Inversion of Dynamical Scattering from Large-Angle Rocking-Beam Electron Diffraction Patterns

A DISSERTATION PRESENTED
BY
FENG WANG
TO
THE FACULTY OF NATURAL SCIENCES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doktor rer. nat.
IN THE SUBJECT OF
Physics

Universität Ulm
Ulm, Germany
August 2016
Dean in office/Amtierender Dekan: Prof. Dr. Peter Dürre
1st expert/supervisor: Professor Dr. Christoph Koch
2nd expert: Professor Dr. Ute Kaiser
Day of doctoral exam: 23 February 2017
Inversion of Dynamical Scattering from Large-Angle Rocking-Beam Electron Diffraction Patterns

Abstract

A method for ab-initio structure factor retrieval from large-angle rocking-beam electron diffraction (LARBED) data of thin crystals is described and tested. This method determines crystal structure factors and specimen thickness from the intensities of the diffraction spots alone, solving a nonlinear least-squares problem. No additional information, such as atomicity or information about chemical composition, have been made use of. In addition to a demonstration of the method on 120 keV experimental data of SrTiO$_3$, where 456 structure factors have been retrieved, we analyze the dependence of the success of the reconstruction on specimen thickness and tilt range by applying this algorithm to simulated data. Our numerical experiments show that the dynamical inversion by gradient optimization works best if the beam tilt range is large and the specimen not too thick. At specimen thicknesses which allow for moderate multiple scattering, the large tilt amplitude effectively removes local minima in this global optimization problem, making ab-initio structure factor retrieval possible. In addition, a dynamic parallelism framework based on the Compute Unified Device Architecture (CUDA) is introduced, reducing the runtime of the optimization routine by two orders of magnitude when compared to running on a CPU.
## CONTENTS

1 **Introduction**.................................................. 5

2 **Quantitative Diffraction Patterns Simulation**.................. 11
   2.1 Electron Source ....................................... 12
   2.2 The Bloch wave method .................................. 14
   2.3 Matrix Exponential Approximations .................... 21
      2.3.1 Padé Approximation ............................... 21
      2.3.2 Optimized Taylor Polynomial Truncation .......... 22
   2.4 Simulation Outline ..................................... 29

3 **Inversion of Dynamical Scattering**.............................. 35
   3.1 Genetic Algorithm ...................................... 36
   3.2 Gradient Based Iteration Algorithm .................... 37
      3.2.1 Levenberg-Marquardt Algorithm .................. 38
      3.2.2 Conjugate Gradient Method ....................... 38
      3.2.3 Line Search ...................................... 41
   3.3 Derivative Evaluation .................................. 42

4 **Inversion from LARBED Patterns**................................. 49
   4.1 Numeric experiments: configuration .................... 51
   4.2 Numeric experiments ................................... 53
      4.2.1 Testing maximum tilt range factor ................ 53
      4.2.2 Testing beam selection schemes ................. 53
      4.2.3 Testing thickness and target functions ........ 54
      4.2.4 Testing beam tilt numbers and noise factors ... 55
      4.2.5 Collaborative test for SrTiO$_3$ ................ 55
      4.2.6 Collaborative test for silicon .................. 56
      4.2.7 Numeric experiment results ...................... 57
      4.2.8 Homotopy experiments ............................ 58
Dedication

This dissertation is dedicated to my family for their unparalleled support and understanding as well as my friends for their helpful advice and kind encouragement.
Foremost, I would like to express my sincere gratitude to my advisor Prof. Christoph Koch, PhD, for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of SEM group: Dr. Alberto Eljarrat Ascunce, Dr. Amin Parvizi, Johannes Müller, Robert S. Pennington, PhD, Dr. Peter Schäfer, Dr. Wouter Van den Broek, Dr. Zdravko Kochovski, Dr. Anna Mogliatenko and our secretary Marion Götsch, for their encouragement, insightful comments and hard questions.

My sincere thanks also goes to all staffs in Ulm group: Dr. Prof. Ute Kaiser, Sr.-Prof. Dr. Harald Rose, Dr. Johannes Biskupek, Dr. Felix Börmert, Dr. Dorin Geiger, Dr. Michael Kiariie Kinyanjui, Dr. Zhongbo Lee, Dr. Haoyuan Qi and secretary Brigitte Kast, for offering me many discussion opportunities in their group seminars and numerous helps.

I would also like to thank Carl Zeiss Foundation and German Research Foundation for funding this project.

Last but not least, I would like to give my deepest gratitude and love to my family, for giving birth to me at the first place and supporting me spiritually throughout my life.

Thank you very much, everyone!
1.1 Phases as information carriers. Top right, an art work of Ulm Cathedral, a Lutheran church located in Ulm, Germany. Bottom left, another art work of Brandenburg Gate, an 18th-century neoclassical monument in Berlin, and one of the best-known landmarks of Germany. Middle row and middle column, the amplitudes and phases of the Cathedral and Brandenburg Gate, generated using Fourier transform operations. Top left, a new image reconstructed using Brandenburg Gate’s phases and Cathedral’s amplitudes with inverse Fourier transform. Bottom right, another new image reconstructed using Cathedral’s phases and Brandenburg Gate’s amplitudes. ............................... 9

2.1 Elastic scattering demonstrated in a two-dimensional case. ................................. 12

2.2 The GPU-based CUDA code is about 2 orders of magnitude faster than the parallelized CPU code [PWK14]. ................................................................. 24

2.3 CBED Si(001) simulation, with thickness of 130 nm and a max tilt angle of 8 mrad at 120 keV. ................................................................. 25

2.4 CBED Si(110) simulation, with thickness of 130 nm and a max tilt angle of 4 mrad at 120 keV. ................................................................. 26

2.5 LARBED TiNb7O18(001) pattern simulation, with thickness of 10 nm, max tilt angle of 70 mrad and post-specimen beam-tilt compensation of 99.2% at 120 keV. 27
2.6 The multiple LARBED patterns computation is divided into a two-dimensional grid, and each block in this grid contains 64 threads. In each one of these threads, a diffraction pattern corresponding to a certain beam tilt is being calculated. And when the thread is being processed, the algorithmic control flow will determine when and where to launch child grids to calculate the matrix-matrix multiplications. These matrix-matrix multiplications grids is dedicated to implement the Taylor polynomial evaluations and the matrix squaring in the matrix exponential algorithm listed in Algorithm 2 on Page 33. Using built-in device synchronizing mechanism, the threads launching a matrix-matrix multiplication sub-grid will wait until this sub-grid comes to an end, then continues the procession after the place where the sub-grid is originally launched.

2.7 Detailed LARBED patterns simulation routine. ........................................ 29

3.1 Step length should be proper, not too large, not too long. .................... 31

3.2 Convergence curve for three test cases with different configurations. Blue: 9 beams with 256 individuals; Red: 25 beams with 128 individuals; Green: 9 beams with 128 individuals; ................................................................. 42

4.1 A numeric experiment with a pseudo specimen inherited from a SrTiO3(001) crystal at 120 keV and a thickness of 12 nm, but adding two Silicon atoms to break the symmetry. The target potential field is demonstrated at the left most side; the reconstructed potential fields at specified iterations are visualized as well. .................. 47

4.1 A numeric experiment with a pseudo specimen inherited from a SrTiO3(001) crystal at 120 keV and a thickness of 12 nm, but adding two Silicon atoms to break the symmetry. The target potential field is demonstrated at the left most side; the reconstructed potential fields at specified iterations are visualized as well. .................. 47
1.1 Phase methods. ......................................................... 6
1.2 Electron diffraction techniques. ................................. 8

4.1 Triplet error of SrTiO₃ numerical experiments. .............. 56
4.2 Experimental data obtained. ........................................ 62
4.3 Left: SrTiO₃ structure. Right: silicon structure. ............... 62
4.4 Simulated SrTiO₃ patterns with different beam tilts. .......... 63
4.5 Simulated silicon patterns (120 keV, 30nm) with different max tilt angles. .............. 63
4.6 Left: simulated SrTiO₃ pattern without noise. Right: simulated SrTiO₃ pattern without Poison noise (1,000 electrons per tilt assumed). .............. 63
4.7 SrTiO₃ patterns simulated with a maximum tilt angle 70 mrad, thickness of 11 nm (Left) and 12 nm (Middle), and their difference (Right). .............. 64
4.8 Beam selection strategies. Left: For SrTiO₃, 45 beams are selected. Middle and Right: Beam selection schemes for numerical experiments of Silicon(001). These two patterns are simulated with a high tension of 120 keV and 40 nm, and their colors are enhanced to show the selected beams clearly. Middle: same beam selection scheme as our SrTiO₃ experiments, including 45 beams, but only 5 discs have visible details. Right: skipping the forbidden reflections, including 45 beams that have higher intensities than the forbidden ones. .............. 64
4.9 Testing the impact of maximum tilt range. Upper row: target patterns with different maximum tilt ranges at 40 nm; Middle row: reconstructed patterns by minimizing R-value function; Lower row: reconstructed potential fields. .............. 65
4.10 Row 1: Target patterns simulated including forbidden reflections; Row 2: Reconstructed patterns from Row 1 by minimizing R-value; Row 3: Reconstructed potential fields corresponding to patterns in row 2; Row 4: Target patterns simulated excluding forbidden reflections; Row 5: Reconstructed patterns from Row 4 by minimizing R-value; Row 6: Reconstructed potential fields corresponding to patterns in row 5; ............ 66
4.11 Testing the impact of specimen thickness. First row: target patterns to be reconstructed from, SrTiO$_3$ with a maximum tilt range of 40 mrad; Second row: LARBED patterns reconstructed by minimizing R-value function; Third row: potential fields corresponding to the patterns in the second row; Fourth row: LARBED patterns reconstructed by minimizing merit function; Last row: potential fields corresponding to the patterns in the fourth row. .............................................. 67

4.12 SrTiO$_3$ numerical experiment with different beam tilt sets and noise levels. To better present the weak details of the diffraction patterns, all intensities are visualized in logarithm scale. ......................................................... 68

4.13 SrTiO$_3$ refinement (1/2). .................................................. 69

4.14 SrTiO$_3$ refinement (2/2). .................................................. 70

4.15 Silicon refinement (1/4). ..................................................... 71

4.16 Silicon refinement (2/4). ..................................................... 72

4.17 Silicon refinement (3/4). ..................................................... 73

4.18 Silicon refinement (4/4). ..................................................... 74

4.19 Testing the performance of homotopy function as target function. First column: target patterns to be reconstructed from, SrTiO$_3$ with a maximum tilt range of 40 mrad; Second column: LARBED patterns reconstructed by minimizing homotopy function function; Third column: potential fields corresponding to the patterns in the second column. ................................................................. 75

4.20 Left: Original experimental pattern, presented with a wrong rotation angle $\theta = 0$. Middle: Kinematically fitted pattern rotated with recovered rotation angle $\theta$. Right: Rotated experimental pattern with recovered rotation angle $\theta$. .......................... 76

4.21 (a) Experimental LARBED pattern of SrTiO$_3$ with 1009 tilts at 120 Kv. (b) Reconstructed LARBED pattern, assuming no previous knowledge about the specimen. (c) Differential intensity between the experimental and the reconstructed LARBED patterns. (d) Theoretical SrTiO$_3$ (00l) potential map with 121 beams. (e) Reconstructed potential map with 121 beams, assuming no previous knowledge about the specimen. (f) Reconstructed potential map with 456 beams, enforcing that diffraction discs with similar total intensities agree in their structure factors’ amplitudes. .......................... 76
In order to determine the positions of atoms within the unit cell of a crystalline material from diffraction experiments the crystallographic phase problem must be solved. Kinematic X-ray crystallography methods handle this problem assuming some prior knowledge, and the convergent beam electron diffraction applications requires very thick specimens with small unit cells, and is not capable of solving new structures. This project addresses this problem in the context of thin specimens with large unit cells, and takes the advantages of the high performance modern general purpose graphics processing unit to accelerate the computation and to solve the optimization problem of big data set using the large-angle rocking-beam electron diffraction technique.

As a crystal can be described mathematically as a convolution between a lattice and a unit cell content, it is consequently possible to describe the atomic positions inside the crystal by means of a Fourier series, and furthermore using this series to represent the electron density distribution [Pat34, DK68]. Diffraction methods have been developed to examine the crystal structures by employing a short wavelength ray to impinge on the crystal substance with pre-selected geometry configuration, measuring the scattered intensities and reconstructing the Fourier series from the measurement.

X-ray diffraction in crystals was discovered by Laue, Friedrich and Knipping in München early
in 1912. Since then, as the quick development of this technique, phase problem [BFGR76, Tay03] arises when aiming to reconstruct the Fourier series from physically measured diffraction patterns. The recorded diffraction data gives the intensities but drops their phases.

The importance of phases in reconstructing the correct crystal structure is illustrated in Figure 1.1 on page 9. The phases of the Cathedral and the Gate in this figure are very hard to interpret visually, and both of them look like pure noises. Nevertheless, rich information is hidden in them, and this information is essential in the reconstruction of the original images. To demonstrate this point, we exchange the magnitudes and phases of these two figures in Fourier domain, and then perform two inverse Fourier transforms on them. From the amplitudes of the two reconstructed images, it is easy to see that a reconstruction using the Gate's phases and the Cathedral's amplitudes can result in a Gate and using the Cathedral's phases and the Gate's amplitudes can result in a Cathedral, and we can conclude that most of the information in the image that we perceive is stored in phase.

There is no formal relationship between the intensities and the phases, therefore, some prior knowledge of the structure, such as electron density and atomic information, must be assumed before recovering the phase information. Typical phasing methods [HAU92, Kar86, Pat34, Pat35, PT71, SHW+06] in X-ray crystallography from data obtained by regular equipment are reviewed by Taylor [Tay03] in Table 1.1. And the charge-flipping method [OS04, OS07] implements the atomicity constraint in an iterative algorithm assuming a real, positive, and sharply peaked density in real space.

