Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this diploma or master thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Multislice Approach for High Resolution Energy Filtered Inelastic Image Simulation

ausgeführt am

Institut für Festkörperphysik

unter der Anleitung von

a.o. Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn.
Peter Schattschneider

durch

Martin Zauner
Reittern 24
4890 Weißenkirchen i. A.

Heidelberg, Mai 2010

TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

A thesis submitted in partial fulfillment for the degree of Master of Science

Multislice Approach for High Resolution Energy Filtered Inelastic Image Simulation

in the

Institut für Festkörperphysik

supervised by

a.o. Univ.Prof. Dipl.-Ing. Mag.rer.nat. Dr.techn.
Peter Schattschneider

by

Martin Zauner
Reittern 24
4890 Weißenkirchen i. A.

Heidelberg, May 2010

"The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong it usually turns out to be impossible to get at or repair."

Douglas Adams

Abstract

An approach is presented to simulate energy filtered inelastic images of crystalline materials with atomic resolution. To this purpose the multislice method (used to calculate multiple elastic scattering events) was adopted in such a way that it includes inelastic scattering events. The approach is described with a density matrix formalism. The implementation into the software program is explained and simulation results for SrTiO3 and Si are presented. The program shows a simple way to parallelize the calculations and proves that it is possible to run simulations on home computers in affordable time.

Danksagung

Ich bedanke mich bei Prof. Peter Schattschneider für seine Unterstützung und die vielen interessanten Gespräche. Sein großes Maß an Geduld und sein Motivationsgeschick haben diese Arbeit ermöglicht.

Dipl.-Ing. Stefan Löffler gilt mein Dank fur die vielen hilfreichen Tips und Korrekturvorschläge.

Ein besonderer Dank geht an meine Eltern Ernst und Brigitte sowie an meine Schwestern. Sie haben mich über die Jahre immer unterstützt und mir ein festes Zuhause geboten.

Meiner Freundin Tatjana danke ich für ihre Geduld und Unterstützung und für die wunderbare gemeinsame Zeit.

Mein Dank geht zudem an Stefan Arnez der immer hilfsbereit zur Stelle war und sich geduldig allen Arten von Aufgaben widmete.

Contents

List of Figures

Abbreviations

CCD charge-coupled device

CTF contrast transfer function

EFTEM energy-filtered transmission electron microscopy

EELS electron energy-loss spectrometry

FFT fast Fourier transform

IFD image file directory

MDFF mixed dynamic form factor

MDOS mixed dynamic object spectrum

PSF point spread function

TEM transmission electron microscope

TIFF tagged image file format

XAS X-ray absorption spectroscopy

Chapter 1

Transmission Electron Microscopy

1.1 Introduction

The electron energy-loss spectrometry (EELS) in the electron microscope has evolved to a routine method for materials characterisation. Fast electrons, after having interacted with the specimen, are analysed with respect to the energy lost during interaction. Quite similar to X-ray absorption spectroscopy (XAS) ionisation edges in the spectra serve as fingerprints for particular chemical elements. In combination with the high spatial resolution of the electron microscope, this method has been used for chemical microanalysis, usually allowing a spatial resolution in the nm range and a detection sensitivity of a few at% of foreign elements in a matrix. With the last generation of instruments equipped with lens correctors and high brightness electron sources the detection of chemical signal from single atoms and with atomic resolution has been reported [3]. The actual rapid evolution of instruments for electron microscopy will without doubt allow the direct visualisation of the ionisation event in real space in the next years. Many new and unexpected structures will be seen, such as anisotropy in bonding, or single spin polarized electronic transitions. The interpretation of such spatially resolved signal is not straightforward, the reason

being the coupling of (elastic) Bragg scattering with the (inelastic) ionisation interaction. Whereas the former is coherent, the later leads to a collapse of the probe electron's wave function at the ionized site, and as such is at least partially incoherent. It has been shown in detail that the intricate combination of both types of scattering may lead to rich and unexpected structure in energy filtered or energy spectroscopic images [4]. The simulation of the image formation process is indispensable for the interpretation of experiments. The respective theory has been developed in detail by several groups and is well understood [5, 6]. What is still missing is an easy to handle numerical treatment in order to simulate energy-filtered transmission electron microscopy (EFTEM) images of crystalline material. The aim of the present diploma thesis is the development of a simulation program. It is based on the use of the density matrix for the probe electron, and its propagation through the lattice. For the propagation part, a program package written by Earl J. Kirkland [7] for the simulation of high resolution lattice images is used as template and adapted to the inelastic case. Inelastic scattering factors are taken from atomic models described by Clementi and Raimondi[8], but can in principle be replaced by more sophisticated ones. The program was tested for Si and SrTiO3.

1.2 EELS in the TEM

The transmission electron microscope (TEM) operates with a beam of fast (100 - 300 keV) electrons. The beam can be manipulated with magnetic lenses, forming either a highly collimated spot on the specimen or illuminating a larger area with variable convergence angle. The microscope column consists of a beam forming part (the equivalent to the condenser system in light optics), a specimen stage where the beam traverses a thinned (5 - 100 nm) specimen and an image forming part consisting of several lenses which allow to project both the image plane and the diffraction plane onto the detecting device (a fluorescent screen, photographic film, or CCD camera). The spectrometer is mounted either in the column or (when used as an add-on device) after the final image plane. Modern spectrometers can also be operated as energy filters. This last mode

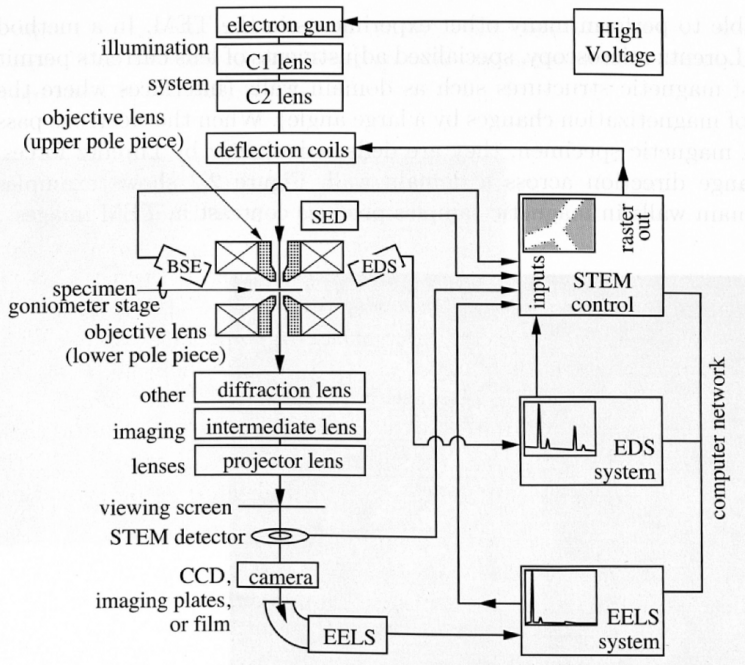

FIGURE 1.1: Block diagram of typical TEM with STEM capability.[1]

allows the imaging of the specimen with the signal corresponding to a particular energy loss, in the present context this is a particular ionisation event, with high spatial resolution. The principal scheme of the instrument is sketched in Fig. 1.1. Details can be found in textbooks [1, 9, 10].

1.3 Structure of this Thesis

After this short introduction, the thesis at hand starts with a short repetition of elastic and inelastic scattering theory. The chapter includes the most fundamental facts and equations. The third chapter contains the main theory that leads to the design of the program. First the solution of the double channeling problem using a density matrix approach[11] is explained. A revision of the multislice method follows. Finally the elementary way of coherent image formation is described. The fourth chapter describes the structure of the designed software package and the way calculations are done. The usage of the programs

is explained and the alterable parameters are listed. An estimation of the calculation effort of simulations ends this chapter. The fifth chapter presents some simulation results, namely focal series for SrTiO3 and Si. In the last chapter the results of the work are discussed.

Chapter 2

Scattering Theory

2.1 Elastic Scattering

2.1.1 Coherent and Incoherent Scattering

Coherent scattering demands the summation of individual wavelets under consideration of their relative phases. This leads to constructive or destructive interference of the complete wavefunction.

$$\Psi_{coh} = \sum_i \psi_i \tag{2.1}$$

$$I_{coh} = \Psi^*_{coh}\Psi_{coh} = \left|\sum_i \psi_i\right|^2 \tag{2.2}$$

In other words: terms containing different wavefunctions in the form of $\psi_i^*\psi_j$ account to the sum. In contrary incoherent scattering demands a summation over the intensities of every wavefunction.

$$I_{inc} = \sum_i I_i = \sum_i |\psi_i|^2 \tag{2.3}$$

When we look at the threatment of inelastic scattering below we will see, that the system we are looking at demands both, coherent and incoherent summation of wavelets. To employ a stringent approach a density matrix approach is used.

2.1.2 Scattering Amplitude and Cross-Section

We treat scattering as a wave phenomenon that obeys the Schrödinger wave equation. We assume an incoming plane wave of the form

$$\Psi_{in} = e^{i(\mathbf{kr}-\omega t)} \tag{2.4}$$

and separate into a spatial and a time depending part. Now we focus on the spatial part. When the wave scatters at a small charge distribution, spherical waves will evolve from the centre in the form:

$$\Psi_{scatt} = f(\mathbf{k}_0, \mathbf{k}) \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \tag{2.5}$$

where $f(\mathbf{k}_0, \mathbf{k})$ is the scattering amplitude, $\mathbf{k}_0$ and $\mathbf{k}$ are the incident and the outgoing wave vector and $|\mathbf{r}-\mathbf{r}'|$ is the distance from the scattering centre. To obtain the scattering amplitude we have to solve the Schrödinger equation for the assumed scattering potential. Starting with

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r}')\right)\Psi(\mathbf{r}') = E\Psi(\mathbf{r}') \tag{2.6}$$

and the following two definitions:

$$k_0^2 = \frac{2mE}{\hbar^2} \tag{2.7}$$

$$U(\mathbf{r}') = \frac{2mV(\mathbf{r}')}{\hbar^2} \tag{2.8}$$

the equation is transformed to the following form:

$$\left(\nabla^2 + k_0^2\right)\Psi(\mathbf{r}') = U(\mathbf{r}')\Psi(\mathbf{r}') \tag{2.9}$$

The next step to find a solution is to find the Green's function for the case of a point scatterer at the origin:

$$\left(\nabla^2 + k_0^2\right) G(\mathbf{r}, \mathbf{r}') = \delta(\mathbf{r}') \tag{2.10}$$

The Green's function for this case is known:

$$G\left(\mathbf{r}, \mathbf{r}'\right) = -\frac{1}{4\pi} \frac{e^{ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} \tag{2.11}$$

and by integration we get:

$$\Psi_{scatt}(\mathbf{r}) = \int U(\mathbf{r}')\Psi(\mathbf{r}')G(\mathbf{r}, \mathbf{r}')d\mathbf{r}' \tag{2.12}$$

Finally by adding the unscattered component,

$$\Psi(\mathbf{r}) = e^{i\mathbf{k}_0\mathbf{r}} + \frac{2m}{\hbar^2} \int V(\mathbf{r}')\Psi(\mathbf{r}')G(\mathbf{r}, \mathbf{r}')d\mathbf{r}' \tag{2.13}$$

it leads to the Lippmann-Schwinger equation.
If we are interested in the scattered current dI_{scatt} that passes through the area $dS = r^2 d\Omega$ we get

$$dI_{scatt} = j_{scatt} r^2 d\Omega = j_{in} d\sigma \tag{2.14}$$

Which means the part of a parallel incidenting beam passing through $d\sigma$ that is scattered into a cone of solid angle $d\Omega$. With $j_{scatt} = j_{in}|f(\mathbf{k}_0, \mathbf{k})|^2/r^2$ the ratio known as differential cross section is calculated.

$$\frac{d\sigma}{d\Omega} = |f(\mathbf{k}_0, \mathbf{k})|^2 \tag{2.15}$$

2.1.2.1 Born Approximation

For weak scattering events a useful approximation for 2.13 is to insert the incoming plane wave

$$\Psi_{in} = e^{i\mathbf{k}_0\mathbf{r}'} \tag{2.16}$$

on the right side. This is known as the born approximation. Further we assume that the detector is far from the scattering centre so that $|\mathbf{r} - \mathbf{r}'|$ is equal to $|\mathbf{r}|$. After some transformations and substitution with:

$$\Delta\mathbf{k} = \mathbf{k} - \mathbf{k}_0 \tag{2.17}$$

we obtain the approximate form:

$$\Psi(\mathbf{r}) = e^{i\mathbf{kr}} - \frac{m}{2\pi\hbar^2}\frac{e^{i\mathbf{kr}}}{|\mathbf{r}|}\int V(\mathbf{r}')e^{-i\Delta\mathbf{kr}'}d\mathbf{r}' \tag{2.18}$$

If we compare the second term with Eq. 2.5 it leads to an equation for the scattering amplitude.

$$f(\Delta\mathbf{k}) = -\frac{m}{2\pi\hbar^2}\int V(\mathbf{r}')e^{-i\Delta\mathbf{kr}'}d\mathbf{r}' \tag{2.19}$$

It is noteworthy, that the scattering amplitude is proportional to the Fourier transform of the scattering potential. If $V(\mathbf{r})$ is the potential of a single atom, $f(\Delta\mathbf{k})$ is called the atomic form factor.

2.1.3 Scattering in a Crystal

Like the atomic form factor describes the interference between the beam and the electrons in an atom, the structure factor describes the interference between the beam and the atoms in one unit cell. The whole interaction of all unit cells in a crystal is further described by the crystal shape factor.

2.1.4 Potential

Traditionally the electron scattering factor $f_e(q)$ of the first Born approximation is tabulated. With $\mathbf{q} = \frac{1}{2\pi}(\mathbf{k} - \mathbf{k}_0)$.Although the Born approximation is too inaccurate to calculate the electron scattering, it is useful to calculate the scatterer potential. Like we saw in Eq. 2.19 the scattering factor is the Fourier transform of the potential times some constants. The bonding effects in the

solid produce an error of magnitude 5% to 10% compared to an isolated single atom[7]. The electron scattering factors were calculated using a Dirac-Fock calculation. The values of $f_e(q)$ are tabulated in a file and are provided with the Kirkland TEMSIM source code. An elegant way to avoid the handling of the large amount of data is to parametrize the function. To do this two functions are used. First a function of Lorentzian form which fits well at high scattering angles. Second a function of Gaussian form for low scattering angles:

$$f_e(q) = \sum_{i=1}^{N_L} \frac{a_i}{q^2 + b_i} + \sum_{i=1}^{N_G} c_i \exp\left(-d_i q^2\right) \tag{2.20}$$

To fit the parameters best a least square fit was performed. The parameters are listed in the following form:

Z= 14,	chisq= 0.094315		
a1	b1	a2	b2
a3	b3	c1	d1
c2	d2	c3	d3

Rewriting Eq. 2.19 leads to:

$$f_e(\mathbf{q}) = \frac{1}{2\pi e a_0} \int V(\mathbf{r}) \exp(2\pi i \mathbf{q} \cdot \mathbf{r}) d\mathbf{r} \tag{2.21}$$

where $a_0 = \frac{\hbar^2}{m_0 e^2}$ is the first Bohr radius. Now we take the Fourier transform:

$$V(\mathbf{r}) = 2\pi a_0 e \int f_e(q) \exp(-2\pi \mathbf{q} \cdot \mathbf{r}) d\mathbf{q} \tag{2.22}$$

We now insert the parametric form of $f_e(q)$ and get:

$$\begin{aligned} V(\mathbf{r}) = {} & 2\pi^2 a_0 e \sum_i \frac{a_i}{r} \exp\left(-2\pi r \sqrt{b_i}\right) + \\ & + 2\pi^{5/2} a_0 e \sum_i c_i d_i^{-3/2} \exp\left(-\frac{\pi^2 r^2}{d_i}\right) \end{aligned} \tag{2.23}$$

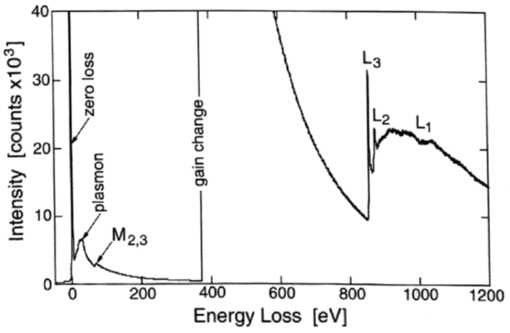

FIGURE 2.1: Energy loss with oxygen K-edges from various manganese oxides[1]

With further calculation we get the projected atomic potential

$$V_z(\mathbf{x}) = \int_{-\infty}^{\infty} V(\mathbf{r})dz = \\ = 4\pi^2 a_0 e \sum_i a_i K_0\left(2\pi r\sqrt{b_i}\right) + 2\pi^2 a_0 e \sum_i \frac{c_i}{d_i} \exp\left(-\frac{\pi^2 r^2}{d_i}\right) \quad (2.24)$$

where $K_0(x)$ is the modified Bessel function.

