NTNU
Norwegian University of
Science and Technology

Refinement of the ζ-factor Method for Quantitative Energy-Dispersive X-ray Spectroscopy in Scanning Transmission Electron Microscopy

Andreas Garmannslund

Nanotechnology
Submission date: June 2016
Supervisor: Antonius Theodorus Johann Van Helvoort, IFY

Norwegian University of Science and Technology
Department of Physics

Abstract

The ζ-factor method for quantitative energy-dispersive X-ray spectroscopy (EDS) has been implemented using the open source Python-based HyperSpy library. The ζ-factors for Ga, As, Al, C and Sb have been determined experimentally for a JEOL ARM200F with a JEOL Centurio silicon drift detector. Least-square fitting has been applied to estimate the ζ-factor for other elements. The ζ-factor method is refined by investigating orientation and tilt effects on the ζ-factors using GaAs nanowires as a model system. The most prominent effect is shadowing of X-rays from the specimen holder, carrier or grid, which leads to a higher ζ-factor being measured when the specimen is tilted away from the detector. Tilt series were taken from the GaAs core of a focused ion beam (FIB) cross section of an GaAs/AlGaAs core-shell nanowire in [111] and around two perpendicular axes of a the GaAs region of whole GaAs/GaAsSb axial-insert nanowires in $[\bar{1}10]$. The ζ-factors allows for fitting shadowing models based on different detector geometries. For the used experimental set-up, shadowing does not occur (i.e. the active detector area and hence the signal is maximized) for specimen tilt (X-tilt) above 10° towards the detector. Quantitative compositional and thickness maps have been constructed using the refined ζ-factors for axial GaAs/GaAsSb and radial GaAs/AlGaAs nanowire heterostructures. The compositional maps were compared with maps constructed using the standard Cliff-Lorimer (CL) ratio technique, while thickness maps were compared with energy-filtered transmission electron microscopy (EFTEM) maps. Incorporating absorption corrections into the ζ-factor method gives only a small improvement in the accuracy of the quantification for the FIB made GaAs/AlGaAs and whole GaAs/GaAsSb nanowires. The refined ζ-factor method gives better results than the CL ratio technique in the regions of pure GaAs, while CL gives better results in the alloyed regions. This is believed to be due to uncertainties in the thickness determination used for calibrating the ζ-factors, especially for Al. Computational routines based on the ζ-factor method have been developed and made accessible for others. The routines can easily be adapted and applied to user-specific data sets as they are developed within an open source software platform.

Sammendrag

ζ-faktor metoden for kvantitativ energidispersiv røntgenspektroskopi har blitt implementert ved bruk av det Python baserte åpne kildekodebiblioteket HyperSpy. ζ-faktorer for Ga, As, Al, C og Sb har blitt bestemt eksperimentelt for en JEOL ARM200F med en JEOL Centurio silisiumdriftdetekor. Minste kvadraters metode har blitt anvendt til å estimere ζ-faktorer for andre elementer. ζ-faktor metoden har blitt utbedret ved å undersøke hvilken påvirkning orientering og tilt har på ζ-faktorene ved å bruke GaAs nanotråder som modellsystem. Den mest fremtredende effekten er skyggelegging av røntgenstråler fra prøveholderen eller gitteret, som fører til at de målte ζ-faktorer øker i verdi når man tilter prøven vekk fra detektoren. Tiltserier ble tatt fra GaAs kjernen av tversnitt, laget ved bruk av fokusert ionestråle, av en GaAs/AlGaAs kjerne-skall nanotråd i [111] og rundt to vinkelrette akser av GaAs segmentet fra hele GaAs/GaAsSb aksiale nanotråder i $[\bar{1}10]$. ζ-faktorene har blitt brukt til å tilpasse skyggeleggingsmodeller basert på ulik detektorgeometri. I det anvendte utstyret er skyggeleggelse ikke tilstedeværende (med andre ord, det aktive detektorarealet og dermed signalet er maksimalt) for når prøven er tiltet mer enn 10° mot detektoren. Kvantitative bilder av sammensetning og tykkelse har blitt lagd ved å bruke de forbedrede ζ-faktorene for heterogene aksielle GaAs/GaAsSb og radielle GaAs/AlGaAs nanotråder. Sammensetningsbildene har blitt sammenlignet med bilder som er laget ved å bruke den mer vanlige Cliff-Lorimer forholdsmetoden, mens tykkelsesbilder har blitt sammenlignet med energifiltrerte transmisjonselektronmikroskopi bilder. Absorpsjonskorrigert ζ-faktor metode gir bare en liten presisjonsforbedring for de undersøkte nanotrådene. Den forbedrede ζ-faktor metoden gir bedre resultater enn Cliff-Lorimer forholdsmeoden i områder bestående av ren GaAs, mens Cliff-Lorimer gir bedre resultater i de legerte områdene. Årsaken til dette er antatt å være usikkerhet i tykkelsesmålingene av prøvene som ble brukt til å kalibrere ζ-faktorene, spesielt for Al. Datarutiner basert på ζ-faktor metoden har blitt utviklet og gjort tilgjengelig for andre. Rutinene kan enkelt bli tipasses og anvendt til brukerspesifikke datasett da de er blitt utviklet ved i og ved hjelp av åpen kildekode programvare.

Preface

The presented work has been done in the TEM Gemini Centre at Department of Physics at the Norwegian University of Science and Technology (NTNU) during the spring 2016. The Master's project has been supervised Professor Dr. A.T. J. van Helvoort. who together with PhD student Julie Stene Nilsen has collected the data used for the analysis. The Nanowire group at the Department of Electronics and Telecommunications at NTNU (Prof. Helge Weman and Prof. Bjørn-Ove Fimland) has provided the material used in this study. This Master's thesis is a continuation of the work in TFY4520 (Nanotechnology, Specialization Project) done in the same group in autumn 2015 and was supervised by Prof. A.T.J. van Helvoort and co-supervised with Prof. II J.C. Walmsley at SINTEF. My focus in this study has been on the data analysis, but I have been involved in the planning and attended some of the experimental sessions at the JEOL-ARM200F STEM at the TEM Gemini Centre.

I hereby declare that this Master's thesis is my own independent work and is in accordance with the regulations for the Master's degree programme in Nanotechnology at NTNU.

Andreas Garmannslund

Acknowledgements

Several people have been involved during the work with my Master's thesis and they all deserve recognition for their help and support.

First of all, I would like to thank my supervisor, Prof. A. T. J. van Helvoort. Thank you for all your encouragement during the last year. In times when work has been challenging, your enthusiasm and support has always managed to bring back my motivation. You have done so much more than what is expected from a supervisor for which I am truly grateful.

I would also like to thank PhD student Julie Stene Nilsen who have spent countless hours doing specimen preparation and collecting data for me.

I would also like to thank all the HyperSpy developers, especially Pierre Burdet who already had done some work on implementing the ζ-factor method (without absorption correction), Katherine E. MacArthur who integrated Burdet's code into HyperSpy, Duncan Johnstone, who helped reviewing the code and Fransisco de la Peña for creating HyperSpy and coming with useful input in the discussion. I would also like to thank Vidar Tonaas Fauske and Magnus Nord for helping me getting into HyperSpy at the start of last year.

Finally, I would like to thank my parents for their endless love and support since the day I was born.

List of abbreviations

ADF	annular dark field
AEM	analytical electron microscopy
BF	bright field
CL	Cliff-Lorimer (ratio technique)
DF	dark field
DP	diffraction pattern
EELS	electron energy-loss spectroscopy
EDS/EDX/XEDS	energy-dispersive x-ray spectroscopy
EFTEM	energy-filtered transmission electron microscopy
FEG	field emission gun
FIB	focused ion beam
GIF	Gatan imaging filter
HAADF	high-angle annular dark field
ICA	independent component analysis
NMF	non-negative matrix factorization
PCA	principal component analysis
SEM	scanning electron microscopy
SDD	silicon-drift detector
SI	spectrum imaging
STEM	scanning transmission electron microscopy
SVD	singular value decomposition
TEM	transmission electron microscope
ZB	zinc blende
WZ	wurtzite

Contents

1 Introduction **1**

2 Theory **4**

2.1 Scanning Transmission electron microscopy 4
2.2 The physics behind X-ray emission 5
2.3 Energy-dispersive X-ray spectroscopy 8
2.3.1 Measuring X-rays and the EDS detector 8
2.3.2 EDS spectrum 9
2.4 Quantitative X-ray analysis 11
2.5 Absorption correction 13
2.6 Shadowing 15
2.7 Multivariate statistical analysis 18
2.8 Fitting of the ζ-factors to theoretical values 19
2.8.1 Chi-square fitting 19
2.8.2 Model from the theoretical ζ-factor 20
2.9 Thickness determination 21
2.9.1 Nanowire thickness from projected width 21
2.9.2 Energy-filtered transmission electron microscopy 22

3 Experimental **24**

3.1 Microscope 24
3.2 Material 24
3.2.1 Nanowires 24
3.2.2 GaAs/GaAsSb nanowires (SC48) 26
3.2.3 Other materials 26
3.3 Data processing 26

4 Results **27**

4.1 Measured ζ-factors 27
4.1.1 ζ-factors for gallium and arsenide from nanowires 27
4.1.2 ζ-factors for antimony 29

4.1.3 ζ-factors for aluminium and carbon 29
4.2 Estimated ζ-factors from least square fitting 30
4.2.1 ζ-factors for other elements 30
4.2.2 Shadowing model 32
4.3 Quantification 34
4.3.1 GaAs/AlGaAs core-shell nanowire (SC365) 34
4.4 GaAs/GaAsSb nanowire (SC48) 39
4.5 Thickness measurements 39

5 Discussion **40**
5.1 Relative error of ζ-factors 40
5.2 ζ-factors from GaAs nanowires 41
5.3 Shadowing model 42
5.3.1 Robustness of the fitting 42
5.3.2 Analysis of the fitted models 43
5.3.3 Why we use ζ-factors for the active detector area ratio 43
5.4 Quantitative analysis 44
5.5 Epilogue: Contribution to HyperSpy 45

6 Conclusion **47**

A SCANDEM 2016 **54**

B Source code **58**

Chapter 1

Introduction

Structural and compositional information can be acquired down to atomic scale in (scanning) transmission electron microscopy ((S)TEM). This information is essential for understanding material properties so that they can be improved for technological applications. A wide variety of different imaging and analysis modes are available including bright-field imaging (BF) for morphology analysis, high-angle annular dark-field imaging (HAADF) depicting contrast due to difference in average atomic number and thickness variation analysis, diffraction pattern (DP) for crystal structure analysis and compositional analysis using electron energy loss spectroscopy (EELS) or X-ray energy dispersive spectroscopy (EDS). The development of new detectors and improved imaging techniques significantly increases the amount of data that need to be processed. Combining the analytical spectroscopy with imaging, so called spectrum imaging (SI), further increases this amount as a spectrum is collected at each pixel of the image as the probe is raster scanned over the region of interest. In principle, it is possible to create even larger and more complex data sets by collecting data from several different spectrometers simultaneously. It is essential to develop computational processing routines to deal with these big data sets in order to fully utilise the potential of state of the art electron microscopes. Reproducible results are of uttermost importance for proper scientific research and progress. Post-acquisition data processing by computational routines would enable the analysis to be re-run at any time, always yielding the same output. Ideally, free open source software packages should be used to develop the routines as to facilitate other researchers to reproduce the outcome and apply the routines to their own data sets. In quantitative microanalysis the chosen processing routines such as peak-fitting and background subtraction for EDS, might have a big impact on the final result and it should therefore be transparent which routines were used. Open source software provides full insight into the internal workings of the implemented routines. This is a huge advantage compared to commercial black box software, as the user can check that the routines are correctly implemented in case any unexpected results are obtained during analysis. Most free open source software projects also encourages people to contribute and collaborate to improve the software. This is the idea behind HyperSpy[1], a Python library for multidimensional data analysis. HyperSpy was created specifically due to the increasing need for more flexible and advanced processing tools for electron microscopy analysis. HyperSpy contains many features that are not available in commercial packages and give the user much more control over the data processing. Despite its young age, it has been used in many research articles, including renowned journals such as *Nature*[2] and *NanoLetters*[3, 4].

The Cliff-Lorimer ratio technique[5], relates the ratio of intensity with the ratio of composition through an empirical factor and it has become the dominant quantification routine for EDS in (S)TEM. It was designed to use ratios due to instrumental limitations in older microscopes. The ζ-factor method[6] is a more recent approach for quantitative EDS in (S)TEM. ζ is a proportionality factor that relates the composition of an element to experimental measurements. In the ζ-factor method, mass-thickness is determined simultaneously with composition, which allows to correct for absorption effects from EDS data alone. Pure element thin-films can be used for the calibration of the ζ-factors for a given setup. This is a major advantage over the Cliff-Lorimer ratio technique where multicomponent standards with similar composition to the unknown sample is needed for an accurate calibration of the k-factors. Unfortunately, most (S)TEM operators uses theoretically calculated k-factors due to the difficulties of finding good standards and because the Cliff-Lorimer method is the only routine widely available through commercial (S)TEM EDS software packages.

The main motivation and aim behind this project has been to improve the quantitative EDS analysis within the TEM Gemini Centre at NTNU, by introducing the ζ-factor method for improve quantitative EDS analysis and to develop related routines and procedures that can easily be accessed and adapted by other users of the facility. By doing this through the use of open source software within the fastest growing platform for TEM analysis, the analysis is transparent and verifiable. The work has given an opportunity to contribute to the scientific community through open source software development. In relation to this project, this has specifically involved aiding and reviewing the implementation of the basic (non-absorption corrected) version of the ζ-factor in HyperSpy. A routine that incorporate the thin-film absorption correction in the quantification has been developed in this project and is structured with the goal of incorporating it into the HyperSpy framework during the summer 2016 in time for the next major release (v.1.0.0).

GaAs nanowires has been used as the model material for the development of the ζ-factor method in Python. The direct band gap of III-V semiconductor nanowires can be tailored by alloying with for example Al (increases the band gap) or Sb (decreases the band gap). This makes them promising candidates as building blocks in future optoelectronic devices such as lasers, solar cells and light-emitting diodes[7, 8]. In this study, alloying of GaAs nanowires was done by introducing lighter Al into Ga lattice sites and heavier Sb into As lattice sites. Active research on the optimization of these nanowires for applications in devices is ongoing at the Nanowire Group at NTNU[9–11]. A more robust and accurate compositional analysis than the currently used Cliff-Lorimer ratio technique is required for further development of these heterogeneous semiconductor nanowires which makes them an ideal model case for this study. The materials are pure and the specimen thickness, which is important for accurate calibration of ζ-factors, can be deduced from the projected width. In addition to looking at whole nanowires, focused ion beam (FIB) made cross sections were also investigated, as FIB has become an important and common preparation method. As part of the process of testing and improving the accuracy of the quantification, specimen tilt dependency effects have been systematically studied. In comparison, the Cliff-Lorimer ratio technique does not normally take s effects into account, despite it being crucial for an accurate outcome and in particular for III-V materials[12].

The thesis is structured as followed. Chapter 2 gives a short theoretical background for the physics behind X-ray emission and EDS analysis in (S)TEM. Chapter 3 explain the methods used and other experimental details. Results are presented in chapter 4, followed by discussion in Chapter 5. Conclusions of the work is given in chapter 6. The appendices contains the source code developed during this project, in addition to an abstract and poster that were submitted and presented at the SCANDEM conference in June 2016.

Chapter 2

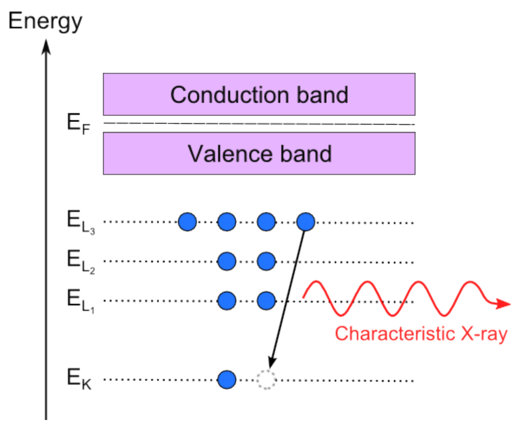
Theory

2.1 Scanning Transmission electron microscopy

Scanning Transmission electron microscopy (STEM) allows for imaging of thin samples (i.e. electron transparent, normally < 100 nm) with subnanometre resolution. An electron emission gun situated at the top of the instrument column creates electrons. The beam diameter and the energy spread are both characteristics that depends of the type of electron gun. In thermionic emission guns, a tungsten or single-crystal lanthanum hexaboride filament is heated giving the electrons enough energy to overcome the work function and thereby escaping the surface. A high negative voltage is applied to the filament and an electrode, known as a Wehnelt cup, situated between the filament and the anode plates. In analytical electron microscopy field emission guns (FEG) are usually preferred as they generate higher brightness (current density per solid angle) and coherence (smaller beam spread). In a FEG, a strong electric field causes tunnelling from a tungsten filament the surrounding vacuum. Cold FEGs, which are operated at room temperature, are preferred if you want high-energy resolution because the energy spread is small (0.3-0.5 eV) [13]. However, the emission current decreases slowly over time in cold FEGs due to contamination from residual gas being adsorbed on the emitter. If emission current stability is a concern a thermal FEG can be used. The emitter is then heated, decreasing the potential barrier in a process known as the Schottky effect. However, thermal FEGs have a larger energy spread (0.6-0.8 eV). The accelerated voltage in STEMs are usually in the range of 100-300 kV, which corresponds to a velocity of 50-80 % of the speed of light, meaning relativistic effects must be taken into account when calculating the electron wavelength. The electrons passes through a set of electromagnetic lenses. By adjusting the current applied to the lenses, we can adjust the applied magnetic field and thereby controlling the electron trajectory. A set of condenser lenses focus a small convergent beam onto the specimen. The beam is raster scanned across the specimen. This technique is called spectrum imaging (SI) where the illuminated area in each step of the raster corresponds to a pixel in the recorded image. The specimen is placed in a holder which can be tilted around two axes (usually) and translated in the x-, y- and z-direction. Some STEMs also allows the sample to be rotated around the optical axis. The various interactions between the incoming electrons and the specimen make it possible to investigate a wide range of material properties. Bright field (BF) images are constructed from electrons that are transmitted through the specimen, but have not significantly changed direction relative to the incoming beam (scattering angle less than 10 mrad). An aperture is inserted in order to isolate the electrons that make up the BF image. The electrons continue through the objective lens (surrounding the specimen stage) and another set of imaging lenses before they hit a fluorescent screen which forms the image. The image can also be digitally recorded by for example using a charge-coupled devices (CCDs). Detectors for large scattering angles such as annular dark-field (ADF) and high-angle annular dark-field (HAADF) are commonly used detectors in STEM. Almost all electrons recorded by HAADF (scattering angle greater than 50 mrad [14]) are inelastically scattered electrons. This gives images, often called Z-contrast images, with an intensity that are directly related to the local mass-thickness of the sample. Inelastic scattered electrons are also used in electron energy-loss spectroscopy (EELS) for element mapping and thickness measurements. The incoming electrons might also lead to emission of X-rays that are used for compositional microanalysis in energy-dispersive X-ray spectroscopy (EDS).

2.2 The physics behind X-ray emission

X-rays are generated from electrostatic inelastic interactions between the incoming beam electron and an inner-shell electron of an atom in the specimen. If the energy transfer from the incoming electron is higher than the critical ionization energy of the inner-shell electron, the electron escapes the atom. The excited atom wants to return to a lower energy state and does so by filling the vacancy with an electron from another shell. The energy released by this de-excitation results in either the emission of an X-ray or an Auger electron as shown in Figure 2.1. The atom returns to its ground state through a series of similar relaxation steps until finally an electron from the conduction band fills the last hole. Only the transitions between the inner-shell electrons releases enough energy to emit an X-ray, while the other transitions might emit photons with lower energy or generate phonons. The X-ray has an energy equal to the energy difference between the ionized and lower energy state of the atom. Because the electron configuration is different for each element, the emitted X-ray can be used for characterization. The characteristic X-ray (or X-ray line) is named according to which energy shells are involved in the transition using Siegbahn notation. The X-ray lines are further grouped into families (K, L, M) according to which shell the initial hole was created. The $K_\alpha$line results from the transition from the L to the K-shell, while a L_{β_1} is caused



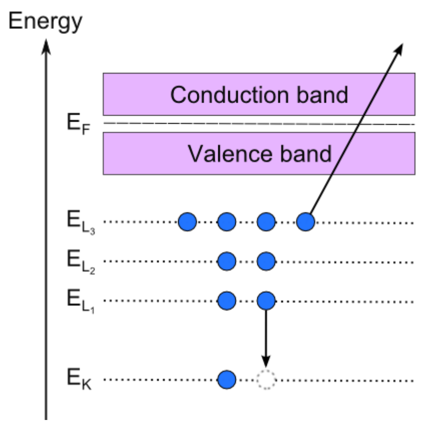


Figure 2.1: The excited atom returns to a lower energy state by filling the hole in the inner-shell. The energy released can cause the emission of (a) an characteristic X-ray or (b) an Auger electron.

by a transition between M_4 and L_2 subshells. For characterization in analytical electron microscopy (AEM) usually only the α-line in each family is used as they have the highest relative intensity (weight).
The relationship between atomic number and the wavelength of the X-ray is given by Moseley's law[15]:

$$\lambda = \frac{B}{(Z - C)^2} \tag{2.1}$$

where Z is the atomic number, B and C constant related to the specific X-ray line, and λ is the wavelength of the X-ray.

Instead of an X-ray, an Auger electron, with a characteristic energy related to the binding energy of the outer-shell electrons, might be emitted when the atom return to a lower energy state. The electron escapes from one of the outer-shells and have an energy similar to the characteristic X-ray. However, most Auger electrons are absorbed in the specimen and it is therefore considered a surface technique and not used in TEM. The fluorescence yield is a measurement of how many X-rays are generated compared to Auger electrons. It increases with atomic number as shown in Figure 2.2. This term must not be confused with fluorescence that refers to X-rays caused by other sources than electron irradiation.

Another type of X-rays is bremsstrahlung, which is generated due to electrostatic interactions as the incoming electron is deflected by the atomic nucleus. An X-ray is generated due to the loss of momentum of the incoming electron. In X-ray spectroscopy, bremsstrahlung give rise to a continuous spectrum upon which the peaks of the char-

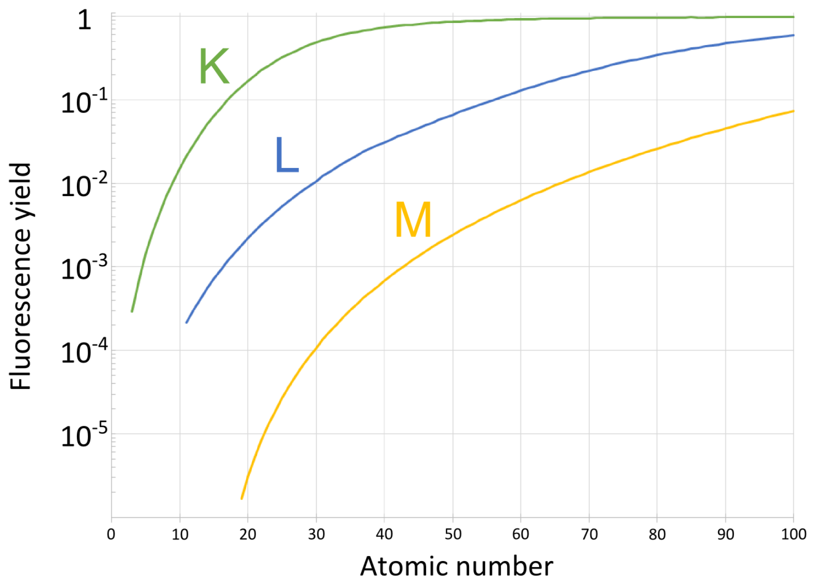


Figure 2.2: The fluorescence yield for the K-shell and the average fluorescence yield for the L- and M-subshells increases for higher atomic numbers. The figure is made from fluorescence yields given in Table 8 in [16, 17]

.

acteristic X-ray lines are superimposed. Bremsstrahlung is mainly forward scattered, compared to characteristic X-rays which are emitted in all directions. In material characterization bremsstrahlung is regarded as noise and must be taken into account during analysis. The intensity of the bremsstrahlung decreases with increasing energy. In experimental results, it is normally not detected for energies $<$ 2 keV as low-energy X-rays are absorbed in specimen or by the EDS detector.

The ionization cross section describes the likelihood for an incoming electron to ionize an atom. There is some disagreement in literature regarding the best value for the ionization cross section, particularly in the range used for analytical electron microscopes (100-400 kV). Most models for the ionization cross sections are based on the model derived by Bethe:

$$\sigma_{nl} = \frac{6.51 \cdot 10^{-24} Z_n b_{nl}}{E_0 E_{nl}} \ln \left(c_{nl} \frac{E_0}{E_{nl}} \right) \tag{2.2}$$

where E_0 is the energy of the incident electron beam in keV, $E_n l$ is the ionization energy of electrons in the shell with principal quantum number n and orbital quantum number l, Z_n is the number of electrons in the n-shell, i.e. 2 for K ($n = 1$), 8 for L ($n = 2$) and 18 for M ($n = 3$). $b_n l$ and $c_n l$ are called the "Bethe parameters" and most studies are trying to find the best values for these variables. Powell[18] did an extensive review of the ionization cross section in 1976 and later, in 1981, Schreiber and Wims[19] introduced another fitting parameter d_n. In practice the fitting parameters are only dependent on the principal quantum number giving the modified expression:

$$\sigma_n = \frac{6.51 \cdot 10^{-24} Z_n b_n}{E_n^2 U^d} \ln(c_n U) \tag{2.3}$$

where the quantity $U = \frac{E_0}{E_{nl}}$ is known as the overvoltage. Both (2.2) and (2.3) has the units m^2.

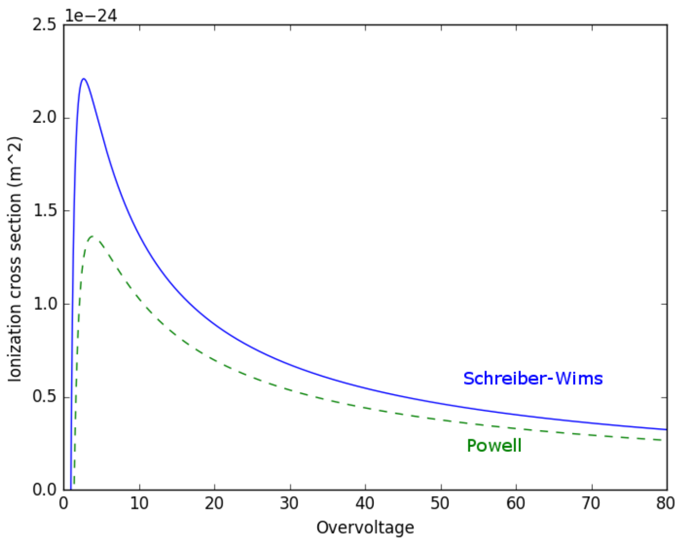


Figure 2.3: The inner-shell ionziation cross section for $\mathrm{Al}_{\mathrm{K}_\alpha}$ as a function of overvoltage. The blue curve uses values for b_k, c_k and d_k provided by Schreiber and Wims[19] and the green line uses parameters recommended by Powell[18].

2.3 Energy-dispersive X-ray spectroscopy

2.3.1 Measuring X-rays and the EDS detector

Maximizing the number of X-rays reaching the detector is essential for accurate energy-dispersive spectroscopy. The small interaction volume in TEM, $10^{-5} - 10^{-8}$ μm^3 compared to $\approx 1\mu$m^3 in scanning electron microscopes, greatly limits the number of generated X-rays[20]. The placement constraints of the EDS detector is another limiting factor in TEMs. The EDS detector is situated above the specimen (to minimize the contribution of bremsstrahlung to the recorded spectrum), between the pole pieces of the objective lens (Figure 2.4). The solid angle from the analysis point to the active area of the detector defines the collection angle (Ω). The detector should be placed such that the collection angle is maximized to limit absorption of created X-rays within the specimen, but its position is limited by the upper pole piece of the objective lens. Due to this restriction only about $0.01 - 1.2\%$ of the emitted X-rays are collected by the detector. The take-off angle is the angle between the specimen surface and the normal to the detector surface. This must not be confused with the elevation angle, which is the take-off angle when the specimen stage is perpendicular to the incoming beam (i.e. at $0°$ tilt). In order to minimize absorption effects the take-off angle should be maximized, but if this is done by altering the elevation angle, this will again limit the collection angle and therefore count rate. The elevation angle is usually restricted to about $20°$. However, in TEM the absorption effect can usually be neglected with regard to take-off angle[21] so a higher collection angle is preferred. The usual way of improving counts in a given setup is to acquire data over a longer period. However, this is not trivial in TEM as the high acceleration voltage combined with the small probe size leads to a very high electron dose which might damage the specimen. The acquisition time is also limited by specimen drift. On top of these poor counting statistics, spurious X-rays (X-rays generated in the specimen, but not from the where the probe is situated) and system X-rays (created by the TEM column or other parts of the system) contributes to the X-ray spectra. EDS detectors are usually silicon semiconductors, so when the X-rays are hitting the sample, a number of electron-hole-pairs proportional to the X-ray energy are generated. They create a voltage signal, which is amplified by a field-effect transistor and then digitized and stored into an appropriate energy channel. Modern TEMs uses Silicon-Drift Detectors (SDD), but Si(Li) detectors are still common on older microscopes. One of the advantages with SDD over Si(Li) is that they offer much better counting statistics as it can be placed closer to the specimen and only need Peltier cooling compared to Si(Li) which need to be continuously cooled by liquid N_2.

