2017

Application of Grating-Based Interferometry to Additive Manufacturing, Lithium-ion Batteries, and Crystals

Adam Joseph Brooks
Louisiana State University and Agricultural and Mechanical College, Abroo38@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

Part of the Chemistry Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_dissertations/4420

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
APPLICATION OF GRATING-BASED INTERFEROMETRY TO ADDITIVE MANUFACTURING, LITHIUM-ION BATTERIES, AND CRYSTALS

A Dissertation

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Doctor of Philosophy in The Department of Chemistry

by

Adam Joseph Brooks
B.S. in Chemistry, Syracuse University, 2012
August 2017
Acknowledgments

This dissertation would not be possible without the help of many people. First I would like to thank my advisor, Dr. Leslie Butler, for all of his knowledge and assistance over the past several years. My love for travel was made possible by his countless proposals and support in running X-ray and neutron experiments at facilities all around the world. I would also like to thank my committee members Professor Andrew Maverick, Professor Louis Haber, and Professor Kim Skinner. With countless emails and random office visits, it was a pleasure probing your thoughts and knowledge during my time here.

Dr. Jinghua Ge from the LSU Center for Computation and Technology was our visualization expert and showed me new ways to visualize tomographic volumes so I am thankful for her support. Dr. Kyungmin Ham from the LSU Center for Advanced Microstructures and Devices (CAMD) made running experiments at CAMD fast, simple, and efficient. I would like to thank Dr. Daniel Hussey from the National Institute of Standards and Technology (Gaithersburg) for his willingness to host myself and Dr. Butler on our quest for battery and additive manufacturing neutron imaging. Also, Dr. Nikolay Kardjilov from the Helmholtz Zentrum Berlin facility provided countless hours of help with design, setup, and implementation of Talbot-Lau neutron interferometry. I am also grateful for Dr. Hassina Bilheux from Oak Ridge National Laboratory for her support and graciousness in hosting myself and Dr. Butler.

For providing samples, I would like to thank Dr. Shengmin Guo and Dr. Michael Khonsari from the LSU Department of Mechanical Engineering. Their interest in fatigue of additively manufactured parts led us to the exploration of imaging
for additive manufacturing. I would also like to thank the graduate students of the Butler research group during my time here: Mutairu Olatinwo, Jumao Yuan, and Omoefe Kio. Without their help, the research presented herein would have taken much longer to understand, process, and develop. Lastly, I would like to thank our undergraduate researchers Bridget Cadigan, Gerald Knapp, and Caroline Lowery. Their assistance with programming and data processing made this research possible.
Table of Contents

Acknowledgments ................................................................. ii

List of Tables ........................................................................ vi

List of Figures ........................................................................ xiii

List of Nomenclature .............................................................. xiv

Chapter 1: Introduction .......................................................... 1

Chapter 2: Imaging ................................................................. 3
  2.1 Background ................................................................. 3
  2.2 X-ray Imaging .............................................................. 3
  2.3 Neutron Imaging ......................................................... 10
  2.4 Bragg Edge Imaging ................................................... 13
  2.5 Image Processing ....................................................... 14
  2.6 Reconstruction .......................................................... 15
  2.7 Visualization ............................................................. 15

Chapter 3: Interferometry ....................................................... 21
  3.1 Single-Shot vs Stepped-Grating Interferometry .................. 22
  3.2 Grating Fabrication ..................................................... 24
  3.3 Absorption/Attenuation ............................................... 26
  3.4 Phase Contrast .......................................................... 28
  3.5 Dark-Field ............................................................... 30
  3.6 Talbot-Lau ............................................................... 31
  3.7 Far-Field ................................................................. 33

Chapter 4: Lithium-Ion Battery Imaging ................................. 41
  4.1 Introduction .............................................................. 41
  4.2 Battery Characterization ............................................. 44
  4.3 Experimental ............................................................ 44
  4.4 Results and Discussion ............................................... 58
  4.5 Conclusions ............................................................. 81

Chapter 5: Porosity in Additive Manufacturing ....................... 88
  5.1 Introduction .............................................................. 88
  5.2 Experimental ............................................................ 90
  5.3 Ti-6Al-4V Paper ....................................................... 92
  5.4 Conclusions ............................................................. 101
Chapter 6: Fatigue in SS316 Additive Manufacturing ......................... 105
  6.1 Introduction ................................................. 105
  6.2 Experimental ................................................. 106
  6.3 Results and Discussion ....................................... 112
  6.4 Conclusions .................................................... 131

Chapter 7: Observation of Geometric Crystal Twinning and Phase ............ 137
  7.1 Introduction .................................................... 137
  7.2 Experimental .................................................... 138
  7.3 Results and Discussion ......................................... 143
  7.4 Conclusions .................................................... 148

Chapter 8: Conclusions and Future Studies ....................................... 154
  8.1 Conclusions .................................................... 154
  8.2 Future Studies ................................................ 157

Appendix A: Copyright Permission for Porosity Chapter ....................... 159
Appendix B: X-ray Linear Attenuation Coefficient Mathematica Code ........ 163
Appendix C: Mathematica Code for the TOF Neutron Diffraction Experiment
  at SNS VULCAN .................................................... 179
Appendix D: Neutron Far-field Interferometry Mathematica Code for Experi-
  ments at NIST .................................................... 213
Appendix E: TomoPy Code for Neutron Talbot-Lau Interferometry Experi-
  ments at HZB ..................................................... 231
Appendix F: Neutron Bragg Edge Mathematica Code for Experiments at HZB

Vita ................................................................. 267
### List of Tables

2.1 CAMD Synchrotron Storage Ring Parameters ........................................ 10

6.1 SS316 (CL 20ES) Elemental Composition and Corresponding Neutron Attenuation ................................................................. 108

6.2 AM and Conventional Samples and Experiments ................................. 115
List of Figures

2.1 Sketch of a microfocus X-ray tube ........................................ 7

3.1 Example of a grating interferometry setup .................................... 23

3.2 Gadox grating fabrication process ........................................... 27

3.3 Reference transmission, visibility, and phi plots for open beam images taken from CAMD in April, 2017 .................................................. 28

3.4 Sketch of refraction ............................................................ 29

3.5 Elastic and inelastic scattering of the vector $Q$ ............................ 30

3.6 Moiré fringes generated by the Talbot effect ............................... 32

4.1 Layering within a Li-ion battery .............................................. 42

4.2 TOF spectra for the VULCAN beamline ................................. 46

4.3 Neutron diffraction of $\text{CeO}_2$ at the VULCAN beamline .............. 47

4.4 Sample neutron diffraction plots based off state of charge for copper and $\text{LiCoO}_2$ ................................................................. 49

4.5 Interpolation functions generated in Mathematica for $\text{LiCoO}_2$, $\text{Li}_{0.75}\text{CoO}_2$, $\text{Li}_{0.5}\text{CoO}_2$, $\text{LiC}_6$, $\text{LiC}_{12}$, and $\text{C}_6$ .......................................................... 50

4.6 Simulated battery dataset with Gaussian noise .......................... 50

4.7 Setup of the near-field interferometer at CAMD .......................... 52
4.8 Autocorrelation length calculations for March 2017 experiments at NIST 53

4.9 Li-ion and Li-po batteries glued to an aluminum sheet for imaging at HZB 54

4.10 Two 43 mAh Li-po batteries for Bragg edge imaging at HZB .............. 55

4.11 Setup for far-field interferometry experiments for a worn (left), fresh (middle), and slightly worn (right) 43 mAh battery at NIST ............... 56

4.12 Autocorrelation length calculations for June 2016 experiments at NIST . 57

4.13 Neutron diffraction for a fresh Li-ion battery ......................... 59

4.14 Neutron diffraction for a worn, 25% of original SOC, Li-ion battery ... 60

4.15 Reduced spectral information of 7 component battery system using MCR-ALS .............................................................. 60

4.16 Reduced concentration information of a 7 component battery system using MCR-ALS .......................................................... 61

4.17 X-ray absorption projections for a charged and discharged 43 mAh battery at 0 degrees ...................................................... 62

4.18 Dark-field projections for a charged and discharged 43 mAh battery at 0 degrees ................................................................. 62

4.19 X-ray dark-field image of scattering when ξ = 20 nm ................. 63

4.20 X-ray dark-field image of scattering when ξ = 133 nm ............... 64

4.21 X-ray dark-field images for a fresh 43 mAh battery when charged and discharged at ξ = 496 nm ........................................... 64

4.22 X-ray dark-field images for a worn 43 mAh battery when charged and discharged at ξ = 496 nm ........................................... 64
4.23 Comparison of dark-field signal for fresh and worn 43 mAh batteries when 
\( \xi = 496 \text{ nm} \) ................................................................. 65

4.24 Image of fresh and worn 43 mAh batteries at 3.18 Å .......................... 66

4.25 Bragg edge graph from 3 - 4.4 Å of a worn 43 mAh battery .............. 67

4.26 Bragg edge graph from 3 - 4.4 Å of a fresh 43 mAh battery .............. 68

4.27 Dark-field images of charged and discharged 15 mAh Li-po batteries .... 68

4.28 Neutron scattering images of charged and discharged 90 mAh batteries 
when \( \xi = 1.125 \mu \text{m} \) ................................................................. 70

4.29 Neutron scattering images of charged and discharged 90 mAh batteries 
when \( \xi = 2.25 \mu \text{m} \) ................................................................. 70

4.30 Neutron scattering images of charged and discharged 90 mAh batteries 
when \( \xi = 2.813 \mu \text{m} \) ................................................................. 71

4.31 Mean scattering values across the fresh and worn, charged and discharged 
90 mAh Li-po batteries .......................................................... 72

4.32 Neutron scattering images after the first charge and discharge cycles of 
43 mAh batteries when \( \xi = 788 \text{ nm} \) .............................................. 72

4.33 Neutron scattering images after the first charge and discharge cycles of 
43 mAh batteries when \( \xi = 1.125 \mu \text{m} \) ............................................. 73

4.34 Neutron scattering images after the first charge and discharge cycles of 
43 mAh batteries when \( \xi = 2.363 \mu \text{m} \) ............................................. 74

4.35 Neutron scattering images after the first charge and discharge cycles of 
43 mAh batteries when \( \xi = 2.588 \mu \text{m} \) ............................................. 74

4.36 Neutron scattering images after the first charge and discharge cycles of 
43 mAh batteries when \( \xi = 2.925 \mu \text{m} \) ............................................. 75
4.37 Mean scattering values across the charged 43 mAh Li-po batteries ...... 75

4.38 Mean scattering values across the discharged 43 mAh Li-po batteries ... 76

4.39 Neutron scattering images after the first charge and discharge cycles of 15 mAh batteries when $\xi = 1.125 \mu m$. ................................. 77

4.40 Neutron scattering images after the first charge and discharge cycles of 15 mAh batteries when $\xi = 2.25 \mu m$. ................................. 77

4.41 Neutron scattering images after the first charge and discharge cycles of 15 mAh batteries when $\xi = 2.8125 \mu m$. ................................. 78

4.42 Neutron scattering images after the first charge and discharge cycles of 15 mAh batteries when $\xi = 3.375 \mu m$. ................................. 78

4.43 Mean scattering values as a function of autocorrelation length for the fresh and worn 15 mAh batteries ........................................ 79

4.44 Neutron scattering images of 50 mAh batteries when $\xi = 1.35 \mu m$ .... 80

4.45 Neutron scattering images of 50 mAh batteries when $\xi = 2.25 \mu m$ .... 80

4.46 Neutron scattering images of 50 mAh batteries when $\xi = 2.813 \mu m$ .... 81

4.47 Mean scattering values across the fresh and worn, charged and discharged 50 mAh Li-po batteries ........................................ 81

5.1 Flow chart showing the many printing methods possible. Of focus in this work are Electron Beam Melting and Selective Laser Melting............. 89

5.2 Image of the 1.5 cm$^3$ Ti-6Al-4V EBM cubes printed at ORNL ........... 91

5.3 Visibility calculation of the neutron beam at HZB .......................... 91

6.1 Image of fractured, pristine, and half-life 2 mm thick SS316 AM samples 107
6.2 Image of pristine AM and conventional 3 mm thick SS316 samples...... 108
6.3 Autocorrelation length calculations for January 2017 experiments at NIST111
6.4 Transmission images for a thin SS316 AM sample before bending ...... 113
6.5 Transmission images for a thin SS316 AM sample after bending 200 times with pliers......................................................... 114
6.6 Dark-field image of a pristine, 3 mm thick additive manufactured dogbone of SS316 at z =0.4 m (ξ = 606 nm) showing six regions of interest ...... 116
6.7 Graph of the DF signal versus region in the 3 mm AM pristine sample... 116
6.8 Dark-field images of the 3 mm SS316 AM pristine and fractured samples at scattering lengths of 600 nm and 960 nm............................... 117
6.9 Dark-field images of the 3 mm SS316 AM pristine and fractured samples at scattering lengths of 1.31 μm, 1.66 μm, and 2 μm......................... 118
6.10 Dark-field image of a 3 mm half-life additive manufactured dogbone of SS316 at z =0.4 m (ξ = 606 nm) ............................................. 119
6.11 Graph of the DF signal versus region in the 3 mm AM half-life sample when ξ = 606 nm .............................................................. 119
6.12 Dark-field images of the SS316 AM half-life sample at scattering length values of 600 nm, 720 nm, and 840 nm................................. 120
6.13 Dark-field images of the SS316 AM half-life sample at scattering length values of 960 nm, 1.07 μm, and 2 μm................................. 122
6.14 Plot of darkfield signal versus autocorrelation length for the pristine, half-life, and fractured 3 mm samples ........................................... 123
6.15 Vertical orientation of the 3 mm AM half-life sample at ξ of 600 nm and 2.0 μm................................................................. 124
6.16 Dark-field volume of the 2 mm AM samples showing the pristine sample at the top, the half-life sample in the middle, and the fractured sample at the bottom.......................... 125

6.17 Line probes of the half-life and pristine AM 2 mm samples............ 126

6.18 Front view of the conventionally machined 3 mm SS316 fatigued and half-life samples................................................................. 127

6.19 Side view of the conventionally machined 3 mm SS316 half-life sample . 127

6.20 Bragg edge image generated from 3.8 - 4.1 Å divided by 4.3 - 4.5 Å...... 128

6.21 Bragg edge image generated from 2.2 - 2.6 Å divided by 3.8 - 4.1 Å..... 128

6.22 Bragg edge image of the 3 mm AM sample generated from the mean (3.8 Å to 4.1 Å) divided by mean (4.3 Å to 4.5 Å)........................... 129

6.23 Bragg edge comparison of the conventional and AM 3 mm printed samples130

6.24 Bragg edge spectra for the pristine, half-life and fractured 2 mm pulled samples ................................................................. 130

7.1 Titanite and strontium ruthenate samples imaged at HZB and NIST..... 139

7.2 An uncorrelated data set in the Blomqvist β test ...................... 141

7.3 A partially correlated positive data set in the Blomqvist β test ......... 142

7.4 A partially correlated negative data set in the Blomqvist β test ......... 142

7.5 Two slightly different rotation angles for the titanite sample at ......... 144

7.6 Statistical analysis for two slightly different rotation angles of titanite... 145
7.7 Titanite projections from both Talbot-Lau interferometry at HZB and far-field interferometry at NIST are assessed with the Blomqvist-\(\beta\) statistic as a function of tomography rotation angle ........................................ 146

7.8 Two rotation angles for the \(\text{Sr}_3(\text{Ru}_{0.95}\text{Mn}_{0.05})_2\text{O}_7\) sample ......................... 147

7.9 Statistical analysis for two rotation angles of the \(\text{Sr}_3(\text{Ru}_{0.95}\text{Mn}_{0.05})_2\text{O}_7\) sample at 133.3\(^\circ\) and 127.3\(^\circ\) ...................................................... 149
List of Nomenclature

ABS - Absorption
AM - Additive Manufacturing
CAD - Computer-Aided Design
CAMD - Center for Advanced Microstructures and Devices
CCD - Charged Couple Device
CLS - Compact Light Source
CMOS - Complementary Metal-Oxide Semiconductor
CPU - Central Processing Unit
CT - Computed Tomography
DF - Darkfield
DPC - Differential Phase Contrast
FIB - Focused Ion Beam
FBP - Filtered Back Projection
Gadox - Gadolinium Oxysulfide
GPU - Graphics Processor Unit
GUI - Graphical User Interface
HZB - Helmholtz-Zentrum Berlin
ICSD - Inorganic Crystal Structure Database
LANSCE - Los Alamos Neutron Science Center
LIB - Lithium-ion Batteries
LIGA - Lithographie, Galvaniformung and Abformung
LIGO - Laser Interferometer Gravitational-Wave Observatory
Li-po - Lithium-polymer
MCP - Microchannel Plate
NIH - National Institutes of Health
NIST - National Institute for Standards and Technology
ORNL - Oak Ridge National Laboratory
PSI - Paul Scherrer Institut
SEI - Solid Electrolyte Interface
SART - Simultaneous Algebraic Reconstruction Technique
SEM - Scanning Electron Microscope
SINQ - Swiss Spallation Neutron Source
SIRT - Simultaneous Iterative Reconstruction Technique
SOC - State of Charge
SS316 - Stainless Steel 316
Ti6Al4V - Titanium Aluminum Vanadium
TOF - Time-of-flight
TRIGA - Training, Research, Isotopes, General Atomics
µXCT - Micro X-ray Computed Tomography
Abstract

X-ray and neutron imaging are convenient ways to non-destructively observe novel materials. X-rays provide advantages of low cost and high brilliance while neutrons show bulk and isotopic sensitivity. Imaging provides a way for observing chemical and physical properties of materials without the need for destruction. The way of the imaging future is utilizing imaging with grating-based interferometry. In comparison to traditional radiography and tomography, by using absorption and phase gratings in the beam path, the absorption, phase, and scattering of a sample can be detected. In essence, three image datasets can be obtained in one experiment, saving substantially on costs (especially at expensive neutron facilities), time and materials. With several methods of interferometry available, the focus in this work is Talbot-Lau interferometry and newer designs referred to as near-field and far-field interferometry.

X-ray Talbot-Lau interferometry experiments were performed at the LSU synchrotron, Center for Advanced Microstructures and Devices (CAMD), using a microfocus X-ray tube and synchrotron X-rays (38 keV). Neutron Talbot-Lau experiments were performed at the CONRAD2 beamline (HZB, Berlin, Germany) and far-field experiments at the NG6 beamline (NIST, Gaithersburg, USA). Neutron imaging of the additive manufactured samples revealed pore structures and evidence of fracture as a function of fatigue. Battery imaging shows the migration of lithium across battery layers on a visual and quantitative level. X-ray and neutron imaging of potentially twinned crystals revealed the importance of preserving data in the 2D projection images that was lost in volume reconstruction. A comparison of Talbot-Lau, near-field, and far-field interferometry with application to additively
manufactured samples, lithium-ion batteries, and geometrically twinned crystals
is presented.
Chapter 1
Introduction

The field of materials science continues to grow at a rapid pace, with novel materials enhancing previous materials’ physical and mechanical properties. From batteries to crystals to 3D printing, new methods are needed to characterize and understand the mechanisms of modern materials in a non-destructive manner.

An easy solution to this problem is through imaging. Imaging provides a way to view the inner workings of materials while keeping the sample intact and available for further characterization. X-rays and neutrons are both highly respected imaging methods, with X-rays advantages including highly coherent beams, low cost, and high penetration for low atomic Z elements. Neutron advantages include obtaining bulk properties, isotopic sensitivity, and wavelengths on the order of inter-atomic spacing (Angstrom size). The complementary nature of X-rays and neutron imaging can result in valuable datasets on both the 2D and 3D scale.

One benefit of imaging is the non-destructive capability of X-rays and neutrons. The field of material science is actively looking for non-destructive testing opportunities. In the case of additive manufacturing, defects within the printed samples like porosity or cracks are challenging to notice on the micrometer scale length. In-situ X-ray and neutron imaging during fatigue and stress procedures would greatly advance knowledge of how the crystalline or amorphous structures change as a function of wear. By obtaining chemical and physical properties from imaging, samples can be better utilized under a series of experiments rather than a create and destroy methodology.
As for a specific imaging method, of recent interest is grating-based interferometry. Grating-based interferometry relies on the use of absorption gratings to add coherence to the beam. Once the beam is coherent, a change in phase of the wavefront can be altered using a phase grating. The change in phase of the wavefront is propagated down the beam’s path and can be detected using an analyzer grating. When a sample is introduced into the system, alterations to the wavefront are analyzed through transmission (neutron) or absorption (X-ray) images, differential phase images, and small angle scattering (dark-field) images.

The goal of this work is to utilize X-ray and neutron interferometry with lithium-ion batteries, additively manufactured metallic samples, and geometrically or synthetically twinned single crystals. With the combination of imaging with visualization, it is possible to observe defects in these samples.
Chapter 2
Imaging

2.1 Background

X-ray and neutron radiography and tomography has advanced greatly over the last century, from new instrument setups to a wide range of applications in the fields of biomedical research and materials science. While X-rays have an advantage of short wavelengths, cost, and high flux, neutrons reign supreme in the case of penetration depth and isotopic sensitivity. When trying to determine whether x-rays or neutrons are the better imaging method for a sample, the focus turns to the chemical or physical properties of interest. To observe lithium movement in Li-ion batteries, X-ray and neutron radiography can observe volume expansion of Li or Si particles during the charging/discharging process [1, 2]. For the case of single crystal imaging, energy-resolved neutron imaging found single-crystalline bulk properties in gold crystals that were too opaque for traditional characterization methods [3]. In our search for signs of geometric and synthetic crystal twinning, neutron imaging appears as a practical option.

2.2 X-ray Imaging

The discovery of X-rays by Roentgen in 1895 truly revolutionized the field of imaging. Roentgen first observed fluorescence from barium platino-cyanide on black paper at a distance of 2 meters, indicating that an opaque agent was capable of penetrating the paper [4]. From this initial finding, Roentgen explored various metals (aluminum, copper, silver) and solutions (water) to determine that the density and thickness of materials are the main property affecting permeability [4]. Roentgen also found that while the retina was unable to observe the new rays, an image of his wife’s hand was possible. Although the risks of radiation exposure
were unknown back then, Roentgen and his wife’s hand showed the world the potential of X-rays for imaging.

In years since, the biomedical field has utilized X-rays for observing dental cavities to computed tomography (CT) scans of the human body. When a person ventures through an airport, they and their baggage are subjected to X-rays in an effort to detect liquid explosives [5]. With low exposures and dosage, X-rays are proven to be a safe imaging method.

X-ray radiography (2D) and tomography (3D) provide ways of viewing the internal features of a sample without manipulation. Of recent X-ray interest is the polymer electrolyte membrane (PEM) in fuel cells, where the dynamics of liquid water transport can be visualized at spatial resolutions below 10 μm [6, 7].

X-rays can be thought of as electromagnetic waves in the wavelength range of 0.01 - 10 nm. X-rays with high energies (above 10 keV, roughly 0.12 nm and below) are referred to as hard X-rays while X-rays below 10 keV are referred to as soft X-rays. To calculate the wavelength of X-rays based upon a given photon energy, Eq. 2.1 is used

\[ E = h\nu = hc/\lambda \]  

(2.1)

where E is the energy in Joules, h is Planck’s constant of 6.626 x 10^{-34} J-s, \( \nu \) is frequency in Hz, and c is the speed of light of 2.998 x 10^8 m/s. One challenge with X-ray energy calculations is that for most experimental setups, X-ray energy is represented in eV or keV. An easy solution to this problem is through the use of Eq. 2.2

\[ \lambda[\text{Å}] = \frac{12.3984}{E[\text{keV}]} \]  

(2.2)
With X-ray contrast generated by a difference in attenuation between elements, weakly absorbing materials are generally not imaged well with X-rays. This is due to the X-ray absorption coefficient, \( \mu \), being proportional to the atomic number \( Z \) (roughly the fourth power of \( Z \)) [8]. For low \( Z \) elements like carbon, this implies that better contrast via staining or media is valuable for soft-tissue imaging.

Transmission of X-rays through a sample is based on the wave nature of light with differences between the refractive index, \( n \), and thickness of samples, \( t \). Eq. 2.3 shows how the refractive index is dependent upon the real and imaginary parts of a sample’s behavior.

\[
n = 1 - \delta - i\beta
\]  
\[(2.3)\]

The real part is represented by \( \delta \) while the imaginary part is represented by \( \beta \).

\[
\delta = \frac{r_e \lambda^2}{2\pi} N_k(Z_k + f'_k)
\]  
\[(2.4)\]

\[
\beta = \frac{\lambda(\mu/\rho)\rho}{4\pi}
\]  
\[(2.5)\]

\( r_e \) is the classical electron radius \((2.82 \times 10^{-15} \text{ m})\), \( \lambda \) is the wavelength of the X-ray beam, \( \mu/\rho \) the mass attenuation coefficient, \( N_k \) is the number density, \( Z_k \) is the number of electrons, and \( f'_k \) is the atomic form factor. \( f'_k \) is a representation of the scattering signal of a wave generated from an isolated atom. The values for \( f'_k \) have previously been determined by Hubbell [9] and are available freely on the National Institute for Standards and Technology’s website \(^1\).

Number density is calculated by using the density of material, \( \rho \), Avogadro’s constant, \( N_A \), and molar mass, \( M \).

\(^1\)http://www.nist.gov/pml/data/ffast/
Mass attenuation coefficients (without coherent scattering) for elements are imported as the value of $(\mu/\rho)$ at certain energies. The variable $\mu$ describes the amount of beam that is absorbed/scattered through a certain thickness of material. This is calculated separately by multiplying the atomic density by $(\mu/\rho)$.

$$N_k = \frac{\text{density}N_A}{M}$$

$$\mu = \frac{\mu}{\rho}d$$

With $\mu$ known, the X-ray transmission, $T$, of compounds can be calculated based upon different thicknesses, $t$.

$$T = e^{-\mu t}$$

X-rays can be produced from several different methods. This study focuses on two of those methods: X-ray tubes and synchrotron radiation sources.

### 2.2.1 X-ray Tubes

Many different types of X-ray tubes have been developed in the past. Coolidge X-ray tubes utilized a fixed tungsten filament, crystal, and thin metal foil such that electrons were generated from heating the tungsten filament (cathode material) and accelerated towards a tungsten target (anode material) due to a difference in voltage potential. Once the electrons are accelerated and hit the anode, they are scattered in different directions and generate bremsstrahlung and characteristic X-rays. The rotating anode gas X-ray tube used metal copper targets and water-cooled aluminum cathode material to provide an intense beam of X-rays with a finer focus and brighter focal spot than stationary anode tubes.

---

Figure 2.1. Sketch of the setup for a microfocus X-ray tube. Electrons generated from heating a cathode material are accelerated towards a target due to a difference in electrochemical potentials. When the electrons reach the target, they are scattered in different directions creating $\beta$-remstrahlung X-rays [12].

Of interest here is the microfocus X-ray tube. Microfocus X-ray tubes have similar features to the Coolidge and rotating anode tubes (cathode filaments, anode material, and targets), however they also include an electromagnetic lens as shown in Fig 2.1 [12]. Microfocus X-ray tubes operate under vacuum by hitting a target material with electrons. In most cases, the filament is a piece of tungsten wire due to tungsten’s high melting point. Properties of the electron beam such as spot size and stability are determined by the interactive size and trajectory of the electron beam with the target material [12].

2.2.2 Synchrotron Sources

Cyclic electron storage rings have been around since 1946, where electrons in a 100 MeV electron accelerator were subjected to a radial acceleration of $10^{17} \text{ m/s}^2$ [13]. In order to increase the energy level of the synchrotron, the speed of the electrons needs to be sufficient to account for energy lost through radiation. Synchrotron radiation sources provide highly brilliant X-rays at a low cost [13]. The storage rings have been through several generations of upgrades, usually referred to
from zeroth to fourth generation sources. Zeroth and first generation synchrotrons were used for high energy physics, with the main difference being that the first generation rings stored electrons for long periods of time, hours [14].

The X-ray radiation produced from synchrotron radiation is over a broad spectral range in comparison to X-ray tubes, which operate in a narrow bandwidth with a cone beam geometry [15]. The parallel beam nature of synchrotron radiation ensures high brightness, polarization, and pulsed time stamps to determine the energy of the beam at specific time points [15]. Recently developed synchrotron sources called compact light sources (CLS) act similarly to a synchrotron source with tunable X-ray energies and a large field of view while only taking up a few meters of space [16]. Bending magnets are used to propel the electrons in the storage ring in a circular path. Despite these advantages, operational costs of synchrotrons remain higher than tube sources due to the large amount of necessary shielding and cost to fix the magnets.

To generate higher energy radiation, insertion devices can be used in the storage rings. Insertion devices operate in straight sections of the storage ring as linear accelerators and produce periodic oscillations of the electron beam through the use of alternating polar magnets [17, 18]. When the electrons pass through the magnets, radiation sources are generated at multiple points due to the oscillations from the magnets. Since the flux of the radiation is dependent on the speed of the electrons, having faster electrons pass through insertion devices implies more radiation [19].

The two main types of insertion devices are undulators and wigglers. The main difference between wigglers and undulators is the K-factor, which is a constant depending on the charge of the particles, q, the magnetic field, B, the period of
the insertion device, \( \lambda_u \), the speed of the particle, \( \beta \), the mass of the particle, \( m \), and the speed of light, \( c \).

\[
K = \frac{qB\lambda_u}{2\pi\beta mc}
\]  

(2.9)

The K-factor determines the amount of radiation that can be generated from an insertion device. In the case of wigglers, K is much greater than 1 while in undulators K is less than 1. The easiest way to alter K is by changing the magnetic field of the wiggler or undulator via the distance between magnets.

The radiation produced from undulators depends on the magnetic period length, strength of the magnetic field (radiation intensity is roughly the square of the magnetic field strength) and electron energy [18, 20]. Slotted steel bars cover the faces of 25 magnets with teeth filling in the gaps, where the magnets are altered North/South and South/North to reduce horizontal radiation [20]. Electrons pass through an evacuated flight tube with a period of 40 mm [20].

The difference between wigglers and undulators is that wigglers have smaller spacings in-between the magnets, leading to a higher number of periods [17]. Wigglers also provide a large fan of radiation, meaning that multiple beamlines can share the radiation generated from one pole of the magnet [15]. In the case of the Synchrotron Radiation Resource Center, a 25-pole, 1.8 Tesla wiggler with a 1.3 GeV storage ring split 13 mrad of radiation into three beamlines to produce photons at 6, 8, 10, and 12 keV [21]. With many options for the number and periods of magnets, wigglers have many ways to tweak their performance to become suitable for imaging.

The LSU synchrotron, the Center for Advanced Microstructures and Devices (CAMD), operates a 1.5 GeV electron storage ring and surprisingly is the only
state-funded synchrotron in the United States \(^3\). Properties of the synchrotron include a 200 MeV low-energy-injected Chasman-Green lattice, eight bending magnets located around the ring structure (radius of 2.928 m), and two straight sections for undulators or wigglers (one 7 T superconducting wiggler is currently installed). Parameters for CAMD can be seen in Table 2.1.

**2.3 Neutron Imaging**

Neutron imaging has provided many advances in the fields of materials science, physics, and scattering. Neutron penetration through a 10 Hz cylindrical steam engine showed water vapor condensation after normalization of images [22]. While neutrons may be more expensive than X-rays, they possess higher penetration power, meaning it is possible to obtain bulk properties of materials. Neutrons have an isotopic sensitivity and magnetic moment, opening the possibility for observing unlikely isotopes and polarized neutron experiments.

Neutron imaging also has a wide range of wavelengths possible, from hot to thermal to cold neutron. Hot neutrons possess wavelengths from 0.4 Å - 1 Å, which are quite fast for neutron imaging. Thermal neutrons are in a more appropriate

---

\(^3\)CAMD; http://www.lsu.edu/camd/about/index.php
range for imaging and have a wavelength range of 1 Å - 3 Å, where they are in equilibrium with the cooling water around the reactor core. Cold neutrons are on the wavelength of 2.5 Å - 20 Å as they pass through liquid hydrogen or deuterium at 20 K. Imaging experiments can be optimized for cold neutrons over thermal neutrons due to the larger cross-section for materials at lower energy [23]. This means that the samples utilized need to be small since the attenuation coefficients are higher than when using thermal neutrons.

2.3.1 Reactors

There are several types of nuclear reactors, from small Training, Research, Isotopes, General Atomics (TRIGA) reactors to fast pulse and high flux reactors [24]. Nuclear research reactors tend to use highly enriched uranium ($^{235}U$) for source brightness, i.e., number of neutrons per unit area per time. When the neutrons continue to interact with other $^{235}U$ nuclei, more neutrons and heat are generated until the fission process becomes self-sustaining. Two fission reactions for $^{235}U$ are shown in Eq. 2.10.

$$^{235}U + ^1_0 n \rightarrow ^{140}_{56} Ba + ^{93}_{36} Kr + 3 ^1_0 n \quad (2.10)$$

$$^{235}U + ^1_0 n \rightarrow ^{144}_{54} Xe + ^{90}_{38} Sr + 2 ^1_0 n \quad (2.11)$$

Many neutron reactors utilize graphite, light and heavy water as moderators, cooling agents, and radiation shields to slow fast neutrons down to speeds where fission can occur [25, 26]. The high flux isotope reactor (HFIR) at Oak Ridge National Laboratory (ORNL) operates at 85 MW and uses highly enriched $^{235}U$ as the fuel element, a 0.30 m thick beryllium reflector, and is light-water-cooled 4.

4http://neutrons.ornl.gov/hfir
2.3.2 Spallation Sources

Spallation neutron sources are anticipated to be the way of the future for neutron imaging. With reactor sources shutting down such as Helmholtz-Zentrum Berlin (HZB) in 2020 \(^5\), spallation sources are becoming the main supply for neutron imaging. There are several spallation sources already around the world, including the Swiss Spallation Neutron Source (SINQ) at the Paul Scherrer Institute (PSI), Switzerland, the Los Alamos Neutron Science Center (LANSCE) in Los Alamos, New Mexico, and the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL), Oak Ridge, Tennessee. Spallation sources operate by having protons shot at a target material made of heavy atomic nuclei. In the case of SNS, the proton beam has an energy of 1 GeV with 2 mA current and the target is made of flowing mercury [27]. When the proton hits the nuclei, neutrons and particles are loosened such that they continue interacting with other nuclei, creating more neutrons.

Spallation sources operate in a time-of-flight (TOF) mode, where neutrons are generated in pulses to each beamline. TOF instruments operate based upon inelastic neutron scattering, where scattering of the neutron beam from a sample can be detected. With spallation neutron imaging, there are several wavelengths possible for each pulse. Since the pulses operate over a wide range of wavelengths, a chopper can be used to select short, near-monochromatic pulses from the neutron beam. To calculate the energy of the scattered neutrons, information regarding the scattering angle and time stamp of a sample’s interactions with the beam are required [28]. By including multiple choppers on a beamline, the energy resolution of the beam can result in a narrow beam, along the order of 15 mm [28]. In the case of

\(^5\)https://www.helmholtz-berlin.de
material characterization, this becomes a great advantage as monochromatic beam can determine scattering parameters as a function of wavelength.

There are several types of neutron detector combinations currently used at neutron imaging beamlines, including Microchannel Plates (MCPs), Charge-Couple Devices (CCDs), and complementary metal-oxide semiconductor (CMOS) sensors with scintillator screens [29, 30, 31]. Advantages of these combinations include spatial resolution down to 10 μm from magnification [32]. Some of these devices operate with $^3$He under the reaction shown in Eq. 2.12.

\[ ^3He + n \Rightarrow ^3H + ^1H + 0.764 \text{ MeV} \]  

(2.12)

However, due to cost and a shortage of $^3$He, other $^3$He-free detectors using Al$_2$O$_3$ lamellas coated with 1 μm $^{10}$B enriched B$_4$C have been proven as viable alternatives [33].

2.4 Bragg Edge Imaging

Bragg edge imaging provides one way to understand the crystallographic information within a sample. Using a monochromator, a series of images are collected at differing wavelengths to determine if a phase change occurs within the sample. In the case of neutrons, cold neutrons have similar wavelengths as the $d_{hkl}$ dimensions of crystalline compounds [34]. If a change of phase occurs, the transmission signal increases greatly once the plane is no longer probed. This is based upon Bragg’s law,

\[ n \lambda = 2d_{hkl} \sin \theta \]  

(2.13)

where neutrons are scattered coherently at angles of $2\theta$ from lattice planes at certain lattice distances, $d_{hkl}$. Cold neutrons are optimal for Bragg edge imaging since materials tend to have intense Bragg edges within this region [23].
2.5 Image Processing

Image processing can be the trickiest part of an experiment as it becomes challenging to determine if features observed in images are real or a characteristic of the beam, camera, detector, or setup. The following describes how images were processed for experiments in Chapters 4, 5, and 6, however similar image processing is performed for experiments around the world.

Raw images collected from experiments can be in several different formats: tiff, jpg, or FITS. Tiff files are generally easy to import and process using a wide range of programs. The size of the images depends upon the pixels of the camera. Binning can occur within the camera (typically 2 x 2 or 4 x 4) or after images are collected during processing. Once the raw images are imported, cropping can occur. Cropping is only needed if image quality is poor from issues such as defects in scintillator screens, optical parts, or beam divergence.

The sample images are processed with the reference images or “white” beam images. Sample images are divided by the reference images to get signal for each transmission/absorption, differential phase contrast (DPC), and dark-field (DF) image. At this point, projections are made for each angle, usually in the FITS format.

During tomography experiments, if the center of rotation is slightly misaligned, the 0 and 180° projections will be off by several pixels. A shift can be applied to the projections such that known points of a sample (such as corners of a cube) can be set to the same location, thus leading to proper registration of all the projections. Once this shift is applied to the absorption, DPC and DF projections, sinograms, also referred to as interferograms, can be generated. Sinograms represent absorption, DPC and dark-field intensity as a ray is propagated through a
sample at different angles. The information embedded in the sinograms is used to help reconstruct an object.

2.6 Reconstruction

Once the sinograms are generated, there are several different methods for reconstructing X-ray and neutron data. The filtered back projection (FBP) method, the Radon transformation, is one of the most common algorithms used. In the case of X-ray biomedical imaging, reducing the number of projections needed with X-rays was found as feasible with FBP [35]. However, one issue with FBP is that it can result in noisy datasets due to amplification of high-frequency signals. Reconstruction software such as SNARK09 [36], Octopus [37], MATLAB, and the ASTRA toolbox [38] have a variety of Radon algorithms including simultaneous iterative reconstruction technique (SIRT) and simultaneous algebraic reconstruction technique (SART) [39]. The time needed to reconstruct large datasets can be long on a single central processing unit (CPU) so the use of graphics processor units (GPUs) is suggested to help speed up reconstruction [40]. A free, open-source python based software called TomoPy was recently discovered and shows great promise in combining reconstruction with functional programming [41, 42].