2.2 Inelastic Scattering

When a electron scatters inelastically it transfers energy to the sample. The most important processes in order of increasing energy loss are phonon creation, plasmon excitation and core electron excitation. In the last case the energy loss is transfered to a core electron. The core electron either jumps from one core state to another or simply leaves the atom. In the latter case an ionized atom remains. Figure 2.1 shows at the left the zero loss peak from electrons that pass through the specimen without interaction, the plasmon peak and the M- and L-edges.

Compared to plasmon excitations, the cross-sections are relatively small for inner shell ionizations. They become even smaller as the energy loss increases. To obtain strong intensities lower energy losses are therefore preferable.

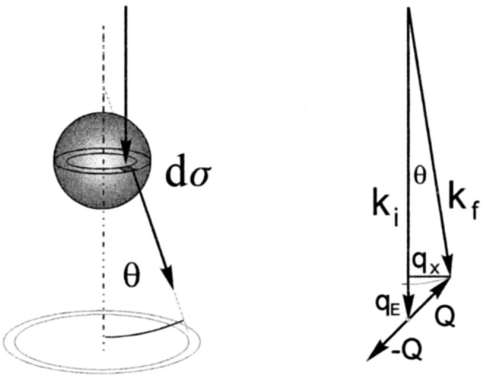

FIGURE 2.2: Inelastic scattering geometry.

2.2.1 Mixed Dynamic Form Factor

The inelastic scattering cross section is the extension of the elastic cross section in Eq. 2.15 to changing energies. In the first Born approximation, and for an incident plane wave, the double differential scattering cross section is [11]

$$\frac{\partial^2\sigma}{\partial E\partial\Omega} = \frac{\gamma^2}{q^4a_0^2}\frac{k}{k_0}S(\mathbf{q},\mathbf{q},E), \tag{2.25}$$

where a_0 is the Bohr radius, and γ is a relativistic factor. $S(\mathbf{q},\mathbf{q},E)$ is the dynamic form factor. In our case we are interested in its more general form $S(\mathbf{q},\mathbf{q'},E)$ called the mixed dynamic form factor (MDFF). For a n-electron system the MDFF reads[11]

$$\begin{aligned} S(\mathbf{q},\mathbf{q'},E) &= \delta(E_i - E_f + E)\sum_i p_i \sum_f (1-p_f). \\ &\quad \cdot \left\langle i \left| \sum_j^n e^{i\mathbf{q}\mathbf{R}_j} \right| f \right\rangle \left\langle f \left| \sum_j^n e^{-i\mathbf{q'}\mathbf{R}_j} \right| i \right\rangle \end{aligned} \tag{2.26}$$

E_i and E_f are the energies of the initial and the final state. p_i and p_f are the occupation probabilities of the initial and final state. R_j are the atom positions.

Chapter 3

Calculation Methods - Wave Solutions

3.1 Multislice Solution with Inelastic Scattering

The crystal can end up in different final states. This leads to an entanglement between final crystal states and electron states. To describe this event it is necessary to apply a density matrix approach. This can be viewed as a sum over many amplitudes of different events that can happen in the crystal in a path integral point of view[11, 12]. For an inelastic scattering event of a single atom at depth d the projection approximation is adopted like this:

$$\rho_{im}(\mathbf{r}, \mathbf{r'}) = T_{mic} T_{el,t-d} T_{inel,d} T_{el,d} \rho_{in}(\mathbf{r},\mathbf{r'}) \tag{3.1}$$

while the intensity in one plane is given by:

$$I_{im} = \rho_{im}(\mathbf{x},\mathbf{x}) \tag{3.2}$$

All the operators act from right to left on the incoming density matrix of the probe ρ_{in}. $\mathbf{x}$ are the coordinates in the respective planes of the microscope.

The first operator acting on the incoming density matrix is $T_{el,d}$. It describes the elastic interaction between the crystal and the electron until it reaches depth d. In this depth the inelastic interaction occurs described by $T_{inel,d}$. $T_{el,t-d}$ finally propagates the density matrix elastically to the exit plane of the crystal. The last acting operator T_{mic} considers the optical effects of the microscope on the density matrix.
Under the assumption of a perfectly coherent probe function ρ_{in} is written as the product state:

$$\rho_{in}(\mathbf{x},\mathbf{x'}) = \phi_{in}(\mathbf{x})\phi^*_{in}(\mathbf{x'}) \tag{3.3}$$

Then we calculate the elastic scattered wave function at depth d. In our case this is done by multislice calculation.

$$\rho_d(\mathbf{x},\mathbf{x'}) = \phi_d(\mathbf{x})\phi^*_d(\mathbf{x'}) = T_{el,d}\rho_{in}(\mathbf{x},\mathbf{x'}) \tag{3.4}$$

Since we treat only elastic scattering this is still a pure state with

$$\phi_d(\mathbf{x}) = \mathtt{T}_d[\phi_{in}] \tag{3.5}$$

while $\mathtt{T}$ is the operator that propagates a wave function from the entrance plan to depth d.
Now we reach the plane in which the inelastic scattering occurs. The next operator acts as a multiplicative kernel in real space:

$$\rho_{d,inel}(\mathbf{x},\mathbf{x'}) = T_{inel}(\mathbf{x},\mathbf{x'})\rho_d(\mathbf{x},\mathbf{x'}) \tag{3.6}$$

For a single inelastic scattering event Schattschneider derived the mixed dynamic object spectrum (MDOS) [6] as the first term in the Rayleigh expansion:

$$MDOS(\mathbf{q},\mathbf{q'}) = f(Q)f(Q')\frac{\mathbf{q}\cdot\mathbf{q'} + q_E^2}{Q^3 Q'^3} \tag{3.7}$$

where $q^2 = q_x^2 + q_y^2$,, $q'^2 = q'^2_x + q'^2_y$, $Q = \sqrt{q_x^2 + q_y^2 + q_E^2}$, $Q' = \sqrt{q'^2_x + q'^2_y + q_E^2}$.

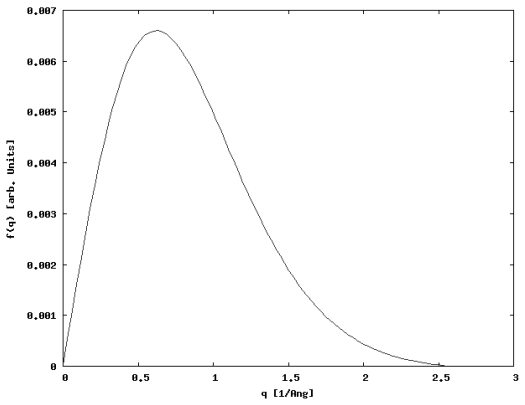

FIGURE 3.1: Angular part $f(Q)$ for Si L_{23} calculated with Clementi-Raimondi wave functions

$q_E = k - k_0$ is the negative difference of radii of the Ewald spheres before and after inelastic scattering. For small energy losses $E \ll E_0$ we have approximately[6]:

$$q_E \approx -k_0 E/2E \tag{3.8}$$

The function $f(Q)$ is shown in Figure 3.1. For small q this function is linear in q. This is called the dipole approximation:

$$MDOS_{dip}(\mathbf{q},\mathbf{q'}) = \frac{\mathbf{q}\cdot\mathbf{q'} + q_E^2}{Q^2 Q'^2} \tag{3.9}$$

and we assume that the inelastic scattering is independent on the incoming direction of the wave vector, and it only depends on the momentum transfer q. This approximation is realistic for a single atom. For a crystal there might be directional bonding effects in the final states which can be calculated with an electronic structure code.
Fourier transforming this according to the rules described in [6] gives the inelastic scattering kernel T_{inel}

$$T_{inel}(\mathbf{x},\mathbf{x'}) = FT_{\mathbf{q},\text{-}\mathbf{q'}}[MDOS_{inel}(\mathbf{q},\mathbf{q'})] \tag{3.10}$$

Calculating component by component results in:

$$W_x(\mathbf{x}) = \int \frac{q_x}{\mathbf{q}^2 + q_E^2} e^{2\pi i \mathbf{q}\cdot\mathbf{x}} dq^2 \tag{3.11}$$

$$W_y(\mathbf{x}) = \int \frac{q_y}{\mathbf{q}^2 + q_E^2} e^{2\pi i \mathbf{q}\cdot\mathbf{x}} dq^2 \tag{3.12}$$

$$W_z(\mathbf{x}) = q_E \int \frac{1}{\mathbf{q}^2 + q_E^2} e^{2\pi i \mathbf{q}\cdot\mathbf{x}} dq^2 \tag{3.13}$$

So we can write

$$T_{inel}(\mathbf{x}, \mathbf{x'}) = W_x(\mathbf{x}) W_x^*(\mathbf{x'}) + W_y(\mathbf{x}) W_y^*(\mathbf{x'}) + W_z(\mathbf{x}) W_z^*(\mathbf{x'}) \tag{3.14}$$

The integrals can be simplified as shown in [6]:

$$FT_q \left[\mathbf{Q} \frac{f(Q)}{Q^3} \right] = iv\mathbf{e}_x + w\mathbf{e}_z \tag{3.15}$$

with

$$v = v(|\mathbf{x}|) = 2\pi \int_0^\infty \frac{q^2 J_1(q|\mathbf{x}|) f(Q)}{Q^3} dq \tag{3.16}$$

and

$$w = w(|\mathbf{x}|) = 2\pi q_E \int_0^\infty \frac{q J_0(q|\mathbf{x}|) f(Q)}{Q^3} dq \tag{3.17}$$

where $\mathbf{e}_x$ is the unit vector in direction of th real space vector $\mathbf{x}$ in the plane, and J are Bessel functions. Making reference to a fixed coordinate system (ξ, η) in the plane at depth d, we put

$$\mathbf{e}_x = \mathbf{e}_\xi \cos\alpha + \mathbf{e}_\eta \sin\alpha \tag{3.18}$$

so that

$$\begin{aligned} FT_q \left[\mathbf{Q} \frac{f(Q)}{Q^3} \right] &= iv\cos\alpha \mathbf{e}_\xi + iv \sin\alpha \mathbf{e}_\eta + w\mathbf{e}_z \\ &= iv\frac{\xi}{|\mathbf{x}|}\mathbf{e}_\xi + iv\frac{\eta}{|\mathbf{x}|}\mathbf{e}_\eta + w\mathbf{e}_z \end{aligned} \tag{3.19}$$

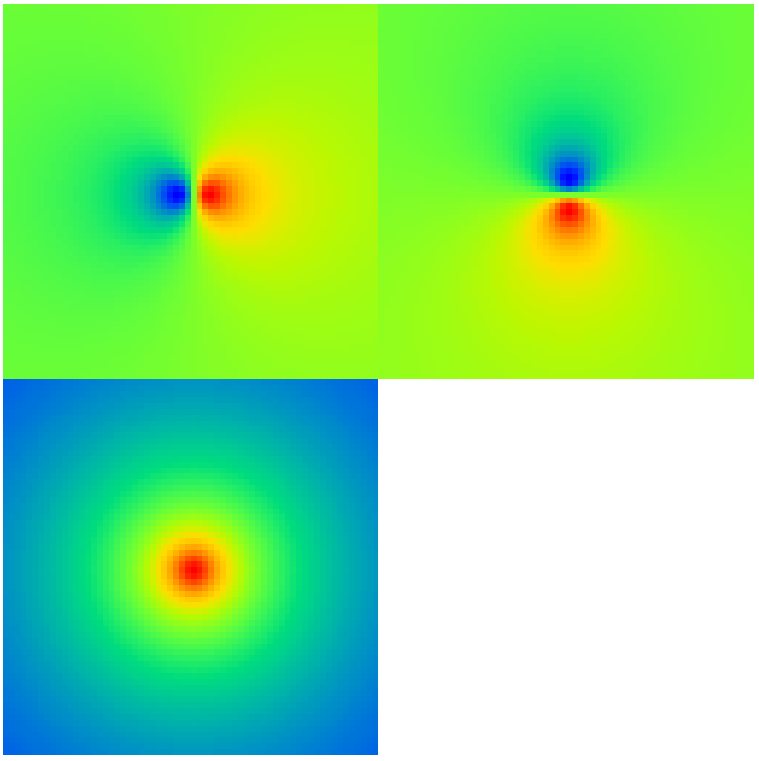

FIGURE 3.2: Plot of W_x, W_y and W_z calculated for oxygen

By comparison with Eq. 3.10 and 3.14 we see that:

$$\begin{aligned} W_x(\mathbf{x}) &= iv(\mathbf{x})\frac{\xi}{|\mathbf{x}|} \\ W_y(\mathbf{x}) &= iv(\mathbf{x})\frac{\eta}{|\mathbf{x}|} \\ W_z(\mathbf{x}) &= w(\mathbf{x}) \end{aligned} \tag{3.20}$$

Applying this propagator on the density matrix at depth d for atom i at x_i gives

$$\rho_{d,inel} = \phi_{W_{x,i,d}}(\mathbf{x})\phi^*_{W_{x,i,d}}(\mathbf{x}') + \phi_{W_{y,i,d}}(\mathbf{x})\phi^*_{W_{y,i,d}}(\mathbf{x}') + \phi_{W_{z,i,d}}(\mathbf{x})\phi^*_{W_{z,i,d}}(\mathbf{x}') \tag{3.21}$$