2.3.2 EDS spectrum

The best way to perform energy-dispersive spectrometry is to use spectrum imaging (SI) in STEM. For each pixel of the image a spectrum of X-ray intensities (counts) vs energy is gathered. The data can be stored in a computer file and processed later. An example of a spectrum is shown in Figure 2.5. The characteristic X-ray lines appears as Gaussian peaks. The poor energy resolution in EDS detectors (typically 125-135 eV) limits the line-width of the peaks, which causes overlapping peaks in the spectrum. This is especially prominent in the low energy regime (i.e. for light-elements and L- and M-edges). Many different artifacts are introduced to the spectrum due to scattering of X-rays after initial creation, signal-processing or characteristics of the detector. Some of these are automatically corrected by the acquisition software, while other needs to be treated manually. The most common artifacts are summarized in table 2.1.

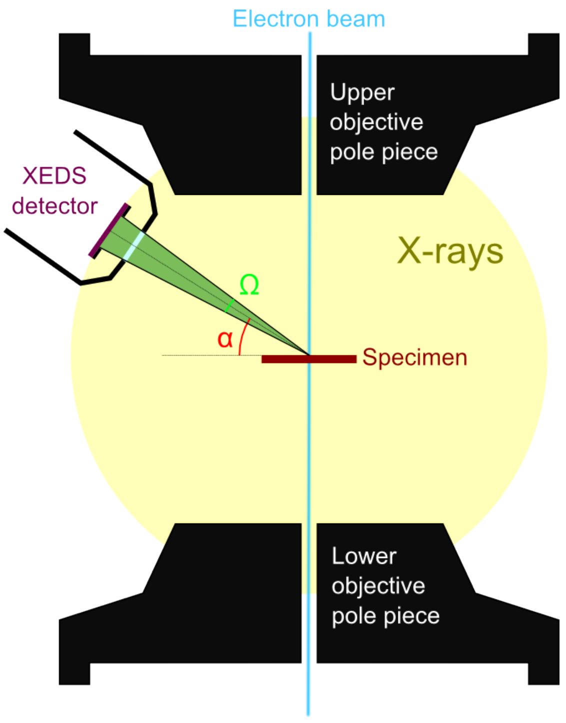


Figure 2.4: The EDS detector interface. Although characteristic X-rays are emitted in all directions (sphere wave), only a small fraction of this, defined by the collection angle Ω (solid angle) actually hits the detector. The take-off angle, α, is also indicated in the figure.

Table 2.1: Common artifacts in EDS spectra

Sum peak	Larger number of incoming X-rays may lead to detection of two X-rays at the same time. Instead of recording the photons as two X-rays, the system recognizes this as one X-ray with an energy equal to the sum of the two X-rays.
Escape peak	Incoming X-ray might generate a Si K_α X-ray in the dead layer of the detector. The recorded X-ray will then have an energy equal to $(E - E_{\mathrm{SiK}_\alpha}) = (E - 1.74\mathrm{keV})$.
Internal fluorescence peak	Si K_αX-rays generated in the dead layer of the detector might be present in the spectrum. In SDDs this effect is small due to the thin dead layer.
Coherent Bremsstrahlung	Gaussian-shaped peaks introduced due to similar coloumb interactions between the electron beem and the periodically located atom nuclei.
System X-rays	X-rays generated in the instrument which don't originate from the specimen. Notable examples are Cu and Fe.
Spurious X-rays	X-rays generated from the specimen, but not from the analysis volume. Created due to post-specimen scattering or fluorescence.

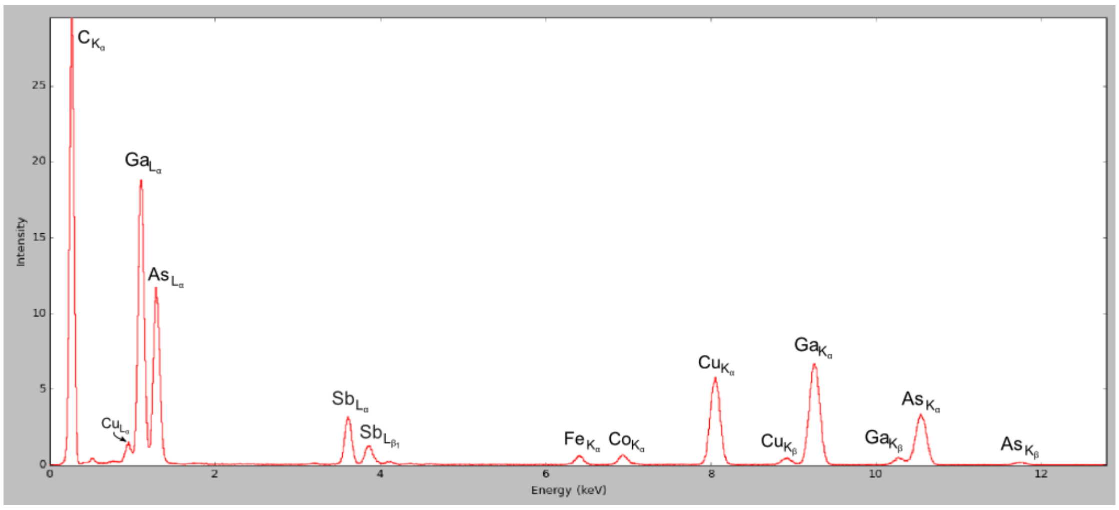


Figure 2.5: EDS spectrum showing the intensity (number of counts) for different X-ray energies. Note that the α lines have a significant number of counts compared to the β lines. The Fe$_{K_\alpha}$ peak is an example of a system X-ray artifact as it comes from a pole piece or an aperture rather than the specimen itself.

2.4 Quantitative X-ray analysis

The Cliff-Lorimer ratio technique[5] relates the composition in weight percent, C_A and C_B of two elements A and B in a sample with their measured X-ray intensity above background, I_A and I_B, through a sensitivity factor k_{AB}as followed

$$\frac{C_A}{C_B} = k_{AB}\frac{I_A}{I_B} \tag{2.4}$$

The composition w_A and w_B can easily be determined in a binary system combining $C_A + C_B = 100\%$ with eq. (2.4). For higher order systems, this can easily be expanded by adding extra eq. in the same form as (2.4) combined with $\sum_j C_j = 100\%$. The Cliff-Lorimer factor (k-factor), k_{AB}, can be calculated theoretically or determined experimentally. However, theoretical values are shown to give an error of 15-20%[14], and experimental measurements should be conducted for an accurate analysis. Unfortunately, the experimental procedure is extremely time consuming and tedious. Even getting below 5% relative error takes a lot of work and if anything changes in the instrumental setup, the procedure must be redone if you want accurate results. It can also be difficult to find suitable multi-component standards for thin-film analysis.

The k-factor can be calculated using eq.(2.5) where Q is the ionization cross section, ω is the fluorescence yield, a is the line weight (relative transition probability), ϵ is the detector efficiency all dependent on the X-ray line while A is the relative atomic mass of the element that emitted the X-ray line.

$$k_{AB} = \frac{(Q\omega a)_A \, A_B \epsilon_A}{(Q\omega a)_B \, A_A \epsilon_B} \tag{2.5}$$

The large error in the k-factor calculation is mainly due to the lack of consensus for a best value for the ionization cross section and uncertainties introduced by detector configurations.

The Cliff-Lorimer method was introduced in 1975[5]. Since then, many of the instrument limitations has been overcome or severly improved in modern analytical electron microscopes. The ζ-factor method is a more recent method for quantitative microanalysis in TEM. It was introduced by Watanabe et al. in 1996[22] and later modified in 2006[6]. The ζ-factor method is based on the principle that the intensity of an X-ray line from an element A in a thin-film (i.e. ignoring absorption or fluorescence effects) is proportional to the mass-thickness and composition and can be expressed as:

$$I_A = \frac{N_v Q_A \omega_A a_a}{M_A} C_A \rho t (\Omega / 4\pi) \epsilon_A D_e \tag{2.6}$$

where Q, ω, a, ϵ_A are the ionization cross section, fluorescence yield and line weight of the X-ray line, N_v is Avogadro's number, M_A is the relative atomic mass of element A, C_A is the composition of element A in weight percent, ρ is the density of the sample, $[\Omega/(4\pi)]$ is the fraction of characteristic X-rays that reach the detector (as Ω is the collection angle). D_e is the total electron dose and can be further expressed as

$$D_e = \frac{I_p \tau}{e} \tag{2.7}$$

where I_p is the in-situ beam current, τ is the acquisition time and $e = 1.6022 \cdot 10^{-19}$ is the elementary charge. The ζ-factor is defined as

$$\zeta_A = \frac{M_A}{N_v Q_A \omega_A a_A [\Omega/(4\pi)] \epsilon_A} \tag{2.8}$$

By inserting (2.8) in (2.6) we obtain

$$\rho t = \zeta_{\mathrm{A}} \frac{I_{\mathrm{A}}}{C_{\mathrm{A}} D_e} \tag{2.9}$$

where ρ is density, t thickness, I_{A} the intensity of the peak above background, C_{A} the composition.
By combining eq.(2.9) and $\sum_j C_j = 100\%$ for all the elements in the analysed region, composition of all the elements can be calculated simultaneously as mass-thickness

using (2.10)

$$\rho t = \sum_{j}^{N} \frac{\zeta_j I_j}{D_e}, \quad C_\mathrm{A} = \frac{\zeta_\mathrm{A} I_\mathrm{A}}{\sum_j^N \zeta_j I_j}, \quad \ldots, \quad C_\mathrm{N} = \frac{\zeta_\mathrm{N} I_\mathrm{N}}{\sum_j^N \zeta_j I_j} \tag{2.10}$$

Note that ζ-factor is, as the k-factor, dependent on the detector efficiency and the acceleration voltage. However, it is much easier to determine the ζ-factors as it can be estimated from pure element films or multi-component specimens with known composition.

2.5 Absorption correction

When characterizing thin-films or similar nanomaterials (e.g. nanowires) absorption effects are usually ignored as they are considered small[14]. This is not the case for all specimens as certain elements, particularly light-elements with X-ray lines of an energy of less than 2 keV, e.g. oxygen, can be highly absorptive. Effectively, in the EDS spectrum, the measured intensity of the absorbing peak will be higher and the intensity of the X-ray peak that is absorbed will be lower. In order to account for this, it is necessary to adjust the measured intensity by introducing an absorption correction term for an X-ray line x such that $I_x = I_x^0 * A_x$ where I_x^0 is the measured intensity and A_x is the absorption correction factor given in (2.12). The amount of absorption is highly dependent of the absorption path length, i.e. the length the emitted X-ray travels through the specimen. As illustrated in Figure 2.6, the absorption path length, t_abs can be expressed as

$$t_\mathrm{abs} = t \csc(\theta_\mathrm{E} + \theta_\mathrm{T}) = t \csc(\alpha) \tag{2.11}$$

where α is the take-off angle which is given by the sum of the elevation angle (θ_E), i.e. the angle between the detector axis and the specimen stage at 0° tilt, and the specimen stage tilt (θ_T). Under the assumption that the X-rays are generated uniformly through the depth of the thin-film, the absorption correction factor, A_x becomes:

$$A_x = \frac{(\mu/\rho)^x_\mathrm{spec} \rho t \csc(\alpha)}{1 - \exp\left[-(\mu/\rho)^x_\mathrm{spec} \rho t \csc(\alpha)\right]} \tag{2.12}$$

$(\mu/\rho)^x_\mathrm{spec}$ is the photoelectric mass absorption coefficient for the X-ray line x in the specimen. It can be calculated as weighted sum of the mass-absorption coefficients for element x for each element in the specimen:

$$(\mu/\rho)^x_\mathrm{spec} = \sum_j \left(w_j (\mu/\rho)^x_j\right) \tag{2.13}$$

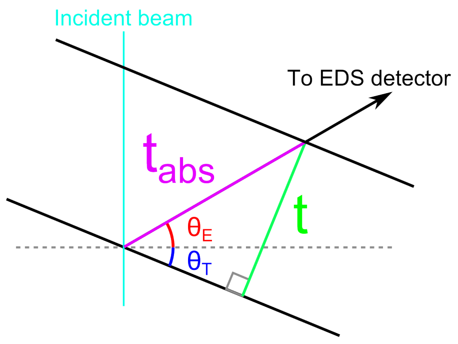


Figure 2.6: The absorption length, t_{abs} given by the specimen tilt, θ_{T},
take-off angle, α, and thickness, t of the thin-film.

where w_j is the weight percent of element j in the specimen and $(\mu/\rho)_j^x$ are the mass absorption coefficient for the X-ray line x for element j.

The other terms in (2.12) are the density of the specimen, ρ, and the absorption path length as defined in (2.11). The mass absorption coefficients for GaAs are shown in Figure 2.7. The sharp tops in the graph are called absorption edges. The absorption edges are located at an energy slightly above the ionization energy of the Ga and As K- and L-lines. Emitted X-rays with an energy close to the top and at a higher energy (to the right side) than the absorption edge will be absorbed by the specimen and ionize the element with an X-ray line with energy just below the edge energy. This causes a new X-ray from the absorber element when this ionized atom relaxes. However, if the original X-ray have an energy just below an absorption edge, no absorption will occur as the energy does not overcome the critical ionization energy of the atom. The vertical scale in Figure 2.7 is logarithmic and it is obvious from the figure that absorption can have a large impact of the measured intensity of light elements ($<$ 2keV).

The absorption correction term (2.12) can be incorporated in the ζ-factor method by substituting I_x with $I_x A_x$:

$$\rho t = \sum_j^N \frac{\zeta_j I_j A_j}{D_e}, \quad C_{\mathrm{A}} = \frac{\zeta_{\mathrm{A}} I_{\mathrm{A}} A_{\mathrm{A}}}{\sum_j^N \zeta_j I_j A_j}, \quad \ldots, \quad C_{\mathrm{N}} = \frac{\zeta_{\mathrm{N}} I_{\mathrm{N}} A_{\mathrm{N}}}{\sum_j^N \zeta_j I_j A_j} \tag{2.14}$$

The absorption correction terms are dependent on mass-thickness and the mass absorption coefficients. The absorption correction therefore an iterative process as both these variables are functions of the composition. For the first iteration all absorption correction factors is set to 1, effectively reducing (2.14) to (2.10), then absorption correction terms are calculated as given by (2.12) and the process is repeated until the solution converges.

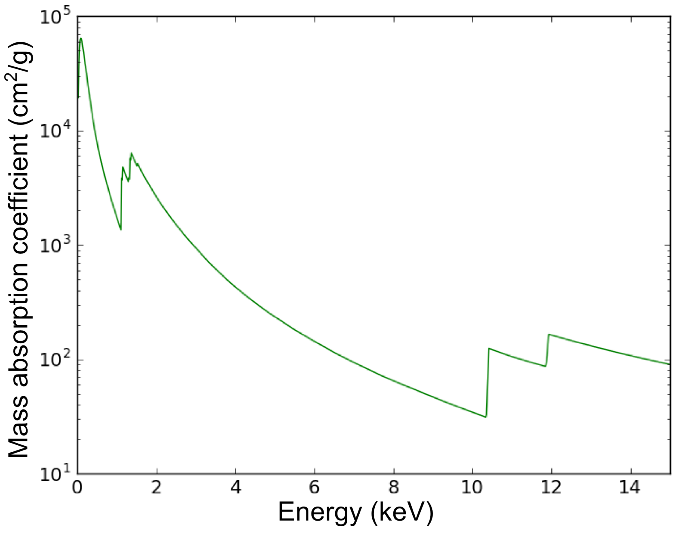


Figure 2.7: The mass absorption coefficients for GaAs as a function of energy. The absorption edges are situated right above the ionization energy for the K- and L-lines of Ga and As. The Figure was made from mass absorption coefficients from the database by Chantler et al.[23] which is included in HyperSpy.

Fluorescence

Fluorescence occurs when the absorbing element emit an X-ray. The composition of the absorbing element is usually much greater than that of the element being absorbed. Thus, although fluorescence has only a small impact on the intensity of the absorbing element, it could contribute greatly to the intensity of the absorbed element. Absorption is often caused by low-energy X-rays, which is often not used for quantitative analysis if higher energy K-lines are available in the spectrum. Fluorescence corrections are usually ignored except in the rare case where it is crucial for an accurate analysis.

2.6 Shadowing

Shadowing is the effect of the specimen holder, carrier and grid blocking the outgoing X-rays from reaching the detector. Under the assumption that all X-rays that goes through the specimen holder is entirely blocked (ignoring other effects such as energy dependent absorption and scattering), the reduction in the measured intensity should be the same for all X-ray lines (i.e. independent of the energy of the X-ray). The shadowing effect can therefore be expressed in terms of an effective collection angle, $\Omega(\theta)$ which is a function of specimen tilt. The ratio of the effective collection angle over the maximum collection angle, Ω^0, gives us the fraction of the X-rays that reach the detector, or in other words the active detector ratio, ΔA:

$$\Delta A(\theta) \equiv \frac{A_{\text{active}}}{A_{\text{full}}} \equiv \frac{\Omega(\theta)}{\Omega^0} \tag{2.15}$$

where Ω^0 is the maximum collection angle when there is no shadowing.

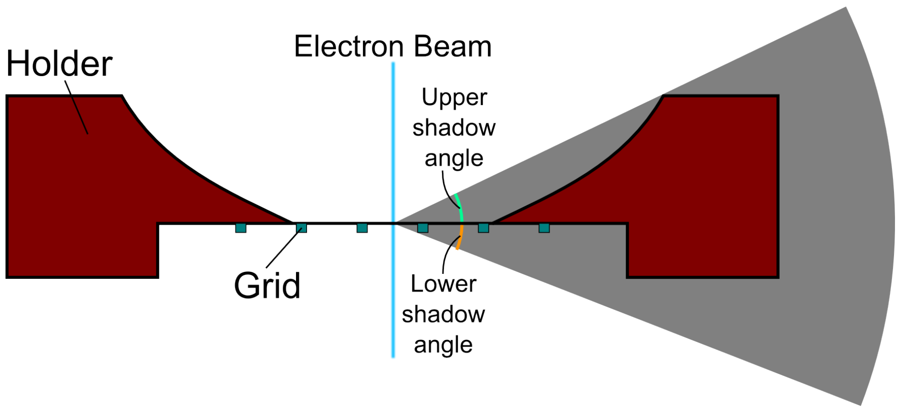


Figure 2.8: Shadowing is caused by the sample holder and grid blocking outgoing X-rays. The figure is adapted from Yeoh et al.[24]

ζ-factors can be used to determine ΔA by taking the ratio between a ζ-factor, ζ_A^0 taken at a high tilt angle and a ζ-factor measured at a chosen tilt angle, $\zeta_A(\theta)$ as demonstrated in (2.16). All element dependent terms are cancelled out, leaving only Δ which is independent of element as it only describes the fraction of the X-rays that was collected.

$$\frac{\zeta_A^0}{\zeta_A(\theta)} = \frac{\frac{M_A}{N_v Q_A \omega_a (\Omega/4\pi)\epsilon_A}}{\frac{M_A}{N_v Q_A \omega_a (\Omega(\theta)/4\pi)\epsilon_A}} = \frac{\Omega(\theta)}{\Omega^0} = \Delta\Omega(\theta) \equiv \Delta A \tag{2.16}$$

The active area will be different for a circular and a rectangular detector for the same shadow height as pictured in Figure 2.9. If the specimen holder casts a shadow of height h on the detector, the shadowed area for a circular detector with radius r is given by the area of a partially filled circle, given as

$$A_{\text{circle}} = r^2 \cos\left(\frac{r-h}{r}\right) - (r-h)\sqrt{2rh - h^2} \tag{2.17}$$

while for a rectangular detector it is simply

$$A_{\text{rectangle}} = (h_d - h) * w_d \tag{2.18}$$

where h_d and w_d is the height and width of the detector respectively.

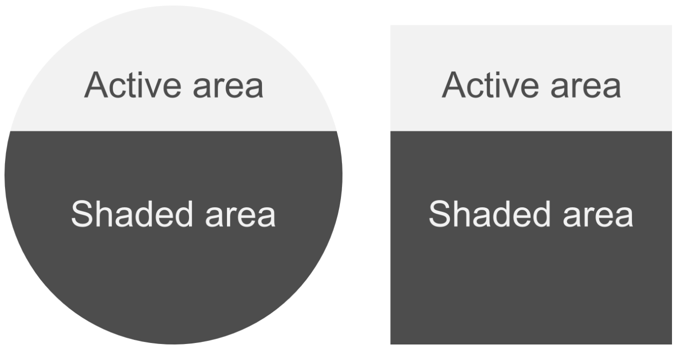


Figure 2.9: The two detectors have the same total area, but the shadowed area is different for the same shadow height.

Yeoh et al.[24] have suggested a simple geometrical model for the shadow height based on the instrumental setup:

$$h = \frac{d\ \sin(\theta_C)}{\cos(\theta_E - \theta_D)\sin(90 - \theta_D + \delta - \theta_C)} \tag{2.19}$$

where d is the distance from the specimen to the centre of the detector, θ_e is the elevation angle (take-off angle at 0°tilt), θ_d is the angle from the specimen to the lower part of the detector, δ is the angle between the detector and the optic axis, and θ_c is range of the shadow on the detector given by

$$\theta_c = (\theta_U - \theta_L) - \theta_T - \theta_D \tag{2.20}$$

where θ_U and θ_L are the upper and lower shadow angle (Figure 2.10)), θ_T is the tilt of the holder (defined as positive when the sample is tilted towards the detector). The lower detector angle is subtracted from the shadow range in order to define that shadowing starts at $\theta_C = 0$.
Once $\Delta A(\theta)$ is known, any ζ-factor taken at any angle, θ_1 can be converted to a ζ-factor for any another angle, θ_2:

$$\begin{aligned} \frac{\zeta_A(\theta_2)}{\zeta_A(\theta_1)} &= \frac{\frac{\zeta(\theta_2)}{\zeta^0}}{\frac{\zeta(\theta_1}{\zeta^0}} = \frac{\Delta A(\theta_1)}{\Delta A(\theta_2)} \\ \zeta(\theta_2) &= \zeta(\theta_1)\frac{\Delta A(\theta_1)}{\Delta A(\theta_2)} \end{aligned} \tag{2.21}$$

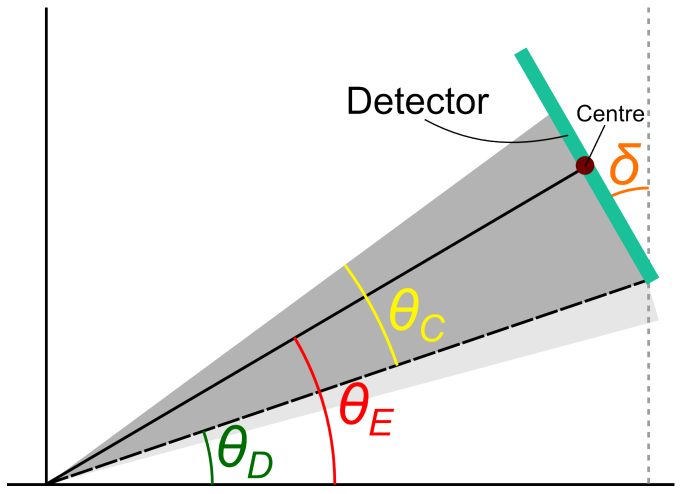


Figure 2.10: Schematic of detector interface showing the lower detector angle (θ_D), the elevation angle (θ_E) and the angular shadow range (θ_C).

2.7 Multivariate statistical analysis

In SI, a focused probe is raster scanned over a region of interest, collecting a full spectrum for each point in a two dimensional spatial array. The technique is commonly used for EDS and EELS in STEM as it allows for spatial compositional analysis. Even a small image contains a vast amount of data. To illustrate this, imagine a 64 x 64 image containing an EDS spectrum with 2048 energy channels. It contains more than eight million data points! Data processing is therefore essential to extract accurate chemical information. A range of multivariate statistical analysis (MSA) techniques exists which facilitates the treatment of large multidimensional data sets. Although these techniques are purely mathematical, they have been shown to be able to extracting elemental information from multidimensional EDS data sets[3].

Principal component analysis (PCA) is one of the most commonly applied MSA techniques and is often used as a first step for other techniques such as independent component analysis (ICA). The main objective with PCA[25] is to reduce the dimensionality of the data set. This is done by transforming the data set of interrelated variables into a new set of uncorrelated variables, more commonly referred to as principal components (PC). The PCs are sorted according to how much of the original data set they explain, i.e. their variance. Usually, only a few PCs are needed to represent most of the data set, while the others can for practical purposes be considered as noise as their contribution is insignificant. A graphical representation showing variance of the components is called a scree plot and is shown in Figure 2.11. PCA can therefore be used as an efficient technique for noise reduction in spectral imaging. Another commonly applied technique is non-negative matrix factorization (NMF) which places an extra constraint on the components so that the loadings (for EDS this correspond to the spectrum) are positive.

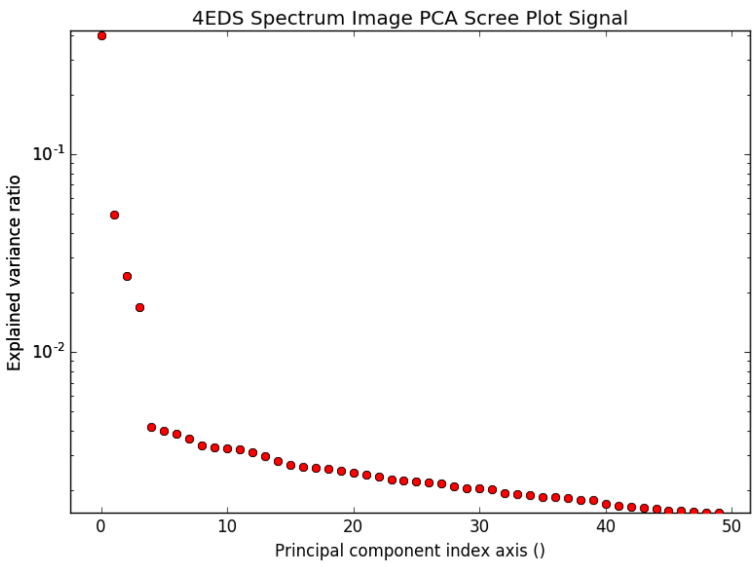


Figure 2.11: The variance of each principal component is given along the vertical axis. Four components will be sufficient to explain most of the data set according to this particular plot.

2.8 Fitting of the ζ-factors to theoretical values

2.8.1 Chi-square fitting

Imagine we have a set of k experimental data points $\boldsymbol{s} = ((x_0, y_0), (x_1, y_1), ..., (x_{k-1}, y_{k-1}))$ that we want to fit to a theoretical model given by $y(x, \boldsymbol{p})$ were $\boldsymbol{p} = (p_0, p_1, ..., p_{m-1})$ contains m parameters to be fitted. Given a fixed set of parameters and assuming that the measurement error in each point in $\boldsymbol{s}$ are independently random and normally distributed around the model $y(x, \hat{\boldsymbol{p}})$ with a standard deviation of σ_i, the probability of obtaining the experimental data, plus/minus a fixed value Δy for each data point[1] from the model is given by

$$\prod_{i=0}^{k-1} \exp\left[-\frac{1}{2}\left(\frac{y_i - y(x_i, \boldsymbol{p})}{\sigma_i}\right)^2\right]\Delta y \tag{2.22}$$

[1]This is a formality to avoid the probabilities to always be zero for continuous y_i values. See [26] for details.

The parameters can be fitted by maximizing this probability. This process is called maximum likelihood estimation. This is equal to minimize the negative of the logarithm of (2.22):

$$\left[\sum_{i=0}^{k-1}\frac{1}{2}\left(\frac{y_i - y(x_i, \boldsymbol{p})}{\sigma_i}\right)^2\right] - k\ln\Delta y \tag{2.23}$$

which again is equivalent to minimize the chi-squared (or weighted least-squares) as given in (2.24)

$$\chi^2 = \sum_{i=0}^{k-1}\left(\frac{y_i - y(x, \boldsymbol{p})}{\sigma_i}\right)^2 \tag{2.24}$$

2.8.2 Model from the theoretical ζ-factor

The theoretical expression for the ζ-factor is given in (2.8). Several of the parameters are uncertain and therefore have to be fitted for experimental conditions such as microscope configuration and detector. For a silicon drift detector with a thin polymer window, the detector efficiency term can be expressed as followed[14]:

$$\begin{aligned}\epsilon_{\mathrm{A}} = &\exp\left[(\mu/\rho)^{\mathrm{A}}_{\mathrm{pw}}\rho_{\mathrm{pw}}t_{\mathrm{pw}}\right]\exp\left[(\mu/\rho)^{\mathrm{A}}_{\mathrm{Au}}\rho_{\mathrm{Au}}t_{\mathrm{Au}}\right]\\ &\exp\left[(\mu/\rho)^{\mathrm{A}}_{\mathrm{Si}}\rho_{\mathrm{Si}}t_{\mathrm{Si}}\right]\left(1-\exp\left[(\mu/\rho)^{\mathrm{A}}_{\mathrm{Si}}\rho_{\mathrm{Si}}\hat{t_{\mathrm{Si}}}\right]\right)\end{aligned} \tag{2.25}$$

In (2.25), (μ/ρ), ρ and t corresponds mass absorption coefficient, density and thickness respectively. The three first factors corresponds to the attenuation of the signal due to the polymer window, the gold contact layer and the silicon dead layer. The last factor accounts for high-energy X-rays that are not absorbed in the active region of the detector and thus not collected. Information about the thickness of the different layers are not easily obtainable and must therefore be fitted numerically from the acquired ζ-factors. Due to the large variations in the available inner-shell ionization cross sections, a scaling factor for Q should also be fitted. From a set of m known ζ-factors from different X-ray lines within the same family, it is possible to fit $m-1$ parameters using least-square fitting by minimizing χ^2 from (2.24). It is important the output parameters make physical sense. The thickness parameters should therefore be restricted to positive values when fitting the ζ-factors. Constrains can be placed on the parameters by using a trust region reflective algorithm which is available through most numerical software packages (e.g. SciPy for Python).

