = kev_distance / channel_distance # kev_per_channel
    offset = peaks_channel[0] - peaks_keV[0] / dispersion # calibrated_zero_channel

    print(
        f"Calibration: {dispersion:.08f} keV/channel, {offset:.03f} channels\u2070 zero offset"
    )

    return dispersion, offset

```

```
[4]: # variables used
channels = np.arange(2048)
```

```

ga_La = 1.098
as_Ka = 10.5436
peaks_keV = [ga_La, as_Ka]
initial_dispersion = 0.01
initial_offset = 20
deg = 12
prominence = 0.01

file = "GaAs_30kV"

ga30, _ = get_counts_and_name(file)
spectrum = ga30

```

```

[5]: # find peaks
peaks_channel_est, _ = find_peaks(spectrum, prominence=0.01)
peaks_keV_est = channel_to_kev(peaks_channel_est, dispersion=initial_dispersion, offset=initial_offset)

# find the peaks that are closest to the peaks_keV
peaks_to_fit_index = []
for peak in peaks_keV:
    peaks_to_fit_index.append(np.argmin(np.abs(peaks_keV_est - peak)))

fit_vals, _ = make_fit(spectrum, channels, deg=deg, prominence=prominence)

fit_peaks = fit_vals[deg + 1::3]
dispersion, offset = calculate_calibration(fit_peaks[peaks_to_fit_index], peaks_keV)
fit_peaks_keV = channel_to_kev(fit_peaks, dispersion=dispersion, offset=offset)
keV = channel_to_kev(channels, dispersion=dispersion, offset=offset)

```

Calibration: 0.01003001 keV/channel, 21.127 channels zero offset

```

[6]: fig = go.Figure()
fit = n_gaussians_and_m_order_bg(channels, deg, *fit_vals)
fig.add_trace(go.Scatter(x=keV, y=spectrum, name='raw data', mode='markers'))
fig.add_trace(go.Scatter(x=keV, y=fit, name='fit'))
fig.add_vline(x=fit_peaks_keV[peaks_to_fit_index[0]], line_width=2, line_dash="dash")
fig.add_vline(x=fit_peaks_keV[peaks_to_fit_index[1]], line_width=2, line_dash="dash")
fig.update_layout(title=f'Calibration of the spectrum with {file}')
fig.update_layout(xaxis_title='Energy [keV]', yaxis_title='Relative intensity [a.u.]')
fig.show()

```

Calibration of the spectrum with GaAs_30kV