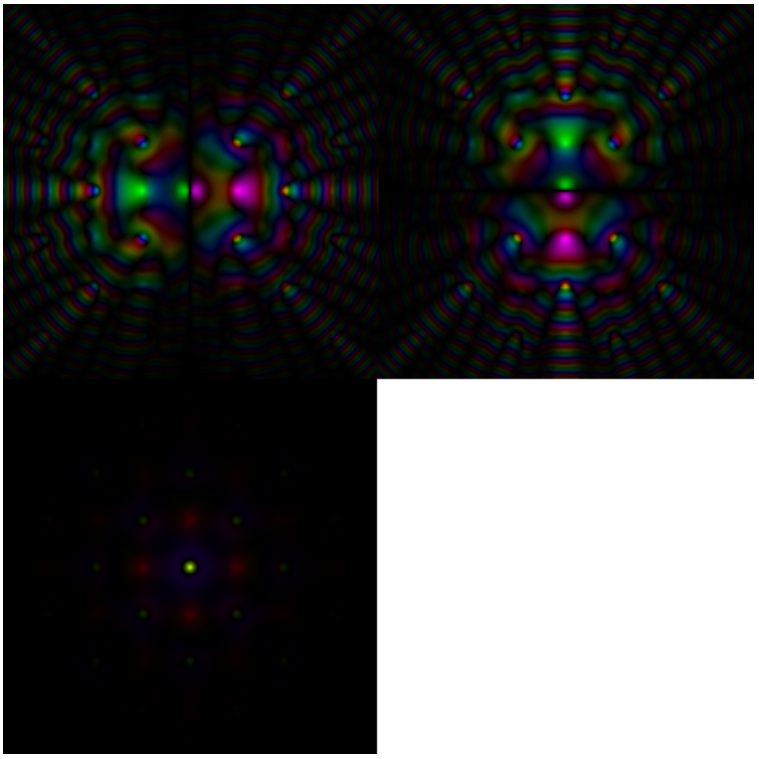

FIGURE 3.3: Inelastic scattering event at slice 24 elastically propagated to the exit plane (slice 160). From left to right: W_x, W_y, W_z. 4x4 UC

with

$$\phi_{W_{x,i,d}}(\mathbf{x}) = W_x(\mathbf{x} - \mathbf{x}_i)\phi_d(\mathbf{x}) \tag{3.22}$$

$$\phi_{W_{y,i,d}}(\mathbf{x}) = W_y(\mathbf{x} - \mathbf{x}_i)\phi_d(\mathbf{x}) \tag{3.23}$$

$$\phi_{W_{z,i,d}}(\mathbf{x}) = W_z(\mathbf{x} - \mathbf{x}_i)\phi_d(\mathbf{x}) \tag{3.24}$$

Now we propagate expression 3.21 to the exit plane t with another elastic propagator. Since the density matrix is nicely split into an incoherent sum (W_x, W_y and W_z) we can apply the elastic propagator separately on each coherent term of the sum

$$\phi_{W_{x,i,t}}(\mathbf{r}) = T_{t-d}[\phi_{W_{x,i,t}}(\mathbf{r})] \tag{3.25}$$

and equal for the others. Then the density matrix at the exit surface is:

$$\rho_{exit}(\mathbf{x},\mathbf{x}') = \phi_{W_{x,i,t}}(\mathbf{x})\phi^*_{W_{x,i,t}}(\mathbf{x}') + \phi_{W_{y,i,t}}(\mathbf{x})\phi^*_{W_{y,i,t}}(\mathbf{x}') + \phi_{W_{z,i,t}}(\mathbf{x})\phi^*_{W_{z,i,t}}(\mathbf{x}') \tag{3.26}$$

Next the action of the objective lens must be applied to each of the 2 wave functions separately.

$$\phi_{W_{x,im}} = T_{mic}[\phi_{W_{x,i,t}}] \tag{3.27}$$

And similar for the other two. The intensity in the image plane is the diagonal element of the density matrix $\rho_{im}(\mathbf{x},\mathbf{x})$, it is obtained as the sum over the 3 squared wave functions

$$I_{im}(x) = \rho_{im}(\mathbf{x},\mathbf{x}) = |\phi_{W_{x,im}}(\mathbf{x})|^2 + |\phi_{W_{y,im}}(\mathbf{x})|^2 + |\phi_{W_{z,im}}(\mathbf{x})|^2 \tag{3.28}$$

Finally the intensities of all atom positions in all intermediate planes from 0 to t are added to obtain the overall intensity measured in the detector plane.

3.2 Conventional Multislice Solution

In the paragraph above we looked in detail at the scattering event in depth d. What is missing is the multislice approach. We will apply it to gain the wavefunction ϕ_d at depth d in the first step, and to calculate the wavefunctions evolving at depth d propagated to the exit plane at depth t in the next step. The method was developed in 1957 by Cowley and Moodie[13]. In the following we will outline the approximations that lead to basically only two operations that allow to calculate the wavefunction at a specific depth[1].

3.2.1 Simplified Wave Equation for Fast Electrons

The microscope works in an energy domain which is high enough to demand a relativistic treatment of the electron. Namely the usage of the Dirac wave equation which contains the spin of an electron in its description. In a simplified way we keep the non-relativistic Schrödinger wave equation and take into

[1] The rest of this chapter is a summary of [1, 7]

account the relativistic mass and wavelength effects. This approach provides reasonable exact solutions for lower energies and introduces small errors at energies of order 1000 keV or higher [7]. To simplify the calculation even more we start again with the time independent form of the Schrödinger equation. Presuming, that we look at a stationary system.

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r})\right)\Psi(\mathbf{r}) = E\Psi(\mathbf{r}) \tag{3.29}$$

Like explained above we substitute the mass by the relativistic mass.

$$m = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}m_0 \tag{3.30}$$

The electron wave function in the specimen in a high energy approximation is further expressed as[2]:

$$\Psi(\mathbf{r}) = \phi(\mathbf{r})\exp(2\pi ikz) \tag{3.31}$$

This splits the wave function into a fast traveling plane wave in z-direction and a small specimen perturbation that varies slowly with position z. Since $k^2 = k_x^2 + k_y^2 + k_z^2$ is only exact for the unscattered case where k_x and k_y are small. The total energy of the electron is

$$E = \frac{h^2k^2}{2m} \tag{3.32}$$

For the following calculations we define the Laplacian to act only in the x-y plane:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.33}$$

We rewrite the derivative on the left side of Eq. 3.29 and insert Eq. 3.31

$$\left(\Delta + \frac{\partial^2}{\partial z^2}\right)\Psi(\mathbf{r}) = \exp(2\pi ikz)\Delta\phi(\mathbf{r}) + \frac{\partial^2}{\partial z^2}\left(\phi(\mathbf{r})\exp(2\pi ikz)\right) \tag{3.34}$$

[2]It is usual in literature on optics to define $k = \frac{1}{\lambda}$ and this is the way it is used in this context

We are now interested in the derivative with respect to z. The first derivative is:

$$\frac{\partial}{\partial z}\left(\phi(\mathbf{r})\exp(2\pi ikz)\right) = \exp(2\pi ikz)\left(\frac{\partial}{\partial z}\phi(\mathbf{r}) + 2\pi ik\phi(\mathbf{r})\right) \tag{3.35}$$

The second derivative is:

$$\begin{aligned}
&\frac{\partial^2}{\partial z^2}\left(\phi(\mathbf{r})\exp(2\pi ikz)\right) = \\
&= \exp(2\pi ikz)\left(\frac{\partial^2}{\partial z^2}\phi(\mathbf{r}) + 4\pi ik\frac{\partial}{\partial z}\phi(\mathbf{r}) + (2\pi ik)^2\phi(\mathbf{r})\right) \\
&= \exp(2\pi ikz)\left(\frac{\partial^2}{\partial z^2}\phi(\mathbf{r}) + 4\pi ik\frac{\partial}{\partial z}\phi(\mathbf{r})\right) - 4\pi^2k^2\Psi(\mathbf{r})
\end{aligned} \tag{3.36}$$

The result is now substituted back into Eq. 3.29 and the factor $\exp(2\pi izk)$ is droped. So we get:

$$-\frac{\hbar^2}{2m}\left(\Delta + \frac{\partial^2}{\partial z^2} + 4\pi ik\frac{\partial}{\partial z} + \frac{2mV(\mathbf{r})}{\hbar^2}\right)\phi(\mathbf{r}) = 0 \tag{3.37}$$

Since ϕ is changing only slowly in respect of z and since k is huge the following approximation is used:

$$\left|\frac{\partial^2}{\partial z^2}\phi\right| << \left|k\frac{\partial}{\partial z}\phi\right| \tag{3.38}$$

therefore is

$$\left(\Delta + 4\pi ik\frac{\partial}{\partial z} + \frac{2mV(\mathbf{r})}{\hbar^2}\right)\phi(\mathbf{r}) = 0 \tag{3.39}$$

This approximation is sometimes referred to as ignoring the backscattered electrons. Probably it is more accurate to refer it as paraxial approximation to the Schrödinger equation[7]. It is now possible to rewrite the Schrödinger equation for fast electrons as first order differential equation in z:

$$\frac{\partial}{\partial z}\phi(\mathbf{r}) = \left(\frac{i}{4\pi k}\Delta + i\sigma V(\mathbf{r})\right)\phi(\mathbf{r}) \tag{3.40}$$

where

$$\sigma = \frac{2\pi m}{h^2 k} \tag{3.41}$$

is denoted as interaction parameter.

3.2.2 The Multislice Method

For the numerical calculations we split the crystal into a series of thin slices[3]. For this reason we use $\mathbf{x}$ as coordinates in the slice plane and z as slice depth. This makes sense if we look at the formal operator solution of Eq. 3.40:

$$\phi(\mathbf{x}, z) = \exp\left(\int_0^z \left(\frac{i}{4\pi k}\Delta + i\sigma V(\mathbf{r})\right) dz\right) \phi(\mathbf{x}, 0) \tag{3.42}$$

Starting the calculation from a depth z we get:

$$\phi(\mathbf{x}, z+\delta z) = \exp\left(\int_z^{z+\delta z} \left(\frac{i}{4\pi k}\Delta + i\sigma V(\mathbf{r})\right) dz\right) \phi(\mathbf{x}, z) \tag{3.43}$$

And after integration we simplify to

$$\phi(\mathbf{x}, z+\delta z) = \exp\left(\delta z \frac{i}{4\pi k}\Delta + i\sigma V_z(\mathbf{x})\right) \phi(\mathbf{x}, z) \tag{3.44}$$

where $V_z(\mathbf{x})$ is similar to Eq. 2.24 the projected potential of the crystal between z and $z+\delta z$.

$$V_z(\mathbf{x}) = \int_z^{z+\delta z} V(\mathbf{x}, z) dz \tag{3.45}$$

In the next step we have to bear in mind the formal treatment of exponentials containing operators. Taking the exponential definition of an operator and inserting two operators A and B multiplied by a small real number ϵ gives the following series:

$$\begin{aligned} \exp\Big(\epsilon(A+B)\Big) &= \sum_{\nu=0}^{\infty} \frac{1}{\nu!}\Big(\epsilon(A+B)\Big)^{\nu} \\ &= E + \epsilon(A+B) + \frac{1}{2!}\epsilon^2(A^2 + AB + BA + B^2) + \cdots \end{aligned} \tag{3.46}$$

[3]This chapter is discussed in more details at[7, 13]

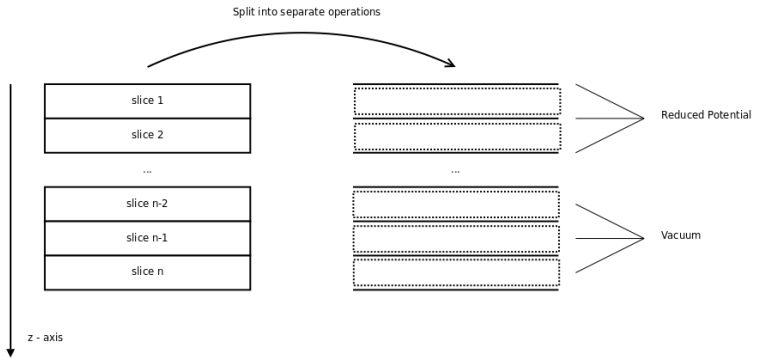

FIGURE 3.4: Multislice scheme

Only for sufficiently small ϵ the approximation

$$\exp\Big(\epsilon(A+B)\Big) \;\cong\; \exp(\epsilon A)\exp(\epsilon B) \tag{3.47}$$

is exact enough. Using this approximation we write Eq. 3.42 in the form:

$$\phi(\mathbf{x};z) = \prod_{j=1}^{n[z]} \exp\left(\frac{i\delta z_j}{4\pi k}\Delta + i\sigma V_z(\mathbf{x})\right)\phi(\mathbf{x},0) \tag{3.48}$$

The next approximation step aims at obtaining two separate operators.

$$\begin{aligned}\phi(\mathbf{x},z+\delta z) &= \exp\left(\delta z\frac{i}{4\pi k}\Delta\right)\exp(i\sigma V_z(\mathbf{x}))\phi(\mathbf{x},z)\\ &= \exp\left(\delta z\frac{i}{4\pi k}\Delta\right)t(\mathbf{x},z)\phi(\mathbf{x},z)\end{aligned} \tag{3.49}$$

For further simplification we calculate the 2D Fourier transformation:

$$\begin{aligned}&FT\left[\exp\left(\delta z\frac{i}{4\pi k}\Delta\right)t(\mathbf{x},z)\phi(\mathbf{x},z)\right] =\\ &= \int\left(\exp\left(\delta z\frac{i}{4\pi k}\Delta\right)t(\mathbf{x},z)\phi(\mathbf{x},z)\exp(2\pi i(k_x x+k_y y)\right)d\mathbf{x}\end{aligned} \tag{3.50}$$

Manipulating the terms and differentiation (details can be found in the appendix) leads to:

$$\begin{aligned} & FT\left[\exp\left(\delta z\frac{i}{4\pi k}\Delta\right)t(\mathbf{x},y)\phi(\mathbf{x},z)\right] = \\ = \;& \exp\left(\frac{-i\pi\delta z}{k}(k_x^2+k_y^2)\right)FT[t(\mathbf{x},z)\phi(\mathbf{x},z)] \qquad (3.51)\\ = \;& FT[p(\mathbf{x},z)]FT[t(\mathbf{x},z)\phi(\mathbf{x},z)] \qquad (3.52)\end{aligned}$$

Now we explicitly calculate $p(\mathbf{x},z)$ by inverse Fourier transformation:

$$\begin{aligned} FT[p(\mathbf{x},z)] &= \exp\left(\frac{-i\pi\delta z}{k}(k_x^2+k_y^2)\right) \qquad (3.53)\\ p(\mathbf{x},z) &= FT^{-1}\left[\exp\left(\frac{-i\pi\delta z}{k}(k_x^2+k_y^2)\right)\right] \\ &= \frac{k}{i\delta z}\exp\left(\frac{i\pi k}{\delta z}(x^2+y^2)\right) \qquad (3.54)\end{aligned}$$

Since a multiplication in Fourier space converts to a convolution in real space Eq. 3.49 becomes:

$$\phi(\mathbf{x},z+\delta z) = p(\mathbf{x},\delta z)\otimes[t(\mathbf{x},\delta z)\phi(\mathbf{x},z)] + \mathcal{O}(\delta z^2) \qquad (3.55)$$

Ordering all the appearing operators by numbers (corresponding to slices instead of depth):

$$\phi_{n+1}(\mathbf{x}) = p_n(\mathbf{x})\otimes[t_n(\mathbf{x})\phi_n(\mathbf{x})] + \mathcal{O}(\delta z^2) \qquad (3.56)$$

with

$$\begin{aligned} p_n(\mathbf{x}) &= \frac{k}{i\delta z_n}\exp\left(\frac{i\pi k}{\delta z_n}(x^2+y^2)\right) \qquad (3.57)\\ t_n(\mathbf{x}) &= \exp(i\sigma V_{z_n}(\mathbf{x})) \qquad (3.58)\end{aligned}$$

We will see in the further progress that it is useful for our numerical approach to separate Eq. 3.53 concerning its variables k_x and k_y:

$$P_n(k_x) = \exp(-i\pi\lambda k_x^2\delta z_n),\; P_n(k_y) = \exp(-i\pi\lambda k_y^2\delta z_n) \qquad (3.59)$$

If we like to combine our results in one operator it has the form

$$T_{PT,j} = \mathcal{F}^{-1}\hat{P}_j\mathcal{F}\hat{t}_j, \tag{3.60}$$

were $\hat{t}_j$ represents the entrywise product with function 3.58, $\mathcal{F}$ the two dimensional Fourier transform and $\hat{P}_j$ the entrywise product of term 3.57 in its Fourier transformed form. $T_{el,d}$ in Eq. 3.1 can now be written as the application of operators over all slices j in the distance d:

$$T_{el,d} = T_{PT,n}T_{PT,n-1}\cdots T_{PT,2}T_{PT,1} \tag{3.61}$$

3.3 Image Formation

3.3.1 Spherical Aberration

In a TEM the objective lens is a coil with large current flow to produce a magnetic field. Symmetry and shape of the magnetic field are determined by Maxwells equations. In practice lenses are afflicted with errors. The effect of the aberration is to shift the phase of each frequency component by a different amount. Since only a small portion of the specimen is imaged, off-axis aberrations can be neglected. So only low order effects like spherical aberration C_s are important. The net phase error due to spherical aberration and defocus is [7]:

$$\chi(\alpha) = \frac{2\pi}{\lambda}\left(\frac{1}{4}C_s\alpha^4 - \frac{1}{2}\Delta f\alpha^2\right) \tag{3.62}$$

C_s produces an error proportional to the fourth power of the focus angle α. Like shown in Fig. 3.5 defocusing the lens moves the ray crossing (circle of least confusion) to the right. Although the spherical aberration C_s is fixed by the microscope it is possible to increase the resolution with smaller wavelength λ which is given by the acceleration voltage. The main problem in high-resolution imaging is to optimize the defocus to provide the best resolution for an image.