2.9 Thickness determination

2.9.1 Nanowire thickness from projected width

For calibrating the ζ factors from (2.9) the thickness need to be known. Nanowires have a hexagonal cross section which allowed the thickness to be calculated based on the projected width. Hexagons have an angle of 120° between each edge. As seen in Figure 2.12, the hexagon can be divided into six equilateral triangles with sides $w/4$ were w is the projected width seen from the top of the hexagon. Using Pytagoras is it then possible to express the thickness (t) of the nanowire as a function of projected width:

$$\left(\frac{w}{4}\right)^2 + \left(\frac{t}{2}\right)^2 = \left(\frac{w}{2}\right)^2 \Rightarrow t = \frac{\sqrt{3}}{2}w \tag{2.26}$$

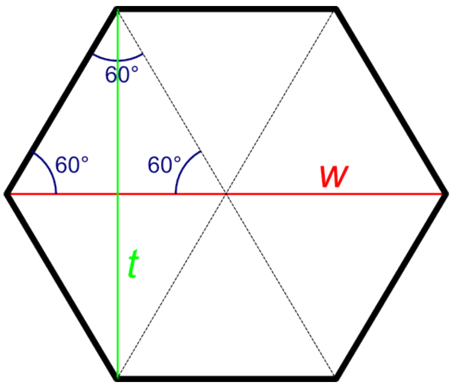


Figure 2.12: Nanowires have hexagonal cross sections which allows us to easily determine the thickness of the nanowire from the projected width. The schematic shows how it is possible to divide the hexagon into six equilateral triangles.

The thickness of a thin-film should be measured when its surface is perpendicular to the incoming beam. When the thin-film is tilted from this position, the thickness (path through which the beam traverses) will increase as shown in Figure 2.13. The adjusted thickness is given by

$$t = t_0/cos(\theta) \tag{2.27}$$

where t_0 is the thickness when the specimen is lying in the plane perpendicular to the beam and θ is the difference in tilt between the two orientations.

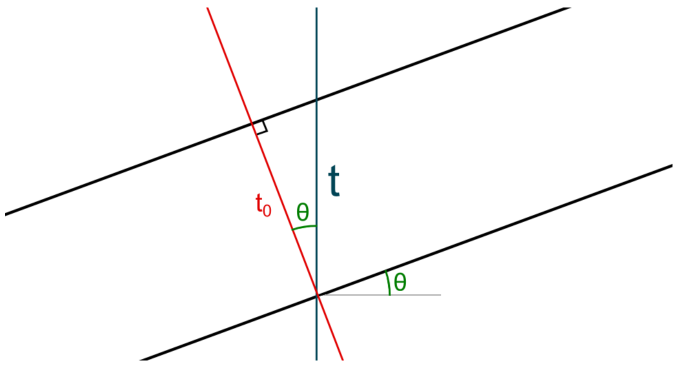


Figure 2.13: For a thin-film the thickness through which the electron beam traverses depends on tilt as $t = t_0/\cos(\theta)$.

2.9.2 Energy-filtered transmission electron microscopy

In electron energy loss spectroscopy (EELS) the transmitted electrons are separated according to the loss in kinetic energy due to different scattering effects when interacting with the atoms in the sample. A magnetic prism, situated below the TEM column, separates the electrons and they are projected onto a detector so that a spectrum of frequency vs energy loss is obtained. EELS can be use to study a lot of different material properties such as band structure, composition, electron density and thickness of the sample. Using a technique called energy-filtered transmission electron microscopy (EFTEM), it is possible to construct images consisting only of electrons with a desired energy loss. A thickness map can be constructed from a zero-loss and unfiltered EFTEM image. The zero-loss image is made up of only elastically scattered electrons, i.e. no energy loss, while the unfiltered image is made up of all the transmitted electrons. It can be shown that the thickness (t), mean free path (λ), the zero-loss intensity (I_0) and the total intensity (I_t) can be related as[27]

$$\frac{t}{\lambda} = \ln\left(\frac{I_t}{I_0}\right) \tag{2.28}$$

The mean free path is the average length the electron travels between scattering events, making t/λ the average number of scattering events for a transmitted electron. The unfiltered, zero-loss and resulting t/λ map are shown in Figure 2.14.

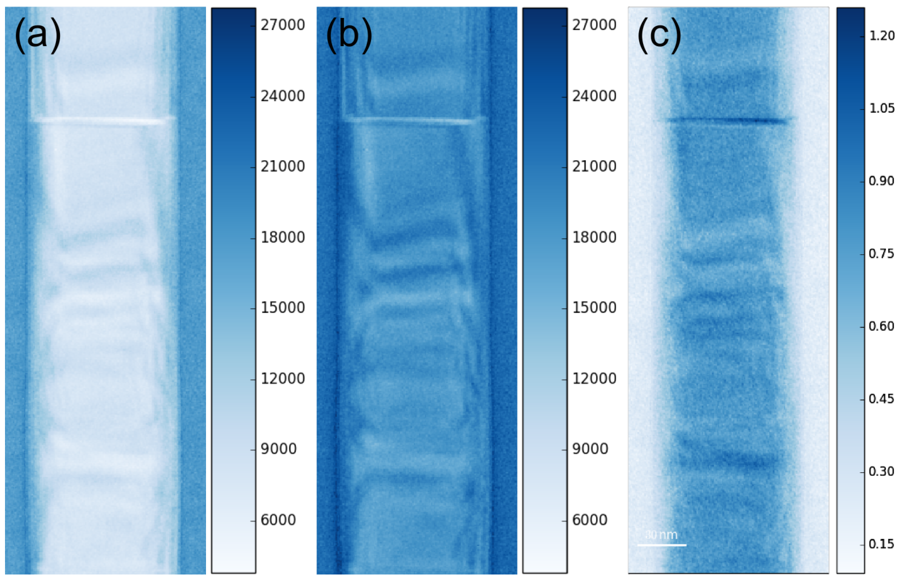


Figure 2.14: The (a) zero-loss, (b) unfiltered and (c) t/λ EFTEM image for a GaAs/GaAsSb nanowire (SC48). The GaAsSb insert is below the distinct horizontal line. The nanowire lies on an evenly thick C-film.

Chapter 3

Experimental

3.1 Microscope

The specimens were studied in a cold-FEG JEOL ARM2000F operated at 200 kV. The data was collected in STEM mode with a condenser aperatures of 10, 30, 40 and 50 mum (specified in results where relevant), a camera length of 2 cm and with a 3C probe size (probe diameter $\approx$1.1 Å) Images and spectra were obtained using Gatan DigitalMicrograph 2.3. EDS data were recorded by a JEOL Centurio SDD with thin windows and a collection angle of 0.98 srad and an elevation angle of 24.3°. The in-situ probe current was measured in vacuum before and after each EDS acquistion by a picoammeter in the drift tube of a GIF Quantum ER by indicating the voltage to 100 kV in the GIF control and using the plugin "Drift Tube Current". A 5 mm entrance aperture was used and it was verified that the entrance aperture was not clipping the field of view for the GIF (and hence not the build in the ampmeter). Thickness maps were constructed from EFTEM maps using the log-ratio method. For the Cliff-Lorimer method, calculated k-factors were as listed by the JEOL EDS software.

3.2 Material

3.2.1 Nanowires

Heterogeneous nanowires from three different batches were investigated in this study. All nanowires contain segments of pure gallium arsenide (GaAs) with inserts and/or shells containing either antimony or aluminium. The nanowires were grown by a Ga-assisted vapor-liquid-solid technique in a Varial Gen II Modular molecular beam epitaxy system.

GaAs/AlGaAs nanowires (SC343 and SC365)

A FEI Helios focused ion beam (FIB) was used to make cross sections of a GaAs/AlGaAs core-shell nanowires (Figure 3.1). FIB preparation was performed by Vidar Fauske (batch SC343) and Julie S. Nilsen (batch SC365). The growth parameters for these nanowires are shown in Table 3.1 and 3.2 respectively. The ζ-factor for Ga and As was determined from EDS spectrum images recorded from the GaAs core of a FIB made cross section in [111] from SC343. Spectra were obtained at different specimen tilt angles in order to investigate orientation effects. FIB made cross sections from SC365 in $[\bar{1}\bar{1}2]$ and [111] was used for quantitative analysis using the calibrated ζ-factors. The nanowires are dispersed on an amorphous 50 nm thick SiN film.

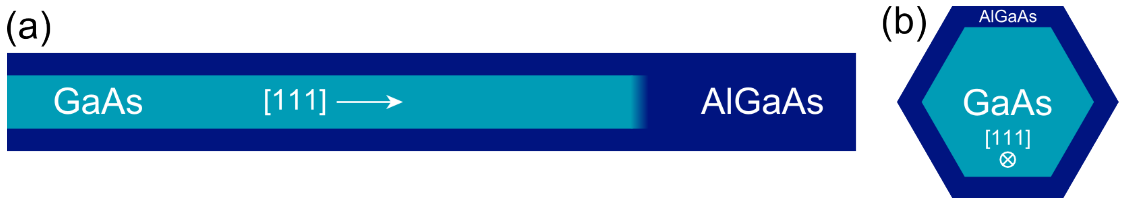


Figure 3.1: Schematic of a GaAs/AlGaAs core-shell nanowire.

Table 3.1: Growth parameters for SC343[28]

Step	Time [min]	Ga flux [ML/s]	As_2 flux [torr]	Al flux [ML/s]	Temperature [°C]
Axial GaAs	25	0.6	$5.5 \cdot 10^{-6}$	-	630
Axial AlGaAs	5	0.3	$5.5 \cdot 10^{-6}$	0.15	630
Solidification	10	-	$1.0 \cdot 10^{-5}$	-	630
AlGaAs shell	20	0.3	$1.0 \cdot 10^{-5}$	0.15	630
GaAs cap	8	0.3	-	-	630

Table 3.2: Growth parameters for SC365[29]

Step	Time [min]	Ga flux [ML/s]	As_2 flux [mbar]	Al flux [ML/s]	Temperature [°C]
Axial GaAs	20	0.7	$5.6 \cdot 10^{-6}$	-	630
Axial AlGaAs	3	0.7	$5.6 \cdot 10^{-6}$	0.3	630
Solidification	10	-	$5.6 \cdot 10^{-6}$	-	630
AlGaAs shell	30	0.2	$9.0 \cdot 10^{-6}$	0.1	630
GaAs cap	15	0.2	$9.0 \cdot 10^{-6}$	-	630

3.2.2 GaAs/GaAsSb nanowires (SC48)

A GaAs nanowire with an axial insert of GaAsSb was used to determine the ζ-factors of Ga and As in [110] (Figure 3.2). The nanowires were scrapped off and dispersed in isopropanol. A drop of the dispersion was put on a 300 mesh Cu grid with a holey carbon film. The sample was also used to create quantitative maps. The nanowire has one segment of GaAs in the zinc blende (ZB) structure, followed by the GaAsSb insert and then a segment of GaAs in the wurtzite (WZ) structure. The difference in crystal structures is not believed to have any significant effect on the ζ-factors or quantification. The growth were done in steps of 20 min (GaAs), 1 min (GaAsSb) and 5 min (GaAs) at 620°. In the GaAsSb region, the Ga, As and Sb fluxes were $3.3 \cdot 10^{-7}$, $4.2 \cdot 10^{-6}$ and $1.1 \cdot 10^{-6}$ respectively. These nanowires have been presented in previous publications[30]. The holey carbon film was used to determine ζ_{C}.

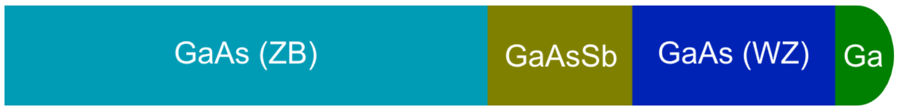


Figure 3.2: Schematic of GaAs nanowire with insert of GaAsSb (SC48)

3.2.3 Other materials

For determining ζ_{Al} an electropolished pure (99.99%) thin foil of Al was used.
$\zeta_{\mathrm{Sb}_{L\alpha}}$ was determined from cleaved GaSb dispersed on a holey carbon film.

3.3 Data processing

All the acquired EDS spectrum images were processed using a developer version of HyperSpy[1]. Note that the developer version has some functionality that are yet not included in the released versions (most recent being v.0.8.5). The ζ-factor method will be included in the release of v.1.0.0 (planned to be released in July 2016). PCA or NMF were performed as an initial step on all spectra to reduce noise. The intensity of the X-ray lines were determined by using an integration window of 1.2 FWHM and the background was substracted by specifying two windows over the background surrounding the peaks. HyperSpy scripts for determining the ζ-factor , applying the ζ-factor method and other analysis are included in appendix B. The *curve_fit* function in SciPy was used for least-square fitting. EFTEM thickness maps were analysed using Gatan DigitalMicrograph 2.3 and HyperSpy.

Chapter 4

Results

4.1 Measured ζ-factors

4.1.1 ζ-factors for gallium and arsenide from nanowires

ζ_{Ga} and ζ_{As} were determined from two different tilt series of GaAs/GaAsSb (SC48) in $[\bar{1}10]$ and a tilt series of a FIB cross section of a GaAs/AlGaAs core-shell nanowire (SC343) in [111]. For each spectrum image, the ζ-factor was calculated individually for each pixel. The average ζ-factors from each EDS spectrum image is plotted in Figure 4.2. The ζ-factors varies with specimen tilt relative to the detector. Tilting the specimen away from the detector significantly increases the measured ζ-factor. For high tilt angles towards the detector (10-30°), the ζ-factor becomes fairly constant.

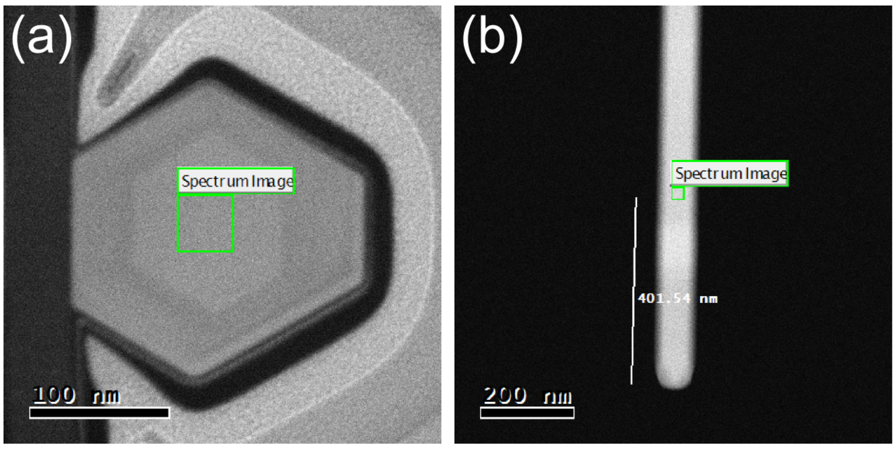


Figure 4.1: Survey images showing the regions from which EDS spectra were collected. (a) GaAs/AlGaAs (SC343) in [111], (b) GaAs/GaAsSb (SC48) in [110]

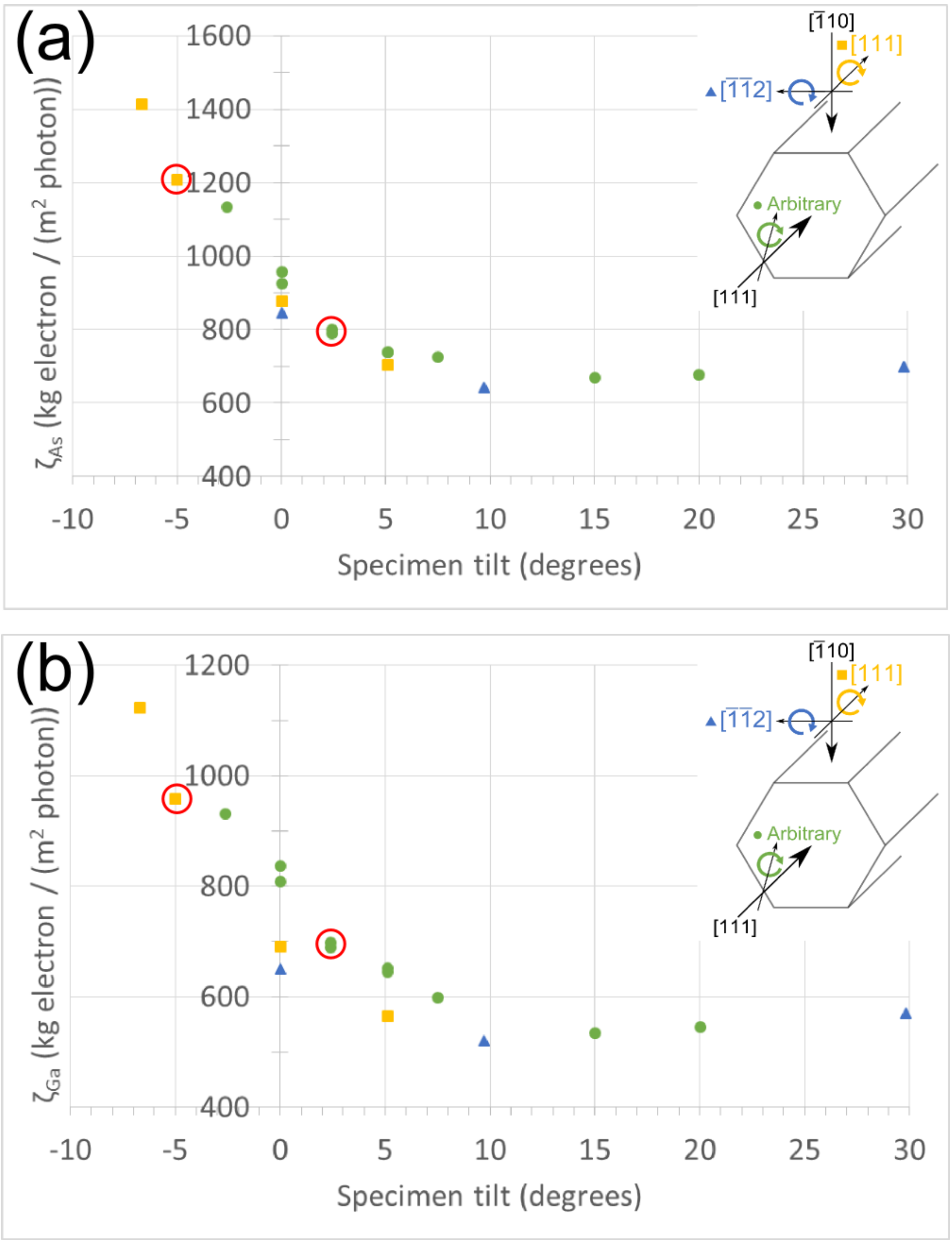


Figure 4.2: (a) ζ_{Ga} and (b) ζ_{As} for K_α lines as a function of specimen tilt relative to the EDS detector. The red circles indicates data collected on the zone axis. Inserts: Schematic showing the direction and rotation axes from which the EDS spectra were collected.

4.1.2 ζ-factors for antimony

The ζ-factors for the Sb L_α line was measured from a wedge shaped foil of GaSb together with additional values for the ζ-factor for the Ga K_α line. The signal from the full signal was added together and then an average thickness, estimated from an EFTEM map was used. The results are shown in Table 4.1.

Table 4.1: Measured ζ-factors from GaSb

Tilt [degrees]	ζ_{Sb} $\left[\frac{\text{kg electron}}{\text{m}^2\text{photon}}\right]$	ζ_{Ga} $\left[\frac{\text{kg electron}}{\text{m}^2\text{photon}}\right]$	Condenser aperture [μm]
6.7	748	1007	10
6.7	663	1160	40
6.7	582	977	50
11.7	437	766	40
11.7	242	369	50
16.8	538	980	40

4.1.3 ζ-factors for aluminium and carbon

The ζ-factor for aluminium was measured from a pure aluminium foil at 0° tilt. The average ζ_{Al} were measured to 242.97, 310.90 and 279.56 kg electron/(m^2 photon) for a 30, 40 and 50 μm condenser aperture respectively. The distribution of the measured ζ-factor for each spectrum image is shown in subfigures (a)-(c) of Figure 4.3. The average value of $\overline{\zeta_{\text{Al}}}$ = 278 kg electron/(m^2 photon) was used for quantification. The ζ-factor for Al was also determined indirectly from a linescan of SC365 in [$\bar{1}$12] (see Figure 4.15(e)) by assuming that the average As composition was exactly 50 at.% and that the ζ-factors for Ga and As was correct. This gave a value of ζ_{Al} = 299 kg electron/(m^2 photon) at 7.5° tilt (the sample was tilted to 8.1, but we used ζ_{Ga} and ζ_{As} based on 7.5°), which corresponds to a value of $\approx$ 367 kg electron/(m^2 photon) for 0° tilt according to (2.21). The ζ-factor for carbon was measured from a holey carbon film, by cropping out a small part of one of the EDS spectrum images of SC48. The sample was tilted -4.6°. The carbon thin film are assumed to have a thickness of 20-30 nm based on the EFTEM map given for SC48 in Figure 2.14(c). For this particular measurement a thickness of 28 nm was used (28.1 nm when adjusted for tilt). The distribution of measured ζ_{C} for the full spectrum image is shown in subfigure (d) of Figure 4.3. An average value of $\overline{\zeta_{\text{C}}}$ = 677 kg electron/(m^2 photon) was obtained.

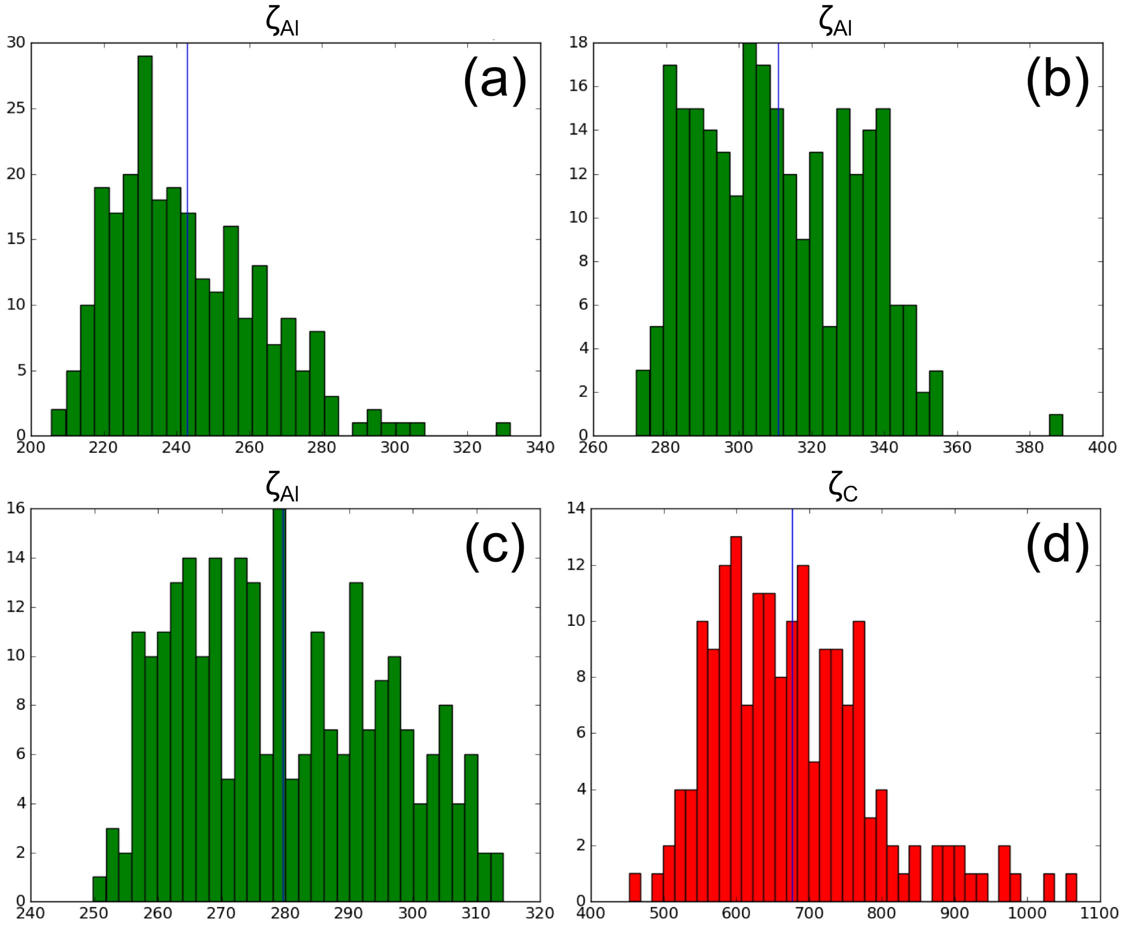


Figure 4.3: Distribution of ζ_{Al} using a condenser aperture of (a) 30 μm, (b) 40 μm and (c) 50 μm at 0°tilt and ζ_{C} at -4.6°tilt. The mean of each distribution is indicated by a solid vertical line.

4.2 Estimated ζ-factors from least square fitting

4.2.1 ζ-factors for other elements

The ζ-factors for other elements were estimated by fitting the theoretical expression (2.8) with the experimental determined K-line ζ-factors for carbon, aluminium, gallium and arsenide. The calculated ζ-factor for antimony could not be used as only the L-line is within the energy range of the EDS detector. In particular, a scaling factor for the ionization cross-section, the silicon layer and gold layer were fitted. The estimated ζ-factors are shown in Figure 4.4. The estimated detector-efficiency based on the fitting parameters is shown in Figure 4.5.

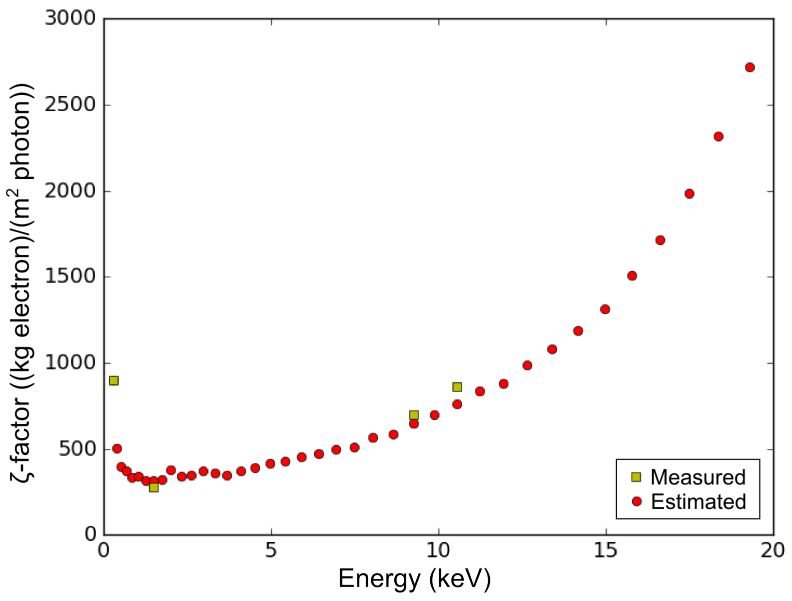


Figure 4.4: (a) Numerically fitted ζ-factors for K-lines for JEOL ARM200F (200 keV) based on experimentally ζ-factors (yellow blocks).

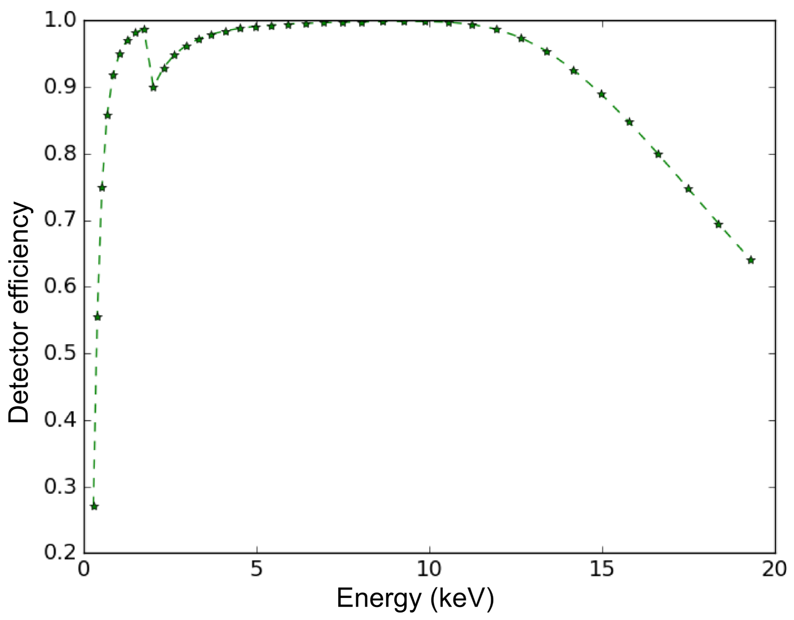


Figure 4.5: Detector efficiency for K-lines for JEOL ARM200F (200 keV) based on numerically fitted values for the thickness of gold contact layer and the silicon dead layer.