2.7 Visualization

Visualization plays a key role in imaging research. Several issues can arise in the determination of whether a feature is truly present or if it is a characteristic of the beam. Visualization software can help manipulate the data such that unexpected consequences of reconstruction like ring artifacts can be easily removed. The Avizo™ software has useful features for materials science imaging with skeletonization, image segmentation, and registration 6. Another Java-based software

---

6https://www.fei.com/software/avizo-for-materials-science/
program is ImageJ, developed at the National Institutes of Health (NIH)\(^7\). ImageJ enables users to import large files (on the order of 10 GB) of many different formats (tiff, jpeg, raw, HDF5) while extracting useful statistics from the images or volumes. These visualization programs are used to view Li-ion/Li-polymer batteries, additively manufactured samples, and crystals in the following chapters.

\(^7\text{https://imagej.nih.gov/ij/}\)
References


Chapter 3
Interferometry

Back in 1887, Albert Michelson and Edward Morley disproved the concept that light must have a supporting medium (referred to as ether/aether) to transmit its wave motions, paving the path to the foundation of the special theory of relativity [1]. To set up the experiment, Michelson used a beamsplitter to split light into two paths, where each path is reflected back toward the beamsplitter using mirrors and the amplitude of the beams are recombined [1]. A slight difference in the angles between the beams can result in a sinusoidal fringe pattern, making it possible to detect intensity changes as a function of angle. While this research was performed over a century ago, the Michelson interferometer design was recently utilized in the Laser Interferometer Gravitational-Wave Observatory (LIGO) in 2015 for the first detection of gravitational waves from two stellar-mass black holes [2].

In comparison to light waves, the wave-particle duality of atoms also makes for interesting features in the field of interferometry. Atomic waves are split into two or more alternating paths and the resulting interference patterns are observed for measuring inertial displacements, probing a material’s properties, and understanding quantum mechanics [3]. In the case of a beam of He atoms shot through gratings, higher order Talbot fringes can result in periodic self-images downstream [4]. To generate the Talbot-Lau effect with gratings, the first grating operates as a multitude of incoherent waves that can be projected on a detector from Fresnel diffraction of a second grating [3].

With a wide range of imaging options to choose from, one recently advanced method called grating interferometry provides several advantages to conventional
radiography. A US patent applied for by John Clauser in 1997 explains how a micro-fabricated grating-interferometer can be tuned to observe an element-specific refractive index via the reduction of blurring and effects of X-ray scattering [5].

Grating interferometry produces conventional attenuation images (absorption in the case of X-rays) along with two new imaging modalities: differential phase contrast (DPC) and dark field (DF), which is synonymous with small angle scattering. Neutron DPC imaging was proven as a viable imaging method using a monolithic silicon crystal interferometer and thermal neutrons [6]. Recently, X-ray dark-field imaging was used to find cracks in a triglyceride stearin sample [7] and grain orientation in wood [8]. By providing three image datasets instead of only attenuation images, neutron grating-interferometry offers many new opportunities for materials science imaging.

So how does the grating interferometer work? Fig. 3.1 shows a generic setup for either X-ray or neutron grating-based interferometry. Three gratings are used in the setup: the source grating, G0, adds phase coherence to the beam, the phase grating, G1, creates a periodic interference pattern that can be detected at specific distances downstream (often called the fractional Talbot distances), and the analyzer grating, G2, detects changes in the interference pattern. When a sample is placed in the beam path, changes to the beam’s intensity make it is possible to determine the attenuation, DPC, and DF signals.

3.1 Single-Shot vs Stepped-Grating Interferometry

In 2008, the spatial harmonic method was introduced by Han Wen from the National Institutes of Health. In this imaging method, a grating-based method collected absorption, differential phase-contrast, and dark-field signals from a single raw image [9]. During an X-ray experiment, a grating made of alternating lead and aluminum stripes is used to mask the beam. When a sample is introduced into
Figure 3.1. Example of a grating interferometry setup. The X-ray or neutron source is located on the right and continues through a source (G0), phase (G1), and analyzer (G2) grating until interference patterns are resolved by a detector. A sample can be placed anywhere along the beampath, in this case it is located between G2 and the detector.

In the system, the projection image is altered based on the grating’s pattern, thus making the Fourier spectrum of the projection a combination from the grating and the object [9]. The Fourier spectrum is characterized by a strong peak at zero spatial frequency and two harmonic peaks at the grating’s period. Using an inverse Fourier transform of band-pass filtered areas near the peaks gives a main image and two harmonic images [9]. With scattering generating a higher signal in high spatial frequency components, the division of the harmonic images by the primary images gives the dark-field image [9].

The harmonic peaks can be calculated using Eqn. 3.1

$$Peaks = \frac{2\pi M}{P} \text{ or } \frac{2\pi N}{P}$$  \hspace{1cm} (3.1)

where M and N are integers and P is the period of the grating [10]. The harmonic spectrum is generated from the fact that the spatial frequency of the sample’s projection image is repeated at each peak. There is also an assumption that the har-
monic spectra do not overlap as the central harmonic image does not contain any grating alterations, meaning it can only be affected by absorption, not diffraction or small angle scattering. In the search for microstructures in cortical and trabecular bone, single-shot interferometry showed how the scattering signal increases as a function of increasing bone thickness while the phase signals decreased from more dispersed X-rays [11].

In comparison to single-shot interferometry, stepped-grating interferometry (also referred to as phase-stepping) has been used in a wide range of applications, from observing mouse joints to understanding phase distribution in MgO crystals [12, 13, 14, 15]. Phase-stepping introduces a phase shift between two interfering beams and determines the phase of an object through a set of images (also referred to as interferograms) [16]. Several algorithms have been designed to determine the modulating phase of the beam based upon the sinograms [17, 18]. A calibration method to track π/2 phase shifts of the beam sums the interferograms at specific spatial locations followed by phase determination [16].

In the phase-stepping process, there are two absorption gratings (G0 and G2) and a phase grating (G1). The phase grating is stepped over the course of at least one period of the grating to generate a sinusoidal profile [19]. The grating is stepped laterally to introduce a phase difference between two interfering beams that can be detected through the use of an imaging detector such as a CCD [13]. When the interferograms are turned into projections as a function of sample rotation (sinograms), tomographic volumes can be reconstructed, providing information regarding the absorption, phase, and scattering of a sample.

3.2 Grating Fabrication

There are two different types of gratings: absorption and phase. Absorption gratings are extremely challenging to make for both X-rays and neutrons while phase
gratings are much easier. The high aspect ratios desired of the substrate material can require the use of lithography and electrochemistry, referred to as LIGA (Lithographie, Galvanoformung and Abformung). The following explains the differences between X-ray and neutron grating fabrication.

### 3.2.1 X-ray Gratings

One X-ray grating fabrication method is using X-ray lithography, where a Ti film is sputtered on a large (100 x 100 mm$^2$) Si substrate [20]. Spin-coating and patterning by UV light can be performed on the substrate, with heights of 4 μm, pitch of 5.3 μm, followed by electroplating of Au in the structure [20]. An epoxy, in this example SU-8, is coated at 40 μm thickness and the final step involves wet etching, where KOH removes the Si substrate and buffered HF removes the Ti film [20].

While the process described above is quite complicated, it is simpler to fabricate X-ray gratings than neutron gratings. X-ray gratings G0 and G2 are usually gold printed on graphite or Si, whereas G1 is Si printed on graphite or Si. Au is used because it is easy to print in a high aspect ratio, where the thickness of Au must be greater than 20 μm for 10-40 keV energies [20]. Unfortunately, Au absorbs a lot of X-rays so the material is desired to be as thin as possible. Graphite is used as the substrate material because it does not absorb X-rays. Silicon is also used because it provides a nice phase shift of the wavefront.

A new non-destructive method for determining the height of gold on gratings was found [21]. In this method, the Beer-Lambert law is used to calculate the height of gold as the grating is rotated in an X-ray synchrotron beam.

### 3.2.2 Neutron Gratings

Previous fabrication methods of neutron gratings included evaporating a 300 nm thick silver film on a silicon wafer, followed by milling of 100 μm long trenches, period of 400 nm, using a focused ion beam (FIB) [22]. A major issue that arises
from this process is the remaining gallium ions on the surface of the material from the FIB milling. The sidewalls of the structure are often sloped near the surface, implying that high precision is needed to fabricate the gratings.

Gadolinium sputtering was a more recent technique to attempt to create high-aspect ratios of Gd on silicon. In this process, gadolinium was sputtered to a height of 11 μm on a silicon grating with a period of 4 μm [23]. Quartz can also be used as a substrate material, but the thermal match between gadolinium and silicon is better [24]. During the sputtering process, temperature changes of more than 30°C can result in film stress/cracked films [24].

An easier method for manufacturing neutron gratings was developed in 2013 using gadolinium oxysulfide (Gadox) on silicon structures [25]. During this process, 15 cm silicon wafers are fabricated with a negative photoresist material followed by photolithography and etching of the wafer [25]. Afterwards, Gadox particles are bound with a texanol/acrylic resin mixture and spread into the silicon structure [25]. Once the wafer is filled, excess Gadox is removed by a knife edge and followed by heat treatment and inspection with a scanning electron microscope (SEM) [26]. Neutron interferometry characterization of the gratings showed a maximum visibility of 9.7% at 2.7 Å [26]. A sketch of the grating fabrication process is shown in Fig. 3.2.

3.3 Absorption/Attenuation

Traditional X-ray and neutron imaging produces absorption images for X-rays and attenuation images for neutrons. When X-rays interact with the electrons of a material, the amount of absorption can be detected through changes in the scattering, whether through coherent/elastic scattering or incoherent scattering.

The intensity of the beam detected at each pixel is represented by Eqs. 3.2 and 3.3 [25].
Figure 3.2. Gadox grating fabrication process. Photo-resist of the material coating (a), UV exposure to the material (b), etching with an ion beam (c), fabricated silicon structure (d), addition of Gadox particles in mixture (e), precipitation of Gadox under vacuum (f), removal of excess Gadox (g), and solidification of resulting structure (h). Reprinted from Kim. et. al 2013 with the permission of AIP Publishing [25].

\[ I(m, n, x_g) = a_0(m, n) + a_1(m, n)\cos\left[\frac{2\pi}{p_g}x_g + \phi_k(m, n)\right] \]  \hspace{1cm} (3.2)

\[ I(m, n, x_g) = \sum_k a_k(m, n)\cos\left[k \frac{2\pi}{p_g}x_g + \phi_k(m, n)\right] \]  \hspace{1cm} (3.3)

where \((m, n)\) are detector row, column coordinates, \(p_g\) is the grating period, \(x_g\) is the grating position, \(a_0\) is the sinusoid offset, \(a_1\) is the sinusoid amplitude, and \(\phi\) is the sinusoid phase shift.

Transmission imaging, Eq. 3.4, is calculated using the sinusoidal nature of the wavefront. The amplitude of an open beam, \(a_0\) reference, is related to the amplitude of the same beam with a sample in the beam’s path, \(a_1\) sample.

\[ T(m, n) = \frac{a_1 \text{sample}(m, n)}{a_0 \text{reference}(m, n)} \]  \hspace{1cm} (3.4)

A sample calculation for the average transmission, dark-field and differential phase contrast signals are shown in Fig. 3.3. In the case of Fig. 3.3, the average transmission is 5828 counts.
3.4 Phase Contrast

The discovery of phase contrast imaging in 1942 by Zernike created a new way to observe chemical and physical phenomena based on the wave nature of light with differences in the refractive index and thickness of samples [27]. Determination of the phase of a sample is dependent upon refraction, which is a measure of the change in direction of a wavefront as it propagates through different mediums. Fig. 3.4 shows a ray as it travels from one medium through another.

Refraction of an object is based off Snell’s law, where

\[
\frac{\sin \theta_1}{\sin \theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1}
\]  

(3.5)

When the ray transitions across the interface between two mediums, the change in velocity from \( v_1 \) to \( v_2 \) causes a change in the wavelength of the ray. A common example of refraction is a straw partially immersed in water. When looking at the straw from many viewpoints, the straw appears to bend at the location of the air/water interface. With the refractive index of air at 1.0 and water at 1.33, the bending of light plays tricks on the eye as to the specific location within water.
Figure 3.4. The incident ray on the left travels at a velocity of \( v_1 \), angle of incidence of \( \theta_1 \), through a medium of refractive index \( n_1 \). When this ray interacts with another medium \( n_2 \), the velocity is changed, \( v_2 \), and the angle of refraction, \( \theta_2 \), is altered.

of the straw. When the angle of incidence is equal to the angle of refraction, the sample is said to have total internal reflection.

For X-ray experiments, phase contrast imaging works by obtaining a very high sensitivity (1000 times higher than absorption imaging) to low atomic number elements as a result of a much larger interaction cross section than absorption. [28, 29]. Advances in X-ray phase contrast imaging have proven valuable to the medical field because of the low X-ray dosage that is non-destructive and limits over-expose of samples. Examples of the benefits of phase contrast computed tomography (CT) include imaging of tumorous cells in breast tissue, liver tissue, and lung tissue [30, 31, 32].

The optical density of a sample is calculated as the negative natural log of X-ray transmittance (Eq. 3.6) [28].
Figure 3.5. On the left, elastic scattering of a particle is shown. On the right, inelastic scattering is shown.

\[ D(x, y) = -\ln T(x, y) = \int \mu(x, y, z) dz \quad (3.6) \]

\[ \Phi(x, y) = \frac{2\pi}{\lambda} \int \delta(x, y, z) dz \quad (3.7) \]

\( \lambda \) is X-ray wavelength, \( \mu(x,y,z) \) is the linear absorption coefficient, and \( \delta \) is refractive index of the material.

The differential phase shift, \( \frac{\partial \Phi}{\partial x} \), generated by an object in the beam’s path causes a deflection of the wavefront at an angle \( \alpha \) in the xz plane [33].

\[ \alpha(x) = \frac{\lambda}{2\pi} \frac{\partial \Phi}{\partial x} = \int \frac{\partial \delta}{\partial x} dz \quad (3.8) \]

### 3.5 Dark-Field

There are multiple kinds of scattering, however the focus below is on elastic and inelastic scattering. Elastic scattering refers to a particle that maintains the same kinetic energy after an interaction with electrons (X-rays) or the nucleus (neutrons). With inelastic scattering, there is a change in the kinetic energy of the particle such that it moves slower or faster after hitting a target atom. Figure 3.5 shows the difference between elastic and inelastic scattering as the Q vector is concerned.
The scattering vector $Q$ is defined by Eq. 3.9.

$$Q = 2k \sin \theta = \frac{4\pi \sin \theta}{\lambda}$$  \hspace{1cm} (3.9)

If an energy beam maintains the same energy after interacting with a particle, $k$ and $k'$ are equal. If energy is lost or gained such that $k'$ is less than $k$ or greater than $k$, then this is said to be inelastic scattering.

In the case of imaging, the scattering from a sample can be detected through the dark-field imaging modality [34]. The dark-field signal is representative of the sample visibility, $V$, in an X-ray or neutron beam. The visibility of a wavefront is dependent upon amplitude of sine wave, which are measurements of the intensity at the peaks and minimums (Eq. 3.10).

$$V = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}}$$  \hspace{1cm} (3.10)

When a sample is placed in the path of the beam, intensity changes of the sine wave are calculated for sample visibility and open beam visibility. The DF signal is the ratio of the sample visibility, $V_s$, to the reference visibility, $V_0$, as calculated from the coefficients in Eq. 3.3.

$$DF = \frac{V_s}{V_0} = \frac{a_{1,\text{sample}}(m, n)/a_{0,\text{sample}}(m, n)}{a_{1,\text{reference}}(m, n)/a_{0,\text{reference}}(m, n)}$$  \hspace{1cm} (3.11)

### 3.6 Talbot-Lau

The concept of Talbot-Lau interferometry must first be described by the Talbot effect. In 1836, H.F. Talbot discovered that when light interacted with a diffraction grating, periodic self-images of the grating could be seen downstream through diffraction interference that came to be known as the Talbot effect [35]. The distance at which the image actually appears is referred to as the Talbot length and self-images are generated at periodic distances based on the grating, referred to as
the Talbot distances [36]. The Talbot length is calculated using the wavelength, \( \lambda \), and the period of the structure, \( P_g \) [36]. By dividing the Talbot lengths in half (1/2, 1/4, 1/8...) the periods and sizes of the images are halved as well, creating a fractal pattern (Fig. 3.6) [37]. Observing the Talbot effect becomes challenging when the wavelength is a function of time, such as the neutron wavelength at a pulsed neutron source [3].

3.6.1 Theory

Neutron grating-based interferometry has previously been based on the Talbot-Lau geometry, in which two absorption gratings (G0, G2) and one phase grating (G1) are employed [38, 39]. The G1 grating adds a phase shift to the lamellas on the grating, which can be detected as an interference pattern of a certain period at the detector [40]. The phase gratings can introduce a \( \pi \) or \( \pi/2 \) phase shift, dependent upon expected wavelengths of the X-ray or neutron beam. Fractional talbot distances, \( D_n \), are calculated using Eqn. 3.12, where the talbot orders are integers, \( n = 1, 2, \ldots \), that depend on the period of the phase grating, \( P_g \), and the wavelength of the beam, \( \lambda \).

\[
D_n = n \frac{P_g^2}{2\lambda} \quad (3.12)
\]
\[ P_g = \frac{p_1}{\eta} \text{ with } \begin{cases} \eta = 1, & \text{for a } \pi/2 \text{ phase shift grating or an absorption grating,} \\ \eta = 2, & \text{for a } \pi \text{ phase shift grating.} \end{cases} \] (3.13)

When the sample is located in front of G1, the effective sample to detector distance, \( L S_{\text{eff}} \) must be taken into account (Eq. 3.14).

\[ L S_{\text{eff}} = \frac{(L_1 + L_2 - L_S) \times L_2}{L_1} \] (3.14)

The period of the moiré fringes can be calculated using Eq. 3.15 based on the total instrument length, L, the period of the phase grating, \( P_g \), and the G1-G2 grating distance, D.

\[ P_{\text{fringe}} = \frac{L P_g}{D} \] (3.15)

### 3.7 Far-Field

Recently, in a search for low-X-ray dose clinical imaging, a new interferometry design called far-field interferometry was discovered with X-rays [41]. In this setup, one absorption grating (G2) is replaced with another phase grating, thus doubling the flux through the system [42]. Flux doubling is very attractive for neutron imaging given the relatively low flux available at most facilities. In comparison to Talbot-Lau interferometry, the main advantage of a far-field interferometer is the fine-tuning of the system to probe a wide range of sample scattering lengths, from roughly 50 nm to 5 \( \mu \)m.

The ability to probe a wide scattering range means that the dark-field imaging is essentially a small angle scattering experiment where particle radius and volume fraction can be determined [43, 44]. When applied to complex systems like additive manufacturing or batteries, the far-field DF imaging has the potential to quantify
and visualize the evolution of particle size during chemical and physical changes to materials.

### 3.7.1 Theory

A parameter called the autocorrelation length is representative of the scattering within a sample at specific sizes. Eqn. 3.16 shows how to calculate the autocorrelation scattering length, $\xi$, knowing the wavelength, $\lambda$, the sample to detector distance, $z$, the G1-G2 distance, $D$, the total length of the system, $L$, and the period of the phase grating, $P_g$. The equation below is representative for a neutron far-field interferometer in a symmetric $L_1 = L_2$ configuration.

$$\xi = \frac{\lambda z D}{L P_g} \tag{3.16}$$

There are two experimental setups that can probe auto-correlation scattering length: one changes the G1-G2 grating distance $D$, the other changes the sample to detector distance, $z$. The reason this is possible is through Eq. 3.16, where $z$ and $D$ are both in the numerator in the autocorrelation length equation. It is possible to alter either $z$ or $D$ in this equation, resulting in the same autocorrelation length. In the case of experimental beamlines with limited space, a strict location on sample placement would require changing the grating distance, $D$, to probe different scattering lengths. In comparison, a several meter long neutron guide hall possesses enough space to translate a sample along the beampath and probe a wide range of $\xi$ values.

With the far-field interferometer operating as a new instrument setup, several factors are taken into account (Eq. 3.17).

$$P_g = \frac{\lambda L_1}{\frac{1}{2} P_s} \tag{3.17}$$
The source, $P_s$, and phase, $P_g$, gratings are technology limited so the periods are known. The only two variables left are $\lambda$ and $L_1$. In most instrument setups, $\lambda$ is known, meaning $L_1$ can be calculated. The benefit of the far-field setup is that $L_1 = L_2$ such that the total instrument length, $L$, can be easily calculated.

In calculating $\delta$, the increments of maximum and minimum visibility are based on the inter-grating spacing $D$. For maximum visibility, $\delta$ occurs at $\frac{n\pi}{2}$, where $n = \text{odd integers}$. Minimum visibility would occur at $\frac{n\pi}{2}$, where $n = \text{even integers}$. [41]

$$\delta_{1,2} = \frac{\lambda L_{1,2}}{P_g P_s}$$

(3.18)

With a known autocorrelation length, the challenge becomes determining the relationship between scattering length and particle size. Recent studies focusing on dark-field imaging of mono-dispersed spheres revealed that shape and size of the samples can be extracted through fits to a Gaussian function [44, 45]. The DF signal of a sample can be equated to the summation of the Gaussian function of $\xi$ and thickness of the sample, $t$.

$$V_s(\xi_{GI})/V_0(\xi_{GI}) = \exp \left[ \int_{\text{path}} \Sigma [G(\xi_{GI}) - 1] dt \right]$$

(3.19)

The Gaussian function is expressed by

$$G(z) = \exp \left[ (-9/8)(\xi/r)^2 \right]$$

(3.20)

and $\Sigma$ equal to

$$\Sigma_s = (3/2)(1 - \phi_v)\phi_v \Delta \rho^2 \lambda^2 r$$

(3.21)

$\phi_v$ is the volume fraction of the scatterer, $\Delta \rho$ is the scattering length density contrast as calculated from the NIST website[^1], $\lambda$ is the wavelength, and $r$ is the

[^1]: https://www.ncnr.nist.gov/resources/activation/
scatterer radius. To simplify the equation, we use the term \( A \) to represent the variables below.

\[
A = \frac{3}{2}(1 - \phi)\phi_{v}\Delta\rho^{2}t
\]  

(3.22)

The DF signal can be represented as \( \ln \frac{V_s}{V_0} \) and set equal to the Gaussian function through Eq. 3.24.

\[
\ln \left( \frac{V_s}{V_0} \right) = \frac{2}{\lambda^2} \left( \Sigma_s/\lambda^2 \right) \left[ G(\xi) - 1 \right] t
\]

(3.23)

\[
= \frac{3}{2}(1 - \phi)\phi_{v}\Delta\rho^{2}r[\lambda(r, \xi) - 1]t
\]

(3.24)

By substituting \( A \) in for the volume fraction and density, it is easier to determine a fitting function with the equation below, where \( A \) and \( r \) can vary.

\[
-ln \left( \frac{V_s}{V_0} \right) = A \times r \times [1 - G(\xi)]
\]

(3.25)

The far-field setup and complicated nature of scattering lengths have not been applied to complex systems like additive manufacturing or lithium ion batteries. The goal of this research is to utilize this setup and Talbot-Lau interferometry for imaging of additively manufactured samples, lithium ion batteries, and crystals.
References


Chapter 4
Lithium-Ion Battery Imaging

4.1 Introduction

Since the discovery of the voltaic cell by Volta in 1800, scientists across the globe have looked at utilizing elements of differing potentials for specific battery applications. Such examples include NiCd batteries for power tools, lead acid batteries for golf carts and automobiles [1], and lithium ion batteries (LIBs) for medical equipment and renewable energy storage [2, 3]. In 1991, the Sony company commercialized the production of LIBs for application in consumer devices [4]. To harness the best characteristics of batteries, there is a trend for higher energy density secondary (rechargeable) batteries like Li-S, LiO₂, or Li-metal to replace older batteries [5, 6, 7, 8, 9].

Understanding lithium battery functionality and performance is key towards advancing energy storage options for the future. On the micron scale, determining the role of lithium dendrite and particle formation/deformation can lead to optimized battery construction. Lithium dendrites are clumps of lithium that become “dead” within the battery chemistry and can penetrate the separator material, leading to short-circuiting of the cell [10]. Previous studies with Li metal anodes have shown promise in the prevention of lithium dendrites via a high surface energy during battery cycling [5]. With safety a primary concern of lithium battery operation, the value of understanding and visualizing the chemical processes within lithium batteries remains important.

There are many different types of lithium batteries: lithium-ion (Li-ion), lithium air, and lithium ion polymer (Li-po) to name a few. The main difference between Li-ion and Li-po batteries is in the electrolytic material. In Li-ion batteries, a
Figure 4.1. The layering within a sample lithium ion battery. On average, a single layer of carbon, LiMO$_2$ or separator material can range from 10 - 100 $\mu$m. In a battery with ten layers, this averages out to roughly 1 mm.

lithium-salt electrolyte (usually LiPF$_6$) is held in an organic solvent to promote lithium ion movement through aqueous material [8, 11, 9]. In Li-po batteries, solid electrolytic materials such as nano-fibrous ethylene carbonate/ dimethyl carbonate (EC/DMC) gels possess high ionic conductivity and high rate capabilities [12]. While both materials have been thoroughly investigated, knowledge regarding their particle size, cycling behaviors, and spatial dependence is limited.

In a typical battery, a difference in chemical potentials between cathode and anode materials is used to convert chemical free energy into electrical energy. For the case of LIBs, the multitude of LIBs vary based on properties of the anode, cathode, and electrolyte, where the anode and cathode are defined for the discharge process. A sample LIB is shown in Fig. 4.1 to show how the electrode materials like LiMO$_2$ and graphite surround metallic elements such as Al and Cu.

Electrolytes form a solid electrolyte interface (SEI) film through reduction on the anode and oxidation on the cathode during the first charge/discharge cycle [13]. While the total thickness of the SEI is thin, between 15-25 Å, it must be a good conductor for lithium ions while preventing direct contact between the electrode and electrolyte material [11]. For a LiCoO$_2$/Li or LiCoO$_2$/C$_6$ cell, the SEI occurs on the anode at a cell voltage between 2.5-3.6 V and on the cathode at 3.95 V [13].
Typical electrolyte solutions are made of an alkyl carbonate, ethylene carbonate, or dimethyl carbonate solution with LiPF$_6$ as a salt [14, 15, 16, 17]. LiFePO$_4$ batteries with LiFePO$_4$/C as the cathode material and lithium metal as the anode have a potential charge capacity of 170 mA·h·g$^{-1}$ using a sol-gel method to ensure a small particle size [18]. In LiMO$_2$ batteries (where M = Ni, Mn, Co), cathode material is combined with binder and pressed onto the electrode, resulting in a potential charge capacity of 140 mAhg$^{-1}$ [18, 19]. Common anode materials include graphite due to the ease of lithium intercalation, where interactions between lithium cations and the graphite layers are strong due to $\pi$ screening of the electrostatic repulsions of the cations [20].

Imaging an intact battery provides unique information not otherwise available. Typical battery components and materials are: (a) a copper negative electrode, (b) lithium ions intercalated into graphite, (c) an electrolyte barrier for electron insulation, but permitting rapid lithium ion diffusion, (d) metal oxide such as Li$_x$CoO$_2$, and (e) an aluminum positive electrode [21]. For the process of battery discharge, electrochemists label the electrodes as anode (negative) and cathode (positive). The solid-state chemist formulates a high Fermi level anode material, such as the LiC$_6$, and a low Fermi level cathode material, such as Li$_x$CoO$_2$, ($x = 0.2$, battery charged). The range of Fermi levels in the charged battery challenges the stability of the organic electrolyte; decomposition of a small amount of electrolyte at the surface of the electrode material creates a solid/electrolyte-interface (SEI) layer. The stability of the SEI layer is very important for the long term performance of the battery.

We note several significant recent works using neutron imaging and diffraction: lithium concentrations and mobilities have been imaged in lithium-ion batteries [22]; neutron diffraction has been reported for several commercial batteries [23]; and
maps of lithium hydride and hydrolysis products [24]. Herein, we briefly discuss our experiences with neutron-based battery imaging, survey recent neutron imaging reports, and speculate upon the future of this field, particularly with respect to the possibilities of time-of-flight neutron beamline imaging, hydrogen scattering, and grating-based interferometry. In our search, we show some of the first spatially resolved scattering images of batteries.

4.2 Battery Characterization

Characterization of Li-ion batteries can be done in several manners. An easy way is to observe the battery capacity after a number of charge/discharge cycles. If the original capacity of a battery is 43 mAh and after 1000 cycles the capacity is 26.5 mAh, the battery is said to be at 50% of its original state of charge (SOC).

Previous characterization methods of Li-ion batteries includes neutron diffraction. Diffraction peaks from a fresh and worn large Li-ion cell (200 x 120 x 5 mm, 15,000 mAh) showed inhomogeneous cell degradation at the edge of the batteries [25]. However, problems with this diffraction experiment included limited d-spacings of 1.6 - 2.0 Å and 2.1 - 2.4 Å [25]. Spatially resolved neutron diffraction of Fe/NaCl and cylindrical Li-ion 18650 cells have shown promise in the search for non-destructive testing of complex batteries [26, 27]. In the experiments performed herein, it is anticipated that probing with wavelengths from 2 Å up to 5 Å will reveal more diffraction peaks and information from within the batteries.

4.3 Experimental

The following sections describe experimental setups and results obtained from a series of experiments performed on Li-ion and Li-po batteries. Neutron diffraction and statistical analysis methods are presented first. Next, X-ray Talbot-Lau and near-field interferometry images at CAMD are described. Third, neutron Talbot-Lau interferometry and Bragg edge imaging experiments show the benefits of neu-
trons versus X-rays. Last, the first application of a neutron far-field interferometer to Li-ion batteries is presented.

4.3.1 Neutron Diffraction at SNS VULCAN

Neutron time-of-flight (TOF) diffraction experiments were performed at the SNS Engineering Materials Diffractometer VULCAN (ORNL, Beamline 7). The VULCAN instrument was set up with a 5 mm x 10 mm x 5 mm sample volume, peak neutron flux at 2 Å, and 30 Hz pulse repetition rate. The beamline length is 43.754 m and the correct two-theta angle is 47.7°. Once a TOF spectrum is obtained for the sample, the peaks are removed by using a sample TOF vanadium spectrum (Fig. 4.2).

The remaining points are fitted to a normalized cubic spline interpolation and compared to a CeO$_2$ diffraction plot from VULCAN based on the Inorganic Crystal Structure Database (ICSD 165720). This CeO$_2$ diffraction plot from VULCAN is generated with the raw counts versus flight time data using the beamline length and effective two-theta (Fig. 4.3).
Figure 4.2. TOF spectra for a vanadium sample with peaks (top) and without peaks (bottom).
Figure 4.3. At the top is a neutron diffraction plot for CeO$_2$ at the VULCAN instrument. Below the spectrum is a table of the interpolated (VULCAN) and literature $hkl$ wavelengths for CeO$_2$. 

<table>
<thead>
<tr>
<th>h</th>
<th>k</th>
<th>l</th>
<th>VULCAN</th>
<th>int.</th>
<th>lit.</th>
<th>int.</th>
<th>d-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1.9141</td>
<td>100</td>
<td>1.91332</td>
<td>100.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1.63245</td>
<td>13</td>
<td>1.63169</td>
<td>12.6</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1.3537</td>
<td>22</td>
<td>1.35292</td>
<td>24.8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1.21086</td>
<td>12</td>
<td>1.21009</td>
<td>13.3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1.10534</td>
<td>51</td>
<td>1.10466</td>
<td>65.7</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.95725</td>
<td>19</td>
<td>0.95666</td>
<td>24.5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>0.91539</td>
<td>7</td>
<td>0.91474</td>
<td>7.8</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0.90258</td>
<td>7</td>
<td>0.90195</td>
<td>9.0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>0.85627</td>
<td>27</td>
<td>0.85566</td>
<td>38.9</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>0.81636</td>
<td>5</td>
<td>0.81584</td>
<td>5.9</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0.78158</td>
<td>9</td>
<td>0.78111</td>
<td>10.7</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.75821</td>
<td>6</td>
<td>0.75779</td>
<td>5.3</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>0.72358</td>
<td>33</td>
<td>0.72317</td>
<td>54.8</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0.70501</td>
<td>7</td>
<td>0.70454</td>
<td>6.8</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0.67602</td>
<td>5</td>
<td>0.67646</td>
<td>5.9</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>0.66122</td>
<td>4</td>
<td>0.66114</td>
<td>2.0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>0.63811</td>
<td>18</td>
<td>0.63778</td>
<td>31.5</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0.61514</td>
<td>3</td>
<td>0.62076</td>
<td>3.3</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.59927</td>
<td>3</td>
<td>0.59401</td>
<td>4.8</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>0.58758</td>
<td>3</td>
<td>0.59046</td>
<td>5.8</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>0.57112</td>
<td>11</td>
<td>0.57689</td>
<td>16.9</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>0.55256</td>
<td>11</td>
<td>0.55233</td>
<td>15.4</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>0.50881</td>
<td>5</td>
<td>0.50464</td>
<td>2.3</td>
</tr>
</tbody>
</table>
4.3.2 Principal Component Analysis

Analysis methods of structural battery information have been limited by the ability to predict a compound’s concentration at a various state of charge (SOC) during cycling. Previous works involving analysis methods called principal component analysis (PCA) and multivariate curve resolution (MCR) have shown success tracking the chemical changes of lithium-ion batteries [28, 29, 30]. These methods focus on the use of a bilinear model to describe how a data set in a rectangular matrix, here 655 x 870, (d-space by state-of-charge) is decomposed into spectra (eigenvectors, 655x7) and concentrations (7 x 870). The spectra are linear combinations of pure d-spacing spectra. While reducing the dataset proved advantageous for a two-component system, it is of interest to separate a battery system into its many components instead of grouping components together.

In this study, we focus on utilizing a synthetic LiCoO$_2$/graphite battery dataset for PCA/MCR analysis. The synthetic system was expected to differentiate seven different components of both cathode and anode materials. By understanding how the state of charge and concentrations of different components are correlated, we can better understand how the chemical compounds in battery systems behave during diffraction and imaging experiments.

In order to generate a synthetic battery dataset, structural information of Cu, Al, LiCoO$_2$, Li$_{0.75}$CoO$_2$, Li$_{0.5}$CoO$_2$, LiC$_6$, LiC$_{12}$, and C$_6$ was obtained through the Inorganic Crystal Structure Database (ICSD). Models were then generated using CrystalMaker and imported into CrystalDiffract to obtain simulated neutron diffraction spectra of individual components. Data was originally viewed in TOF vs. intensity, however after consideration of experimental studies, it was identified that units of d-spacing vs. intensity would be more beneficial towards peak analysis.
Figure 4.4. On the left is the Cu synthetic intensity, on the right LiCoO$_2$ intensity.

An appropriate d-spacing range of 0.5 to 2.5 Å was selected as the optimal range for data analysis due to limited value of peaks below 0.5 Å and above 2.5 Å.

Synthetic data analysis of the d-spacing versus intensity spectra was performed using a Mathematica program. The dataset was imported into Mathematica, where d-space minimums and maximums were determined for each chemical component. Selected data plots of Cu and LiCoO$_2$ can be seen in Fig 4.4.

After importing the datasets, a chemical equilibrium problem needed to be solved as the percentage for each component increases, decreases, or remains the same as the battery is charged or discharged. In the case of this synthetic dataset, as the battery is charged from 0 to 100% SOC, the concentrations of LiCoO$_2$ and C$_6$ decrease while the concentrations of Li$_{0.5}$CoO$_2$ and LiC$_6$ increase. Since Cu and Al are not involved in the lithium intercalation/de-intercalation process, both components were not involved in the equilibrium solution. Interpolation functions of the internal components were then generated from the synthetic data plots and the equilibrium solutions. A plot of the interpolation functions relating SOC to concentration is shown in Fig 4.5.

With the synthetic dataset pertaining to a perfect environment, random noise was added to create more realistic experimental results (Fig 4.6).
Figure 4.5. Interpolation functions generated in Mathematica for LiCoO$_2$, Li$_{0.75}$CoO$_2$, Li$_{0.5}$CoO$_2$, LiC$_6$, LiC$_{12}$, and C$_6$.

Figure 4.6. Simulated battery dataset with Gaussian noise
With a noisy dataset in hand, the next step involved performing singular value decomposition to reduce the dataset into eigenvectors based upon values of eigenvalues. After reduction of the dataset, MCR analysis occurred through the use of free software\(^1\) [31]. The MCR-ALS graphical user interface (GUI) was used to look at spectral and concentration information. The MCR-ALS GUI provided two main benefits over PCA analysis: non-negativity and concentration information. Non-negativity constraints in the program, provided by non-negative least squares (NNLS) and fast non-negative least squares (FNNLS), were selected to ensure that none of the spectra or concentrations would be negative. This benefit enabled accurate and facile determination of the various components in the system.

### 4.3.3 X-ray Experiments at CAMD

Due to issues with monochromator installation at the CAMD imaging beamline, a Hamamatsu microfocus X-ray tube was used in place with a Talbot-Lau interferometry setup. Features of the X-ray tube (model L9181-02) include a voltage of 40 to 130 kV, a focal spot size of 51 \(\mu\)m, and a maximum output of 39 W \(^2\).

Four sets of commercial Li-po batteries of varying capacity and thickness were used for experiments: PGEB 0054018 (15 mAh, 0.5 x 40 x 18 mm), GMB 201030 (43 mAh, 2 x 10 x 30 mm), PGEB 0054338 (50 mAh, 0.5 x 43 x 38 mm), and GMB 161652 (90 mAh, 1.6 x 16 x 52 mm)\(^3\). For the nominal 43 mAh batteries, one fresh (1 cycle, 37.7 mAh), one slightly worn (125 cycles, 39.9 mAh), and one worn (1790 cycles, 38.6 mAh at the last cycle) were taped on a sample stage such that all three were in the beam path during imaging. Batteries were connected to a 8 channel battery analyzer (6-3000 mA) from MTI and charged to 4.2 V \(^4\). After allowing to sit for 20 minutes to minimize lithium movement in the cells, imaging

---

\(^1\)mcrals.info
\(^3\)http://www.powerstream.com/thin-lithium-ion.htm
\(^4\)http://www.mtiixtl.com/8ChannelsBatteryAnalyzer-BST8-3.aspx
was performed for the charged batteries. Upon completion of charged imaging, the batteries were then discharged to 2.7 V at a rate of 0.93 C (40 mA/hr for a 43 mAh battery) and allowed to rest for 20 minutes.

When the monochromator was re-installed, synchrotron X-ray radiation of 38 keV was used. 2 second exposures provided energy flux to penetrate the batteries of differing thicknesses. Under the direction of Dr. Han Wen (NIH), a near-field interferometer was also setup using the 38 keV X-rays. A sample beamline setup of the near-field interferometer is shown in Fig. 4.7.

In this setup, the total distance from the sample to G2 is 390 mm, the distance from G2 to the detector is 118 mm, and the distance between G1 and G2 changed from 3 - 143 mm. A Pilatus 100K position sensitive photon counter with 172 μm pixels was used. In calculating the autocorrelation length, ξ, a different equation must be used than for neutrons (Eq. 4.1). The reason for this discrepancy in autocorrelation length formulas is that given the parallel beam geometry of synchrotron
Figure 4.8. Autocorrelation length calculations for March 2017 experiments at NIST. The range of $\xi$ is from 20 nm to 904 nm. $\lambda = 0.326$ Å, $D$ ranges from 3 mm - 133 mm, and $P_g = 2.4 \mu$m.