<table>
<thead>
<tr>
<th>Method</th>
<th>Prior Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct method</td>
<td>positive electron density, discrete atoms</td>
</tr>
<tr>
<td>Molecular replacement</td>
<td>Homology model</td>
</tr>
<tr>
<td>Isomorphous replacement</td>
<td>Heavy-atom substructure</td>
</tr>
<tr>
<td>Anomalous scattering</td>
<td>Absorption by specific atoms</td>
</tr>
<tr>
<td>Density modification (phase improvement)</td>
<td>Solvent flattening</td>
</tr>
<tr>
<td></td>
<td>Histogram matching</td>
</tr>
<tr>
<td></td>
<td>Non-crystallographic symmetry averaging</td>
</tr>
<tr>
<td></td>
<td>Partial structure</td>
</tr>
<tr>
<td></td>
<td>Some phases</td>
</tr>
</tbody>
</table>

Table 1.1: Phase methods, adopted from [Tay03].

The first transmission electron microscope (TEM) of the world was constructed by Ernst Ruska and Max Knoll [KR32a, KR32b, Rus87] 85 years ago in Germany. As the scattering cross
section of electrons is about 5 orders of magnitude larger than that of X-rays it is able to probe crystal structures a few \( \text{nm}^3 \) in size. This technique is therefore widely used to determine crystal structures.

Crystal structure determination from electron diffraction data has conventionally been an adaptation of kinematic X-ray crystallography methods. The success of these techniques for small structures (i.e. proteins up to about 1000 atoms in the asymmetric unit \([\text{Hau97}]\)) is based on the fact that with the availability of diffraction data up to a sufficiently high resolution the number of recorded diffraction intensities is much greater than the number of atoms in the structure. However, while in beam-sensitive structures radiation damage makes it often impossible to collect diffraction data up to atomic resolution, dynamical scattering in thicker crystals causes electron diffraction intensities to be a highly non-linear function of the structure factors, making the kinematic approximation inapplicable in many cases \([\text{WPK16}]\).

Despite these problems, the refinement of crystal structure factors from electron diffraction patterns has a long history \([\text{Cow95, Vai13, WLZNO6, Dor13}]\). Precession electron diffraction (PED) \([\text{VE93, VE95}]\) has been developed in an attempt to make electron diffraction data more kinematic-like by average over different multiple scattering conditions, where the diffraction intensities are integrated on a ring of a fixed tilt amplitude. Considerable effort has been devoted to matching dynamical diffraction intensities to Bloch wave simulations \([\text{ZS91, SZ92, Spe93, Spe98, SCZ99, PJC+13}]\), multi-slice calculations \([\text{GM74, AJL98, ALS99, AFL00, AKOS01}]\) or other approaches \([\text{ZA97, JTZS98, FJS+04, BMHH92, BS92, WRS+96, IKM+14}]\).

Convergent beam electron diffraction (CBED) \([\text{KM39}]\) experiments provide electron diffraction data for a range of slightly different incident beam directions in a single exposure of the diffraction camera. For each of these different incident beam directions the dynamical scattering conditions are slightly different. Matching structure factors to CBED data \([\text{Zu098, ZKOS99}]\) may thus largely overdetermine the resulting optimization problem, making this technique very sensitive to both amplitude and phase of structure factors.

For conventional CBED patterns the range of incident beam directions is limited to the first Brillouin zone, since otherwise diffraction discs would overlap. For crystals with lattice parameters smaller than 1 \( \text{nm} \) the specimen thickness for quantitative CBED work must thus be rather large (typically 80 - 150 \( \text{nm} \) at electron beam energies of 120 - 200 \( \text{keV} \)) for the diffraction discs
to feature significant changes in the diffraction intensities, making it inapplicable for nanomaterials [LLQ03, MBSS+08, eeReSeSG09].

<table>
<thead>
<tr>
<th>CBED</th>
<th>DRZAP</th>
<th>LACBED</th>
<th>SLACBED</th>
<th>LACDIF</th>
<th>PED</th>
<th>Rotation</th>
<th>LABED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>Small</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Small</td>
<td>Small</td>
<td>Small</td>
</tr>
<tr>
<td>Small</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
<td>Large</td>
</tr>
<tr>
<td>Disc</td>
<td>Square</td>
<td>Disc</td>
<td>Disc</td>
<td>Disc</td>
<td>Ring</td>
<td>Line</td>
<td>Disc/Line</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Short</td>
<td>Long</td>
<td>Long</td>
<td>Short</td>
<td>Short</td>
<td>Long</td>
<td>Long</td>
<td>(Long)</td>
</tr>
</tbody>
</table>

Table 1.2: Comparison of electron diffraction techniques, adopted from [Koc11].

In case of thin specimen, or crystals with lattice parameters much larger than 1 nm, it is very difficult, if not impossible, to obtain CBED patterns with significant variation of intensity within the diffraction discs. By acquiring diffraction data at different beam tilts sequentially, the large-angle rocking-beam electron diffraction (LARBED) technique [Koc11] makes it possible to collect CBED-like data from nano-volumes for beam tilt angles up to about 100 mrad, i.e. more than 20 times larger than the largest possible convergence angle in CBED experiments of even a small unit cell crystal such as silicon. This method is similar to PED [VM94, GZH+03, OSM04, OMS05, AKN+07, DKMM07, NMG07, OHZ07, MGK09, NBP+10, RPN+10, ZGO+10], but PED does not keep track of the variation of diffracted intensities with varying tilt angles. A comparison of different electron diffraction techniques, including CBED, the double-rocking zone-axis pattern (DRZAP) [Ead80], large-angle convergent-beam electron diffraction (LACBED) [TSKH80, TUH80], simulated LACBED (SLACBED), LACDIF [MHR+08], PED and Rotation method [ZOHZ10] are listed on Table 1.2 on page 8. With large beam tilt angle achievable and preserved tilt information, the dynamic inversion problem and the crystal structure reconstruction from LARBED patterns of thin specimens are therefore made feasible from the massive diffraction diffraction data set along, without assuming any prior knowledge of atomicity or symmetry.
Fig. 1.1: Phases as information carriers. Top right, an art work of Ulm Cathedral, a Lutheran church located in Ulm, Germany. Bottom left, another art work of Brandenburg Gate, an 18th-century neoclassical monument in Berlin, and one of the best-known landmarks of Germany. Middle row and middle column, the amplitudes and phases of the Cathedral and Brandenburg Gate, generated using Fourier transform operations. Top left, a new image reconstructed using Brandenburg Gate’s phases and Cathedral’s amplitudes with inverse Fourier transform. Bottom right, another new image reconstructed using Cathedral’s phases and Brandenburg Gate’s amplitudes.
This chapter introduces some basic crystallography related principles and tools, including electron source, electrostatic lattice potential, Bloch wave method and diffraction pattern image simulation routine.

In the TEM, the elastic scattered electrons contribute most part of the intensity in the diffraction patterns, interact strongly with the atoms inside the crystal and can be scattered more than once (dynamic scattering) with a typical thin specimen of several nanometers.

Fig. 2.1 shows a simplified diagram of parallel beam electron diffraction, where only elastic scattering is considered. The process of electron microdiffraction has been fully studied by many authors almost a century ago. Before the invention of electron microscopy, Bethe has already discussed it with the Schrödinger equation, Fourier series of the crystalline potential inside the specimen and the electron wave function of the same periodicity of crystal lattice [Bet28], which is known as Bloch-wave method following Bloch’s Theorem [Kit66, AM03]. Latter the Bloch-wave function is casted into a scattering matrix form in reciprocal space [NW55, Fuj59, Fuj61, Stu62, Tou62], and this method is widely extended and employed in the context of Scanning Transmission Electron Microscopy (STEM) [HW61, HW62, FR81, SOBS83, VD85, Sta87, PJ90, PJ91, PJ92, Wat93, Cow95, Bet97]. The multislice method has been first proposed by Cowley and Moodie from a physical optics point of view [CM57], in which the specimen is divided into a series of two-dimensional phase objects thin enough to make the (strong) phase object approximation valid for each of the thin layers. The propagation between the layers is then described by
Fresnel propagation [BW06] in empty space. This method is implemented numerically first on a computer by Goodman and Moodie [GM74], then quickly becomes very popular thanks to the widely spread simulation wheels invented and expanded throughout the past decades [OB79, Koc02, DG03, KMO05, GRbdRHB10, RōM'11, GR13, KMZ'13, LVD15, ADF15, VJK15].

2.1 ELECTRON SOURCE

The electron source illuminating the specimen is one of the most important parts in a TEM. For an electron accelerated by a potential of $V_0$, it acquires a potential energy of $eV_0$. This energy is completely converted into kinetic energy, enabling this electron traveling at substantial
fraction of light speed $c$. With a typical potential difference of 100 keV, the electron is accelerated to approximately half of the light speed, the wavelength of this electron must be corrected in the framework of the special theory of relativity.

The relativistic mass $m$ of a fast electron traveling at a speed of $v$ can be obtained as

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

where $m_0$ is the electron rest mass.

The kinetic energy $E_0$ is governed by the conservation of energy

$$E_0 = eV_0 = mc^2 - m_0c^2.$$  

The de Broglie wavelength of the electron wave $\lambda$ is given by

$$\lambda = \frac{h}{mv},$$  

where $h$ is Plank constant, $m$ is the relativistic mass and $v$ is the speed of the accelerated electron.

Thereby we eventually obtain the wavelength $\lambda_0$ as

$$\lambda_0 = \frac{hc}{\sqrt{eV_0(2m_0c^2 + eV_0)}}.$$  

or in a numerical electron wavelength form

$$\lambda_0 \approx \frac{12.3984244}{\sqrt{E_0(E_0 + 1021.99812)}},$$  

where $E_0$ is given in keV, and $\lambda_0$ is obtained in Å.

The magnitude of the wavevector $\vec{k}$ is the inverse of the wavelength. The wavevector in vacuum, $k_0^*$, is given by

$$|k_0^*| = \frac{1}{\lambda_0},$$
and the direction is that of the incident beam.

The electrons traveling through the specimen have a bit higher energy than the ones in vacuum, because the electron is further accelerated by the mean inner potential of the material it passes through. This effect can be accounted for by a small change in the acceleration potential energy of $eV_s$, giving the corresponding wave length in the material

$$\lambda = \frac{hc}{\sqrt{e(V_0 + V_s)(2m_0c^2 + e(V_0 + V_s))}}.$$

And the plane electron wave function must have the form

$$\psi(r, \mathbf{k}) = e^{i2\pi \mathbf{k} \cdot \mathbf{r}},$$

where $\mathbf{r}$ is a position vector and $\mathbf{k}$ is the wave vector in specimen.

## 2.2 The Bloch wave method

The eigenvalue decomposition based solutions for Bloch wave method [Bet28] has become quite popular over the last century, with numerous extensions, implementations and enhancements developed [Fuj39, HW61, Stu62, Tou62, Met73, Fuj78, Hum79, FR81, Kam82, Ich83, SOBS83, VD85, Sta87, PW90, SZ92, Wat93, ZW95, ZMo04, KMo05, DG03, WDG16, YGS+16, Kir16]. In this thesis, we simulate the diffraction patterns using matrix exponential, which is fully reviewed by Metherell [Met73], Humphreys [Hum79] and Kirkland [Kir10], and in latter theory deductions of this section we inherit the notation used in *Electron Microdiffraction* [SZ92] and *Introduction to Conventional Transmission Electron Microscopy* [DG03]. The matrix exponential is easier to accelerate in a workstation equipped with multiple GPUs than matrix diagonalization, due to the fact that matrix multiplication optimization is much more mature than matrix decomposition with modern GPU architecture [NTD10, NBGS10, PWK14].

Providing the electron wavelength and mass are relativistically corrected, and electron spin is neglected, the time-independent Schrödinger equation can be used to describe high energy electron diffraction. The electron wave function can generally be expressed as a linear combination of any complete basis sets. If the selected basis sets also satisfy the Schödinger equation in the speci-
men, these bases are called Bloch-waves. The Bloch-waves combination takes a form of

\[ \Psi(\vec{r}, \vec{k}_t) = \sum_j \alpha_j b_j(\vec{k}_j, \vec{r}), \]

where \( \vec{k}_t \) is the incident wave vector, \( \alpha_j \) are Bloch-wave excitation amplitudes and \( b_j(\vec{k}_j, \vec{r}) \) are Bloch-waves.

As elastic scattering is assumed, each wave \( b_j(\vec{k}_j, \vec{r}) \) satisfies the conventional Schrödinger equation in the crystal specimen

\[ \left[ -\frac{\hbar^2}{2m} \Delta - eV_s(\vec{r}) \right] b_j(\vec{k}_j, \vec{r}) = E_0 b_j(\vec{k}_j, \vec{r}) \]

Substitute the kinetic energy \( E_0 \), this equation can be rewritten as

\[ \left[ \Delta + 4\pi^2 K_t^2 \right] b_j(\vec{k}_j, \vec{r}) = -\frac{8\pi^2 m e}{\hbar^2} V_s(\vec{r}) b_j(\vec{k}_j, \vec{r}), \]

where \( K_t = 1/\lambda \) is the relativistic wavenumber.

As each Bloch-wave \( b_j(\vec{k}_j, \vec{r}) \) is basically a plane wave having the same period as the crystalline specimen, they can be expanded as Bloch-wave Fourier combinations taking a form of

\[ b_j(\vec{k}_j, \vec{r}) = e^{i2\pi \vec{k}_j \cdot \vec{r}} \sum_{\vec{g}} C_{\vec{g},j} e^{i2\pi \vec{g} \cdot \vec{r}} = \sum_{\vec{g}} C_{\vec{g},j} e^{i2\pi (\vec{k}_j + \vec{g}) \cdot \vec{r}}, \]

where \( \vec{g} \) are typically the reciprocal lattice vectors of the specimen, and \( C_{\vec{g},j} \) are coefficients for each Bloch-wave \( b_j(\vec{k}_j, \vec{r}) \).