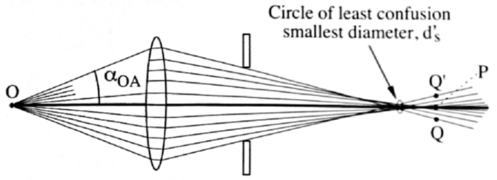

FIGURE 3.5: Lens with positive spherical aberration, showing a closer focus for off-axis rays[1]

To obtain the sharpest image of a point object we look at the following equation:

$$\psi(x) = \int_{-\infty}^{\infty} e^{i\Delta k x} e^{i\chi(\Delta k)} d\Delta k \tag{3.63}$$

The equation shows the Fourier representation of a wave amplitude. To achieve the best point resolution, namely to obtain the Dirac δ-function we would have to set $\chi(\Delta k) = 0$. An analogous consideration tells us that the Fourier representation of a point function consists of a wide range of frequencies Δk. So the problem of picking an optimum defocus value is equal to choosing $\chi(\Delta k)$ in such a way that it has a constant value over a wide range in Δk. The standard criterion for point-to-point resolution uses the range $-3\pi/4 < \chi(\Delta k) < 0$. This range is obtained with the so called Scherzer defocus:

$$\Delta f = \sqrt{\frac{3C_s\lambda}{2}} \tag{3.64}$$

The maximum Δk for which the coherency is preserved is the second root. Its inverse is called the Scherzer resolution:

$$r_{Sch} = \sqrt[4]{\frac{C_s\lambda^3}{6}} \tag{3.65}$$

For weak phase objects the most important part of the transfer function $e^{i\chi(\Delta k)}$ is its imaginary component $\sin(\chi(\Delta k))$. This is called the contrast transfer function (CTF). The CTF in Fig. 3.6 is strongly alternating for increasing Δk. This grounds on the dominance of the quadratic term in Eq. 3.62 for large Δk. However, errors in microscopes limit the coherency for higher values of Δk. This is known as information limit and leads to a damping of the CTF

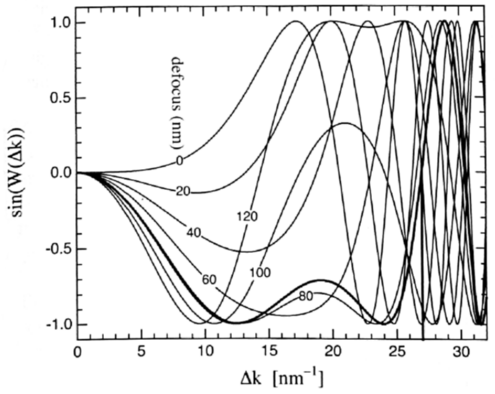

FIGURE 3.6: Contrast transfer function for the lens conditions ($C_s = 2.3$mm, $\lambda = 0.001968$nm, 300 keV) for various values of defocus. [2]

with increasing Δk. The lateral coherence length perpendicular to the optical axis of the illumination electron wave function is approximately given by [7]

$$\Delta x_{coh} \approx \frac{0.16\lambda}{\beta_{max}} \tag{3.66}$$

while β_{max} is the maximum condenser angle. A criteria to classify the image coherency evolves from the comparison of β_{max} with the maximum objective angle α_{max}:

$$\begin{aligned} \beta_{max} &\ll 0.16\alpha_{max} \cdots \text{coherent image} \\ \beta_{max} &\gg 0.16\alpha_{max} \cdots \text{incoherent image} \end{aligned}$$

In between these ranges the image is called partial coherent[7]. We will distinguish between coherent and the partial coherent image calculation in the image formation process. In the extend of this work we will concentrate on the coherent image calculation[4].

[4]Details about partial coherent image calculation can be found in [7, 14]

3.3.2 Coherent Image Calculation

In the case that coherent image calculation is sufficient we calculate a more general form of the aberration function Eq. 3.62 [7].

$$\begin{aligned}\chi(k) &= \frac{\pi}{2}C_s\lambda^3k^4 - \pi\Delta f\lambda k^2 \\ &\quad +\pi f_{a2}\lambda k^2\sin(2(\phi-\phi_{a2})) + \frac{2\pi}{3}f_{a3}\lambda^2k^3\sin(3(\phi-\phi_{a3}))\end{aligned} \tag{3.67}$$

The additional parameters are the two and three fold astigmatisms f_{a2} and f_{a3} and their azimuthal orientations ϕ_{a2} and ϕ_{a3}. By calculation of

$$H_0(\mathbf{k}) = \exp(-i\chi(\mathbf{k})) \tag{3.68}$$

and Fourier back transformation into real space we obtain the complex point spread function (PSF) $h_0(\mathbf{x})$. By convolution of the respective wavefunction with the point spread function we obtain the wavefunction at the detector plane. Therefore the effect of the operator T_{lens} in Eq. 3.27 equates the convolution with the PSF.

Chapter 4

Programming

All the programs are written in the C programming language[1]. The main reasons for this choice are its low runtime demand of system resources and its flexible way to allocate memory.

4.1 Outline

The program consists of three main calculations like the mathematical treatment in the previous chapter (Eq. 3.1) indicates. These calculations are further split into smaller parts in the following way:

[1]For compilation the GNU C compiler Ver. 3.4 was used.

The calculation of the evolving wave functions in the scattering event.

1. Reading respectively calculation of $f(Q)$.
2. Calculation of the integrals $v(|\mathbf{x}|)$ and $w(|\mathbf{x}|)$ Eq. 3.16.
3. Calculation of the 2D arrays for $W_x(\mathbf{x})$, $W_y(\mathbf{x})$ and $W_z(\mathbf{x})$ Eq. 3.20
4. Shifting the centres of the previous arrays to match the scattering centre in the wave function.
5. Obtain the wave function at the current depth.
6. Calculation of three new wave functions by multiplication of the wave function with the previous arrays.

The multislice propagation of a wavefunction through a number of slices.

1. Calculation of a 2D array representing the reduced potential $V_z(\mathbf{x})$ Eq. 3.45 for each used layer.
2. Symmetrically bandwidth limit the potentials like described in [7]
3. Calculation of an 2D array representing the transmission function $t(\mathbf{x})$ Eq. 3.58 for each used layer.
4. Calculation of two 1D arrays representing the propagation functions Eq. 3.59 $P(k_x)$ and $P(k_y)$
5. Entry wise multiplication of the wave function with the transmission function that represents the layer.
6. Forward FFT into k-space of the previous result.
7. Entry wise multiplication of the wave function with the propagation functions $P(k_x, 0)$ and $P(0, k_y)$.
8. Backward FFT into x-space.
9. Repetition of steps 5 to 8 until the wave function reaches the desired depth.

The calculation of the intensity distribution at the detector plane.
1. Calculation of a 2D array representing the transfer function of the objective lens Eq. 3.68. 2. Forward FFT into k-space of the wave function at the exit plane. 3. Entry wise multiplication of the wave function with the transmission function. 4. Limitation of the wave function in k-space to account the objective aperture size. 5. Backward FFT into x-space. 6. Calculate the intensity of the wave function. 7. Sum the previous intensity to the overall image intensity. 8. Repetition of the steps 2 to 7 for every wave function at the exit plane.

The listed calculation steps are separated into three programs. The purpose of this split-up is to allow to vary as many parameters as possible without having to redo all the calculations from the beginning. Of course this approach is limited since usability has to be maintained, calculations have to be grouped and the need of slow memory access minimized. Each program reads the required parameters from the configuration files and stores its results to a file that is read by the next program.

4.2 `ixchel_atompot` Program

The `ixchel_atompot` program calculates and stores the reduced potential for every given layer into a file . The reduced potential only depends on the set up of atoms in the specific layer and the pixel dimension of its array representation[2].

[2]Basically the pixel dimension of the reduced potential could differ from the dimension of the wave function. This was not implemented so that the pixel dimensions match.

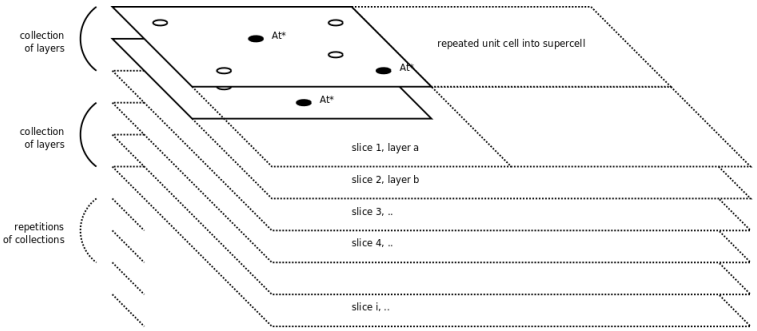

FIGURE 4.1: Figure shows how the crystal is separated into slices. The black dots mark the atoms where inelastic scattering occurs

4.3 `ixchel` Program

In the `ixchel` program a lot of bookkeeping is going on. It propagates the incoming wave function elastically with the multislice method to the crystal exit plane. After each slice a snapshot of the wave function is stored. In slices that contain the defined scattering atom the three evolving wave functions are calculated and propagated to the crystal exit plane by using the multislice method again. Since the multislice calculation of one wave function is independent from other multislice calculations it is quite simple to parallelize these calculations.

4.3.1 Parsing the Crystal Structure

In the crystal structure file the crystal structure is defined. The character of multi slice calculation is to sequence through a set of layers. Each layer holds a specific constellation of atom positions in one crystal plain. Layers can be combined to a sequence of layers. Sequences of layers can be combined with other sequences and other layers and so on. A simple example to combine two layers **a** and **b** is **12(ab)**. This means to repeat the sequence **ab** twelve times resulting in a crystal containing 24 slices. The stacking sequence is defined in the crystal structure file. The first step in the calculation is to parse the stacking sequence. This results in a direct mapping between a slice number and a layer.

4.3.2 Creating and Managing Jobs; Parallel Computing

Unlike forking where the whole program memory is cloned pthread uses the same memory the parent program uses. The library allows the main program to keep track which threads are still running and which are finished.
After having calculated all initially required arrays the program starts to create a list of jobs. This queue (Fig. 4.3) is build in form of an array which contains pointers to structures. For each individual job namely the multislice calculation in Eq. 3.25 a structure is created. The structure in Listing 4.1 contains all the specific informations for this job. A second structure Listing 4.2 exists to pass information that is relevant for every job. While the main program creates new jobs it checks if free cores are available(Fig. 4.2). If so, a new thread is started and the pointer to a free job is provided. When a thread finishes it automatically checks for free jobs and restarts if it finds one. When all jobs are created, the main program waits for the threads to finish. When all jobs are done the threads will finish. The main program waits for the last thread and finishes itself.
Since the calculations run in parallel some care has to be taken to avoid problems. One problem is the freeing of memory. Memory that is needed only by a limited number of jobs has to be freed after all of these jobs are finished. This is implemented with a counter representing the number of jobs still linking to the memory. The second problem is known as race condition.To avoid that two or more threads simultaneously manipulate the same memory addresses mutex locks are used.

```
1  struct ineljob
2  {
3    struct commjob *ptr_masterjob;
4    int islice; // # of slice
5    int delta_nx; // disposition of winel
6    int delta_ny; // disposition of winel
7    float **ptr_wfr; // points to beginning wave function for this depth
8    float **ptr_winel; // points to inelastic winel
9    int atom; // # of atom
10   int pos; // index of atom of this sort
11   int *layer; // array containing layer # of islice
12   int comp; // {0, 1, 2 } => {x, y, z }-component
13   clock_t cput; // required calculation time
14   int state; // 0 ... free, 1 ... in work, >1 ... finished w. success
```

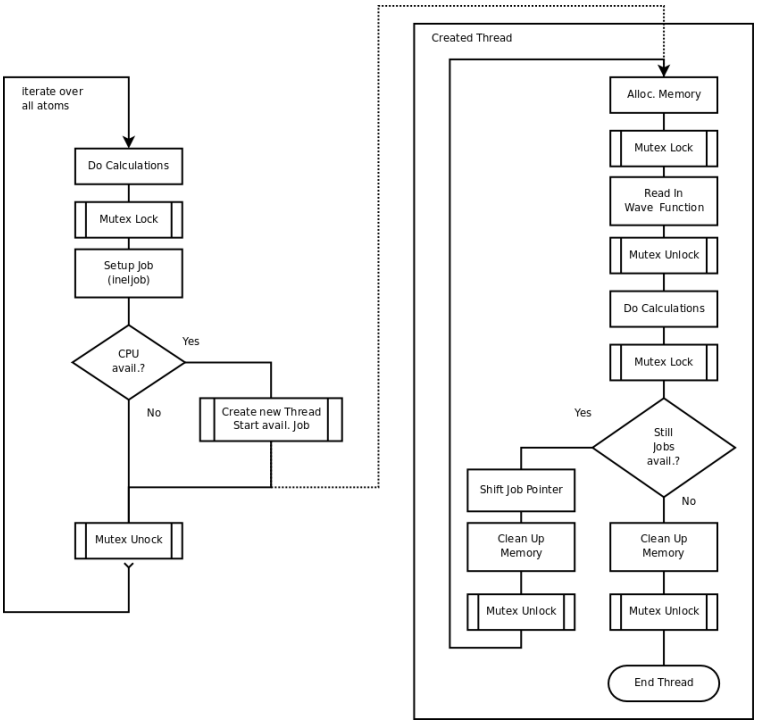

FIGURE 4.2: Parallelization for multi core processors with pthread

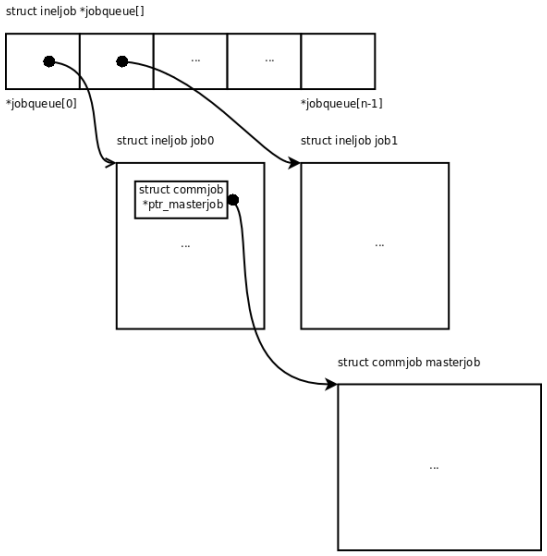

FIGURE 4.3: Figure shows the queue which holds the available jobs

```
15 };
```

Listing 4.1: struct ineljob

```
1  struct commjob
2  {
3    // Do not change order of this lines //
4    pthread_mutex_t job_mutex;
5    int next_job;
6    int last_job;
7    // Do not change order of this lines //
8    pthread_t thread_id[NUM_CPU];
9    float ***wfqueue;
10   int *wfcount;
11   int nx; // size of wave function in pixel
12   int ny;
13   unsigned int *cksums;
14   float **ptr_propxr;
15   float **ptr_propxi;
16   float **ptr_propyr;
17   float **ptr_propyi;
18   float ***ptr_transr;
19   float ***ptr_transi;
20   float *kx2;
21   float *ky2;
22   float k2max;
23   int nout; // quantity of output slices
24   int outslices[NOUTMAX]; // contains the slice numbers for output
25   clock_t cput; // required calculation time
26   int state; // 0 ... free, 1 ... in work, >1 ... finished w. success
27 };
```

Listing 4.2: struct commonjob

4.4 `ixchel_image` Program

The job of the program is to calculate the intensity distribution from the exit plane wavefunctions and sum them up. In the simplest case the convolution of wavefunction and PSF. With little changes it is possible to implement more advanced optics calculations. Like done with the partial coherent mode implemented from the Kirkland package[7]. In difference to the `ixchel` program the calculations were not parallelized. However, it is favourable to run simulations

with different configuration files simultaneously. This can be done with the help of simple scripts (eg. in BASH).