4.2.2 Shadowing model

The active detector area as a function of tilt was plotted based on (2.16) and then fitted to the geometrical model described in section 2.6. The ζ-factors from As and Ga from the nanowire tilt series were used (Figure 4.2). The model were fitted by using shadow-free ζ-factors measured at 29.8° (Figure 4.6 and Figure 4.8) and 20° (Figure 4.7 and Figure 4.9)) assuming circular or rectangular detector geometry. The fitted parameters are given in 4.2 where θ_E is the elevation angle, θ_D is the angle to lower part of detector, θ_U and θ_L is the upper and lower shadow angle respectively (Figure 2.8), δ is the detector tilt (Figure 2.10), d is the distance from the specimen to the centre of the detector and θ^0 is the tilt angle of the ζ-factors that was used for normalization.

Table 4.2: Estimated parameters from the shadow model

Shape	θ^0	Figure	χ^2	θ_E	θ_D	θ_U	θ_L	δ	d
Circular	29.8	4.6	0.0913	25.4	22.9	27.2	-4.4	6.8	10.3
Circular	20	4.7	0.1083	25.3	23.1	27.1	-4.1	33.0	12.4
Rectangular	29.8	4.8	0.0873	24.3	23.9	27.0	-3.6	3.5	13.8
Rectangular	20	4.9	0.0685	25.9	42.9	34.2	-18.1	0.1	10.2

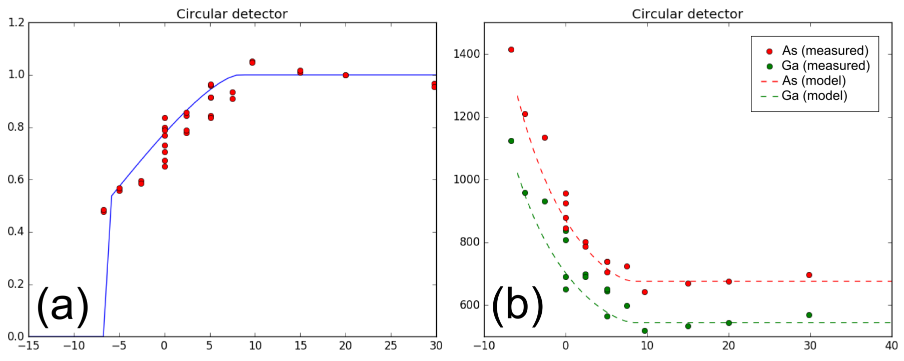


Figure 4.6: (a) The active detector area given by ζ-factors from the nanowire tilt series (Figure 4.2) with ζ^0 taken at 29.8° and the fitted model assuming a *circular* detector. (b) The ζ-factors from the nanowire tilt series together with the model.

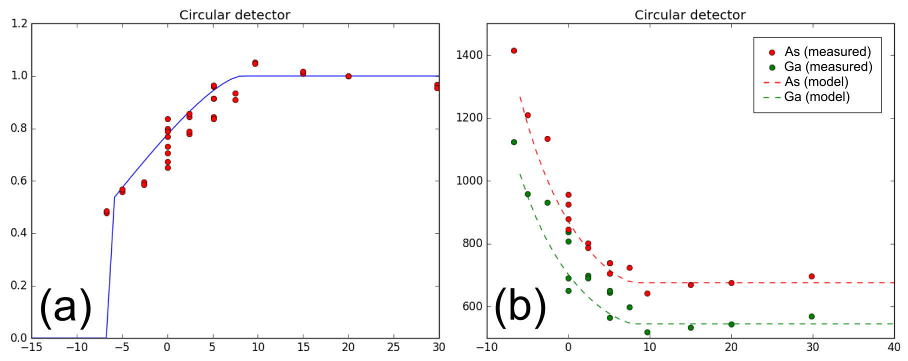


Figure 4.7: (a) The active detector area given by ζ-factors from the nanowire tilt series (Figure 4.2) with ζ^0 taken at 20° and the fitted model assuming a *circular* detector. (b) The ζ-factors from the nanowire tilt series together with the model.

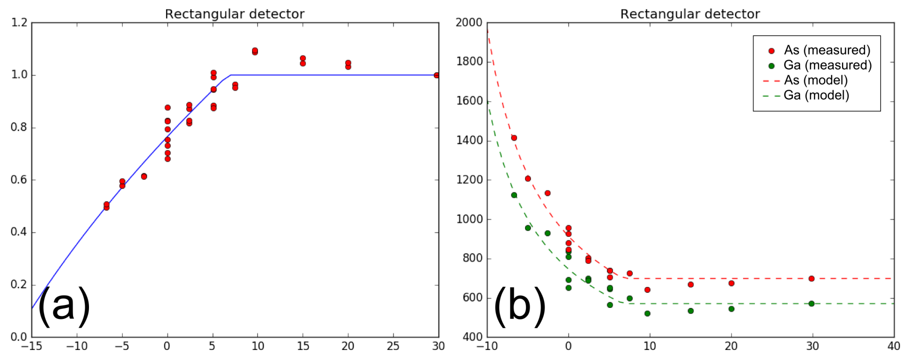


Figure 4.8: (a) The active detector area given by ζ-factors from the nanowire tilt series (Figure 4.2) with ζ^0 taken at 29.8° and the fitted model assuming a *rectangular* detector. (b) The ζ-factors from the nanowire tilt series together with the model.

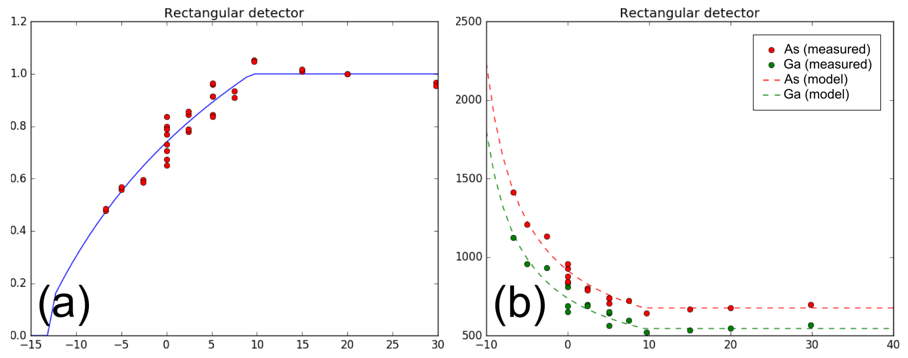


Figure 4.9: (a) The active detector area given by ζ-factors from the nanowire tilt series (Figure 4.2) with ζ^0 taken at 20° and the fitted model assuming a *rectangular* detector. (b) The ζ-factors from the nanowire tilt series together with the model.

4.3 Quantification

Quantitative maps were made from FIB cross sections of GaAs/AlGaAs core-shell (SC365) nanowires and for whole GaAs/GaAsSb (SC48) axial insert nanowires based on the determined ζ-factors.

4.3.1 GaAs/AlGaAs core-shell nanowire (SC365)

FIB made cross sections in [111]

Figure 4.10 shows compositional maps (a-c) for the GaAs/AlGaAs (SC365) core-shell nanowire FIB cross section in [111] (note that this is not the same nanowire as in Figure 4.1(a) which was used for calibrating the ζ-factors). In the core of the nanowire the composition is about 50 atomic percent for both Ga and As as expected. The compositional map using the Cliff-Lorimer method (not shown here) showed no visible differences. The central part of the EDS map was cropped out (insert in Figure 4.12), and the mean composition along $[\bar{1}10]$ was plotted (Figure 4.12). In the core, both the Cliff-Lorimer and ζ-factor method gives about 51 and 52 atomic percent Ga and 49 and 48 atomic percent As respectively. In the shell, both methods show an atomic percent of Al between 12 to 24 percent. Thickness map of the same sample are shown in Figure 4.11.

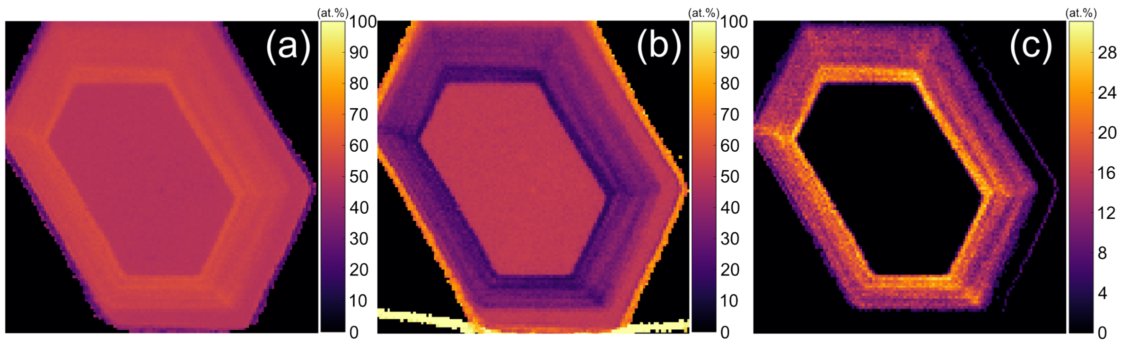


Figure 4.10: Compositional maps of (a) As, (b) Ga, (c) Al of a FIB cross section of a GaAs/AlGaAs core-shell nanowire (SC365) in [111] using the ζ-factor method.

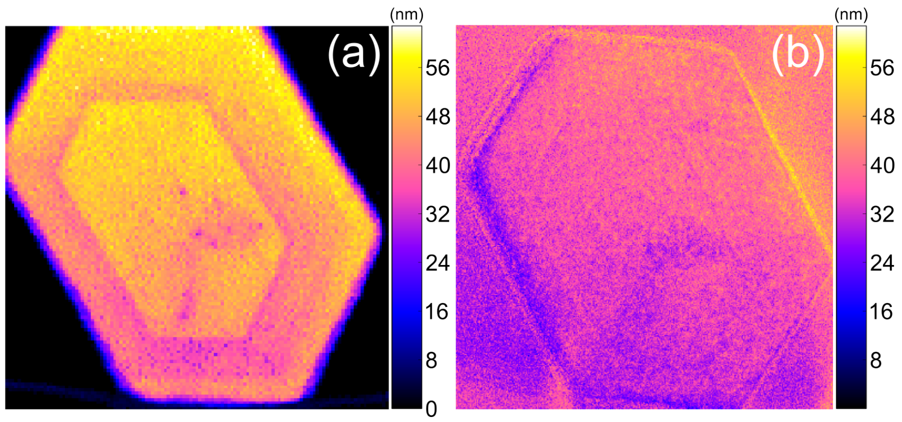


Figure 4.11: Thickness maps of the full FIB cross section of a GaAs/AlGaAs core-shell nanowire (SC365) in [111] from (a) the ζ-factor method and (b) EFTEM (assumed $\lambda \approx 103$). The thickness outside the nanowire has been set to zero.

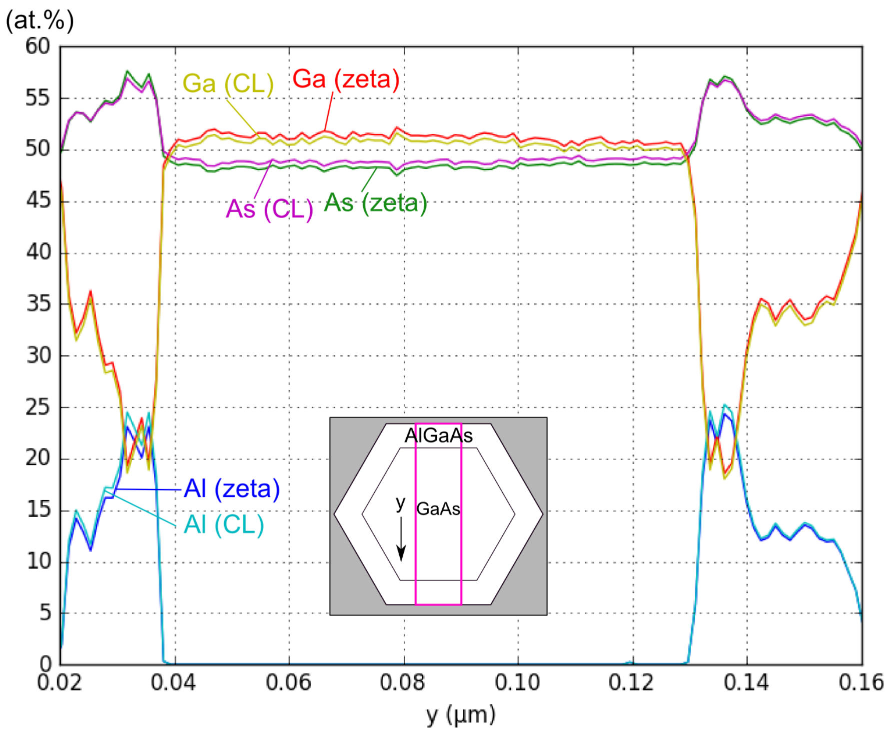


Figure 4.12: Comparison of the ζ-factor method and Cliff-Lorimer for the compositional map shown in Figure 4.10. The insert shows the region that was cropped and averaged.

FIB made cross sections in $[\bar{1}\bar{1}2]$

Compositional and thickness were also made for FIB made cross section of the GaAs/AlGaAs (SC365) core-shell nanowire in $[\bar{1}\bar{1}2]$. Figure 4.13 shows a comparison of the quantification using the ζ-factor method without and with absorption correction and the Cliff-Lorimer ratio technique.

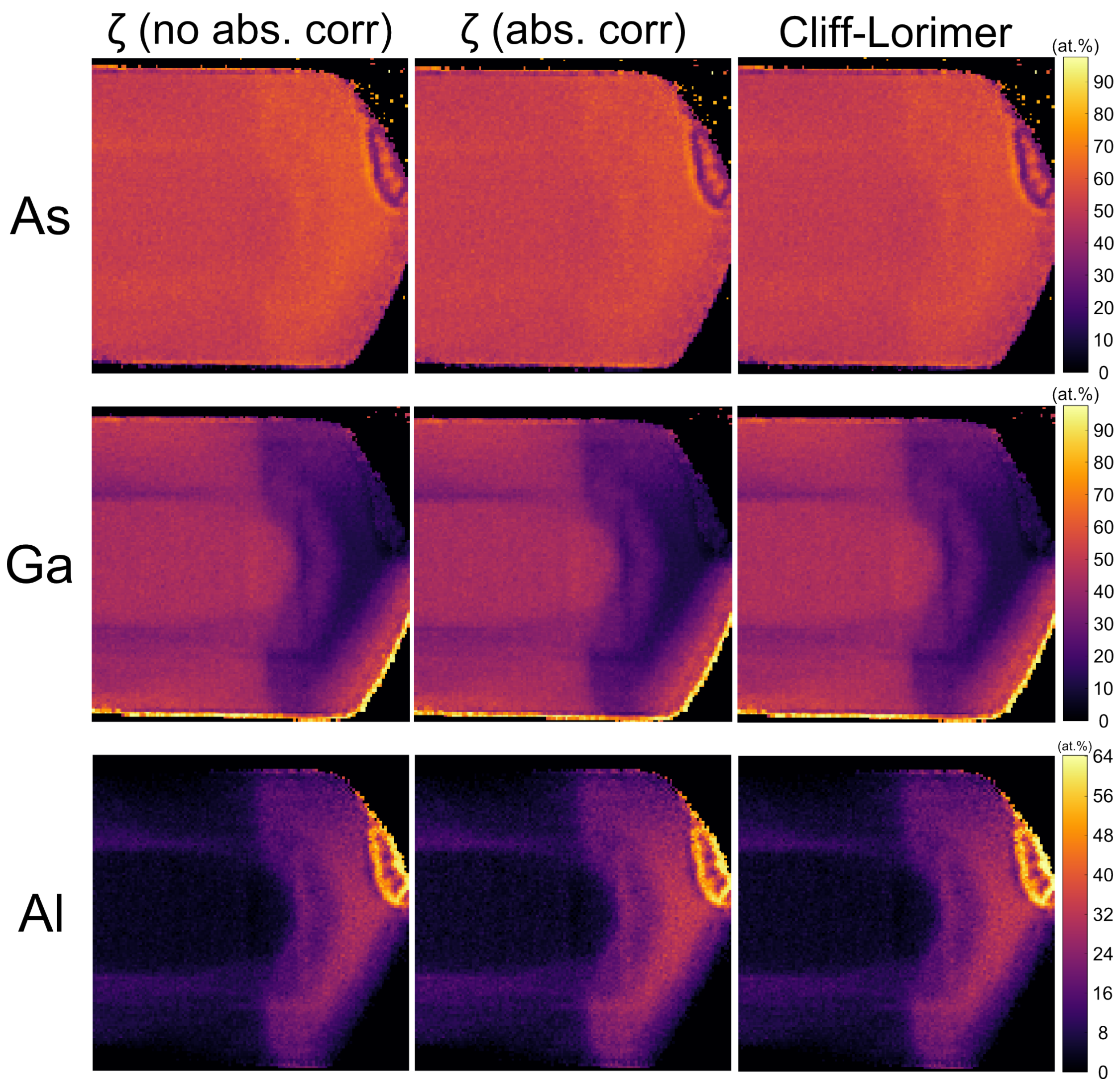


Figure 4.13: Composition of As, Ga and As (vertical axis) of the tip of the nanowire from a FIB made cross section of GaAs/AlGaAs core-shell nanowire (SC365) in $[\bar{1}\bar{1}2]$ using the ζ-factor method withouth absorption correction, the ζ-factor method with absorption correction and the Cliff-Lorimer ratio technique (horizontal axis).

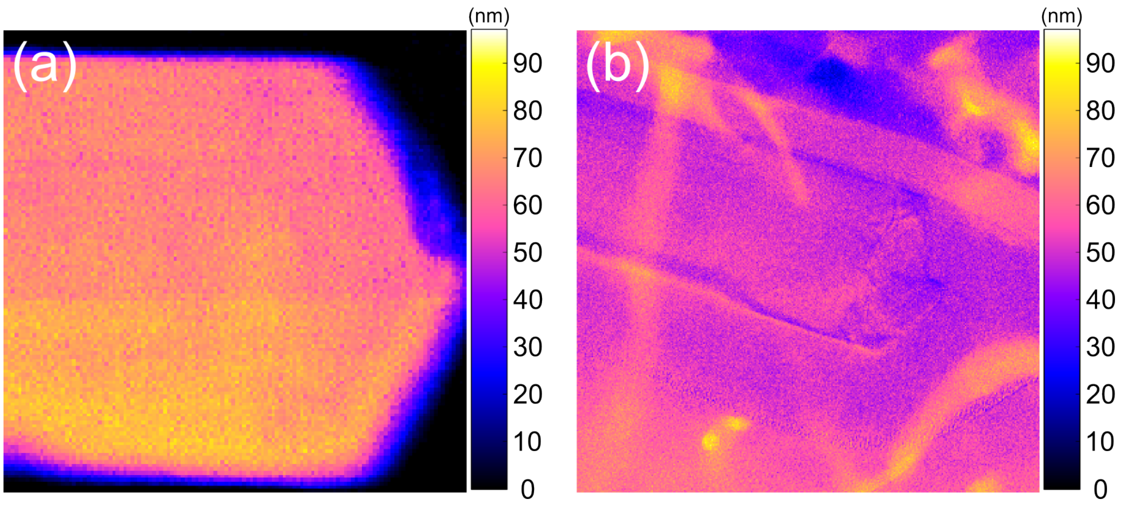


Figure 4.14: Thickness maps of the full FIB cross section of a GaAs/AlGaAs core-shell nanowire (SC365) in $[\bar{1}\bar{1}2]$ constructed by (a) the ζ-factor method (with absorption correction) and (b) EFTEM (assumed $\lambda \approx 103$). The thickness outside the nanowire has been set to zero.

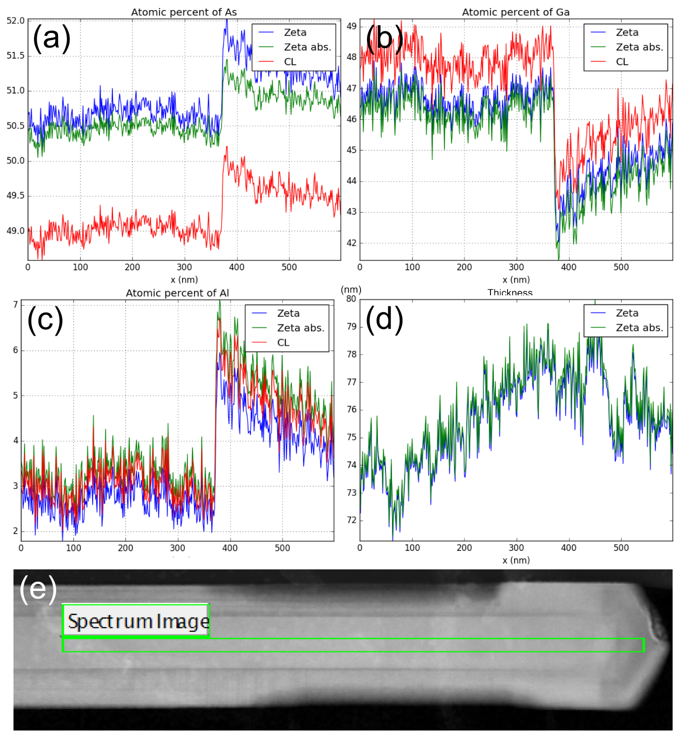


Figure 4.15: Composition of (a) As, (b) Ga, (c) Al and the (d) thickness profile from the core of the GaAs/AlGaAs (SC365) in $[\bar{1}\bar{1}2]$ along the length of the nanowire (scan in [111] direction) and (e) survey image showing the region the EDS data were collected from. Note the transition from the GaAs core to the AlGaAs insert.

To investigate the transition from the GaAs core to the AlGaAs insert, a linescan was taken from the FIB sample in $[\bar{1}\bar{1}2]$ (Figure 4.15). The ζ-factor for Al was also determined indirectly from the linescan in Figure 4.15, by automated trial and error, while keeping ζ_{Ga} and ζ_{As} fixed and fixing the composition of As to 50%. This gave a higher value of ζ_{Al} which indicates that the value from the Al thin-foil might be too low. A close up of the transition is shown in Figure 4.16 included the compositional and thickness map from the indirectly determined ζ_{Al}.

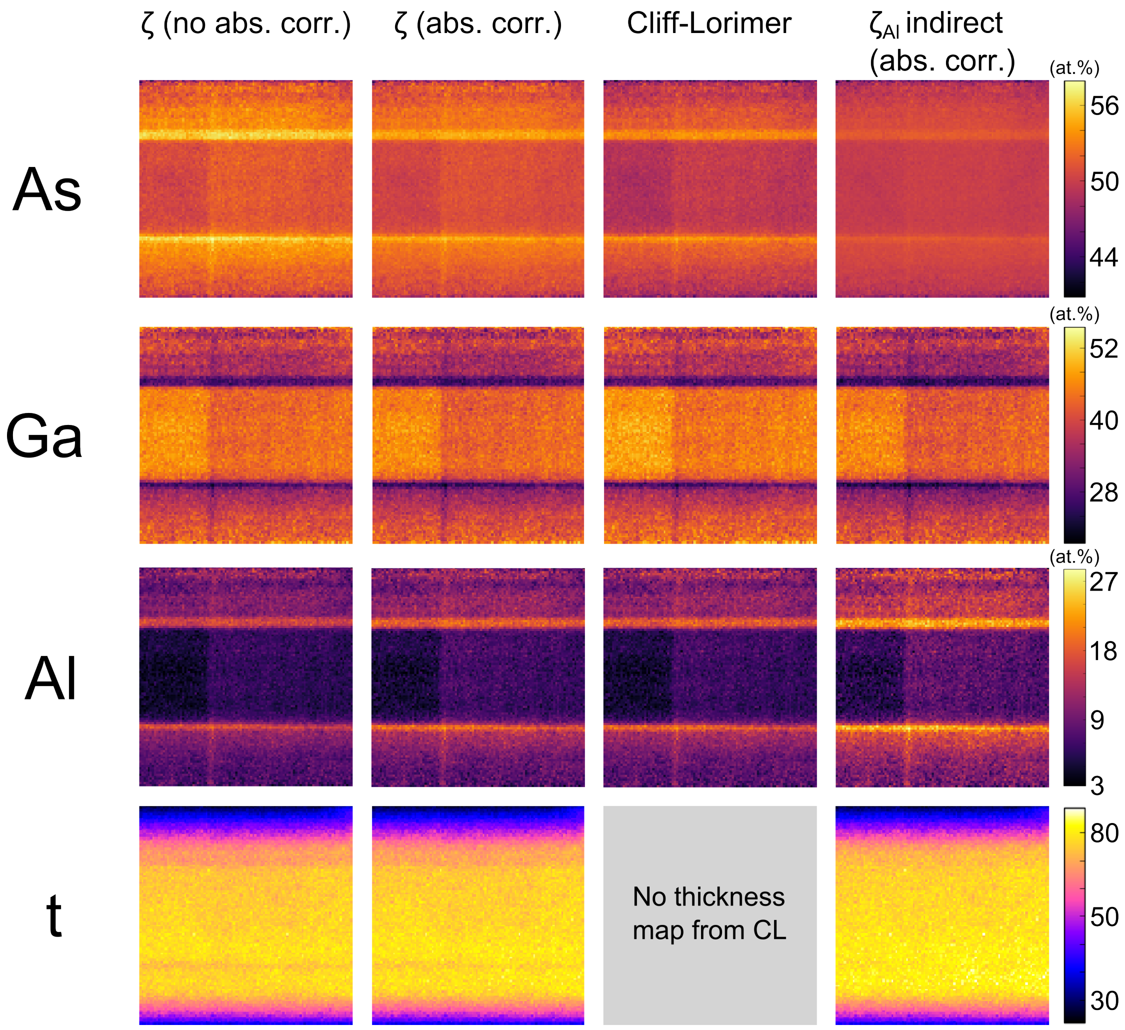


Figure 4.16: Close up of the transition between the GaAs and AlGaAs core of the FIB lamella of SC365 in $[\bar{1}12]$. Quantitative maps from the ζ-factor method without absorption, with absorption, from CL, and using the indirectly measured ζ_{Al} with absorption correction (horizontal axis). The composition of As, Ga and Al are shown along the vertical axis. The last row display thickness maps.

4.4 GaAs/GaAsSb nanowire (SC48)

Quantitative maps and thickness maps were also made for the GaAs/GaAsSb (SC48) nanowire as shown in Figure 4.17. The composition of Ga is unchanged along the length of the wire as the Sb atom take the place of the As atoms. At the end of the nanowire is a Ga droplet.

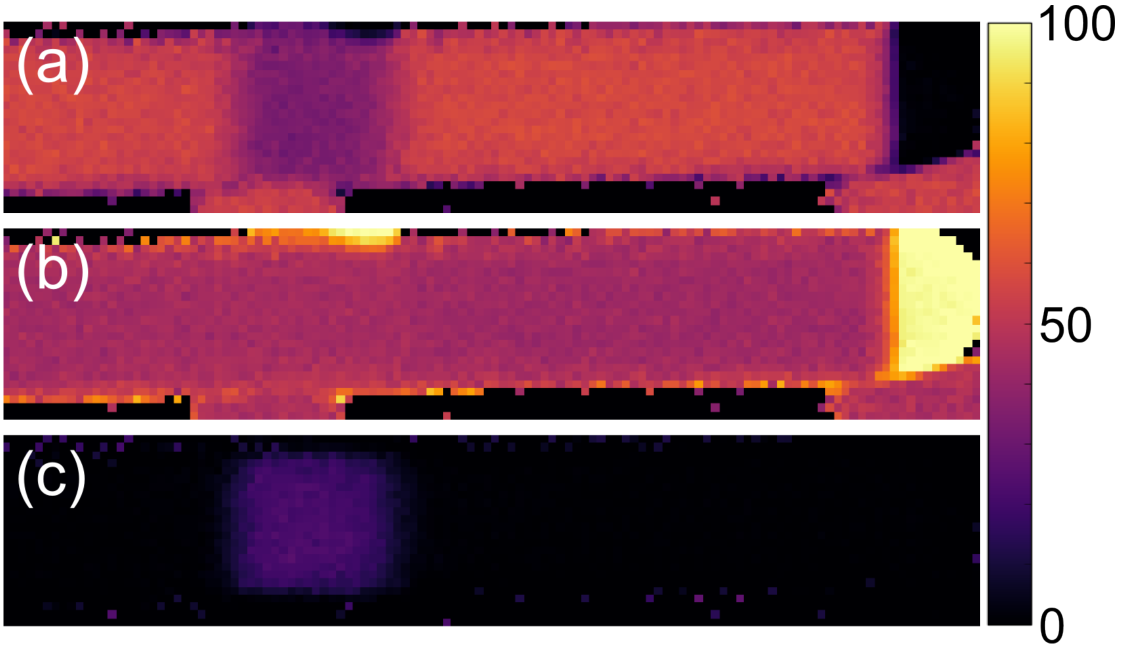


Figure 4.17: Composition of (a) As, (b) Ga and (c) Sb of a GaAs/GaAsSb (SC48) whole nanowire using the ζ-factor method.