The specimen potential \( V_s(\vec{r}) \) can also be expanded into a Fourier series enforcing the same period.
where

\[ V_h(\vec{r}) = \sum_h V_h e^{i2\pi \vec{r} \cdot \vec{h}} \]

in which \( \vec{h} \) are the reciprocal lattice vectors of the specimen, \( f_{ej}(|\vec{h}|) \) is the electron scattering factor in the first Born approximation of the \( j \)th atom, the accumulation over \( j \) is for all the atoms resided in the unit cell, and \( \Omega \) is the volume of the unit cell.

Substituting specimen potential \( V_s(\vec{r}) \) and Bloch-wave linear expansion \( b_j(\vec{k}_j, \vec{r}) \), the Schrödinger equation yields

\[ \sum_g (|\vec{k}_g|^2 - |\vec{k}_j + \vec{g}|^2) C_{g,j} e^{i2\pi (\vec{k}_j + \vec{g}) \cdot \vec{r}} = -\sum_h \left[ \sum_{\vec{g} - \vec{h}} U_{\vec{g} - \vec{h}} C_{\vec{h},j} \right] e^{i2\pi (\vec{k}_j + \vec{g}) \cdot \vec{r}}, \]

where \( \vec{r} \) are the position vectors and the structure factors \( U_{\vec{g} - \vec{h}} \) are defined as

\[ U_{\vec{g} - \vec{h}} = \frac{2meV_{\vec{g} - \vec{h}}}{\hbar^2}. \]

Equating all the coefficients of \( e^{i2\pi (\vec{k}_j + \vec{g}) \cdot \vec{r}} \), this yields

\[ (|\vec{k}_g|^2 - |\vec{k}_j + \vec{g}|^2) C_{g,j} + \sum_{\vec{g}} U_{\vec{g} - \vec{h}} C_{\vec{h},j} = 0. \]

Or in a matrix equation form,

\[
\begin{pmatrix}
  \vdots \\
  \langle \vec{k}_g^2 - |\vec{k}_j + \vec{g}|^2 + U_{\vec{g},\vec{h}} \rangle \\
  U_{\vec{g},\vec{h}} \langle \vec{k}_g^2 - |\vec{k}_j^2 + \vec{g}^2 + U_{\vec{g},\vec{h}} \rangle \\
  \vdots \\
  U_{\vec{g},\vec{h}} \langle \vec{k}_g^2 - |\vec{k}_j^2 + \vec{g}^2 + U_{\vec{g},\vec{h}} \rangle \\
  \vdots \\
  \langle \vec{k}_g^2 - |\vec{k}_j^2 + \vec{g}^2 + U_{\vec{g},\vec{h}} \rangle \\
  \vdots \\
\end{pmatrix}
\begin{pmatrix}
  C_{g,j} \\
  C_{g,j} \\
  C_{g,j} \\
  C_{g,j} \\
  C_{g,j} \\
  C_{g,j} \\
\end{pmatrix}
= 0.}

As the typical acceleration voltage \( V_0 \) is of order of 100 kV, about 4 orders higher than the inner potential \( V_s \), the wave vectors \( \vec{k}_g \) are far larger than the reciprocal lattice vectors \( \vec{g} \).
Reciprocal lattice vector

\[ |\mathbf{k}'_i| \approx |\mathbf{k}_j| >> |\mathbf{g}|. \]

Suppose the Bloch wave vector \( \mathbf{k}_j \) can be written as \( \mathbf{k}'_r \) plus a small change along the beam direction \( \hat{z} \), then wave vector approximation can be employed for simplification.

Wave vector approximation

\[ \mathbf{k}_j \approx \mathbf{k}'_r + \mathbf{y}_j \cdot \hat{z}. \]

In this case, the diagonal elements of structure matrix can be further approximated as

Diagonal element of structure matrix

\[
\begin{align*}
|\mathbf{k}'_r|^2 - |\mathbf{k}_j + \mathbf{g}|^2 &\approx |\mathbf{k}'_r|^2 - |\mathbf{k}'_r + \mathbf{g}|^2 - 2\mathbf{y}_j \cdot (\mathbf{k}'_r + \mathbf{g}) \cdot \hat{z} - \mathbf{y}_j^2 \\
&\approx 2k'_rs_g - 2\mathbf{y}_j \cdot (\mathbf{k}'_r + \mathbf{g}) \cdot \hat{z} \\
&\approx 2k'_rs_g - 2\mathbf{y}_j (\mathbf{k}'_{r.z} + g_z) \\
&\approx 2k'_rs_g - 2\mathbf{y}_j \mathbf{k}'_{r.z},
\end{align*}
\]

where

\[ 2k'_rs_g = |\mathbf{k}'_r|^2 - |\mathbf{k}_r + \mathbf{g}|^2 = -2\mathbf{k}'_{r.z}g_z - |\mathbf{g}|^2, \]

in which the scalar \( \mathbf{k}'_{r.z} \) represents the component of incident wavevector \( \mathbf{k}'_r \) along \( \hat{z} \) direction, and \( g_z \) denoting same direction component of the reciprocal space vector \( \mathbf{g} \).

The matrix equation is then further approximated to a deduced matrix equation taking a form of

Deduced matrix equation

\[
\begin{pmatrix}
2k'_r & u_{r.g} & u_g & u_{g.r} & u_{g.e} & \cdots \\
u_{r.g} & 2k'_r & u_g & u_{g.r} & u_{g.e} & \cdots \\
u_g & u_g & 2k'_r & u_{g.r} & u_{g.e} & \cdots \\
u_{r.e} & u_{g.r} & u_{g.e} & 2k'_r & u_{g.e} & \cdots \\
u_{e.g} & u_{g.e} & u_{g.e} & u_{e.g} & 2k'_r & \cdots \\
u_{r.e} & u_{g.e} & u_{g.e} & u_{e.g} & u_{e.g} & \cdots
\end{pmatrix}
\begin{pmatrix}
C_{g,1} \\
C_{r,1} \\
C_{r,2} \\
C_{e,1} \\
C_{e,2} \\
C_{g,2}
\end{pmatrix} = 2\mathbf{y}_j \mathbf{k}'_{r.z},
\]

in which the mean inner potential \( U_{(0,0,0)} \) is omitted, due to the fact that it substantially serves as
a pure phase factor common to all reflections. Or in a eigenvalue matrix equation form

\[ A \hat{\mathcal{C}} = 2y_j \hat{\mathcal{R}}_{\tau,x} \hat{\mathcal{C}}, \]

where the column vector \( \hat{\mathcal{C}} \) is an eigenvector of the square complex matrix \( A \), and the scalar \( 2y_j \hat{\mathcal{R}}_{\tau,x} \) is the eigenvalue corresponding to \( \hat{\mathcal{C}} \).

Recall the Bloch-wave combination

\[ \Psi(\vec{r}, \vec{\kappa}_r) = \sum_j a_j b_j(\vec{k}_j, \vec{r}), \]

and the Bloch-wave Fourier combinations

\[ b_j(\vec{k}_j, \vec{r}) = \sum_{\vec{\delta}} C_{\vec{g},j} e^{i2\pi(\vec{k}_j + \vec{\delta}) \cdot \vec{r}}. \]

The electron wave function as a combination of all Bloch-waves \( b_j(\vec{k}_j, \vec{r}) \) can be explicitly written as

\[ \Psi(\vec{r}, \vec{\kappa}_r) = \sum_j a_j \sum_{\vec{\delta}} C_{\vec{g},j} e^{i2\pi(\vec{k}_j + \vec{\delta}) \cdot \vec{r}}. \]

And if the wavevector \( \vec{k}_j \) is further supplanted by the simplified approximation, \( \vec{k}_j \approx \vec{\kappa}_r + y_j \hat{\mathbf{a}}, \) the wave function \( \Psi(\vec{r}, \vec{\kappa}_r) \) can be rearranged as

---

**Explicit Bloch-wave Combination**

\[ \Psi(\vec{r}, \vec{\kappa}_r) \approx \sum_{\vec{\delta}} \sum_j a_j C_{\vec{g},j} e^{i2\pi(\vec{\kappa}_r + y_j \hat{\mathbf{a}} + \vec{\delta}) \cdot \vec{r}} \]

\[ \approx \sum_{\vec{\delta}} \sum_j a_j C_{\vec{g},j} e^{i2\pi y_j \hat{\mathbf{a}} \cdot \vec{r}} e^{i2\pi(\vec{\kappa}_r + \vec{\delta}) \cdot \vec{r}} \]

\[ \approx \sum_{\vec{\delta}} \psi_{\vec{g}}(t, \vec{\kappa}_r) e^{i2\pi(\vec{\kappa}_r + \vec{\delta}) \cdot \vec{r}}, \]

with the Fourier coefficients of the wave function

\[ \psi_{\vec{g}}(t, \vec{\kappa}_r) = \sum_j a_j C_{\vec{g},j} e^{i2\pi y_j \hat{\mathbf{a}} \cdot \vec{r}} = \sum_j a_j C_{\vec{g},j} e^{i2\pi \eta_j t}, \]
where $t = \bar{z} \cdot \vec{r}$ is the specimen thickness along the $z$ direction. It is therefore convenient to rewrite this in matrix form

$$
\begin{pmatrix}
\psi(t, \vec{r}) \\
\psi_0(t, \vec{r}) \\
\psi_{(00)}(t, \vec{r}) \\
\psi_{-d}(t, \vec{r}) \\
\psi_{+d}(t, \vec{r}) \\
\end{pmatrix} =
\begin{pmatrix}
C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\
C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\
C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\
C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\
C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \\
\end{pmatrix}
\begin{pmatrix}
\lambda(t) \\
\lambda_{(00)} \\
\lambda_{-d} \\
\lambda_{+d} \\
\end{pmatrix}
\begin{pmatrix}
\left[ \ldots , e^{i2\pi y_2 t}, e^{i2\pi y_3 t}, e^{i2\pi y_1 t}, e^{i2\pi y_1 t}, \ldots \right] \cdot \vec{d}
\end{pmatrix}
$$

where $\lambda(t)$ denotes a diagonal matrix $\text{diag}\{\ldots , e^{i2\pi y_2 t}, e^{i2\pi y_3 t}, e^{i2\pi y_1 t}, e^{i2\pi y_1 t}, \ldots \}$, and $\vec{d}$ is a column array composed of the Bloch-wave excitation coefficients $a_j$.

At the incident surface of the specimen, the corresponding thickness is $t = 0$, the Fourier coefficients $\psi_0(t = 0)$ of the incident plane wave can be represented as a column vector with all elements being zero, except for the $m^{th}$ one representing no transverse momentum, which is the central element in an expression below

**Incident wave function**

$$
\psi(t = 0) = [\ldots, 0, 0, 1, 0, 0, \ldots]^T.
$$

Due to the fact that $\psi(t = 0)$ does not depend on $\vec{r}$, and $\lambda(0)$ is an identity matrix, this leads to a resulting equation of

$$
\begin{pmatrix}
\psi_0(0) \\
\psi_{(00)}(0) \\
\psi_{-d}(0) \\
\psi_{+d}(0) \\
\end{pmatrix} =
\begin{pmatrix}
0 \\
1 \\
0 \\
0 \\
\end{pmatrix}
\begin{pmatrix}
C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\
C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\
C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\
C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\
\end{pmatrix}
\begin{pmatrix}
a_2 \\
a_3 \\
a_4 \\
a_5 \\
\end{pmatrix}
$$

As a straightforward result, the Bloch-wave excitation coefficients $a_j$ is attainable by

$$
\vec{d} =
\begin{pmatrix}
a_2 \\
a_3 \\
a_4 \\
a_5 \\
\end{pmatrix} =
\begin{pmatrix}
C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\
C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\
C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\
C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\
\end{pmatrix}^{-1}
\begin{pmatrix}
0 \\
0 \\
0 \\
1 \\
0 \\
\end{pmatrix}
= C^{-1} \psi(t = 0).
The Fourier coefficients $\psi_g(t, \mathbf{K}_r)$ of the exit wave at thickness $t$ is then subsequently deduced to

$$\psi(t, \mathbf{K}_r) = C \cdot A(t) \cdot \mathbf{A}$$

$$= C \cdot A(t) \cdot C^{-1} \psi(t = 0)$$

$$= e^{\frac{i\pi A(K_r)}{2k}} \psi(t = 0)$$

$$= S(t, \mathbf{K}_r) \psi(t = 0),$$

where $S$ is a scattering matrix.

For incident beam direction nearly parallel to the surface normal $\hat{z}$ of the crystalline slab, the scattering matrix can be further approximated to

$$S(t, \mathbf{K}_r) \approx e^{i\pi A(K_r)}.$$

With this definition, the intensity measured at reflection $g_n$ in the diffraction intensity can be written as

$$I_n(t, \mathbf{K}_r) = |\psi(t, \mathbf{K}_r)_{n}|^2 = |S(t, \mathbf{K}_r)_{n,m}|^2,$$

where the subscript $m$ in the scattering matrix $S(t, \mathbf{K}_r)_{n,m}$ is exactly the same index in $\psi(t = 0)$ where the corresponding value is 1.

Back to the explicit Bloch-wave combination

$$\Psi(t, \mathbf{K}_r) \approx \sum_g \psi_g(t, \mathbf{K}_r) e^{i2\pi K_r \cdot \mathbf{g}}$$

$$\approx \sum_g \psi_g(t, \mathbf{K}_r) e^{i2\pi \mathbf{K}_r \cdot \mathbf{g}} e^{i2\pi \mathbf{K}_r \cdot \mathbf{r}}.$$
Dropping the high frequency term $e^{i2\pi \kappa \cdot r}$, and employing a two dimensional inverse Fourier transform, the exit wave eventually can be approximated to

$$\Psi(r', \kappa') \approx \text{FT}^{-1}_{\xi, \eta} \left[ \sum_{\vec{g}} \psi_{\vec{g}}(t, \kappa_t) e^{i2\pi \vec{g} \cdot \vec{r}_t} \right].$$

### 2.3 Matrix Exponential Approximations

In mathematics, the matrix exponential is a matrix function given by the power series analogous to the ordinary exponential function

$$e^A = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots + \frac{A^n}{n!} + \ldots$$

In principle, the matrix exponential in scattering matrix can be approximated in a number of ways, including direct Taylor expansion, Padé approximation, scaling-and-squaring, Chebyshev rational approximation, single and multiple step(s) ordinary differential equation (ODE) solvers, eigenvalue decomposition, and splitting method. And dozens of methods for this purpose [Le096, MVL03] can be obtained from more or less classical results in numerical approximation [War77, ACF96, vHo6, HL97, Hig09, Bre96, War77, Saa92, Opp88], matrix theory [GVL12, ZMK02, PL10, IZ05] and specific applications [BVoo, PWK14].