4.5 Crystal Structure Configuration File

The crystal structure configuration file (like seen in Listings 5.1 and 5.3) contains the following information:

- `stack_seq`, The stacking sequence of layers contains the number of repetitions and the layer sequence (represented through their associated letters)
- `NCELL`, defines the replication factors of the layer unit cell. The replication factor gives the number of repetitions of a unit cell along an axis to form a super cell. The repetition of the unit cell generates a "zoom-in" in momentum space.
- `dim`, The real space dimensions in pixels of the supercell. The employed FFT algorithm demands dimension 2 to the power of an integer.
- `layers`, contains a list of layer definitions. The defined layers are mapped to the sequence abc...XYZ according to their appearance. Each layer requires the following information:
  - `filename`, the output file name to which the reduced potential should be saved.
  - `size_uc`, the size of the unit cell is defined by the list of dimensions { x= ...; y= ...; z= ...;} in the unit [Å].
  - `atoms`, contains lists. Each list specifies the atomic number and the atom positions:
    - `Z`, the atomic number.
    - `positions`, holds a number of arrays. Each array has two elements, the first is the x-position, the second the y-position in proportion to the unit cell length.

TABLE 4.1: Parameters defined in the configuration files

Parameter	Used by			Data Type
	`ixchel_atompot`			
		`ixchel`		
			`ixchel_image`	
`stack_seq`	X	X	X	string
`dim`	X	X	X	integer array with 2 elements
`NCELL`	X	X	X	integer array with 3 elements
`layers`	X	X	X	group containing an arbitrary number of groups
`size_uc`	X	X	X	group of 3 float values
`filename`	X	X		string
`atoms`	X	X	X	list of groups
`Z`	X	X	X	integer
`positions`	X	X	X	list of float arrays with 2 elements
`filescatt`		X		string
`Eloss`		X		float
`ctilt`		X		float array with 2 elements
`btilt`		X		float array with 2 elements
`v0`		X	X	float
`Zinel`		X	X	integer
`outslices`		X	X	integer array of arbitrary size
`transopt`			X	boolean
`transwf`			X	boolean
`Cs`			X	float
`df0`			X	float
`sigmaf`			X	float
`dfdelt`			X	float
`aobj`			X	float
`cmode`			X	boolean
`dfa2`			X	float
`dfa2phi`			X	float
`dfa3`			X	float
`dfa3phi`			X	float
`objlx`			X	float
`objly`			X	float
`alpha0`			X	float
`ddf`			X	float

The simulation configuration file contains the following information:

- `filescatt`, the file name from where the inelastic scattering amplitude should be read. The file should contain two rows. The first representing q the second $f(q)$. q has to be calculated like $1/\lambda$ rather than $2\pi/\lambda$ and must have the unit [1/Å].
- `Eloss`, the energy transfered to the core electron in [eV].
- `ctilt`, the tilt of the crystal in x- an y-direction in [mrad].
- `btilt`. the tilt of the electron beam [mrad].
- `v0`, the energy of the incident electron beam in [keV].
- `Zinel`, the atomic number of the atoms specific for the energy loss.
- `outslices`, an array of arbitrary size holds the slice numbers where intermediate results are stored to a file. In contrast to the elastically propagated wavefunction which is saved after each slice, the wavefunctions evolving from the inelastic scattering centres are only saved on given positions. This given positions allow image calculations for the corresponding crystal thickness.
- `transopt`, if this switch is true, the simulation in the `ixchel_image` is simplified. Optical effects are only calculated for atoms in the first unit cell. This switch makes only sense if `transwf` is also true.
- `transwf`, if this switch is true, the simulation in `ixchel` and `ixchel_image` are simplified. Only wavefunctions originating from the first unit cell are taken for the simulation but are replicated, so that the image simulation is done for each unit cell.
- `Cs`, the spherical aberration of the lenses in [mm].
- `df0`, the defocus value in [Å]. Have a look at Eq. 3.64 to calculate the best value.

- `sigmaf`, `dfdelt` the standard deviation and the sampling size of `df0` in [Å].
- `aobj`, the object aperture size in [mrad].
- `dfa2, dfa2phi`, magnitude [Å] and angle [mrad] of 2-fold astigmatism.
- `dfa3, dfa3phi`, magnitude [Å] and angle [mrad] of 3-fold astigmatism.
- `cmode`, coherent image formation if true. Otherwise if only partial coherence is assured `false`. The following two parameters are required for this case.
- `alpha0`, illumination semi-angle in [mrad].
- `ddf`, defocus spread in [Å].

4.6 Usage of the Program

1. Create the crystal configuration file. Table 4.1 shows which settings are relevant for which program.
2. Run the `ixchel_atompot` program. It produces a Tiff file for each defined layer. The allowed command-line options are:

   -c FILE Reads the crystal configuration from FILE. When this option is omitted the program will read from `crystal_struct.cfg`

   -h A brief summary of the command-line options.
3. Create or modify the simulation config file.
4. Run the `ixchel` program. It produces Tiff files for the multislice results at each slice and the multislice results of all evolving wavefunction at the end slice and at the slices specified in `outslices`.

   -c FILE Reads the crystal configuration from FILE. When this option is omitted the program will read from `crystal_struct.cfg`

-i FILE Reads the incoming wavefunction from FILE. When omitted a plane wave is calculated using the parameters in the simulation configuration.

-s FILE Reads the simulation configuration from FILE. When this option is omitted the program will read from `simulation.cfg`

-h A brief summary of the command-line options.

5. Create or modify the simulation config file.

6. Run the `ixchel_image` program. The program produces Tiff files showing the intensity distribution at the crystal exit plane and at the slices specified in `outslices`.

   -c FILE Reads the crystal configuration from FILE. When this option is omitted the program will read from `crystal_struct.cfg`

   -o FILE Writes the intensity distribution at the exit plane to FILE. When omitted the intensity is written to `pixsum.tif`.

   -s FILE Reads the simulation configuration from FILE. When this option is omitted the program will read from `simulation.cfg`

   -h A brief summary of the command-line options.

7. Repeat either steps 1-6, 3-6 or 5-6 to modify parameters.

4.7 Calculation Effort and Storage Requirement

Only frequently repeated calculations are relevant for the duration of the simulation. The most frequent calculations are entry wise array multiplication and 2D FFT. The array multiplications scale with N^2, the 2D FFT with $N^2.\log(N)$. As was shown before, the calculation of a succeeding slice needs two 2D FFT and two array multiplications. The time of one such calculation depends on N. If the duration of one single slice calculation say t_{ss} is known the overall time

of an `ixchel` simulation can be estimated as following.

$$T_{ixchel} = t_{ss}\left(n + 3\sum_{i=1}^{n} n_{scatt}[i](n-i)\right) \tag{4.1}$$

n should be the number of slices, $n_{scatt}[i]$ the number of scattering centres in slice i. Through parallelisation the calculation time can be split by the number of available CPU cores to obtain the time a simulation needs.
The procedure to calculate the intensity distribution of one wavefunction in the `ixchel_image` program requires the reading of the wavefunction from the file, two 2D FFT, one entry wise multiplication and one addition of two arrays. If the time needed for one single wavefunction is t_{sw} we can estimate:

$$T_{image} = 3t_{sw}\left(\sum_{i=1}^{n} n_{scatt}[i] + \sum_{j\in N_{out}}\sum_{i=1}^{j} n_{scatt}[i]\right) \tag{4.2}$$

N_{out} is the set of slice numbers in the range $[1, n-1]$ specified in `outslices`. The right term in Eq. 4.2 contains the number of stored wavefunctions. We add the number of slices n since we stored the multislice result after each slice. So we know the number of stored wavefunctions. We further know that the size of a wavefunction is double the size of an intensity image[3]. The overall storage space required is then

$$I = I_{wf}\left(3\sum_{i=1}^{n} n_{scatt}[i] + 3\sum_{j\in N_{out}}\sum_{i=1}^{j} n_{scatt}[i] + \frac{n_{out}}{2}\right) \tag{4.3}$$

I_{wf} is the size of one wavefunction data file (approx. 2.9 Mb for a 512x512 pixel data file). n_{out} is the number of elements in the set N_{out}.

[3]The wavefunction contains additionally an imaginary array of same size.

Chapter 5

Multislice Simulations

5.1 SrTiO3

The first crystal we are looking at is SrTiO3 (tausonite). In the first step we have to transfer the cubic crystal structure in Fig. 5.1 into slices. One possible way with Sr and O in the first slice and Ti and O in the second is shown in Fig. 5.2. Listing 5.1 shows the definition of the slices.

```
# Example crystal structure file
# for SrTiO3

# Stacking sequence
stack_seq = "126(ab)";
# Real space dimension Nx, Ny in pixel
dim = [ 512, 512];
# Replication factors of unit cell: NCELLX, NCELLY, NCELLZ
NCELL = [ 8, 8, 1];
# Layers
layers = {
  a = {
    # Size of unit cell in Angstrom
    size_uc = { x = 3.9051; y = 3.9051; z = 1.9525; };
    # Input file name atomic potential
    filename = "srtapot.tif";
    atoms = (
      {
        # name not mandatory, Z is read
        name = "Sr";
        Z=38;
```

```
22          positions = ( [ 0.0, 0.0] );
23        },
24        {
25          name = "O";
26          Z=8;
27          positions = ( [ 0.5, 0.5] );
28        }
29      );
30    };
31    # END layer a
32    b = {
33      size_uc = { x = 3.9051; y = 3.9051; z = 1.9525; };
34      filename = "srtbpot.tif";
35      atoms = (
36        {
37          name = "Ti";
38          Z=22;
39          positions = ( [ 0.5, 0.5] );
40        },
41        {
42          name = "O";
43          Z=8;
44          positions = (
45            [ 0.5, 0.0],
46            [ 0.0, 0.5]
47          );
48        }
49      );
50    };
51  };
```

Listing 5.1: Crystal structure file for SrTiO3

The simulation was run with energy 300 keV, and inelastic scattering at oxygen (K-edge) with an energy loss of 532 eV. C_s is 0.9 mm. The optimal defocus can be calculated with Eq. 3.64 and is approximately 51.5 nm. Listing 5.2 shows the simulation parameters.

```
1  # Example simulation file
2  # for SrTiO3
3
4  # Real space dimension Nx, Ny in pixel
5  dim = [ 512, 512];
6  # Energy of incident beam in keV
7  v0 = 300.;
8  # Atomnumber at which electron scatters inelastically
9  Zinel = 8;
10 # Energy loss in eV
```

```
11 Eloss = 532.;
12 # Crystal tilt [ xtilt, ytilt ] in mrad
13 ctilt = [0.0, 0.0];
14 # Incident beam tilt [ xtilt, ytilt ] in mrad
15 btilt = [0.0, 0.0];
16 # Define all the additional depth (# slices) were the inelastic wavefunction
17 # should be written to a file; Note: provide in ascending order.
18 outslices = [28, 56, 84, 112, 140, 168, 196, 224, 252];
19 # Translation invariance of exit intensity array?
20 transopt = true;
21 # Translation invariance of exit wavefunction between replicated unit cells?
22 transwf = true;
23 # File name to read inelastic scattering amplitude
24 filescatt = "O_K_Ang.dat";
25 # Spherical aberration Cs in mm
26 Cs = 0.9;
27 # Defocus, mean, standard deviation and sampling size in Ang.
28 df0 = 515.0;
29 sigmaf = 0.;
30 dfdelt =0.;
31 # Object aperture in mrad
32 aobj = 9.42;
33 # Coherent image calculation mode
34 cmode = true;
35 # Magnitude and angle of 2-fold astigmatism in Ang. and degrees
36 dfa2 = 0.;
37 dfa2phi = 0.;
38 # Magnitude and angle of 3-fold astigmatism in Ang. and degrees
39 dfa3 = 0.;
40 dfa3phi = 0.;
41 # Objective lens and aperture centre x, y in mrad
42 objlx = 0.;
43 objly = 0.;
```

Listing 5.2: Simulation file for SrTiO3

In Fig. 5.3 the wavefunction shows the propagation through the crystal. The thickness values equal the number of slices:

```
outslices = [28, 56, 84, 112, 140, 168, 196, 224, 252];
```

The last two intensity plots 5.5 and 5.6 show a variation of crystal slices and defocus values. The first for elastic scattering only the second with inelastic scattering.

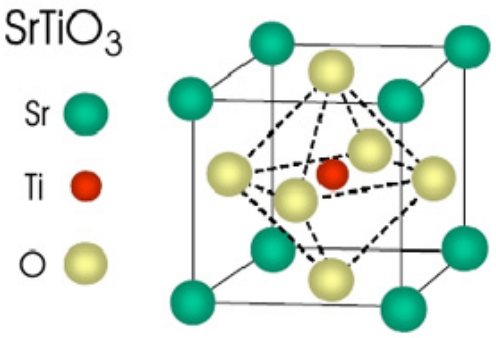

FIGURE 5.1: Figure shows the crystal structure of SrTiO3

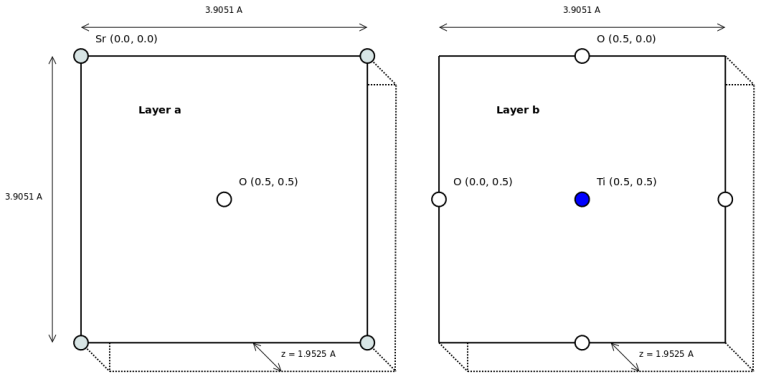

FIGURE 5.2: Figure shows the two layers a and b defined in the crystal structure file Listing 5.1

5.2 Si [110]

The second crystal we are looking at is Si in [110] direction. In Fig. 5.7 the coloured atoms indicate the two slices (Fig. 5.8) that are needed to replicate the structure. Listing 5.3 shows the resulting configuration file.