4.5 Thickness measurements

The thickness of the GaAs/GaAsSb (SC48) nanowire in [110] was estimated to 82.3 nm based on measurements of the projected width. An EFTEM thickness map (Figure 2.14(c)) of the same nanowire gave t/$\lambda \approx 0.79$. Based on this, $\lambda_{\mathrm{GaAs}} = \frac{82.3}{0.79} = 104.17 \approx 104$. Because there were no good sources for λ_{GaAs} at 200 keV, this value was used when determining thickness from other maps.

Chapter 5

Discussion

5.1 Relative error of ζ-factors

It is possible to give an estimate of the error in the measured ζ-factors from the errors of the independent variables in (2.9)

$$\left(\frac{\Delta\zeta}{\zeta}\right)^2 = \left[\left(\frac{\Delta I_p}{I_p}\right)^2 + \left(\frac{\Delta C}{C}\right)^2 + \left(\frac{\Delta t}{t}\right)^2 + \left(\frac{\Delta I}{I}\right)^2\right]^{1/2} \tag{5.1}$$

Note that we have not included the error of density as density and composition are correlated to each other. The acquisition time is also omitted as it is considered a constant. All the studied specimen for the ζ-factor determination have a well-defined composition, so the term $\Delta C/C$ can be considered to be negigible compared to the other terms. The measurements of the probe current had an estimated relative error, $\Delta I_p \approx 0.005$ nA, which gives a value of $\frac{\Delta I_p}{I_p}$ of $\approx 1.2\%$ and 0.8% for a 40 μm and 60 μm condenser aperture respectively. The probe current was measured both before and after each acquisition and the average value was used to minimize this error. The term $\Delta I/I$ describes the error in the measured intensity for the X-ray peak. X-rays counts follow Poisson statistics[31] so ΔI can be expressed as $\nu\sqrt{I}$, where $\nu = 3$ for a confidence limit of 99%. This error should be low as long as the number of counts are significant. The measured intensity, I also depends on which integration windows and background subtraction routines that was used, which is why the same data processing routines must be used for all data. Although, we had to reduce the acquisition time (and thereby the total number of counts) in order to avoid beam damage, this is error is still small compared to the error in thickness. The thickness measurements for EFTEM images are based on the log-ratio method as given by (2.28). In order to determine the thickness, the mean free path for the sample must be known. Most studies regarding the measurement of the mean free path for GaAs are for energies less than 5 keV [32, 33]. Egerton[27] reports a value of $\lambda_{\mathrm{GaAs}} = 95$ for GaAs at 100 keV with a semi collection angle of 10

mrad. Because no reliable literature report on measurements for λ_{GaAs} was available for 200 keV, the mean free path used for GaAs was calculated by comparing an EFTEM thickness map (Figure 2.14(c)) and the thickness measured from the projected width of the GaAs/GaAsSb (SC48) nanowire in [-110]. The values for the projected width measurement varies with ± 5 nm depending on where we define the edge in the intensity profile. This introduces an error in the thickness of $\Delta t/t = 5/95 \approx 5.26\%$ for the two tilt series of SC48. For the FIB made GaAs/AlGaAs (SC343) additional errors as it is difficult to locate exactly which parts of the EFTEM map to use for thickness measurements because it was taken at a different magnification than the EDS SI. However, as the FIB sample is relative flat this should only introduce a small increase in the relative error of the thickness.
An estimate of the relative error in the ζ-factor can be given from the error in thickness alone as all other variables contribute insignificantly to the overall error. For the ζ-factors from Ga and As determined from SC48 and SC365, the relative error are estimated to about 5-7 % as it is based on measurements from the projected width and the error this introduces to the estimated λ_{GaAs}.
The ζ-factors for Al is more likely to have a relative error as large as 15-20% as values are reported from 100 nm[27] at 100 kV and 133.6[34] at 200 kV. For the Al research done at the TEM Gemini Centre, a fixed λ is used, but these measurements are only routinely done at 150 kV. The error in the ζ-factor for carbon is also around 15-20% as the thickness of the thinfilm is somewhere between 20 and 30 nm.
To improve the ζ-factor method the thickness should be determined from the same area and in the same mode. An alternative is to record low-loss EELS in STEM mode prior to EDS collection. This would allow a thickness to be defined for every pixel of the EDS SI. Unfortunately this can not be done in parallel as the GIF is needed for measuring the probe current. However, it might be possible to get the total probe current from the zero loss taken in vacuum.

5.2 ζ-factors from GaAs nanowires

In an ideal setup, the collection angle should be at its maximum (no shadowing) when the specimen stage is perpendicular to the incoming beam (i.e. 0 degrees tilt) or tilted further towards the detector (i.e. positive tilt). From Figure 4.2 we see that there is still a continuous decline in the ζ-factors even for positive tilt angles. It is difficult to say exactly where the shadowing stops due to the low number SI taken at higher tilt angles, but it seems like the ζ-factors are fairly constant for tilts above 10°. There are some differences in the ζ-factor from the FIB lamella (SC343) and the whole nanowires (SC48) at 0 ° tilt. The thickness of the two nanowires from SC48 is likely to be more accurate as they were determined from the projected width, while SC343 was determined from an EFTEM map as discussed in the previous section. However, at higher tilts, the ζ-

factors from SC343 give even lower ζ-factors. Because the ζ-factor is proportional with thickness this deviation might be caused by errors in the estimated thickness for the FIB lamella. However, for higher tilt angles, the FIB lamella seems to give similar results to the whole nanowires, which indicates that the error is due to something else. It is very likely that absorption effects cause this, as the effect seems to disappear for higher angles as the absorption path length decrease. In fact, looking at the sum of the spectra from the GaAs core region of the SC343 specimen (Figure 5.1) shows X-rays from other elements such as Al, O, and Si. These are all examples of spurious X-rays (i.e. secondary X-rays that is generated in the specimen, but not from the analysed region). Si origins from the substrate the nanowires in the FIB lamella lies on. Although, these low energy X-ray lines might cause absorption, this effect should not be very large as we are using the K-lines for Ga and As which lies far away from the absorption edges of the respective absorbing elements.

5.3 Shadowing model

5.3.1 Robustness of the fitting

According to JEOL, the elevation angle in the used set-up should be around 24.3°, we therefore restricted the bounds of the elevation angle to 20-30°. Note that none of the model described in the result gave any meaningful results when a wider range (10-30°) was applied, which might seem odd at first as 20-30° is within the same range. However, the trust region reflective algorithm from the curve fit method in SciPy only find the local minima of the function. Because there are a lot of free parameters to be fitted, a lot of such local minima exist and the model is therefore very sensitive to the bounds and it also define the starting guess for the parameters. The starting parameters could also be given explicitly, but due to lack of knowledge of what the real values are, it's better to adjust this by giving meaningful bounds. Another issue that makes the model less robust is that data has only been collected from a limited tilt range. It would have been especially useful if more data had been collected from a larger range of negative tilts in order to identify at which angle the detector is completely shadowed. Note that if the shadow range, θ_U - θ_L is very small, this would never happen, but it is impossible to know without having any data for large negative tilts). The finer the tilt series, the better model. Another way to improve the current method is to try to find better bounds for the parameters by investigation the geometry of the set-up or getting more of this information from the manufacturer. Another problem with the current implementation is that there is no constrains on the relationship between the parameters. This could be implemented as a future improvement.

5.3.2 Analysis of the fitted models

The active detector area was determined by taking a tilt series of the core of GaAs nanowires and the plotting the measured ζ-factors normalized by the minimum ζ-factor (or any ζ factor that is taken under shadow free conditions). Although bounds were set on the allowed range for the fitted parameters, the current implementation does not set any restriction on the relationship between the different parameters. Due to this, not all the estimated models make physical sense. For example, at first glance, the rectangular model with $\theta^0 = 20°$ seems to give the best fit to the experimental data as it has the lowest χ^2 value. However, the model estimates the lower detector angle to be higher than the elevation angle which doesn't make any sense as by definition the elevation angle is the angle from the specimen to the centre of the detector, while the lower detector angle is the from the specimen angle to the lower edge of the detector. Thus, these model parameters can be rejected. However, it does not imply that the detector geometry are not rectangular as it might be an effect of the sensitivity of the current implementation as mention in the previous section. The three other fits listed in 4.2 gives very similar parameters. The main difference is the value of δ which for the circular detector model for $\theta^0 = 20°$ gives a value of 33° compared to 6.8 for the other circular fit for $\theta^0 = 29.8°$ and 3.5 for the rectangular fit for $\theta^0 = 20°$. A value of 33° is more likely as the collection angle is maximized when $\delta = \theta_E$. It is impossible to conclude that one model is better than the other, because the models are very similar in the limited acquired tilt range. A full tilt series from a large negative tilt with full shadowing to a large positive tilt with no shadowing is needed in order to identify the detector geometry in the used experimental set-up.

5.3.3 Why we use ζ-factors for the active detector area ratio

Yeoh et al.[24] used the intensity ratio ($I(\theta)/I^0$) to determine the active detector area. This worked in their case because they used a powdered sample of cobalt oxide nanoparticles were the thickness can be regarded as independent of specimen tilt due to the particles spherical shape. On the other hand, when using a thin-film the specimen thickness *will* vary as a function as shown in Figure 2.13. Based on (2.6), the intensity ratio for a thin film is given by

$$\frac{I(\theta)}{I^0} = \frac{\Omega(\theta)\cos(\theta)}{\Omega^0\cos(\theta^0)} \neq \Delta A \tag{5.2}$$

From the definition of the ζ-factor in (2.9), we see that the inverse of the ζ-factors is the intensity normalized by the thickness, density, composition and total electron dose. This is why we used ζ-factors in (2.15) and not intensities.

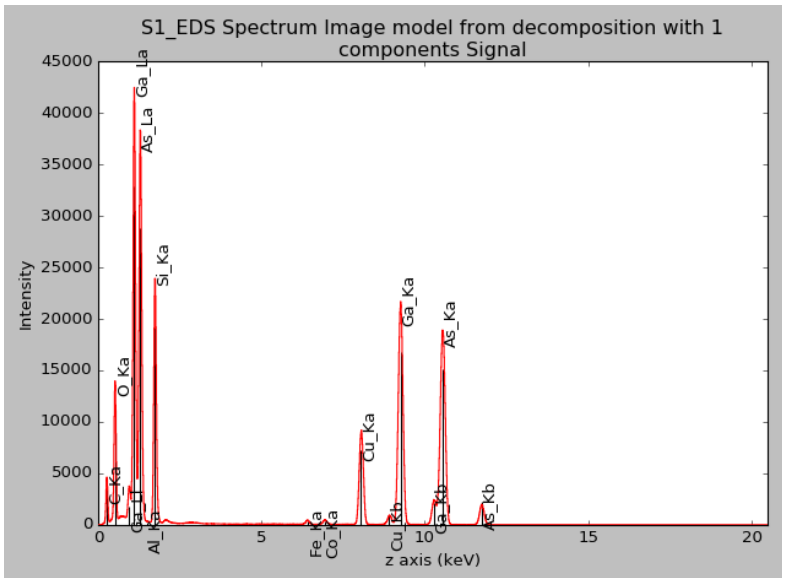


Figure 5.1: Sum of spectra from the EDS SI of the FIB cross section of GaAs/AlGaAs (SC343) used to calibrate the ζ-factors for the tilt series in [111] (indicated in Figure 4.1).

5.4 Quantitative analysis

The composition of As in all GaAs/AlGaAs should always be 50 at.% as the Al atom replaces Ga in the lattice. In Figure 4.15(a) we see that the composition of As is in the GaAs core is about 50.5 at.% for the ζ-factor method. The absorption-corrected version gives a slightly better result than the non-absorptive corrected version, but the difference is small. At the same time, the CL ratio technique gives about 49 at.% As and is therefore off by 0.5 at.%. However, in the AlGaAs region CL gives a better result than the ζ-factor method. Note that also in this region, the absorption corrected version gives an improvement over the basic ζ-factor method (0.5 at.% closer to the true value), however, CL still give the best result. The fact that the ζ-factor gave very promising results in the GaAs segment, but then suddenly went off in the AlGaAs segment indicates that there might be something wrong with the value of ζ_{Al}. The ζ-factor method will always compute the right composition as long as the ζ-factors for you system are calibrated correctly relatively to each other. This is a consequence of the composition is calculated by $\zeta_{\mathrm{A}} I_{\mathrm{A}}$ divided by the sum $\sum_j \zeta_j I_j$. If the ζ-factors used for quantification have been determined simultaneously from the same sample, which is the case for our values of ζ_{Ga} and ζ_{As}, they would give the correct composition if we then later quantified the same kind of sample, here GaAs, independently of whatever thickness, density and electron dose was used to calculate the ζ-factors, because they would scale with the same factor. However, they would give a completely wrong thickness if the electron dose and mass-thickness was inaccurate. This explains why the ζ-factor method gives so good results (almost 50 at.% As) in the GaAs core.

The large deviation from 50 at.% for As in the AlGaAs region is a strong indication that something is off with the relationship between the ζ-factors. It is most likely that the error lies with ζ_{Al} and especially due to error in the thickness as discussed earlier. For the ζ-factor method the composition of As is given as $C_{\mathrm{As}} = \zeta_{\mathrm{As}} I_{\mathrm{As}} / \zeta_{\mathrm{Al}} I_{\mathrm{Al}} \zeta_{\mathrm{As}} I_{\mathrm{As}} \zeta_{\mathrm{Ga}} I_{\mathrm{Ga}}$. Because a too high value of C_{As} was observed, the value of $\zeta_{A} l$ should therefore be higher if we keep all the other parameters fixed. This was investigated further by writing a computer script that by trial and convergence found the $\zeta_{A} l$ that minimized the mean of $|C_{\mathrm{As}}(at.\%)–50|$ along the linescan in Figure 4.15. This process gave a value of ζ_{Al} = 299 kg electron/(m^2 photon) , compared to the value of 227 kg electron/(m^2 photon) which was used for the ζ-factor method. Note also that the value of 227 is the tilt-adjusted value according to (2.21) for 7.5° using the values given by ζ_{As} and ζ_{Ga}. The new value of 299 kg electron/(m^2 photon) for ζ_{Al} was used to quantify a map of the transition between the GaAs and AlGaAs segment of the same nanowire cross section. The result, as well as a comparison with the ζ-factor method and CL, are shown in Figure 4.16. In this figure we see that for As, the composition is more evenly distributed (close to 50 at.%) between the two segments for CL than for the ζ-factor method when using ζ_{Al} = 227 kg electron/(m^2 photon) . The indirect ζ_{Al} gives a very even As composition of around 50 at.% as expected, because it was estimated to give this value. The thickness maps, given in the lower row of Figure 4.16 shows a smooth an even thickness for the indirect ζ_{Al} compared to the other maps based on the ζ-factor method which shows distinct sharp lines in the intersection between the core and the shell. This is a good verification that the indirect ζ_{Al} gives the correct result. Please note that when finding ζ_{Al} indirectly, the average As composition was brought as close to 50 at.% as possible, but no restrictions were set on the thickness. For completeness, it should be mention that the absolute thickness (or values) given by the thickness map is not necessarily correct as it might be scaled due to a systematic error in the measured ζ-factors. However, the composition should still be accurate, as it is only the relationship between them that determine the composition.

5.5 Epilogue: Contribution to HyperSpy

Originally, the plan was to implement the zeta-factor method in HyperSpy based on the code I (Garmannslund) developed as part of my project work in TFY4520. However, it turned out that some work on implementing the basic ζ-factor method (not including absorption correction) had already been done by Dr. Pierre Burdet, a developer who no longer have time to contribute to the project. I managed to get Burdet's code working and verified that it produced the same results as my own code. Meanwhile, Dr. Katherine E. MacArthur at University of Oxford/Forschungzentrum Jülich was implementing the related EDX cross section method, which is a mathematically transformation of the zeta-factor method to give the composition in number of atoms rather than weight percent. She had also incorporated Burdet's zeta-factor method with help of D. Johnstone at University of Cambridge. Instead of having two people working on the same thing, it was commonly decided to keep Katherine's implementation since she was also integrating the EDX cross section method. My contribution to the HyperSpy consist of reviewing this code. This reviewing process included bug fixes, suggesting changes to the application programming interface and verifying that it produced the correct output. In this Master's thesis I have developed a routine (see zeta_factor_method.py in appendix B) that incorporate the thin-film absorption correction in the quantification for the ζ-factor method. The developed routine is planned to be integrated into the EDS TEM module of the HyperSpy library during the early summer 2016 and will hopefully be incorporated ready for release of the next major release (v.1.0.0) in the late summer. This story is a good description of both the good sides (fast cooperation across borders) and the challenges (coordinating ongoing parallel work) of open source development. It is very positive to experience that the barrier to contribute is low so that even MSc students can contribute and cooperate with more experienced researchers at other institutions.

Chapter 6

Conclusion

In this work, several computational routines related to the ζ-factor method have been developed. This includes routines for determining the ζ-factors, fitting the ζ-factors to the theoretical expression, fitting the ζ-factors from a tilt series to a model for shadowing with two different detector geometries and most importantly the implementation of the thin-film absorption correction for the ζ-factor method in HyperSpy. Hopefully, with some further adjustments and tests, the latter will become part of the next release of HyperSpy, scheduled for release late summer 2016.

The ζ-factors have been determined experimentally for Ga, As, Sb, C and Al. The K-line ζ-factors for other elements have been estimated based on these measurements (except Sb which is for the L-line) by fitting a scaling factor to the ionization cross-section and the thickness of the silicon dead layer and the gold contact layer. The detector efficiency is highest (about 100%) for X-rays from 4-13 keV which is the typical range for EDS in (S)TEM.

The ζ-factors have been determined for Ga and As in $[\bar{1}10]$ and [111] for different X-tilts. The effect of shadowing from the specimen holder has been shown to have a large impact in the measured ζ-factors. These effects can be avoided by tilting to a high angle, but this is not an option if you want to study certain orientations of your specimen. It has been demonstrated that it is possible to take these effects into account by fitting ζ-factors from tilt series to a simple geometrical model. However, the tilt range was too limited to identify the detector geometry. This method can be further improved by either taking a tilt series over a larger range or acquiring more information about the experimental set-up. In principle, this should only be needed to be done once, preferably by using a pure element thin-film to avoid other tilt or orientation dependent effects.

The quantitative analysis clearly illustrates how important it is to be accurate when calibrating the ζ-factors. The largest source of error in the ζ-factor is caused by errors in the thickness. While the error is estimated to be around 5-6% when the thickness is measured using the projected width, errors as large as 15-20% are not unrealistic for the thickness measurments for ζ_{Al} and ζ_{C}. This large error can explain why CL in some of the studied cases appears to give more accurate results. However, if the ζ-factors are properly calibrated, the method offers much more than what is possible with CL. This includes built-in absorption correction, the possibility to create thickness maps from EDS SI alone, and the possibility of easily accounting for shadowing effects by calculating the active detector area. The ζ-factor method is therefore very promising technique for accurate microanalysis as long as the calibration is accurate.

Bibliography

[1] Francisco de la Peña et al. *HyperSpy 0.8.5*. June 2016.

[2] A. S. Eggeman, R. Krakow, and P. A. Midgley. “Scanning precession electron tomography for three-dimensional nanoscale orientation imaging and crystallographic analysis”. In: *Nat. Commun.* 6 (June 2015).

[3] David Rossouw et al. “Multicomponent Signal Unmixing from Nanoheterostructures: Overcoming the Traditional Challenges of Nanoscale X-ray Analysis via Machine Learning”. In: *Nano Letters* 15.4 (2015). PMID: 25760234, pp. 2716–2720. eprint: `http://dx.doi.org/10.1021/acs.nanolett.5b00449`.

[4] Vidar T. Fauske et al. “In Situ Heat-Induced Replacement of GaAs Nanowires by Au”. In: *Nano Letters* 16.5 (2016). PMID: 27104293, pp. 3051–3057. eprint: `http://dx.doi.org/10.1021/acs.nanolett.6b00109`.

[5] G. Cliff and G. W. Lorimer. “The quantitative analysis of thin specimens”. In: *Journal of Microscopy* 103.2 (1975), pp. 203–207.

[6] M. Watanabe and D.B. Williams. “The quantitative analysis of thin specimens: a review of progress from the Cliff-Lorimer to the new ζ-factor methods”. In: *Journal of Microscopy* 221 (2006), pp. 89–109.

[7] Mark S. Gudiksen et al. “Growth of nanowire superlattice structures for nanoscale photonics and electronics”. In: *Nature* 415 (Feb. 2002), pp. 617–620.

[8] Jeppe V. Holm et al. “Surface-passivated GaAsP single-nanowire solar cells exceeding 10% efficiency grown on silicon”. In: *Nature Communications* 4.1498 (2013).

[9] Lyubomir Ahtapodov et al. “A Story Told by a Single Nanowire: Optical Properties of Wurtzite GaAs”. In: *Nano Letters* 12.12 (2012). PMID: 23131181, pp. 6090–6095. eprint: `http://dx.doi.org/10.1021/nl3025714`.

[10] Dingding Ren et al. “New Insights into the Origins of Sb-Induced Effects on Self-Catalyzed GaAsSb Nanowire Arrays”. In: *Nano Letters* 16.2 (2016). PMID: 26726825, pp. 1201–1209. eprint: `http://dx.doi.org/10.1021/acs.nanolett.5b04503`.

[11] Junghwan Huh et al. “Rectifying Single GaAsSb Nanowire Devices Based on Self-Induced Compositional Gradients”. In: *Nano Letters* 15.6 (2015). PMID: 25941743, pp. 3709–3715. eprint: `http://dx.doi.org/10.1021/acs.nanolett.5b00089`.

[12] Marc De Graef. *Introduction to Conventional Transmission Electron Microscopy*. Cambridge University Press, 2003.

[13] D. Shindo and T. Oikawa. *Analytical Electron Microscopy for Materials Science*. Springer, 2002.

[14] D. B. Williams and C. B. Carter. *Transmission Electron Microscope*. English. Second. Springer, 2009.

[15] J.I. Goldstein et al. *Scanning Electron Microscopy and X-Ray Microanalysis*. Second. Plenum Press, 1992.

[16] J. H. Hubbell et al. “A Review,Bibliography, and Tabulation of K, L, and Higher Atomic Shell X-Ray FLuorescence Yields”. In: *J. Phys. Chem. Ref. Data* 23.2 (1994), pp. 339–364.

[17] J. H. Hubbell et al. “Erratum: “A Review, Bibliography, and Tabulation of K, L, and Higher Atomic Shell X-Ray Fluorescence Yields” [J. Phys. Chem. Ref. Data 23, 339 (1994)]”. In: *J. Phys. Chem. Ref. Data* 33.2 (2004), pp. 621–621.

[18] C. J. Powell. “Cross sections for ionization of inner-shell electrons by electrons”. In: *Rev. Mod. Phys.* 48 (1 Jan. 1976), pp. 33–47.

[19] Schreiber T.P. and A.M. Wims. “A Quantitative X-ray Microanalysis Thin Film Method Using K-, L- and M-lines”. In: *Ultramicroscopy* 6 (1981), pp. 323–334.

[20] M. Watanabe. “X-Ray Energy-Dispersive Spectrometry in Scanning Transmission Electron Microscopes”. English. In: *Scanning Transmission Electron Microscopy*. Ed. by Stephen J. Pennycook and Peter D. Nellist. Springer New York, 2011, pp. 291–351.

[21] Watanabe, Ackland, and Williams. “The effect of large solid angles of collection on quantitative X-ray microanalysis in the AEM”. In: *Journal of Microscopy* 195.1 (1999), pp. 34–43.

[22] Masashi Watanabe, Zenji Horita, and Minoru Nemotos. “Absorption correction and thickness determination using the ζ factor in quantitative X-ray microanalysis”. In: *Ultramicroscopy* 65.3 (1996), pp. 187 –198.

[23] R.A. Dragoset J. Chang A.R. Kishore S.A. Kotochigova C.T. Chantler K. Olsen and D.S. Zucker. *X-Ray Form Factor, Attenuation, and Scattering Tables*. Online. Available: http://www.nist.gov/pml/data/ffast/index.cfm, National Institute of Standards and Technology, Gaithersburg, MD. 2005.

[24] Catriona S.M. Yeoh et al. “The Dark Side of EDX Tomography: Modeling Detector Shadowing to Aid 3D Elemental Signal Analysis”. In: *Microscopy and Microanalysis* 21 (03 June 2015), pp. 759–764.

[25] I.T. Jolliffe. *Principle Component Analysis*. 2nd ed. Springer, 2002.

[26] W.H. Press et al. *Numerical Recipes in C++*. 2nd ed. Cambridge University Press, 2005.

[27] R.F. Egerton. *Electron Energy-Loss Spectroscopy in the Electron Microscope*. 3rd ed. Springer, 2011.

[28] J. S. Nilsen. “Position controlled Growth of GaAs/AlGaAs core-shell Nanowires - more uniform in their structural and optical Properties?” MA thesis. NTNU, June 2014.

[29] A. B. Mosberg. “Examining the three-dimensional structure of an AlGaAs shell on GaAs nanowires”. MA thesis. NTNU, June 2015.

[30] H. Kauko et al. “Near-surface depletion of antimony during the growth of GaAsSb and GaAs/GaAsSb nanowires”. In: *Journal of Applied Physics* 116.23, 239901 (2014).

[31] M. R. Keenan and P.G. Kotula. “Accounting for Poisson noise in multivariate analysis of ToF-SIMS spectrum images”. In: *Surf. Interface Anal.* 36 (2004), pp. 203–212.

[32] C.M. Kwei and Y.C. Li. “Electron inelastic mean free paths and surface excitation parameters for GaAs”. In: *Applied Surface Science* 238.1–4 (2004). APHYS’03 Special IssueA. Mendez-Vilas and M.L. Gonzalez-Martin, pp. 151 –154.

[33] L. Zommer et al. “Determination of the inelastic mean free path of electrons in GaAs and InP after surface cleaning by ion bombardment using elastic peak electron spectroscopy”. In: *Journal of Electron Spectroscopy and Related Phenomena* 87.3 (1998), pp. 177 –185.

[34] H. Shinotsuka et al. “Calculations of electron inelastic mean free paths. X. Data for 41 elemental solids over the 50 eV to 200keV range with the relativistic full Penn algorithm”. In: *Surface and Interface Analysis* 47.9 (2015). SIA-15-0197, pp. 871–888.

[35] T J A Slater et al. “Understanding the limitations of the Super-X energy dispersive x-ray spectrometer as a function of specimen tilt angle for tomographic data acquisition in the S/TEM”. In: *Journal of Physics: Conference Series* 522.1 (2014), p. 012025.

[36] J. Thomas and Thomas G. *Analytical Transmission Electron Microscopy: An Introduction for Operators*. Springer, 2014.

[37] *Introduction: What is EELS?* Accessed 19.06.16.

[38] D. Brandon and D. K. Wayne. *Microstructural Characterization of Materials*. 2nd ed. Wiley, 2008.

[39] Pierre Burdet et al. "A novel 3D absorption correction method for quantitative EDX-STEM tomography". In: *Ultramicroscopy* 160 (2016), pp. 118 –129.

[40] R. Brydson and N. Hondow. "Aberration-Corrected Analytical Transmission Electron Microscopy". In: John Wiley and Sons, 2011. Chap. Electron Energy Loss Spectrometry and Energy Dispersive X-ray Analysis, pp. 163–210.

[41] Stefanie Fladischer and Werner Grogger. "Quantitative {EDXS} analysis of organic materials using the ζ-factor method". In: *Ultramicroscopy* 136 (2014), pp. 26 –30.

[42] Takeshi Fujita et al. "Advanced form of ζ-factor method in analytical electron microscopy". In: *Journal of Electron Microscopy* 48.5 (1999), pp. 561–568. eprint: `http://jmicro.oxfordjournals.org/content/48/5/561.full.pdf+html`.

[43] Joseph Goldstein, David C Joy, and Alton D Romig Jr. *Principles of analytical electron microscopy*. Springer Science & Business Media, 1986.

[44] Humphreys J. Beanland R. Goodhew P.J. *Electron Microscopy and Analysis*. 3rd ed. Taylor & Francis, 2001.

[45] Suk Chung, David J. Smith, and Martha R. Mccartney. "Determination of the Inelastic Mean-Free-Path and Mean Inner Potential for AlAs and GaAs Using Off-Axis Electron Holography and Convergent Beam Electron Diffraction". English. In: *Microscopy and Microanalysis* 13.5 (Oct. 2007), pp. 329–35.

[46] H. Kauko. "Quantitative scanning transmission electron microscopy studies on heterostructured GaAs nanowires". PhD thesis. NTNU, 2013.

[47] M.R. Keenan. "Multivariate Analysis of Spectral Images composed of Count Data". In: *Techniques and Applications of Hyperspectral Image Analysis*. Ed. by H.F. Grahn and Geladi P. Wiley, 2007.

[48] C.N.R. Rao and A. Govindaraj. *Nanotubes and Nanowires*. 2nd ed. Royal Society of Chemistry, 2011.

[49] Katsunori Ohshima et al. "Determination of absolute thickness and mean free path of thin foil specimen by ζ-factor method". In: *Journal of Electron Microscopy* 53.2 (2004), pp. 137–142. eprint: `http://jmicro.oxfordjournals.org/content/53/2/137.full.pdf+html`.