X-rays, the fringe lines generated by G1 are projected onto G2 with little magnification and converted into broad fringes by G2. Scattering effects blur out the $4.8 \mu$m fringes and beyond that the moire fringes are broad.

$$\xi = \frac{\lambda D}{P_g}$$

The range of autocorrelation values for this near-field is from 20 nm – 904 nm as shown in Fig. 4.8.

### 4.3.4 Neutron Talbot-Lau Grating Interferometry at HZB

Talbot-Lau neutron interferometry experiments were performed at the HZB-CONRAD2 beamline using polychromatic beam at a designed wavelength of 3.5 Å [32, 33]. Distance between the G0-G1 gratings, $L_1$, was 4.78 m, G1-G2 distance, $L_2$, was 2.27 cm. The instrument operated in the first order Talbot distance with a grating
period $P_g$ of 7.97 $\mu$m, a sample-to-detector distance, $z$, of 5.0 cm, and a $G_1-G_2$ distance of 2.27 cm, yielding an auto correlation scattering length, $\xi$, of 1.97 $\mu$m.

Li-ion and Li-po batteries of differing capacities (360 mAh, 90 mAh, 43 mAh, 15 mAh, and 50 mAh) were glued to an aluminum plate (Fig. 4.9). Batteries were first imaged at their initial resting voltage (around 3.7 V) with 20 second exposures at three different angles ($0^\circ$, $10^\circ$, and $20^\circ$) with 14 grating steps over 1.2 mm, slightly above the 0.8 mm period of the grating. For the 15 mAh batteries, they were charged up to 4.2 V at 15 mA and left to rest for 20 minutes, following by “charged” imaging. After “charged” imaging, the batteries were discharged at 15 mA down to 2.7 V and left to rest for 20 minutes. At this point, “discharged” imaging took place.

### 4.3.5 Neutron Bragg Edge Imaging at HZB

Bragg edge imaging was performed with monochromatic beam at the HZB CONRAD2 beamline on a fresh and worn 43 mAh Li-po battery (Fig. 4.10). Imaging was performed with 120 second exposures from 3 - 4.2 Å by steps of 0.02 Å. No binning was required for this imaging in comparison to the tomography datasets.
4.3.6 Neutron Imaging at NIST

In the first application of neutron far-field interferometry to batteries, imaging was done at the NG6 cold neutron imaging beamline (NCNR NIST, Gaithersburg). The neutron flux at the detector has not yet been measured; it is estimated at $10^6$ n-cm$^{-2}$s$^{-1}$. The slit-to-detector distance was 8.71 m ($L$); detector-to-G2 was 4.355 m ($L_2$) as was slit-to-G2 ($L_1$), hence the interferometer was operated in a symmetric geometry. The D-scanning direction of G1 was towards the slits, i.e., upstream. The beam defining slits were fabricated from tungsten and $^6$Li/borated polyethylene. The slits were adjusted to an uncalibrated opening of 0.5 mm horizontal and 5 cm vertical; the images are blurred along the vertical direction. The interferometry image sharpness is excellent along the horizontal direction with $L/D \approx \frac{8.71 \text{ m}}{0.5 \text{ mm}} = 17,000$ while only 170 along the vertical direction. The wavelength of the beam ranged from 5.2 - 6.3 Å and the period of the gratings was 2.4 μm. Boraflex (silicone polymer and boron carbide powder) was used to cover the slit and reduce streaking from the neutron beam. The scintillator was a 150 μm thick $^6$LiF:ZnS screen.

Both gratings were mounted on stages for independent roll, pitch, yaw adjustments as well as collective roll, pitch, yaw adjustments. In addition, G1 was on
linear translation stage for scanning $D(G1-G2)$ from closest possible approach, 3 mm, to over 40 mm separation. A critical adjustment is the roll positioning of $G1$ with respect to $G2$; an angular settability of $0.001^\circ$ was needed. The centers of angular adjustments did not coincide; this is an area worthy of improvement.

The same set of Li-po batteries from the experiment at CAMD (see Section 4.3.3) were also imaged using the neutron far-field interferometer at NIST. Batteries were charged to 4.2 V, left to rest for 20 minutes, then imaged. Once the charged imaging was complete, the batteries were discharged to 2.7 V at a rate of 0.93 C (40 mA/hr for a 43 mAh battery), left to rest for 20 minutes, and imaged. After the first set of discharged battery imaging was performed, a second dataset of discharged images was obtained two hours later to observe any differences in battery quality. A sample image showing a worn, fresh, and slightly worn battery is shown in Fig. 4.11.

The sample was mounted on rotation stage and positioned 5 cm downstream of $G2$. This position yields an optical magnification of $m = \frac{L_2}{L_1} = 2$. A 150 $\mu$m thick $^6\text{LiF/ZnS}$ scintillator was imaged with an Andor NEO sCMOS, air and Peltier
The interferometer was operated in D-scan mode. In D-scan mode, a fixed sample rotation was used and D (G1-G2 distance) was scanned in 1 mm increments, typically from 7 mm to 35 mm corresponding to a range of autocorrelation scattering lengths 338 nm to 4.5 \( \mu \text{m} \). Calculations for the autocorrelation length, \( \xi \), in the 2016 experiments are shown below in Fig. 4.12.
The interferometer was also used in stepped-grating mode, typically with 9 step positions evenly spaced over 3 μm, slightly more than the 2.4 μm phase grating period. In one test, 12 positions over 3 μm gave slightly better results as measured by $\chi^2$ in the visibility images. Two software packages were used for reduction of the interferograms to projections of attenuation, differential phase contrast, and dark-field (small angle scattering) images. The NIH software, graciously provided by Dr. Han Wen, was used for initial processing at the beamline and is based off Fournier analysis [34]. The LSU software was used for production work off-line [35]. The tomography projections were reconstructed with SIRT (ASTRA toolbox) [36]. The small angle scattering signal extracted from the dark-field images as a function of the D-scans was analyzed for the autocorrelation lengths.

4.4 Results and Discussion

Battery diffraction and X-ray/neutron imaging was utilized in a step-wide manner to try and observe battery characteristics after certain experiments. The discussion below explains how neutron diffraction of a fresh and worn Li-ion battery led to the trial of a statistical analysis method (principal component analysis/multivariate curve resolution). Although this option was explored, Talbot-Lau X-ray imaging was thought to be a simple way to reveal information within the thin Li-ion and Li-po batteries. Neutron Talbot-Lau interferometry and Bragg edge imaging were then explored as a more costly, but useful option. Finally a new interferometry setup called far-field interferometry was explored with the batteries using both X-rays and neutrons.

4.4.1 Neutron Diffraction at SNS VULCAN

Neutron diffraction experiments are meant to reveal specific information about $hkl$ planes within a polycrystalline material. In the case of a Li-ion/Li-po battery, this becomes more complex due to the wide range of materials used for anode, cath-
ode, electrolytic, and separator material. Figures 4.13 and 4.14 show the neutron diffraction patterns taken from 0.5 - 2.5 Å at the VULCAN instrument at ORNL, SNS.

Li_{0.5}CoO_2 and Li_xC_6 peaks at 1.85 and 2.15 Å, respectively, are absent in the worn battery as compared to the fresh battery. While the understanding is that lithium is not leaving the cell, it is trapped in a non-electrochemically active phase within the battery such that lithium intercalation cannot be fully achieved as in the first charge/discharge cycle. Also observable is a decrease in LiC_{12} at 1.78 Å and an increase of C_6 at 1.69 Å.

To further understand the relevance of the neutron diffraction data, spectral analysis using PCA/MCR was performed. MCR analysis of a synthetic dataset generated from CrystalMaker and CrystalDiffract proved valuable in understand-
Figure 4.14. Neutron diffraction for a worn, 25% of original SOC, Li-ion battery.

Spectral analysis based on the d-spacing of the components is shown in Fig 4.15. In the spectrum, it is observed that MCR separates the different components dependent upon their likelihood of having the characteristics at a specific d-space. Separation of the three LiCoO$_2$ species and the three graphite species show a

Figure 4.15. Reduced spectral information of 7 component battery system using MCR-ALS. The Cu + Al signal is not changing at 100%.
Figure 4.16. Reduced concentration information of a 7 component battery system using MCR-ALS. The x-axis is in experimental units of diffraction data set sequence number.

The relative shift in d-space from one component to the next. Further explanation of this can be seen in Fig 4.16, with the SOC and concentration of each component.

In comparison to the interpolation functions generated in Mathematica in Fig 4.5, Fig 4.16 exemplifies how well MCR represents the original synthetic dataset. Even with a reduction in the total number of data points, MCR still was able to separate the individual components consistently based upon the parameters set at the beginning of generating the dataset.

When applied to experimental data, it is anticipated that this new workflow will enable chemical tracking of individual components of unique battery systems. Altering the chemical equilibrium calculations is facile and dependent upon the chemical substances of the cathode and anode materials. From there, the interpolation functions generated can provide a way to measure the reliability of MCR analysis in application to other batteries. Future work involves the incorporation of neutron imaging in combination with PCA/MCR analysis. In doing so, we will provide a new way to visualize internal battery components and understand on a chemical level how the concentration of compounds increase and decrease during the cycling process.
4.4.2 X-ray Talbot-Lau Imaging at CAMD

Synchrotron radiation X-rays at CAMD provided enough flux to penetrate through 0.5 - 2 mm thick Li-po batteries. Layering within the batteries can be seen through absorption images in both the charged and discharged states using a Talbot-Lau interferometry setup (Fig. 4.17).

![Figure 4.17. Absorption projections for a charged (left) and discharged (right) 43 mAh battery at 0 degrees.](image)

The absorption images were expected to show minor changes due to the intercalation of lithium increasing the volume in a charged battery. However, even with this expansion around 10%, this expansion is not observed in these unconstrained batteries. In comparison, major changes in the dark-field images are detected between charged and discharged states (Fig. 4.18).

![Figure 4.18. Dark-field projections for a charged (left) and discharged (right) 43 mAh battery at 0 degrees.](image)

The scattering signal is becoming stronger in the discharged state due to particle size changes at a $\xi$ value of 575 nm. The migration of scattering from both sides of
the charged battery to solely the left side of the battery in the discharged image implies inhomogeneous migration within the cell.

4.4.3 Near-Field X-ray Imaging at CAMD

With the help of Dr. Han Wen (NIH) in March 2017, a near-field X-ray interferometry setup was designed and implemented at the CAMD imaging beamline. This setup of the near-field interferometer was designed to observe smaller scattering features than in the far-field neutron interferometer experiments at NIST (Section 4.4.6). The near-field experiments showed changes in battery scattering for a 43 mAh battery at resting voltage (3.7 V) from 20 nm to 910 nm (Figs. 4.19 and 4.20).

Figure 4.19. Dark-field image of scattering at a grating distance of 3 mm. This corresponds to an autocorrelation length, \( \xi \), of 20 nm.

With the battery at rest for imaging, the differences of the images at 20 nm versus 904 nm are quite large. At 20 nm, there is little to no scattering in most of the battery. The few bright spots in the center-bottom of the battery are due to scintillator defects and do not change throughout imaging. At the largest autocorrelation length of 904 nm, the battery is observed to have inhomogeneous scattering centers throughout the battery. The brightest (red-colored) regions have an average scattering signal around 0.58 while the green and blue regions have signals around...
Figure 4.20. Dark-field image of scattering at a grating distance of 133 mm. This corresponds to an autocorrelation length, $\xi$, of 904 nm.

0.80. The gap between bright colored regions on the left side of the battery indicates a region of smaller particles, possibly due to battery degradation. Remnants from the moiré fringes are observable in the middle of the batteries, indicating the need for higher quality data processing.

Figure 4.21. Dark-field images for a fresh battery when charged (left) and discharged (right). The autocorrelation length probed equates to 496 nm.

Figure 4.22. Dark-field images for a worn battery when charged (left) and discharged (right). The autocorrelation length probed equates to 496 nm.
Figure 4.23. Comparison of dark-field signal for fresh and worn 43 mAh batteries when $\xi = 496$ nm

To determine the best scattering length to observe changes in the batteries, all of the images were explored from 3 mm to 133 mm in ImageJ. By taking histograms of a 650 x 650 pixel region for the center of the sample, it was determined that the 63 mm grating distance ($\xi$ of 428 nm) provided the most contrast between fresh and worn batteries, with the 73 mm grating distance ($\xi$ of 496 nm) in a close second (Fig. 4.23).

4.4.4 Neutron Bragg Edge Imaging at HZB

Neutron Bragg Edge imaging of fresh and worn, charged and discharged batteries revealed several interesting features. The plots of the macroscopic attenuation cross section, $\Sigma$, versus wavelength are generated from the standard

$$\Sigma = \ln \frac{\text{open beam} - \text{closed beam}}{\text{sample} - \text{closed beam}}$$

but an additional correction was needed to account for a change in reactor power between measurement of the open beam and the sample measurement. A reference area was chosen in the images and a correction applied to the discharged data as

$$\Sigma'(\text{discharged}) = \Sigma(\text{discharged}) + [\Sigma(\text{charged(reference)}) - \Sigma(\text{discharged(reference)})]$$

(4.3)
Figure 4.24. Image of fresh (left) and worn (right) 43 mAh batteries at 3.18 Å.

The transmission signal through the batteries was always higher for the discharged battery versus the charged battery. There are two possible explanations as to why the signal is higher in the fresh battery than the worn battery. With “dead” Li not getting involved in migration from one electrode to another, this results in a larger difference between the charged and discharged states. The more likely reasoning is from a remaining graphite, LiC\textsubscript{x}, or LiCoO\textsubscript{2} Bragg peak above 4.2 Å. Since the signal for a Bragg edge increases for an \textit{hkl} peak up to a specific wavelength, a peak just past the edge of the spectra would result in higher signal just before that point. At very large wavelengths, the samples become opaque so it is anticipated that a Bragg edge is located shortly above 4.2 Å.

The feature at 3.6 to 3.8 Å in the worn battery was expected due to LiC\textsubscript{6} depletion as the battery is worn [37]. A strong LiC\textsubscript{6} peak from the 111 plane is located at 3.65 Å and in combination with peaks from large carbon compounds like C\textsubscript{60} at 3.77 Å, there is overlap from several Bragg edges. The strongest peak around
3.2 Å is likely a result of $\text{Li}_x\text{CoO}_2$, where $x = 1$ for a charged battery and 0.5 for a discharged battery [38]. If the lithium is becoming inactive in the chemical process, this could explain the large difference between charged and discharged states.

### 4.4.5 Neutron Talbot-Lau Interferometry at HZB

This section and the following section, Sec. 4.4.6, are linked. The Talbot-Lau gives good spatial resolution, but at a fixed interferometer autocorrelation scattering length. The far-field interferometer has poor spatial resolution, but scans a wide range of scattering lengths. For all of the radiography images below, air has a value of 0 while the highest amount of scattering is 1. Neutron radiography utilizing Talbot-Lau interferometry showed a slight difference between charged and discharged 15 mAh Li-po batteries in the dark-field at $\xi$ of 1.97 µm, $\lambda$ of 3.5 Å (Fig. 4.27).

When both the fresh and worn batteries are charged, there is not a large difference in the scattering within the samples except at well-defined locations. Average dark-field signals for battery cross sections are 0.21 for the fresh battery and 0.20 for the worn battery. The major difference occurs at the top of the fresh battery,
Figure 4.26. Bragg edge graph from 3 - 4.4 Å of a fresh battery. The (+) symbol is representative of the charged state and the (-) symbol a discharged state.

Figure 4.27. Dark-field images of charged (left) and discharged (right) 15 mAh Li-po batteries. In each image, the battery on the left is the fresh battery and the battery on the right is the worn battery. The scattering length $\xi$ was 1.97 μm and $\lambda$ was 3.5 Å.

where the higher scattering values (around 0.65) in this location are associated with electrolytic material. This area of high scattering would also be observed in the worn battery, however due to field of view constraints for the gratings, this feature was unobserved. Also of note is the vertical scattering features in both batteries, specifically in the middle and edges of the sample.
When the battery is discharged, the scattering values increase to 0.29 for the fresh battery and 0.27 for the worn battery. While this is not a large increase in the scattering signal, it implies that lithium migration is easier to observe in a discharged battery than a charged battery. The more uniform distribution of dark-field signal in the discharged batteries versus the charged batteries supports this hypothesis.

4.4.6 Far-Field Neutron Imaging at NIST

Thicker batteries (1 mm and up) resulted in more scattering data with the far-field neutron interferometer as compared to the X-ray data obtained at CAMD. The 43 and 90 mAh batteries exhibited the largest changes between charged and discharged, fresh and worn samples but results from all four battery sets will be shown.

Of interest first are the highest capacity batteries of 90 mAh. While these batteries are not the thickest of the group of four battery sets (1.6 mm), it was anticipated that the multiple electrode layers would provide useful scattering signal. In Fig. 4.28, we observe a large difference in the scattering signal between a fresh and worn battery at an autocorrelation length of 1.125 μm. The worn battery had 1780 charge/discharge cycles to a 81% SOC while the fresh battery only had one cycle previously to build up the solid electrolyte interface.

The signal in the worn battery hovers around 0.40 while the fresh battery is homogeneous around 0.80. Of interest in the worn battery is the left edge, which has a high scattering signal of 0.75. As this region is not in a fold of the battery (located at the top and bottom of each battery), it is unique that the scattering signal would be high on just one side of the battery. When the grating spacing is increased to 15 mm, the scattering signal increases for both batteries compared to D(G1-G2) of 10 mm. With the autocorrelation function going from 1.125 μm to
Figure 4.28. Neutron scattering images of charged (left) and discharged (right) 90 mAh batteries. In the mount are a worn (left) and fresh (right) battery. The neutron scattering autocorrelation length depends on the D(G1-G2) grating spacing of 10 mm. Here, $\xi = 1.125 \, \mu m$.

1.688 $\mu m$, this is to be expected. The autocorrelation function includes scattering up to a certain value so at 1.688 $\mu m$, this implies that any scattering from particles 1.688 $\mu m$ or smaller will be included.

Upon increasing the scattering length to 2.25 $\mu m$, the worn battery appears to gain uniformity across the left side of the sample while the fresh battery remains homogeneous (Fig. 4.29).

Figure 4.29. Neutron scattering images of charged (left) and discharged (right) 90 mAh batteries. In the mount are a worn (left) and fresh (right) battery. The neutron scattering autocorrelation length depends on the D grating spacing of 20 mm. Here, $\xi = 2.25 \, \mu m$.

When the scattering length is increased to 2.813 $\mu m$, the highly intense scattering centers in the worn battery diminish (Fig. 4.30).
Figure 4.30. Neutron scattering images of charged (left) and discharged (right) 90 mAh batteries. In the mount are a worn (left) and fresh (right) battery. The neutron scattering autocorrelation length depends on the D grating spacing of 25 mm. Here, $\xi = 2.813 \, \mu\text{m}$.

The fresh battery also has a small reduction in signal, however it is not as noticeable as in the worn battery. One possible explanation for this decrease in signal is that the amount of particles at the 2.813 $\mu\text{m}$ size are more prevalent in the fresh battery than the worn battery. While this could be related to battery degradation within the cell of certain components, it is challenging to specify which component would cause this phenomenon. Having multiple layers of electrolytic and electrode material (graphite, LiCoO$_2$, Al, Cu) do not make it simple to determine what is causing the degradation.

To understand the trend of scattering across the wide range of autocorrelation lengths probed for the fresh and worn batteries, we observe how the signal increases and then sharply decreases when the grating distance reaches a certain point. The dark-field signal (scattering) as a function of grating distance is shown in Fig. 4.31 and as a function of $\xi$ in Eq. 3.16.

In the 43 mAh batteries, large differences in the mean scattering values across the batteries were observed. The figures below exhibit the drastic change in battery particle size as a function of state of number of cycles. Here, the fresh battery has undergone a single cycle to generate the solid-electrolyte interface. The slightly worn battery still has the initial capacity. The worn battery after 1790 cycles still
has a capacity of 38.6 mAh, close to its advertised capacity of 43 mAh. The worn battery is indeed worn, but has not been mistreated.

At a D(G1-G2) spacing of 7 mm (ξ of 788 nm), we can still observe the effects of harmonics from the gratings on the batteries. The harmonics are a partial failure or imperfection in the far-field experimental optics. Figure 4.32 shows this harmonic generation on the batteries in a vertical orientation.

The harmonics begin to disappear around a D(G1-G2) spacing of 10 mm when the gratings move further apart, although slight remnants of them are still observable in the samples. In this image (Fig. 4.33), the worn battery on the left exhibits
a higher degree of scattering than the fresh battery in the middle or slightly worn battery on the right.

![Image of neutron scattering images after the first charge (left) and discharge (right) cycles.](image_url)

Figure 4.33. Neutron scattering images after the first charge (left) and discharge (right) cycles. In the images, a mounted sample consisted of worn (left), fresh (middle), and slightly worn (right) 43 mAh batteries. The D grating spacing of 10 mm corresponds to $\xi$ of 1.125 $\mu$m.

Physical alterations during battery manufacturing could result in some of the scattering changes observed within the same fresh, slightly worn, and worn battery. If more electrode material is manufactured in one battery than another, this would lead to more scattering within that battery. However, the changes seen between the charged and discharged states imply that chemical changes within the batteries are the main cause. While these images do show the spatial locations of scattering centers between the charged and discharged states, it is a challenge to determine what elements and compounds within the battery are responsible. The thought was that Bragg Edge imaging (Figs. 4.25 or 4.26) would provide major $hkl$ peak differences to relate to the spatial image locations here, however this was not the case.

When the D-spacing increased up to 21 mm, a change in the worn battery is observable (Fig. 4.34). During this change, the scattering value becomes more similar to air (closer to 0).

The first sign of deterioration in the discharged batteries is at a D spacing of 23 mm, where the worn battery begins showing widespread inhomogeneous
Figure 4.34. Neutron scattering images of charged (left) and discharged (right) 43 mAh batteries. A mounted sample consisted of worn (left), fresh (middle), and slightly worn (right) batteries in each image. The D grating spacing of 21 mm corresponds to $\xi$ of 2.363 $\mu$m. 

scattering. The slightly worn battery shows small regions of deterioration and the fresh battery still appears homogeneous.

While the change in the worn battery at a D spacing of 21 mm is only slightly noticeable, it becomes more pronounced at 23 and 26 mm (Figs. 4.35 and 4.36).

Figure 4.35. Neutron scattering images of a charged (left) and discharged (right) 43 mAh battery. A mounted sample consisted of a worn (left), fresh (middle), and slightly worn (right) battery in each image. The autocorrelation length depends on the D grating spacing of 23 mm, which corresponds to $\xi$ of 2.588 $\mu$m.

Similar to the case of the 2 mm thick 90 mAh batteries, we observe a large difference in the scattering between charged and discharged 43 mAh batteries at larger scattering lengths.

In the charged batteries, the worn battery has almost no signal in comparison to air while the slightly worn battery has some signal along the edges and the fresh battery is still near homogeneous. The worn battery still possesses scattering along
Figure 4.36. Neutron scattering images of a charged (left) and discharged (right) 43 mAh battery. A mounted sample consisted of a worn (left), fresh (middle), and slightly worn (right) battery in each image. The D grating spacing of 26 mm corresponds to $\xi$ of 2.925 $\mu$m.

the edge, however the middle area of the battery now has scattering values similar to air. In comparing the mean scattering values for charged and discharged cycles, we observe Figs. 4.37 and 4.38.

Figure 4.37. Mean scattering values across the charged 43 mAh Li-po batteries. The fresh battery is shown in blue, the slightly worn in red, and the worn in green.

Of interest in the scattering values is how the signal increases up to a grating spacing of 21 mm for the fresh battery, 16 mm for the slightly worn battery, and 15 mm for the worn battery before dropping. This could indicate that the fresh battery has particles of larger size in comparison to the worn battery as a D space of 21 mm corresponds to scattering from 2.363 $\mu$m and D of 15 mm corresponds
to 1.688 μm. As a battery undergoes degradation from its original capacity, larger particles within the battery are breaking down into smaller particles. The difficult challenge to analyzing this issue is whether the degradation is occurring on the solid electrolyte interface or the electrodes (Li$_{0.5}$CoO$_2$/graphite).

As anticipated, the fresh battery possesses the highest capacity at 39.5 mAh, the slightly worn battery in the middle at 36.7 mAh, and the worn battery the lowest at 36 mAh. Even after 1790 charge/discharge cycles, it is quite remarkable that the “worn” battery still possesses roughly 83% capacity while the fresh battery with only one cycle is at 92%. The variability in manufacturing of electrode and electrolytic material from one battery to the next is quite large knowing that the fresh battery only maintains 90% higher capacity with much less cycling.

Also of note is how the dark-field signal drops below 0 for D-spaces above 29 mm in the worn battery. The dark-field signal should never drop below the value for air, implying that some parameters in this experimental setup need to be altered for future experiments.

Figure 4.38. Mean scattering values across the discharged 43 mAh Li-po batteries. The fresh battery is shown in blue, the slightly worn in red, and the worn in green.
Thin batteries on the order of 0.5 mm had limited scattering value differences between charged and discharged cycles in comparison to the 1.5 mm thick batteries (Fig. 4.39). A portion of the aluminum holder is visible in the neutron image in the center. The image resolution is excellent along the horizontal direction, but poor along the vertical, due to the neutron slit dimensions.

Figure 4.39. Neutron scattering images of the first charge (left) and discharge (right) cycles. In the images, a mounted sample consisted of a worn (left) and fresh (right) 15 mAh battery. The D(G1-G2) grating spacing of 10 mm corresponds to $\xi$ of 1.125 $\mu$m.

There is limited difference in the scattering signal between the charged and discharged state at $\xi$ of 1.125 $\mu$m. A simple explanation is that while this 15 mAh worn battery had over 1228 cycles, it still retained 80% of its original SOC (12 mAh out of the original 15 mAh).

Figure 4.40. Neutron scattering images of the first charge (left) and discharge (right) cycles. In the images, a mounted sample consisted of a worn (left) and fresh (right) 15 mAh battery. The D grating spacing of 20 mm corresponds to $\xi$ of 2.25 $\mu$m.
At a 20 mm grating distance ($\xi$ of 2.25 $\mu$m, Fig. 4.41), there is a noticeable difference in scattering between the charged and discharged states. The overall signal is higher in the discharged states, especially along the edges of the batteries.

Figure 4.41. Neutron scattering images of the first charge (left) and discharge (right) cycles. In the images, a mounted sample consisted of a worn (left) and fresh (right) 15 mAh battery. The D grating spacing of 25 mm corresponds to $\xi$ of 2.8125 $\mu$m.

At a 25 mm grating distance ($\xi$ of 2.8125 $\mu$m, Fig. 4.41), we begin to see the scattering signal drop in both of the fresh and worn batteries. While this drop is slightly observable, it is continued further at longer scattering lengths (Fig. 4.42). Regions of high scattering located in air (to the left and right of the batteries) imply an optical flaw with the interferometer design at long scattering lengths.

Figure 4.42. Neutron scattering images of the first charge (left) and discharge (right) cycles. In the images, a mounted sample consisted of a worn (left) and fresh (right) 15 mAh battery. The D(G1-G2) grating spacing of 30 mm corresponds to $\xi$ of 3.375 $\mu$m.

In comparison to the thicker 43 and 90 mAh batteries, the overall smaller dark-field signal in the 15 mAh batteries can be observed in Fig. 4.43.
There are two features of interest in the 15 mAh battery scattering. The dark-field signal alternates much more than the 43 or 90 mAh batteries between the charged and discharged states. At \( \xi \) values of 3.5 - 4 \( \mu m \) we observe several oscillations of the signal for all four charged/discharged states. When the fresh battery signal increases at 3.75 \( \mu m \) for both the charged and discharged states, the dark-field signal decreases for both states in the worn battery. These oscillations are likely caused by optical inconsistencies in the interferometer at large \( \xi \) lengths.

With another batch of thin 50 mAh batteries (0.5 mm thickness), the scattering signal was anticipated to be lower than in the 1.6 and 2 mm batteries that possess more layering and material. Figures 4.44, 4.45, and 4.46 show the progression of dark field signal as a function of D-spacing. The high intensity of scattering at the bottom of each image is representative of the terminals of the batteries while an aluminum holder is seen in the top center of each image.

At an autocorrelation length of 1.35 \( \mu m \) (Fig. 4.44), there is a large difference in scattering signal between the fresh and worn battery. The fresh battery averages dark-field signal along the edges near 0.50 and in the center at 0.375 while the worn battery is near 0.46 along the edges and 0.28 in the center. As these batteries
Figure 4.44. Neutron scattering image of after the second discharge cycle of the mounted sample consisting of a fresh (left) and slightly worn (right) 50 mAh battery. The neutron scattering autocorrelation length depends on the $D(G_1-G_2)$ grating spacing of 12 mm, $\xi$ of 1.35 $\mu$m.

only have one layer of material, the degradation within the battery can be spatially identified.

Figure 4.45. Neutron scattering image of after the second discharge cycle of the mounted sample consisting of a fresh (left) and slightly worn (right) 50 mAh battery. The neutron scattering autocorrelation length depends on the $D(G_1-G_2)$ grating spacing of 20 mm, $\xi$ of 2.25 $\mu$m.

The signal of the scattering in these 50 mAh batteries turned negative at long grating distances/autocorrelation lengths (Fig. 4.47) which is physically impossible. This indicates an instrument problem at long D spacings. We are reassured that moderate D spacings give valid results.
Figure 4.46. Neutron scattering image of after the second discharge cycle of the mounted sample consisting of a fresh (left) and slightly worn (right) 50 mAh battery. The neutron scattering autocorrelation length depends on the $D(G_1-G_2)$ grating spacing of 25 mm, $\xi$ of 2.813 $\mu$m.

Figure 4.47. Mean scattering values across the fresh and worn, charged and discharged 50 mAh Li-po batteries. The fresh battery is shown in blue and the worn in red.

4.5 Conclusions

The first spatially resolved scattering images for four lithium polymer (Li-po) battery sets of differing thickness (0.5 mm, 1.6 mm, and 2.0 mm) and capacity (15, 43, 50, and 90 mAh) were imaged using X-ray and neutron Talbot-Lau, near-field and far-field interferometry. Despite the worn batteries having over 1500 charge/discharge cycles, the capacities were still above 80%, indicating a well-manufactured battery.

X-ray imaging performed at CAMD utilized the Talbot-Lau and a new interferometry design (near-field) that probed scattering lengths from 20 nm up to 906 nm.
In the 43 mAh batteries, absorption and dark-field images revealed two layers of battery materials with locational scattering between charged and discharged states.

Neutron diffraction performed on a Li-po battery revealed a decrease in LiC\textsubscript{6} and LiCoO\textsubscript{2} peak intensity from a fresh and a worn battery. A statistical analysis method (Principal Component Analysis/ Multivariate Curve Resolution) was attempted to separate battery components using a synthetic dataset and correlate to the diffraction data. While a 7-component PCA system did not behave as anticipated, a 3-component system showed a decrease in C\textsubscript{6} concentration and an increase in LiC\textsubscript{6} concentration as a function of state of charge. The possibilities of separating battery components is dependent upon known crystallographic and neutron diffraction information, some of which is unknown for lithium graphite compounds like LiC\textsubscript{12}, LiC\textsubscript{18}, and LiC\textsubscript{24}.

Neutron Bragg edge imaging performed on 2 mm thick 43 mAh batteries revealed changes between the fresh and worn, charged and discharged states. The worn batteries always possessed a higher attenuation signal than the fresh batteries. The most logical explanation is that during lithium intercalation when a battery is charged, the battery swells up to 15% of it’s original volume. When a battery is worn, there is less lithium to migrate between electrodes, thus reducing the signal.

Neutron Talbot-Lau interferometry showed spatial differences in the scattering signal at the 1.97 \( \mu \)m scattering range. Vertical features of high scattering in a charged and discharged battery were observed in 0.5 mm thick 15 mAh batteries. However, with only one layer of material in the battery, the signal was relatively weak in comparison to the 2 mm thick batteries.

In a new far-field interferometry experiment, the autocorrelation length, \( \xi \), was probed from 600 nm up to 4.5 \( \mu \)m. The scattering changes dramatically as a function of grating distance D(G1-G2) and autocorrelation length, \( \xi \), between worn
and fresh batteries. While it is unknown as to what is causing this phenomenon, the dark-field images show locations of high scattering near the edges of all batteries.

The difference between fresh and worn batteries is seen in the evolution of the average scattering with $D(G1-G2)$ values. When the $G1-G2$ distance was small (between 3 - 10 mm), harmonics generated from the gratings were projected onto the battery images. At large grating distances (above 30 mm), the dark-field signal turned negative in all batteries, indicating a design flaw in the setup. To understand the difference between fresh and worn batteries, the average scattering signal was plotted as a function of grating distance. The plots for the 43 mAh batteries show a maximum scattering signal for the worn battery at 18 mm, the slightly worn battery at 24 mm, and the fresh battery at 27 mm.
References


Chapter 5
Porosity in Additive Manufacturing

5.1 Introduction

Additive manufacturing (AM) has become a buzzword within the past two years. From the building of 3D printed houses and cars to rethinking battery design [1], the goal of 3D printing is to build materials used in everyday life on a much cheaper scale. With little known about the stability of 3D printed parts, questions arise as to whether a million dollar AM printer provides more value than a $5,000 printer. Advantages of AM include the ability to build complex shapes unobtainable by conventional printing [2], the re-usability of material, and a lack of a tool access requirement [3]. Issues with the printers, materials, or software import files all play a major role in validating the AM printing process.  

To print an object by additive manufacturing, a computer-aided design (CAD) file for an object is uploaded and converted into a stereolithography (STL) file [4]. During this conversion, the object is cut into slices (similar to the slices from a tomographic volume) to give information regarding thickness and resolution about each layer. At this point, an electron beam, laser, polyjet, or other energy source is used to physically print material on a layer by layer basis. Optimization of different printing parameters such as laser power, scan speed, and hatch spacing are important in determining how the microstructure of the resulting builds are altered from differing prints [2, 5]. Figure 5.1 shows the wide range of printing

---

1This chapter previously appeared as Brooks et al. 2017, “Porosity Detection in Electron Beam Melted Ti-6Al-4V using High-Resolution Neutron Imaging and Grating-Based Interferometry”, and is available at Springer http://dx.doi.org/[10.1007/s40964-017-0025-z]. It is reprinted by permission of Springer International Publishing Switzerland.
Figure 5.1. Flow chart showing the many printing methods possible. Of focus in this work are Electron Beam Melting and Selective Laser Melting.

methods [4]. Herein, samples fabricated with electron beam melting (EBM) will be discussed.

EBM uses a metal powder bed fusion AM technology to print objects generated from computer models in 3D [6]. During EBM printing, a metallic powder – roughly 50 μm diameter – layer is laid out and selectively melted by an electron beam formed from a tungsten filament [6]. Once one layer is complete, another batch of powder is dispensed and a pattern created by the computer is formed from the beam. When the electrons interact with the powder particles, the kinetic energy of the electrons is transferred to the powder dependent upon the size and depth of penetration of the particles [7, 8]. EBM systems are controlled under vacuum such that the electrons are not reflected or interact with random atoms in the residual atmosphere [9].
The three main types of additive manufacturing are ceramics, metals, and polymers. Polymer AM has designed and built cars and houses on large scales. One interesting feature missing from these prints are flame retardants, which can help reduce the amount of char layers or bubbles in a polymer material [10, 11]. Ceramic AM could play a key role in the printing of piezo-electro ceramic materials for medical imaging or in sand casting molds for mining, oil, and gas industries [12]. Metal AM shows promise in the fabrication of precision machines and implants as a more resourceful method than conventional machining [13].

Metallic printing of titanium alloys are used in a wide variety of applications, from manufactured pores on the surface of orthopedics [14] to dental implants [15] to bone implants [16]. Advantages of Ti alloys include great tensile strength, light weight, heat treatability, and high corrosive resistance [14, 17, 18]. However, disadvantages of Ti-6Al-4V involve lack of fusion, trapped gas, and keyhole porosity that affect the microstructural properties of the material [19, 20, 21].

Of interest in this chapter is the observation of porosity in AM Ti-6Al-4V alloys. μXCT imaging provides an easy way to view pores, cracks, and potential defects in AM parts using small volumes [20]. Synchrotron X-ray radiography of 2mm thick titanium AM dogbones observes the initiation and propagation of cracks during fretting fatigue [22]. While this shows the benefits of X-ray imaging, sample thickness becomes a factor due to the high absorption of X-rays by metals. It is quite impractical to use a 400-800 keV X-ray source when neutrons provide bulk sensitivity of metal samples rather easily. For this reason, neutron imaging is the focus of this chapter.

5.2 Experimental
Grating-based interferometry and traditional neutron radiography experiments of Ti-6Al-4V cubes were performed at the HZB-CONRAD2 beamline [23, 24]. Ti-
Figure 5.2. Image of the 1.5 cm$^3$ Ti-6Al-4V EBM cubes printed at ORNL

Figure 5.3. Visibility calculation of the neutron beam at HZB. Figure was made using Mathematica.

6Al-4V cubes were EBM printed at the Manufacturing Demonstration Facility at ORNL and can be seen in Fig. 5.2. Raw data for the tomography datasets was converted to projections then sinograms using Mathematica. A sample calculation of the neutron beam visibility can be seen in Fig. 5.3. Reconstruction was done using the ASTRA toolbox [25] simultaneous iterative reconstruction technique (SIRT). Visualization was performed on the volumes using Avizo.
5.3 Ti-6Al-4V Paper

Attached is a copy of a recently accepted paper titled “Porosity Detection in Electron Beam Melted Ti-6Al-4V using High-Resolution Neutron Imaging and Grating-Based Interferometry”. “The final publication is available at Springer via http://dx.doi.org/[10.1007/s40964-017-0025-z]”
Porosity detection in electron beam-melted Ti-6Al-4V using high-resolution neutron imaging and grating-based interferometry

Adam J. Brooks1 · Jinghua Ge2 · Michael M. Kirka3 · Ryan R. Dehoff3 · Hassina Z. Bilheux4 · Nikolay Kardjilov5 · Ingo Manke5 · Leslie G. Butler1

Received: 21 December 2016 / Accepted: 28 April 2017
© Springer International Publishing Switzerland 2017

Abstract A high-resolution neutron tomography system and a grating-based interferometer are used to explore electron beam-melted titanium test objects. The high-resolution neutron tomography system (attenuation-based imaging) has a pixel size of 6.4 μm, appropriate for detecting voids near 25 μm over a (1.5 cm)³ volume. The neutron interferometer provides dark-field (small-angle scattering) images with a pixel size of 30 μm. Moreover, the interferometer can be tuned to a scattering length, in this case, 1.97 μm, with a field-of-view of (6 cm)³. The combination of high-resolution imaging with grating-based interferometry provides a way for nondestructive testing of defective titanium samples. A chimney-like pore structure was discovered in the attenuation and dark-field images along one face of an electron beam-melted (EBM) Ti-6Al-4V cube. Tomographic reconstructions of the titanium samples are utilized as a source for a binary volume and for skeletonization of the pores. The dark-field volume shows features with dimensions near and smaller than the interferometer auto-correlation scattering length.