In this work, a Padé approximation based scaling-and-squaring method is implemented in CPU space, and an optimized Taylor truncation based scaling-and-squaring method is implemented in GPU space.

### Padé Approximation

Mathematically Padé approximation performs very well in evaluating a function that is a rational function of a given order, and the approximant’s power series agrees with the power series of the function it is approximating.

To approximate the matrix exponential $e^A$, two polynomials are constructed
\[ P_r(A) = \frac{N(A)}{M(A)} = \frac{a_0 + a_1 A + a_2 A^2 + \ldots + a_n A^n}{b_0 + b_1 A + b_2 A^2 + \ldots + b_m A^m} \]

where the \( \{a_n\} \) and \( \{b_m\} \) series are selected as

\[ a_j = \frac{(m + n - j)!n!}{(m + n)!j!(n - j)!} \]
\[ b_j = \frac{(m + n - j)!m!}{(m + n)!j!(m - j)!} (-1)^j. \]

For this set of specified parameters, \( N(A) \) tends to \( e^{A/2} \) when increasing \( n \), and \( M(A) \) tends to \( e^{-A/2} \) when \( m \) increases.

In the approximation above, to eliminate round-off errors as much as possible, a scaling-and-squaring method is employed to pre-process and post-process the matrix \( A \). Detailed \( \{a_n\} \) and \( \{b_m\} \) parameters used in this approximation are listed in Algorithm 1 on page 32.

**Optimized Taylor Polynomial Truncation**

Padé approximation requires inversion of a matrix, and this would be problematic if this matrix is poorly conditioned, even if add a small element to its diagonal, the evaluation of the matrix inverse is still computationally expensive.

Providing the \( L_2 \) norm of the matrix \( A \) is small enough, the \( n \)th term \( \frac{A^n}{n!} \), tends to 0, this case, a polynomial truncation of the conventional matrix exponential can also be employed to approximate the matrix exponential in the following way

\[ e^A \simeq I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \ldots + \frac{A^n}{n!}, \]

which requires \( n - 1 \) matrix-matrix multiplications.

To accelerate the computation of the truncated approximation above, a polynomial-polynomial multiplication optimization trick can be used. This trick starts with the solution of a polynomial equation
which can be easily solved with root-finding algorithms either numerically or analytically [Ost66, OR70, Act90], or even with optimization algorithms [DJS96].

Suppose the solutions of the polynomial equation are \( \{c_1, c_2, \ldots, c_n\} \), the polynomial truncation then can be written as

\[
\mathcal{A} \approx \frac{1}{n!} (A - c_1)(A - c_2) \cdots (A - c_n),
\]

which requires \( n - 1 \) multiplications as before.

However, if the truncation is written in a form of two-ordered polynomial series multiplications

\[
\mathcal{A} \approx \frac{1}{n!} [A^2 - (c_1 + c_2)A + c_1 c_2] [A^2 - (c_3 + c_4)A + c_3 c_4] \cdots
\]

\[
[A^2 - (c_{n-2} + c_{n-1})A + c_{n-2} c_{n-1}],
\]

this would only require \( \left\lceil \frac{n}{2} \right\rceil \) multiplications, as \( A^2 \) can be pre-calculated and reused. And for the three-ordered optimized polynomial truncation

Three-ordered Optimized Polynomial Truncation

\[
\mathcal{A} \approx \frac{1}{n!} [A^3 - (c_1 + c_2 + c_3)A^2 + (c_1 c_2 + c_2 c_3 + c_3 c_4)A - c_3 c_4] \cdots
\]

\[
[A^3 - (c_{n-1} + c_{n-2} + c_{n-3})A^2 + (c_{n-1} c_{n-2} + c_{n-2} c_{n-3} + c_{n-3} c_{n-4})A - c_{n-3} c_{n-4}],
\]

only \( 1 + \left\lceil \frac{n}{3} \right\rceil \) multiplications required.

Figure 2.2 shows the performance benefit obtained from a single GPU-based CUDA acceleration [PWK14]. The purple and the green curves present the computation performance on the CPU, where
The GPU-based CUDA code is about 2 orders of magnitude faster than the parallelized CPU code [PWJ14].

- The purple curve is a single-threaded native C++ implementation, using Padé approximation of order 14. And

- The green one is a LAPACK [ABB+99]/BLAS [LHKK79]-based implementation, using 4 CPUs and the SciPy Python package, using Padé approximation of order 7.

The blue and red curves show the performance on the GPU, where

- The blue one is based on CULA [HPS+10] and is coded in Python, using Padé approximation of order 7. And

- The red one is a native CUDA implemented in C++, using scaling-squaring algorithm combining optimized polynomial expansion of order 9.

Some of electron diffraction patterns generated by our GPU-based approximation are rich of details and are visualized in Figure 2.3, 2.4 and 2.5.
Fig. 2.3: CBED Si(001) simulation, with thickness of 130 nm and a max tilt angle of 8 mrad at 120 keV.
Fig. 2.4: CBED Si(110) simulation, with thickness of 130 nm and a max tilt angle of 4 mrad at 120 keV.
Fig. 2.5: LARBED TINb$_7$O$_{14}(001)$ pattern simulation, with thickness of 10 nm, max tilt angle of 70 mrad and post-specimen beam-tilt compensation of 99.2% at 120 keV.
The GPU based exponential implementation employs this optimized polynomial truncation approximation trick inside a scaling-and-squaring method. All the details and parameters are listed in Algorithm 2 on page 33.

The evaluation of the matrix exponential is quite computation intensive. Possible solutions for this problem could be

A **Hardware accelerator**, FPGA for example, but is very difficult to design and program on;

B **Multicore CPU**, but with small speedup;

C **Grid computing**, very expensive; and

D **GPU**, simple, cheap and commercially available.

LARBED simulation requires evaluation of thousands matrix exponentials simultaneously, and depends on the beams selected, their sizes are usually limited by several hundreds. Benefiting from the revolutionary design and implementation changes in commodity processor architectures [NBGS08], the significant computational resources consumed by dynamical diffraction pattern simulations can be accelerated on modern GPUs. The CUDA toolkit allows end users to leverage the power of the GPUs using a general-purposed C/C++ programming language without the expense and complexity of hardware-specified assembly language, enabling dramatic increases in computing performance by harnessing the power of the GPUs.

CUDA’s dynamic parallelism is a feature that aims to improve the performance of data-dependent functions on GPUs and is supported since the release version 5. In the previous release of CUDA, only CPU code can determine when and how to launch subsequent computations based on the particular algorithm dependencies. But with the dynamic parallelism feature, the GPU space code can freely determine when and how to launch matrix-matrix computations based on the control flow of the matrix exponential algorithm, without CPU interruption.

In this work, a parallel framework is implemented in GPU space, with two layers of parallelism coded. The first layer involves computing multiple matrix exponentials simultaneously, as every beam tilt in LARBED pattern simulation would lead to a modification of the diagonal elements in structure matrix $A$, and sequential computation of the corresponding scattering matrices $S$ is computationally inefficient, especially for the case of thousands of beam tilts. The second
layer involves matrix-matrix multiplication parallelism. The matrix-matrix product is such a fundamental linear algebra routine, many optimization methods have been published from algorithm levels\cite{Str69, CU03, CKSU05, Will12, Will14} to application levels\cite{NTD15}. The implementation selected in this layer simply follows the classical CUDA matrix-matrix multiplication algorithm listed in textbooks or CUDA manuals, considering shared memory, thread wrapping, memory alignment, matrix partitioning and other optimization tricks \cite{Coo12, KH13, Bar14}.

The details of this method is demonstrated in Figure 2.6. With this method, the computation time for the fitting program is reduced from weeks to hours.

### 2.4 SIMULATION OUTLINE

This section outlines the computation routine for the electron diffraction numerical simulation, taking the unit cell of the specimen (including the positions of all the atoms), the thickness of the specimen, the acceleration voltage and the tilt angle of the incoming electron beam as input.
Known information

Unit cell of the specimen, which is represented in terms of lattice parameters, including the lengths of the cell edges and the angles between these edges.

Zone axis and surface normal of the specimen;

Thickness of the specimen \( t \);

Acceleration voltage;

Tilt angle of the incident beam relative to the zone axis;

Select reflections \( \mathcal{R} \) to be included in the simulation, which means only these reflection intensities will be calculated in the program. Or in another word, select a set of reflections and place them sequentially in a preselected column of the structure matrix \( \mathbf{A} \). And reflection \( \{0,0,0\} \) must be included.

Simulation

Generate full reflection set \( \mathcal{S} \) that will be used in the simulation, including two steps

1. Transform all reflections from \( \mathcal{R} \) to \( \mathcal{S} \), now set \( \mathcal{S} \) holds all the reflections in the selected column of the structure matrix \( \mathbf{A} \);

2. Enumerate all reflection pairs \( (\mathbf{g}, \mathbf{h}) \) from set \( \mathcal{R} \), and for each pair, insert their difference, \( \mathbf{g} - \mathbf{h} \), into \( \mathcal{S} \);

Calculate the electron wavelength \( \lambda \) for the acceleration voltage \( V_0 \);

Calculate all the structure factors for reflections in set \( \mathcal{S} \);

Calculate the diagonal elements in structure matrix \( \mathbf{A} \) with incident beam tilt;

Generate the structure matrix \( \mathbf{A} \) with calculated structure factors and diagonal elements;

Calculate the scattering matrix \( \mathbf{S} \) with wavelength \( \lambda \), specimen thickness \( t \) and structure matrix \( \mathbf{A} \), using scattering matrix formulation;

Calculate the diffraction intensity from the moduli of the preselected column in scattering matrix \( \mathbf{S} \).

Details of the LARBED patterns simulation routines are listed in Figure 2.7 on page 31.
Fig. 2.7: Detailed LARBED patterns simulation routine.
Data: complex structure matrix $A$, thickness $t$, wavelength $\lambda$

Result: Scattering matrix $S$

1. $A_1 \leftarrow i\pi\lambda t A$
2. $s_0 \leftarrow \|A_1\|_2$
3. $\sigma \leftarrow 5.371920351148152$
4. $s_1 \leftarrow max \left(1, \left|\log_2 \left(\frac{s_0}{\sigma}\right)\right|\right)$
5. $A_2 \leftarrow \frac{A_1}{s_1}$
6. $c \leftarrow \{64764752532480000, 32382376266240000, 77717703038976000, 11873537964288000, 1290601952640000, 105594705216000, 6704425728000, 33522128640, 1323241920, 408408000, 960960, 16380, 182, 1\}$
7. $U \leftarrow A_2 (A_2^6 (c_{14} A_2^6 + c_{12} A_2^4 + c_{10} A_2^2) + c_8 A_2^8 + c_6 A_2^6 + c_4 A_2^4 + c_2)$
8. $V \leftarrow A_2^6 (c_{13} A_2^6 + c_{11} A_2^4 + c_9 A_2^2) + c_7 A_2^8 + c_5 A_2^6 + c_3 A_2^4 + c_1$
9. $S \leftarrow \frac{V + U}{V - U}$

10. while $s_1 > 1$ do
11.     $S \leftarrow S^2$
12.     $s_1 \leftarrow s_1 - 1$
13. end
14. return $S$

Algorithm 1: evaluation of matrix exponential using the scaling and squaring algorithm combined with the Padé approximation.
Data: complex structure matrix $A$, thickness $t$, wavelength $\lambda$

Result: Scattering matrix $S$

1. $A_1 \leftarrow i \pi \lambda A$
2. $s_0 \leftarrow ||A_1||_2$
3. $\sigma \leftarrow 5.371920351148152$
4. $s_1 \leftarrow \max \left( 1, \left[ \log_2 \left( \frac{s_0}{\sigma} \right) \right] \right)$
5. $A_2 \leftarrow \frac{A_1}{s_1}$
6. $c_0 \leftarrow 2.697333461536989227389605 + 5.184162062649414177834087i$
   $c_1 \leftarrow -0.3810698456631129990312942 + 4.384644533145397950369203i$
   $c_2 \leftarrow -2.110839800302654737498705 + 3.089910928725500922777702i$
   $c_3 \leftarrow -3.038648072936697089212469 + 1.58680119575838328803868i$
   $c_4 \leftarrow -3.33551485269048803294274$
   $c_5 \leftarrow -3.038648072936697089212469 - 1.58680119575838328803868i$
   $c_6 \leftarrow -2.110839800302654737498705 - 3.089910928725500922777702i$
   $c_7 \leftarrow -3.810698456631129990312942 - 4.384644533145397950369203i$
   $c_8 \leftarrow 2.697333461536989227389605 - 5.184162062649414177834087i$
7. $S \leftarrow \frac{1}{s_1} \left( 1 + c_0 A_2 + c_1 A_2^2 + c_2 A_2^3 \right) \left( 1 + c_3 A_2 + c_4 A_2^2 + c_5 A_2^3 \right) \left( 1 + c_6 A_2 + c_7 A_2^2 + c_8 A_2^3 \right)$
8. while $s_1 > 1$ do
    9.     $S \leftarrow S^2$
10.    $s_1 \leftarrow s_1 - 1$
11. end
12. return $S$

Algorithm 2: evaluation of matrix exponential using scaling-squaring algorithm combing Taylor polynomial expansion.
The inversion problem is to find all the amplitudes and all their relative phases of structure factor $U_g$’s and other experiment related parameters from a series of observations of the diffraction intensities, $I$. A common approach is to minimize the square of the mismatch between the measurement and the simulation, i.e., to minimize a merit function $\chi^2$

$$\chi^2(U_g) = \sum_{k_i} \sum_{m} |I_{m}^{\exp} - I_{m}(U_g, \vec{k})|^2,$$

or a non-normalized R-value expression $R$

$$R(U_g) = \sum_{k_i} \sum_{m} |I_{m}^{\exp} - I_{m}(U_g, \vec{k})|.$$

This is a highly nonlinear problem, especially when the specimen is thick, which means a lot of dynamic diffraction. For a diffraction pattern including $N$ beams, there exist about $4N$ different complex parameters, i.e. $U_g$’s, to be fitted, and $N$ is usually of the order of one hundred or even more. The number of possible solutions in the search space is very large, therefore a brute-force based search algorithm is not technically feasible.
No free lunch theorem [WM095, WM97, Köpoo, KWM01, HP02, WW05] asserts that there does not exist a general-purpose, universal optimization strategy for all problems. Aware of this, we test several optimization methods with this very complex nonlinear optimization problem, considering different experimental related setups, including various heuristic optimization methods and classic gradient based methods.