[50] A.J. Garrat-Reed and D.C. Bell. *Energy-Dispersive X-Ray Analysis in the Electron Microscope*. Bios Scientific Publishers Ltd, 2003.

[51] H.D. Young and Freedman R.A. *University Physics with Modern Physics*. Vol. 2. Pearson Education Limited, 2011.

[52] E.W. Weisstein. *Maximum Likelihood*. `http://mathworld.wolfram.com/MaximumLikelihood.html`. Accessed: 13.06.16.

[53] A. C. et al. Thompson. *X-ray data booklet*. Lawrence Berkeley National Laboratory University of California Berkeley, CA 94720, Oct. 2009.

[54] G. Zschornack. "Physical Fundamentals". English. In: *Handbook of X-Ray Data*. Springer Berlin Heidelberg, 2007, pp. 9–97.

[55] W. Xu et al. "A numerical model for multiple detector energy dispersive X-ray spectroscopy in the transmission electron microscope". In: *Ultramicroscopy* 164 (2016), pp. 51 –61.

Appendix A: SCANDEM 2016

ζ-factor Tilt Dependency For Improved Quantitative Microanalysis

Andreas Garmannslund*, Julie S. Nilsen and Antonius T.J. van Helvoort

Department of Physics, NTNU, 7491 Trondheim, Norway
*E-mail: garmanns@stud.ntnu.no

Keywords: ζ-factor, GaAs, heterostructures, quantitative XEDS, nanowires

Compositional analysis with nanometre scale resolution is important for technological development. The ζ-factor method for quantitative X-ray energy-dispersive spectroscopy (XEDS) in scanning transmission electron microscopy (STEM) overcomes several limitations of the well-known Cliff-Lorimer (CL) ratio technique [1]. ζ is a proportionality factor that relates the composition to the experimental conditions. The mass-thickness is determined simultaneously with compositional quantification in the ζ-factor method. Therefore, corrections for thickness-related effects, such as absorption, can be taken into account. Another major advantage is that pure element standards can be used in the calibration of ζ-factors. Thus, the ζ-factor method offers improved quantification accuracy compared to CL.

In this study, the tilt dependency on the quantification by the ζ-factor method is investigated. Tilt series were acquired from GaAs nanowires in the $\bar{1}10$ direction and focused ion beam (FIB) made cross-sections of GaAs/AlGaAs core-shell nanowires in the 111 direction (see insert Fig.1(a)). A routine for determining the ζ-factors was developed in HyperSpy [2].The data were collected on a cold-FEG JEOL ARM2000F with a JEOL Centurio SSD (solid angle: 0.98 srad). Quantitative maps of axial GaAs/GaAsSb and radial GaAs/AlGaAs nanowire heterostructures were constructed using the ζ-factor method and compared to maps based on CL analysis. The detected X-ray signal varies as a function of specimen tilt due to absorption, channelling and detector shadowing effects [3]. The most notable effect is shadowing from the specimen holder and support grid (dashed box in Fig.1(a)). Absorption effects are negligible in these nanowires. The ζ-factor is constant for high tilt angles (10-30°) facing the detector. The refined method allows for more accurate quantitative mapping (see for example Fig.1(b)) across a large range of specimen tilts.

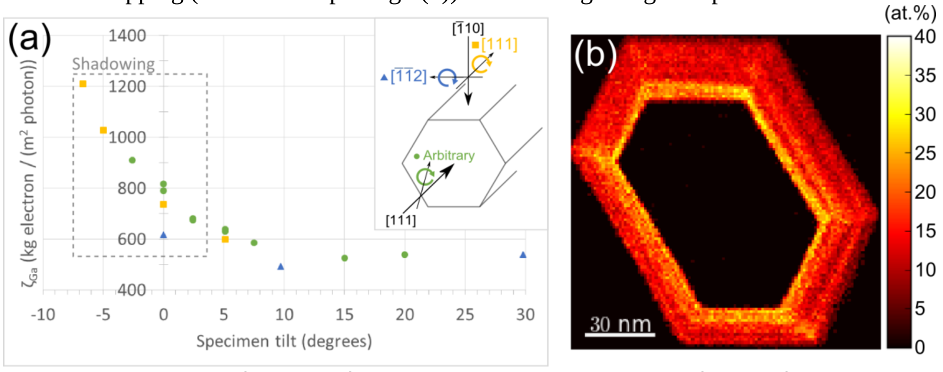


Figure 1: (a) ζ_{Ga} as a function of specimen tilt. Insert: schematic of set-up for tilt series. (b) Quantitative XEDS map based on the refined ζ-factor method showing at.% Al of a FIB made cross-section in the 111 direction of a GaAs/AlGaAs core-shell nanowire.

[1] M. Watanabe and D. Williams, *Journal of Microscopy* **221**, 89-109 (2006).
[2] F. de la Peña et al. Hyperspy 0.8.1 doi: 10.5281/zenodo.27735
[3] W. Xu et al, *Ultramicroscopy* **164**, 51-61 (2016).

ζ-factor Tilt Dependency For Improved Quantitative Energy Dispersive Spectroscopy

Andreas Garmannslund*, Julie S. Nilsen and Antonius T.J. van Helvoort
Department of Physics, NTNU, 7491 Trondheim, Norway
andreas.garmannslund@gmail.com

Summary

The tilt dependency on the quantification by the ζ-factor method is investigated by using GaAs and GaAs/AlGaAs core-shell nanowires as model systems. A routine for determining the ζ-factors was developed in the open source Python library HyperSpy. The detected X-ray signal varies as a function specimen tilt due to absorption, channelling effects and detector shadowing. The most notable effect is shadowing from the specimen holder. The ζ-factor is constant for high tilt angles (10-30°) facing the detector. The refined method allows for more accurate quantitative mapping across a large range of specimen tilts.

Theory

Compositional analysis with nanometre scale resolution is important for understanding and improving materials. The ζ-factor method for quantitative X-ray energy-dispersive spectroscopy (XEDS) in scanning transmission electron microscopy (STEM) overcomes several limitations of the commonly used Cliff-Lorimer (CL) ratio technique [1]. The advantages is due to the mass-thickness (ρt) being determined simultaneously with compositional quantification and the ζ-factors being more robust than the k-factors used in CL. ζ-factors need to be determined from a sample of well-known composition, density and thickness:

$$\zeta_A = \frac{\rho t\, C_A}{I_A}\left(\frac{I_p \tau}{e}\right)$$

where C_A is the composition of an element A in atomic (or weight) percent, I_A is the X-ray counts (intensity), I_p is the probe current, τ is the acquisition time and e is the elementary charge. Pure element samples can be used for the calibration, which is an advantage compared to CL.

ζ-factor tilt dependency

The largest variation in the experimental ζ-factors was found for low (< 5°) and negative tilt angles (Fig. 2). This is due the specimen holder and grid shadowing X-rays and thereby reducing the efficient collection angle. Channelling effects can, except in [111], contribute significantly, but only near a zone axis. Absorption effects can be ignored for thin GaAs NWs.

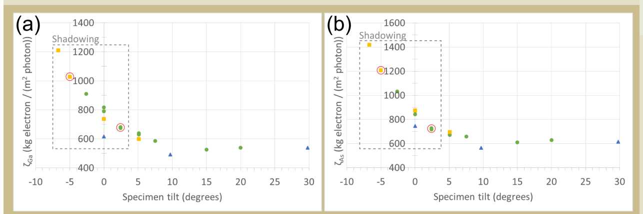


Fig. 2: (a) ζ_{Ga} and (b) ζ_{As} as a function of specimen tilt. The colours and shapes corresponds the directions and rotation axes (Fig. 1). The red circles shows data points taken on the corresponding zone axis.

Quantitative element maps

Quantitative element maps of FIB cross sections of GaAs/AlGaAs core-shell NW in [$\bar{1}\bar{1}2$] were constructed from the calibrated ζ-factors. Fig. 3 shows the transition from the (almost) pure GaAs core to a segment with AlGaAs in both core and shell.

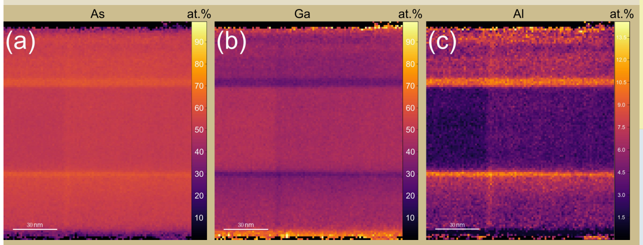


Fig. 3: Quantitative element maps of (a) As, (b) Ga and (c) Al from FIB cross sections of GaAs/AlGaAs core-shell NW in [$\bar{1}\bar{1}2$]

Experimental

XEDS data were collected using a JEOL Centurio SSD with thin window and a collection angle of 0.98 srad in a JEOL ARM200F operated in STEM mode (probe size: ~1.2 Å, convergence angle: ~27 mrad) at 200 kV. Nanowires (NW) were grown by molecular beam epitaxy. Tilt series were acquired from GaAs NWs in [$\bar{1}10$] and focused ion beam (FIB) made cross sections of GaAs/AlGaAs core-shell NWs in [111] as shown in Fig. 1. The probe current was measured in vacuum using a Gatan Quantum ER GIF. The thickness was measured using the projected width of the NW or based on EFTEM thickness maps. ζ_{Ga} and ζ_{As} were determined as a function of tilt assuming the composition ratio to be 1:1 for Ga and As. ζ_{Al} was determined from a pure Al foil. All data were processed using HyperSpy [2]. Principal component analysis (PCA) was applied to all spectra for noise reduction prior to determining the zeta-factors. For the quantitative maps, non-negative matrix factorization was applied instead of PCA. An integration window of 1.2 FWHM was used and background was removed by subtracting the mean intensity windows at each side of the targeted characteristic peak.

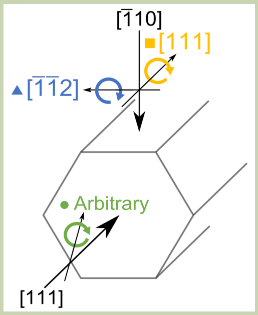


Fig. 1: Schematic of tilt series showing direction and rotation axes.

Thickness maps

Thickness maps were constructed from the mass-thickness calculated by the ζ-factor method. The density of the sample was estimated based on a harmonic mean of the pure element densities and the calculated compositions. The thickness appears to vary more than expected in a FIB lamella (Fig. 4) due to this rough estimate.

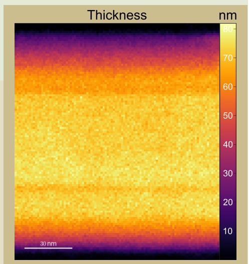


Fig. 4: Thickness map from FIB cross section of GaAs/AlGaAs core-shell NW in [$\bar{1}\bar{1}2$]

Conclusions

- The ζ-factor method for XEDS quantification has been implemented in HyperSpy, an open source Python library.
- Shadowing effects from the specimen holder and grid is the main contribution to the variation in the determined ζ-factors.
- Absorption effects can be neglected for thin specimens such as GaAs nanowires.
- For an accurate analysis it is recommended to determine the ζ-factors under the same tilt conditions as you want to use for characterization, preferably under high tilt angles (10-30°).

References

[1] M. Watanabe and D. Williams, Journal of Microscopy 221, 89-109 (2006).
[2] F. de la Peña et al. Hyperspy 0.8.1 doi: 10.5281/zenodo.27735
[3] W. Xu et al, Ultramicroscopy 164, 51-61 (2016).

Acknowledgements

We are grateful to the Nanowire Group at NTNU (Weman/Fimland) for supplying the material, the Research Council of Norway for financial support to the NORTEM project (197405) and the GRANASOL project (239206/O70), and the HyperSpy developers for the collaboration and invaluable support, advice and encouragement.

NTNU
Norwegian University of Science and Technology

HyperSpy

TEM Gemini Centre

Please note that the figures in the submitted abstract and poster have some mistakes. The correct figures are given in Figure 4.2 and Figure 4.10 for the abstract. The correct figures for the poster is given in Figure 4.2 and Figure 4.16.

Appendix B: Source code

1 Implementation of the ζ-factor method with absorption correction . . . 58
2 General code for calculating the ζ-factor 62
3 Auxiliary methods 64
4 Fitting ζ-factors 65
5 Fitting ζ-factors to the shadowing model 74
6 Determine ζ_{Ga} and ζ_{As} from tilt series of SC343 in [111] 78
7 Determine ζ_{Ga} and ζ_{As} from tilt series of SC48 in $[\bar{1}10]$ 80
8 Determine ζ_C from thin carbon foil 82
9 Determine ζ_{Ga} and ζ_{Sb} from GaSb 83
10 Determine ζ_{Al} from Al thin film 84
11 Quantification of GaAs/GaAsSb (SC48) 86
12 Quantification of linescan and close up of transition from GaAs to AlGaAs 87
13 Quantification of tip of FIB lamella of SC365 in $[\bar{1}\bar{1}2]$ 90
14 Quantification of tip of FIB lamella of SC365 in [111] 91
15 Thickness maps from EFTEM t/λ maps 93

Listing 1: Implementation of the ζ-factor method with absorption correction

```python
"""
The zeta-factor method has implemented been with optional absorption correction.
This is planned to be implemented in the HyperSpy library in the summer 2016.
The functions are implemented with this in mind, so only small changes need to be
done for fitting this into the current code framework.
"""

import hyperspy.api as hs
import scipy.constants
import numpy as np

def thinfilm_absorption_terms(mass_thickness, take_off_angle, macs_specimen):
    """
    Calculate absorption correction terms.

    Parameters
    ----------
    mass_thickness: signal
        Density-thickness map in kg/m^2
    take_off_angle: float
        X-ray take-off angle in degrees.
    macs_specimen: np.array
            Mass absorption coefficients for xray-lines in specimen.
            The first axis should be element_axis.
    """


    toa_rad = np.radians(take_off_angle)
    csc_toa = 1.0/np.sin(toa_rad)

    # Multiply macs_specimen by 0.1 to convert from cm^2/g to m^2/kg
    x = macs_specimen * 0.1 * mass_thickness * csc_toa
    x = x/(1.0 - np.exp(-(x)))

    return x

def quantification_zeta(s, intensities, zeta_factors, composition_units='atomic', convergence_precision=None):
        """
        Returns quantification and mass-thickness

        Parameters
        ----------
        s: signal
                The signal/spectrum
        intensities: numpy.array
                The intensities for each X-ray line with first axis as element axis.
        zeta_factors: list of float
                The zeta-factors for the elements in same order as intensities
        convergence_precision: float
                Absorption correction will stop when atomic percent is less than convergence_precision.
                If None, no absorption correction will not be performed.
        """

        dose = s._get_dose('zeta')
        ints_data = hs.stack(intensities).data

        composition = hs.stack(intensities) # signal list

        if convergence_precision: #absorption correction
                lines = [i.metadata.Sample.xray_lines[0] for i in intensities]
                elements = [l.split('_')[0] for l in lines]

                take_off_angle = s.get_take_off_angle()

                macs_line = []
                for line in lines:
                        macs_line.append([hs.material.mass_absorption_coefficient(el, line) for el in elements])

                abs_corr = None
                comp_old = np.zeros_like(ints_data) # np.array

                iteration = 1
                cp = convergence_precision/100
                while(True): # todo -> check for convergence
                        comp, pt = _quantification_zeta(ints_data, zeta_factors, dose, abs_corr)

                        if np.max(comp-comp_old) < cp:
                                print("Solution converged to less than", convergence_precision, "weight percent, after", iteration, "iterations.")
                                break
                        elif iteration >= 100:
                                import warnings
                                warnings.warn("Solution did not converge after ", iteration, " iterations!")
                                break

```

```python
            macs_specimen = hs.material.mass_absorption_mixture(comp, elements, lines)

            abs_corr = thinfilm_absorption_terms(pt, take_off_angle, macs_specimen)
            comp_old = comp

            iteration += 1
    else:
        comp, pt = _quantification_zeta(ints_data, zeta_factors, dose)

    composition.data = comp * 100
    composition = composition.split()

    # Convert to atomic percent
    if composition_units == 'atomic':
        composition = hs.material.weight_to_atomic(composition)

    for c in composition:
        md = c.metadata
        element, line = md.Sample.xray_lines[0].split('_')
        md.General.title = 'Atomic percent of ' + element

    mass_thickness = intensities[0].deepcopy()
    mass_thickness.data = pt

    return composition, mass_thickness

def _quantification_zeta(intensities, zeta_factors, dose, absorption_correction=None):
    """
    Returns quantification and mass-thickness

    Parameters
    ----------
    intensities: numpy.array
        The intensities for each X-ray line with first axis as element axis.
    zeta_factors: list of float
        The zeta-factors for the elements in same order as intensities
    dose: float
        Total electron dose given by i*t*N, where i is the beam_current, t is the acquisition time,
        and N the number of electrons per unit electric charge (1/e).
    """
    if absorption_correction is None:
        absorption_correction = np.ones_like(intensities, dtype='float')

    comp = np.zeros_like(intensities, dtype='float')

    zi_list = [i * z * a for i,z,a in zip(intensities, zeta_factors, absorption_correction)]
    zi_sum = sum(zi_list)
    comp = np.asarray([zi / zi_sum for zi in zi_list])
    comp = np.nan_to_num(comp) # if divided by zero, set to zero
    pt = zi_sum / dose

    return comp, pt

# Assumes input in atomic percent so converts it
def get_thickness_map(c, pt, p='auto', composition_units='atomic'):
    if p == 'auto':
        if composition_units == 'atomic':
```

```
                        p =
↪   hs.material.density_of_mixture(hs.material.atomic_to_weight(c)) * 1e3
                else:
                        p = hs.material.density_of_mixture(c) * 1e3
        thickness = (pt*1e9/p)
        thickness *= (~np.isinf(thickness.data))
        thickness.data = np.nan_to_num(thickness.data)
        thickness.metadata.General.title = 'Thickness (nm)'
        return thickness
```

Listing 2: General code for calculating the ζ-factor

```python
import hyperspy.api as hs
import numpy as np
import scipy

from hyperspy import utils

def determine_zeta_factor(s, intensities, composition, thickness, density):
        """
        Determine the zeta-factors from a sample with known mass-thickness and composition.

        Parameters
        ----------
        s: signal
                Temporary, replace later by self when moved to eds_tem.py
        intensities: list of signal
                The intensity for each X-ray line
        composition: list of float
                Composition of the elements in the same order as intensities.
        elements:
                Elements in same order as composition.
        mass_thickness: float or signal
                Mass-thickness for the sample. If signal, must be same shape as intensities.
        print_average:
                If True, the function prints the average zeta-factor for each element.

        Returns
        -------
        A signal in the same shape as intensities, giving the zeta-factors for each X-ray line.
        """

        zfactors = utils.stack(intensities)
        zfactors.data = _determine_zeta_factor(zfactors.data, composition, s._get_dose('zeta'), thickness*1e-9*density)
        zfactors = zfactors.split()

        xray_lines = [xray.metadata.Sample.xray_lines[0] for xray in intensities]
        return zfactors
        for i, line in enumerate(xray_lines):
            zfactors[i].metadata.General.title = "Zeta-factor for " + line

        return zfactors

def _determine_zeta_factor(intensities, composition, dose, mass_thickness):
        """
        Determine zeta-factors

        Parameters
        ----------
        intensities: numpy.array
                The intensities for each X-ray line. The first axis should be the element axis.
        composition: list of float
                Composition of the elements given in weight-percent in the same order as intensities.
        dose: float
```

```
                Total electron dose given by i*t*N, where i is the beam_current, t is
↪   the acquisition time,
                and N the number of electrons per unit electric charge (1/e).

        Returns
        -------
        A numpy.array containing the zeta_factors with the same shape as intensities
        """

        zfactors = np.zeros_like(intensities, dtype='float')

        for i, (intensity, comp) in enumerate(zip(intensities, composition)):
                zfactors[i] = (dose * mass_thickness * comp) / intensity

        return zfactors
```

Listing 3: Auxiliary methods

```
import hyperspy.api as hs

""" Some general functions that are shared among many of the scripts """
def load_pca(filepath, pca_comps):
        s = hs.load(filepath)
        s.change_dtype('float')
        s.decomposition()
        return s.get_decomposition_model(pca_comps)

def load_nmf(filepath, nmf_comps):
        s = hs.load(filepath)
        s.change_dtype('float')
        s.decomposition(algorithm='nmf', output_dimension=nmf_comps)
        return s.get_decomposition_model()

def apply_mask(signal, s_ints, threshold_counts):
        mask = [si > threshold_counts for si in s_ints]
        signal = [s * m for (s,m) in zip(signal,mask)]
        for s in signal:
                s.data = np.nan_to_num(s.data)
        return signal

def um_to_nm_scale(s):
        s.axes_manager['x'].scale *= 1000
        s.axes_manager['x'].units = 'nm'
        s.axes_manager['y'].scale *= 1000
        s.axes_manager['y'].units = 'nm'
```

Listing 4: Fitting ζ-factors

```
"""
This script contains a lot of useful functions:
 -         Calculating the ionization cross section using parameters from
         Schreiber and Williams or Powell (see thesis for references).
 -        Calculate the theoretical zeta-factor and k-factor.
 -        Least-Square fitting of zeta-factors by fitting scaling factors
         to the ionization cross section, Au contact layer and Si dead layer
         of the detector.
 -         Plots the fitted model and detector efficiency as function of
         energy and element.
 -        Plots the ionization cross section as a function of overvoltage.
 """
from hyperspy.misc.elements import elements as elements_db
from hyperspy.misc.eds.utils import _get_element_and_line
from hyperspy.misc.material import mass_absorption_coefficient

import numpy as np
import scipy
import scipy.constants
from scipy.constants import e as elementary_charge
from scipy.constants import pi as PI
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams.update({'font.size': 14})

# Fluorescence yield from Hubbel et al. (See Reference section of thesis)
# Fluorescence yields (K (3 < Z < 100), L (11 < Z < 100), 19 < Z < 100)
# The given fluorescence yield for L- and M-lines are the average of all subshells
fluorescence_yield_K_lines = [2.93E-04, 6.93E-04, 0.001409, 0.002575, 0.004349,
 ↪   0.006909, 0.01045, 0.01519, 0.02133, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.12,
 ↪   0.14, 0.17, 0.20, 0.23, 0.26, 0.29, 0.32, 0.35, 0.39, 0.42, 0.45, 0.49, 0.52,
 ↪   0.55, 0.57, 0.60, 0.63, 0.65, 0.67, 0.70, 0.72, 0.73, 0.75, 0.77, 0.78, 0.80,
 ↪   0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.88, 0.88, 0.89, 0.90, 0.90,
 ↪   0.91, 0.91, 0.92, 0.92, 0.93, 0.93, 0.93, 0.93, 0.94, 0.94, 0.94, 0.94, 0.95,
 ↪   0.95, 0.95, 0.95, 0.95, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96, 0.96,
 ↪   0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.97,
 ↪   0.97, 0.97, 0.97, 0.97]
fluorescence_yield_L_lines = [2.17E-04, 3.04E-04, 4.15E-04, 5.53E-04, 7.24E-04,
 ↪   9.30E-04, 1.18E-03, 1.47E-03, 1.81E-03, 2.21E-03, 2.68E-03, 3.21E-03, 3.81E-03,
 ↪   4.50E-03, 5.27E-03, 6.14E-03, 7.11E-03, 8.19E-03, 9.39E-03, 1.07E-02, 1.22E-02,
 ↪   1.38E-02, 1.55E-02, 1.74E-02, 1.95E-02, 2.18E-02, 2.42E-02, 2.63E-02, 2.85E-02,
 ↪   3.09E-02, 3.35E-02, 3.63E-02, 3.93E-02, 4.25E-02, 4.59E-02, 4.95E-02, 5.34E-02,
 ↪   5.75E-02, 6.18E-02, 6.65E-02, 7.14E-02, 7.65E-02, 8.20E-02, 8.77E-02, 9.38E-02,
 ↪   1.00E-01, 1.07E-01, 1.14E-01, 1.21E-01, 1.29E-01, 1.37E-01, 1.45E-01, 1.53E-01,
 ↪   1.63E-01, 1.72E-01, 1.82E-01, 1.92E-01, 2.02E-01, 2.12E-01, 2.23E-01, 2.34E-01,
 ↪   2.45E-01, 2.57E-01, 2.69E-01, 2.81E-01, 2.93E-01, 3.05E-01, 3.18E-01, 3.31E-01,
 ↪   3.43E-01, 3.56E-01, 3.69E-01, 3.82E-01, 3.95E-01, 4.09E-01, 4.22E-01, 4.35E-01,
 ↪   4.48E-01, 4.61E-01, 4.74E-01, 4.86E-01, 4.99E-01, 5.11E-01, 5.24E-01, 5.36E-01,
 ↪   5.48E-01, 5.60E-01, 5.72E-01, 5.83E-01, 5.95E-01]
```

```
fluorescence_yield_M_lines = [1.67E-06, 3.10E-06, 5.28E-06, 8.46E-06, 1.29E-05,
↪   1.89E-05, 2.67E-05, 3.68E-05, 4.96E-05, 6.53E-05, 8.45E-05, 1.08E-04, 1.35E-04,
↪   1.68E-04, 2.06E-04, 2.51E-04, 3.02E-04, 3.61E-04, 4.28E-04, 5.04E-04, 5.90E-04,
↪   6.86E-04, 7.93E-04, 9.12E-04, 1.04E-03, 1.19E-03, 1.35E-03, 1.53E-03, 1.72E-03,
↪   1.93E-03, 2.17E-03, 2.42E-03, 2.69E-03, 2.98E-03, 3.30E-03, 3.65E-03, 4.01E-03,
↪   4.41E-03, 4.84E-03, 5.29E-03, 5.78E-03, 6.29E-03, 6.85E-03, 7.44E-03, 8.06E-03,
↪   8.73E-03, 9.43E-03, 1.02E-02, 1.10E-02, 1.18E-02, 1.27E-02, 1.36E-02, 1.46E-02,
↪   1.56E-02, 1.67E-02, 1.79E-02, 1.91E-02, 2.03E-02, 2.16E-02, 2.30E-02, 2.45E-02,
↪   2.60E-02, 2.75E-02, 2.92E-02, 3.10E-02, 3.28E-02, 3.47E-02, 3.66E-02, 3.87E-02,
↪   4.08E-02, 4.30E-02, 4.53E-02, 4.77E-02, 5.02E-02, 5.28E-02, 5.55E-02, 5.83E-02,
↪   6.12E-02, 6.42E-02, 6.73E-02, 7.06E-02, 7.39E-02]

def get_fluorescence_yield(xray_line):
        """ Returns the fluorescence yield for a given X-ray line """
        element, line = xray_line.split('_')
        family = line[0]

        el_data = elements_db[element]
        z = el_data['General_properties']['Z']

        if z > 100:
                raise ValueError("No data for Z > 100")

        if family == 'K':
                if z < 3:
                        raise ValueError("Error: No data for Z < 3 for K-lines")
                return fluorescence_yield_K_lines[z-3]
        elif family == 'L':
                if z < 11:
                        raise ValueError("Error: No data for Z < 11 for L-lines")
                return fluorescence_yield_L_lines[z-11]
        elif family == 'M':
                if z < 19:
                        raise ValueError("Error: No data for Z < 19 for M-lines")
                return fluorescence_yield_M_lines[z-19]

        raise ValueError('Not a valid X-ray line!')

def schreiber_wims_cross_section(z, line_family):
        """ Returns parameters for the inner-shell ionization cross section
                as described by Schreiber and Wims """
        ln_z = np.log(z)

        if line_family == 'K':
                if z <= 30:
                        b = 8.874 - (8.158 * ln_z) + (2.9055 * ((ln_z)**2)) - (0.35778
↪   * (ln_z**3))
                else:
                        b = 0.661

                d = 1.0667 - (0.00476 * z)
        elif line_family == 'L':
                b = 0.2704 + (0.00726 * ((ln_z)**3))
                d = 1
        elif line_family == 'M':
                b = 11.33 - 2.43 * ln_z
                d = 1
        else:
                raise Exception("Invalid line")

        c = 1