Keywords Neutron interferometry · Electron beam melting · Additive manufacturing · Porosity · Tomography

1 Introduction

Ti-6Al-4V additive manufacturing (AM) has played a key role in modern scientific exploration, from biocompatibility studies with mouse fibroblast cells [1] to dental implants [2]. As the role of AM grows larger, so does the need for understanding the effects of printing methods on materials. A recent model comparing the differences between electron beam melting and selective laser melting of Ti-6Al-4V serves as a guide as to how the microstructure can be controlled through AM [3]. While modeling may be
valuable, it does not reveal any information as to whether a desired manufactured object was printed correctly.

One simple way to look at this problem is through imaging. X-ray and neutron imaging provide a way to visualize 3D printed objects without the need for destruction of samples. Traditional imaging involves 2D radiography, where a wavefront from a source (X-ray or neutron) travels in a designated path and interacts with a sample placed in the beam path at a certain distance. Alterations to the wavefront are detected through attenuation/absorption images and the object can then be reconstituted.

In comparison to the one dataset provided from attenuation-based radiography, grating-based interferometry generates three datasets: traditional attenuation/absorption, differential phase contrast, and dark-field (synonymous with small-angle scattering). With many different methods of X-ray and neutron interferometry developed over the years (Michelson, Talbot-Lau [4], and Far-Field [5]), of interest here is the Talbot-Lau interferometer. In a Talbot-Lau interferometer, a source grating, G0, adds coherence to an incoming wavefront (either X-rays or neutrons). The wave proceeds along the beam path and interacts with a phase grating, G1, which alters the wave by a certain period that is then propagated further downstream. Intensity changes of the beam can be detected using an analyzer grating, G2, after an object is placed in the beam path.

While interferometry is a well-known technique, one recent hindrance was grating fabrication. Until 2006, grating fabrication was only well established for X-ray interferometry (X-ray lithography) [6]. Challenges with gadolinium sputtering for source and analyzer gratings limited the production of neutron gratings [7] until Kim/ Lee began filling silicon gratings with Gadox powder to produce the same optical properties [8]. Interferometry has now become a valuable tool for neutron imaging of materials.

With two main types of imaging (X-rays and neutrons), questions arise as to which is the more viable imaging option for AM materials. X-ray computed tomography (XCT) has proven valuable for observing polymer AM pores [9]. Micro X-ray CT (µXCT) offers very high spatial resolution (as low as 5 µm) for observing Ti-6Al-4V lack of fusion, keyhole, and gas porosity [10, 11]. While higher spatial resolution is preferred, the high absorbance of X-rays by metals and a small sample size requirement for µXCT limits the reproducibility of imaging large-scale samples. In comparison, neutron imaging provides bulk sensitivity and can show strain distribution in Inconel 625 [12]. In the case of Ti-6Al-4V samples with micron-sized spacing, the interaction of neutrons with the metal results in an expected phase shift of the beam and an altered signal in the dark-field. In comparison to attenuation imaging, where resolution is limited down to 15 microns, the dark-field signal can reach as low as 1-micron resolution [13].

The possibility exists for one imaging method, such as attenuation, to measure defects in AM samples undetected with another imaging modality. Due to the chance of porosity defects occurring at multiple length scales (nanometer and micrometer), the need exists for performing high-resolution attenuation imaging and grating-based interferometry. The interferometry can detect changes in the small-angle scattering (dark field) while high-resolution imaging can magnify noticeable features through attenuation. For a several cm thick AM Ti-6Al-4V sample, bulk properties regarding the shape, size and location of porosity defects can be detected on the micron scale.

2 Experimental

2.1 Additive manufacturing- ORNL MDF 3D printing of the titanium cubes

Samples of interest were electron beam-melted (EBM) in 2015 and 2016 from Ti-6Al-4V gas atomized powder. The build setup is shown in Fig. 1.

The powder is estimated to have a diameter ranging between 45 and 150 µm. Composition of the Ti-6Al-4V includes a majority titanium (90%), aluminum (6%), and vanadium (4%), with trace amounts of iron, nitrogen, carbon and oxygen. An Arcam Q10 EBM machine printed layers from used powder in 0.05 mm height increments. A symbol, such as M3 or M5, is incorporated into one face of each cubic sample to distinguish samples. The beam spot size was not measured. For batch 4.2.75, the speed function of the Arcam Q10 system is 46 mm/s and the focus Offset is 32 mA. For batch, 4.2.87, the speed function is 30 mm/s and the focus offset is 15 mA. Hatch spacing of the system is 0.2 mm. After the EBM printing is complete, the nonsintered powder is kept under a vacuum atmosphere, so that it can be used in future builds. Printing parameters for Ti-6Al-4V samples are shown in Table 1 below. Similar AM printing procedures can be seen in [14–16].

2.2 HZB CONRAD-2 imaging beamline

Grating interferometry and high-resolution neutron imaging experiments were performed at the HZB BER II reactor neutron imaging instrument (CONRAD-2) [17, 18]. High-resolution imaging was performed without grating interferometry due to absorption of the beam by the gratings and to achieve higher resolution. A sample setup of the grating-based interferometry is shown in Fig. 2.
The beamline operated a pinhole of 3 cm, allowing sufficient neutron flux \(2.4 \times 10^7 \text{n cm}^{-2} \text{s}^{-1}\) at the detector (5 m, \(L/D\) value of 167). An Andor Ikon L-936 camera (2048 \times 2048 pixels, pixel size 30 \(\mu\text{m}\)) was used for imaging. The effective pixel size was 6.4 \(\mu\text{m}\) for high-resolution imaging and 30 \(\mu\text{m}\) for interferometry. A 100 \(\mu\text{m}\) thick \(\text{LiZnS:Ag}\) scintillator was used to convert neutron flux into visible light.

In the grating interferometry setup, the source (G0), phase (G1) and analyzer (G2) gratings were placed at their respective first Talbot order distances. Grating position, alignment, and stability is critical for maintaining interferometer performance, as measured in percent visibility of a moiré pattern at the detector. In this experiment, the gratings were placed horizontal to the neutron beam to observe horizontal microstructure sensitivity in the samples. Alignment of the gratings began by observing the neutron beam with G0 (period of 790 \(\mu\text{m}\)) and G2 (period of 4 \(\mu\text{m}\)), where horizontal moiré fringes were observed. G2 was aligned close to the detector to eliminate high-angle scattering from the sample. During the initial setup and alignment, G1 was the last grating to be installed; the G0, G2 gratings and the detector were mutually aligned to give a low-frequency moiré pattern. The interferometer was operated in a stepped-grating mode, typically with twelve G1 positions evenly spaced over 12 \(\mu\text{m}\), slightly more than the 7.96 \(\mu\text{m}\) G1 period.

Interferometry and tomography imaging of two Ti-6Al-4V cubes was performed with 20 s exposures, with one sample from 2015 (M5) and one from 2016 (M3). High-resolution attenuation tomography was performed on the 2016 cube, M3, after porosity was discovered with the gratings. An exposure time of 6 s provided good visibility (5\%) over 500 image projections. For tomography experiments, 181 steps from 0\(^\circ\) to 360\(^\circ\) were used.

### 2.3 Neutron interferometry analysis

A vectorized least squares algorithm was used for processing the raw images of the stepped-grating interferogram into projections of absorption, differential phase contrast, dark-field, percent visibility of the reference image, and \(\chi^2\) of the sample images [19]. Across most of the sample interferograms, \(\chi^2\) averaged near 2, indicating an acceptable fit. The interferometer visibility in the reference images averaged 5\% over the field of view, with the exception of a few damaged regions in the gratings. For regions in the reference visibility image with less than 3\%, a mask was generated and used to guide an inpainting correction to the projection [20]. Inpainting has characteristics similar to a median filter, but when guided by a visibility mask, it offers a more targeted image correction.

The tomography projections for attenuation and dark field were reconstructed with SIRT (ASTRA toolbox) [21]. The dark-field projections were reconstructed as \((1 - \text{dark-field})\) to set the air-region around the sample to zero. The reconstructed differential phase contrast volume was found to be essentially featureless. With some tomography runs producing 2534 projections, a dataset could result in well over
17 GB. Binning of 2 or 4 was originally utilized for faster image processing. Cropping resulted in projections of 1.8 MB per image, or 4.7 GB per dataset.

2.4 Visualization

Three volumes were obtained and compared pairwise. Two volumes show neutron small-angle scattering. The attenuation volume with pixel size of 6.4 μm is regarded as the benchmark for feature detection; this volume is labeled A6.4. The attenuation volume collected simultaneously with the dark-field image has a pixel size of 60 μm after binning of 2 and is labeled A60. The dark-field volume is labeled DF60.

Control points in the A6.4 and A60 volumes were manually selected in Avizo and used to generate a rigid-body affine transformation for the A60 and DF60 volumes. The high image contrast in A6.4 supported segmentation and morphological component analysis to provide a label field of the porosity structure (not shown). This label field was visualized as a skeletonization, Fig. 5.

3 Theory

To properly analyze the raw attenuation and dark-field datasets for the AM cubes, theory regarding the properties of materials and imaging must be understood. Attenuation images detect the sample composition based on neutron scattering and absorption. The dark-field image has a high connection with small-angle neutron scattering (SANS) and shows the microstructure of samples. The interferometer can be tuned to an auto-correlation length, ζ, and the dark-field image can show the effect of features with dimensions at or smaller than ζ. The theory behind quantitative neutron dark-field imaging has recently been published [22, 23]. A condensed explanation is presented here.

The real component of the neutron refractive index, n, is determined by:

\[ n = 1 - \frac{j^2 N b_c}{2\pi} = 1 - \frac{j^2}{2\pi} (\text{NS LD}), \tag{1} \]

where λ is the neutron wavelength, \(N_b\) is the number density, \(N_{b_c}\) is the coherent neutron scattering length, and NS LD is the neutron scattering length density as tabulated by NIST.\(^1\)

The dark-field intensity (DFI) is dependent on the visibility of the beam, showing how intensity changes of the beam are generated by the phase grating, G1.

\[ V = \frac{I_{\text{max}} - I_{\text{min}}}{I_{\text{max}} + I_{\text{min}}} \tag{2} \]

When a sample is introduced into the beam path, modulations to the beam intensity can be detected at specific Talbot distances. The dark-field signal is a determination of sample visibility, \(V\), over open beam visibility, \(V\), as shown in Eq. 3.

\[ \text{DFI} = \frac{V}{V_{\text{max}}} \tag{3} \]

DFI is dependent upon several factors: period of the phase grating, \(p_2\), sample-to-detector distance, \(L_S\), and auto-correlation length, \(\zeta\).

\[ \zeta = \frac{\lambda \times L_{S_{\text{eff}}}}{p_2} \tag{4} \]

To include sensitivity changes in the scenario of a sample being placed in front of G1 in the grating interferometer, the real sample-to-detector distance, \(L_{S_{\text{eff}}}\), must be accounted for. \(L_1\) is the G0-to-G1 distance and \(L_2\) is the G1-to-G2 distance. A large sample-to-detector distance implies the scattering angles probed are larger than those if the sample was placed close to the detector.

\[ L_{S_{\text{eff}}} = \frac{(L_1 + L_2 - L_S) \times L_2}{L_1} \tag{5} \]

For this experiment, \(\lambda = 3.5 \, \text{Å}, L_S = 5 \, \text{cm}, L_{S_{\text{eff}}} = 2.257 \, \text{cm},\) and \(p_2 = 4 \, \mu\text{m},\) giving a calculated autocorrelation scattering length of \(\zeta = 1.97 \, \mu\text{m}.\) The dark-field image gains intensity for features of this size or smaller.

4 Results and discussion

4.1 Grating interferometry of Ti-6Al-4V cubes

From the grating interferometry experiment, it was discovered that one of the AM cubes was not as uniform as expected. In both the attenuation and dark-field projections, a pore-like structure was observed in the 2016 cube (M3) but not in the 2015 cube (M5). Figure 3 shows the change in sample uniformity from 2015 to 2016 in attenuation and dark-field slices. In principle, the attenuation image can be used to measure the material density in the pores through the composition-weighted attenuation values, \(\Sigma\) (see Table 2) and the dark-field can measure pore sizes through Eq. 4, the autocorrelation scattering length.

In determining the quality of the data, histograms revealed a slightly higher degree of correlation in the uniform cube (Fig. 4). While the porous cube has a similar range of attenuation values from 0.0028 to 0.0038, the dark-field values are more obscure in the defective cube as

\(^1\) NIST Neutron activation and scattering calculator; http://www.ncnr.nist.gov/resources/activation/.
seen by the lighter shade of orange above values of 0.001. For the defective cube, the dark-field signal is much broader across the full attenuation range as compared to the concentrated intense signal in the good cube. This could indicate that even though the pores are microns in size (as compared to the 1.5 cm sample size), they greatly reduce the amount of sample scattering to behave more similarly to the environment.

4.2 High-resolution imaging

The main benefit of high-resolution attenuation imaging is obtaining roughly twice the resolution compared to grating interferometry. With an increase in effective pixel size from 30 to 6.4 μm, the pore structure of an AM sample can undergo further evaluation through segmentation and skeletonization. Skeletonization was found to be a valuable tool for observing porosity, where the connectivity of pores relates to the design parameters of the material. Figure 5 shows a skeletonization of the pore structure, ranging from radius sizes of 5–50 μm.

Once a skeletonization showed the pore structure, it was necessary to visualize this structure in the sample. Figure 5 (right) shows the combination of skeletonization with the porous M3 cube. One great feature is the ability to observe the M3 symbol on the cube, indicating which face hosts the defects. In this case, the bottom face hosts the chimney pore structure and shows how the pores build vertically across the sample. This information is highly valuable to the AM field to help alleviate any potential porosity issues during the printing or relaxation process.

Since the pore structure became more pronounced with high-resolution imaging, a new challenge arose to register Ti-6Al-4V cube volumes from differing experiments. With three known registration options (auto, manual, and landmarks), the best choice for the cubes was to utilize segmented data with correlation. The resulting Fig. 5 (left) shows a successful registration from the grating interferometry attenuation volume with the high-resolution attenuation volume. The two bounding boxes indicate the transformation applied to the interferometry dataset to register with the high-resolution dataset. The porous volume rendering of the high-resolution data is in white with the interferometry in yellow.

4.3 Pore structure

The attenuation and dark-field images provide complementary information about the pore structure. The attenuation image can have a high contrast to noise ratio, which allows binarization, segmentation, and skeletonization. The skeletonization defines the internal pore structure, yet makes the erroneous assumption that the pore/matrix interface is

Table 2 Ti-6Al-4V composition, thermal neutron scattering, absorption, and attenuation lengths (ρ = 4.43 g/cm³)

<table>
<thead>
<tr>
<th>Element</th>
<th>wt%</th>
<th>Scattering (barn)</th>
<th>Absorption (barn)</th>
<th>Weighted attenuation (cm⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ti</td>
<td>90</td>
<td>4.35</td>
<td>6.09</td>
<td>0.524</td>
</tr>
<tr>
<td>Al</td>
<td>6</td>
<td>1.503</td>
<td>0.231</td>
<td>0.010</td>
</tr>
<tr>
<td>V</td>
<td>4</td>
<td>5.1</td>
<td>4.49</td>
<td>0.020</td>
</tr>
<tr>
<td>Σ(cm⁻¹)</td>
<td></td>
<td></td>
<td></td>
<td>0.554</td>
</tr>
</tbody>
</table>
discrete. The dark-field image more accurately shows the range of porosity at the pixel-by-pixel level.

Shown in Fig. 6 is a pore observed with both interferometry and high-resolution attenuation imaging. With the high-resolution imaging, the pore has been segmented, skeletonized (Fig. 5) and converted to a label field. The dark-field image, Fig. 6d, shows the complicated structure in a pore. As the dark field is synonymous with small-angle scattering, brighter features are those regions exhibiting reduced attenuation combined with increased neutron scattering.

Fig. 4 Histograms of the M3 Ti-6Al-4V cube (A60 and DF60) for the portion with defects (a) and no defects (b). Defects are observed as reduced attenuation combined with increased neutron scattering.

Fig. 5 On the left, the grating interferometry attenuation volume (yellow, A60) is registered to the high-resolution (white, A6.4) attenuation volume. Average values for the interferometry attenuation range from 0–0.00295 to 0–0.00119 for the high-resolution volume. On the right, a skeletonization of the pore structure from A6.4. The pore segments are depicted in tubes scaled and colored by thickness (thickness min–max: 0.5–12.1, colormap: blue to red) (color figure online).
strong scattering at the autocorrelation scattering length set by the interferometer.

Attenuation and dark-field reconstructions of the 2016 M3 sample exhibit a high amount of overlap in the chimney-like structure. Still, the dark field shows evidence of the feature outside the regions detected by attenuation. This suggests two different levels within the features. The larger chimney-like structures are highlighted by both attenuation and dark field.

The voids were originally treated as spherical; however, the discovery of the chimney structure helped to determine that the porosity is either a result of a lack of fusion or trapped gas. With the composition of Al rather small (only 10%) in Ti-6Al-4V, it is difficult to determine if the pores are composed of aluminum or if the pores are truly voids in material. Pure aluminum could have attenuation values low enough to show up as a pore (as calculated in Table 2). However, the dark-field volumes disagree with this hypothesis as there is scattering from the pore structure at different locations than the attenuation volume. This likely indicates that the observed pores are a result of a lack of fusion (irregularly shaped pores) rather than keyhole or trapped gas (spherical voids). The attenuation values in the histograms in Fig. 4 are also above 0, indicating that any voids are likely composed of material rather than gas.

5 Conclusions

A Ti-6Al-4V AM cube was imaged using neutron grating interferometry and high-resolution imaging. Synergistic analysis of high-resolution neutron tomography is achievable with lower resolution neutron interferometry/tomography. The dark-field image from neutron grating interferometry reveals sub-pixel scattering features at near micron dimensions. A comparison of neutron attenuation versus dark-field intensity shows that the chimney-like features attenuate neutron flux nearly as much as the bulk, but contain efficient neutron scattering sites with an autocorrelation length of 2 lm. There are two possibilities for the scattering differences: small gas-filled pockets present in the sample are less than 2 lm or micron-scale phase separation of aluminum and titanium occurs in the material. If phase separation is causing the difference in scattering, the reduced neutron attenuation in the chimney-like features suggests this is due to aluminum enrichment within the chimney. A destructive test, like scanning electron microscopy, would show the true composition of the pores; however, further imaging experiments of the sample could no longer be explored.

This work shows one application of neutron grating interferometry for non-destructive evaluation of AM. Unfortunately, we note the HZB reactor is scheduled to close in 2020. The forthcoming VENUS beamline at the
Spallation Neutron Source, Oak Ridge National Laboratory, USA could be an excellent site for grating interferometry.

Acknowledgements AIB and LGB gratefully acknowledge support from the Louisiana Consortium for Neutron Scattering (LaCNS), funded by the US Department of Energy- EPSCoR Cooperative Agreement No. EPS-3003897 and the Louisiana Board of Regents. MMK and RRD are supported by the Additive Manufacturing Demonstration Facility of Oak Ridge National Laboratory and by the US Department of Energy, Office of Energy Efficiency and Renewable Energy, Advanced Manufacturing Office, under contract DE-AC05-00OR22725 with UT-Battelle, LLC. HZB is supported by the DOE Office of Science under contract DE-AC05-00OR22725 with UT-Battelle, LLC.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References


5.4 Conclusions

Porosity in Ti-6Al-4V Electron Beam Melted cubes was discovered with neutron Talbot-Lau grating interferometry and high-resolution neutron imaging. This is the first application of grating-based interferometry to additively manufactured samples. Once features of interest were observed in the attenuation and dark-field images of the grating-based interferometry experiments (60 μm resolution), high resolution tomographic imaging (30 μm resolution) was performed to zoom in on the regions. The regions were found to change from slice to slice so visualization was performed using Avizo. A chimney-like pore structure was found in a 2016 printed cube while a 2015 printed cube was homogeneous.

The chimney was discovered to have locations of varying thickness propagating from the first level of the build to the middle of the sample. This discovery implies that altering parameters during the printing process can cause defects during printing. The pores showed up in both the attenuation and dark-field datasets, however there was a difference in locations of these pores. With the scattering length of the Talbot-Lau interferometer set to 2 μm, this indicated that the scattering was caused by gas-filled pores or phase separation occurred during the printing process. The hypothesis is that the pores are attributed to a lack of fusion during the printing process rather than gas due to the irregular shape of the chimney. Spherical (keyhole) pores are likely to be caused by gas and that was not the case in this experiment.
References


Chapter 6
Fatigue in SS316 Additive Manufacturing

6.1 Introduction

Another area of focus in additive manufacturing besides porosity is fatigue. Materials such as magnesium alloys are attractive components in airplanes and transportation vehicles due to their light weight and high machine strength [1, 2]. Unfortunately, tension created from cyclic loading can introduce crack formation and propagation throughout a sample [1]. In the case of fatigue of aircraft parts, with estimates of aircraft flying roughly 3,000 flying hours in a year, safety regulations regarding design, maintenance, and damage tolerance are necessary [3]. While models to understand fatigue propagation under multi-axial loading have been developed [4], prediction of fatigue is needed to prevent catastrophic incidents such as the crash of El Al flight 1862 [5] or the collapse of the Minneapolis Mississippi River bridge in 2007 [6].

One such method to observing the formation of cracks during the fatigue process is through X-ray and neutron imaging and diffraction. Neutron Bragg edge transmission imaging has shown microstructural changes in steel plates due to plastic bending deformation [7].

In this chapter we focus on marine grade stainless steel, SS316, which can be used for fuel cells [8], piping material in the food and drink industry [9], and piping material for water and gas distribution [10]. Low cycle fatigue tests on SS316 found that cracks originating at the surface were trans-granular while an increase in temperature resulted in intergranular cracks [11]. The two common phases of stainless steel are austenite and martensite, where the softer austenite phase can transform into a harder martensite phase after deformation or strain-induction of the mate-
Neutron Bragg edge imaging is also able to observe this phase transition from austenite to martensite [13]. In austenite, the formation of δ-ferrite can prevent cracking at high temperatures, limit segregation, and remove the chance of melting at low temperatures [14, 15].

Selective Laser Melting (SLM) uses a high-energy laser beam to melt evenly spread powder across the build area [16]. This melting process takes place in an inert atmosphere (Ar or N₂) to reduce the chance of oxidation or degradation of the materials [17]. SLM can be used with titanium alloys [14, 18], steel [19, 20], aluminum [21, 22], and bronze [23]. The builds of SLM parts often possess high strength due to grain size and dislocation density leading to strengthening of the material [19]. A large benefit of SLM printing includes a reduction in atmospheric emissions due to the reuse of powder materials from build to build without major differences in the properties of materials [22].

The following describes results obtained from neutron interferometry and Bragg edge imaging experiments at ORNL, HZB and NIST to observe fatigue in SLM printed SS316 materials. The goal is to observe fatigue of SS316 AM and conventionally machined samples as a function of fatigue. Differences between AM and conventional samples are anticipated to be minimal. The more tension or pulling on a sample, the easier it should be to view crack formation as a function of fatigue.

### 6.2 Experimental

The following sections describe the printing process and neutron imaging of 2 and 3 mm SS316 dogbone samples. Two neutron interferometry methods (Talbot-Lau and far-field) were utilized to observe crack formation in the dogbone samples. Neutron Bragg edge imaging was also explored as a method to understand phase information of the SS316.
6.2.1 Selective Laser Melting SS316 at LSU

Selective laser melting (SLM) was used to additively manufacture 2 mm (Fig. 6.1) and 3 mm (Fig. 6.2) thick dog-bones at LSU Mechanical Engineering. 3 mm dog-bones were also made from conventional stainless steel plates as a comparison. In the SLM printing, the process uses a laser to selectively melt a bed of spherical metallic powder approximately 20 $\mu$m thick. Each layer corresponds to a cross section of the desired parts. A Concept Laser Mlab cusing R machine was used for part manufacturing. The material used was Concept Laser CL 20ES, an austenitic stainless steel powder following formulations for grade 316L, and the machine’s built-in processing parameter set was used. The neutron attenuation for SS316L (CL 20ES) is calculated from the NIST database and listed in Table 6.1.

Tensile testing on 2 mm dog-bone samples was performed at LSU Mechanical Engineering with a 5,000 N mechanical tester with a strain rate of $1 \times 10^{-4}s^{-1}$. Dog-bones were left in their pristine state, pulled to 75% elongation (referred to...
Figure 6.2. Image of a pristine AM (top) and conventional (bottom) 3 mm thick SS316 sample. Dimensions of the samples are 16.09 mm (length) by 3 mm (width/thickness) by 110 mm (height).

Table 6.1. SS316 (CL 20ES) Elemental Composition and Corresponding Neutron Attenuation

<table>
<thead>
<tr>
<th>Element</th>
<th>Wt % Comp. (min-max)</th>
<th>$\sigma (cm^{-2})$</th>
<th>$\mu_{\text{min}} (cm^{-1})$</th>
<th>$\mu_{\text{max}} (cm^{-1})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fe</td>
<td>(balance: 62.895 - 71.5)</td>
<td>$1.416 \times 10^{-23}$</td>
<td>0.776</td>
<td>0.861</td>
</tr>
<tr>
<td>Cr</td>
<td>16.5-18.5</td>
<td>$6.54 \times 10^{-24}$</td>
<td>0.098</td>
<td>0.113</td>
</tr>
<tr>
<td>Ni</td>
<td>10-13</td>
<td>$2.30 \times 10^{-24}$</td>
<td>0.186</td>
<td>0.248</td>
</tr>
<tr>
<td>Mo</td>
<td>2.0-2.5</td>
<td>$8.19 \times 10^{-24}$</td>
<td>0.008</td>
<td>0.010</td>
</tr>
<tr>
<td>Mn</td>
<td>0 - 2.0</td>
<td>$1.545 \times 10^{-23}$</td>
<td>0</td>
<td>0.027</td>
</tr>
<tr>
<td>Si</td>
<td>0 - 1.0</td>
<td>$2.338 \times 10^{-24}$</td>
<td>0</td>
<td>0.004</td>
</tr>
<tr>
<td>P</td>
<td>0 - 0.045</td>
<td>$3.484 \times 10^{-24}$</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>C</td>
<td>0 - 0.030</td>
<td>$5.555 \times 10^{-24}$</td>
<td>0</td>
<td>0.001</td>
</tr>
<tr>
<td>S</td>
<td>0 - 0.030</td>
<td>$1.556 \times 10^{-24}$</td>
<td>0</td>
<td>0.000</td>
</tr>
</tbody>
</table>

$\Sigma (cm^{-1})$ | 1.068 | 1.264 |
henceforth as half-life), or pulled to fracture. In the half-life sample, elongation of the material was discovered to be around 5% with tensile strength over 570 MPa.

6.2.2 Neutron Imaging at ORNL-CG1D

Traditional radiography and tomography was performed at ORNL CG1D on a thin SS316 sample (dimensions 60 mm height x 12 mm width x 2 mm thick) with a manufactured crack. The sample was imaged, then bent 300 times using a set of pliers, imaged, bent again 300 more times and finally imaged again. The instrument operated with a diffuser made of 50 nm Al$_2$O$_3$ nanoparticles of 0.2 cm thickness, an aperture size of 8.2 mm, and a $^6$LiF/ZnS scintillator. An Andor CCD detector was optically coupled to a 7.5 x 7.5 cm scintillator. The position of the sample along the beampath was 189.7 mm with a sample to detector distance of 5.5 cm. Tomography experiments were performed with 90 second exposures from 0 - 183$^\circ$ in increments of 1$^\circ$. Radiography experiments used 3 images of 200 second exposures.

6.2.3 Neutron Talbot-Lau Interferometry at HZB

Talbot-Lau grating interferometry and Bragg edge imaging experiments of 2 and 3 mm thick SS316 AM and conventional samples were performed at the HZB-CONRAD2 beamline [24, 25]. Grating interferometry of 2 mm and 3 mm SS316 samples was performed with gratings in the vertical orientation. 2D and 3D datasets were collected, with radiography of the 3 mm samples using 20 second exposures and the source grating, G0, stepped from 0 - 1.2 mm over 14 steps with 200 projections. Three tomography datasets of the 2 mm samples were collected: one with the samples in vertical orientation, two with the samples horizontal. Exposure times of 15 seconds provided enough flux through the thinner samples. Bragg edge imaging was performed with monochromatic beam, 100 second exposures from 2 - 4.5 Å by steps of 0.02 Å, and 2 x 2 binning.
Raw data for the tomography datasets was converted to projections then sinograms. Reconstruction was done using Tomopy’s simultaneous iterative reconstruction technique (SIRT). Visualization was done using Avizo. In ImageJ, the amplitude and offset for the sample and flatfield projection images are imported (File/Import/ImageSequence). A z-project with a median of 2 is applied so only one image remains (Image/Stacks/Z-project/median), followed by division of the amplitude by the offset AmplitudeSamp/OffsetSamp and AmplitudeRef/OffsetRef (Process/ImageCalculator). At this point, the reference Amplitude/Offset image is translated from -7 to 0 pixels (Image/Transform/Translate). Finally the sample image is divided by the reference image to generate the dark-field image.

6.2.4 Neutron Far-Field Interferometry at NIST

In the 2017 far-field experiments at NIST, $\lambda = 5.4$ Å, $D = 26$ mm, $L = 4.6$ m, $z$ ranges from 0.48 m to 1.58 m, and $P_g = 2.4 \mu$m. An Andor Neo camera was cooled to $-30^\circ$C with a pixel size of 6.5 $\mu$m. The readout was 200 MHz, with a 12 bit low noise readout option. A 50 mm Nikon F.12, with closing focus of 50 cm and a reproduction ratio of 7.9 was used. The pixel size was 51.35 $\mu$m. A geometric magnification of 2 resulted in an effective pixel size of 25.7 $\mu$m. 6 mm of Si powder helped diffuse the beam structure while a liquid $H_2$ moderator was also used to cool the setup.

Autocorrelation plots were generated using Mathematica’s nonlinear least squares fitting. Dark-field images for each sample were imported into ImageJ, then filtered using a median filter (r = 2.0) and then a mean filter (r= 3.0). Histograms of areas of interest (fractured neck region in middle of sample, non-fractured good region on left side of sample) revealed the average DF signal and standard deviation. The standard deviation was divided by the square root of the number of pixels used in the average to obtain uncertainty in the images.
All of the $\xi$, DF, and uncertainty values were loaded into an XLS spreadsheet for the five different areas of interest. The XLS file was loaded into a Mathematica notebook designed to explore non-linear least squares fits for the experimental data. Fig. 6.14 shows estimated values for $r$ and $A$ based on the non-linear fitting for the AM half-life sample in the neck region.

In 2017, the far field interferometer setup was altered to move the sample to detector distance instead of the grating distance $D$ as in 2016. Calculations for the autocorrelation length, $\xi$, in 2017 are shown in Fig. 6.3.

The calculation of $z$ was specific for the sample holder used for this run. A 1.5 m translation stage had a free range of motion from a setting of 0 mm to a setting of 1100 mm. The formula for the sample-to-detector was $z=1577$ mm - sample
position (in mm).

\[
\xi = \frac{\lambda z D}{(L_1 + D + L_2)P_{g1}^2} \quad (6.1)
\]

\[z_{\text{min}} = 1577 \text{ mm} - 1100 \text{ mm} = 477 \text{ mm}\]

\[\xi(477 \text{ mm}) = 603 \text{ nm}\]

\[z_{\text{max}} = 1577 \text{ mm} - 40 \text{ mm} = 1537 \text{ mm}\]

\[\xi(1577 \text{ mm}) = 1.99 \mu m\]

With some change in the motor stage holding the sample, an even larger range of
\[\xi (z=10 \text{ mm}) = 13 \text{ nm} \text{ to } \xi (z=2200 \text{ mm}) = 2.78 \mu m\] could be possible.

The pinhole optical system used in neutron imaging will create geometric blur
as the sample-to-detector distance increases. The pinhole diameter \(D\) was 1.5 cm;
the pinhole-to-detector distance \(L\) was 6 m. The geometric blur is blur= \(zL/D\) and is 1.2 mm at \(z=477 \text{ mm}\) and 3.9 mm at \(z=1577 \text{ mm}\).

In summary, the NIST run January, 2017 at 5.4 \(\text{Å}\) accessed \(\xi=0.60\) to 2.0 \(\mu m\)
with blur ranging from 1.2 to 3.9 mm. Under these conditions, conventional SS316
showed little dark-field scattering except at fracture. The AM SS316 75\% fatigued
sample showed an increase in dark-field scattering in the neck region, especially
with \(\xi=1 \mu m\).

6.3 Results and Discussion

Thin SS316 samples revealed several interesting characteristics in the attenuation
and dark-field images. The following sections describe observation of crack forma-
tion in fatigued SS316 samples and information regarding their crystallographic
phases through Bragg edge imaging.
6.3.1 Traditional Radiography and Tomography at ORNL

A thin SS316 SLM AM sample with a manufactured crack was imaged before and after bending the samples with pliers. Figure 6.4 shows the images of the sample before bending with front and side views.

![Figure 6.4. A front (left) and two side (middle, right) views of the thin SS316 sample before bending. The manufactured crack can be seen in the top 2 cm of the sample with a zoomed in view of the area around the crack in the image on the right.]

When the thinnest portion of the sample (2 mm) is perpendicular to the beam, the sample appears uniform. However, when the sample is rotated such that the beam must penetrate through 12 mm of SS316, we can observe the manufactured crack and a screw holder quite well. A zoomed in view of the crack shows several small regions of high attenuation near where the crack stops in the middle of the sample. To understand if these regions extended after bending the sample with pliers, we look at Fig. 6.5.

After the sample was bent, we can immediately notice a difference in the scattering signal in the sample’s front view. The average dark-field signal in the fabricated crack portion before bending is 0.173 while after bending it drops to 0.141. With air
having a value of 0 in these images, this implies that the attenuation of the sample is becoming more similar to air with more bending. In the side view comparison of the sample, we also observe this drop in signal after bending. The area where the crack meets the middle of the material shows regions of diminished attenuation, possibly indicating a likely region for formation of a crack. One explanation for this is that during the printing process, a “pristine” sample should undergo an annealing process to ensure homogeneity throughout the sample. In the case of Li-ion batteries, the first charge/discharge cycle is used to build up the solid electrolyte interface. In the case of AM samples, a similar type of procedure (heat treatment to prevent porosity, light bending to prevent fatigue) should be developed.

One challenge with neutron radiography for observing crack formation is that while the attenuation imaging is useful, it does not say anything about the internal scattering of the sample. Grating-interferometry provides these same attenuation images with the added benefit of differential phase contrast and dark-field im-
ages. With the combination of scattering images and attenuation images, a high flux research reactor like ORNL HFIR has plenty of beneficial use for grating-interferometry.

6.3.2 Far-Field Experiments at NIST

A summary of the experiments run at NIST using the far-field interferometry setup are described in Table 6.2. Of note for these experiments is the change of the dark-field value for air equaling 1. The projections for dark-field images below utilize 1-DF so values close to 1 mean regions of more opaqueness.

Table 6.2. AM and Conventional Samples and Experiments

<table>
<thead>
<tr>
<th>Experiment #: and Sample</th>
<th>Number of Sample-Detector Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: AM fractured, pristine</td>
<td>5 steps</td>
</tr>
<tr>
<td>2: AM 0.75 fatigue</td>
<td>13 steps</td>
</tr>
<tr>
<td>3: Vertical AM 0.75 fatigue</td>
<td>5 steps.</td>
</tr>
</tbody>
</table>

6.3.3 Fractured and Pristine SS316 AM Samples

With a wide range of scattering lengths probed (606 nm - 2 \( \mu \)m), imaging of the fractured, half-life, and pristine samples revealed interesting features. Dark-field images of a pristine 3 mm SS316 sample revealed that the sample was not in fact so pristine at small scattering lengths. From left to right in Fig. 6.6, we can observe areas of higher and lower scattering when \( \xi = 606 \) nm. This dark-field projection is shown without coloring to exemplify the ease it was in spotting the differences in uniformity during the imaging experiment.

A plot of the average signal for these six regions in the pristine sample is shown in Fig. 6.7. While this is only one scattering range, the wide range of signals from 0.55 up to 0.82 indicate the need for annealing.

A comparison of the fractured and pristine samples at \( \xi \) values of 600 nm and 960 nm can be observed in Fig. 6.8.
Figure 6.6. A pristine, 3 mm thick additive manufactured dogbone of SS316 at $z = 0.4$ m ($\xi = 606$ nm) showing six regions of interest. Across the sample, one can observe differing areas of heterogeneity in a sample that should be homogeneous.

At 600 nm, the fractured sample shows small amounts of scattering except for the location of the crack. In comparison to the pristine sample, the scattering signal is more homogeneous, but weak in the fractured sample at this scattering length. When the probed scattering length increases to 960 nm, two areas of interest in the fractured center appear. The diagonal region on the left is where a crack was visible, however another vertical region on the right also appears to have high scattering. Meanwhile, the pristine sample gains uniformity with strong scattering signal throughout the sample. This indicates that during the SLM printing process, pristine samples are manufactured with larger scattering centers. Proof of this
hypothesis is shown in Fig. 6.9, where $\xi$ is increased to 1.31 $\mu$m, 1.66 $\mu$m, and 2.0 $\mu$m.

From 1.31 - 2.0 $\mu$m, only the fractured sample appears to gain in scattering signal. The pristine sample’s homogeneity is finally shown in comparison to the fractured sample, indicating that it truly is pristine at higher scattering lengths. One worthy mention is a reminder that the scattering signal includes scattering from all particles of a certain size and smaller. At 2 $\mu$m, the scattering region of the crack in the fractured sample is included in the dark-field image. However, it is covered up by remainder scattering from nearby larger particles. This shows
Figure 6.9. Images of the 3 mm SS316 AM pristine and fractured samples at scattering lengths of 1.31 μm (top), 1.66 μm (middle), and 2 μm (bottom).
the importance of probing a wide range of scattering lengths in understanding the fatigue process.

### 6.3.4 Half-life SS316 AM Sample

The half-life 3 mm AM sample showed interesting features in the neck region as seen in Fig. 6.10 after median ($r = 2.0$ pixels) and mean ($r = 3.0$ pixels) filters are applied. The average dark-field signal across the sample is more homogeneous than in the pristine sample as well (Fig. 6.11).

![Figure 6.10. A half-life additive manufactured dogbone of SS316 at $z =0.4$ m ($\xi = 606$ nm). In the middle of the neck region, a darker signal can be observed, indicating a higher amount of scattering.](image)

![Figure 6.11. Graph of the DF signal versus region in Figure 6.10. The sample is quite uniform except for the neck region, where the scattering is stronger.](image)
Crack formation can be initially observed in the middle of the half-life sample across a range of scattering lengths (Fig. 6.12).

Figure 6.12. Dark-field images of the SS316 AM half-life sample at scattering length values of 600 nm (top), 720 nm (middle), and 840 nm (bottom). The formation of a crack can be observed in the neck region at 720 nm and expands in the 840 nm image.
At a $\xi$ value of 720 nm, a small area of interest in the middle of the neck region shows higher amounts of scattering. One possibility for this signal increase is pore migration towards the neck region. When the sample is fatigued to 75% of its initial strength, pores within the microstructure are concentrating in the neck region. By the time the scattering length increases to 840 nm, the scattering becomes more widespread with a much higher concentration in the neck region. Figure 6.13 shows the extended range of $\xi$ values probed from 960 nm to 2 $\mu$m.