### 3.1 Genetic Algorithm

A genetic algorithm (GA) is a typical heuristic mimicking the process of Darwin natural selection and Mendelian inheritance. It maintains a population of potential solutions, through successive iterations, applies selection on the whole population based on the fitness of individuals in a way of nonuniform Roulette game, then execute crossover and mutation process to generate a new population. As it performs well in approximating solutions to all types of problems without assuming the underlying target function landscape, GA is widely used in the many fields of numeric optimization [RE73, Dav91, GD91, DSM94, HNG94, HJK95, WPGC96, Mit98, B099, Col99, DAPM00, SD07].

In this work, three cases are tested with different population sizes and different diffracted beams, where the specimen is Silicon (001) with a thickness of 10 nm at 120 keV, 1009 beam tilts, a maximum tilt angle of 70 mrad and using the simulated patterns as target patterns. And the genes in this algorithm are represented in binary as 64-bit integers with Gray encoding, with shuffle crossover and the bit-flipping mutation method enabled.

Figure 3.2 on page 47 presents these three test cases, which are all running in a Linux workstation equipped with Intel(R) Xeon(R) CPU W3550 enabling parallelism acceleration. These cases are

- **A** 9 beams with a population of 256, the blue curve — The GA solver quickly converges to a very small residual in a few hours.

- **B** 9 beams with a population of 128, the green curve — The solver converges, but much slower when comparing with the blue one.

- **C** 25 beams with a population of 128, the red curve — The solver does not converge, and runs very slowly (several weeks).
From our numerical experiments, it can be concluded that GA tends to converge providing

A the population size is large enough,

B the evolution generation is large enough.

C the number of beams included in the fitting is not that large, and

D a computer powerful enough.

However, when including more beams, the GA solver does not scale well with the increasing complexity unless increasing population size in the swarm, and the repeated merit function evaluation dramatically slows down the performance, making the fitting not feasible with hardwares in current generation.

### 3.2 Gradient Based Iteration Algorithm

Iterative optimization methods always start with an initial guess $\mathbf{x}^{(0)}$, compute a search direction $\mathbf{p}^{(k)}$ and a step length $\alpha^{(k)}$, then iterate in a way like

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}.$$  

To ensure the search direction $\mathbf{p}^{(k)}$ is a descent direction, it must hold that

$$\mathbf{p}^{(k)\top}\mathbf{d}^{(k)} < 0,$$

where $\mathbf{d}^{(k)}$ is the merit function gradient at $\mathbf{x}^{(k)}$.

There are two kinds of classic iterative methods widely used for nonlinear optimization, the line search methods [Aka59, OR70, Luc73, MS79, Gol80, Lem81, AB85, ABF86, BNY87, MT94, DJS96, ZHO4, HZ05, WB06, BSS13, Bre13, Fle13] and trust region methods [Pow75, MS83, Mor83, Ste83, SSB85, BSS88, DS89, CL96, BGN00, CGToo]. The trust region methods first choose a step length then a direction, while the line search methods choose direction first and then iteration step length.
Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm, which is invented by K. Levenberg and later rediscovered by D. Marquardt [Lev44, FP63, Mar63], is widely used and considered as the semi-standard [BA77, Mor78, Yeh86, Low91, Bjö96, DJ96, Kel99, NW06, Bet0, Gavi1, Flet13] for least-square nonlinear problems in practice, for its robust and quick convergence.

This method approximates the merit function using a quadratic function, or more detailedly, this method updates the parameters in a way like

$$x^{(k+1)} = x^{(k)} - [D + \lambda I]^{-1} \Delta \chi^2(x^{(k)})$$

where $D$ is an approximation of the Hessian matrix using only the first-order derivative of the model, and $\lambda$ is the parameter that governs the step length of the search region. This strategy means a Hessian-like matrix $D$ needs maintaining and updating at every iteration.

For a merit function with $n$ parameters, one iteration requires $\frac{n(n-1)}{2}$ times evaluation of the first-order partial derivatives of the target function. This is quite computationally expensive, especially when there are thousands of parameters needing refining, as in our case. Therefore, we switch to line search methods, which determines the search direction $\rho^{(k)}$ first, then the step length $\sigma^k$.

Conjugate Gradient Method

In a line search scheme, the search direction can be computed by many methods, such as gradient descent, Newton’s method and Quasi-Newton methods. In this work, the conjugate gradient method is employed, as gradient descent is relatively slow when close to the minima, and Newton’s method and Quasi-Newton methods requires a Hessian (or a Hessian-approximated) matrix for the second-order derivatives. The conjugate gradient method is initially proposed by M. R. Hestenes and E. Stiefel [HS52], intended to solve a linear equation in a form of

$$Ax = b,$$
where $A$ is a square symmetric positive definite matrix. It can be stated that this equation is equivalent to a minimization problem taking a form of

$$\chi'^{(b)} = \frac{1}{2}\dot{x}^T A \dot{x} - \dot{b}^T \dot{x},$$

as they share the same unique solution. This equivalence can be interpreted that solving a linear equation and minimizing a convex quadratic function can be done in a mutual way.

In this work, conjugate gradient method is employed to compute the search direction $p^{(k)}$. At the first step, $p^{(0)}$ is simply the opposite direction of the gradient of the initial guess $\chi^{(0)}$

$$p^{(0)} = -\left\{ \frac{\partial \chi^2(\chi^{(0)})}{\partial x_0^{(0)}}, \frac{\partial \chi^2(\chi^{(0)})}{\partial x_1^{(0)}}, \ldots, \frac{\partial \chi^2(\chi^{(0)})}{\partial x_{n-1}^{(0)}} \right\}^T,$$

and at the intermediate step $k$, the conjugate direction $p^{(k+1)}$ is updated with a parameter $\beta$

$$p^{(k+1)} = \beta p^{(k)} - d^{(k)}$$

where $d^{(k)}$ is the numerical gradient at $\chi^{(k)}$

$$d^{(k)} = \left\{ \frac{\partial \chi^2(\chi^{(k)})}{\partial x_0^{(k)}}, \frac{\partial \chi^2(\chi^{(k)})}{\partial x_1^{(k)}}, \ldots, \frac{\partial \chi^2(\chi^{(k)})}{\partial x_{n-1}^{(k)}} \right\}^T.$$

There are numbers of methods [HS52, FR64, LMW67, ERE69, Pol69, Lue73, CGO76, Fle76, Pow77, Ker78, Sha78, O’L80, YJ80, EIS81, Ste83, NL87, LS91, GN92, M093, She94, CN96, Gon97, DY99, HZ05] designed to calculate the conjugate gradient direction factor $\beta$ for unconstrained nonlinear optimization problems. Some of the best known ones are

A. Fletcher-Reeves [FR64]

$$\beta = \frac{d^{(k)\top} d^{(k)}}{d^{(k-1)\top} d^{(k-1)}},$$

converges if the initial guess is close to the desired minimum.

B. Polak-Ribiére [ERE69, Pol69]
\[ \beta = \frac{d^{(k)} \top (d^{(k)} - d^{(k-1)})}{d^{(k-1)} \top d^{(k-1)}}. \]

might fail to converge, but if converging, it converges quickly.

**Hestenes-Stiefel [HS52]**

\[ \beta = \frac{d^{(k)} \top (d^{(k)} - d^{(k-1)})}{\rho^{(k-1)} \top (d^{(k-1)} - d^{(k)})}, \]

**Dai-Yuan [DY99]**

\[ \beta = \frac{d^{(k)} \top d^{(k)}}{\rho^{(k-1)} \top (d^{(k-1)} - d^{(k)})}, \]

convergences as long as strong Wolfe conditions [Wol69, Wol71] are satisfied.

**Hager-Zhang [HZ05]**

\[ \beta = \left( \frac{(d^{(k)} - d^{(k-1)}) - 2\rho^{(k)} \frac{d^{(k)} \top (d^{(k)} - d^{(k-1)})}{\rho^{(k)} \top (d^{(k)} - d^{(k-1)})}}{\rho^{(k-1)} \top (d^{(k-1)} - d^{(k)})} \right) \top d^{(k)}, \]

a global convergence result is established when the line search fulfills the Wolfe conditions.

In this work, Dai-Yuan’s scheme is used. The detailed conjugate gradient strategy for \( \rho^{(k+1)} \) updating is

\[ d^{(k)} = \left\{ \frac{\partial x^2(x^{(k)})}{\partial x_0^{(k)}}, \frac{\partial x^2(x^{(k)})}{\partial x_1^{(k)}}, \frac{\partial x^2(x^{(k)})}{\partial x_2^{(k)}}, \ldots, \frac{\partial x^2(x^{(k)})}{\partial x_{n-1}^{(k)}} \right\} \top, \]

\[ \beta_1 = d^{(k)} \top d^{(k)}, \]

\[ \beta_2 = d^{(k-1)} \top \beta^{(k)}, \]

\[ \beta_3 = d^{(k)} \top \beta^{(k)}, \]

\[ \beta = \frac{\beta_1}{\beta_2 - \beta_3}, \]

\[ \rho^{(k+1)} = \beta \rho^{(k)} - d^{(k)}. \]
With this scheme, only \( n \) evaluations of the first-order partial derivatives of the target function is required, and this is a big advantage comparing with the search algorithms using a quadratic approximation, which requires \( O(n^2) \) evaluations at every updating.

**Line Search**

As mentioned earlier, a line search method is one of basic iterative approaches to a nonlinear optimization problem, while the other one is trust region. For the optimization algorithm implemented in this work, the search direction is first determined from conjugate gradient method, and the step length is determined using line search algorithm.

Basically, the step length \( \alpha^{(k)} \) at step \( k \) is selected to minimize a modified target function

\[
\chi'(\alpha^{(k)}) = \chi^2 \left( x^{(k)} + \alpha^{(k)} \tilde{p}^{(k)} \right),
\]

where \( \alpha^{(k)} \) must be positive, which makes a one-dimensional search problem. The challenge here is to find a proper step length \( \alpha^{(k)} \) that is

A **not too large** – the decrease must be sufficient, otherwise, at local minima the iteration will vibrate and slows down; and

B **not too small** – or the iteration will be very slow, and will probably get stuck at one of the local minima.

This is is demonstrated in Figure 3.1.

The step length \( \alpha^{(k)} \), or the maximum amount to move along the calculated search direction \( \tilde{p}^{(k)} \), can be computed either exactly or inexactly.

The typical inexact line search methods [Arm66, Wol69, Wol71] involve Wolfe conditions, which are a combination of Armijo condition, curvature condition and Armijo-Goldstein conditions, but these methods require evaluating derivatives of the modified target function.

Exact line search methods are derivative-free, including golden section and Fibonacci search [Kie53, HJ61, AW66, Fer60] respectively, which are relatively easy to implement comparing with the inexact ones.
In practical computations, theoretically exact optima cannot be determined directly, and it is also considered very expensive to find a semi-exact one, therefore the inexact methods are more popular. But for our case, there are thousands of parameters to be fitted from the merit function, the direction search of $\hat{\rho}^{(k)}$ consumes most of the computation time, the benefit from inexact line search methods is neglectable.

As golden section and Fibonacci search only differ in the way where the trial step lengths are iterated, in this work, golden section method is employed. The details of our implementation is listed in Algorithm 4 on page 47.

### 3.3 DERIVATIVE EVALUATION

From our conjugate gradient strategy, to calculate the conjugate gradient of the merit function, evaluation of the derivatives, $\bar{j}^{(k)}$, is a must. In our case, the target function is too complicated, and it is not possible to calculate them analytically. There are a number of approaches for numerical derivative evaluation exist [RG27, Hen61, LM66, Ebe03, Liu06]. In this work, a combination
of Richardson Extrapolation method \([\text{RG27}]\) and Neville’s algorithm are employed.

The precise definition of the derivative is a limit of difference quotient

\[
f'(x) = \lim_{h \to 0} \frac{f(x + h) - f(x)}{h}.
\]

In practice it is usually approximated with central differential method

\[
f'(x) \approx \frac{f(x + h) - f(x - h)}{2h},
\]

but in this work, the derivatives are approximated in a more precise way.

Starting with the Taylor expansion, \(f(x + h)\) and \(f(x - h)\) can be expanded in polynomial forms of

\[
f(x + h) = f(x) + hf'(x) + \frac{1}{2} h^2 f''(x) + \frac{1}{6} h^3 f'''(x) + \frac{1}{24} h^4 f''''(x) + \ldots,
\]

and

\[
f(x - h) = f(x) - hf'(x) + \frac{1}{2} h^2 f''(x) - \frac{1}{6} h^3 f'''(x) + \frac{1}{24} h^4 f''''(x) + \ldots,
\]

therefore, \(f'(x)\) can be obtained from

\[
f'(x) = \frac{f(x + h) - f(x - h)}{2h} - \left[ \frac{h^2}{3} f''(x) + \frac{h^4}{120} f''''(x) + \ldots \right];
\]

Or in a central differential way

\[
f'(x) = \frac{f(x + h) - f(x - h)}{2h} + O(h^2)
\]

for short.