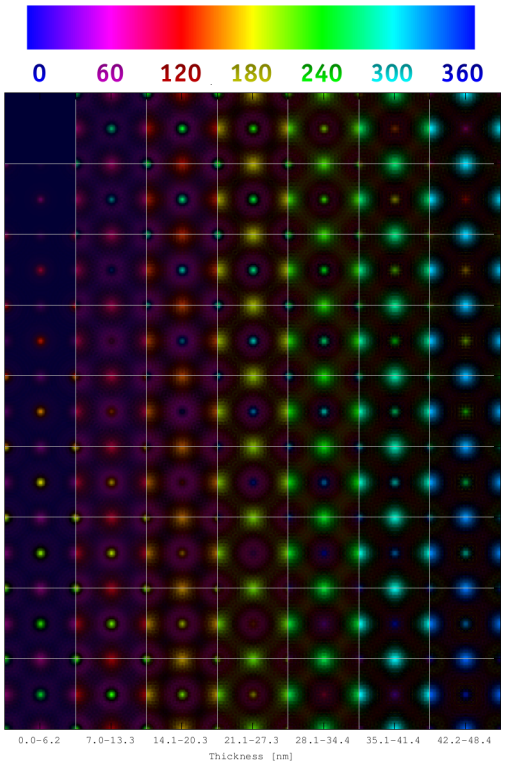

FIGURE 5.3: Wavefunction with progressing thickness (from top to down) for SrTiO3 (1 unit cell) with an energy of 300 keV. The colour-bar indicates the phase shift in degrees.

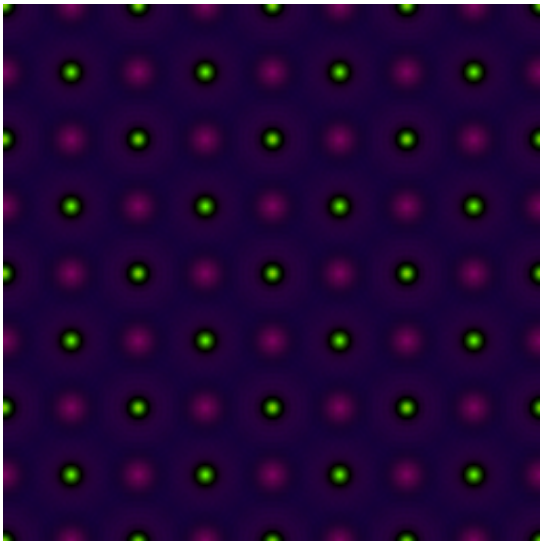

FIGURE 5.4: 4x4 unit cells of the elastic propagated wavefunction at slice 24 for SrTiO3 with an energy of 300 keV

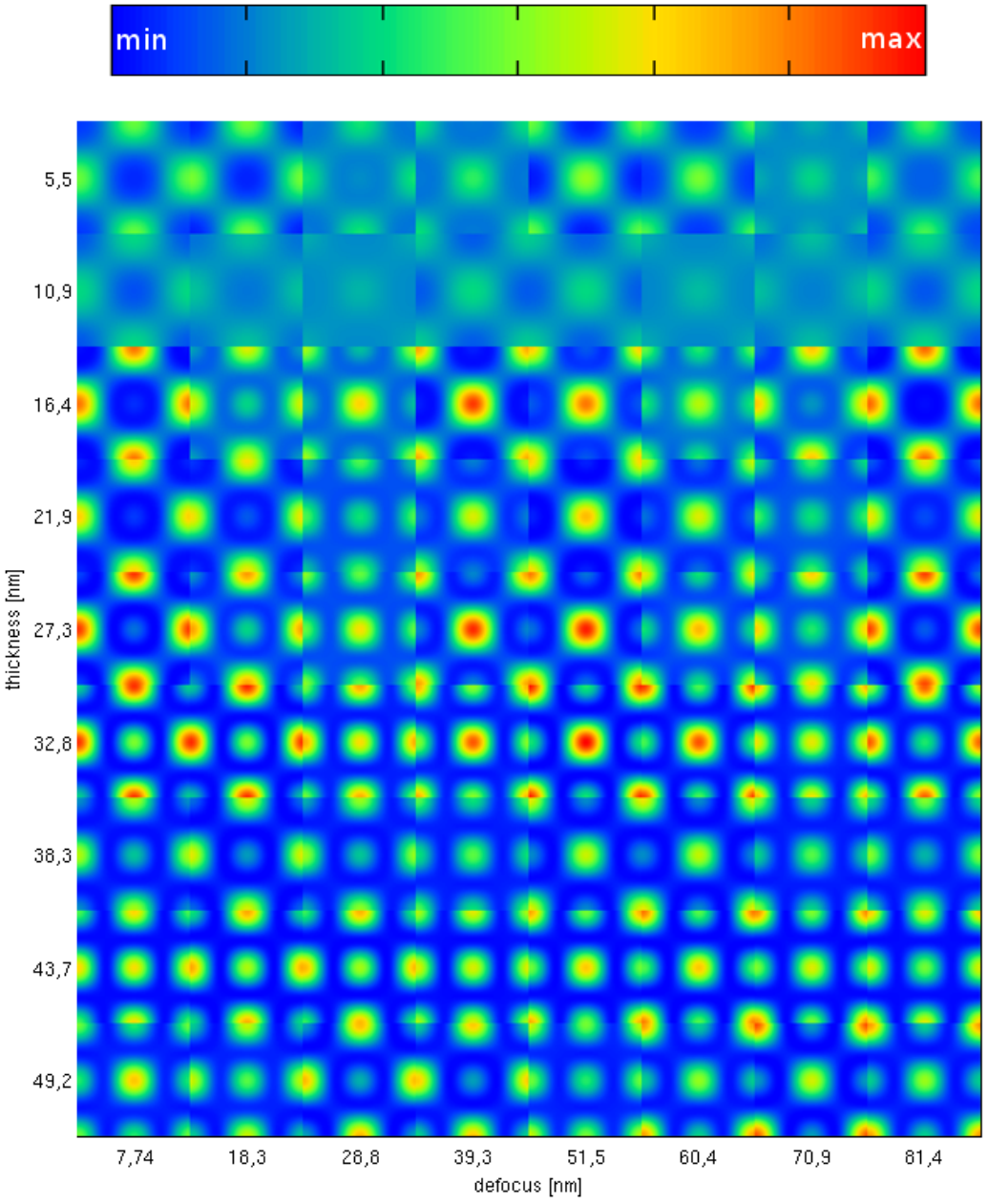

FIGURE 5.5: Elastic focal series of SrTiO3 with energy 300 keV and C_s 0.9 mm. The colour-bar indicates the intensity.

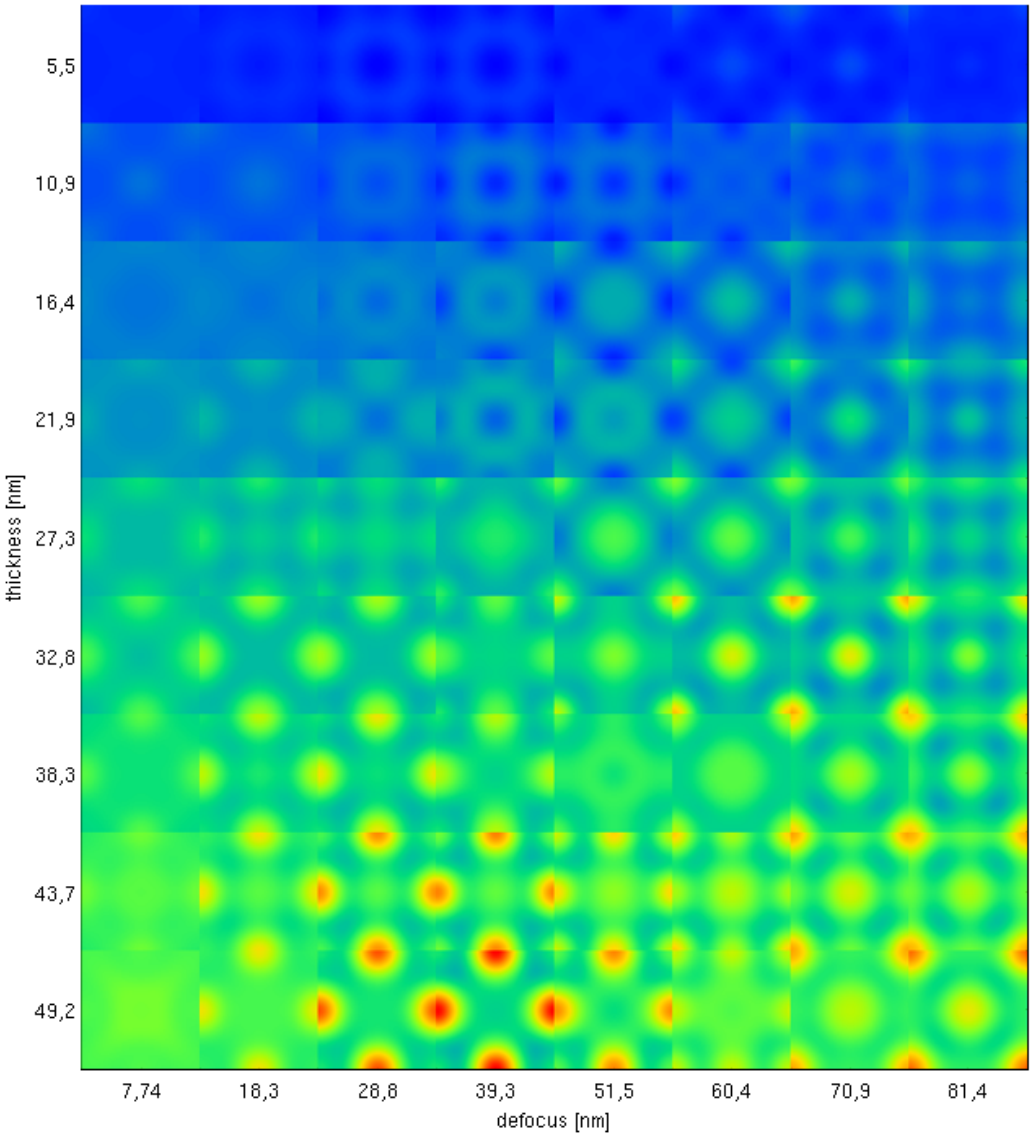

FIGURE 5.6: Focal series for SrTiO3 with energy 300 keV and C_s 0.9 mm

```
# Example crystal structure file
# for Si (110)

stack_seq = "126(ab)";
dim = [ 512, 512];
NCELL = [ 8, 8, 1];
layers = {
    a = {
      size_uc = { x = 3.84009; y = 5.43071; z = 1.92005; };
      filename = "siapot.tif";
      atoms = (
          {
          name = "Si";
          Z=14;
          positions = (
            [ 0.0, 0.0],
            [ 0.5, 0.25]
            );
          }
          );
    };
# END layer a
    b = {
      size_uc = { x = 3.84009; y = 5.43071; z = 1.92005; };
      filename = "sibpot.tif";
      atoms = (
          {
          name = "Si";
          Z=14;
          positions = (
            [ 0.5, 0.5],
            [ 0.0, 0.75]
            );
          }
          );
    };
  };
```

Listing 5.3: Crystal structure file for Si110

The simulation was run with energy 300 keV, and inelastic scattering at silicon ($L_{2,3}$ transition) with an energy loss of 99 eV. C_s is 0.9 mm. The optimal defocus is approximately 51.5 nm. The intensity plots Fig. 5.9 and Fig. 5.10 show again a thickness and focal series for the elastic and inelastic case. The thickness values equal the slice number:

`outslices = [28, 56, 84, 112, 140, 168, 196, 224, 252];`

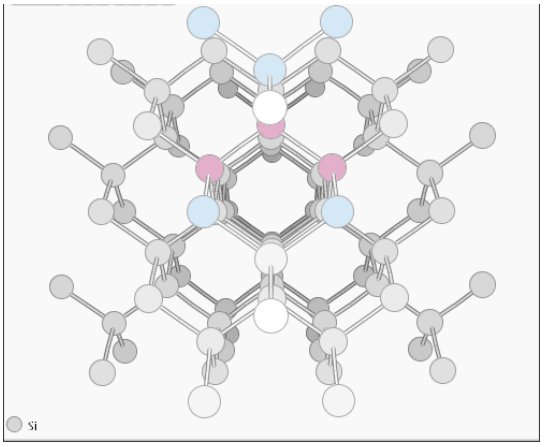

FIGURE 5.7: Look on Si surface in [110] direction.

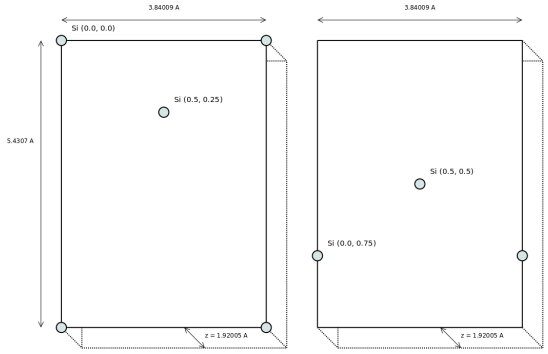

FIGURE 5.8: Split-up of Si [110] into unit cells.