```

```python
        return b,c,d

def powell_cross_section(line_family):
        """ Returns parameters for the inner-shell ionization cross section
                as described by Powell """
        if line_family == 'K':
                b = 0.9
                c = 0.70
        elif line_family == 'L':
                # It was not clear from Powell(1976) what the recommended values for L-lines were,
                # so uses L-lines from Powell as given in
                # "Analytical Transmission Electron Microscopy: An Introduction for Operators (2014)
                c = 0.59
                b = 0.63
        else:
                raise Exception('Powell cross section only accepts K- or L-lines')

        d = 1
        return b,c,d


def ionization_cross_section(xray_line, beam_energy, method, normalize=False):
        """
        Calculates the inner-shell ionization cross section in m^2
        If normalize is True, units are m^2 * eV^2

        Parameters
        ----------
        xray_line: str
                The name of the X-ray line, i.e. Al_Ka
        beam_energy: float
                Beam energy in keV
        method: str
                'sw': Schreiber-Wims
                'powell': Powell
        normalize: bool
                If True: The cross section is normalized by ionization energy
        """
        element, line = xray_line.split('_')
        line_family = line[0]
        ionization_energy = elements_db[element]['Atomic_properties']['Xray_lines'][line]['energy (keV)']
        z = elements_db[element]['General_properties']['Z']

        # Get number of electrons in shell
        if line_family == 'K':
                ns = 2
        elif line_family == 'L':
                ns = 8
        elif line_family == 'M':
                ns = 18
        else:
                raise Exception("Invalid line family: Need to be K-, L- or M-line")

        # Get ionization cross section
        if method == 'sw':
                b,c,d = schreiber_wims_cross_section(z, line_family)
        elif method =='powell':
                b,c,d = powell_cross_section(line_family)

        U = beam_energy/ionization_energy # overvoltage

        num = 6.4924e-24 * ns * b * np.log(c * U)

        if normalize:
                den = U**d
        else:
                den = (ionization_energy**2) * (U**d)

        Q = num / den
        return Q


def plot_Q_vs_overvoltage(xray_line, overvoltage_bounds, normalize=False, method='sw', format='r-', new_figure=True):
        element, line = xray_line.split('_')
        line_energy = elements_db[element]['Atomic_properties']['Xray_lines'][line]['energy (keV)']
        z = elements_db[element]['General_properties']['Z']

        beam_min = overvoltage_bounds[0] * line_energy
        beam_max = overvoltage_bounds[1] * line_energy
        print("beam_min:", beam_min, "beam_max:", beam_max)

        U = []
        Q = []
        for beam_energy in np.linspace(beam_min, beam_max, (beam_max-beam_min+1) * 50):
                q = ionization_cross_section(xray_line, beam_energy, method, normalize)

                Q.append(q)
                U.append(beam_energy / line_energy)

        if new_figure:
                plt.figure()
        plt.plot(U,Q, format)
        plt.xlabel('Overvoltage')

        if normalize:
                plt.ylabel('Ionization cross section (m^2) * (eV^2)')
        else:
                plt.ylabel('Ionization cross section (m^2)')

# Plots inner-shell ionization cross sections for both implemented methods.
overvoltage_bounds = (1,30)
plot_Q_vs_overvoltage('As_Ka', overvoltage_bounds, normalize=False, new_figure=True, method='sw', format='b-')
plot_Q_vs_overvoltage('As_Ka', overvoltage_bounds, normalize=False, new_figure=False, method='powell', format='g--')
plt.ylim(0)


# Detector configuration
# Thicknesses in cm
detector_data = {
        'collection_angle': 0.98,        # srad
        't_polymer': 0, # Don't know the composition of this, therefore this term is ignored (assume windowless)
        't_Au': 1e-6, # From metadata - from m to cm
        't_Si': 1e-5, # From metadata from m to cm
        't_Si_2': 0.1 # From Williams & Carter
}
```

```python
def calculate_theoretical_zeta_factor(xray_line, detector, beam_energy, cross_section='sw'):
        """
        Parameters
        ----------
        xray_line: str
                Name of the X-ray line
        detector: dictionary
                Dictionary containing the window thickness of
                the polymer window, gold layer, silicon dead layer
                and high energy layer.
        beam_energy: float
                Beam energy in keV
        cross_section: 'sw' or 'powell'
                If 'sw' use the ionization cross section defined by Schreiber and Wims (default)
                If 'powell' use the ionization cross section defined by Powell
        """

        element, line = xray_line.split('_')
        el_data = elements_db[element]
        #z = el_data['General_properties']['Z']

        atomic_weight = el_data['General_properties']['atomic_weight'] # Check units of this - should be g/mol

        a = el_data['Atomic_properties']['Xray_lines'][line]['weight']
        w = get_fluorescence_yield(xray_line)
        collection_angle_4pi = detector['collection_angle'] / (4 * scipy.pi)

        q = ionization_cross_section(xray_line, beam_energy, cross_section)

        detector_efficiency = estimate_detector_efficiency(xray_line, detector['t_polymer'], detector['t_Au'], detector['t_Si'], detector['t_Si_2'])

        return atomic_weight / (scipy.constants.N_A * q * w * a * collection_angle_4pi * detector_efficiency)

def _get_element_density(element):
        """ Returns density of the given element in g/cm^3 """
        return elements_db[element]['Physical_properties']['density (g/cm^3)']

def estimate_detector_efficiency(xray_line, t_polymer, t_Au, t_Si, t_Si_2):
        """ Estimates the detector efficiency. Thickness given in cm. """
        polymer_window = 1 # Don't know the composition of this, but it's thin, so assume 0 - can be improved if window is known
        Au_contact = np.exp(-mass_absorption_coefficient('Au', xray_line) * _get_element_density('Au') * t_Au)
        Si_dead_layer = np.exp(-mass_absorption_coefficient('Si', xray_line) * _get_element_density('Si') * t_Si)
        high_energy_layer = 1 - np.exp(-mass_absorption_coefficient('Si', xray_line) * _get_element_density('Si') * t_Si_2)

        return polymer_window * Au_contact * Si_dead_layer * high_energy_layer

# Reference is usually 'Si_Ka' or 'Fe_Ka'
def calculate_theoretical_k_factor(xray_lines, detector, beam_energy):
        """
        Calculates the theoretical k-factor.

        Parameters
        ----------
        xray_lines: list of strings
                The two X-ray lines for the k-factor
        detector: dictionary
                Thicknesses of detector windows/layers and collection angle.
        beam_energy: float
                The energy of the beam
        """
        atomic_weight = []
        res = []
        for xl in xray_lines:
                element, line = xl.split('_')

                el_data = elements_db[element]
                z = el_data['General_properties']['Z']

                atomic_weight.append(el_data['General_properties']['atomic_weight'])
                w = get_fluorescence_yield(xl)
                a = el_data['Atomic_properties']['Xray_lines'][line]['weight']
                q = ionization_cross_section(xl, beam_energy, 'sw')
                eff = estimate_detector_efficiency(xl, detector['t_polymer'], detector['t_Au'], detector['t_Si'], detector['t_Si_2'])

                res.append(q * w * a * eff)

        return (res[1] / res[0]) * (atomic_weight[0]/atomic_weight[1])

def _plot_model(zeta_factors, line_energies, xray_lines, detector_efficiency,offset_atomic_number):
        num = len(zeta_factors)
        # Zeta-factors by energy
        plt.figure()
        plt.plot(line_energies, zeta_factors, 'ro')
        plt.plot(known_energies, zeta_factors_known, 'ys')
        plt.ylabel("Zeta-factors")
        plt.ylabel("Zeta-factors")
        plt.xlabel("Energy (keV)")

        # Detector efficiency by energy
        plt.figure()
        plt.plot(line_energies, detector_efficiency, 'g*--')
        plt.title("Detector efficiency")
        plt.xlabel("Energy (keV)")

        # Plot by element
        elements = [i[0] for i in xray_lines][:num]
        atomic_numbers = np.arange(offset_atomic_number, len(xray_lines)+offset_atomic_number-1)[:num]

        # Atomic number help
        PLOT_SPACE = 1
        elements_plot_labels = [el for (i, el) in enumerate(elements) if not i%PLOT_SPACE]
        atomic_numbers_plot_labels = [a for (i, a) in enumerate(atomic_numbers) if not i%PLOT_SPACE]

        # Zeta-factor by atomic number/element
        plt.figure()
        plt.plot(atomic_numbers, zeta_factors, 'ro')
        plt.plot(known_atomic_numbers, zeta_factors_known, 'ys')
        plt.ylabel("Zeta-factors")
        plt.ylabel("Zeta-factors")
        plt.xlabel("Element")

        plt.xticks(atomic_numbers_plot_labels, elements_plot_labels)
        plt.grid(axis='x')

        # Detector efficiency by atomic number
        plt.figure()
        plt.plot(atomic_numbers, detector_efficiency, 'g*--')
        plt.xticks(atomic_numbers_plot_labels, elements_plot_labels)
        plt.title("Detector efficiency")
        plt.xlabel("Element")
        plt.grid(axis='x')


def calculate_and_plot_model_K_lines(s, detector, known_energies, zeta_factors_known,
 ↪   plot=True):
        """
        Returns a list of fitted zeta-factors for
        all elements from C to U.

        Parameters
        ----------
        s: float
                scaling factor for cross-section
        detector: dictionary
                Detector configuration: thickness of windows/layers and collection
 ↪   angle.
        known_energies: list of floats
                Energies in (keV) of the X-ray lines used for fitting
        zeta_factors_known: list of float
                The zeta-factors of the X-ray lines used for fitting.
        plot: bool
                If True, plots the model

        """
        xray_lines_K_alpha = [['C', 'C_Ka'], ['N', 'N_Ka'],['O', 'O_Ka'],['F',
 ↪   'F_Ka'],['Ne', 'Ne_Ka'],['Na', 'Na_Ka'],['Mg', 'Mg_Ka'],['Al', 'Al_Ka'],['Si',
 ↪   'Si_Ka'],['P', 'P_Ka'],['S', 'S_Ka'],['Cl', 'Cl_Ka'],['Ar', 'Ar_Ka'],['K',
 ↪   'K_Ka'],['Ca', 'Ca_Ka'],['Sc', 'Sc_Ka'],['Ti', 'Ti_Ka'],['V', 'V_Ka'],['Cr',
 ↪   'Cr_Ka'],['Mn', 'Mn_Ka'],['Fe', 'Fe_Ka'],['Co', 'Co_Ka'],['Ni', 'Ni_Ka'],['Cu',
 ↪   'Cu_Ka'],['Zn', 'Zn_Ka'],['Ga', 'Ga_Ka'],['Ge', 'Ge_Ka'],['As', 'As_Ka'],['Se',
 ↪   'Se_Ka'],['Br', 'Br_Ka'],['Kr', 'Kr_Ka'],['Rb', 'Rb_Ka'],['Sr', 'Sr_Ka'],['Y',
 ↪   'Y_Ka'],['Zr', 'Zr_Ka'],['Nb', 'Nb_Ka'],['Mo', 'Mo_Ka'],['Tc', 'Tc_Ka'],['Ru',
 ↪   'Ru_Ka'],['Rh', 'Rh_Ka'],['Pd', 'Pd_Ka'],['Ag', 'Ag_Ka'],['Cd', 'Cd_Ka'],['In',
 ↪   'In_Ka'],['Sn', 'Sn_Ka'],['Sb', 'Sb_Ka'],['Te', 'Te_Ka'],['I', 'I_Ka'],['Xe',
 ↪   'Xe_Ka'],['Cs', 'Cs_Ka'],['Ba', 'Ba_Ka'],['La', 'La_Ka'],['Ce', 'Ce_Ka'],['Pr',
 ↪   'Pr_Ka'],['Nd', 'Nd_Ka'],['Pm', 'Pm_Ka'],['Sm', 'Sm_Ka'],['Eu', 'Eu_Ka'],['Gd',
 ↪   'Gd_Ka'],['Tb', 'Tb_Ka'],['Dy', 'Dy_Ka'],['Ho', 'Ho_Ka'],['Er', 'Er_Ka'],['Tm',
 ↪   'Tm_Ka'],['Yb', 'Yb_Ka'],['Lu', 'Lu_Ka'],['Hf', 'Hf_Ka'],['Ta', 'Ta_Ka'],['W',
 ↪   'W_Ka'],['Re', 'Re_Ka'],['Os', 'Os_Ka'],['Ir', 'Ir_Ka'],['Pt', 'Pt_Ka'],['Au',
 ↪   'Au_Ka'],['Hg', 'Hg_Ka'],['Tl', 'Tl_Ka'],['Pb', 'Pb_Ka'],['Bi', 'Bi_Ka'],['Po',
 ↪   'Po_Ka'],['At', 'At_Ka'],['Rn', 'Rn_Ka'],['Fr', 'Fr_Ka'],['Ra', 'Ra_Ka'],['Ac',
 ↪   'Ac_Ka'],['Th', 'Th_Ka'],['Pa', 'Pa_Ka'],['U', 'U_Ka']]
        zeta_factors = []
        line_energies = []
        detector_efficiency = []

        for element_line in xray_lines_K_alpha:
                energy =
 ↪   elements_db[element_line[0]]['Atomic_properties']['Xray_lines']['Ka']['energy
 ↪   (keV)']
                if energy > 20:
                        continue
                line_energies.append(energy)
                zeta_factors.append(calculate_theoretical_zeta_factor(element_line[1],
 ↪  detector, 200)/s)
                detector_efficiency.append(estimate_detector_efficiency(element_line[1],
 ↪  detector['t_polymer'], detector['t_Au'], detector['t_Si'], detector['t_Si_2']))

        if plot:
                _plot_model(zeta_factors, line_energies, xray_lines_K_alpha,
 ↪  detector_efficiency, 6)

        return zeta_factors, line_energies

def zeta_factor_model(x, s, s_Si, s_Au):
        det = detector_data.copy()
        det['t_Si'] *= s_Si
        det['t_Au'] *= s_Au
        return [calculate_theoretical_zeta_factor(z, det, 200)/s for z in x]

def cal_chi_sq(xray_lines, zexp, zerr, det, s):
        chi_sq = 0
        for (xl, z, err) in zip(xray_lines,zexp, zerr):
                zmodel = calculate_theoretical_zeta_factor(xl, det, 200)/s
                temp = (z-zmodel)/err
                chi_sq += temp**2
        return chi_sq

def fit_model_zeta_factor(xray_lines, zexp, zerr, p0=None):
        """
                Returns:
                s: The scaling factor for the ionization cross section.
                det_adj: The adjusted detector configuration
                chi_sq: The chi-squared value after the fitting
                cov: The covariance matrix after the fitting.
        """
        import scipy.optimize as optim
        res, cov = optim.curve_fit(zeta_factor_model, xdata=xray_lines, ydata=zexp,
 ↪  p0=None, sigma=zerr, bounds=([0, 0, 0],np.inf), method='trf')
        s = res[0]  # scaling factor for ionization cross-section
        s_Si = res[1]
        s_Au = res[2]
        det_adj = detector_data.copy()
        det_adj['t_Si'] *= s_Si
        det_adj['t_Au'] *= s_Au
        chi_sq = cal_chi_sq(xray_lines, zexp, zerr, det_adj, s)

        return s, det_adj, chi_sq, cov, res

# Experimental values for plot
known_energies = [0.2774, 1.4865, 10.5436, 9.2517] # Al, As, Ga
known_atomic_numbers = [6, 13, 33, 31]

zeta_As = 902 # average 0 degrees
zeta_Ga = 747 # average 0 degrees
zeta_As = 859 # average, 0 degrees
zeta_Ga = 699 # average 0 degrees
zeta_Al = 278 # 0 degrees
zeta_C = 885 # estimated from effective area
zeta_factors_known = [zeta_C, zeta_Al, zeta_As, zeta_Ga]
zerr = [z * 0.10 for z in zeta_factors_known] # assume 10 % error
s, det_adj, chi_sq, cov, res = fit_model_zeta_factor(['C_Ka', 'Al_Ka', 'As_Ka',
 ↪  'Ga_Ka'], zeta_factors_known, zerr)
```

```
model_zeta_factors, line_energies = calculate_and_plot_model_K_lines(s, det_adj,
 ↪   known_energies, zeta_factors_known)

def test_k_zeta_same_val(detector=detector_data, beam_energy=200):
        """ Running this will give an error if there is any inconsistency
                between the theoretical calculated zeta-factor and k-factor """
        k_AlSi = calculate_theoretical_k_factor(['Al_Ka', 'Si_Ka'], detector_data,
 ↪   beam_energy)
        zeta_Al = calculate_theoretical_zeta_factor('Al_Ka', detector_data,
 ↪   beam_energy)
        zeta_Si = calculate_theoretical_zeta_factor('Si_Ka', detector_data,
 ↪   beam_energy)
        k_AlSi_from_zeta = zeta_Al/zeta_Si

        np.testing.assert_allclose(k_AlSi, k_AlSi_from_zeta)

test_k_zeta_same_val() # Should not do anything, unless error
```

Listing 5: Fitting ζ-factors to the shadowing model

```python
%matplotlib qt4
import matplotlib.pyplot as plt
import numpy as np
import scipy.optimize as optim

def active_area(h, shape, b):
        r = (np.sqrt(100)/np.pi) #* 1e-3
        if (h <= 0):
                return 1
        elif (((h > r) and shape == 'circle') or (shape=='rect' and (h >= b))):
                return 0
        else:
                if shape == 'rect':
                        A_full = 100
                        A = h * b
                else:
                        A = (r**2) * np.arccos((r-h)/r)-((r-h) * np.sqrt(2*r*h - (h**2)))
                        A_full = (np.pi * r**2)

        return (1-(A/A_full))

def get_height(tilt, tu, td, te, delta, d, tl):
        tc = ((tu-tl)-tilt-td)
        a = d * np.sin(np.radians(tc))
        b = np.sin(np.radians(90-td+delta-tc))
        c = np.cos(np.radians(te-td))
        return a/(b*c)

def shadow_model_rect(tilts, tu, td, te, delta, d, tl, b):
        return shadow_model(tilts, tu, td, te, delta, d, 'rect', tl, b)

def shadow_model_circle(tilts, tu, td, te, delta, d, tl, b):
        return shadow_model(tilts, tu, td, te, delta, d, 'circle', tl, b)

def shadow_model(tilts, tu, td, te, delta, d, shape, tl, b):
        """
        Parameters
        ----------
        tilts: array
                Tilts of measured values
        tu:        float
                The upper shadow angle
        td:        float
                Angle between specimen and the lower part of the detector
        te: float
                Elevation angle: From specimen to centre of detector
        delta: float
                Tilt of the detector from the axis perpendicular to the specimen stage at 0 degrees tilt
        tl: float
                The lower shadow angle
        shape: 'rect' or 'circle'
                Shape of detector, used to calculate active area.
        b: float
                The height (or width) of the detector if 'rectangular'

        """
        return np.array([active_area(get_height(t, tu, td, te, delta,d, tl), shape, b)
↪   for t in tilts])

def fit_and_plot_both(tilts, zeta_norm):
        models = []
        models.append(fit_and_plot_model(tilts, zeta_norm, 'rect'))
        models.append(fit_and_plot_model(tilts, zeta_norm, 'circle'))
        return models

def fit_and_plot_model(tilts, zeta_norm, shape, tu='auto', td='auto', te='auto',
↪   delta='auto', d='auto', tl='auto', b='auto'):
        """
        Parameters
        ----------
        tilts: list of floats
                Tilts of measured values
        zeta_norm: list of floats
                The measured normalized zeta-factors

        tu:        tuple of floats
                Bounds on the upper shadow angle
        td:        tuple of floats
                Bounds on lower detector angle
        te: tuple of floats
                Bounds on elevation angle
        delta: tuple of floats
                Bounds on detector tilt
        tl: tuple of floats
                Bounds on the lower shadow angle
        b: tuple of floats
                Height of rectangular detector
        """
        # Default parameters
        bounds = ([0, 0, 10, 0, 1, -50, 0], [50, 50, 30, 80, 100, 50, 100])
        if tu != 'auto':
                bounds[0][0] = tu[0]
                bounds[1][0] = tu[1]
        if td != 'auto':
                bounds[0][1] = td[0]
                bounds[1][1] = td[1]
        if te != 'auto':
                bounds[0][2] = te[0]
                bounds[1][2] = te[1]
        if delta != 'auto':
                bounds[0][3] = delta[0]
                bounds[1][3] = delta[1]
        if d != 'auto':
                bounds[0][4] = d[0]
                bounds[1][4] = d[1]
        if tl != 'auto':
                bounds[0][5] = tl[0]
                bounds[1][5] = tl[1]
        if b != 'auto':
                bounds[0][6] = b[0]
                bounds[1][6] = b[1]

        plt.figure()
        if shape == 'rect':
                res, cov = optim.curve_fit(shadow_model_rect, xdata=tilts,
↪   ydata=zeta_norm, bounds=bounds, method='trf')
                plt.title('Rectangular detector')
        else:
                res, cov = optim.curve_fit(shadow_model_circle, xdata=tilts, ydata=zeta_norm, bounds=bounds, method='trf')
                plt.title('Circular detector')

        x = np.linspace(-15, 30)
        y = np.array(shadow_model(x, res[0], res[1], res[2], res[3], res[4], shape, res[5], res[6]))
        plt.plot(x,y)
        plt.plot(tilts, zeta_norm, 'ro')

        model = shadow_model(tilts, res[0], res[1], res[2], res[3], res[4], shape, res[5], res[6])
        chi_sq = 0
        for i,z in enumerate(zeta_norm):
                diff = z - model[i]
                chi_sq += (diff * diff)
        print("Chi_square:", chi_sq)

        print("Upper shadow angle:", res[0], ", lower:", res[5], "range:", res[0]-res[5])
        print("Lower detector angle:", res[1])
        print("Elevation angle:", res[2])
        print("Delta:", res[3])
        print("Distance:", res[4], "mm")
        if shape == 'rect':
                print("Detector height X width:", res[6], "mm X", 100/res[6], " mm =", res[6]/25.4, "inches X", 100/res[6], "inches.")

        return res

def plot_GaAs(model_params, shape, zeta_As_ref=698, zeta_Ga_ref=570):
        x = np.linspace(-10, 40)
        effective_area = shadow_model(x, model_params[0], model_params[1], model_params[2], model_params[3], model_params[4], shape, model_params[5], model_params[6])

        plt.figure()
        if shape == 'rect':
                plt.title('Rectangular detector')
        else:
                plt.title('Circular detector')

        tilts_Ga_measured = [-6.7,-5,-2.6,0,0,0,0,2.4,2.4,5.1,5.1,5.1,7.5,9.7,15,20,29.8]
        zeta_Ga_measured = [1123,958,931,809,837,651,691,699,690,645,651,565,599,520,535,545,570]

        zeta_Ga = (zeta_Ga_ref/effective_area)
        plt.plot(x, zeta_Ga, 'g--')
        plt.plot(tilts_Ga_measured,zeta_Ga_measured, 'go')

        tilts_As_measured = [-6.7,-5,-2.6,0,0,0,0,2.4,2.4,5.1,5.1,5.1,7.5,9.7,15,20,29.8]
        zeta_As_measured = [1414,1209,1134,926,956,846,879,801,788,739,738,705,724,642,669,676,698]


        zeta_As = (zeta_As_ref/effective_area)
        plt.plot(x, zeta_As, 'r--')
        plt.plot(tilts_As_measured,zeta_As_measured, 'ro')

```

```

tilts =
 ↪   [2.4,2.4,0,0,-2.6,5.1,5.1,7.5,15,20,0,9.7,29.8,0,-5,-6.7,5.1,2.4,2.4,0,0,-2.6,5.1,5.1,7.5,15,20,0,9.7,29
zeta_norm
 ↪   =[0.872366742,0.886096115,0.754406271,0.730313727,0.615892807,0.94578534,0.945982151,0.964358346,1.04459
# Fit and plot models
#model_circle = fit_and_plot_model(tilts, zeta_norm, 'circle')
#model_circle = fit_and_plot_model(tilts, zeta_norm, 'circle')
model_circle = fit_and_plot_model(tilts, zeta_norm, 'circle', te=(20,30))
plot_GaAs(model_circle, 'circle')

#model_rect = fit_and_plot_model(tilts, zeta_norm, 'rect')
model_rect = fit_and_plot_model(tilts, zeta_norm, 'rect', te=(20,30))
plot_GaAs(model_rect, 'rect')

# Normalize with respect on 20 degrees instead
#zeta_norm2 =
 ↪   [0.844243841,0.857530614,0.730086118,0.706770256,0.596037978,0.915295609,0.915486075,0.933269868,1.01091
 ↪   0.779364329,0.78946529,0.673151572,0.650964929,0.584937343,0.84507137,0.836197722,0.908991996,1.01838676

model_circle = fit_and_plot_model(tilts, zeta_norm2, 'circle', te=(20,30))
plot_GaAs(model_circle, 'circle', 676, 545)

model_rect = fit_and_plot_model(tilts, zeta_norm2, 'rect', te=(20,30))
plot_GaAs(model_rect, 'rect', 676, 545)

#model_rect = fit_and_plot_model(tilts, zeta_norm2, 'rect', te=(15,30))
#plot_GaAs(model_rect, 'rect', 676, 545)
```

Listing 6: Determine ζ_{Ga} and ζ_{As} from tilt series of SC343 in [111]

```python
"""
This script determines the zeta-factors of Ga and As
from the GaAs core of SC343_NW112 in [111].
Github branch: siriagus/zeta_factor_method_combined
"""

get_ipython().magic(u'matplotlib qt4')
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
from determine_zeta_factor import *
from misc import load_pca

# The number of components for PCA has been determined in a former interactive analysis
s1_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S1_EDS Spectrum
 ↪   Image.dm3", 1)
s2_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S2_EDS Spectrum
 ↪   Image.dm3", 1)
s3_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S3_EDS Spectrum
 ↪   Image.dm3", 1)
s4_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S4_EDS Spectrum
 ↪   Image.dm3", 1)
s5_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S5_EDS Spectrum
 ↪   Image.dm3", 1)
s6_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S6_EDS Spectrum
 ↪   Image.dm3", 1)
s7_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S7_EDS Spectrum
 ↪   Image.dm3", 2)
s8_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S8_EDS Spectrum
 ↪   Image.dm3", 1)
s9_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S9_EDS Spectrum
 ↪   Image.dm3", 1)
s10_pca = load_pca("D:/TEM_Master/TEM-Master-Data/160208_training/S10_EDS Spectrum
 ↪   Image.dm3", 1)
spectra = [s1_pca, s2_pca, s3_pca, s4_pca, s5_pca, s6_pca, s7_pca, s8_pca, s9_pca,
 ↪   s10_pca]

# Background window for As and Ga
BW = np.array([[  9.79326174,   9.99588858,  11.09131142,  11.29393826],
       [  8.47288491,   8.66369057,   9.66970943,   9.86051509]])
IW = 1.2 # Integration window
p_GaAs = 5317.6 # density of GaAs in kg/m^3

# Experimental conditions
tilts = [2.4, 2.4, 0, 0, -2.6, 5.1, 5.1, 7.5, 15, 20]
probe_currents = [0.4228, 0.4235, 0.4156, 0.4178, 0.419, 0.4128, 0.4197, 0.416, 0.4103,
 ↪   0.4118]
acquisition_time = [0.2, 0.4, 0.4, 0.2, 0.2, 0.2, 0.4, 0.2, 0.2, 0.2]
thickness = [154.0, 154.0, 154.1, 154.1, 154.6, 154.2, 154.2, 154.6, 157.8, 161.6]

COMP_AS_GA_WT = (hs.material.atomic_to_weight([0.5, 0.5], ['As', 'Ga'])/100)
def run_analysis_GaAs(s, acquisition_time, probe_current, thickness, density, tilt):
        s.add_elements(['As', 'Ga'])
        s.add_lines()
        s.set_microscope_parameters(real_time = acquisition_time,
 ↪   beam_current=probe_current)
        s.set_microscope_parameters(tilt_stage=tilt, elevation_angle=24.3,
 ↪   azimuth_angle=0.0)
```

```
        s_ints = s.get_lines_intensity(integration_windows=IW, background_windows=BW)
        zeta_factors = determine_zeta_factor(s, s_ints, COMP_AS_GA_WT, thickness,
 ↪   density)
        return zeta_factors, s_ints
# ANALYSIS #
zeta_As = []
zeta_Ga = []

absorption_terms = []
for i,s in enumerate(spectra):
        zeta_factors, s_ints = run_analysis_GaAs(s, acquisition_time[i],
 ↪   probe_currents[i], thickness[i], p_GaAs, tilts[i])
        zeta_As.append(zeta_factors[0])
        zeta_Ga.append(zeta_factors[1])


print("zeta_As")
for z in zeta_As:
        print(np.mean(z.data.flatten()))
print("zeta_Ga")
for z in zeta_Ga:
        print(np.mean(z.data.flatten()))

### Qualitative analysis ###
# For reference peaks
spectra[0].sum('x').sum('y').plot(['Cu_Ka', 'Al_Ka', 'Ga_Ka', 'As_Ka', 'Ga_La', 'C_Ka',
 ↪   'Si_Ka', 'As_La', 'Ga_Kb', 'As_Kb', 'O_Ka', 'Cu_Kb', 'Ga_Ll', 'Fe_Ka', 'Co_Ka'])
```