To further improve the central difference approximation, we rewrite this equation as

\[
f'(x) = E(h) + mh^2 + O(h^4),
\]
where \( E(h) = \frac{f(x+h) - f(x-h)}{2h} \) and \( m = -\frac{f''}{3} \). If \( h \) is replaced with \( 2h \), there we get

\[
f'(x) = E(2h) + 4mh^2 + O(h^4).
\]

Combing these two equations, the \( h^2 \) terms can be eliminated with

\[
f'(x) = \frac{4E(h) - E(2h)}{3} + O(h^4),
\]

which can then be expanded as

\[
f'(x) = -f(x + 2h) + 8f(x + h) - 8f(x - h) + f(x + 2h) + O(h^4).
\]

This approximation, with an error of \( O(h^4) \), is more precise than the approximation from central difference method. Using similar method, we can eliminate terms of \( h^4 \), \( h^6 \) and even higher order terms of \( h \), to further approximate the derivatives more precisely.

In our implementation, we approximate the derivatives to

\[
f'(x) = -f(x + 3h) + 9f(x + 2h) - 45f(x + h) + 45f(x - h) - 9f(x + 2h) + f(x + 3h) + O(h^6).
\]

For the derivative at a fixed point \( x \), the above equation can be seen as a function of \( h \), a sixth-order approximation

\[
f'(x) \approx g(h) = -f(x + 3h) + 9f(x + 2h) - 45f(x + h) + 45f(x - h) - 9f(x + 2h) + f(x + 3h)
\]

However, direct approximation this way is not numerically stable, as the step length \( h \) is not so easy to choose. If \( h \) is too large, error accumulates from the higher order terms; and if \( h \) is too small, the round-off error matters.

In this work, a polynomial interpolation algorithm based on Neville’s formula is implemented to obtain the derivatives precisely from the 6th-order approximation. This method starts from a series of trial step lengths

\[
\{h_0, h_1, \ldots, h_k, \ldots\}, \quad h_k \neq 0,
\]

3.3 Derivative Evaluation
and the series of the corresponding derivative approximations

\[ \{g(h_0), g(h_1), \ldots, g(h_k), \ldots\}, \]

then constructs an interpolating polynomial \( P_{k,n}(h) \) of order \( n \) to approximate the derivative at \( h = 0 \).

This method recursively constructs \( P_{k,n}(h) \) at \( h = 0 \) from very poor approximations of order 0

\[ P_{k,0} = g(h_k), \]

to good polynomial approximations at higher order of \( n \)

\[ P_{k,n} = \frac{h_k P_{k-1,n-1} - h_{k-n} P_{k,n-1}}{h_k - h_{k-n}}. \]

Detailed implementation is listed in Algorithm 3 on page 46.
Data: function $f$, variable $x$

Result: derivative of function $f$ at point $x$, $f'(x)$

1 Define:
$$g(h) = \frac{-f(x + 3h) + 9f(x + 2h) - 45f(x + h) + 45f(x - h) - 9f(x + 2h) + f(x + 3h)}{60h}$$

2 $d = \frac{1 + \sqrt{5}}{2};$
3 $h = d \times 10^{-4};$
4 $N = 32;$
5 Define:
$$P(i, j) = \begin{cases} g\left(\frac{h}{d^j}\right) & \text{if } j = 0, \\ \frac{h^j P_i - h^{j-1} P_{i-1}}{d^j} & \text{otherwise.} \end{cases}$$
6 Define:
$$\epsilon(i, j) = \max(|P(i, j) - P(i, j - 1)|, |P(i, j) - P(i - 1, j - 1)|);$$
7 for $i \leftarrow 1$ to $N$ do
8     for $j \leftarrow 1$ to $i$ do
9         if $|P(i, i) - P(i - 1, i - 1)| > d^2 \epsilon(i, j)$ then
10            return $P(i, j)$
11         end
12     end
13 end
14 return $P(N, N);$ 

Algorithm 3: Evaluation of a derivative.
Fig. 3.2: Convergence curve for three test cases with different configurations. Blue: 9 beams with 256 individuals; Red: 25 beams with 128 individuals; Green: 9 beams with 128 individuals.

Algorithm 4: Golden section algorithm.

```plaintext
golden_section(χ′, α₀, αₙ, ε):

Data: target function χ′, step length α₀, step length αₙ, precision ε

Result: step length α

1. A = χ′(α₀);
2. B = χ′(αₙ);
3. if |A − B| <= ε : return α₀;
4. φ = \(\sqrt{5} - 1\);
5. α₁ = (1 − φ)(αₙ − α₀) + α₀;
6. α₂ = φ(αₙ − α₀) + α₀;
7. C = χ′(α₁);
8. D = χ′(α₂);
9. if C > D : return golden_section(χ′, α₁, α₂, ε);
10. return golden_section(χ′, α₀, αₙ, ε);
```

Chapter 3: Inversion of Dynamical Scattering
This chapter presents the results of the numeric experiments designed to test the performance of our fitting program. This chapter also presents the fitting results when applying our program to acquired experimental data.

A fitting program implementing the numerical tools discussed last chapter is developed in C++ and runs at a Linux workstation including an Intel(R) Xeon(R) CPU E5-2650 and 3 Nvidia Tesla K20C GPUs and 1 Nvidia Tesla K40C GPU.

Generally speaking, this program takes a target function $f$ as input, then outputs $f$’s global minimum, a vector $\vec{x}$.

A series of LARBED patterns are simulated using Padé approximation combining scaling-squaring algorithm running in the CPU space, then the fitting program using optimized Taylor expansion combined with scaling-and-squaring method running in the GPU space to reconstruct the diffraction patterns.

In this chapter, all the figures with hotblue color scheme (color varies from White to Blue then to Black) are presented in logarithmus scale to give special emphasis to weak instensities (the right figure in Table 4.2 at page 62 for example); while the figures with bluehot color scheme (color varies from Black to Blue then to White) are in linear scale (the left figure in Table 4.2 at page 62 for example). And all the potential maps are present using bluehot with unit cells two by two.

In Table 2.7 on page 31, those nodes in red rectangular represents the parameters to be fitted, including the structure factors $U_q$s and specimen thickness; the pattern rotation angle is consid-
Fig. 4.1: A numeric experiment with a pseudo specimen inherited from a SrTiO$_3$ (001) crystal at 120 keV and a thickness of 12 nm, but adding two Silicon atoms to break the symmetry. The target potential field is demonstrated at the left most side; the reconstructed potential fields at specified iterations are visualized as well.

In our numeric experiments, and is not fitted. But when fitting the experimental patterns latter, the rotation angle and other experiment related parameters are considered.

A typical fitting procedure, including the target specimen potential field, the convergence curve and the reconstructed potential field at specified iterations, are demonstrated in Figure 4.1 on page 50. Before the fitting, a LARBED pattern is simulated with the pseudo structure factors, including 45 beams measured and 1009 tilt angles. The potential map corresponding to the structure factors is shown on the left part of the figure, which is a SrTiO$_3$ unit cell with two silicon atoms to break the symmetry. Then the fitting program randomly picks up an initial guess for the structure factors, which is corresponding to the potential field map at step 1. Our optimization program minimize the $\chi^2$ over iterations, and alongside this procedure, the potential maps corresponding to the fitted structure factors are becoming more and more regular. It can be seen that the potential map at step 5000 is almost identical to the initial potential field, despite a shift in the origin.
4.1 NUMERIC EXPERIMENTS: CONFIGURATION

To figure out the suitability and robustness of this program under different situations, we have considered seven scenarios of different LARBED pattern simulation parameters related to the target function $f$ in the performance benchmark, including

**Crystal species** Strontium titanate and silicon are tested. The silicon crystallizes in a diamond cubic crystal structure with a lattice spacing of 0.5430710 nm, while the lattice parameter for strontium titanate, which is in a perovskite structure, is 0.3905 nm. Their unit cells and potential fields are shown in Table 4.3 on page 62.

**Maximum tilt range** Four cases are tested, which are \{5 mrad, 10 mrad, 20 mrad, 40 mrad\}. The patterns with larger maximum tilt angles provide more details than the ones with smaller maximum tilt angles. This is very apparent by observing the silicon diffraction pattern figures shown in Table 4.5 on page 63.

**Noise** Noise free and Poission noise cases are tested. In our simulation program, the Poission noise is generated using the rejection method [Dev86] and the Mersenne Twister engine [MN98]. For each beam tilt, 1000 incident electrons per tilt are assumed. On page 63, Table 4.6 demonstrates these patterns for SrTiO$_3$.

**Total beam tilts** Seven beam tilt cases are tested, which are \{1024, 512, 256, 128, 64, 32, 16\}. Typical LARBED patterns with different beam tilts are demonstrated in Table 4.4 on page 63. These numbers are specified to fit the warp size of the CUDA mechanisms for best performance. As the intensities are generally weak with small number of beam tilts, all these patterns are presented in a logarithm scale.

**Specimen thickness** Five thicknesses are tested, including \{5 nm, 10 nm, 20 nm, 30 nm, 40 nm\}. Table 4.7 on page 64 shows SrTiO$_3$ patterns simulated with 11 nm and 12 nm at 120 keV. The difference between these two patterns suggests that the LARBED patterns are not very sensitive to thickness variance. The idea that using difference between data recorded at two slightly different acceleration voltages [SZ92], which is equivalent to two slightly different
thicknesses, is therefore not so promising when only matching the reconstructed intensities to experimental ones.

**Beams selected** On the beam selection method in the Bloch-wave simulation, a better one should consider the proximity to the Ewald sphere (the excitation error), length of $\vec{g}$ and perturbation strength length $[ZW95]$. In this experiment, two different beam selection schemes are tested in the context of weak intensities. One includes all beams up to a small cutoff length of $\vec{g}$, the other also includes the larger $\vec{g}$, but omits forbidden or very weak reflections. In accordance with previous experiements, there are 45 beams selected in this setup, which are visualized using Web-Emaps[ZMo4], and 157 $U_g$s are being fitted. As shown in Table 4.8 on page 64, beam selection strategies for SrTiO$_3$ and silicon are demonstrated. It should be emphasised that they are generated using the same maximum tilt range, i.e., all disks are exactly of the same size. The different disk size in these two figures occurs because of the different scaling factors.

**Target function** Two different target functions are tested in the numeric experiments, including the least-squared merit function

$$
\chi^2(U_g) = \frac{\sum_{\vec{k}_i} \sum_m |I^\text{exp}_m - I_m(U_g, \vec{k}_i)|^2}{\sum_{\vec{k}_i} \sum_m |I^\text{exp}_m|^2},
$$

and R-value function

$$
R(U_g) = \frac{\sum_{\vec{k}_i} \sum_m |I^\text{exp}_m - I_m(U_g, \vec{k}_i)|}{\sum_{\vec{k}_i} \sum_m |I^\text{exp}_m|}.
$$

Additionally, in latter section, we compose and test a 0-1 convex homotopy function, a simple linear combination of the merit function and the R-value function, hoping this new target function could take the advantages of its parents and outperform throughout all thicknesses.
4.2 NUMERIC EXPERIMENTS

As indicated in the previous section, we have seven degrees of freedom to test our fitting program, and the total combinations can go as large as 2240 (2 × 4 × 2 × 7 × 5 × 2 × 2). It is tedious to present the target diffraction pattern, the reconstructed diffraction pattern, the reconstructed potential field, the convergence curve and the quantitative evaluation (triplet error and R-value) of the fitting program for each case in the combination. Therefore we choose and present a subset of the results in groups covering every aspect of the listed seven scenarios. As the diffraction patterns for Silicon are very weak in most of the disks, we presented them in a log scale and a bluehot color scheme.

Testing maximum tilt range factor

In this unit test, we test the impact of maximum tilt range has on the performance of the fitting program, with a setup of

<table>
<thead>
<tr>
<th>Crystal</th>
<th>Maximum tilt range</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selenium tinmate</td>
<td>5 mrad, 10 mrad, 20 mrad, 40 mrad</td>
<td>noise free</td>
</tr>
<tr>
<td>Beam tilt</td>
<td>Thickness</td>
<td>Beams selected</td>
</tr>
<tr>
<td>10°</td>
<td>40 nm</td>
<td>40 beams</td>
</tr>
<tr>
<td>Target function</td>
<td>R-value function</td>
<td></td>
</tr>
</tbody>
</table>

Experiment results of this unit are presented in Table 4.9 on page 65. When the tilt range is smaller, i.e., in the 5 mrad and 10 mrad case, the fitting program gets stuck at a local minimum, and fails to reconstruct the correct potential field. But when it is larger, i.e., 20 mrad and 40 mrad, the fitting program successfully regenerates the potential field for the SrTiO₃ crystal. The result also shows the fitting result improves as the tilt ranges are increasing. This result suggests there has been a steady improvement in the quality of the fitting program as the maximum tilt range are increasing, and the large tilt range effectively removes the local minima.

Testing beam selection schemes

In this unit test, we test the impact of beam selection schemes. The detailed setup is
Results are presented in Table 4.10 on page 66. For beam selection with only small $g$, the diffraction patterns match very well, but the potential fields reconstructed are generally distorted, because we did not include enough beams to correctly represent the diamond cubic structure of silicon. And the thinner the specimen is, the worse the potential field are distorted. But for the beam selection scheme considering larger $g$, all silicon atom positions are revealed, with similar peak intensities. Visually comparing the fitted potential fields, it can be easily conclude that, for a successfully fitting program, we should always prefer larger $g$ reflections when the diffraction pattern intensities are weak.