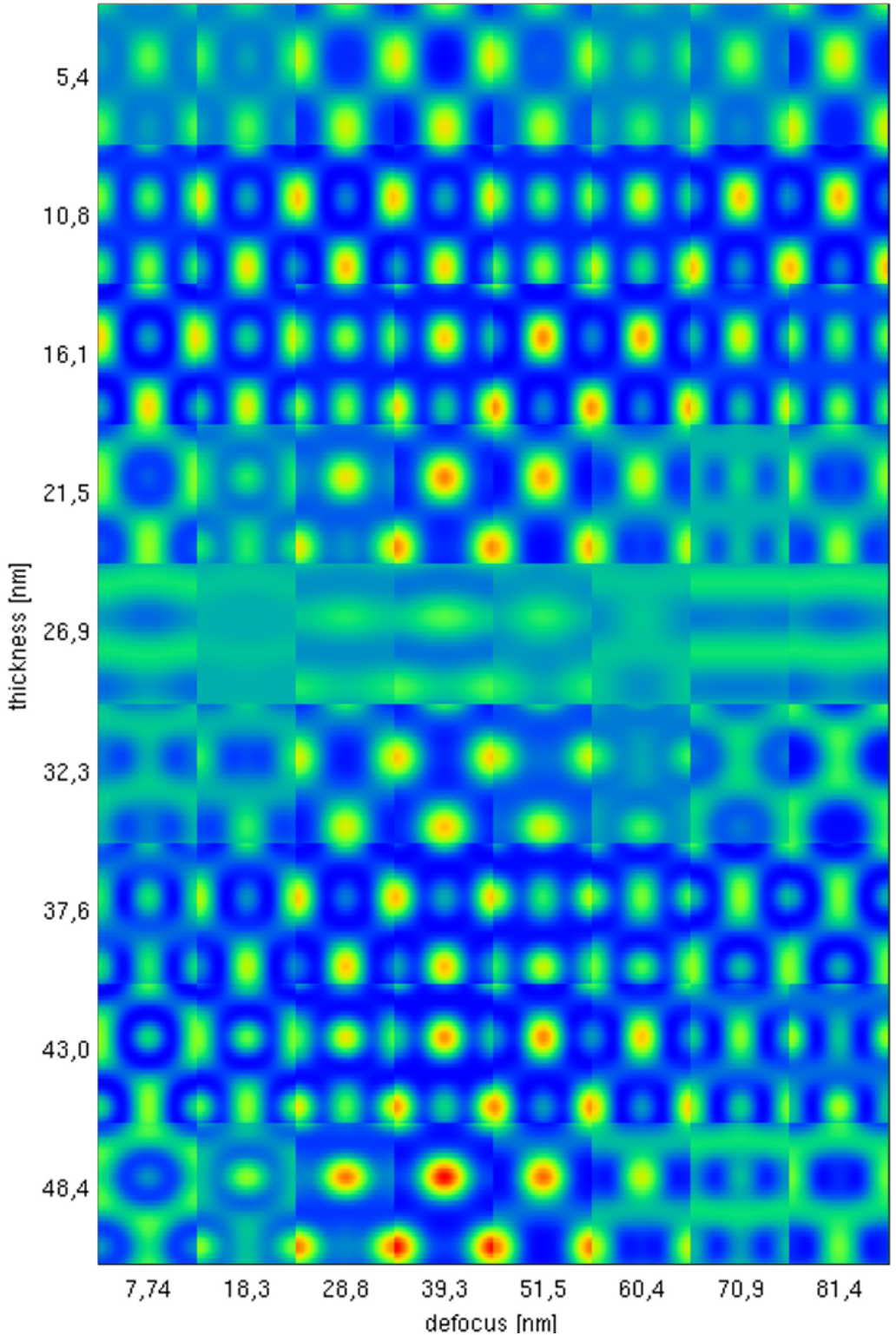

FIGURE 5.9: Elastic focal series of Si [110] with energy 300 keV and C_s 0.9 mm

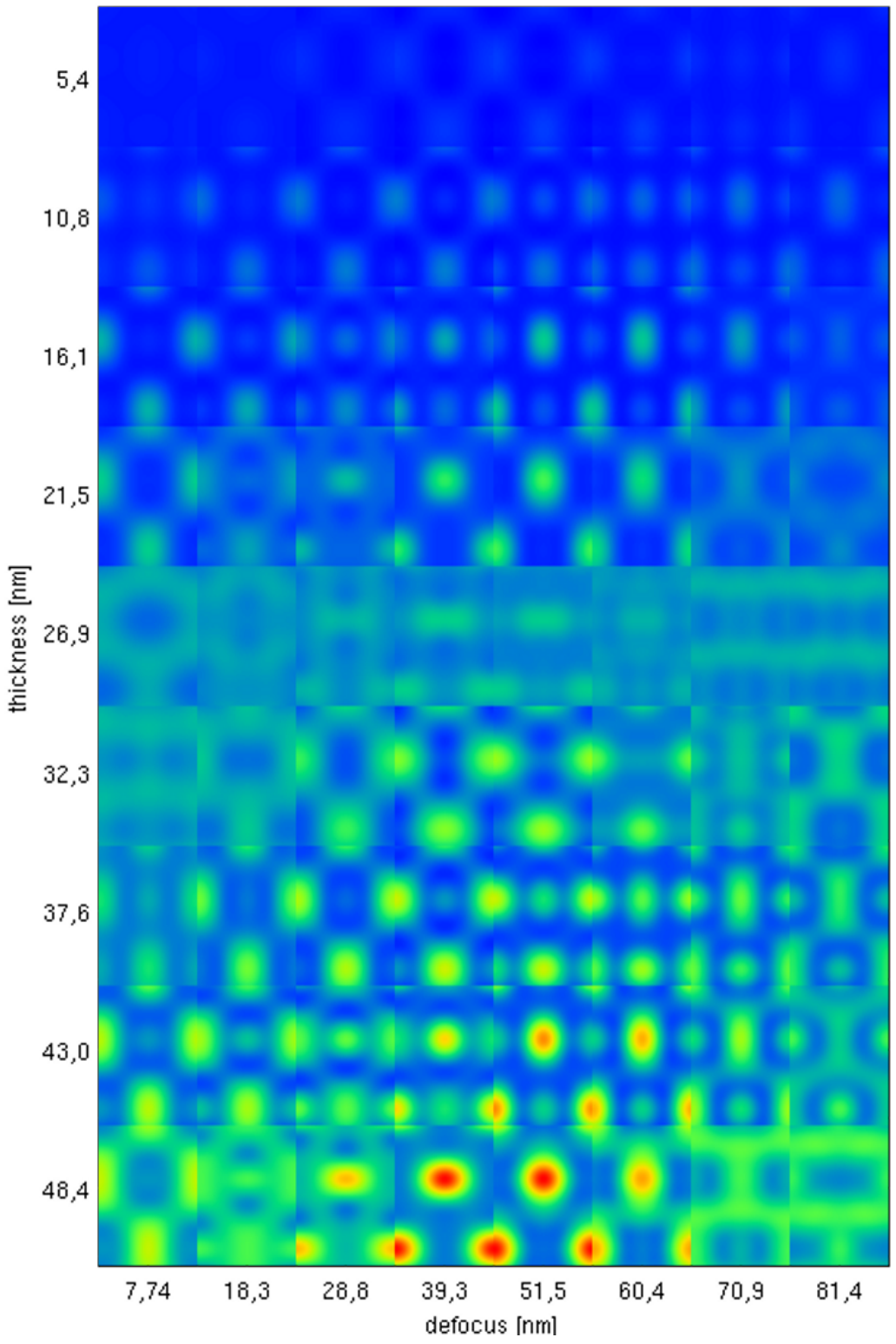

FIGURE 5.10: Focal series for Si [110] with energy 300 keV and C_s 0.9 mm

Chapter 6

Conclusion

The introduced method shows a way to implement inelastic scattering by core electron transition into the multislice method. It became clear that the calculation time required for a simulation scales with the power of 2 of the sample thickness and linear with the number of scattering atoms in a slice. There are a considerable number of simplifications for special configurations that immensely reduce the calculation time. The simulations can be run in affordable time on home computers.[1] The independence of the single calculations in the model allows a simple approach to parallelize them. The tendency to extend the number of processor cores in home computers will improve the run time even more. Therefore even more complex structures than given here should not present a problem on modern home computers.

The memory usage scales with the resolution of the image and with the number of simulations running in parallel but remains in a neglect able range. The hard disk storage requirements are a bit higher and quickly grow in sizes of gigabytes. The storage requirement scales with the power of 1 to 2 with the thickness of the sample (depending on the number of intermediate results we are interested in). Data compression provides one simple way to reduce the required storage on the expense of calculation time. Ratios up to 1:2, for periodic data even up to 1:10 are possible. Exploitation of symmetries and limiting factors would

[1] The presented SrTiO3 simulations took approximately five hours on a low cost system (4 CPU cores with 1.7 GHz and 1024 KiB cache)

reduce the storage even more. For example the objective aperture size limits the relevant domain in the k-space representation of a wavefunction.
The written software provides a practical tool to interpret high resolution EFTEM images and to find new structures. Versatility was achieved by a modular design of the program. In the future it can be used as a framework for improved and more specialized programs.

Appendix A

Programming Details

A.1 Functions

A.1.1 Initial Calculations

```
int incwf(config_t *cfg_cry, config_t *cfg_sim,
    float **waver, float **wavei, float ***transr, float ***transi,
    float **propxr, float **propxi, float **propyr, float **propyi,
    float *kx2, float *ky2, float *k2max)
```

The function `incwf()` is doing most of the initially required calculations to start the `ixchel` program. The results are stored in arrays of different size and dimension and are repeatedly used in the following calculations.
`transr` and `transi` are calculated for each defined layer once. This is done by reading the reduced potential information from the corresponding atompot data file and by multiplication with a scaling factor. The scaling factor contains σ (Eq.3.41) and an relativistic correction factor.
`propxr`, `propyr` and their associated imaginary arrays are calculated like seen in Eq. 3.59.

A.1.2 Elastic Propagation

```
void transmit( float** waver, float** wavei,
    float** transr, float** transi, int nx, int ny )
```

The function `transmit()` is doing the calculation $t_n(\mathbf{x})\phi_n(\mathbf{x})$ in Eq.3.56. This is basically a point for point array multiplication in real space. The arrays `transr[][]` and `transi[][]` have been calculated once for each layer at the program start. Since `waver[][]` and `wavei[][]` are the real and imaginary part as are `transr[][]` and `transi[][]` the complex product looks like:

```
waver[ix][iy] = waver[ix][iy]*transr[ix][iy] - wavei[ix][iy]*transi[ix][iy];
wavei[ix][iy] = waver[ix][iy]*transi[ix][iy] + wavei[ix][iy]*transr[ix][iy];
```

The results are stored in the arrays `waver[][]` and `wavei[][]`. Only these two arrays change during the function call.

```
void propagate( float** waver, float** wavei, float* propxr,
  float* propxi, float* propyr, float* propyi,
  float* kx2, float* ky2, float k2max, int nx, int ny )
```

The function `propagate()` calculates the convolution in Eq. 3.56. Since it bears advantages to use a Fourier transform we employ the convolution theorem and use Eq.3.53 instead. This means that `waver[][]` has to contain the Fourier transformed before it is provided to `propagate()`. We would end in a simple array multiplication like in `transmit()`. Eq. 3.56 allows us to separate for the two variables k_x and k_y. Instead of one "propagation" array of size `nx*ny` we remain with one array of size `nx` and one of size `ny`. This are the arrays `propxr[]` and variations. The wavefunction arrays are now multiplied in the same way it is done in `transmit()`. Only this time multiplied with the two components of the propagation arrays, and only until the sum of `kx2` and `ky2` reaches the bandwidth limit `k2max`.

A.1.3 CRC calculation

Although changes in the configuration files are possible between the three simulation steps, the dependencies in Table 4.1 have to be considered. If data files that were produced with different configuration parameters are used in a following simulation step a warning is printed. To check for changes every data file contains in the third TIFF directory CRC-checksums of the basic configuration files that were used. The calculation method for the CRC-checksums is the same which is implemented in the coreutils tool `cksum`.

```
unsigned int cksum(const char *filename);
```

The function reads the passed file and returns a 32 bit checksum.

A.1.4 HSV Colour mapping

```
int hsv_rgb(float* c_hv, unsigned short * c_rgb );
```

As explained in the appendix the TIFF format allows two ways to create colour images[16]. The wave function results we obtain consist of two arrays containing an real and imaginary value: $z_{kl} = x_{kl} + iy_{kl}$. One way to create images was to draw the real and imaginary array in grey scale next to each other. An easier way to interpret the results is to plot the intensity and the - phase shift ($z_{kl} = r_{kl}.\exp(i\phi_{kl})$)in a colour RGB image. It is common practice to plot complex values in the HSV model. The RGB model requires three values per pixel as does the HSV model. To reduce the degrees of freedom we set the saturation to 1. Next we map the intensity and the phase shift in the following

way:

$$\begin{aligned} r &\mapsto v \\ [r_{min} \ldots r_{max}[&\mapsto [0 \ldots 1[\\ \text{and} & \\ \phi &\mapsto h \\ [0 \ldots 2\pi[&\mapsto [0 \ldots 1[\end{aligned}$$

`hsv_rgb()` converts the h and v values (passed in the form of the two value array `float c_hv[2]`) into three corresponding (`ushort c_rgb[3]`) values with the following algorithm:

```
1  h = c_hv[0];
2  v = c_hv[1];
3
4  i = floor(6*h);
5  f = 6*h - i;
6
7  x = v*(1-f);
8  y = v*f;
9  if i==0:
10   r = v; g = y; b = 0;
11 if i==1:
12   r = x; g = v; b = 0;
13 if i==2:
14   r = 0; g = v; b = y;
15 if i==3:
16   r = 0; g = x; b = v;
17 if i==4:
18   r = y; g = 0; b = v;
19 if i==5:
20   r = v; g = 0; b = x;
21
22 c_rgb[0] = 256*r;
23 c_rgb[1] = 256*g;
24 c_rgb[2] = 256*b;
```

LISTING A.1: Pseudo code of HSV to RGB colour space conversion in the case s=1. h, s, v and r, g, b are the coordinates in the corresponding colour space.[2]

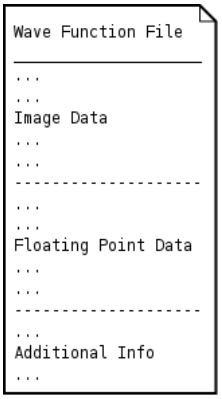

FIGURE A.1: Data file structure

The code uses an array of two float elements to calculate three 8 bit values.

A.2 The Data Files

In every simulation step we obtain different kind of data which is saved as a TIFF file. The TIFF data format offers an almost unlimited number of data containers, like described in the TIFF documentation [16]. Each container is called a directory. There are no limitations on the data type we like to store. We use this container to store independent data depending on the information we obtain in every simulation step. The data files are designed as three separate directories. The first directory contains a demonstrative improved representation of the image data. This allows a simple interpretation of the obtained data. The second directory contains the `float` valued result of the simulation. The third directory contains further information about the simulation. The first value in this directory is an `unsigned int` checksum. The checksum allows a comparison of the current configuration files with the present data file. This prevents the usage of data files which were obtained with different configuration parameters.

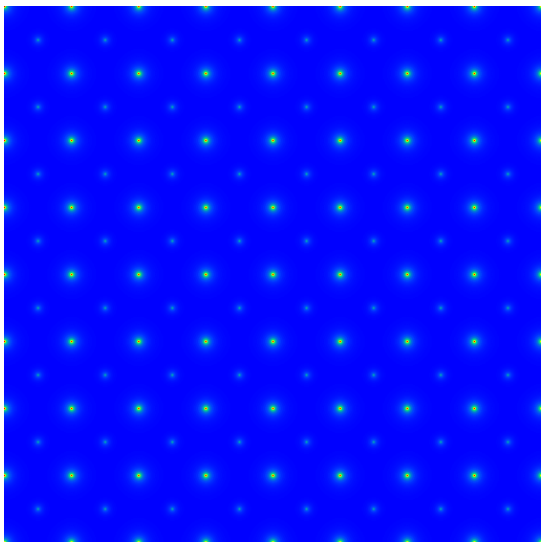

FIGURE A.2: The reduced potential of a SrTiO3 layer (8x8 unit cells) calculated with the `ixchel_atompot` program.

A.2.1 `ixchel_atompot` and `ixchel_image` data files

The `ixchel_atompot` and`ixchel_image` data files are build in the same way. The first IFD holds the data of the image representation. The second all the data in `float` precision. The third additional simulation informations. To achieve a color mapping from the data values the palette-colour method for TIFFs was used.

```
short** createColorMap()
```

The function returns the used colour mapping. It is the same mapping one would get by using gnuplot with the option `rgbformulae 22,13,-31`.