Listing 7: Determine ζ_{Ga} and ζ_{As} from tilt series of SC48 in $[\bar{1}10]$

```python
""" SC48 - [110] """
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
from misc import load_pca

from determine_zeta_factor import *

BW = np.array([[  9.79326174,   9.99588858,  11.09131142,  11.29393826],
       [  8.47288491,   8.66369057,   9.66970943,   9.86051509]])
IW = 1.2 # Integration window
p_GaAs = 5317.6 # density of GaAs in kg/m^3

# Experimental conditions
# Rot.111 (4) + Rot [-1-12] (3)
tilts = [0, -5, -6.7, 5.1, 0, 9.7, 29.8]
probe_currents = [0.62905, 0.6311, 0.63015, 0.6289, 0.6449, 0.6355, 0.6343]
acquisition_time = [0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2]
#thickness = [82.6, 82.3, 82.3, 83.6, 82.3, 83.5, 94.8]
thickness = [82.6, 82.3, 82.3, 83.6, 92, 93.3, 106]

# Rotation around
t1_pca = load_pca("D:/TEM_Master/160203_training/T1_EDS Spectrum Image.dm3", 1)
t2_pca = load_pca("D:/TEM_Master/160203_training/T2_EDS Spectrum Image.dm3", 1)
t3_pca = load_pca("D:/TEM_Master/160203_training/T3_EDS Spectrum Image.dm3", 1)
t4_pca = load_pca("D:/TEM_Master/160203_training/T4_EDS Spectrum Image.dm3", 1)

# Rotation around [111]
s1_pca = load_pca('D:/TEM_Master/TEM-Master-Data/160113/1_EDS Spectrum Image.dm3', 1)
s3_pca = load_pca('D:/TEM_Master/TEM-Master-Data/160113/3EDS Spectrum Image.dm3', 1)
s5_pca = load_pca('D:/TEM_Master/TEM-Master-Data/160113/5EDS Spectrum Image.dm3', 1)

spectra = [t1_pca, t2_pca, t3_pca, t4_pca, s1_pca, s3_pca, s5_pca]
s5_pca.sum('x').sum('y').plot(['Cu_Ka', 'Al_Ka', 'Ga_Ka', 'As_Ka', 'Ga_La', 'C_Ka',
 ↪  'Si_Ka', 'As_La', 'Ga_Kb', 'As_Kb', 'O_Ka', 'Cu_Kb', 'Ga_Ll', 'Fe_Ka', 'Co_Ka'])

COMP_AS_GA_WT = (hs.material.atomic_to_weight([0.5, 0.5], ['As', 'Ga'])/100)
def run_analysis_GaAs(s, acquisition_time, probe_current, thickness, density, tilt):
        s.add_elements(['As', 'Ga'])
        s.add_lines()
        s.set_microscope_parameters(real_time = acquisition_time,
 ↪  beam_current=probe_current)
        s.set_microscope_parameters(tilt_stage=tilt, elevation_angle=24.3)
        s_ints = s.get_lines_intensity(integration_windows=IW, background_windows=BW)
        zeta_factors = determine_zeta_factor(s, s_ints, COMP_AS_GA_WT, thickness,
 ↪  density)
        return zeta_factors, s_ints, s._get_dose('zeta'), absorption_terms

zeta_As = []
zeta_Ga = []

i_As = []
i_Ga = []
for i,s in enumerate(spectra):
        zeta_factors, s_ints, dose, absterm = run_analysis_GaAs(s, acquisition_time[i],
 ↪  probe_currents[i], thickness[i], p_GaAs, tilts[i])
        zeta_As.append(zeta_factors[0])
        zeta_Ga.append(zeta_factors[1])
        i_As.append(np.mean(s_ints[0].data)/dose)
        i_Ga.append(np.mean(s_ints[1].data)/dose)

# RESULTS #
# Plot average values #
for i in range(len(spectra)):
        print("Tilt: %.1f" % (tilts[i]))
        print("----------------------")
        print('|-- Zeta_As = %.2f' % (zeta_As[i].mean('x').mean('y').data[0]))
        print('|-- Zeta_Ga = %.2f' % (zeta_Ga[i].mean('x').mean('y').data[0]))

print("Rotation around [111]")
print("zeta_As")
for z in zeta_As[:4]:
        print(np.mean(z.data.flatten()))
print("zeta_Ga")
for z in zeta_Ga[:4]:
        print(np.mean(z.data.flatten()))

print ("------------------------")
print ("Rotation around [-1-12]")
print("zeta_As")
for z in zeta_As[4:]:
        print(np.mean(z.data.flatten()))
print("zeta_Ga")
for z in zeta_Ga[4:]:
        print(np.mean(z.data.flatten()))
for i in absorption_terms[4:]:
        print(i[0])
```

Listing 8: Determine ζ_C from thin carbon foil

```
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams.update({'font.size': 14})
from determine_zeta_factor import *

# Constants
IW = 1.2
BW_C = [[ 0.11733517,  0.17069011,  0.38410989,  0.43746483]] # only works for pure
  ↪  carbon
p_Carbon = hs.material.elements.C.Physical_properties.density_gcm3 * 1000

# Load spectra
s = hs.load('D:/TEM_Master/TEM-Master-Data/2015/S5_EDS Spectrum Image.dm3')
s = s.inav[:, 385:]
s.change_dtype('float')
s.decomposition(algorithm='nmf', output_dimension=3)
s.set_microscope_parameters(real_time=0.2, beam_current=0.6063)
s_ints = s.get_lines_intensity(xray_lines=['C_Ka'], integration_windows=1.2)

t = 28/np.cos(np.radians(-4.6)) # +- 5 nm
zeta_C_data = determine_zeta_factor(s, s_ints, [1], t, p_Carbon)[0].data.flatten()
zeta_C_mean = np.mean(zeta_C_data)

print("Mean:", zeta_C_mean, "Median:", np.median(zeta_C_data))
plt.figure()
plt.hist(zeta_C_data, 40, facecolor='red')
plt.title('Zeta factor for Carbon (-4.6 degrees tilt)')
plt.xlabel('Zeta-factor C')
plt.axvline(zeta_C_mean)
```

Listing 9: Determine ζ_{Ga} and ζ_{Sb} from GaSb

```python
""" This script determines the zeta-factors of Ga and Sb """

get_ipython().magic(u'matplotlib qt4')
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
from determine_zeta_factor import *
from misc import load_pca

FILE_PATH = 'D:/TEM_Master/TEM-Master-Data/160530_ZetaFactorSamples/GaSb/'
EDS_FILES = ['GaSb_A2_50um_11.7deg_EDS Spectrum Image.dm3','GaSb_A2_40um_16.8deg_EDS
 ↪  Spectrum Image.dm3',
                         'GaSb_A2_40um_11.7deg_EDS Spectrum Image.dm3',
 ↪  'GaSb_A2_50um_6.7deg_EDS Spectrum Image.dm3',
                         'GaSb_A2_10um_6.7deg_EDS Spectrum Image.dm3',
 ↪  'GaSb_A2_40um_6.7deg_EDS Spectrum Image.dm3']

IW = 1.2
BW = np.array([[  8.47468585,   8.63369057,  10.56970943,  10.72871415],
       [  2.78770631,   2.89337087,   4.81602913,   4.92169369]])
spectra = []
p_GaSb = 5614 # kg/m^3

# Experimental conditions
tilts = [11.7, 16.8, 11.7, 6.7, 6.7, 6.7]
probe_currents = [0.6751,0.41705,0.4208,0.6864,0.02685,0.417]
acquisition_time = [1024 * 0.1] * 6 # sum
thickness = [122.5, 43.5, 30, 37.5, 47.5, 47.5]
for i in range(6):
        s = load_pca(FILE_PATH + EDS_FILES[i], 1).sum('x').sum('y')
        s.add_elements(['Ga', 'Sb'])
        s.add_lines()
        s.set_microscope_parameters(real_time = acquisition_time[i],
 ↪  beam_current=probe_currents[i], tilt_stage=tilts[i], azimuth_angle=0.0,
 ↪  elevation_angle=24.3)
        spectra.append(s)


COMP_GA_SB_WT = (hs.material.atomic_to_weight([0.5, 0.5], ['Ga', 'Sb'])/100)

zeta_Ga = []
zeta_Sb = []
for i,s in enumerate(spectra):
        s_ints = s.get_lines_intensity(integration_windows=IW, background_windows=BW)
        zeta_factors = determine_zeta_factor(s, s_ints, COMP_GA_SB_WT, thickness[i],
 ↪  p_GaSb)
        zeta_Ga.append(zeta_factors[0])
        zeta_Sb.append(zeta_factors[1])

print("zeta_Ga_Ka")
for z in zeta_Ga:
        print(np.mean(z.data.flatten()))
print("zeta_Sb_La")
for z in zeta_Sb:
        print(np.mean(z.data.flatten()))
```

Listing 10: Determine ζ_{Al} from Al thin film

```
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib
matplotlib.rcParams.update({'font.size': 14})
import matplotlib.pyplot as plt
from determine_zeta_factor import *

BW = np.array([[ 1.01666201,  1.09327468,  1.87972532,  1.95633799]])

MASTER_FILE_PATH = 'D:/TEM_Master/TEM-Master-Data/'
FILE_PATH = MASTER_FILE_PATH + '160319/Al/'
# STEM images

def load_pca(filepath, pca_comps):
        s = hs.load(filepath)
        s.change_dtype('float')
        s.decomposition()
        return s.get_decomposition_model(pca_comps)

s1 = load_pca(FILE_PATH + '1EDS Spectrum Image.dm3', 8)
s2 = load_pca(FILE_PATH + '2EDS Spectrum Image.dm3', 6)
s3 = load_pca(FILE_PATH + '3EDS Spectrum Image.dm3', 6)
spectra = [s1, s2, s3]

# CONSTANTS #
probe_currents=[0.4063, 0.66865, 0.2236]

acquisition_time = [1.0, 1.0, 1.0]
thickness = [115, 110, 100] # average
density_Al = hs.material.elements.Al.Physical_properties.density_gcm3 * 1000 # kg/m^3
IW = 1.2


def run_analysis_Al(s, acquisition_time, probe_current, thickness, density):
        s.add_lines(['Al_Ka'])
        s.set_microscope_parameters(real_time = acquisition_time,
 ↪   beam_current=probe_current)
        s_ints = s.get_lines_intensity(integration_windows=IW, background_windows=BW)
        zeta_factors = determine_zeta_factor(s, s_ints, [1], thickness, density)[0]
        return zeta_factors

# ANALYSIS #
# Do analysis for every data point, then get mean #
zeta_factors = []
for i,s in enumerate(spectra):
        zeta_factors.append(run_analysis_Al(s, acquisition_time[i], probe_currents[i],
 ↪   thickness[i], density_Al))

# RESULTS #
print("Zeta first, then get mean:")
print("--------------------------")
for i in range(len(spectra)):
        print('#%d: Zeta_Al = %f' % (i, zeta_factors[i].mean('x').mean('y').data[0]))

print()

```

```python
# Sum all data, then do analysis #
zeta_factors_sum = []
for i,s in enumerate(spectra):
        zeta_factors_sum.append(run_analysis_Al(s.sum(’x’).sum(’y’), acquisition_time[i]*256, probe_currents[i], thickness[i], density_Al))

# RESULTS #
print("Sum first, get zeta:")
print("--------------------------")
for i in range(len(spectra)):
        print(’#%d: Zeta_Al = %f’ % (i, zeta_factors_sum[i].data[0]))

aperture = [40, 50, 30]
means = [np.mean(z.data.flatten()) for z in zeta_factors]
medians = [np.median(z.data.flatten()) for z in zeta_factors]

for z,app,m in zip(zeta_factors, aperture, means):
        plt.figure()
        plt.hist(z.data.flatten(), 32, facecolor=’green’)
        plt.title(’Zeta-factor for Aluminium (’ + str(app) + ’um aperature)’)
        plt.xlabel(’Zeta-factor Al’)
        plt.axvline(m)
```

Listing 11: Quantification of GaAs/GaAsSb (SC48)

```python
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
import time
from zeta_factor_method import *
from misc import load_nmf

IW = 1.2

BW = np.array([        [9.79326174,    9.99588858,   11.09131142,   11.29393826],
                                [8.47468585,    8.63369057,   10.56970943,   10.72871415],
                               [2.78770631,    2.89337087,    4.81602913,    4.92169369]])


FILE_PATH = 'D:/TEM_Master/TEM-Master-Data/160113/'
FN = ['2_EDS Spectrum Image-NMF6.hdf5', '4EDS Spectrum Image-NMF4.hdf5', '6EDS Spectrum
↪    Image-NMF4.hdf5']

for i in range(3):
        s = hs.load(FILE_PATH + FN[i])

# Use zeta factors for 0 degrees (average)
zeta_factors = [1197,859,699] # 0 degrees

s2 = hs.load('D:/TEM_Master/TEM-Master-Data/160113/2_EDS Spectrum Image-NMF6.hdf5')
s2.add_elements(['As', 'Ga', 'Sb'])
s2.add_lines()
s2.set_microscope_parameters(real_time=0.2, beam_current=0.64225)
s2.set_microscope_parameters(tilt_stage=0.0, elevation_angle=24.3, azimuth_angle=0.0)
s2_ints = s2.get_lines_intensity(integration_windows=IW, background_windows=BW)

mask = (s2_ints[1] > 5)
for i in s2_ints:
        i *= (i > 0) * mask

s2_comp, s2_pt = s2.quantification(s2_ints, 'zeta', zeta_factors)
s2_comp_linescan = [c.isig[10:17].mean('x') for c in s2_comp]
hs.plot.plot_spectra(s2_comp_linescan, legend=s2.metadata.Sample.elements)


kfactors = [2.835, 2.419, 3.976]
s2_comp_CL = s2.quantification(s2_ints, 'CL', kfactors)
s2_comp_CL_linescan = [c.isig[10:17].mean('x') for c in s2_comp_CL]
hs.plot.plot_spectra(s2_comp_CL_linescan, legend=s2.metadata.Sample.elements)
```

Listing 12: Quantification of linescan and close up of transition from GaAs to AlGaAs

```
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
from misc import *
from zeta_factor_method import *

IW = 1.2

FILE_PATH = 'D:/TEM_Master/TEM-Master-Data/160428_SC365_2xFIB/'

BW = np.array([[  1.61561341,   1.62327468,   1.59972532,   1.60738659],
       [  9.79326174,   9.99588858,  11.09131142,  11.29393826],
       [  8.47288491,   8.66369057,   9.66970943,   9.86051509]])

##############################################################################################################
# SAMPLE A - "line-scan" #
a = load_nmf(FILE_PATH + 'A_EDS Spectrum Image.dm3', 3)
um_to_nm_scale(a)
a = a.inav[:600., :]
a = a.mean('y')

a.add_elements(['Al', 'Ga', 'As'])
a.add_lines()
a.set_microscope_parameters(real_time=0.1, beam_current=0.68435, elevation_angle=24.3,
 ↪   tilt_stage=8.3, azimuth_angle=0.0)
a_ints = a.get_lines_intensity(integration_windows=IW, background_windows=BW)

for ai in a_ints:
        ai *= (ai > 0)

zeta_factors = [227, 724, 599] # uses values from 7.5 degrees (Al estimated from 0
 ↪   degree measurement)
comp_a, pt_a = a.quantification(a_ints, 'zeta', factors=zeta_factors,
 ↪   composition_units='atomic')
#comp_a_abs, pt_a_abs = quantification_zeta(a, a_ints, zeta_factors,
 ↪   composition_units='atomic', convergence_precision=0.001)

kfactors = [1.04, 2.835, 2.4919]
comp_a_CL = a.quantification(a_ints, 'CL', factors=kfactors,
 ↪   composition_units='atomic')

ta = get_thickness_map(comp_a, pt_a_abs)
ta_abs = get_thickness_map(comp_a_abs, pt_a_abs)

hs.plot.plot_spectra([ta, ta_abs], legend=['Zeta', 'Zeta abs.'])

elements = ['Al', 'As', 'Ga']
for i in range(3):
        hs.plot.plot_spectra([comp_a[i], comp_a_abs[i], comp_a_CL[i]], legend=['Zeta',
 ↪   'Zeta abs.', 'CL'])
        plt.title('Atomic percent of ' + elements[i])
        plt.grid(True)

# Compare zeta and CL
compDiffAbsCL = [(cz-cl) for cz,cl in zip(comp_b_abs, compCL)]
hs.plot.plot_images(comp_b_abs + compCL + compDiffAbsCL, cmap='inferno',
 ↪   colorbar='single', axes_decor='off')
```

```python

# Compare Al
hs.plot.plot_images([comp_b[0], comp_b_abs[0], compCL[0]], cmap='inferno', colorbar='single', axes_decor='off')

# Compare As in the sample
hs.plot.plot_images([comp_b[1], comp_b_abs[1], compCL[1]], cmap='inferno', axes_decor='off', colorbar='single')


zeta_factors = [299, 724, 599]
comp_a_abs, pt_a_abs = quantification_zeta(a, a_ints, zeta_factors, composition_units='atomic', convergence_precision=0.001)

# Find zeta_Al indirectly, assuming composition of As = 50% over linescan
zeta_Al = 227 # Start value
while(True):
        zeta_factors = [zeta_Al, 724, 599]
        comp_a_abs, pt_a_abs = quantification_zeta(a, a_ints, zeta_factors, composition_units='atomic', convergence_precision=0.001)
        delta = np.mean(comp_a_abs[1].data) - 50
        print("Delta:", delta)
        if (np.abs(delta) < 0.001):
                break
        else:
                zeta_Al += delta * 30
print("Zeta_Al indirect value: %.2f ~= %.0f" % (zeta_Al, zeta_Al))
# Gives Zeta_Al indirect value: 298.74 ~= 299

# SAMPLE B - [-1-12] - on zone #

b = load_nmf(FILE_PATH + 'B_EDS Spectrum Image.dm3', 6)
b = b.inav[:, 6:109]
um_to_nm_scale(b)
b.add_elements(['Al', 'Ga', 'As'])
b.add_lines()
b_ints = b.get_lines_intensity(integration_windows=IW, background_windows=BW)

mask = (b_ints[2] > 6)
for bi in b_ints:
        bi *= (bi > 0) * mask

b.set_microscope_parameters(real_time=0.1, beam_current=0.6848, elevation_angle=24.3, tilt_stage=8.3, azimuth_angle=0.0)

zeta_factors = [227, 724, 599] # uses values from 7.5 degrees (Al estimated from 0 degree measurement)

kfactors = [1.04, 2.835, 2.4919]
compCL = b.quantification(b_ints, 'CL', factors=kfactors, composition_units='atomic')

comp_b, pt_b = b.quantification(b_ints, 'zeta', factors=zeta_factors, composition_units='atomic')
compCL = b.quantification(b_ints, 'CL', factors=kfactors, composition_units='atomic')
comp_b_abs, pt_b_abs = quantification_zeta(b, b_ints, zeta_factors, composition_units='atomic', convergence_precision=0.001)
comp_b_abs_indirect, pt_b_abs_indirect = quantification_zeta(b, b_ints, [299, 724, 599], composition_units='atomic', convergence_precision=0.001)

tb = get_thickness_map(comp_b, pt_b)
tb2 = get_thickness_map(comp_b_abs, pt_b_abs)
tb3 = get_thickness_map(comp_b_abs_indirect, pt_b_abs_indirect)
```

```

hs.plot.plot_images([tb, tb2, tb3], cmap='gnuplot2', axes_decor='off',
 ↪   colorbar='single')

# Compare zeta and CL
compDiffAbsCL = [(cz-cl) for cz,cl in zip(comp_b_abs, compCL)]
hs.plot.plot_images(comp_b_abs + compCL + compDiffAbsCL, cmap='inferno',
 ↪   colorbar='single', axes_decor='off')

# Compare Al
hs.plot.plot_images([comp_b[0], comp_b_abs[0], compCL[0], comp_b_abs_indirect[0]],
 ↪   cmap='inferno', colorbar='single', axes_decor='off')

# Compare As in the sample
hs.plot.plot_images([comp_b[1], comp_b_abs[1], compCL[1], comp_b_abs_indirect[1]],
 ↪   cmap='inferno', axes_decor='off', colorbar='single')
hs.plot.plot_images([comp_b[2], comp_b_abs[2], compCL[2], comp_b_abs_indirect[2]],
 ↪   cmap='inferno', axes_decor='off', colorbar='single')
```

Listing 13: Quantification of tip of FIB lamella of SC365 in $[\bar{1}\bar{1}2]$

```python
get_ipython().magic(u'matplotlib qt4')
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
from zeta_factor_method import *

def apply_mask(signal, s_ints, threshold_counts):
        mask = [si > threshold_counts for si in s_ints]
        signal = [s * m for (s,m) in zip(signal,mask)]
        for s in signal:
                s.data = np.nan_to_num(s.data)
        return signal

IW = 1.2

zeta_factors = [278, 859, 699]

s = hs.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc2_EDS Spectrum
 ↪   Image.dm3')
s.learning_results.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc2_EDS
 ↪   Spectrum Image_NMF.npz')
s = s.get_decomposition_model()
s.add_elements(['Al', 'As', 'Ga'])
s.add_lines()
s.set_microscope_parameters(real_time=0.2, beam_current=0.5859)
s.set_microscope_parameters(elevation_angle=24.3, tilt_stage=0.0, azimuth_angle=0.0)

s_ints = s.get_lines_intensity(integration_windows=IW)
comp, pt = s.quantification(s_ints, 'zeta', factors=zeta_factors,
 ↪   composition_units='atomic')
compAbs, ptAbs = quantification_zeta(s, s_ints, zeta_factors,
 ↪   convergence_precision=0.001)
compCL = s.quantification(s_ints, 'CL', factors =[1.040, 2.835, 2.419])

comp = apply_mask(comp, s_ints, 6)
compAbs = apply_mask(compAbs, s_ints, 6)
compCL = apply_mask(compCL, s_ints, 6)

t_tip = hs.load('D:/TEM_Master/TEM-Master-Data/160218_SC365_NW331_2xcc/Tip_Thickness
 ↪   Map.dm3')
t_tip *= 103 # mfp_GaAs

hs.plot.plot_images([get_thickness_map(comp, pt), get_thickness_map(compAbs, ptAbs),
 ↪   t_tip], axes_decor='off', cmap='gnuplot2', colorbar='single')
hs.plot.plot_images([comp[1], comp[2], compAbs[1], compAbs[2], compCL[1], compCL[2]],
 ↪   cmap='inferno', axes_decor='off', colorbar='single', per_row=2)
hs.plot.plot_images([comp[0], compAbs[0], compCL[0]], cmap='inferno', axes_decor='off',
 ↪   colorbar='single')
```

Listing 14: Quantification of tip of FIB lamella of SC365 in [111]

```
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
import time
from zeta_factor_method import *

IW = 1.2

s = hs.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc1_EDS Spectrum
  ↪ Image.dm3')
s.learning_results.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc1_EDS_NMF.npz')
s = s.get_decomposition_model()
s = s.inav[7:, 7:] # Only NW
s.add_elements(['Al', 'As', 'Ga'])
s.add_lines()
s.set_microscope_parameters(real_time=0.2, beam_current=0.6546)
s.set_microscope_parameters(tilt_stage=1.7, elevation_angle=24.3, azimuth_angle=0.0)

def apply_mask(signal, s_ints, threshold_counts):
        mask = [si > threshold_counts for si in s_ints]
        signal = [s * m for (s,m) in zip(signal,mask)]
        for s in signal:
                s.data = np.nan_to_num(s.data)
        return signal


# Uses values from 2.4 instead of 1.7 (closest)
zeta_factors = [247, 794, 694]

k_factors = [1.040, 2.835, 2.419]
s_ints = s.get_lines_intensity(integration_windows=IW)

t0 = time.clock()
comp, pt = s.quantification(s_ints, 'zeta', factors=zeta_factors)
implemented_time = time.clock() - t0

t0 = time.clock()
compCL = s.quantification(s_ints, 'CL', factors=k_factors)
cliff_lorimer_time = time.clock() - t0

t0 = time.clock()
compCustom, pt_custom = quantification_zeta(s, s_ints, zeta_factors)
myversion_time = time.clock() - t0

t0 = time.clock()
compCustomAbs, pt_customAbs = quantification_zeta(s, s_ints, zeta_factors,
  ↪ convergence_precision=0.001)
absorption_correction_time = time.clock() - t0

# Compare custom method with the one currently implemented in HyperSpy
# Ignore nan-values
for c,cc in zip(comp, compCustom):
        c.data = np.nan_to_num(c.data)
        cc.data = np.nan_to_num(cc.data)

# Check to see if the custom version (withouth abs.corr.) give the same result as
  ↪ HyperSpy version
```

```
assert np.allclose(hs.stack(comp).data, hs.stack(compCustom).data), "The custom method
 ↪   and HyperSpy method gives different results!"

# Compare time used for quantification
print("HyperSpy zeta version used ", implemented_time, " seconds.")
print("HyperSpy CL version used ", cliff_lorimer_time, " seconds.")
print("Custom version used ", myversion_time, " seconds.")
print("Custom version with absorption correction used", absorption_correction_time,
 ↪   "seconds.")

comp = apply_mask(comp, s_ints, 10)
compCustomAbs = apply_mask(compCustomAbs, s_ints, 10)
compCL = apply_mask(compCL, s_ints, 10)

hs.plot.plot_images([comp[1], comp[2], compCustomAbs[1], compCustomAbs[2], compCL[1],
 ↪   compCL[2]], cmap='inferno', axes_decor='off', colorbar='single')
hs.plot.plot_images([comp[0], compCustomAbs[0], compCL[0]], cmap='inferno',
 ↪   axes_decor='off', colorbar='single')

linescan_comp = [c.isig[0.08383:0.1031, :].mean('x') for c in comp]
linescan_compAbs = [c.isig[0.08383:0.1031, :].mean('x') for c in compCustomAbs]
linescan_compCL = [c.isig[0.08383:0.1031, :].mean('x') for c in compCL]

compare = [linescan_comp[0], linescan_comp[1], linescan_comp[2], linescan_compAbs[0],
 ↪   linescan_compAbs[1], linescan_compAbs[2], linescan_compCL[0], linescan_compCL[1],
 ↪   linescan_compCL[2]]
#hs.plot.plot_spectra(compare, legend=['Al (zeta)', 'As (zeta)', 'Ga (zeta)', 'Al (abs.
 ↪   corr.)', 'As (abs. corr.)', 'Ga (abs. corr.)', 'Al (CL)', 'As (CL)', 'Ga (CL)'])
hs.plot.plot_spectra([linescan_compAbs[0], linescan_compAbs[1], linescan_compAbs[2],
 ↪   linescan_compCL[0], linescan_compCL[1],linescan_compCL[2]], legend=['Al (zeta
 ↪   abs.)', 'As (zeta abs.)', 'Ga (zeta abs.)', 'Al (CL)', 'As (CL)', 'Ga (CL)'])
hs.plot.plot_spectra([linescan_compAbs[0], linescan_compAbs[1], linescan_compAbs[2],
 ↪   linescan_compCL[0], linescan_compCL[1],linescan_compCL[2]])
plt.yticks(np.arange(0,75,5))
plt.ylim((0, 60))
plt.xlim((0.02, 0.16))
plt.grid(True)

# Reference
t_SC365 = hs.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc1_Thickness
 ↪   Map.dm3')
t_SC365 *= 103 # mfp_GaAs
t_SC365 = t_SC365.isig[32.43:192.8, 19.28:182.8]
centre_core_thickness = np.mean(t_SC365.isig[99.38:132.7, 80.13:111.6].data.flatten())
print("Centre of GaAs core has a thickness of", centre_core_thickness, "nm.")

t = get_thickness_map(comp, pt)
t2 = get_thickness_map(compCustomAbs, pt_customAbs)

hs.plot.plot_images([t, t2, t_SC365], cmap='gnuplot2', axes_decor='off',
 ↪   colorbar='single')
```

Listing 15: Thickness maps from EFTEM t/λ maps

```
%matplotlib qt4
import hyperspy.api as hs
import numpy as np
import matplotlib.pyplot as plt
import scipy

# SC48
unfiltered = hs.load('D:/TEM_Master/TEM-Master-Data/160203_training/Unfiltered_TM.dm3')
zero_loss = hs.load('D:/TEM_Master/TEM-Master-Data/160203_training/Elastic_TM.dm3')
thickness_map = hs.load('D:/TEM_Master/TEM-Master-Data/160203_training/Thickness
 ↪   Map.dm3')
lambda_map = thickness_map.deepcopy()

known_thickness = 95 * np.sqrt(3)/2 # nm

lambda_map.map(scipy.ndimage.rotate, angle=16.4, reshape=False) # rotate
lambda_map = lambda_map.isig[338:355, 3:479] # crop to middle of NW
lambda_map.data = 1/lambda_map.data # lambda / t
lambda_map.data = known_thickness * lambda_map.data # Converted to mfp map assuming
 ↪   known thickness
mfp = np.mean(lambda_map.data.flatten())
lambda_map.mean('x').plot()
plt.axhline(mfp)
print("Mean free path GaAS:", mfp) # Output: 103.224

mfp = 103 # use from now on for GaAS

## SC48 EFTEM maps ##
unfiltered.map(scipy.ndimage.rotate, angle=16.4, reshape=False)
zero_loss.map(scipy.ndimage.rotate, angle=16.4, reshape=False)
thickness_map.map(scipy.ndimage.rotate, angle=16.4, reshape=False)
hs.plot.plot_images([zero_loss.isig[213.8:340.8, 1.374:362.3],
 ↪   unfiltered.isig[213.8:340.8, 1.374:362.3]], cmap='Blues', axes_decor='off',
 ↪   colorbar='single')
thickness_map.isig[213.8:340.8, 1.374:362.3].plot(cmap='Blues')

# SC343: Used for zeta factors
thickness_SC343 = hs.load('D:/TEM_Master/TEM-Master-Data/160208_training/Thickness
 ↪   Map.dm3')
thickness_SC343 *= mfp
thickness_SC343.plot(cmap='Blues')
thickness_SC343 = thickness_SC343.isig[170.1:268.2, 225.1:311.5] # middle of nanowire
tdata = thickness_SC343.data.flatten()
print("Thickness of SC343 [111]:", np.mean(tdata), "nm, std:", np.std(tdata))

# SC365 - Used for quantification
t_SC365 = hs.load('D:/TEM_Master/TEM-Master-Data/160304_SC365_2xcc/cc1_Thickness
 ↪   Map_b.dm3')
t_SC365 *= mfp
t_SC365 = t_SC365.isig[32.43:192.8, 19.28:182.8]
t_SC365.plot(cmap='Blues')
centre_core_thickness = np.mean(t_SC365.isig[99.38:132.7, 80.13:111.6].data.flatten())
print("Centre of GaAs core has a thickness of", centre_core_thickness, "nm.")

t_Al = hs.load('D:/TEM_Master/TEM-Master-Data/160319/Al/Thickness Map.dm3')
```