To understand the size of the particles causing the scattering within the pristine, half-life, and fractured samples, a nonlinear fitting of the dark-field signal and its relationship with the Gaussian function that is dependent upon radius and volume fraction (Equations 3.20, 3.22, and 3.24) is shown in Fig. 6.14.

The nonlinear fitting gives representations of the particle radius and the variable, $A$, which is determined by the volume fraction of pores within the sample. The estimate of particle size for the pristine sample is 0.72 $\mu$m, the half-life sample of 0.88 - 0.89 $\mu$m, and the fractured sample of 0.70 $\mu$m (neck region) - 1.05 $\mu$m (good region). While it was already explained how the pristine sample was not necessarily as homogeneous across the sample as expected, the darkfield signal was taken across the entire specimen for simplicity.

The radius size of the particles causing the scattering appears to decrease from the pristine to half-life to fractured samples, indicating a progression from larger to smaller sized particles in the neck region. In comparing the “good” and neck regions, the fractured sample is of particular interest. With a change from 1.05 $\mu$m in the good region to 0.70 $\mu$m around the crack, this suggests a migration of pores towards areas of fatigue. However, this trend is not observed yet in the half-life sample as both samples were sized near 0.88 $\mu$m.
Figure 6.13. Dark-field images of the SS316 AM half-life sample at scattering length values of 960 nm (top), 1.07 μm (middle), and 2 μm (bottom). Crack formation observed in the 720 nm and 840 nm images is well defined in the 960 nm image. The signal becomes saturated at the 2 μm scale range.
Figure 6.14. Plot of darkfield signal versus autocorrelation length for the pristine, half-life, and fractured 3 mm samples. 75%neck: r is .8808, A is 0.03906. 75%Good: r is 0.8930, A is 0.0247. PristineNeck: r is 0.7211, A is 0.04468. FracturedCrack: r is 0.7038, A is 0.03420. FracturedGood: r is 1.0526, A is 0.010343. The “good” region refers to the wider end regions of the dogbone samples while the neck region is self-explanatory. Comparisons of these two regions was thought to describe the progressive change of scattering as a function of fatigue.

6.3.5 Half-life SS316 AM Sample in Vertical Orientation

AM and conventional samples were rotated to be imaged in the vertical direction. The thought behind rotating the samples is that the scattering within the sample may be orientation dependent, meaning a potential fracture point may only be observed in one direction. Fig 6.15 shows the 3 mm AM samples in the vertical orientation.

Unfortunately, evidence of scattering amongst potential fracture locations in this orientation was unseen. This experiment proved that dogbone samples need to be placed in a perpendicular direction to the gratings (i.e. if vertical gratings, dogbones should be horizontal to the beam). By doing so, any potential fracture in the neck region is in the same orientation as the gratings.

6.3.6 Neutron Talbot-Lau Interferometry at HZB

Talbot-Lau interferometry using neutron radiography and tomography resulted in many interesting features in the 2 and 3 mm SLM AM SS316 samples. At HZB,
Figure 6.15. Vertical orientation of the AMHalf sample at $\xi$ of 600 nm (left) and 2.0 $\mu$m (right). Anticipated crack formation observed with the sample in the horizontal location is not seen when the samples are rotated to their vertical orientation.

the Talbot-Lau interferometer operated at a $\xi$ scattering length of 1.97 $\mu$m. While it would have been optimal to perform radiography and tomography at multiple scattering lengths, long distance requirements for higher Talbot orders meant only the first Talbot order was used.

Of interest first are the 2 mm thick fatigued, half-life, and pristine samples. As these samples were not pulled or pulled to their half-life/fracture, it was anticipated that scattering lengths of interest would be similar to the experiments at NIST with the 3 mm bent samples. This did not turn out to be the case. Tomography of the dark-field signal from the 2 mm samples is shown in Fig. 6.16.

Line probes of the fractured and pristine samples are shown in Fig. 6.17. A line probe for three tomography experiments of the same 2 mm pristine and half-life samples reveals the scattering increases in the neck region where a point of fracture may be forming. In one experiment the samples were placed in a vertical orientation whereas the other two experiments had the samples mounted horizontally. There are two regions in the half-life sample that have scattering from the top of the neck
Figure 6.16. Dark-field volume of the 2 mm samples showing the pristine sample at the top, the half-life sample in the middle, and the fractured sample at the bottom. Two regions of interest in the half-life sample show higher regions of scattering similar to the fracture point in the cracked sample.

to the bottom, as indicated by the dips in the DF signal to 0.513 in the middle of the neck and 0.544 in the middle right neck region.

In the half-life sample, there are two regions of interest. The first is at the 10 mm mark, where a sudden dip in the signal could indicate possible crack formation. The other location similar to this is from 18 - 21 mm. This region has a much larger drop in the signal and likely indicates a crack will form.

The 3 mm conventionally made SS316 samples that did not show features of interest in the far-field interferometer were studied again. Figures 6.18 and 6.19 show features of interest at the 1.97 μm scattering length.

To the human eye, the fatigued conventional sample only had one visible crack. However, upon inspection, there are many small cracks propagating near the main fracture point. In the half-life sample at the bottom, two small cracks appear in the neck region. While the scattering is not as high as in the fatigued sample, the appearance of two cracks that were not visually apparent proved the viability of the experiment. To further understand the depth of crack formation, the sample was rotated on its side (Fig. 6.19).
Figure 6.17. Line probes of the half-life (top) and pristine (bottom) AM 2 mm samples.
6.3.7 Neutron Bragg Edge Imaging at HZB

Bragg edge imaging revealed several interesting features in the 2 and 3 mm AM printed samples as well as the conventionally machined samples. The Bragg edge
imaging only utilized transmission imaging so no dark-field or phase images were
collected with this experiment.

Previous Bragg edge studies divided Bragg images of stainless steel from two
wavelengths in attempts at observing fatigue [26]. Attempts in this work to do the
same with the half-life additive manufactured samples resulted in Figures 6.20 and
6.21.

Figure 6.20. An image generated from the $\overline{T}([3.8,4.1] \, \text{Å})$ divided by $\overline{T}([4.3,4.5] \, \text{Å})$. The
pixel size is 60 $\mu$m (binning=2). There is a very slight change in Bragg transmission in
the neck region, but crack formation is not localized based on this data.

Crack formation observed in the dark-field volumes was anticipated to be ob-
served when dividing the areas from 3.8 - 4.1 Å by 4.3 - 4.5 Å and 2.2 - 2.6 Å
by 3.8 - 4.1 Å. However, neither region showed evidence of crack formation. One
possibility is that transmission imaging is not the best method for viewing fatigued
regions, whereas the dark-field imaging shows scattering for small features. Further validation of this hypothesis is shown in Figure 6.22 for the Bragg imaging of the half-life conventional sample.

![Image](image.png)

Figure 6.22. An image generated from the mean (3.8 Å to 4.1 Å) divided by mean (4.3 Å to 4.5 Å). The pixel size is 60 μm (binning=2).

Crack formation seen in this sample using grating-interferometry was not observed in the Bragg edge imaging. This may imply that while transmission imaging of fatigued samples is useful, there are better methods such as dark-field imaging that can show features of interest clearer.

An interesting difference was observed between conventionally machined and additively manufactured SS316 in the transmission signals. Figure 6.23 shows how the additively manufactured steel was of a different crystallographic form than expected.

Transmission through the fractured sample is the highest, followed by the half-life and pristine samples. Of interest is the region between 3.5 - 4.2 Å, where a change in phase from pristine to fractured can be observed. The pristine sample has characteristics of austenite, however the fractured sample has much less of a change in this region. This could be related to a transition from γ-austenite’s (111) at 4.17 Å to α-martensite (110) at 4.10 Å.

When looking at the 2mm fractured, half-life and pristine samples, we notice a change in the peak region between 3.7 - 4.2 Å (Figure. 6.24).
Figure 6.23. Bragg edge of the conventional (left) and AM (right) 3 mm printed samples. In the conventional sample, the spectra fit similar to α-martensite as seen in Robin Woracek’s paper. In comparison to the conventional samples, a peak at 3.65 Å and 4.2 Å indicate the AM samples are of a different crystalline form, austenite, instead of α-martensite as seen in the conventional.

Figure 6.24. Bragg edge spectra for the pristine, half-life and fractured 2 mm pulled samples.

The fractured sample always had the highest transmission of the three samples, which supports the concept that as a sample is pulled to fracture it becomes longer and thinner. The main region of interest in the Bragg spectra is between 3.7 - 4.2 Å, where it appears a Bragg edge is disappearing in the fractured sample as compared to the pristine sample. During this transition from pristine to fractured, the pristine sample appears to be austenite while transitioning towards α-martensite in the fractured sample. This trend implies a reorientation of the grains during the tension process.
Several scanning electron microscope (SEM) experiments on the pristine and half-life samples performed at LSU Mechanical engineering revealed porosity (several $\mu$m up to 10 $\mu$m) in the pristine samples. When the samples were fatigued through bending or pulling, the pores within the samples changed. The new porosity was found to exhibit rearrangement of the grain orientation to a finer microstructure. This supports the Bragg edge data for the 2 mm samples in Figure 6.24 and the hypothesis that AM samples need some sort of annealing treatment to distribute porosity and grain orientation equally within samples.

6.4 Conclusions

Traditional neutron radiography performed on a 2 mm thick SS316 sheet at ORNL did not reveal crack formation after several hundred attempts at bending, despite a powerful reactor with high counting statistics. Focus then turned to neutron sources with Talbot-Lau and far-field interferometry for imaging. Visualization of fatigue in conventional and additively manufactured SS316 was possible using interferometry and also Bragg edge imaging.

3 mm SLM printed SS316 dogbones in their pristine, half-life, and fractured states were imaged with far-field interferometry at NIST. Probing a wide range of autocorrelation lengths from 606 nm - 2.01 $\mu$m, the dark-field signal changed between different scattering lengths for all three samples. At 606 nm, the pristine sample was not shown to be as uniform as expected, with six different regions of scattering ranging from 0.38 up to 0.6. A likely cause for this inhomogeneity is from a lack of annealing after the printing process. In the half-life sample, evidence of crack formation can be observed at 840 and 960 nm when the sample is horizontal to the beam. When the sample is switched to a vertical orientation, a concentrated region of scattering is not observed, leading to the conclusion that scattering is directional in these SS316 dogbones.
In studying the dark-field images for the pristine, half-life, and fractured samples, a connection to a Gaussian function relating particle size and volume fraction with dark-field signal was made. The pristine sample had the smallest particle radius near 0.72 μm, the fractured sample the widest range of particle sizes (0.70 - 1.05 μm), and the half-life sample in-between (0.88 μm). The wider regions at the ends of the half-life and fatigued samples had larger particle sizes in comparison to the neck regions, implying a migration of porosity as the samples underwent fatigue.

Traditional Talbot-Lau interferometry at HZB was performed on the same 3 mm bent AM dogbones as well as conventionally manufactured 3 mm thick dogbones and 2 mm thick SLM printed SS316 dogbones that were pulled via a jig. With a set autocorrelation length of the instrument, ξ of 1.97 μm, crack formation was observed in the 3 mm conventional half-life and fatigued samples. While a large visible crack was expected in the dark-field images for the fatigued sample, a microstructure of small cracks was discovered surrounding the large crack. In the case of the half-life conventional sample, two small cracks not visible to the human eye were observed to penetrate 1 mm into the 3 mm samples.

Volumes of the 2 mm pulled samples showed potential regions of interest for crack formation. In the half-life 2 mm sample, a line probe of three different tomography experiments (one with samples vertical, two horizontal) revealed similar drops in the DF/Attenuation signal, indicating possible crack formation. For comparison, the pristine sample showed less variability from one tomography dataset to another across the same probed distance.

Bragg edge imaging of the 2 and 3 mm AM and conventional samples revealed two interesting features. First, the 3 mm SLM AM printed parts were originally thought to be of the α-martensite phase, similar to conventionally machined sam-
amples. However, this was not the case. Instead, the crystalline form represented austenite, indicating a required change needed for printing. The second feature involved a gradual change from 3.8 - 4.1 Å for the 2 mm pristine, half-life, and fractured samples. The fractured sample appears to be of the α-martensite phase, however after pulling of the samples the higher transmission in this region indicates a crystallographic phase change towards the austenite phase. In combination with SEM imaging performed with LSU Mechanical Engineering, this turned out to be proven true.
References


Chapter 7
Observation of Geometric Crystal Twinning and Phase

7.1 Introduction

The use of grain boundary engineering in the synthesis of single crystals plays a key role in the future of materials science. It is thought that grain boundaries play an important role in the deformation and fracture of crystalline material due to high temperatures, boundary movement, and migration [1, 2, 3]. Previous experiments relied on “designing and controlling” a material with specific grain boundaries to optimize the electrical and physical properties of that material [3]. With the relationship between grain boundaries and magnetism in α and γ phases of Fe taken as a background [4], recent studies have focused on engineering grain orientation for strong magnetic materials [5]. By altering the grain orientation in a material, the possibilities are high for advanced magnetic materials like superconductors, maglev trains, and imaging devices.

While synthesizing materials with specific grain boundaries may be possible, there is a need for effective means of characterizing these materials. Detection of microstructure, twinning and texture are possible using X-ray diffraction [6], electron backscatter diffraction [7], and X-ray phase contrast imaging [8]. Although the information may be valuable for small samples, the interaction of X-rays with a crystalline material mostly occurs near the surface. This makes it difficult to obtain crystal lattice information for a sample of reasonable thickness [9].

Neutron imaging is looked at as a viable option for crystal imaging due to its bulk sensitivity in comparison to X-rays. Previous neutron Bragg edge imaging of polycrystalline samples has viewed intensity changes as a function of \( hkl \) reflections at specific wavelengths due to differences in strain [10, 11], texture [12], and phase
The differences in the atomic cross sections and path length make it possible to determine the various phases within a material using a Rietveld fitting [14]. Anisotropy and preferred orientation is also possible to determine within textured materials through neutron Bragg edge imaging [15].

For the case of single crystal imaging, energy-resolved neutron imaging found single-crystalline domains within bulk gold/palladium samples that were otherwise too opaque for traditional characterization methods [16]. Diffraction contrast tomography with neutrons excels at revealing multiple grain structures, but is a challenging experiment in terms of instrument time and data processing [17].

The hypothesis driving this work is that crystal twinning can be rapidly detected using Talbot-Lau and far-field neutron grating-interferometry due to the possible change of the level of porosity at the twinning interface. In our search for an efficient procedure for detection of crystal twinning in synthetic and geological samples, neutron interferometry imaging is explored. In principle, the phase contrast imaging modality should reveal twinning based on porosity properties in the twinning interface. The dark-field imaging modality may show crystal twinning via porosity at the crystal domain interface over a range of auto correlation scattering lengths in both interferometers.

7.2 Experimental

The following sections describe the neutron interferometry methods explored to observe crystal twinning. Talbot-Lau interferometry was utilized in Berlin while far-field interferometry was used at NIST.

7.2.1 Neutron Talbot-Lau Interferometry at HZB

Neutron Talbot-Lau interferometry experiments of crystals were performed at HZB CONRAD2 [18, 19]. The geological example was a classic twinned “fishtail” titanite, CaTiSiO₅, and provided by the LSU Department of Geology & Geophysics. A
synthetic twinned crystal examined was a strontium ruthenate prepared by high-pressure, floating-zone crystallization, provided by the LSU Department of Physics & Astronomy.

![Image of Titanite and Strontium Ruthenate](image)

Figure 7.1. Titanite (a) and strontium ruthenate (b) samples imaged at HZB and NIST.

An Andor Neo SCMOS camera was used (2560×2160 pixels, effective pixel size of 30 μm) with a 100 μm thick 6LiF/ZnS scintillator. Three crystals were mounted on a tomography rotation stage in a vertical stack assembled with thermoplastic from a consumer-quality hot glue gun. The hydrogen content of the thermoplastic strongly attenuates the neutron beam and provides a convenient demarcation between samples. The instrument operated with a polychromatic neutron beam with a maximum intensity at 2.5 Å, a grating period $p_1$ of 7.97 μm, a sample-to-detector distance, $z$, of 5.0 cm, and a $G_1$-$G_2$ distance, $D$, of 2.27 cm, yielding an auto correlation scattering length, $\xi$, of 1.97 μm.

### 7.2.2 Statistics

The interferogram projections and tomography volumes are analyzed by the correlating pixel values in one image versus pixel values in another. For example, the DPC image will be compared to the ratio of DF/Attenuation pixel-by-pixel in a scatterplot; the structure of the scatterplot will be quantitatively assessed by a statistical measure of the correlation. Other image pairs will also be assessed; the dark-field versus attenuation is a good measure of surface scattering.
The images are reduced to vectors of pixel intensities. It is important that the vectors have equal length and consistent order. This is done by generating a binary image mask from the attenuation image, closing holes within the mask, and erosion of the surface by a few pixels to reduce surface effects on the statistical analysis. A list of the \{row, column\} indices of the mask elements is used to extract pixel values from the attenuation, dark-field, and DPC images. These lists of pixel values are the vectors used for the correlation testing.

Correlation testing between two or more vectors is a common activity in economics. A recent review was published by Schmid et al. from the Department of Economic and Social Statistics, University of Cologne.\(^1\) Consider two vectors of data, \(\nu_1\) and \(\nu_2\), representing for example, education and income. A first test is to reject the null hypothesis of no correlation. The second test is to measure the correlation between the two vectors. In this crystal imaging work, we are at the first test.

The null hypothesis of correlation starts with the assumption of true, there is no correlation, and this assumption is verified if the \(p\)-value remains small. A large \(p\)-value means the null hypothesis is false and there may be a correlation between the two vectors. The null hypothesis can also be combined with an alternative hypothesis that a correlation does exist. This sounds like a lot of work. Mathematica has a command, `IndependenceTest`, which combines null and alternative hypothesis with various tests. We are starting with a monotonic test, that is an increase in one image is correlated with an increase in another image. The Blomqvist’s \(\beta\) is a test for monotonic correlation. The Blomqvist’s \(\beta\) test is one of group of test called copula. The word copula comes from the Latin “connection, linking of words”.

Blomqvist’s $\beta$ (see Schmid footnote, Eq. 10.11) is calculated for two vectors $\nu_1$ and $\nu_2$ (higher dimensions can also be calculated) as:

$$\beta = P\{(\nu_1 - \bar{\nu}_1)(\nu_2 - \bar{\nu}_2) > 0\} - P\{(\nu_1 - \bar{\nu}_1)(\nu_2 - \bar{\nu}_2) < 0\}$$  \hspace{1cm} (7.1)

where $\bar{\nu}_1$ is the mean value of $\nu_1$ and $P$ is the probability. A value of zero means no correlation and $\pm 1$ indicates complete, monotonic correlation.

The examination of two data sets for independence can be done by several tests. The Mathematica command IndependenceTest will perform several tests on a pair of data sets. The test Blomqvist $\beta$ appears to have the largest dynamic range of all the tests for the comparison of (dark-field)/(attenuation) versus DPC. Figs. 7.2, 7.3, and 7.4 show an uncorrelated, positive correlated, and negatively correlated dataset. The results of independence testing where the null hypothesis is independence. A small $p$-value suggests the hypothesis is unlikely.

Figure 7.2. An uncorrelated data set: $x = N(\mu, \sigma^2)$ and $y = N(\mu, \sigma^2)$ with $\mu = 1$ and $\sigma = 3$.

7.2.3 Neutron Far-Field Interferometry at NIST

Experiments at NIST were performed using a far-field interferometer, with a design wavelength of 5.4 Å, a grating period, $P_g$, of 2.4 $\mu$m, a sample-to-detector distance, $z$, of 2.257 cm, and $D(G_1-G_2)$ distances of 3 to 40 mm, equating to auto correlation scattering lengths, $\xi$, of 360 nm to 4.75 $\mu$m [20]. Smaller scattering lengths ran
Figure 7.3. A partially correlated data set: $x = N(\mu, \sigma^2)$ and $y = x + N(\mu, \sigma^2)$ with $\mu = 1$ and $\sigma = 3$.

Figure 7.4. A partially correlated data set with a negative correlation: $x = N(\mu, \sigma^2)$ and $y = -x - N(\mu, \sigma^2)$ with $\mu = 1$ and $\sigma = 3$.

into issues with harmonics projected onto images of the samples so scattering lengths below 606 nm are disregarded. The far-field interferometer was optimized for measuring autocorrelation scattering lengths and not image resolution.

The neutron interferograms were processed in Mathematica with a vectorized linear algebra method [21]. The algorithm has recently been converted to Python and is in the process of installation into the Advanced Photon Source tomopy project 3. Interferometry performance was noted with the open beam average visibility and was 5% for the Talbot-Lau system and approaching 20% for the far-field system. This algorithm is more robust than the more commonly used Fourier analysis [22]. Volume reconstructions for absorption, dark-field, and DPC were done

---

3http://tomopy.readthedocs.io
using simultaneous iterative reconstruction technique (SIRT) from the ASTRA toolbox [23].

7.3 Results and Discussion

All crystals were imaged on a tomography rotation stage such that radiography and tomography could be performed without sample manipulation. Tomography reconstructions of the attenuation, DPC, and DF projections of titanite did not reveal any crystal twinning. One reason for this is the filtering used in tomography reconstruction of coarsely spaced datasets will blur sharp features. By solely looking at projection data at specific angles, a more concentrated effort can be placed on observing the absorption, DPC, and dark-field signal over the course of multiple grating steps.

Therefore our attention turned to the projection images. The projections were then analyzed on a pixel-by-pixel basis for each rotation angle and assessed with a statistical parameter sensitive to monotonic correlations, the Blomqvist’s $\beta$ parameter. To illustrate the pixel-wise correlations, a pair of closely related rotation angles are selected for the figures below. For the rotation angle shown on the top side for Figs 7.5 to 7.9, crystal twinning is most visible in both the images and the Blomqvist’s $\beta$ parameter. For reference, the image at the bottom is at a rotation angle showing less obvious twinning or a smaller Blomqvist’s $\beta$ parameter.

7.3.1 Titanite (HZB)

Imaging of the titanite sample led to the observation of features looking like crystallographic plane excitations at several rotation angles. Two Talbot-Lau image projection sets for the titanite sample are shown in Fig. 7.5.

At 96.0° (top set of images), the DF scattering between the two domains increases markedly; see also Fig. 7.6. The DPC image shows a change left to right across the sample, specifically at the interface between the two crystal domains.
Figure 7.5. Two slightly different rotation angles for the titanite sample at 96.0° (top) and 88.8° (bottom). Scale for attenuation and DF: 0 to 1, with air having a value of 0. DPC scale -π (black) to + π (white). Surface roughness is removed from the projection, especially the DF projection, by application of an eroded binary mask generated from the attenuation projection.

This is attributed to path length differences. At 88.8°, the change between the left and right portions of the crystal are not as well defined as at 96.0°. While this could imply that twinning can be observed in the sample at specific rotation angles, these features could also be dependent upon direction of the phase gradient in the sample. Such orientations could induce features that appear similar to twinning.

The Blomqvist-β analysis in Fig. 7.6 shows the differences between 88.8° and 96.0°. When the crystallographic plane comes into view, there is a high degree of correlation at 88.8° in the DPC vs DF/attenuation and DPC vs darkfield images. At 96.0° a split in the DPC signal is observed, indicating a possible twinning.

The plot of DPC versus (DF/Attenuation) shows the most dramatic change with a rotation angle of 96° indicating detection of a crystal twinning feature. It could
Figure 7.6. Statistical analysis for two slightly different rotation angles of titanite. The top set of images shows titanite at 96.0° and the bottom set shows titanite at 88.8°. The scatterplot is a pixel-by-pixel comparison from attenuation, DPC, and DF images. The Blomqvist-β parameter assesses the independence of the two pixel sets, with a value of 1 showing perfect monotonic correlation.
also be related to behavior of the phase signal that yields positive or negative values depending on phase gradient direction. The splitting of the signal could be generated by geometrical considerations and not necessarily twinning.

Figure 7.7. Titanite projections from both Talbot-Lau interferometry at HZB (a) and far-field interferometry at NIST (b) are assessed with the Blomqvist-\(\beta\) statistic as a function of tomography rotation angle. Note: the different sample mounts at the two facilities removed the correlation of rotation angles. In the far-field setup, a grating difference \(D(G_1 - G_2)\) of 15 mm corresponds to an autocorrelation scattering length of 2 \(\mu\)m, equivalent to the interferometry setup of the Talbot-Lau system. In summary, both systems are sensitive to crystal twinning.

When comparing the tomography datasets from Talbot-Lau and far field interferometry, the crystallographic twinning feature can be visually observed in the projections as a function of rotation angle. Likewise, the Blomqvist-\(\beta\) parameter for some correlations also reveals evidence of an anomaly. In Fig. 7.7, the sharp peaks in the Blomqvist-\(\beta\) parameter indicate the appearance and disappearance of the twinning plan, while for most rotation angles, the Blomqvist-\(\beta\) value remains near constant. One issue that appears with the far-field interferometer is geometric...
blurring. When the sample to detector distance ranges from 0.5 to 1.2 meters as in this experiment, the blurring at far distances is noticeable. In the case of Fig. 7.7 taken with far-field interferometry, the correlations for titanite are not as marked as for the Talbot-Lau interferometer.

### 7.3.2 Strontium Ruthenate

As an example of a topological insulator, a sample of strontium ruthenate, which was thought to be a single crystal, was imaged. Tomography of the crystal revealed several angles of interest so projection images from those specific angles are shown in Fig. 7.8.

![Figure 7.8](image-url)

Figure 7.8. Two rotation angles for the $\text{Sr}_3(\text{Ru}_{0.95}\text{Mn}_{0.05})_2\text{O}_7$ sample; see also Fig. 7.9. At 133.3° (top set of images), the DF scattering within the sample is larger than at 127.3° (bottom set of images). Scale for attenuation and DF: 0 to 1, with air having a value of 0 (black). The DPC scale ranges from $-\pi$ (black) to $+\pi$ (white).

Absorption, DPC, and dark-field images show evidence of possible twinning at rotation angles near 133.3°. The easiest image to observe this is in the absorption image, where a change from the left side to the right side of the sample is indicated by a jump in the signal. The dark-field is also of interest as it indicates scattering features at the 1.97 μm length scale, where scattering changes from the top of the sample to the bottom in both projection images. At 133.3°, a small gap in the
scattering signal is observed in the crystal. This may be indicative of directional crystal grain orientation within the sample.

When the sample is rotated further to 127.3°, evidence of twinning is minimized. The DPC signal appears homogeneous throughout the sample, explaining how the sample appears homogeneous. This explains the possibility of twinning being detected only within a small range of angles in tomography experiments. In comparison to the titanite sample, the projections for strontium ruthenate look similar. Again, a possible crystal twinning feature is detected at a few tomography rotation angles, however this could be due to the geometry of the crystal in the neutron beam.

At 133.3°, the correlation plots, Fig. 7.9, show substantial structure. The correlation plots at 127.3° are representative of most other rotation angles, thus angle 133.3° is unique and believed to be aligned with a twinning feature.

7.4 Conclusions

Neutron grating-based interferometry using Talbot-Lau and far-field interferometry setups was performed for crystalline samples at HZB and NIST, respectively. Titanite and strontium ruthenate crystals were chosen due to anticipated twinning within the structure. Tomographic datasets were disregarded due to spatial information lost over the course of several hundred rotation angles. Projection images of absorption, DPC and dark-field were the main focus for this chapter. Visually, the DPC and dark-field images show features which are consistent with twinning provided the crystal orientation with respect to the interferometer is optimized.

Pixel-wise correlations of DPC and/or dark-field versus attenuation (see Fig. 7.6) also show evidence of crystal twinning and indicate a route for faster crystal inspection. While the features are thought to be a result of crystal twinning, another explanation exists. Depending upon the direction of the phase gradients in titanite
Figure 7.9. Statistical analysis for two rotation angles of the Sr$_3$(Ru$_{0.95}$Mn$_{0.05}$)$_2$O$_7$ sample at 133.3° and 127.3°. The plot of DPC versus (DF/Attenuation) shows the most dramatic change with rotation angle 133.3°. The parameters for 127° are typical for most of the rotation angles. The parameters at 133.3° are unique and taken as evidence of detection of a crystal twinning feature.

and strontium ruthenate, crystal geometry in the neutron beam path could explain the splitting of the absorption, DPC, and dark-field signals. Evidence of this splitting was seen in the DPC/(DF/attenuation) images, where both positive and negative correlation were observed at specific rotation angles in the Blomqvist-β parameter. The Blomqvist-β test has potential application to easily detect crystal geometry or twinning.

At the beginning of this project, the operational hypothesis for identifying internal structures of twinned crystals was based on the differential phase contrast image. The project has now revealed the utility of dark-field imaging when used with an autocorrelation scattering length near 2 μm. Localized scattering observed in the
strontium ruthenate sample showed how features of interest change from projection to projection. In comparison to the additive manufacturing studies performed in Chapters 5 and 6, the scattering of crystallographic gradients is anticipated to be several magnitudes of order smaller (Angstrom instead of micrometers). With neutron far-field and X-ray near-field interferometry gaining access to smaller and smaller scattering length scales, it is expected that higher quality X-ray and neutron imaging results can be obtained.
References


8.1 Conclusions

Talbot-Lau and far-field grating-based interferometry have shown great promise in the imaging of additive manufacturing, lithium ion batteries, and single crystals. In combination with other techniques like Bragg edge imaging and electron microscopy, X-ray and neutron interferometry provide quantitative scattering and transmission/absorption information of the samples. This is the first known report on the application of far-field interferometry to additive manufacturing, batteries or crystals. Also, these are some of the first experiments for spatially resolved scattering imaging.

X-ray synchrotron imaging provided high enough flux to observe five cathode/anode layers in fresh and worn (over 1000 cycles) Li-polymer batteries. As X-rays are cheaper to use than neutrons, the possibilities of utilizing X-ray grating interferometry for batteries are great. Experiments using an X-ray tube source did not provide enough flux to gain good contrast through thin (0.5 - 2.0 mm) batteries even with high exposure times. Due to the parallel beam geometry at the synchrotron sources, 2 second exposures showed electrode and separator material through the thinnest and thickest portions of all batteries. Although this imaging does not describe what accounts for the scattering features, it shows the locations of inhomogeneity within batteries.

Neutron diffraction and Bragg edge imaging of batteries showed the wavelength dependence of $hkl$ peaks between charged and discharged, fresh and worn batteries. Principal component analysis and multivariate curve resolution was attempted to understand the equilibrium within a seven-component battery system, however
this was not as successful as separating the battery into three \( \text{Li}_x\text{CoO}_2 \) and three graphite/LiC\(_n\) components. Resolving Bragg edge peaks for both the fresh and worn battery was a challenge due to the peaks not corresponding with known LiCoO\(_2\) or graphite hkl values. One possibility for this discrepancy is the degradation of the internal solid electrolyte interface to form metal oxides or lithium carbide structures (LiC\(_{12}\), LiC\(_{18}\)) for which there is little available diffraction data. Also of note in the Bragg edge 3 to 4.5 Å spectra is the transparency of charged versus discharged batteries. For both fresh and worn batteries, the charged state has overall greater transmission than the discharged state. This could be explained by an as-yet unidentified large unit cell species with a strong Bragg edge just above 4.2 Å. By understanding the evolution of battery components as a function of state of charge, changes to the internal battery chemistry can be made to ensure long lifetimes.

In the first application of far-field neutron interferometry to batteries, changes in the scattering locations within fresh and worn batteries was observed. With useful autocorrelation lengths (\(\xi\)) ranging from 600 nm to 3.5 \(\mu\)m, the dark-field signal in 2 mm thick, 43 mAh batteries increased to a saturation point when the gratings were set at certain distances. The fresh batteries reached this saturation point at longer autocorrelation lengths (2.813 \(\mu\)m) in comparison to slightly worn batteries (2.588 \(\mu\)m) and worn batteries (2.25 \(\mu\)m). Even with many unknowns as to what components within the batteries are causing this scattering, this implies that as the batteries go through charging and discharging cycles, the particle size within the batteries is decreasing. Slight differences in the scattering signal were also observed between the same batteries charged and discharged, although these changes were not as noticeable as between fresh and worn batteries.
In another application of neutron Talbot-Lau grating interferometry, porosity in additively manufactured EBM Ti-6Al-4V samples was observed. Features discovered in the transmission and dark-field interferometry images at 60 μm spatial resolution led to tomographic high-resolution transmission imaging at 30 μm. During reconstruction of the sample, a chimney-like pore structure was observed beginning at the location of the first print layer continuing to halfway through the sample. The pore structure is likely due to a lack of fusion during the printing process, indicating the need for annealing of samples to ensure homogeneity.

Points of potential fracture were discovered in 2 mm and 3 mm thick SS316 selectively laser manufactured dogbones with both Talbot-Lau and far-field neutron interferometry. The 2 mm samples (pristine, half-life (75% fatigued), and fractured) were pulled using a jig while the 3 mm samples were bent similar to a diving board. Far-field interferometry showed the progression of crack formation in the half-life sample from 600 nm up to 2 μm. Good scattering lengths for observation of crack formation in the half-life 3 mm sample were 960 nm and 1.07 μm. In the experiment at HZB using Talbot-Lau neutron interferometry, a scattering length of 1.97 μm showed two regions of possible crack formation. Determining the location of fatigue in additive manufacturing can play a large role in preventing catastrophic events.

Talbot-Lau and far-field interferometry of crystals showed excitation of the absorption and dark-field signal at certain angles. The anticipation was that grating interferometry could be used to detect crystal twinning through the absorption, phase, or dark-field images. The excitations were originally thought to be a result of neutron birefringence, however this has not been observed to date. A Blomqvist beta correlation test comparing the three datasets for strontium ruthenate showed positive and negative splitting of the DF/attenuation signal, possibly indicating
twinning. While this project was not as successful as the batteries or additive manufacturing, there is still promise that interferometry optimized under different scattering lengths could reveal twinning.

### 8.2 Future Studies

With the success of X-ray and neutron experiments presented, there are several methods of improvement in reference to both the experimental setups and applications. In the case of X-ray imaging, the near-field interferometry setup designed by Dr. Han Wen showed great promise in observing scattering on the 20–900 nm length scale. Application of this new setup to samples with small features can provide users with results similar to a small angle X-ray scattering (SAXS) experiment while also including images of those features. Easier samples to begin with include monodispersed spheres of known size and radius such that a comparison dark-field imaging and SAXS experiment could be performed.

The far-field neutron interferometry experiments at NIST were performed in two orientations: one setup where the grating distance moved and the other where the sample moved. When the gratings were moved, issues with harmonics and the scattering signal going negative at high grating distances limited the scattering range. In the second experiment where the gratings were set to a maximum visibility and the sample was moved, problems with geometric blurring at far sample to detector distances also limited useful scattering lengths. As a new experimental setup, parameters such as sample to detector distance, grating distance, slit width, and the use of flight tubes need to be optimized.

The neutron Talbot-Lau interferometry setup was designed for an autocorrelation length of 1.97 μm. By altering the total length of the system or moving the gratings closer to one another, it would be possible to probe smaller autocorrelation lengths. In the case of the 2 and 3 mm SS316 AM samples, far-field
interferometry showed that crack formation occurs at scattering lengths around 1 μm. Another option is to fabricate a new grating with a smaller period than the 0.8 mm period currently used. However, this option is considerably more expensive and dangerous if gadolinium sputtering is required to make similar gratings. If the Gadox fabrication method for neutron gratings is utilized, this would make for a quick and easy solution.

As far as applications, the goal for most of these projects is to gain in-situ measurements of battery degradation or crack formation in additive manufacturing. By imaging a battery while it is charging or discharging, it would be possible to determine how scattering in the battery is changing spatially. One challenge to this scenario involves the high cross section of lithium with neutrons, making it difficult to determine whether scattering in a Li-ion or Li-po battery is a result of lithium movement or component degradation. In the case of additive manufacturing, the next step is to image dogbone samples while they are undergoing the fatigue process. The HZB facility has a jig designed and set up for in-situ measurements with Talbot-Lau interferometry so this experiment is anticipated to be performed soon.
Appendix A: Copyright Permission for Porosity Chapter

The following is a copyright transfer of the article “Porosity detection in electron beam-melted Ti-6Al-4V using high-resolution neutron imaging and grating-based interferometry” to Springer from myself, Adam J. Brooks. Springer states that I retain the rights of the article “for his/her further scientific career by including the final published journal article in other publications such as dissertations and post-doctoral qualifications provided acknowledgement is given to the original source of publication.”
Copyright Transfer Statement (CTS) for your article in Progress in Additive Manufacturing (25)

Springer <springerauthorquery@springeronline.com>
Fri 5/5/2017 9:11 AM

To Adam J Brooks <abroo38@fsu.edu>

Copyright Transfer

05/05/2017
visit us at springer.com

Confirmation of your Copyright Transfer

Dear Author,

Please note: This e-mail is a confirmation of your copyright transfer and was sent to you only for your own records.

The copyright to this article, including any graphic elements therein (e.g. illustrations, charts, moving images), is hereby assigned for good and valuable consideration to Springer International Publishing Switzerland effective if and when the article is accepted for publication and to the extent assignable if assignability is restricted for by applicable law or regulations (e.g. for U.S. government or crown employees). Author warrants (i) that he/she is the sole owner or has been authorized by any additional copyright owner to assign the right, (ii) that the article does not infringe any third party rights and no license from or payments to a third party is required to publish the article and (iii) that the article has not been previously published or licensed.

The copyright assignment includes without limitation the exclusive, assignable and sublicensable right, unlimited in time and territory, to reproduce, publish, distribute, transmit, make available and store the article, including abstracts thereof, in all forms of media of expression now known or developed in the future, including pre- and reprints, translations, photographic reproductions and microform. Springer may use the article in whole or in part in electronic form, such as use in databases or data networks for display, print or download to stationary or portable devices. This

https://outlook.office.com/owa/?viewmodel=ReadMessageItem&itemId=PB0mT79nGAAF0cyjwAAK2D&isPrintView=1&wid=74&ispopup=1&path=
includes interactive and multimedia use and the right to alter the article to the extent necessary for such use.

Authors may self-archive the Author's accepted manuscript of their articles on their own websites. Authors may also deposit this version of the article in any repository, provided it is only made publicly available 12 months after official publication or later. He/she may not use the publisher's version (the final article), which is posted on SpringerLink and other Springer websites, for the purpose of self-archiving or deposit. Furthermore, the Author may only post his/her version provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be provided by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]."

Prior versions of the article published on non-commercial pre-print servers like arXiv.org can remain on these servers and/or can be updated with Author's accepted version. The final published version (in pdf or html/xml format) cannot be used for this purpose. Acknowledgement needs to be given to the final publication and a link must be inserted to the published article on Springer's website, by inserting the DOI number of the article in the following sentence: "The final publication is available at Springer via http://dx.doi.org/[insert DOI]." Author retains the right to use his/her article for his/her further scientific career by including the final published journal article in other publications such as dissertations and postdoctoral qualifications provided acknowledgement is given to the original source of publication.

Articles disseminated via http://link.springer.com are indexed, abstracted and referenced by many abstracting and information services, bibliographic networks, subscription agencies, library networks, and consortia.