**Testing thickness and target functions**

In this unit test, we test the impact of specimen thickness. The configuration is

Results are presented in Table 4.11 on page 67. When the specimen is thin (5 nm case), there is not so much details in the diffraction pattern across the disks, the fitting program easily fails. But when it is too thick (40 nm case), there occurs too much multiple diffraction, and the program cannot properly handle that situation. However, the merit function minimization performs better than R-value minimization when the specimen is thin; and for the thick specimen, R-value minimization is much better. So for a successful program, it is important to make sure the specimen thickness is moderate to make the reconstruction work.
Testing beam tilt numbers and noise factors

In this unit test, we test the beam tilt numbers and the noise impact on the optimization program. The configuration is

<table>
<thead>
<tr>
<th>Beam tilts and noise test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal: SrMoO$_4$ crystal</td>
</tr>
<tr>
<td>Beam tilt: 512, 256, 128, 64, 32, 16</td>
</tr>
<tr>
<td>Target function: $R$-value, norm function</td>
</tr>
<tr>
<td>Maximum tilt range: 60 mrad</td>
</tr>
<tr>
<td>Thickness: 30 nm</td>
</tr>
<tr>
<td>Noise: noise free, Poisson noise</td>
</tr>
<tr>
<td>Beams selected: 45 beams</td>
</tr>
</tbody>
</table>

Results are presented in Table 4.12 on page 68. As the intensities involved in this experiment are generally weak, all the diffraction patterns (target, noised and fitted) are displayed in a reverse-colored logarithm scale. Our fitting program shows good robustness against noise and limited numbers of beam tilt for this unit test. For all the noise free cases, our program can successfully reconstruct the correct structure potential field; Even with as small as 16 beam tilts and with Poisson noise, all atoms show up except the Oxygen atoms positions blur a bit. This means our program is robust against small amounts of noise. And our program can also work as expected even with only tens of beam tilts, providing the tilt angles are large enough to get rid of local minima. However, the more the beam tilts are, the more robust our program performs against noise.

Collaborative test for SrTiO$_3$

In this unit test, we test all cases for SrTiO$_3$; the detailed configuration is

<table>
<thead>
<tr>
<th>SrTiO$_3$ collaborative test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal: SrTiO$_3$ crystal</td>
</tr>
<tr>
<td>Beam tilt: 5144</td>
</tr>
<tr>
<td>Target function: $R$-value, norm function</td>
</tr>
<tr>
<td>Maximum tilt range: (60 mrad, 20 mrad, 10 mrad, 5 mrad)</td>
</tr>
<tr>
<td>Thickness: (40 nm, 20 nm, 10 nm, 5 nm)</td>
</tr>
<tr>
<td>Noise: noise free</td>
</tr>
<tr>
<td>Beams selected: 45 beams</td>
</tr>
</tbody>
</table>

Results are presented in Table 4.13 on page 69 and Table 4.14 on page 70. To further quantify the ability of our approach to determine the correct structure factor phases, three-phase invariant errors have been evaluated. The sum of the phases of three structure factors corresponding to
lattice vectors that form a closed loop, i.e.

\[ \phi_{\vec{g}, \vec{h}} = \phi_{\vec{g}} + \phi_{\vec{h}} + \phi_{-\vec{g} - \vec{h}}, \]

is independent of the choice of unit cell origin and thus a well-defined quantity. We have thus evaluated the following three-phase invariant error

\[ E_{\text{triplet}} = \frac{\sum_{\vec{g}, \vec{h}} |\phi_{\vec{g}, \vec{h}} - \phi_{\text{theory}}| \sqrt{|U_{\vec{g}} U_{\vec{h}} U_{-\vec{g} - \vec{h}}|}}{\sum_{\vec{g}, \vec{h}} \sqrt{|U_{\vec{g}} U_{\vec{h}} U_{-\vec{g} - \vec{h}}|}}, \]

for all possible structure factor phase triplets that can be constructed from the 45 structure factors of the measured reflections. Here \( \phi_{\text{theory}} \) is the theoretical phase invariant, and for centrosymmetric SrTiO\(_3\), this value is either 0 or \( \pi \). Table 4.1 lists all the average phase triplet errors for all our 16 fitting results. The errors for the patterns with larger tilt amplitudes are generally smaller than for the patterns with smaller tilt amplitudes. Setting a threshold of 0.3 we can say that reconstructed structure factor sets with a mean phase triplet error below this value correspond to a potential map which agrees reasonably with the correct one, at least when not considering very high order reflections, diffraction intensities of which have not been measured.

<table>
<thead>
<tr>
<th>40 mrad</th>
<th>20 mrad</th>
<th>10 mrad</th>
<th>5 mrad</th>
</tr>
</thead>
<tbody>
<tr>
<td>40 nm</td>
<td>0.24</td>
<td>0.36</td>
<td>0.72</td>
</tr>
<tr>
<td>20 nm</td>
<td>0.053</td>
<td>0.33</td>
<td>0.80</td>
</tr>
<tr>
<td>10 nm</td>
<td>0.16</td>
<td>0.24</td>
<td>0.39</td>
</tr>
<tr>
<td>5 nm</td>
<td>0.19</td>
<td>0.29</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Table 4.1: Mean of 1189 phase triplet errors \( E_{\text{triplet}} \) estimated from those reconstructed \( U_{\vec{g}} \) for which intensity measurements have been included in the merit function optimization, i.e. which appear in the column \( m \) of the structure factor matrix.

**Collaborative test for silicon**

In this unit test, we test all cases for silicon; the detailed configuration is

<table>
<thead>
<tr>
<th>SrTiO(_3) collaborative test</th>
<th>Crystal</th>
<th>Maximum tilt range</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sr titanate</td>
<td>(40 mrad, 20 mrad, 10 mrad, 5 mrad)</td>
<td>noise free</td>
</tr>
<tr>
<td>Beam tilt</td>
<td>3144</td>
<td>Thickness</td>
<td>Beams selected</td>
</tr>
<tr>
<td>Target function</td>
<td></td>
<td>(50 nm, 20 nm, 10 nm, 5 nm)</td>
<td>(with, without) forbidden reflections</td>
</tr>
</tbody>
</table>
The difference between Silicon and SrTiO₃ is that, there are much more forbidden reflections in the Silicon diffraction patterns than the SrTiO₃ diffraction patterns. But SrTiO₃ also produces some very weak, almost forbidden, reflections. In real word, the diffraction intensities corresponding to the forbidden reflections are always very difficult to read out precisely. To figure out the behaviour of our fitting program when including and excluding those very weak reflections, we setup two groups of numerical experiments, run our program on them and compare their performance. These beam selection schemes are shown in Table 4.8 in the middle and right column.

The beam selection for the first group is the same as our previous experiments of SrTiO₃, including a lot of forbidden reflections, where their normalized intensities can be as small as of order 10⁻²⁰, which is not measurable in reality. while the second group, which also takes 45 beams for fitting, but exclude most of the forbidden reflections, their smallest intensities is of order 10⁻⁸, but with a larger length of \( \vec{g} \). For Bravais lattice type of face centered cubic, the forbidden reflections include all reflections who’s Miller indices (h, k and l) are mixed odd and even, or all even with \( h + k + l \neq 4n \). Detailed results are presented from Table 4.15 on page 71 to Table 4.18 on page 74.

**Numeric experiment results**

Summarizing all the numerical tests presented above, we conclude the impact of all factors
Homotopy experiments

Examining Figures in Table 4.11 on page 67, we can see that neither R-value function nor merit function based optimization works throughout all thicknesses. As one works better on thin specimen, the other one works better on thick specimen, we design a convex homotopy method [WH89, Wat89, SS95, Hilo1, DO05, VUM12] and managed to make this target function works throughout all thicknesses.

The essential step is the composition of a 0-1 convex homotopy function combining the R-value function and the merit function, which is designed as follows

Convex homotopy

\[ H(\lambda, U_g) = \lambda \chi^2(U_g) + (1 - \lambda) R(U_g) \]

To take the advantage of the merit function and R-value function convergence, we design the \( \lambda \)
factor to be a function of optimization program iteration step \( k \)

\[
\lambda(k) = \begin{cases} 
0 & \text{if } \frac{k}{10} \text{ AND } 1 \\
1 & \text{otherwise}
\end{cases}
\]

where \textbf{AND} represents a logical AND operation.

The test is carried out with a configuration of

<table>
<thead>
<tr>
<th>Crystal</th>
<th>Maximum tilt range</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>SrTiO$_3$</td>
<td>40 mrad</td>
<td>noise free</td>
</tr>
<tr>
<td>Beam tilt</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{3044}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\textit{convex homotopy function}</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Minimizing this convex target function, all thickness work ranging from 5 nm to 40 nm, as are shown in Table 4.19 on page 75.

### 4.3 PERFORMANCE IN REAL WORLD

Fig. (a) of Figure 4.21 shows experimental LARBED data of SrTiO$_3$ oriented near the (001) zone axis with 1009 tilts covering a disc of maximum radius of 70 mrad. This pattern is captured by a Gatan Orius 200W CCD camera installed on a Zeiss EM912 operated at an accelerating voltage of 120 kV.

Before feeding the recorded data to our fitting program, we estimate a parameter angle \( \theta \) representing the rotation angle of the recorded disks. The original recorded LARBED pattern is demonstrated in Table 4.2 on page 62. The left one is presented in common linear scale; to better present the details in this pattern, a log-scaled image is present altogether.

The rotation angle \( \theta \) is discovered by fitting the kinematic approximation [KS03] to the measurement. For an incident beam \( \vec{k}_i \), after applying a tilt rotation, it generates a new incident beam \( \vec{k}_i' \):

\[
\begin{vmatrix} 
\vec{k}_i^{(x)} \\
\vec{k}_i^{(y)} 
\end{vmatrix} = \begin{vmatrix} 
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta 
\end{vmatrix} \begin{vmatrix} 
\vec{k}_i^{(x)} \\
\vec{k}_i^{(y)} 
\end{vmatrix}.
\]

This case, in the corresponding deduced matrix equation, the diagonal element of structure
matrix at reflection \( \hat{g} \) is changed to

\[
D_{\hat{g}} = -2\hat{k}_1^\top \hat{g} - \hat{g}^\top \hat{g}.
\]

The kinematic approximation of the scattering matrix takes a form of

\[
S_{\hat{g}, \hat{h}} = C_{\hat{g}, \hat{h}} \mathcal{U}_{\hat{g}},
\]

where the subscripts \( \hat{g} \) and \( \hat{h} \) represent the row and column index of the scattering matrix \( S \), and the complex factor \( C_{\hat{g}, \hat{h}} \) is given by

\[
C_{\hat{g}, \hat{h}} = \frac{e^{iD_{\hat{g}}} - e^{iD_{\hat{h}}}}{D_{\hat{g}} - D_{\hat{h}}},
\]

where \( t \) is the thickness of the specimen.

The target function being minimized takes a least-square form

\[
\chi^2(\mathcal{U}_{\hat{g}}, t, \theta) = \sum_{\hat{k}^{(n)}, m} \left| f_{\hat{k}^{(n)}, m}^{\text{exp}} - |S_{\hat{k}^{(n)}, m}|^2 \right|^2,
\]

but only the norm of structure factors shown in the selected column in the structure matrix \( A \) can be fitted. And the term \( S_{\hat{k}^{(n)}, m} \) represents the complex entry at beam tilt \( \hat{k}^{(n)} \) in the \( m \)th element in the first column of the scattering matrix.

Table 4.20 on page 76 lists the raw LARBED pattern, the kinematically fitted and the rotated pattern corrected with rotation angle \( \theta \).

Since no energy filtering applied, and the expected localization of much of the inelastic scattering occurs at small scattering angles, we have omitted the undiffracted and the first five weak reflections from the fit, i.e., intensities corresponding to reflections 000, 100, 010, 100 and 010 are omitted in our program.

Adding a structure factor amplitudes penalty as a constraint to prevent them from iterating to very big range, we minimized a modified R-value function.
Modified R-value

\[ R'(U_d) = \sum_{k_i} \sum_{m} \left| I_{m}^{exp} - I_{m}(U_d, k_i) \right|^2 + \alpha \sum_{g} |U_g|^2 \]

where \( \alpha = 100.9 \).

Additionally, we design another target function taking the similarity of the measured disks’ total energies into consideration. Assuming the disks of roughly same intensities indicates their corresponding structure factors are of similar moduli, we design another constraint to improve the performance of our fitting algorithm. First we classify all the structure factors corresponding to their measured intensities into different sets, \( \{ S_0, S_1, \ldots, S_{n-1} \} \), according to their integrated intensities, ensuring in each set the difference between the maximum and the minimum one is less than 10% of the maximum one. Then we append the total amount of the variances of structure factors’ moduli in each set to the modified R-value function with a factor \( \beta \).

Another modified R-value function

\[ R''(U_d) = \sum_{k_i} \sum_{m} \left| I_{m}^{exp} - I_{m}(U_d, k_i) \right|^2 + \alpha \sum_{g} |U_g|^2 + \beta \sum_{i=0}^{n-1} \text{Var}(S_i) \]

where \( \text{Var}(S_i) \) represents the variance of the moduli of the structure factors in set \( S_i \), \( \alpha = 10.09 \) and \( \beta = 10090 \).

From this experimental LARBED data for 121 reflections we have successfully determined 456 \( U_d \)'s, the specimen thickness and other parameters related to the experiment, such as a fitting of the beam tilts and the disk rotation angle. The fitted LARBED pattern simulated from the recovered structure factors is shown in Fig. 4.21 (b), the difference between it and the experimental data is shown in Fig. 4.21 (c). The specimen thickness has been found to be 11.5 nm. Fig. 4.21 (d) shows the expected projected potential corresponding to the neutral atom scattering factors for the 121 observed reflections. Fig. 4.21 (e) shows the map of the projected potential corresponding to the structure factors of the 121 measured reflections (omitting the 100 reflections). No symmetry constraints have been applied during the fit. Fig. 4.21 (f) shows the map of the projected potential obtained from all 456 structure factors that have been fitted to the experimental data, but enforcing that diffraction discs with similar total intensities agree in their structure factors’ amplitudes.
Table 4.2: Experimental data obtained.