A.2.2 `ixchel` data files

The required wave functions are calculated in the second simulation step. Two ways to draw the graphical interpretation of the complex wave function are implemented. The first way is an 8 bit grey scale image plotting the real and

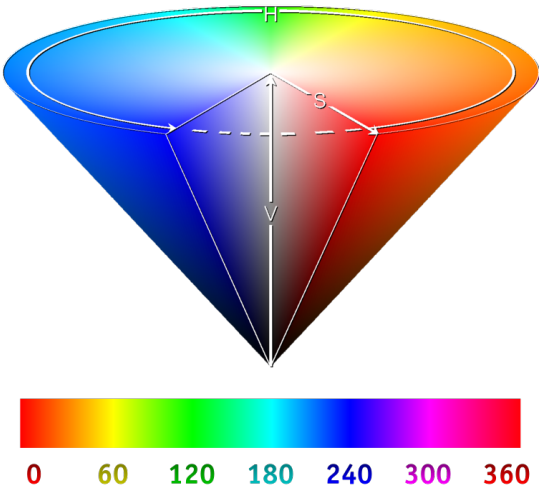

FIGURE A.3: Top: Conical representation of the HSV model. Bottom: Hue of HSV colour space for maximum saturation and value.[2]

complex part of the wave function next to each other. The second possible way is a colour representation in an RGB valued image. Each colour value is represented with an 8 bit value. The wave function is mapped to the HSV model as explained above.

A.3 Configuration Files

A.3.1 Realisation of the Configuration Files

There exist several realisations of configuration file reader routines, so there is no point in writing one. The most common files used are INI and XML files. Both of them allow implementations into C and can be manipulated with standard text editors. Nevertheless neither of those two was used in this program. The INI files are mainly used on Windows systems, but can also be used on different platforms. They imply a rather simple to use syntax. For more complex configurations like the configuration of the crystal slices with a whole set of parameters INI files are not appropriate.
The Extensible Markup Language (XML) is a good choice to define complex

structured data. XML implies an elaborated syntax. However, the files are not so simple to read and to create by hand. Therefore it is harder to maintain readability.

To avoid the mentioned weaknesses a third solution, `libconfig`[17] was chosen. The library uses a simple to understand syntax. Variable names can be assigned with all needed data types. Namely integer, float values and strings. Furthermore it is possible to assign arrays and lists with arbitrary complex nesting.

To give a picture of the configuration files syntax an example from the libconfig Documentation is printed in Listing A.2.

```
# Example application configuration file
version = "1.0";
application:
{
  window:
  {
    title = "My Application";
    size = { w = 640; h = 480; };
    pos = { x = 350; y = 250; };
  };
  list = ( ( "abc", 123, true ), 1.234, ( /* an empty list */) );
  books = ( { title = "Treasure Island";
              author = "Robert Louis Stevenson";
              price = 29.95;
              qty = 5; },
            { title = "Snow Crash";
              author = "Neal Stephenson";
              price = 9.99;
              qty = 8; } );
  misc:
  {
    pi = 3.141592654;
    bigint = 9223372036854775807L;
    columns = [ "Last Name", "First Name", "MI" ];
    bitmask = 0x1FC3;
  };
};
```

Listing A.2: Example of a general configuration file [17]

Appendix B

TIFF File Format

B.1 TIFF Specification

he simulation software uses the tagged image file format (TIFF)[1] version 6.0 for data in and output. The complete specification can be found at [16]. The TIFF data format was primarily developed to store two-coloured images. It was then extended to store grey scaled images and finally to store full coloured images.

The format allows additionally to store any amount of random information. The standard provides different types of compression. So it is possible to minimize size or calculation time. The file format uses a 32 bit offset which allows a maximum file size of 2^{32}byte, that is about 4 GB. Most image processing software uses quadratic pixel size, although the format does not limit the ratio of the pixel sizes. If necessary the image is simply stretched in one direction.

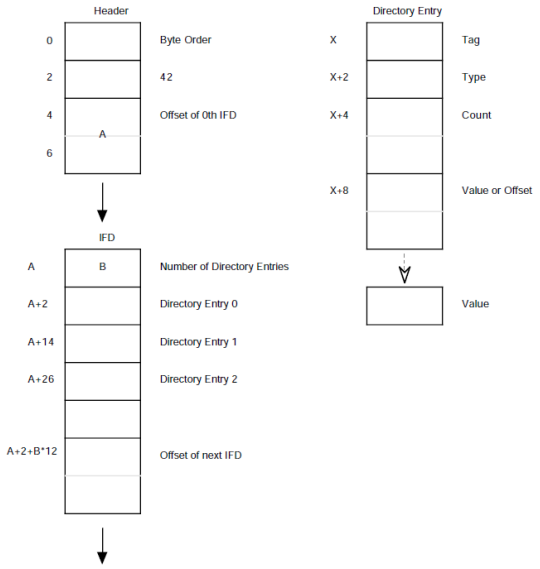

FIGURE B.1: Schematic structure of the TIFF file format

B.2 TIFF Structure

Any TIFF file starts with an eight byte image file header which points to an image file directory (IFD). The IFD contains all the necessary information of the image and a pointer which points to the actual data. The TIFF files are limited to 4 GiB (4,294,967,296 bytes).

[1] Although TIFF is the abbreviation for tagged image file format (TIFF) it is common to call it "TIFF format"

B.2.1 Image File Header

bytes 0-1 sets the type of byte ordering which depends on processor and operating system. Possible values are little-endian (eg. x86 systems), big-endian (eg. M68000)

bytes 2-3 "An arbitrary but carefully chosen number (42) that further identifies the file as a TIFF file."

bytes 4-7 Determines the offset of the first IFD. It can succeed directly after the image data but it must begin at a word boundary!

B.2.2 Image File Directory

The IFD starts with 2 bytes. They determine the number of directory entries. Then 12 bytes field entries follow. At the end follow 4 bytes which determine the offset for an eventually following IFD entry. They are set to 0 if it was the last entry.
The 12 byte field has the following format:

bytes 0-1 identification tag

bytes 2-3 type of the field

bytes 4-7 quantity of contained values

The TIFF functions provide additionally to grey scale images different ways to produce colour images.

B.2.3 Palette-Colour Images

One way is to define a field of RGB colour map (often called lookup-table). For every grey scale value a red, green and blue value is assigned. Since there doesn't exist a predefined lookup table the program has to create it first. This is done with the `TIFFSetField` function and the TIFF tag `TIFFTAG_COLORMAP`. For every RGB colour the address of an array with the assigned values is provided,

namely `*rgbcm[0...2]`. Eg. if we want to represent an 8 bit grey scale image as 8-bit colour image every one of the three arrays contains 2^8 values, and each element is of 8 bit size. To set the palette the following function is used:

```
TIFFSetField(tif, TIFFTAG_COLORMAP, rgbcm[0], rgbcm[1], rgbcm[2]);
```

B.2.4 RGB Full Color Images

The second way is to save an 8 bit value for every colour in the RGB model. Additionally there are two ways to store the three colour values. First it is possible to save the colour values for every pixel in the form of R,G,B next to each other. The second possibility is to store three fields each containing all the pixel information for one colour.

Appendix C

Fouriertheory

C.1 Fourier Transform and Fast Fourier Transformfast Fourier transform (FFT)

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k x} dx \tag{C.1}$$

$$f(x) = \int_{-\infty}^{\infty} F(k) e^{-2\pi i k x} dk \tag{C.2}$$

It is useful to think of $f(x)$ and $F(k)$ as different representations of the same function. Some important properties of the Fourier transform are the following:

$$f(x) = f^*(x) \qquad \Longleftrightarrow \qquad F(-k) = F^*(k) \tag{C.3}$$

$$f(ax) \leftrightarrow \frac{1}{|a|} F(\frac{k}{a}) \qquad \text{scaling} \tag{C.4}$$

$$\frac{1}{|b|} f(\frac{x}{b}) \leftrightarrow F(k) \qquad \text{reciprocal scaling} \tag{C.5}$$

$$f(x - x_0) \leftrightarrow F(k) e^{2\pi i k x_0} \qquad \text{shifting} \tag{C.6}$$

$$f(x) e^{2\pi i k_0 x} \leftrightarrow F(k - k_0) \qquad \text{reciprocal shifting} \tag{C.7}$$

C.1.1 Parseval's Theorem

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |H(k)|^2 dk \tag{C.8}$$

Parseval's theorem has different interpretations depending on its application. In signal processing it means that the total power of a signal is the same in the time and in the frequency domain. In quantum mechanics its interpretation as probability makes clear that the wave function has the same overall probability in space and reciprocal space representation.

C.1.2 Convolution

Two functions $f(x)$ and $g(x)$ and their corresponding Fourier transforms $F(k)$ and $G(k)$ allow a combination of interest. For this two functions the convolution $f \otimes g$ is defined in the following form:

$$(f \otimes g)(x) = \int_{-\infty}^{\infty} f(\xi) g(x - \xi) d\xi \tag{C.9}$$

The convolution has the following properties:

$$(f \otimes g)(x) = (g \otimes f)(x) \tag{C.10}$$

$$(f \otimes g)(x) \leftrightarrow F(k)G(k) \qquad \text{convolution theorem} \tag{C.11}$$

The last property means, that the Fourier transform of the convolution matches the product of the individual Fourier transforms.

C.2 Fourier Projection Theorem

For a three dimensional function $f(x, y, z)$ the Fourier transform is given by:

$$\begin{aligned} F(k_x, k_y, k_z) &= FT_{3D}[f(x, y, z)] \\ &= \iiint f(x, y, z) \exp(2\pi i(xk_x + yk_y + zk_z)) dx dy dz \end{aligned} \tag{C.12}$$

Next we calculate the two dimensional inverse Fourier transform of the three dimensional function $F(k_x, k_y, k_z)$

$$\begin{aligned} &FT_{2D}^{-1}[F(k_x, k_y, k_z)] = \\ &= \iint F(k_x, k_y, k_z) \exp(2\pi i(xk_x + yk_y + zk_z))dk_x dk_y \end{aligned} \tag{C.13}$$

$$\begin{aligned} &FT_{2D}^{-1}[F(k_x, k_y, k_z)] = \\ &= \iint \left(\iiint f(x', y', z') \exp(2\pi i(x'k_x + y'k_y + z'k_z))dx'dy'dz' \times \right. \\ &\left. \times \exp(-2\pi i(xk_x + yk_y)) \right) dk_x dk_y \end{aligned} \tag{C.14}$$

Changing the order of integration leads to

$$\begin{aligned} &FT_{2D}^{-1}[F(k_x, k_y, k_z)] = \iiint f(x', y', z') \times \\ &\times \iint (\exp(2\pi i((x' - x)k_x + (y' - y)k_y))dk_x dk_y)dx'dy'dz' = \\ &= \iiint f(x', y', z')\delta(x' - x)\delta(y' - y) \exp(2\pi iz'k_z)dx'dy'dz' \end{aligned} \tag{C.15}$$

Finally setting $k_z = 0$ we receive

$$FT_{2D}^{-1}[F(k_x, k_y, k_z = 0)] = \int f(x, y, z')dz' \tag{C.16}$$

The inverse Fourier transform applied to a three dimensional function and setting the third reciprocal space coordinate to zero results in the inverse transform over two dimensions and a projection of the third dimension.

C.3 Discrete Fourier Transform

For N sampled points with equal spacing the discrete Fourier transform is defined:

$$F_n = \sum_{k=0}^{N-1} f_k e^{2\pi i k n/N} \tag{C.17}$$

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{-2\pi i k n/N} \tag{C.18}$$

Bibliography

[1] B. Fultz and J.M. Howe. *Transmission Electron Microscopy and Dirffractometry of Materials*. Springer Verlag, 2002.

[2] Representation of the hsv model, 2010. URL http://commons.wikimedia.org.

[3] David A. Muller. Structure and bonding at the atomic scale by scanning transmission electron microscopy. *Nat Mater*, 8(4):263–270, 2009. ISSN 1476-1122. doi: 10.1038/nmat2380. URL http://dx.doi.org/10.1038/nmat2380.

[4] A. Weickenmeier and H. Kohl. Computation of the atomic inner-shell excitation cross-sections for fast electrons in crystals. *Philosophical Magazine Part B*, 60(4):1364–2812, 1989. ISSN 1364-2812. URL http://www.informaworld.com/10.1080/13642818908205920.

[5] L.J. Allen. Electron energy loss spectroscopy in a crystalline environment using inner-shell ionization. *Ultramicroscopy*, 48(1-2):97 – 106, 1993. ISSN 0304-3991. doi: DOI:10.1016/0304-3991(93)90175-W. URL http://www.sciencedirect.com/science/article/B6TW1-46JGMVR-JK/2/f1a38932923b53004e338d0dff8381ac.

[6] P. Schattschneider, M. Nelhiebel, and B. Jouffrey. Density matrix of inelastically scattered fast electrons. *Phys. Rev. B*, 59(16):10959–10969, April 1999. doi: 10.1103/PhysRevB.59.10959.

[7] Earl J. Kirkland. *Advanced Computing in Electron Microscopy*. Plenum Press, New York and London, 1998.

[8] E. Clementi and D. L. Raimondi. Atomic screening constants from scf functions. *J. Chem. Phys.*, 38(11):2686, 1963. ISSN 0021-9606. doi: doi: 10.1063/1.1733573. URL http://link.aip.org/link/JCPSA6/v38/i11/p2686/s1.

[9] D. B. Williams and C. B. Carter. *Transmission electron microscopy : a textbook for materials science*. Springer Verlag, 2009.

[10] R. F. Egerton. *Physical principles of electron microscopy : an introduction to TEM, SEM, and AEM*. Springer Verlag, 2005.

[11] P. Schattschneider, M. Nelhiebel, H. Souchay, and B. Jouffrey. The physical significance of the mixed dynamic form factor. *Micron*, 31(4): 333 – 345, 2000. ISSN 0968-4328. doi: DOI:10.1016/S0968-4328(99)00112-2. URL http://www.sciencedirect.com/science/article/B6T9N-3YMFKW6-3/2/7cb3bfe0e52ecd2a115cf15fc7317fba.

[12] Jo Verbeeck, Peter Schattschneider, and Andreas Rosenauer. Image simulation of high resolution energy filtered tem images. *Elsevier Science*, 1 (1):1–20, December 2008.

[13] J. M. Cowley and A. F. Moodie. The scattering of electrons by atoms and crystals. I. A new theoretical approach. *Acta Crystallographica*, 10 (10):609–619, Oct 1957. doi: 10.1107/S0365110X57002194. URL http://dx.doi.org/10.1107/S0365110X57002194.

[14] Kazuo Ishizuka. Contrast transfer of crystal images in tem. *Ultramicroscopy*, 5(1-3):55 – 65, 1980. ISSN 0304-3991. doi: DOI:10.1016/0304-3991(80)90011-X. URL http://www.sciencedirect.com/science/article/B6TW1-4D5X3YH-8/2/5df8f70d97b713248f618c9b4fee684f.

[15] R. Kilaas and R. Gronsky. Real space image simulation in high resolution electron microscopy. *Ultramicroscopy*, 11(4):289 – 298, 1983. ISSN 0304-3991. doi: DOI:10.1016/0304-3991(83)90008-6. URL http://www.sciencedirect.com/science/article/B6TW1-46MKWG9-3V/2/7126692577892c8c3b15d73e915457e9.

[16] Adobe. Tiff revision 5.0. *a*, a(1):1, June 1992. URL http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf.

[17] M. Lindner. libconfig. *a*, a(a):a, September 2008. URL http://www.hyperrealm.com/libconfig/.