After submission of the agreement signed by the corresponding author, changes of authorship or in the order of the authors listed will not be accepted by Springer.

This is an automated e-mail; please do not reply to this account. If you have any questions, please go to our help pages.

Thank you very much.

Kind regards,

Springer Author Services

<table>
<thead>
<tr>
<th>Journal title</th>
<th>Article title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Progress in Additive Manufacturing</td>
<td>Porosity detection in electron beam-melted Ti-6AI-4V using high-resolution neutron imaging and grating-based interferometry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DOI</th>
<th>Corresponding Author</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B: X-ray Linear Attenuation Coefficient Mathematica Code

The following code is used to calculate the linear attenuation coefficient, $\mu$, for various chemical elements and compounds at differing thicknesses and X-ray energies. This is based upon atomic form factor, $f'$, and mass attenuation information. Values were obtained from http://www.nist.gov/pml/data/ffast/ for $f'$ and from http://www.nist.gov/pml/data/xcom/ for mass attenuation.
Gold: beta and delta for phase and analyzer gratings

Silicon: beta and delta for phase and analyzer gratings

Aluminum: beta and delta for phase and analyzer gratings

Iron: beta and delta for phase and analyzer gratings

Beryllium: beta and delta for phase and analyzer gratings

Lithium: beta and delta for phase and analyzer gratings

Oxygen: beta and delta for phase and analyzer gratings

Cobalt: beta and delta for phase and analyzer gratings

Li0.2CoO2: beta and delta for phase and analyzer gratings
Li0.8CoO2: beta and delta for phase and analyzer gratings

mu/rho mass attenuation coefficient: For mixtures, sum weighted components
f' atomic form function: For mixtures, sum the (i-th number density)x(i-th f1 atomic form function).

(i-th number density) = (atom fraction) (density) x AvogadroConstant/(molar mass)

Import, Interpolate Atomic Form Factor data (f1) from NIST:
http://www.nist.gov/pml/data/ffast/
Import, Interpolate mass attenuation data from NIST:
http://www.nist.gov/pml/data/xcom/

Unit Cell Volume

Calculate density, mass fractions, and number densities

xLithium = 0.8
strPlotLabel = "Li" <> ToString[xLithium] <> "CoO2"

unitCellVolume = (unitCellVolumeLinearFit /. x -> xLithium) Angstrom^3
unitCellZ = 3;
molarMass = {xLithium*ElementData["Li", "AtomicWeight"] +
    ElementData["Co", "AtomicWeight"] +
    2 + ElementData["O", "AtomicWeight"]} (Gram / Mole)
densityLixCoO2 = Convert[(unitCellZ x molarMass x AvogadroConstant) /
    unitCellVolume, Gram / (Centi Meter)^3]

0.8

Li0.8CoO2

97.0715 Angstrom^3

96.4848 Gram

Mole

4.95151 Gram

Centi^3 Meter^3
massFracLithium = (xLithium * ElementData["Li", "AtomicWeight"] (Gram / Mole)) / molarMass;
massFracCobalt = (ElementData["Co", "AtomicWeight"] (Gram / Mole)) / molarMass
massFracOxygen = (2 * ElementData["O", "AtomicWeight"] (Gram / Mole)) / molarMass;
0.610803

atomFracLithium = \frac{xLithium}{xLithium + 1 + 2};
atomFracCobalt = \frac{1}{xLithium + 1 + 2};
atomFracOxygen = \frac{2}{xLithium + 1 + 2};

numberDensityFormulaUnits = \frac{densityLixCoO2 \times AvogadroConstant}{molarMass}
3.09051 \times 10^{22}
Centi^{3} Meter^{3}

Calculate delta and beta at 20 keV

energyPhoton = 20 Kilo ElectronVolt;
waveLengthPhoton = Convert[\frac{\text{PlanckConstant SpeedOfLight}}{\text{Convert[energyPhoton, Joule]}}, Meter]
6.19921 \times 10^{-11} Meter

formFactor = atomFracLithium \times \text{interFormFactorLithium[First[energyPhoton]]} +
atomFracCobalt \times \text{interFormFactorCobalt[First[energyPhoton]]} +
atomFracOxygen \times \text{interFormFactorOxygen[First[energyPhoton]]}
muOverRho = (massFracLithium \times 
\text{interAttenuationWithoutLithium[First[energyPhoton]]} +
massFracCobalt \times \text{interAttenuationWithoutCobalt[First[energyPhoton]]} +
massFracOxygen \times \text{interAttenuationWithoutOxygen[First[energyPhoton]]}) \times (\text{Centi Meter})^2 / \text{Gram}
12.0583

17.0544 Centi^{2} Meter^{2}
Gram

delta = Convert[\frac{1}{2 \pi^2} \times \text{ClassicalElectronRadius} \times \text{waveLengthPhoton}^2 \times
\text{numberDensityFormulaUnits} \times \text{formFactor},]
6.42305 \times 10^{-7}
mu = \text{Convert}[\text{muOverRho} \times \text{densityLixCoO2}, \{\text{Micro Meter}\}^{-1}]

\text{beta} = \text{Convert}\left[\frac{\text{waveLengthPhoton} \times \mu}{4 \pi}\right],

0.00844451
\begin{array}{l}
\text{Meter Micro} \\
4.16582 \times 10^{-8}
\end{array}

\textbf{Define functions for Mu, Atomic Form Function, Beta, and Delta}

densityLixCoO2
4.95151 \text{ Gram} \\
\text{Centi}^3 \text{ Meter}^3

\text{funcMu}[energy_] := \text{Module}[\{\text{energyPhoton}, \text{muOverRho}\},
\text{energyPhoton} = \text{energy Kilo Electron Volt};
\text{muOverRho} = \{\text{massFracLithium} \times
\text{interAttenuationWithoutLithium}[\text{First}[\text{energyPhoton}]] +
\text{massFracCobalt} \times \text{interAttenuationWithoutCobalt}[\text{First}[\text{energyPhoton}]] +
\text{massFracOxygen} \times \text{interAttenuationWithoutOxygen}[\text{First}[\text{energyPhoton}]]\} \times \text{Centi Meter}^{-2} / \text{Gram};
\text{Convert}[\text{muOverRho} \times \text{densityLixCoO2}, \{\text{Micro Meter}\}^{-1}] // \text{Quiet}
]

\text{funcAtomicFormFunction}[\text{energy}_] := \text{Module}[\{\text{energyPhoton}\},
\text{energyPhoton} = \text{energy Kilo Electron Volt};
\text{atomFracLithium} \times \text{interFormFactorLithium}[\text{First}[\text{energyPhoton}]] +
\text{atomFracCobalt} \times \text{interFormFactorCobalt}[\text{First}[\text{energyPhoton}]] +
\text{atomFracOxygen} \times \text{interFormFactorOxygen}[\text{First}[\text{energyPhoton}]]
]

\text{funcDelta}[\text{energy}_] := \text{Module}[\{\text{energyPhoton}, \text{waveLengthPhoton}\},
\text{energyPhoton} = \text{energy Kilo Electron Volt};
\text{waveLengthPhoton} = \text{Convert}\left[\left(\text{PlanckConstant SpeedOfLight} / \text{Convert}[\text{energyPhoton}, \text{Joule}], \text{Meter}\right)\right];
\text{Convert}\left[\frac{1}{2 \pi} \text{ClassicalElectronRadius} \times \text{waveLengthPhoton}^2 \times \text{numberDensityFormulaUnits} \times \text{funcAtomicFormFunction}[\text{energy}_],\right]
]

\text{funcBeta}[\text{energy}_] := \text{Module}[\{\text{energyPhoton}, \text{waveLengthPhoton}, \mu\},
\text{energyPhoton} = \text{energy Kilo Electron Volt};
\text{waveLengthPhoton} = \text{Convert}\left[\left(\text{PlanckConstant SpeedOfLight} / \text{Convert}[\text{energyPhoton}, \text{Joule}], \text{Meter}\right)\right];
\mu = \text{funcMu}[\text{energy}_];
\text{Convert}\left[\frac{\text{waveLengthPhoton} \times \mu}{4 \pi},\right]
]
funcAtomicFormFunction[energy = 5]
funcMu[energy]
funcDelta[5]
funcBeta[energy]
11.7811
0.0537176
Meter Micro
0.0000100406
1.05999 \times 10^{-6}
funcAtomicFormFunction[1.1]
11.2345
funcDelta[1.1]
0.000197826

Plot $\mu$

gXCOMInterpolation = LogPlot[First[funcMu[energykeV]],
{energykeV, 2, 100}, PlotRange -> All, GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
{["$\mu/(\text{micron}^{-1})$", ""], ["keV", strPlotLabel]}, ImageSize -> 500]

Plot atomic form factor
(atomFracLithium * 3 + atomFracCobalt * 27 + atomFracOxygen * 8)
11.9474
gFormFactor = Plot[funcAtomicFormFunction[energykeV],
{energykeV, 2, 100}, PlotRange -> All, GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18],
FrameLabel -> {"f_1 (e/atom)", ""}, {"Energy (keV)", strPlotLabel}];
Plot beta and delta

Needs["PlotLegends")
gDeltaBeta = LogLogPlot[{10^6 funcDelta[energykeV], 10^6 funcBeta[energykeV]}, 
  {energykeV, 2, 100}, 
  PlotStyle -> {Directive[Thick, Blue, Dashed], 
    Directive[Thick, Black]}, 
  PlotRange -> {Full, {0.0001, 100}}, PlotPoints -> 100, 
  GridLines -> Automatic, 
  Frame -> True, FrameStyle -> Directive[18], 
  FrameLabel -> {"\(\delta\), \(\beta\) (10^6)", ""}, {"Energy (keV)", strPlotLabel}], 
  PlotLegend -> {Style["\(\delta\)", 20], Style["\(\beta\)", 20]}, LegendShadow -> None, 
  LegendPosition -> {0.4, 0.3}, LegendSize -> {0.3, 0.2}, 
  ImageSize -> 500] // Quiet

![Graph showing the relationship between energy (keV) and \(\delta\), \(\beta\) (10^6) for Li0.8CoO2]
How thick is the LixCoO2 structure for $\pi$-phase shift and $T=0.05$ (absorbance=2.996) for 25 keV

```
energy = 25
energyPhoton = energy Kilo ElectronVolt;
transmission = 0.05;
absorbance = -Log[E, transmission]
delta = funcDelta[energy]
beta = funcBeta[energy]
mu = funcMu[energy]

waveLengthPhoton = Convert[\frac{PlanckConstant SpeedOfLight}{Convet[energyPhoton, Joule]}, Pico Meter]
25
2.99573
4.10468 \times 10^{-7}
1.77395 \times 10^{-8}
0.00449495
Meter Micro
49.5937 Meter Pico

thicknessPhaseMicron = Convert[\frac{\pi \text{ waveLengthPhoton}}{2 \pi \delta \text{ delta}}, \text{ Micro Meter}]
60.4112 Meter Micro

thicknessAnalyzerMicron = \frac{-\text{ Log}[E, transmission]}{\mu}
thicknessAnalyzerMicron \times \mu;
E^\left(-1 \times \text{ thicknessAnalyzerMicron} \times \mu\right);
666.466 \text{ Meter Micro}
```
Plot LixCuO2 thickness for π-phase shift and T=0.10 structures

funcTransmission2[energy_] := Module[{},
          energyPhoton = energy Kilo ElectronVolts;
          mu = funcMu[energy];
          Convert[E^(-mu * 100 Micro Meter), ]
          funcPhaseShift2[energy_] := Module[{},
          delta = funcDelta[energy];
          waveLengthPhoton =
          Convert[PlanckConstant SpeedOfLight, Pico Meter];
          Convert[energyPhoton, Joule]
          Convert\[\frac{2 \pi \text{ delta } * 100 \text{ Micro Meter}}{\text{ waveLengthPhoton}}\], ]
energy = 85
funcTransmission[energy]
funcPhaseShift[energy]

0.981939
1.52003
Needs["PlotLegends""]
gDeltaBeta =
Plot[{funcTransmission2[energykeV]}, {energykeV, 0, 100},
PlotStyle ->
{Directive[Thick, Blue, Dashed], Directive[Thick, Black]},
PlotRange -> {Full, {0, 1}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
{"Transmission", ""}, {"Energy (keV)", strPlotLabel},
PlotLegend -> {Style["Li0.8CoO2", 18], LegendShadow -> None,
LegendPosition -> {0.2, 0.1}, LegendSize -> {0.5, 0.2},
LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

![Graph showing transmission vs. energy for Li0.8CoO2](image-url)
Needs["PlotLegends"]
gDeltaBeta =
LogLogPlot[{funcTransmission2[energykeV]}, {energykeV, 10, 100},
PlotStyle ->
{Directive[Thick, Blue, Dashed], Directive[Thick, Black]},
PlotRange -> {Full, {0, 100}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
{{"Transmission", ""}, {"Energy (keV)", strPlotLabel}},
PlotLegend -> {Style["Li0.8CoO2", 18], LegendShadow -> None,
LegendPosition -> {0.2, 0.1}, LegendSize -> {0.5, 0.2},
LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

![Graph showing transmission as a function of energy for Li0.8CoO2](image-url)
<<PlotLegends``

gDeltaBeta = Plot[funcPhaseShift2[energykeV], {energykeV, 0, 100},
PlotStyle ->
{Directive[Thick, Blue, Dashed], Directive[Thick, Black]},
PlotRange -> {Full, {0, 100}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18],
FrameLabel -> {{"PhaseShift", ""}, {"Energy (keV)", strPlotLabel}},
PlotLegend -> {Style["Li0.8CoO2", 18]},
LegendShadow -> None, LegendPosition -> {0.2, 0.1},
LegendSize -> {0.5, 0.2}, LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{phase_shift.png}
\caption{Phase shift for Li0.8CoO2 as a function of energy.}
\end{figure}
Needs["PlotLegends""]

gDeltaBeta = LogLogPlot[{{funcPhaseShift2[energykeV]}, {energykeV, 10, 100},
PlotStyle -> {Directive[Thick, Blue, Dashed], Directive[Thick, Black]},
PlotRange -> {Full, {0, 100}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18],
FrameLabel -> {{"PhaseShift", ""}, {"Energy (keV)", strPlotLabel}},
PlotLegend -> {Style["Li0.8CoO2", 18]},
LegendShadow -> None, LegendPosition -> {0.2, 0.1},
LegendSize -> {0.5, 0.2}, LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

[[Fig. 1]]

Li0.8CoO2

![Phase Shift vs. Energy Graph](figures/thick_phase_analyz_Li0.8CoO2.pdf)

Export[NotebookDirectory[] <> "figures/thick_phase_analyz_" <> strPlotLabel <> ".pdf", gDeltaBeta, "PDF"]
/Users/tomo4/Desktop/figures/thick_phase_analyz_Li0.8CoO2.pdf
Needs["PlotLegends"]
strPlotLabel = "LixCoO2";
gDeltaBeta =
Plot[
{funcPhaseShift1[energykeV], funcPhaseShift2[energykeV]},
{energykeV, 0, 100},
PlotStyle -> {Directive[Thick, Blue],
   Directive[Thick, Yellow, Dotted]},
PlotRange -> {Full, {0.1, 50}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18],
FrameLabel -> {{"Phasshift",""}, {"Energy (keV)", strPlotLabel}},
PlotLegend -> {Style["x=0.2", 18], Style["x=0.8", 18]},
LegendShadow -> None, LegendPosition -> (0.2, 0.1),
LegendSize -> {0.5, 0.2}, LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

Export[NotebookDirectory[] <> "figures/thick_phase_analyz_" <>
strPlotLabel <> ".pdf", gDeltaBeta, "PDF"]
/Users/tomo4/ Desktop/figures/thick_phase_analyz_LixCoO2.pdf
Needs["PlotLegends\"]

strPlotLabel = "LixCoO2";

gDeltaBeta = LogLogPlot[
{funcTransmission1[energykeV],
funcTransmission2[energykeV]},
{energykeV, 10, 100},
PlotStyle -> {Directive[Thick, Blue],
Directive[Thick, Yellow, Dotted]},
PlotRange -> {Full, {0.1, 1}}, PlotPoints -> 100,
GridLines -> Automatic,
Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
{("Transmission", ""), {"Energy (keV)", strPlotLabel}},
PlotLegend -> {Style["x=0.2", 18], Style["x=0.8", 18]},
LegendShadow -> None, LegendPosition -> {0.2, 0.1},
LegendSize -> {0.5, 0.2}, LegendTextOffset -> {-0.5, 0},
ImageSize -> 500] // Quiet

Export[NotebookDirectory[] <> "figures/Transmission_analyz_" <>
strPlotLabel <> ".pdf", gDeltaBeta, "PDF"]
Appendix C: Neutron Diffraction Mathematica Code

The following Mathematica code was used to analyze a time-of-flight neutron diffraction experiment performed at Oak Ridge National Lab, SNS VULCAN. The samples of interest were lithium ion batteries. The code initially defines functions for the instrument, followed by import of raw time-of-flight data for vanadium and cerium oxide references. Diffraction spectra for the references are compared with raw battery data and thus diffraction plots of fresh and worn, charged and discharged batteries are made.
v2p092_Vulcan_PeakFit.nb
IPTS-2946

Needs["Units""]
Needs["PhysicalConstants""]
pathBatteryData =
  pathVulcanData = NotebookDirectory[] <> "VULCAN_data/";
correctTwoThetaAngle = 47.7 Degree;
pathLengthVulcan = 43.754 Meter;

Define functions

flightTimeToWavelength[time_, pathLength_] :=
  Module[{velocity, wavelength},
    velocity = pathLength
    time
    wavelength = Convert[ PlanckConstant
    NeutronMass × velocity
    wavelength]
  ]

wavelengthToFlightTime[wavelength_, pathLength_] :=
  Module[{velocity, time},
    velocity = Convert[ PlanckConstant
    NeutronMass × wavelength
    time = pathLength
    velocity
    time]
  ]

flightTimeToWavelength[0.011 Second, pathLengthVulcan]
0.994569 Angstrom

Initial plots of time-of-flight data: Vanadium, Cerium Oxide, and Charged, Fresh Battery

4576 Cerium Oxide measured early Friday morning 4AM to 7AM with 5x10x5, 30Hz, 2Å center
4577 Vanadium measured early Friday morning 7AM to 9AM with 5x10x5, 30Hz, 2Å center
4579 charged, fresh battery #2740

Read 4577 Vanadium prof.dat data

pathVulcanData <> "4577.prof.dat"
/Users/tomo/Dropbox/Battery_manuscript/Mathematica/VULCAN_data/4577.
prof.dat
dataString = Import[pathVulcanData <> "4577.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4
lineStart = 5; lineEnd = Dimensions[dataString][[1]];
dataVanadium = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex <= lineEnd, lineIndex++,
   line = dataString[[lineIndex]];
dataVanadium[[index, All]] = line;
index++;
];
gVanadiumspacing = ListPlot[
   Transpose[{{dataVanadium[[All, 1]], dataVanadium[[All, 2]]}}],
   Joined -> True, PlotStyle -> Black,
   PlotRange -> {{0, 50000}, {0, 40000}},
   ImageSize -> 600, AxesOrigin -> {0.6, 0},
   Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
   {{"counts", ""}, "time of flight/μs", "vanadium (with peaks)"},
   Epilog -> Inset[Graphics[Text[Style["30 Hz, 2 Å", 18]]],
   {40000, 35000}]]

4576 - Cerium Oxide, 5×10×5, 30Hz, 2Å center

4579 - charged, fresh

Intensity correction for 5×10×10 mm^3 based on vanadium scattering
Cerium Oxide peak (via 2-theta angle) calibration:

4576 Cerium Oxide measured early Friday morning 4AM to 7AM with 5x10x5, 30Hz, 2Å center
The cerium oxide peak calibration result is:  \[ \text{correctTwoThetaAngle} = 47.7 \text{ Degree} \]

4576 - Cerium Oxide, 5x10x5, 30Hz, 2Å center

dataString = Import[pathVulcanData <> "4576.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4
lineStart = 5; lineEnd = Dimensions[dataString][[1]];
dataCeriumOxide = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex ≤ lineEnd, lineIndex++,
    line = dataString[[lineIndex]];
    dataCeriumOxide[[index, All]] = line;
    index++;
];
timeTOF = dataCeriumOxide[[All, 1]];
wavelengthTOF =
    Table[FlightTimeToWavelength[(timeTOF[[i]]), Micro Second,
            pathLengthVulcan] / (Angstrom), {i, Length[timeTOF]}];

dSpacingTOF = \[ \frac{1 \times \text{wavelengthTOF}}{2 \sin[\text{correctTwoThetaAngle}]} \]
dataCeriumOxide[[All, 1]] = dSpacingTOF;
y = dataCeriumOxide[[All, 2]];
y = 100 * y / Total[y];
x = dataCeriumOxide[[All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
            y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataCeriumOxide[[All, 2]] = y;
gCeO2dspacing = ListPlot[
    Transpose[{dataCeriumOxide[[All, 1]], dataCeriumOxide[[All, 2]]}],
    Joined -> True, PlotStyle -> Black, PlotRange -> {{0.6, 2.5}, {0, 4}},
    ImageSize -> 600, AxesOrigin -> {0.6, 0},
    Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
    {"intensity/arb. units", ""}, {"d-spacing/Å", "cerium oxide"}],
    Epilog -> Inset[Graphics[Text[Style["30 Hz, 2 Å", 18]], (2.2, 3.3)]]

2487

182
Find Cerium Oxide peak positions in the spectrum

interpCeriumOxide =
  Interpolation[ Transpose[{x, y}], InterpolationOrder \[Rule] 3 ];
Clear[z]; threshold = 0.1;
answer = Union[Table[
  FindMaximum[
    {interpCeriumOxide[z], Min[x] < z < Max[x]}, {z, zInit},
    MaxIterations \[Rule] 500, PrecisionGoal \[Rule] 0.0001] // Quiet,
    {zInit, 0.5, 0.5, 0.02}]
  ];

peakData =
  Table[{z /. answer[[i, 2]], answer[[i, 1]]}, {i, Length[answer]}];

peakData = DeleteDuplicates[Round[peakData * 10^5] / 10^5 // N;

index =
  Flatten[Position[peakData[[All, 2]], p_?(## \[GreaterEqual] threshold &)]]; 

peakData = peakData[[index, All]];

index = Flatten[Position[peakData[[All, 1]], p_?(## \[GreaterEqual] 0.5 &)]]; 

peakData = peakData[[index, All]];

index = Reverse[Ordering[peakData[[All, 1]]]]; 

peakData = peakData[[index, All]];

peakData[[All, 2]] =
  Round[100 * peakData[[All, 2]] / Max[peakData[[All, 2]]]];

TableForm[peakData, TableHeadings \[Rule] {None, {"d-spacing", "rel. int."}}]

d-spacing rel. int.  
1.9141 100  
1.63245 13  
1.3537 22  
1.21086 12  
1.10534 51  
0.95725 19  
0.91539 7  
0.90258 7  
0.85627 27  
0.81637 5  
0.78158 9  
0.75822 6  
0.72358 33  
0.70501 7  
0.67683 5  
0.66123 4  
0.63811 18  
0.62515 5  
0.59927 3  
0.58758 3  
0.57712 11  
0.55257 11  
0.50881 5  

184
Import Cerium Oxide peak positions from cif file (CrystalDiffact)

dataString = Import[pathVulcanData <> "CeO2_5.4117A.txt", "Data"];
numberOfReflections = Dimensions[dataString][[1]] - 8;
lineStart = 9; lineEnd = Dimensions[dataString][[1]];
dataCeriumOxideReflections =
  ConstantArray[0, {numberOfReflections, 5}];
index = 1;
For[lineIndex = lineStart, lineIndex <= lineEnd, lineIndex++,
  line = dataString[[lineIndex]];
  temp = StringSplit[line];
  {h, k, l} = ToExpression[temp[[1, 3 ;; 5]]];
  dhkl = ToExpression[temp[[1, 6]]];
  relativeIntensity = ToExpression[temp[[1, 9]]];
  dataCeriumOxideReflections[[index, All]] =
    {h, k, l, dhkl, relativeIntensity};
  index++;
];
TableForm[dataCeriumOxideReflections,
  TableHeadings -> {Automatic, {"h", "k", "l", "d(hkl)", "rel. int."}}]
Compare Cerium Oxide (expt. to literature)

Fresh Battery #2740:

4579 charged, fresh battery #2740
4580 discharging; split into 5 minute windows
4581 discharged by -401 Coulombs

4580 - explore timing code
4580 - read V,I and compute Coulombs at 5 minute intervals
4579 - charged, fresh

dataString = Import[pathVulcanData <> "4579.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4
lineStart = 5; lineEnd = Dimensions[dataString][[1]];
dataCharged = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex <= lineEnd, lineIndex++,
  line = dataString[[lineIndex]];
  dataCharged[[index, All]] = line;
  index++;
];
timeTOF = dataCharged[[All, 1]];
wavelengthTOF =
  Table[flightTimeToWavelength[
    (timeTOF[[i]] Micro Second, pathLengthVulcan] / (Angstrom), {i, Length[timeTOF]]};
  dSpacingTOF = 1 \times \text{wavelengthTOF} \\
  2 \text{Sin[correctTwoThetaAngle]}
  dataCharged[[All, 1]] = dSpacingTOF;
y = dataCharged[[All, 2]];
y = 100 \times y / Total[y];
x = dataCharged[[All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
  y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataCharged[[All, 2]] = y;
gCharged =
  ListPlot[Transpose[
    {dataCharged[[All, 1]], dataCharged[[All, 2]]}],
  Joined -> True, PlotStyle -> (Black, Thick),
  PlotRange -> {{0.5, 2.5}, {-0.15, 0.30}},
  ImageSize -> 600, AxesOrigin -> {0.5, -0.15},
  Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
  {"Intensity/arb. units", ""}, {"d-spacing/Å", "fresh battery"}],
  Epilog -> Inset[Graphics[Text[Style["charged", 18]], {2.2, 0.2}]]
Dimensions[dataCharged[[All, 1 ;; 2]]]
dataCharged[[1 ;; 10, 1 ;; 2]]
Export[NotebookDirectory[] <> "FreshBattery_Charged_Run4579.xls",
dataCharged[[All, 1 ;; 2]], "XLS"]
{2487, 2}

{{0.30561, 0}, {0.305915, 0}, {0.306221, 0},
{0.306528, 0}, {0.306834, 0}, {0.307141, 0},
{0.307448, 0}, {0.307756, 0}, {0.308063, 0}, {0.308371, 0}}

/Users/tomo/Dropbox/Battery_manuscript/Mathematica/
FreshBattery_Charged_Run4579.xls
dataString = Import[pathVulcanData <> "4581.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4
dataDischarged = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex ≤ lineEnd, lineIndex++,
   If[lineIndex > Dimensions[dataString][[1]], Break[]];
   line = dataString[[lineIndex]];
   dataDischarged[[index, All]] = line;
   index++;
];
timeTOF = dataDischarged[[All, 1]];
wavelengthTOF =
   Table[flightTimeToWavelength[(timeTOF[[i]]) Micro Second,
      pathLengthVulcan] / (1 Angstrom), {i, Length[timeTOF]}]
   1 x wavelengthTOF;
   2 Sin[correctTwoThetaAngle]
dataDischarged[[All, 1]] = dSpacingTOF;
y = dataDischarged[[All, 2]];
y = 100 * y / Total[y];
x = dataDischarged[[All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
   y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataDischarged[[All, 2]] = y;
gDischarged = ListPlot[Transpose[
   {dataDischarged[[All, 1]], -0.10 + dataDischarged[[All, 2]]}],
   Joined -> True, PlotStyle -> {Gray, Thick},
   PlotRange -> {{0.5, 2.5}, {-0.15, 0.30}},
   ImageSize -> 600, AxesOrigin -> {0.5, -0.15},
   Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
   {"Intensity/arb. units", ""}, {"d-spacing/Å", "fresh battery"}],
   Epilog -> Inset[Graphics[Text[Style["discharged", 18]], {2.2, 0.2}]]
fresh battery

intensity/arb. units

d–spacing/Å

fresh battery

intensity/arb. units

d–spacing/Å
Dimensions[dataDischarged[[All, 1 ;; 2]]]
dataDischarged[[1 ;; 10, 1 ;; 2]]
Export[NotebookDirectory[] <> "FreshBattery_Discharged_Run4581.xls",
dataDischarged[[All, 1 ;; 2]], "XLS"]

{2487, 2}

{{0.30561, 0}, {0.305915, 0}, {0.306221, 0}, {0.306528, 0}, {0.306834, 0}, {0.307141, 0}, {0.307448, 0}, {0.307756, 0}, {0.308063, 0}, {0.308371, 0}}

/Users/tomo/Dropbox/Battery_manuscript/Mathematica/
FreshBattery_Discharged_Run4581.xls
4580 - discharging, fresh

Length[listCoulomb];
Length[listTimes];
temp = Table[StringReplace[ToString[listTimes[[i, 4 ;; 6]]],
  {"" → "", " " → ":", "" → ":"}], {i, Length[listTimes]}];
TableForm[Transpose[{temp, listCoulomb}],
  TableHeadings → {Automatic, None}]

1     6:15:10  30.8199
2     6:20:10  30.2155
3     6:25:10  30.2149
4     6:30:10  30.2155
5     6:35:10  30.2155
6     6:40:10  30.2149
7     6:45:10  30.2155
8     6:50:10  30.2155
9     6:55:10  30.2155
10    7:00:10  30.2155
11    7:05:10  30.2155
12    7:10:10  30.2155
13    7:15:10  30.2155
14    7:20:10  9.9061
15    7:21:34  1.18564
numberTimeWindows = 15; gDischarging = {};
dataDischarging = ConstantArray[{{0, 0, 0}, {numberTimeWindows, numberOfPoints}}];
For[i = 1, i \leq numberTimeWindows, i++,
dataString = Import[
  pathVulcanData <> "/4580/" <> ToString[i] <> ".prof.dat", "Data"];
index = 1;
For[lineIndex = lineStart, lineIndex \leq lineEnd, lineIndex++,
  If[lineIndex > Dimensions[dataString][[1]], Break[]];
  line = dataString[[lineIndex]];
dataDischarging[[i, index, All]] = line;
index++;
];
timeTOF = dataDischarging[[i, All, 1]];
wavelengthTOF =
  Table[flightTimeToWavelength[timeTOF[[i]]] Micro Second,
    pathLengthVulcan] / (1 Angstrom), {i, Length[timeTOF]}];
daSpacingTOF = 1 \times wavelengthTOF / (2 Sin[correctTwoThetaAngle]);
dataDischarging[[i, All, 1]] = daSpacingTOF;
y = dataDischarging[[i, All, 2]];
y = 100 \times y / Total[y];
x = dataDischarging[[i, All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
  y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataDischarging[[i, All, 2]] = y;
gDischarging =
  Append[gDischarging, ListPlot[Transpose[{x, y}], Joined -> True, 
    PlotStyle -> {Black, Thick}, PlotRange -> {{0.6, 2.5}, {0, 0.35}},
    ImageSize -> 600, AxesOrigin -> {0.6, 0},
    Frame -> True, FrameStyle -> Directive[18],
    FrameLabel -> {"intensity/arb. units", ""},
    {"d-spacing/Å", "fresh battery"}],
    Epilog -> Inset[Graphics[Text[Style[
        ToString[listCoulombCumulative[[i]]] <> " C", 
        18]], {2.0, 0.3}]] ];
];
gAll = Flatten[{gCharged, gDischarging, gDischarged}];
ListAnimate[gAll];
6.709 / 4
1.67725

Table[
  Export[NotebookDirectory[] <> 
    "freshbattery_discharging_" <> ToString[i] <> ".jpg", gAll[[i]]], 
  {i, Length[gAll]}];
Show[{gCharged, gDischarged}, Epilog -> Inset[ Graphics[Text[Style["charged (black)\n discharged (gray)"", 16]], {2.2, 0.25}]]]

fresh battery

charged (black)
discharged (gray)

intensity/arb. units

d-spacing/Å

fresh battery

cu 046
graphite
graphite

Cu 111
Al

Li0.5CoO
LiGraphite
Li(NiCoMn)O2

Cu 002
Cu 002
Cu 002

LiGraphite
Graphite
Graphite

0.5 1.0 1.5 2.0 2.5

-0.1 0.0 0.1 0.2 0.3

d-spacing/Å

intensity/arb. units

fresh battery

charged (black)
discharged (gray)
4579, 4580, 4581 - integration of selected diffraction peaks

4579 charged, fresh battery #2740: blue peaks = 2.145(0.02), 2.018(0.01), 1.847(0.04), 1.764(0.03), 1.575(0.01), 1.45(0.01),
4580 discharging; split into 5 minute windows
4581 discharged by -401 Coulombs: black peaks = 2.007(0.01), 1.683(0.04), 1.565(0.02), 1.222(0.01), 0.993(0.01)

peaksBlue = {{2.145, 0.02}, {2.018, 0.01},
{1.847, 0.04}, {1.764, 0.03}, {1.575, 0.01}, {1.45, 0.01}};
peaksBlack = {{2.007, 0.01}, {1.683, 0.04},
{1.565, 0.02}, {1.222, 0.01}, {0.993, 0.01}};

For[i = 1, i <= Length[peaksBlue], i++,
{center, fullwidth} = peaksBlue[[i]];
areaPeaksBlue[i, 1] = ToString[NumberForm[center, 3]] <>
" (" <> ToString[fullwidth] <> ")";

For[i = 1, i <= Length[peaksBlack], i++,
{center, fullwidth} = peaksBlack[[i]];
areaPeaksBlack[i, 1] = ToString[NumberForm[center, 3]] <>
" (" <> ToString[fullwidth] <> ")";

x = dataCharged[[All, 1]]; y = dataCharged[[All, 2]];
indexPeak = Flatten[Position[x, p_ ? ((# > (center - fullwidth / 2)) &&
(# < (center + fullwidth / 2))) &]];

For[indexCoulomb = 1, indexCoulomb <= Length[axisCoulomb], indexCoulomb++,
Which[indexCoulomb = 1, y = dataCharged[[All, 2]];
indexCoulomb > 1 && indexCoulomb < 17, Module[{i},
1 = indexCoulomb - 1; y = dataDischarging[[i, All, 2]];
indexCoulomb = 17, y = dataDischarging[[All, 2]]];

areaPeakPlusBaseline = Total[Take[y, {indexMin, indexMax}] -
(xMax - xMin) / (indexMax - indexMin + 1)];
areaPeak = areaPeakPlusBaseline - areaBaseline;
(* Print[{i, indexCoulomb, areaPeak}]; *)}
areaPeaksBlue[[i, indexCoulomb + 1]] = areaPeak;
];
]
For[i = 1, i ≤ Length[peaksBlue], i++,
{center, fullwidth} = peaksBlue[[i]]; x = dataCharged[[All, 1]]; y = dataCharged[[All, 2]]; indexPeak = Flatten[Position[x, p_?((p > ((center - fullwidth/2)) && (p < (center + fullwidth/2)) &))]]
{indexMin, indexMax} = {First[indexPeak], Last[indexPeak]};
gFillBlue[i] = ListPlot[
 Transpose[{{x[[indexMin ;; indexMax]], y[[indexMin ;; indexMax]]}, AxesOrigin -> (0, 0), Joined -> True, Filling -> Axis, FillingStyle -> Directive[Black, Opacity[0.5]]}];
];

For[i = 1, i ≤ Length[peaksBlack], i++,
{center, fullwidth} = peaksBlack[[i]]; areaPeaksBlack[[i, 1]] = ToString[NumberForm[center, 3]] <> "(" <> ToString[fullwidth] <> ")"; x = dataDischarged[[All, 1]]; y = dataDischarged[[All, 2]]; indexPeak = Flatten[Position[x, p_?((p > ((center - fullwidth/2)) && (p < (center + fullwidth/2)) &))]]
{indexMin, indexMax} = {First[indexPeak], Last[indexPeak]}; {xMin, xMax} = {x[[indexMin]], x[[indexMax]]}; {yMin, yMax} = {y[[indexMin]], y[[indexMax]]}; areaBaseline = Mean[{yMin, yMax}] × (xMax - xMin);
For[indexCoulomb = 1,
indexCoulomb ≤ Length[axisCoulomb], indexCoulomb++,
Which[indexCoulomb = 1, y = dataCharged[[All, 2]], indexCoulomb > 1 && indexCoulomb < 17, Module[{i},
   i = indexCoulomb - 1; y = dataDischarging[[i, All, 2]]];
indexCoulomb = 17, y = dataDischarged[[All, 2]]];
areaPeakPlusBaseline = Total[Take[y, {indexMin, indexMax}]] × (xMax - xMin)/(indexMax - indexMin + 1);
areaPeak = areaPeakPlusBaseline - areaBaseline;
(* Print[{i, indexCoulomb, areaPeak}]; *)
areaPeaksBlack[[i, indexCoulomb + 1]] = areaPeak;
];
]

For[i = 1, i ≤ Length[peaksBlack], i++,
{center, fullwidth} = peaksBlack[[i]]; x = dataDischarged[[All, 1]]; y = dataDischarged[[All, 2]]; indexPeak = Flatten[Position[x, p_?((p > ((center - fullwidth/2)) && (p < (center + fullwidth/2)) &))]]
{indexMin, indexMax} = {First[indexPeak], Last[indexPeak]};
gFillBlack[i] = ListPlot[
 Transpose[{{x[[indexMin ;; indexMax]], y[[indexMin ;; indexMax]]}, AxesOrigin -> (0, 0), Joined -> True, Filling -> Axis, FillingStyle -> Directive[Blue, Opacity[0.5]]}];
];
Show[{gCharged, Table[gFillBlue[i], {i, Length[peaksBlue]}], gDischarged, Table[gFillBlack[i], {i, Length[peaksBlack]}]}];
\[ x = \text{axisCoulomb}; \quad y = \text{areaPeaksBlue}[[\text{All}, 2 ;; (\text{Length}[x] + 1)]]; \]
\[
gPeakBlue = \text{ListPlot}[\text{Table}[\text{Transpose}[[x, y[[i, \text{All}]]]], \{i, \text{Length}[\text{peaksBlue}]\}], \text{PlotStyle} \rightarrow \text{Blue}, \text{Joined} \rightarrow \text{True}, \text{PlotMarkers} \rightarrow \{"1", "2", "3", "4", "5", "6"\}, \text{PlotRange} \rightarrow \{(-100, 450), \text{All}\}, \text{ImageSize} \rightarrow 600, \text{AxesOrigin} \rightarrow \{0.0, 0\}, \text{Frame} \rightarrow \text{True}, \text{FrameStyle} \rightarrow \text{Directive}[18], \text{FrameLabel} \rightarrow \{\{"peak areas", ""\}, \{"discharging/Coulomb", "fresh battery"\}\};
\]
\[\text{For}[i = 1, i \leq \text{Length}[\text{peaksBlue}], i++, \]
\[\text{xText} = -90; \quad \text{yText} = \text{Take}[\text{areaPeaksBlue}[[i, \text{All}]], \{2\}][[1]]; \quad \text{gTextBlue}[i] = \]
\[\text{Graphics}[\text{Text}[\text{Style}[\text{areaPeaksBlue}[[i, 1]] <> " \text{"} Å", 12, \text{Blue}], \]
\[\quad \text{xText}, \text{yText}], \{-1, 0\}]]; \quad \]
\[
\text{Show}[\{\text{gPeakBlue}, \text{Table}[\text{gTextBlue}[i], \{i, \text{Length}[\text{peaksBlue}]\}] \}]
\]
\[ x = \text{axisCoulomb}; \quad y = \text{areaPeaksBlack}[[\text{All}, 2 ;; (\text{Length}[x] + 1)]]; \]
\[
gPeakBlack = \text{ListPlot}[\text{Table}[\text{Transpose}[[x, y[[i, \text{All}]]]], \{i, \text{Length}[\text{peaksBlack}]\}], \text{PlotStyle} \rightarrow \text{Black}, \]
\[\text{Joined} \rightarrow \text{True}, \text{PlotMarkers} \rightarrow \{"1", "2", "3", "4", "5"\}, \]
\[\text{PlotRange} \rightarrow \{(0, 500), \text{All}\}, \text{ImageSize} \rightarrow 600, \text{AxesOrigin} \rightarrow \{0.0, 0\}, \]
\[\text{Frame} \rightarrow \text{True}, \text{FrameStyle} \rightarrow \text{Directive}[18], \text{FrameLabel} \rightarrow \{\{"peak areas", ""\}, \{"discharging/Coulomb", "fresh battery"\}\};
\]
\[\text{For}[i = 1, i \leq \text{Length}[\text{peaksBlack}], i++, \]
\[\text{xText} = 420; \quad \text{yText} = \text{Last}[\text{areaPeaksBlack}[[i, \text{All}]]]; \quad \text{gTextBlack}[i] = \]
\[\text{Graphics}[\text{Text}[\text{Style}[\text{areaPeaksBlack}[[i, 1]] <> " \text{"} Å", 12, \text{Black}], \]
\[\quad \text{xText}, \text{yText}], \{-1, 0\}]]; \quad \]
\[
\text{Show}[\{\text{gPeakBlack}, \text{Table}[\text{gTextBlack}[i], \{i, \text{Length}[\text{peaksBlack}]\}] \}]
\]