Table 4.3: Left: SrTiO$_3$ structure. Right: silicon structure.
Table 4.4: Simulated SrTiO$_3$ patterns with different beam tilts.

Table 4.5: Simulated silicon patterns (120 keV, 30nm) with different max tilt angles.

Table 4.6: Left: simulated SrTiO$_3$ pattern without noise. Right: simulated SrTiO$_3$ pattern without Poisson noise (1,000 electrons per tilt assumed).
Table 4.7: SrTiO$_3$ patterns simulated with a maximum tilt angle 70 mrad, thickness of 11 nm (Left) and 12 nm (Middle), and their difference (Right).

Table 4.8: Beam selection strategies. Left: For SrTiO$_3$, 45 beams are selected. Middle and Right: Beam selection schemes for numerical experiments of Silicon(001). These two patterns are simulated with a high tension of 120 keV and 40 nm, and their colors are enhanced to show the selected beams clearly. Middle: same beam selection scheme as our SrTiO$_3$ experiments, including 45 beams, but only 5 discs have visible details. Right: skipping the forbidden reflections, including 45 beams that have higher intensities than the forbidden ones.
Table 4.9: Testing the impact of maximum tilt range. Upper row: target patterns with different maximum tilt ranges at 40 nm; Middle row: reconstructed patterns by minimizing R-value function; Lower row: reconstructed potential fields.
Table 4.10: Row 1: Target patterns simulated including forbidden reflections; Row 2: Reconstructed patterns from Row 1 by minimizing R-value; Row 3: Reconstructed potential fields corresponding to patterns in row 2; Row 4: Target patterns simulated excluding forbidden reflections; Row 5: Reconstructed patterns from Row 4 by minimizing R-value; Row 6: Reconstructed potential fields corresponding to patterns in row 5;
Table 4.11: Testing the impact of specimen thickness. First row: target patterns to be reconstructed from SrTiO$_3$ with a maximum tilt range of 40 mrad; Second row: LARBED patterns reconstructed by minimizing R-value function; Third row: potential fields corresponding to the patterns in the second row; Fourth row: LARBED patterns reconstructed by minimizing merit function; Last row: potential fields corresponding to the patterns in the fourth row.
Table 4.12: SrTiO$_3$ numerical experiment with different beam tilt sets and noise levels. To better present the weak details of the diffraction patterns, all intensities are visualized in logarithm scale.
<table>
<thead>
<tr>
<th>Potential field from R-value</th>
<th>Pattern reconstructed from R-value</th>
<th>Target intensity (± mrad, 5 nm)</th>
<th>Pattern reconstructed from merit function</th>
<th>Potential field from merit function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 31]</td>
<td>![Image 32]</td>
<td>![Image 33]</td>
<td>![Image 34]</td>
<td>![Image 35]</td>
</tr>
<tr>
<td>![Image 36]</td>
<td>![Image 37]</td>
<td>![Image 38]</td>
<td>![Image 39]</td>
<td>![Image 40]</td>
</tr>
<tr>
<td>![Image 41]</td>
<td>![Image 42]</td>
<td>![Image 43]</td>
<td>![Image 44]</td>
<td>![Image 45]</td>
</tr>
</tbody>
</table>

Table 4.15: Silicon refinement (1/6)
<table>
<thead>
<tr>
<th>Potential field from R-value</th>
<th>Pattern reconstructed from R-value</th>
<th>Target intensity (4 mrad, 20 nm)</th>
<th>Pattern reconstructed from merit function</th>
<th>Potential field from merit function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potential field from R-value</td>
<td>Pattern reconstructed from R-value</td>
<td>Target intensity (4 mrad, 20 nm)</td>
<td>Pattern reconstructed from merit function</td>
<td>Potential field from merit function</td>
</tr>
<tr>
<td>Potential field from R-value</td>
<td>Pattern reconstructed from R-value</td>
<td>Target intensity (4 mrad, 20 nm)</td>
<td>Pattern reconstructed from merit function</td>
<td>Potential field from merit function</td>
</tr>
<tr>
<td>Potential field from R-value</td>
<td>Pattern reconstructed from R-value</td>
<td>Target intensity (4 mrad, 20 nm)</td>
<td>Pattern reconstructed from merit function</td>
<td>Potential field from merit function</td>
</tr>
<tr>
<td>Potential field from R-value</td>
<td>Pattern reconstructed from R-value</td>
<td>Target intensity (4 mrad, 20 nm)</td>
<td>Pattern reconstructed from merit function</td>
<td>Potential field from merit function</td>
</tr>
<tr>
<td>Potential field from R-value</td>
<td>Pattern reconstructed from R-value</td>
<td>Target intensity (4 mrad, 20 nm)</td>
<td>Pattern reconstructed from merit function</td>
<td>Potential field from merit function</td>
</tr>
</tbody>
</table>

Table 4.16: Silicon refinement (2/4).

4.3 Performance in real world
Table 4.7: Silicon refinement (3/4)
Table 4.19: Testing the performance of homotopy function as target function. First column: target patterns to be reconstructed from SrTiO$_3$ with a maximum tilt range of 40 mrad; Second column: LARBED patterns reconstructed by minimizing homotopy function; Third column: potential fields corresponding to the patterns in the second column.
Table 4.20: Left: Original experimental pattern, presented with a wrong rotation angle $\theta = 0$. Middle: Kinematically fitted pattern rotated with recovered rotation angle $\theta$. Right: Rotated experimental pattern with recovered rotation angle $\theta$.

Table 4.21: (a) Experimental LARBED pattern of SrTiO$_3$ with 1009 tilts at 120 kV. (b) Reconstructed LARBED pattern, assuming no previous knowledge about the specimen. (c) Differential intensity between the experimental and the reconstructed LARBED patterns. (d) Theoretical SrTiO$_3$ (001) potential map with 121 beams. (e) Reconstructed potential map with 121 beams, assuming no previous knowledge about the specimen. (f) Reconstructed potential map with 456 beams, enforcing that diffraction discs with similar total intensities agree in their structure factors’ amplitudes.
In this thesis, the *ab initio* structure factor fitting algorithm has been presented and tested on the simulated and experimental patterns, without applying any additional constraints.

In Chapter 2, the Bloch-wave methods has been introduced, and optimization strategies to accelerate the computation in both the algorithm and hardware levels have been presented.

In Chapter 3, several nonlinear programming algorithms have been introduced, and an optimized derivative evaluation algorithm derived from Richardson Extrapolation method and Neville’s algorithm has been implemented and applied to this project.

In Chapter 4, we simulate a series of target LARBED patterns of Silicon and SrTiO$_3$ under (a) different beam selection schemes, (b) different specimen thicknesses, (c) different ranges of maximum tilt, (d) different numbers of beam tilts included and (e) noise levels, then carry out our numeric experiments on them by minimizing the merit function and the non-uniformed R-value function. Also we have invented a 0-1 homotopy target function to make our program works from 5 nm to 40 nm for SrTiO$_3$ at a maximu tilt angle of 40 mrad. From these experiments, we understand that it is important to beam selections, it is essential to enrich the diffraction pattern features across the disks and allowing moderate multiple scattering to make the fitting successful by select proper specimen thickness and tilt angle ranges. The fitting program also shows good robustness over noise and limited data set. Finally, We are able to determine 456 structure factors from experimental LARBED data of 11.5 nm thick SrTiO$_3$, even though only 121 diffraction spots have been included in the analysis.

Simulating diffraction images from given crystal structures is difficult enough, but the inversion problem of reconstructing crystal potentials from the recorded images is much more difficult. In the past, only a few people had the courage to try it, mainly because of a limitation of the computational power available at hand. The fact that we can recover unknown structure factors even without measuring them indicates that, rocking curve information and dynamical scattering in TEM must be hiding a wealth of information that is waiting to be explored.
$A$ structure matrix.
$a_i$ Bloch-wave coefficient.
$\alpha^{(k)}$ step length of an iteration optimization program at step $k$.
$\vec{a}$ a column vector of Bloch-wave coefficients.

$b_i(\vec{k}, \vec{r})$ Bloch-wave.

$c$ speed of light, $299792458 \text{ m} \cdot \text{s}^{-1}$.
$C_{g,j}$ coefficient for Bloch-wave inside a specimen.
$\chi^2$ merit function to be minimized.
$C$ matrix composed of eigenvectors of structure matrix $A$.
$\vec{C}$ an eigenvector of the structure matrix $A$.

$\Delta$ Laplacian operator.

e elementary charge, $1.6021766208(98) \times 10^{-19} \text{ C}$.
$E_0$ potential energy.
e Euler’s number, the base of the natural logarithm, $2.71828182845904523536$.
e $V_0$ kinetic energy.
e $V_s$ potential energy inside of the specimen.

$f_{ej}(\vec{H})$ electron scattering factor in the first Born approximation of the $j$th atom.

$\gamma_i$ a small scalar.
$\vec{g}$ reciprocal lattice vector.
$\vec{g}_x$ reciprocal lattice vector component along x direction.
\( \gamma_y \) reciprocal lattice vector component along \( y \) direction.
\( \gamma_z \) reciprocal lattice vector component along \( z \) direction.

\( \hbar \) Plank constant, \( 4.135667662(25) \times 10^{-15} \text{ eV} \cdot \text{s} \).
\( \hbar \) reduced Plank constant, \( 6.582119514(40) \times 10^{-16} \text{ eV} \cdot \text{s} \).
\( \hhat \) reciprocal lattice vector.

\( l \) scattering intensity.

\( \vec{k} \) wavevector.

\( |k_0| \) wavenumber of incident wave.
\( \vec{k}_j \) wavevector for Bloch-wave.
\( \vec{k}_e \) wavevector for incident wave.
\( \vec{k}_{e,z} \) wavevector component along \( z \) direction.

\( \lambda \) de Broglie wavelength.
\( \lambda_0 \) de Broglie wavelength of electron.
\( \Lambda(t) \) a diagonal matrix.

\( m \) relativistic mass of electron.
\( m_0 \) rest mass of electron.

\( \beta^{(k)} \) search direction of an iteration optimization program at step \( k \).
\( \psi \) wave function.
\( \psi_s \) Fourier coefficients of the wave function.

\( r \) position vector.
\( r_z \) position vector component along \( z \) direction.

\( S \) scattering matrix.

\( t \) specimen thickness.

\( \mathcal{U}_g \) crystal structure factor.

\( v \) speed of electron.
\( V_0 \) potential drop.
\( V^*_h \) Fourier coefficient of the electrostatic lattice potential.
\( V_s \) potential drop inside the specimen.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAS</td>
<td>Basic Linear Algebra Subprograms</td>
</tr>
<tr>
<td>CBED</td>
<td>Convergent Beam Electron Diffraction</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CUDA</td>
<td>Compute Unified Device Architecture</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphical Processing Unit</td>
</tr>
<tr>
<td>LAPACK</td>
<td>Linear Algebra PACKAGE</td>
</tr>
<tr>
<td>LARBED</td>
<td>large-Angle Rocking-Beam Electron Diffraction</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equation</td>
</tr>
<tr>
<td>PED</td>
<td>Precession Electron Diffraction</td>
</tr>
<tr>
<td>STEM</td>
<td>Scanning Transmission Electron Microscopy</td>
</tr>
<tr>
<td>TEM</td>
<td>Transmission Electron Microscopy</td>
</tr>
<tr>
<td>DRZAP</td>
<td>Double-Rocking Zone-Axis Pattern</td>
</tr>
<tr>
<td><strong>LACBED</strong></td>
<td>Large-Angle Convergent-Beam Electron Diffraction</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>SLACBED</strong></td>
<td>Simulated LACBED</td>
</tr>
</tbody>
</table>


[GRBdRH10] A. Gómez-Rodríguez, L.M. Beltrán-del Río, and R. Herrera-Becerra. Simu-


[Hig09] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revis-


B
Bloch-wave Fourier combinations, 15
Bloch-waves combination, 14

C
central differential, 43
conjugate gradient strategy, 40
conventional Schrödinger equation, 15
convex homotopy, 52, 58

D
de Broglie wavelength, 13
deduced matrix equation, 17, 59
diagonal element of structure matrix, 59
diagonal elements of structure matrix, 17
diffraction intensity, 20, 30

H
homotopy, 58

K
kinetic energy, 13

M
matrix equation, 16
matrix exponential, 21
merit function, 35, 52, 58
modified R-value function, 60
modified target function, 41

N
Neville’s formula, 44
non-normalized R-value expression, 35
numerical electron wavelength, 13

P
Padé approximation, 21
polynomial equation, 22
polynomial truncation, 22

R
reciprocal lattice vectors, 16
relativistic mass, 13
R-value function, 52, 58
S
scattering matrix, 20, 21, 30, 60
sixth-order approximation, 44
specimen potential, 15, 50
structure matrix, 60
structure factors, 16, 60

T
Taylor expansion, 43
three-ordered optimized polynomial truncation, 23

W
wave vector approximation, 17
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Title and Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christoph T. Koch</td>
<td></td>
</tr>
<tr>
<td>and Christoph T. Koch</td>
<td></td>
</tr>
</tbody>
</table>

Feng Wang

Resumé Feng Wang

Education

2012–2015 Ph.D., Ulm Universität, Ulm, Experimental Physics.
2006–2010 M.A., Purple Mountain Observatory, Graduate School of Chinese Academy of Sciences, Nanjing, China, Astrometry and Celestial Mechanics.
2000–2004 B.S., Zhejiang University, Hangzhou, China, Optical Electrical Information Engineering.

Master Thesis

title Design and implementation of two new fitting algorithms for optical readouts of a heterodyne interferometer
supervisor W.-T. Ni

Doctoral Thesis

title Inversion of Dynamical Scattering from Large-Angle Rocking-Beam Electron Diffraction Patterns
supervisor Christoph T. Koch

Publication

Parts of this dissertation have been published in the following journal articles:


and