198
Combination plot of charged, discharged with selected peak picking

```
peaksBlue = {{1.012, 0.1301}, {1.035, 0.1241}, {1.073, 0.123},
             {1.154, 0.1328}, {1.17, 0.1404}, {1.239, 0.1474}, {1.365, 0.1034},
             {1.405, 0.123}, {1.45, 0.1056}, {1.573, 0.1062}, {1.763, 0.1279},
             {1.846, 0.1105}, {2.017, 0.1149}, {2.144, 0.09313}};
peaksBlack = {{0.9947, 0.1295}, {1.123, 0.1322}, {1.156, 0.1491},
               {1.231, 0.1599}, {1.434, 0.1094}, {1.556, 0.1029},
               {1.683, 0.1534}, {2.005, 0.1127}, {2.032, 0.09747}};
Clear[gPeakLabelBlue]
For[i = 1, i ≤ Length[peaksBlue], i++,
gPeakLabelBlue[i] = Module[{},
  text = ToString[peaksBlue[[i, 1]]];
  coords =
    peaksBlue[[i, All]] + {0, 0.02 * Sign[peaksBlue[[i, 2]]]};
  Graphics[Text[Style[text, 10, Blue], coords,
                 {0, 0}, {0, 1}]]];
Clear[gPeakLabelBlack]
For[i = 1, i ≤ Length[peaksBlack], i++,
gPeakLabelBlack[i] = Module[{},
  text = ToString[peaksBlack[[i, 1]]];
  coords =
    peaksBlack[[i, All]] + {0, 0.08 * Sign[peaksBlack[[i, 2]]]};
  Graphics[Text[Style[text, 10, Black],
                 coords, {0, 0}, {0, 1}]]];
Show[{gCharged, gDischarged,}
     Table[gPeakLabelBlue[i], {i, Length[peaksBlue]}],
     Table[gPeakLabelBlack[i], {i, Length[peaksBlack]}]],
Epilog -> Inset[Graphics[Text[
     Style["charged (blue)\n discharged (black) ", 16]], {2.1, 0.3}]
```

![Graph showing d-spacing vs intensity](image)

<table>
<thead>
<tr>
<th>d-spacing/Å</th>
<th>fresh battery</th>
<th>discharged (black)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9947</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.012</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.035</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.073</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.239</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.405</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.573</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.846</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.144</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4579, 4580, 4581 - save to hdf5 file

Worn Battery #3319:

- 4573 charged, worn battery #3319
- 4574, 4575 discharging; split into 5 minute windows
- 4576 cerium oxide
- 4577 vanadium
- 4578 discharged by -100 Coulombs

4574, 4575 - explore timing code

4574, 4575 - read V, I and compute Coulombs at 5 minute intervals. Make table of Run number, split number, time-of-day, index, and coulombs of discharge in a split.
dataString = Import[pathVulcanData <> "4573.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4
lineStart = 5; lineEnd = Dimensions[dataString][[1]];
dataChargedWorn = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex <= lineEnd, lineIndex++,
    line = dataString[[lineIndex]];
    dataChargedWorn[[index, All]] = line;
    index++;
];
timeTOF = dataChargedWorn[[All, 1]];
wavelengthTOF =
    Table[flightTimeToWavelength[{timeTOF[[i]]} Micro Second,
        pathLengthVulcan] / (Angstrom), {i, Length[timeTOF]}];
dSpacingTOF = 
    1 x wavelengthTOF /
    2 Sin[c correctTwoThetaAngle];
dataChargedWorn[[All, 1]] = dSpacingTOF;
y = dataChargedWorn[[All, 2]];
y = 100 * y / Total[y];
x = dataChargedWorn[[All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
        y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataChargedWorn[[All, 2]] = y;
gChargedWorn = ListPlot[
    Transpose[{dataChargedWorn[[All, 1]], dataChargedWorn[[All, 2]]}],
    Joined -> True, PlotStyle -> {Blue, Thick},
    PlotRange -> {{0.6, 2.5}, {0, 0.25}},
    ImageSize -> 600, AxesOrigin -> {0.6, 0},
    Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
    {"intensity/arb. units", ""}, {"d-spacing/Å", "worn battery"}],
    Epilog -> Inset[Graphics[Text[Style["charged", 18]], {2.0, 0.2}]]}
4578 - discharged to -101C, worn

dataString = Import[pathVulcanData <> "4578.prof.dat", "Data"];
numberOfPoints = Dimensions[dataString][[1]] - 4;
dataDischargedWorn = ConstantArray[{0, 0, 0}, numberOfPoints];
index = 1;
For[lineIndex = lineStart, lineIndex <= lineEnd, lineIndex++]
    If[lineIndex > Dimensions[dataString][[1]], Break[]];
    line = dataString[[lineIndex]];
dataDischargedWorn[[index, All]] = line;
    index++;
];
timeTOF = dataDischargedWorn[[All, 1]];
wavelengthTOF = Table[flightTimeToWavelength[[timeTOF[[i]]]] Micro Second, 
    pathLengthVulcan] / (Angstrom), {i, Length[timeTOF]}];
dSpacingTOF = \frac{1 \times \text{wavelengthTOF}}{2 \sin [\text{correctTwoThetaAngle}]};
dataDischargedWorn[[All, 1]] = dSpacingTOF;
y = dataDischargedWorn[[All, 2]];
y = 100 * y / Total[y];
x = dataDischargedWorn[[All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
    y[[i]] / interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataDischargedWorn[[All, 2]] = y;
gDischargedWorn = ListPlot[Transpose[
    {dataDischargedWorn[[All, 1]], dataDischargedWorn[[All, 2]]}],
    Joined -> True, PlotStyle -> {Black, Thick},
    PlotRange -> {{0.6, 2.5}, {0, 0.25}},
    ImageSize -> 600, AxesOrigin -> {0.6, 0},
    Frame -> True, FrameStyle -> Directive[18], FrameLabel ->
    {"intensity/arb. units", ", "}, {"d-spacing/Å", "worn battery"}],
Epilog -> Inset[Graphics[Text[Style["discharged (101 C)", 18]],
    {2.0, 0.2}]]];
2487

204
worn battery

discharged (101 C)

Dimensions[dataDischargedWorn[All, 1 ;; 2]]
dataCharged[1 ;; 10, 1 ;; 2]
Export[NotebookDirectory[] <> "WornBattery_Discharged_Run4578.xls",
dataDischargedWorn[All, 1 ;; 2], "XLS"
{2487, 2}

{{0.30561, 0}, {0.305915, 0}, {0.306221, 0},
 {0.306528, 0}, {0.306834, 0}, {0.307141, 0},
 {0.307448, 0}, {0.307756, 0}, {0.308063, 0}, {0.308371, 0}}

/Users/tomo/Dropbox/Battery_manuscript/Mathematica/
WornBattery_Discharged_Run4578.xls
Show[{gChargedWorn, gDischargedWorn}, Epilog -> Inset[Graphics[
  Text[Style["charged (blue)\n discharged (101 C) (black) ", 16]],
  {2.1, 0.2}]]

worn battery

charged (blue)
discharged (101 C) (black)

4574,4575 - discharging, worn

tmp = Table[ToString[listTimes[[i, 4 ;; 6]]], {i, Length[listTimes]}];
TableForm[
  Transpose[{listRun, listProfDat, tmp, listIndex, listCoulomb}],
  TableHeadings ->
  {Automatic, {"Run", "split", "time", "i", "coulomb"}}];
numberTimeWindows = (95 - 18 + 1);
indexOffset = 17; gDischargingWorn = {};

listCoulombCumulative =
    Table[Total[listCoulomb[[1 + indexOffset ;; i + indexOffset]]],
        {i, numberTimeWindows}];
dataDischargingWorn = ConstantArray[{0, 0, 0},
        {numberTimeWindows, numberOfPoints}];
For[i = 1, i ≤ numberTimeWindows, i++,
    runNumber = listRun[[i + indexOffset]]; splitNumber = listProfDat[[i + indexOffset]];
    Print[{runNumber, splitNumber}];
    If[runNumber = 4574,
        dataString = Import[pathVulcanData <> "/4574/" <>
            ToString[splitNumber] <> ".prof.dat", "Data"],
        dataString = Import[pathVulcanData <> "/4575/" <>
            ToString[splitNumber] <> ".prof.dat", "Data"],
        ];
    index = 1;
    For[lineIndex = lineStart, lineIndex ≤ lineEnd, lineIndex++,
        If[lineIndex > Dimensions[dataString][[1]], Break[]];
        line = dataString[[lineIndex]];
        dataDischargingWorn[[i, index, All]] = line;
        index++;
    ];
    timeTOF = dataDischargingWorn[[i, All, 1]];
wavelengthTOF =
    Table[flightTimeToWavelength[[timeTOF[[i]]]]] Micro Second,
        pathLengthVulcan] / (1 Angstrom), {i, Length[timeTOF]}];
    dSpacingTOF = \frac{1 \times wavelengthTOF}{2 \sin[correctTwoThetaAngle]};
dataDischargingWorn[[i, All, 1]] = dSpacingTOF;
y = dataDischargingWorn[[i, All, 2]];
y = 100 * y / Total[y];
x = dataDischargingWorn[[i, All, 1]];
y = Table[If[interpVanadiumCubic[x[[i]]] > 0,
        y[[i]]/ interpVanadiumCubic[x[[i]]], 0], {i, Length[y]}];
dataDischargingWorn[[i, All, 2]] = y;
gDischargingWorn = Append[gDischargingWorn,
    ListPlot[Transpose[{x, y}], Joined -> True,
        PlotStyle -> {Black, Thick}, PlotRange -> {{0.6, 2.5}, {0, 0.25}},
        ImageSize -> 600, AxesOrigin -> {0.6, 0},
        Frame -> True, FrameStyle -> Directive[18],
        FrameLabel -> {{"intensity/arb. units", ""},
            {"d-spacing/Å", "worn battery"}},
        Epilog -> Inset[Graphics[Text[Style[ToString[listCoulombCumulative[[i]]] <> " C",
            18]], {2.0, 0.2}]]];
    ];
gAll = Flatten[{gChargedWorn, gDischargingWorn, gDischargedWorn}];
ListAnimate[gAll]
4573, (4574,4575), 4578 - integration of selected diffraction peaks

4573 charged, worn battery #3319: blue peaks = 2.145(0.02), 2.018(0.01), 1.847(0.04), 1.764(0.03), 1.575(0.01), 1.45(0.01), 4574,4575 discharging; split into 5 minute windows
4578 discharged by -100 Coulombs: black peaks = 2.007(0.01), 1.683(0.04), 1.556(0.02), 1.122(0.01), 0.993(0.01)

peaksBlue =  
  (2.145, 0.02), (2.018, 0.01), 
  (1.847, 0.04), (1.764, 0.03), (1.575, 0.01), (1.45, 0.01);
peaksBlack =  
  (2.007, 0.01), (1.683, 0.04), 
  (1.556, 0.02), (1.122, 0.01), (0.993, 0.01);

For[i = 1, i ≤ Length[peaksBlue], i++,
  {center, fullwidth} = peaksBlue[[i]];
  areaPeaksBlue[[i, 1]] = ToString[NumberForm[center, 3]] <>
    " (" <> ToString[fullwidth] <> ")";
  x = dataChargedWorn[[All, 1]]; y = dataChargedWorn[[All, 2]];
  indexPeak = Flatten[Position[x, p_?((center - fullwidth/2) &&
    (p < (center + fullwidth/2)) &&] ];
  {indexMin, indexMax} = {First[indexPeak], Last[indexPeak]};
  xMin, xMax} = {x[[indexMin]], x[[indexMax]]};
  {yMin, yMax} = {y[[indexMin]], y[[indexMax]]};
  areaBaseline = Mean[({xMin, xMax}) x (xMax - xMin)];

For[i = 1, i ≤ Length[peaksBlue], i++,
  {center, fullwidth} = peaksBlue[[i]];
  x = dataChargedWorn[[All, 1]]; y = dataChargedWorn[[All, 2]];
  indexPeak = Flatten[Position[x, p_?((center - fullwidth/2) &&
    (p < (center + fullwidth/2)) &&] ];
  {indexMin, indexMax} = {First[indexPeak], Last[indexPeak]};
  xMin, xMax} = {x[[indexMin]], x[[indexMax]]};
  {yMin, yMax} = {y[[indexMin]], y[[indexMax]]};
  areaBaseline = Mean[({xMin, xMax}) x (xMax - xMin)];

areaPeaksBlue[[i, indexCoulomb + 1]] = areaPeak;
worn battery

charged (blue)
discharged (black)
\[ x = \text{axisCoulomb}; \ y = \text{areaPeaksBlue}[\{\text{All}, 2 \}; (\text{Length}[x] + 1)]; \]
\[
g\text{PeakBlue} = \text{ListPlot}[\text{Table}[\text{Transpose}[\{x, y[[i, \text{All}]]\}], \{i, \text{Length}[\text{peaksBlue}]\}], \text{PlotStyle} \to \text{Blue}, \]
\[
\text{Joined} \to \text{True}, \text{PlotMarkers} \to \{"1", "2", "3", "4", "5", "6"}, \]
\[
\text{PlotRange} \to \{(-35, 80), \text{All}\}, \text{ImageSize} \to 600, \text{AxesOrigin} \to \{0.0, 0\}, \]
\[
\text{Frame} \to \text{True}, \text{FrameStyle} \to \text{Directive}[18], \text{FrameLabel} \to \{
\{\text{"peak areas"}, ""}, \{\text{"discharging/Coulomb"}, \"worn battery\"\} \}; \]
\[
\text{For}[i = 1, i \leq \text{Length}[\text{peaksBlue}], i++], \]
\[
xText = -30; \ yText = \text{Take}[\text{areaPeaksBlue}[\{i, \text{All}\}], \{10\}][[1]]; \]
\[
g\text{TextBlue}[i] = \]
\[
\text{Graphics}[\text{Text}[\text{Style}[\text{areaPeaksBlue}[\{i, 1\}] <> \text{\" Â\"}, 12, \text{Blue}], \]
\[
\{xText, yText\}, \{-1, 0\}]]; \]
\[
\]; \]
\[
\text{Show}[\{g\text{PeakBlue}, \text{Table}[g\text{TextBlue}[i], \{i, \text{Length}[\text{peaksBlue}]\}] \}] \]

\[
x = \text{axisCoulomb}; \ y = \text{areaPeaksBlack}[\{\text{All}, 2 \}; (\text{Length}[x] + 1)]; \]
\[
g\text{PeakBlack} = \text{ListPlot}[\text{Table}[\text{Transpose}[\{x, y[[i, \text{All}]]\}], \{i, \text{Length}[\text{peaksBlack}]\}], \text{PlotStyle} \to \text{Black}, \]
\[
\text{Joined} \to \text{True}, \text{PlotMarkers} \to \{"1", "2", "3", "4", "5"}, \]
\[
\text{PlotRange} \to \{(-35, 100), \text{All}\}, \text{ImageSize} \to 600, \text{AxesOrigin} \to \{0.0, 0\}, \]
\[
\text{Frame} \to \text{True}, \text{FrameStyle} \to \text{Directive}[18], \text{FrameLabel} \to \{
\{\text{"peak areas"}, ""}, \{\text{"discharging/Coulomb"}, \"worn battery\"\} \}; \]
\[
\text{For}[i = 1, i \leq \text{Length}[\text{peaksBlack}], i++], \]
\[
xText = 75; \ yText = \text{Take}[\text{areaPeaksBlack}[\{i, \text{All}\}], \{60\}][[1]]; \]
\[
g\text{TextBlack}[i] = \]
\[
\text{Graphics}[\text{Text}[\text{Style}[\text{areaPeaksBlack}[\{i, 1\}] <> \text{\" Â\"}, 12, \text{Black}], \]
\[
\{xText, yText\}, \{-1, 0\}]]; \]
\[
\]; \]
\[
\text{Show}[\{g\text{PeakBlack}, \text{Table}[g\text{TextBlack}[i], \{i, \text{Length}[\text{peaksBlack}]\}] \}] \]
4573, (4574,4575), 4578 - save to hdf5 file

Thin Battery #92:
Appendix D: Neutron Far-field Interferometry Mathematica Code

The following code is used from a January 2017 far-field neutron interferometry experiment at NIST, Gaithersburg. Specifically, this is the code used to make the projections for the 3 mm SS316 additive manufactured fatigued samples. This phase-stepping code can be utilized for Talbot-Lau interferometry or far-field interferometry. Mathematica is used simply as a check to ensure input/output parameters are workable.

In the code, functions are defined for the experimental setup, calculations for the reference and sample images, and plotting of the resulting figures. Files from the experiment are imported and cropped to focus on regions of interest within the samples. Absorption, DPC, and dark-field signals are calculated for each angle of sample imaging then exported as FITS projection files.
Expt2_AMHalf_25Jan2017.nb

25Jan2017
NIST far-field
Open beam: the real open beam images are “OPEN_AM3DHalf__p0120d00000”
“OPEN_AM3DHalf__p0002d00000” have sample in position

Frustrating issues.
1) the FITS format varies with Import dimensionallity of \{1,2160,2560\} and \{1,1,2160,2560\}.
The former requires \text{data}=\text{Import[filenamesSample[[1]],"RawData"]}\[1\];
The latter requires \text{data}=\text{Import[filenamesSample[[1]],"RawData"]}\[1,1\];

\text{ClearAll["Global`*"]}
\text{NotebookDirectory[]}
\$HomeDirectory
\text{ToString[Round[MemoryInUse[] / 2^30 // N]]}<>" Gb"

Step 1: Define functions

Functions for reading file, finding number of groups of reference files and sample files (Andor NEO, Andor iKon)

\text{funcReadPilatusFile[filename_] } This will read data from a Pilatus integer-32 FIT file. The standard \text{Mathematica Import[filename]} cannot read integer-32 FIT.
\text{funcFindReferenceFITfiles[pathFIT_] } Finds files based on ".white." and orders the list based on the image sequence number.
\text{funcFindSampleFITfiles[pathFIT_] } Finds files based on "raw." and orders the list based on the image sequence number.
\text{funcFindReferenceGroups[]} Based on sequence number, finds the grouping of the reference files.
\text{funcFindSampleGroups[]} Based on sequence number, finds the grouping of the sample files.

(* Modified for NIST Andor camera *)
\text{funcReadAndorFITFileMedianFilter[listfilenames, cropRows_, cropColumns_, binning_, referenceDarkfield_]} :=
\text{Module[{filename, image, data, rows, columns, alldata},
data = \text{Import[listfilenames[[1]],"RawData"]}\[1,1\];
\{rows, columns\} = \text{Dimensions[data];}
alldata =
\text{ConstantArray[0, \{rows, columns, numberImagesForMedianFilter\}];}
\text{For[index = 1, index <= Length[listfilenames], index++,}
\text{filename = listfilenames[[index]];}
data = \text{Import[filename, "RawData"]}\[1,1\];
alldata[\{All, All, index\}] = data;
];
data = Map[\text{Median, alldata, \{2\}}];
\text{image = Image[data, "Real32"]};
\text{image = ImageTake[image, cropRows, cropColumns];}
\text{image = ImageResize[image, Scaled[1/binning]]};
data = \text{ImageData[image, "Real"] } - \text{referenceDarkfield  }];
funcFindReferenceFITfiles[pathFIT_, strOpen_] := Module[{},
filenamesReference = FileNames[strOpen <> "*.fit", pathFIT];
numberOfReferenceFiles = Length[filenamesReference];
listReferenceIndexSequenceGratingNumbers = Table[Module[{},
text1 = Last[FileNameSplit[filenamesReference[[i]]]];
text2 = StringSplit[text1, {".", "_"}];
rotationAngle = 0;
sampleTranslation = 0;
text3 = Take[text2, {-3}][[1]];
text4 = StringSplit[text3, {"p", "m", "d"}];
phaseStepDistance = 1000 * (ToExpression[First[text4]] +
ToExpression[Last[text4]] / 100000) // N;
{i, sampleTranslation, phaseStepDistance, filenamesReference[[i]]},
{1, numberOfReferenceFiles}]; ];

funcFindSampleFITfiles[pathFIT_, strRaw_] := Module[{},
filenamesSample = FileNames[strRaw <> "*.fit", pathFIT];
numberOfSampleFiles = Length[filenamesSample];
listSampleIndexSequenceGratingNumbers = Table[Module[{},
text1 = Last[FileNameSplit[filenamesSample[[i]]]];
text2 = StringSplit[text1, {".", "_"}];
distanceGIG2 = 23;
rotationAngle = 0;
text3 = Take[text2, {-4}][[1]];
text4 = StringSplit[text3, {"p", "m", "d"}];
sampleTranslation = ToExpression[First[text4]] +
ToExpression[Last[text4]] / 100000 // N;
text3 = Take[text2, {-3}][[1]];
text4 = StringSplit[text3, {"p", "m", "d"}];
phaseStepDistance = 1000 * (ToExpression[First[text4]] +
ToExpression[Last[text4]] / 100000) // N;
{i, sampleTranslation, phaseStepDistance, filenamesSample[[i]]},
{1, numberOfSampleFiles}]; ];

funcFindDarkfieldFITfiles[pathFIT_] := Module[{},
filenamesDarkfield = FileNames["*.fit", pathFIT]]; 

funcReadDarkfield[listfilenamesDarkfield_,
cropRows_, cropColumns_, binning_] :=
Module[{filename, allDarkfield, image, data, rows, columns},
data = Import[listfilenamesDarkfield[[1]], "RawData"][[1, 1]]; 
{rows, columns} = Dimensions[data];
allDarkfield = ConstantArray[0, {rows, columns, Length[listfilenamesDarkfield]]};
For[index = 1, index <= Length[listfilenamesDarkfield], index++,
filename = listfilenamesDarkfield[[index]];
data = Import[filename, "RawData"][[1, 1]];
allDarkfield[[All, All, index]] = data; ];
data = Map[Median, allDarkfield, {2}];
image = Image[data, "Real32"];
(*image = ImageReflect[image, Top→Bottom];*)
image = ImageTake[image, cropRows, cropColumns];
image = ImageResize[image, Scaled[1/binning]];
referenceDarkfield = ImageData[image, "Real"]; 
referenceDarkfield = funcCorrectForZeroValuePixels[referenceDarkfield] ];

215
Functions for interferometry

```plaintext
cfuncPrepareBvectorArbitrarySteps[gratingPeriodMicron_,
    listGratingStepsMicron_] := Module[{b1, b2, b3, numberGratingSteps},
    numberGratingSteps = Length[listGratingStepsMicron];
    b1 = Table[1, {i, numberGratingSteps}];
    b2 = Table[Sin[2 π listGratingStepsMicron[[i]] / gratingPeriodMicron] //
                 N, {i, numberGratingSteps}];
    b3 = Table[Cos[2 π listGratingStepsMicron[[i]] / gratingPeriodMicron] //
                 N, {i, numberGratingSteps}];
    Chop[Transpose[{b1, b2, b3}]]];

cfuncPrepareAllVectors[gratingPeriodMicron_,
    listGratingStepsMicron_, rows_, columns_] := Module[{},
    bVector = cfuncPrepareBvectorArbitrarySteps[
                 gratingPeriodMicron, listGratingStepsMicron];
    aVector = cVector = ConstantArray[0, {3, rows * columns}];
    aMatrix = ConstantArray[0, {rows, columns, 3}];
    visibility = phi = ConstantArray[0, {rows, columns}];
    gMatrix = Inverse[Transpose[bVector] . bVector] . Transpose[bVector]; ]
```
funcReadOneInterferogramMedianFilterReference[
  listIndexSequenceGratingNumbers_, listGratingStepsMicron_] :=
  Module[{listFilenames, intensity, gratingStep, listIndexMedian},
    allData = ConstantArray[0.0, {rows, columns, numberGratingSteps}];
    For[indexSteps = 1, indexSteps <= numberGratingSteps, indexSteps++,
      gratingStep = listGratingStepsMicron[[indexSteps]];
      listIndexMedian = Position[listIndexSequenceGratingNumbers[[All, 3]],
        p_ ? (# = gratingStep &)] // Flatten;
      listFilenames = Table[listIndexSequenceGratingNumbers[
        listIndexMedian[[i]], 4], {i, numberImagesForMedianFilter}];
      Print[{gratingStep, " um", listFilenames}];
      intensity = funcReadAndorFITFileMedianFilter[listFilenames,
        cropRows, cropColumns, binning, referenceDarkfield];
      intensity = funcCorrectForZeroValuePixels[intensity];
      allData[[All, All, indexSteps]] = intensity;
    ];
    allData];

funcReadOneInterferogramMedianFilterSample[
  listIndexSequenceGratingNumbers_, listGratingStepsMicron_,
  translation_] := Module[{listFilenames, intensity, gratingStep,
    listIndexTranslation, listIndexMedian, listIndexFilenames},
    allData = ConstantArray[0.0, {rows, columns, numberGratingSteps}];
    listIndexTranslation = Position[listIndexSequenceGratingNumbers[[All, 2]],
      p_ ? (# = translation &)] // Flatten;
    For[indexSteps = 1, indexSteps <= numberGratingSteps, indexSteps++,
      gratingStep = listGratingStepsMicron[[indexSteps]];
      listIndexMedian = Position[listIndexSequenceGratingNumbers[[All, 3]],
        p_ ? (# = gratingStep &)] // Flatten;
      listIndexFilenames = Intersection[listIndexTranslation,
        listIndexMedian];
      listFilenames = Table[listIndexSequenceGratingNumbers[
        listIndexFilenames[[i]], 4], {i, numberImagesForMedianFilter}];
      Print[{gratingStep, " um", listFilenames}];
      intensity = funcReadAndorFITFileMedianFilter[listFilenames,
        cropRows, cropColumns, binning, referenceDarkfield];
      intensity = funcCorrectForZeroValuePixels[intensity];
      allData[[All, All, indexSteps]] = intensity;
    ];
    allData];

funcCalculateTransmissionVisibilityPhi[data_] :=
  Module[{transmission, visibility, phi},
    cVector = Transpose[Flatten[data, {1, 2}]];
    aVector = gMatrix.cVector;
    aMatrix = Partition[Transpose[aVector], columns];
    transmission = aMatrix[[All, All, 1]];
    visibility = Sqrt[aMatrix[[All, All, 2]]^2 + aMatrix[[All, All, 3]]^2 ];
    phi = ArcTan[aMatrix[[All, All, 2]], aMatrix[[All, All, 3]]];
    {transmission, visibility, phi}];
funcCorrectForZeroValuePixels[dataOriginal_] :=
Module[{coordinatesZeroIntensityPixels, dataCorrected, dataMedianFilter, r, c},
dataCorrected = dataOriginal;
coordinatesZeroIntensityPixels = Position[dataCorrected, _?(_ <= 0 &)];
If[Length[coordinatesZeroIntensityPixels] > 0,
Module[{},
dataMedianFilter = MedianFilter[dataOriginal, 1];
For[index = 1, index <= Length[coordinatesZeroIntensityPixels], index++,
{r, c} = coordinatesZeroIntensityPixels[[index]];
dataCorrected[[r, c]] = dataMedianFilter[[r, c]];
]
];
coordinatesZeroIntensityPixels = Position[dataCorrected, _?(_ <= 0 &)];
If[Length[coordinatesZeroIntensityPixels] > 0,
Module[{},
For[index = 1, index <= Length[coordinatesZeroIntensityPixels], index++,
{r, c} = coordinatesZeroIntensityPixels[[index]];
dataCorrected[[r, c]] = 1;
]
];
dataCorrected];

funcCorrectLargeDPCvalue[data_] :=
Module[{coordinatesLargeDPC, dataCorrected, r, c},
dataCorrected = data;
coordinatesLargeDPC = Position[data, _?(_ >= π &)];
If[Length[coordinatesLargeDPC] > 0,
Module[{},
For[index = 1, index <= Length[coordinatesLargeDPC], index++,
{r, c} = coordinatesLargeDPC[[index]];
dataCorrected[[r, c]] = data[[r, c]] - 2 π;
]
];
coordinatesLargeDPC = Position[data, _?(_ < -π &)];
If[Length[coordinatesLargeDPC] > 0,
Module[{},
For[index = 1, index <= Length[coordinatesLargeDPC], index++,
{r, c} = coordinatesLargeDPC[[index]];
dataCorrected[[r, c]] = data[[r, c]] + 2 π;
]
];
dataCorrected];
Functions for finding correct files for a given translation

    funcGetSampleFileNamesForSpecificTranslation[translation_] := Module[{},
        indexTranslation = Flatten[Position[listSampleIndexSequenceGratingNumbers[[All, 2]],
            p_? (p == translation &)]];
        indexFilename = listSampleIndexSequenceGratingNumbers[[
            indexTranslation, 1]];
        listSampleFileNamesAtThatTranslation =
            filenamesSample[[indexFilename]]];
```
funcPlotTransmissionVisiblityPhiFit[
  testPoint_, plotLabel_, transmissionMatrix_,
  visibilityMatrix_, phiMatrix_, interferogram_] := Module[{},
  gReferenceTransmission = ArrayPlot[transmissionMatrix,
    PlotRange -> {All, All, climTrans}, ColorFunction -> "GrayTones",
    ClippingStyle -> {Black, White}, Frame -> False,
    PlotLabel -> plotLabel <> " transmission, dist=" <>
    ToString[transmission] <> ", test point=" <> ToString[testPoint],
    Axes -> True, AspectRatio -> rows / columns, ImageSize -> 300,
    PlotLegends -> {Placed[BarLegend[{{"GrayTones", climTrans}, Right}],
        Epilog -> {PointSize -> Large, Point[[testPoint]]]}},
  visibilityPercentMatrix = 100 * visibilityMatrix / transmissionMatrix;
  gReferenceVisibility = ArrayPlot[visibilityPercentMatrix,
    PlotRange -> {All, All, climVis}, ColorFunction -> "GrayTones",
    ClippingStyle -> {Black, White}, Frame -> False,
    PlotLabel -> plotLabel <> " visibility/"%
    Axes -> True, AspectRatio -> rows / columns, ImageSize -> 300,
    PlotLegends -> {Placed[BarLegend[{{"GrayTones", climPhi}, Right}],
        Epilog -> {Line[{{0, 0}, {1, 1}}]}]},
  yExpt = interferogram[[testPointOne[[2]], testPointOne[[1]], All]]; xExpt = listGratingStepsMicron;
  transmission = transmissionMatrix[[testPoint[[2]], testPoint[[1]]]];
  visibility = visibilityMatrix[[testPoint[[2]], testPoint[[1]]]];
  visibilityPercentTestPoint = Round[100 * visibility / transmission, 0.1];
  phi = phiMatrix[[testPoint[[2]], testPoint[[1]]]]; yCalc = Table[
    transmission + visibility Sin[2 Pi xExpt[[i]] / gratingPeriodMicron + phi],
    {i, Length[xExpt]}];
  sigma = Sqrt[yExpt] // N;
  chi = (1 / numberGratingSteps - 1) Total[(yExpt - yCalc)^2 / sigma];
  g1 = ListPlot[Transpose[{xExpt, yExpt}], Joined -> {False, True},
    Frame -> True, FrameLabel -> {"intensity", ""},  
    plotLabel <> ", test point=" <> ToString[testPoint],
    Epilog -> Inset[Style[Text[plotString], 12], Scaled[{0.4, 0.85}],
    ImageSize -> 300];
  g2 = Plot[transmission + visibility Sin[2 Pi x / gratingPeriodMicron + phi],
    {x, Min[listGratingStepsMicron], Max[listGratingStepsMicron]}];
  gFitReference = Show[{g1, g2}];
  gAllReference = Grid[{{gReferenceTransmission, gReferenceVisibility},
    {gReferencePhi, gFitReference}}]
]
```
funcPlotAbsorptionDPCDarkField[plotLabel_,
  absorptionMatrix_, dpcMatrix_, darkfieldMatrix_] := Module[{},
  gAbsorption = ArrayPlot[absorptionMatrix,
    PlotRange -> (All, All, climAbs), ColorFunction -> "GrayTones",
    ClippingStyle -> {Black, White}, Frame -> False,
    PlotLabel -> "absorption", dist" -> ToString[translation],
    Axes -> True, AspectRatio -> rows / columns, ImageSize -> 300,
    PlotLegends -> {Placed[BarLegend["GrayTones", climAbs]], Right}];
  gDPC = ArrayPlot[dpcMatrix,
    PlotRange -> (All, All, climDPC), ColorFunction -> "GrayTones",
    ClippingStyle -> {Black, White}, Frame -> False,
    PlotLabel -> "DPC", dist" -> ToString[translation],
    Axes -> True, AspectRatio -> rows / columns, ImageSize -> 300,
    PlotLegends -> {Placed[BarLegend["GrayTones", climDPC]], Right}];
  gDarkField = ArrayPlot[darkfieldMatrix,
    PlotRange -> (All, All, climDarkField), ColorFunction -> "GrayTones",
    ClippingStyle -> {Black, White}, Frame -> False,
    PlotLabel -> "dark-field", dist" -> ToString[translation],
    Axes -> True, AspectRatio -> rows / columns, ImageSize -> 300,
    PlotLegends -> {Placed[BarLegend["GrayTones", climDarkField]], Right}];

  x = Range[rows];
  listColumns = Range[50, columns - 50];
  numberOfColumns = Length[listColumns];
  lineProbeVertical = Take[absorptionMatrix, All, {50, columns - 50}, 50];
  gAbsorptionLineProbe = ListPlot[Table[
    Transpose[{x, (i - 1) + Max[climAbs] + lineProbeVertical[[All, i]]},
      {i, numberOfColumns}], Joined -> True, Frame -> True,
      FrameLabel -> {{"absorption", ""}, {"row", plotLabel}},
      Epilog -> Table[Style[Text[listColumns[[i]], {10, (i - 1) + Max[climAbs]}], 12,
        Background -> White], {i, numberOfColumns}, ImageSize -> 300];
  lineProbeVertical = Take[dpcMatrix, All, {50, columns - 50}, 50];
  gDPCLineProbe = ListPlot[Table[
    Transpose[{x, (i - 1) + Max[climDPC] + lineProbeVertical[[All, i]]},
      {i, numberOfColumns}], Joined -> True, Frame -> True,
      FrameLabel -> {{"DPC", ""}, {"row", plotLabel}},
      Epilog -> Table[Style[Text[listColumns[[i]], {10, (i - 1) + Max[climDPC]}], 12,
        Background -> White], {i, numberOfColumns}, ImageSize -> 300];
  lineProbeVertical = Take[darkfieldMatrix, All, {50, columns - 50}, 50];
  gDarkFieldLineProbe = ListPlot[Table[Transpose[{x, (i - 1) + Max[climDarkField] +
    lineProbeVertical[[All, i]]}, {i, numberOfColumns}],
    PlotRange -> (All, (1 + numberOfColumns) + Max[climDarkField]},
    Joined -> True, Frame -> True,
    FrameLabel -> {{"dark-field", ""}, {"row", plotLabel}},
    Epilog -> Table[Style[Text[listColumns[[i]], {10, (i + Max[climDarkField])}],
      12, Background -> White], {i, numberOfColumns}, ImageSize -> 300];
  gAllAbsDPCDarkField = Grid[{{gAbsorption, gDPC, gDarkField},
    {gAbsorptionLineProbe, gDPCLineProbe, gDarkFieldLineProbe}}] ]
Image correction functions

funcRemoveFringeWithLowPassFilter[data_] :=
Module[{rows, columns, rowsTopRows, columnsTopRows,
    topRows, smoothedTopRows, dataSmoothedFringe},
  {rows, columns} = Dimensions[data];
  topRows = Take[data, {1, 20}, All];
  {rowsTopRows, columnsTopRows} = Dimensions[topRows];
  topRows = Total[topRows, {1}] / rowsTopRows // N;
  smoothedTopRows = LowpassFilter[topRows, 0.5];
  dataSmoothedFringe =
    Table[(rows - i) / rows * smoothedTopRows, {i, rows}];
  data = dataSmoothedFringe;

funcUnwrapAndMedianFilter[data_] :=
Module[{indexPhaseWrap, dataUnwrapped},
  dataUnwrapped = data;
  indexPhaseWrap = Position[data, p_? (p ≤ -0.5 π &)];
  For[i = 1, i <= Length[indexPhaseWrap], i++,
    {r, c} = indexPhaseWrap[[i]];
    dataUnwrapped[[r, c]] = data[[r, c]] + 2 * π;
    ];
  dataUnwrapped = MedianFilter[dataUnwrapped, 1];

Interferometer steps and period, Image cropping

Paths, filenames, reference.

pathRoot = NotebookDirectory[];
testNameStr = "AMHalf";
pathRaw = pathRoot <> "Raw/";
pathOpenBeam = pathRoot <> "Openbeam/";
pathDarkField = pathRoot <> "Dark/";
pathProjections = pathRoot <> "projections/";
pathHDF5 = pathRoot <> "HDF5/";
pathFigures = pathRoot <> "figures/";
gratingPeriodMicron = 2.4;
numberOfImagesForMedianFilter = 3;

funcFindReferenceFITfiles[pathOpenBeam, "OPEN_AMHalf__p0120d00000" ]
numberOfReferenceFiles
Take[listReferenceIndexSequenceGratingNumbers, 6] // TableForm
funcFindSampleFITfiles[pathRaw, "Xi_AMHalf"];
numberOfSampleFiles
Take[listSampleIndexSequenceGratingNumbers, 6] // TableForm

listSampleTranslation =
  Union[listSampleIndexSequenceGratingNumbers[[All, 2]]]
numberTranslations = Length[listSampleTranslation]

listGratingStepsMicron =
  Union[listReferenceIndexSequenceGratingNumbers[[All, 3]]]
numberGratingSteps = Length[listGratingStepsMicron]
TableForm[{{"sample name", testNameStr},
{"# of reference images", numberOfReferenceFiles},
{"# of sample images", numberOfSampleFiles},
{"# of gratings steps", numberGratingSteps},
{"grating steps", listGratingStepsMicron},
{"# images for median filter", numberImagesForMedianFilter},
{"# of translations", numberTranslations},
{"sample translations", listSampleTranslation} ]

Image cropping and binning: Crop image with ImageTake[]. Binning with ImageResize[___, Scaled[1/binning]]
{columns,rows}=ImageDimensions[image] and {NY,NX}=Dimensions[data]
Determine the depth of the raw data; this leads to [[1]] or [[1,1]]

Import[filenamesSample[[1]], "Elements"]
Import[filenamesSample[[1]], "DataType"]
Max[Import[filenamesSample[[1]], "Data"]]
data = Import[filenamesSample[[1]], "RawData"];
Dimensions[data]
data = Import[filenamesSample[[1]], "RawData"][[1, 1]];
Dimensions[data]
Max[data]

image = Image[Import[filenamesSample[[1]], "RawData"][[1, 1]], "Real32"];
ImageQ[image]
image = ImageAdjust[ImageReflect[image, Top -> Bottom]];  
{columnsOriginal, rowsOriginal} = ImageDimensions[image]
ImageHistogram[image]
ImageAdjust[image, {0, 0, 1}, {0, 0.05}]

cropRows = {1, 2160};
cropColumns = {1, 2560};
binning = 2;

image = Image[Import[filenamesSample[[1]], "RawData"][[1, 1]], "Real32"];
image = ImageAdjust[ImageReflect[image, Top -> Bottom]];  
image = ImageTake[image, cropRows, cropColumns];
image = ImageResize[image, Scaled[1/binning], Resampling -> "Linear"];
{columns, rows} = ImageDimensions[image]
data = Import[data, "Real"];  
{NY, NX} = Dimensions[data]
image = ImageAdjust[ImageResize[image, 200]];
GraphicsRow[{image, ArrayPlot[data],  
ListPlot[data[[Round[NY/2, All]]]], ImageSize -> 800}]

Based on cropping and binning, calculate a dark-field.
In the absence of dark-field measurements, use referenceDarkfield = 100

funcFindDarkfieldFITfiles[pathDarkField]
data = Import[filenamesDarkfield[[1]], "RawData"];
Dimensions[data]
data = Import[filenamesDarkfield[[1]], "RawData"][[1, 1]];
Dimensions[data]
Max[data]
funcReadDarkfield[filenamesDarkfield, cropRows, cropColumns, binning];
Dimensions[referenceDarkfield]
{Min[referenceDarkfield],
Mean[Flatten[referenceDarkfield]] // N, Max[referenceDarkfield]}
Image[referenceDarkfield, "Real"] // ImageAdjust
Histogram[Flatten[referenceDarkfield]]

Verify interferometry steps and period at first reference

(*filenamesReference;*)
funcPrepareAllVectors[gratingPeriodMicron,
listGratingStepsMicron, rows, columns]
{Dimensions[gMatrix], Dimensions[bVector], Dimensions[aVector],
Dimensions[aMatrix], Dimensions[visibility], Dimensions[phi]}
interferogramReference = funcReadOneInterferogramMedianFilterReference[
listReferenceIndexSequenceGratingNumbers, listGratingStepsMicron];
Dimensions[interferogramReference]
{referenceTransmission, referenceVisibility, referencePhi} =
funcCalculateTransmissionVisibilityPhi[interferogramReference];
testPointOne = {Round[columns / 2], Round[rows / 2]};
climTrans = {Min[referenceTransmission],
Max[MedianFilter[referenceTransmission, 1]]};
referenceVisibilityPercent = 100 * -referenceVisibility
referenceTransmission;
climVis = {0, 1.5 * Median[Flatten[referenceVisibilityPercent]]};
climPhi = {-1, 1} * Max[MedianFilter[referencePhi, 1]];
gAllReference = funcPlotTransmissionVisibilityPhiFit[
    testPointOne, "ref", referenceTransmission,
    referenceVisibility, referencePhi, interferogramReference]
Manipulate[Module[
  {x, b1, b2, b3, b, g, a1, a2, a3, visibility, phi, data, y, sigma, chi},
  b1 = Table[{i, numberGratingSteps}];
  b2 = Table[Sin[2 π listGratingStepsMicron[[i]] / pg] // N,
    {i, numberGratingSteps}];
  b3 = Table[Cos[2 π listGratingStepsMicron[[i]] / pg] // N,
    {i, numberGratingSteps}];
  b = Chop[Transpose[{b1, b2, b3}]];
  g = Inverse[Transpose[b] . b] . Transpose[b];
  data = Flatten[Take[allData, {r, r + (bin - 1)}, {c, c + (bin - 1)}], 1];
  data = Total[data, {1}];
  {a1, a2, a3} = g . data;
  visibility = Sqrt[a2^2 + a3^2];
  phi = ArcTan[a2, a3];
  y = Table[a1 + visibility Sin[2 π listGratingStepsMicron[[i]] / pg + phi],
    {i, numberGratingSteps}];
  sigma = Sqrt[data] // N;
  chi = (1 / (numberGratingSteps - 1)) Total[(data - y)^2 / sigma];
  visibilityPercent = Round[100 * visibility / a1, 0.1];
  plotString = " a=" <> ToString[{a1, a2, a3}] <> "\n"
    <> " visibility =" <> ToString[visibilityPercent] <> "%\n"
    <> " phi=" <> ToString[phi] <> "\n"
    <> " rad; " <> ToString[Round[phi * 180 / Pi]] <> "°\n"
    <> " χ^2 = " <> ToString[chi];
  g1 = ListPlot[
    Transpose[{listGratingStepsMicron, data}], Joined -> {False, True},
    Epilog -> Inset[Style[Text[plotString], 12], Scaled[{0.3, 0.75}]]];
  g2 = Plot[a1 + visibility Sin[2 π x / pg + phi],
    {x, Min[listGratingStepsMicron], Max[listGratingStepsMicron]}];
  Show[{g1, g2},
    {{r, 1}, 1, rows, 1, Appearance -> "Labeled"},
    {{c, 1}, 1, columns, 1, Appearance -> "Labeled"},
    {{bin, 2}, 1, 5, 1, Appearance -> "Labeled"},
    {{pg, gratingPeriodMicron}, 0.5 * gratingPeriodMicron, 1.5 * gratingPeriodMicron, Appearance -> "Labeled"}]

Step 3: For any angle, process reference and sample

initialize vectors used for the calculation

funcPrepareAllVectors[gratingPeriodMicron,
  listGratingStepsMicron, rows, columns] = {
  Dimensions[gMatrix], Dimensions[bVector], Dimensions[aVector],
  Dimensions[aMatrix], Dimensions[visibility], Dimensions[phi]}

set angle for calculation, find the correct filenames

translation = listSampleTranslation[[2]]
listFilenamesSampleOneInterferogram = funcGetSampleFilenamesForSpecificTranslation[translation]
listFilenamesReferenceOneInterferogram = filenamesReference;
calculate transmission, visibility, and phi for reference and sample: simple phase unwrap

```plaintext
interferogramReference = funcReadOneInterferogramMedianFilterReference[
  listReferenceIndexSequenceGratingNumbers, listGratingStepsMicron];
Dimensions[interferogramReference]
{referenceTransmission, referenceVisibility, referencePhi} =
  funcCalculateTransmissionVisibilityPhi[interferogramReference];
{Dimensions[referenceTransmission], Min[referenceTransmission],
  Mean[Flatten[referenceTransmission]] // N, Max[referenceTransmission]}  
{Dimensions[referenceVisibility], Min[referenceVisibility],
  Mean[Flatten[referenceVisibility]] // N, Max[referenceVisibility]}  
{Dimensions[referencePhi], Min[referencePhi],
  Mean[Flatten[referencePhi]] // N, Max[referencePhi]}  

interferogramSample = funcReadOneInterferogramMedianFilterSample[
  listSampleIndexSequenceGratingNumbers, listGratingStepsMicron, translation];
Dimensions[interferogramSample]
{sampleTransmission, sampleVisibility, samplePhi} =
  funcCalculateTransmissionVisibilityPhi[interferogramSample];
{Dimensions[sampleTransmission], Min[sampleTransmission],
  Mean[Flatten[sampleTransmission]] // N, Max[sampleTransmission]}  
{Dimensions[sampleVisibility], Min[sampleVisibility],
  Mean[Flatten[sampleVisibility]] // N, Max[sampleVisibility]}  

correct for bad pixels: referenceTransmission, sampleTransmission, referenceVisibility

Position[referenceTransmission, p_?((# == 0 &)]]
referenceTransmission =
  funcCorrectForZeroValuePixels[referenceTransmission];
Position[referenceTransmission, p_?((# == 0 &)]]

Position[sampleTransmission, p_?((# == 0 &)]]
sampleTransmission = funcCorrectForZeroValuePixels[sampleTransmission];
Position[sampleTransmission, p_?((# == 0 &)]]

Position[referenceVisibility, p_?((# == 0 &)]]
referenceVisibility = funcCorrectForZeroValuePixels[referenceVisibility];
Position[referenceVisibility, p_?((# == 0 &)]]

plot transmission, visibility, and phi for reference and sample

testPointOne = {Round[columns / 2], Round[rows / 1.2]};
climTrans = {Min[referenceTransmission],
  Max[MedianFilter[referenceTransmission, 1]]};
referenceVisibilityPercent = 100 * referenceVisibility
  referenceTransmission
climVis = {0, 1.5 * Median[Flatten[referenceVisibilityPercent]]};
climPhi = {-1, 1} * Max[MedianFilter[referencePhi, 1]];
gAllReference = funcPlotTransmissionVisibilityPhiFit[
  testPointOne, "ref", referenceTransmission,
  referenceVisibility, referencePhi, interferogramReference]
Export[pathFigures <> testNameStr <> ", trans_vis_DP_fit_" <>
  "reference" <> ".png", gAllReference, "PNG"]
```
transmission, dist = 91.6667, test point = {640, 900}
gAllSample = funcPlotTransmissionVisibilityPhiFit[testPointOne, "sample",
  sampleTransmission, sampleVisibility, samplePhi, interferogramSample]
Export[pathFigures <> testNameStr <> "_trans_vis_DF_fit_" <>
  ToString[translation] <> ".png", gAllSample, "PNG"]
create ImageBadPixelMask from referenceVisibility <3%

referenceVisibilityPercent = 100 * referenceVisibility / referenceTransmission;
indexBadMask = Position[referenceVisibilityPercent, p_7 (p <= 3 &)];
dataBadMask = ConstantArray[0, Dimensions[referenceVisibilityPercent]];
Table[Module[{},
    {r, c} = indexBadMask[[i]];
    dataBadMask[[r, c]] = 1, {i, Length[indexBadMask]}];
imageBadPixelMask = Image[dataBadMask, "Bit"]
Clear[dataBadMask]

calculate absorption, differential phase contrast, and dark-field
Inpaint using imageBadPixelMask. Apply medianFilter (r=1)

{Min[absorption], Mean[Flatten[absorption]], Max[absorption]}
differentialPhase = samplePhi - referencePhi;
{Min[differentialPhase], 
Mean[Flatten[differentialPhase]], Max[differentialPhase]}
darkfield = sampleVisibility / referenceVisibility
sampleTransmission / referenceTransmission;
{Min[darkfield], Mean[Flatten[darkfield]], Max[darkfield]}

plot absorption, differential phase contrast, and dark-field

plotLabel = "horizontal sensitivity";
climAbs = {-0.05, 0.5}; climDPC = 1*{-\pi, \pi};
climDarkField = {0, 1.2};
gAbsDpcDF = funcPlotAbsorptionDPCDarkField[
    plotLabel, absorption, differentialPhase, darkfield]
With the mouse pointer positioned over an image, right-click and select Get Indices. The mouse pointer then changes into a cross with a tooltip next to it that indicates the current row and column position of the pointer.

(rows, columns)

ImageAdjust[Image[absorption, "Real"], {0, 0, 1}, {-0.05, 0.05}]

doseROI = {{1, 1}, {100, 100}}
rowROI = {doseROI[[1, 2]], doseROI[[2, 2]]}
columnROI = {doseROI[[1, 1]], doseROI[[2, 1]]}

ImageResize[ImageAdjust[
    Image[Take[absorption, columnROI, rowROI], "Real"], {100, 100}]
meanBackgroundAbsorptionSelectedAngle =
    Median[Flatten[Take[absorption, columnROI, rowROI]]]
meanBackgroundDPCSelectedAngle =
    Median[Flatten[Take[differentialPhase, columnROI, rowROI]]]
meanBackgroundDarkFieldSelectedAngle =
    Median[Flatten[Take[darkfield, columnROI, rowROI]]]

plot absorption, differential phase contrast, and dark-field

plotLabel = testNameStr <> " dist" <> ToString[translation]
climAbs = {0, 0.8}; climDPC = 0.5 * {-\pi, \pi};
climDarkField = {0, 1};
gAbsDpcDF = funcPlotAbsorptionDPCDarkField[
    plotLabel, absorption, differentialPhase, darkfield]

(*Export[pathFigures<>testNameStr<>"_abs_DPC_DF_corrected_"<>ToString[translation]<>".png",gAbsDpcDF,"PNG"]*)

Step 4: For all angles, process reference and sample

initialize vectors used for the calculation

funcPrepareAllVectors[gratingPeriodMicron,
    listGratingStepsMicron, rows, columns]
{Dimensions[gMatrix], Dimensions[bVector], Dimensions[aVector],
    Dimensions[aMatrix], Dimensions[visibility], Dimensions[phi]}

for all angles, calculate transmission, visibility, and phi for reference and sample, store HDF5, FITS

listSampleTranslation
interferogramReference = funcReadOneInterferogramMedianFilterReference[
    listReferenceIndexSequenceGratingNumbers, listGratingStepsMicron];
{referenceTransmission, referenceVisibility, referencePhi} =
    funcCalculateTransmissionVisibilityPhi[interferogramReference];

listTimeTranslation = Table[
    Module[{},
        translation = listSampleTranslation[[indexListTranslation]];
        interferogramSample = funcReadOneInterferogramMedianFilterSample[
            listSampleIndexSequenceGratingNumbers, listGratingStepsMicron, translation];
        Print[{indexListTranslation, translation}];
        {sampleTransmission, sampleVisibility, samplePhi} =
            funcCalculateTransmissionVisibilityPhi[interferogramSample];
        absorption = -Log[
            sampleTransmission
            ] / referenceTransmission; // N;
        differentialPhase = samplePhi - referencePhi;
        darkfield = sampleVisibility / referenceVisibility;
        sampleTransmission / referenceTransmission;
        
        filenameHDF5 = testNameStr <> "_" <> ToString[NumberForm[
            translation, {6, 4}, NumberPadding -> {"0", "0"}]] <> ".h5";
        filenameFITsAbs = testNameStr <> "_abs_" <> ToString[NumberForm[
            translation, {6, 4}, NumberPadding -> {"0", "0"}]] <> ".mm.fits";
        filenameFITsDpc = testNameStr <> "_dpc_" <> ToString[NumberForm[
            translation, {6, 4}, NumberPadding -> {"0", "0"}]] <> ".mm.fits";
        filenameFITsDarkfield = testNameStr <> "_darkfield_" <>
            ToString[NumberForm[translation, {6, 4},
            NumberPadding -> {"0", "0"}]] <> ".mm.fits";
        Export[pathHDF5 <> filenameHDF5, {absorption, differentialPhase, darkfield}, 
            "Datasets", 
            {"absorption", "dpc", "darkfield"}]]; 
        Export[pathProjections <> filenameFITsAbs, absorption];
        Export[pathProjections <> filenameFITsDpc, differentialPhase];
        Export[pathProjections <> filenameFITsDarkfield, darkfield];
        {translation} ,
        {indexListTranslation, 1, Length[listSampleTranslation]]};

---

Step 5: Scatterplots of the dark-field and attenuation

230
Appendix E: Neutron Talbot-Lau Interferometry TomoPy Code

The following TomoPy code is used for neutron Talbot-Lau interferometry experiments of additively manufactured samples at HZB in March 2017. This code was utilized for 2 and 3 mm thick SS316 AM and conventional samples at HZB, but also for image processing of the thin SS316 samples imaged at Oak Ridge National Laboratory. In comparison to the Mathematica code in Appendix D, this Python based code is constantly updated from users at X-ray and neutron facilities across the globe.
white beam for tomography/interferometry (March, 2017, Les)

Step 1: Initialization

```
In [3]: # -*- coding: utf-8 -*-
from __future__ import print_function
import os
from os import listdir
from os.path import isfile, join
import numpy
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
import string
from numpy.linalg import inv, matrix
import matplotlib
import scipy
import scipy.signal
import scipy.ndimage
import scipy.signal
import scipy.stats
import scipy.optimize
from matplotlib import pyplot
np.set_printoptions(suppress=True)  # no scientific notations
import re
import fits
from Python.display import clear_output, #clear_output
```

Descriptions: make sure you put "raw" file in the correct path

```
In [4]: # user defined section
workdir = '/run/media/tomosauer/Mercury550/HZB_March2017/Expt0_AM1ens_Helix_3D_v2'  # change your path here
rawPath = workdir
yourRawFilename = '0_Tomography/'
darkFilename = '1_Darkfield/'
whiteFilename = '0_Flatfield/
angleIncrement = 360.0 / (1238 - 1)  # degree
par = angleIncrement
try:
    os.stat(os.path.join(workdir, 'FITS'))  # create export file: FITS
except:
    os.mkdir(os.path.join(workdir, 'FITS'))
```

Step 2: functions

Step 2.1: Reading files

```
In [5]: def readFile(path, string):
   file = 'for f in listdir(path) if isfile(f) end(path, f)]
   allFiles = [f for f in listdir(path) if isfile(f) join(path, f)]
   allFiles = [join(path, allFiles[n])
   allFiles.sort(key=lambda var:int(var)) if x.isdigit() else x for x in re.findall('[\d-]+', var)]
   # use "regular expression" to sort filenames
   listfile = []
   for seq in range(0, len(allFiles)):
       if string in allFiles[seq]:
           listfile.append(allFiles[seq])
   print ('There are ' + str(len(listfile)) + ' files)
return (listfile)
```
def funcReadAndDownloadTIFFFile(filename, cropRows, cropColumns, binning, referenceDarkfield):
    binning = 2
    image = plt.imread(filename)
    data = np.array(image)
    data = data[cropRows[0]:cropRows[1], cropColumns[0]:cropColumns[1]]
    data = scipy.ndimage.zoom(data, 1.0/binning, order=1) #linear resampling
    data = data - referenceDarkfield
    return(data)

Step 2.2: grouping

def funcGetSampleFileNamesForSpecificAngle(numSample, angle):
    numSteps = 22
    end = (angle/par1)*12 - 1
    start = end - (numSteps - 1)
    sampleFileForThisAngle = []
    for i in range(start, end+1):
        sampleFileForThisAngle.append(sampleFile[i])
    return(sampleFileForThisAngle)

def funcGetWhiteFileNamesForSpecificAngle(stack):
    n=5 stacks of white images
    numSteps = 12
    end = stack*12-1
    start = end - (numSteps - 1)
    whiteFileForThisStack = []
    for i in range(start, end+1):
        whiteFileForThisStack.append(whiteFile[i])
    return(whiteFileForThisStack)

Step 2.3: Function for interferometry

def funcPrepareVectorArbitrarySteps(gratingPeriodMicrons, listGratingStepsMicrons):
    numberGratingSteps = len(listGratingStepsMicrons)
    b1 = np.ones([numberGratingSteps])
    b2 = np.zeros([numberGratingSteps])
    b3 = np.zeros([numberGratingSteps])
    for i in range(numberGratingSteps):
        [b1[i], b2[i], b3[i]] = np.cos(2*np.pi*listGratingStepsMicrons[i] / gratingPeriodMicrons)
    return(np.column_stack([b1, b2, b3]))

def funcPrepareAllVectors(gratingPeriodMicrons, listGratingStepsMicrons, rows, columns):
    bVector = funcPrepareVectorArbitrarySteps(gratingPeriodMicrons, listGratingStepsMicrons)
    phiVector = np.zeros((rows, columns))
    visibility = np.zeros((rows, columns))
    gMatrix = np.dot(np.dot(np.transpose(bVector), bVector)), np.transpose(bVector))
    return(gMatrix, bVector, phiVector, visibility, phi)

def funcReadOneInterferogram(listFileNamesOneInterferogram, rows, columns, listGratingStepsMicrons):
    allData = np.zeros([rows, columns, numberGratingSteps])
    for index in range(numberGratingSteps):
        fileName = listFileNamesOneInterferogram[index]
        intensity = funcReadAndDownloadTIFFFile(fileName, cropRows, cropColumns, binning, referenceDarkfield)
        allData[:,:,index] = intensity
    return(allData)
In [12]:
def funcCalculateTransmissionVisibilityPhi(data):
    numberGratingSteps = len(listGratingStepsMicrons)
    cVector = np.transpose(np.reshape(data, (rows*columns, numberGratingSteps)))
    aMatrix = np.reshape(np.transpose(cVector). (rows, columns, 3))
    transmission = aMatrix[:,:,0]
    visibility = np.sqrt(aMatrix[:,:,1]**2 + aMatrix[:,:,2]**2)
    phi = np.arctan2(aMatrix[:,:,2], aMatrix[:,:,1])
    visibilityPercent = 100 * visibility / transmission
    return(transmission, visibility, phi)

In [13]:
def funcCorrectForZeroValuePixels(dataOriginal):
    coordinatesZeroIntensityPixels = np.transpose(np.where(dataOriginal == 0))
    dataCorrected = dataOriginal
    if len(coordinatesZeroIntensityPixels > 0):
        for index in range(len(coordinatesZeroIntensityPixels):
            [r, c] = coordinatesZeroIntensityPixels[index]
            dataCorrected[r, c] = dataMedianFilter[r, c]
    return(dataCorrected)

In [14]:
def funcCorrectLargeDPCValue(data):
    dataCorrected = data
    coordinatesLargeDPC = np.transpose(np.where(data > np.pi))
    if len(coordinatesLargeDPC) > 0:
        for index in range(len(coordinatesLargeDPC):
            [r, c] = coordinatesLargeDPC[index]
            dataCorrected[r, c] = data[r, c] - 2*np.pi
        coordinatesLargeDPC = np.transpose(np.where(differentialPhase < -np.pi))
    if len(coordinatesLargeDPC) > 0:
        for index in range(len(coordinatesLargeDPC):
            [r, c] = coordinatesLargeDPC[index]
            dataCorrected[r, c] = data[r, c] + 2*np.pi
    return(dataCorrected)

Step 3: Paths, filenames and grouping

In [15]:
rawData = os.path.join(rawPath, yourRawFilename)
fitsFile = os.path.join(workdir, 'FITS/'
print ("Your working directory is " + workdir + "")
print ()
Your working directory is " /run/media/tomouser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2 "

In [48]:
    whiteFiles = readFile(whiteFilename, 'Flat') #flat-field
    sampleFiles = readFile/rawData, 'Tomography_' ) #1506, 1501, 2806, 2808 image files are missing
    darkFiles = readFile/darkFilename, 'Dark'
    print()
    numWhite = len(whiteFiles)
    numSample = len(sampleFiles)
    numDark = len(darkFiles)
    print(np.transpose(whiteFiles[0:4]))
    print()
    print(np.transpose(sampleFiles[0:4]))

There are 60 Flat files
There are 2760 Tomography_ files
There are 15 Dark files

['0_Flatfield/Flatfield_0001.tif', '0_Flatfield/Flatfield_0002.tif',
 '0_Flatfield/Flatfield_0003.tif', '0_Flatfield/Flatfield_0004.tif']
['/run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0001.tif',
 '/run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0002.tif',
 '/run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0003.tif',
 '/run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0004.tif']

Cropping

235
In [46]: oneImg = np.array(plt.imread(sampleFiles[0]))
[NY, NX] = oneImg.shape
[rows, columns] = oneImg.shape
print("Before cropping:")
plt.imshow(oneImg, cmap = "gray")
plt.colorbar()
plt.show()
cropRows = [300, 1500]
cropColumns = [9, NX]
binning = 2
oneImageCrop = oneImg[cropRows[0]: cropRows[1], cropColumns[0]: cropColumns[1]]
oneImageCrop = scipy.ndimage.zoom(oneImageCrop, 1./binning, order=1) #Linear resampling
print("After cropping and binning:", oneImageCrop.shape)
plt.imshow(oneImageCrop, cmap = "gray")
plt.colorbar()
plt.show()

Before cropping:

After cropping and binning: (800, 1024)

based on cropping and binning, calculate dark-field
In [47]: [NY, NX]=[cropRows[1]-cropRows[0]+1, cropColumns[1]-cropColumns[0]+1]
print (['FileInfo', NX, NY, 'FileInfo', numDark])
binning = 2
referenceDarkfield = np.zeros((numDark, NY/2, NX/2))
for i in range(numDark):
    image = np.array(pil.imread('FileInfo')
    image = image[cropRows[0]:cropRows[1], cropColumns[0]:cropColumns[1]]
    image = scipy.ndimage.zoom(image, 1.0/binning, order=1)
referenceDarkfield[0,..., ] = image
print (['FileInfo', refDarkfield.shape])
referenceDarkfield = np.median(referenceDarkfield, axis=0)
print (['FileInfo', refDarkfield.shape])
print (['FileInfo', np.min(referenceDarkfield), np.mean(referenceDarkfield), np.max(referenceDarkfield)])
[rows, columns] = referenceDarkfield.shape

1200.0
796.8

Out[47]: (600, 1024)

Step 4: Look at one angle here

In [60]: # user defined
    angle = 360.0/299.0 # must be divided by 360/299
    print ("the one angle you want to look at is: ", angle, '°')
    print ('if not, please change angle = ', angle, '° above. Thanks!')

the one angle you want to look at is: 1.20401337793

if not, please change angle = 1.20401337793 above. Thanks!

In [61]: numSteps = 12
finalAngle = 360.0
print ("the number of steps is: ", numSteps, "the sample rotates from 0 degree to", finalAngle,
    "degree")
angleIncrement = 360/299
numberOfAngles = numSample / numSteps
print ("the number of angles is": numberOfAngles)

number of steps is: 12
the sample rotates from 0 degree to 360 degree angle increment is 360/299 = 1.57205240175 degrees
the number of angles is: 230
In [62]:
listFilenamesSampleOneInterferogram = funcGetSampleFilenamesForSpecificAngle(numSample, angle)
print('Number of angle', angle, 'sample files: ', len(listFilenamesSampleOneInterferogram))

'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0104.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0105.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0106.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0107.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0108.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0109.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0110.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0111.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0112.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0113.tif'
'run/media/tomuser/MercurySSD/HZB_Mar2017/Expt8_AM_TENS_Horiz_3D_v2/0_Tomography/Tomography_0114.tif'

Number of angle 1.20401337793 sample files: 12

In [63]:
gratingPeriodMicrons = 790 #pixel size = 7 microns
listGratingStepsMicrons = np.zeros(12)
numGratingSteps = numSteps
for i in range(0, 12):
    listGratingStepsMicrons[i] = 1200/(numGratingSteps-1)*i
print(listGratingStepsMicrons)

cropping range:
[306, 1508]
[0, 2048]
[0, 1099]

In [64]:
sample name: 'sampleFiles[0].split("/\")[6]
print('grating period (in microns): ', gratingPeriodMicrons)
print('grating steps (in microns): ', listGratingStepsMicrons)
print('number of grating steps: ', len(listGratingStepsMicrons))
print('number of dark images: ', len(darkFiles))
print('number of reference images (white): ', len(whiteFiles))
print('number of sample images (raw): ', len(sampleFiles))

sample name: Expt8_AM_TENS_Horiz_3D_v2
grating period (in microns): 790
grating steps (in microns): 15
number of grating steps: 12
number of dark images: 15
number of reference images (white): 60
number of sample images (raw): 2760
```python
In [65]: listSampleAngles = np.zeros((np.int(finalAngle//par)+1))
for i in range(len(listSampleAngles)):
    listSampleAngles[i] = i*par
print('There are', len(listSampleAngles), 'angles', '\\n', '\n', 'numberOfAngles = len(listSampleAngles)
#clear_output()')
368/199
0.0

There are 230 angles:

[ 0.  1.572614  3.141593  4.716372  6.283185]
15.7263342  17.3275673  18.8646286  20.4017106  21.9378428
23.5699742  25.1075556  26.7469075  28.3865578  29.9912773
31.6217529  33.2412083  34.8606953  36.4801503  38.1095954
39.7391455  41.3685905  42.9980351  44.6274797  46.2569243
47.8863693  50.1075426  51.7369767  53.3664108  54.9958448
56.6252791  58.2547132  59.8841473  61.5135813  63.1429153
64.7723494  66.4017935  68.0312276  69.6606717  71.2891059
72.9185400  74.5479741  76.1774081  77.8068422  79.4362863
81.0657305  82.6951646  84.3245987  85.9540328  87.5834669
89.2128610  90.8422951  92.4717292  94.1011633  95.7305974
97.3596015  98.9890346  100.6184687  102.2479028  103.8773369
105.5067710  107.1362051  108.7656392  110.3950733  112.0245074
113.6539415  115.2833756  116.9128097  118.5422438  119.1716779
120.8011120  122.4305461  124.0599802  125.6904143  127.3198484
128.9492825  130.5787166  132.2081507  133.8375848  135.4663519
137.0950860  138.7245201  140.3531542  141.9826883  143.6122224
145.2417565  146.8712906  148.5008247  149.1303588  150.7598929
152.3894270  154.0189611  155.6484952  157.2780293  158.9075634
160.5370975  162.1666316  163.7961657  165.4257007  167.0552348
168.6847689  170.3143030  171.9438371  173.5733712  175.2029053
176.8324394  178.4619735  180.0915076  181.7210417  183.3505757
184.9801098  186.6096439  188.2391780  189.8687121  191.4982463
193.1277803  194.7573144  196.3868485  198.0163826  199.6459167
201.2754508  202.9049849  204.5345190  206.1640531  207.7935872
209.4231213  211.0526554  212.6821895  214.3117236  215.9412577
217.5707918  219.2003259  220.8298600  222.4593941  224.0889282
225.7184623  227.3480064  228.9775405  230.6070746  232.2366087
233.8661428  235.4956769  237.1252110  238.7547451  240.3842792
241.0138133  242.6433474  244.2728815  245.9024156  247.5319497
249.1614838  250.7900179  252.4195520  254.0490861  255.6786201
257.3081542  258.9376883  260.5672224  262.1967565  263.8262906
265.4558247  267.0853588  268.7148929  270.3444270  271.9739611
273.6034952  275.2330293  276.8625634  278.4921668  280.1217610
281.7513541  283.3809482  285.0105423  286.6401364  288.2697305
289.8993246  291.5289187  293.1585128  294.7881069  296.4176990
298.0472931  299.6768872  301.3064813  302.9360754  304.5656695
306.1952636  307.8248577  309.4544518  311.0840459  312.7136399
314.3432340  315.9728281  317.6024222  319.2320163  320.8616104
322.4912045  324.1207986  325.7503927  327.3800868  329.0096809
330.6392750  332.2688671  333.8984592  335.5280513  337.1576454
338.7872395  340.4168336  342.0464277  343.6760218  345.3056159
346.9352099  348.5648040  350.1943981  351.8239922  353.4535863
355.0831804  356.7127745  358.3423686  359.9719627  361.6015568]

Out[65]: 1.8999452261306533

initialize vectors used for the calculation

239
In [66]:

```
print (gMatrix.shape)
print (bVector.shape)
print (aVector.shape)
print (gMatrix.shape)
print (visibility.shape)
print (phi.shape)

(3, 12)
(12, 3)
(3, 614400)
(600, 1024, 3)
(600, 1024)
```

calculate white/reference — transmission, phi and visibility

In [67]:

```
nnumStacksForWhite = numWhite / numSteps
print ("There are", numStacksForWhite, "stacks flat-field/white samples")
stack = 1 #stack = 1, 2, 3, 4, 5
print ("stack", stack, ", stack")
listfilenamesReferenceOneInterferogram = getWhiteFilenamesForSpecificAngle(stack)
print (np.transpose(listfilenamesReferenceOneInterferogram))
```

There are 5 stacks flat-field/white samples

```
stack: 1

['O_Flatfield/Flatfield_0001.tif', 'O_Flatfield/Flatfield_0002.tif',
'0_Flatfield/Flatfield_0003.tif', 'O_Flatfield/Flatfield_0004.tif',
'O_Flatfield/Flatfield_0005.tif', 'O_Flatfield/Flatfield_0006.tif',
'O_Flatfield/Flatfield_0007.tif', 'O_Flatfield/Flatfield_0008.tif',
'O_Flatfield/Flatfield_0009.tif', 'O_Flatfield/Flatfield_0010.tif',
'O_Flatfield/Flatfield_0011.tif', 'O_Flatfield/Flatfield_0012.tif']
```

In [68]:

```
listinterferogramReference = readOneInterferogram(listfilenamesReferenceOneInterferogram, rows, columns,
listGratingStepsMicrons)
print (listinterferogramReference.shape)
```

```
(600, 1024, 12)
```

In [69]:

```
np.seterr(divide='ignore', invalid='ignore') #code have some "divided by zero" or "Nan" appears
referenceTransmission, referenceVisibility, referencePhi = \
calculateTransmmisionVisibilityPhiReferenceInterferogramReference)
print ("reference Transmission:"
print (referenceTransmission.shape, np.min(referenceTransmission), np.max(referenceTransmission))
print ("reference Visibility:")
print (referenceVisibility.shape, np.min(referenceVisibility), np.max(referenceVisibility))
print ("reference Phi")
print (referencePhi.shape, np.min(referencePhi), np.max(referencePhi))
```

```
reference Transmission:
(600, 1024) 1.480 3407 0185 5489.58351978 9284.432344818
reference Visibility
(600, 1024) 1.62101585672 358.633963214 1193.77965972
reference Phi
(600, 1024) -3.14158664210 0.197292643825 3.1415863598
```

240
```python
In [79]:
print("reference transmission:")
plt.imshow(referenceTransmission, cmap = "gray")
plt.show()
print("reference phi")
plt.imshow(referencePhi, cmap = "gray")
plt.show()
print("reference visibility")
plt.imshow(referenceVisibility, cmap = "Greys_r")
plt.show()
```

reference transmission:

![Reference Transmission Image](image1)

reference phi

![Reference Phi Image](image2)

reference visibility

![Reference Visibility Image](image3)
median processing for flat-field/white/reference 12 steppings

In [71]:
refTrans = np.zeros((numStacksForWhite, rows, columns))
refVis = np.zeros((numStacksForWhite, rows, columns))
refPhi = np.zeros((numStacksForWhite, rows, columns))
for i in range(numStacksForWhite):
    #print (i)
    listFilenamesReferenceOneInterferogram = funcGetWhiteFilenamesForSpecificAngle([i])
    #print (np.transpose(listFilenamesReferenceOneInterferogram))
    interferogramReference = funcReadOneInterferogram(listFilenamesReferenceOneInterferogram, rows, columns, 
    listRatingStepsMicrons)
    [referenceTransmission, referenceVisibility, referencePhi] = 
    funcCalculateTransmissionVisibilityPhi(interferogramReference)
    refTrans[i,:,:] = referenceTransmission
    refVis[i,:,:] = referenceVisibility
    refPhi[i,:,:] = referencePhi
    ***
    plt.imshow(refTrans[i,:,:], cmap = 'Greys_r')
    plt.show()
    plt.imshow(refVis[i,:,:], cmap = 'Greys_r')
    plt.show()
    ***

In [72]:
index=0
print ("reference transmission")
plt.imshow(refTrans[index,:,:], cmap = 'gray')
plt.show()
print ("reference phi")
plt.imshow(refPhi[index,:,:], cmap = 'gray')
plt.show()
print ("reference visibility")
plt.imshow(refVis[index,:,:], cmap = 'Greys_r')
plt.show()
clean_output()

In [73]:
print (np.median(refTrans, axis=0).shape)
referenceTransmission = np.median(refTrans, axis=0)
referencePhi = np.median(refPhi, axis=0)
referenceVisibility = np.median(refVis, axis=0)

(600, 1024)
In [74]:

    print("reference transmission")
    plt.imshow(referenceTransmission, cmap = "gray")
    plt.show()

    print("reference phi")
    plt.imshow(referencePhi, cmap = "gray")
    plt.show()

    print("reference visibility")
    plt.imshow(referenceVisibility, cmap = "gray")
    plt.show()
calculate samples -- transmission, phi and visibility

In [75]: interferogramSample = funcReadOneInterferogram(listFileNamesSampleOneInterferogram, rows, columns, ListGratingStepsMicrons)
   print [interferogramSample.shape]
   (600, 1024, 12)

In [76]: [sampleTransmission, sampleVisibility, samplePhi] = funcCalculateTransmissionVisibilityPhi(interferogramSample)
   print ('sample Transmission:')
   print (sampleTransmission.shape, np.min(sampleTransmission), np.mean(sampleTransmission), np.max(sampleTransmission))
   print ('sample Visibility')
   print (sampleVisibility.shape, np.min(sampleVisibility), np.mean(sampleVisibility), np.max(sampleVisibility))
   print ('sample Phi')
   print (samplePhi.shape, np.min(samplePhi), np.mean(samplePhi), np.max(samplePhi))

sample Transmission:
(600, 1024) 664.576376281 5229.69710017 9479.18488831
sample Visibility
(600, 1024) 8.116209318563 218.372651894 1862.7055448
sample Phi
(600, 1024) -3.341598648351 -8.379684548073 3.341598515448

In [77]:
   print [np.transpose(np.where(refernceTransmission == 0))]
   referenceTransmission = funcCorrectForZeroValuePixels(refernceTransmission)
   print [np.transpose(np.where(refernceTransmission == 0))]
   sampleTransmission = funcCorrectForZeroValuePixels(sampleTransmission)
   print [np.transpose(np.where(sampleTransmission == 0))]
   print [np.transpose(np.where(refernceVisibility == 0))]
   referenceVisibility = funcCorrectForZeroValuePixels(refernceVisibility)
   print [np.transpose(np.where(refernceVisibility == 0))]

plot trans, vis, and phi for ref and sample

In [78]: testPointOne = (np.round(columns/2), np.round(rows/2))
   print [testPointOne]
   limTrans = [np.min(refernceTransmission), np.max(refernceTransmission)]
   limPhi = [np.min(referncePhi), np.max(referncePhi)]
   limVis = [np.min(refernceVisibility), np.max(refernceVisibility)]

   (512, 300)

In [79]:
   plt.imshow(refernceTransmission, cmap='gray')
   plt.colorbar()
   plt.show()
   plt.imshow(refernceVisibility, cmap='gray')
   plt.show()
   plt.imshow(referncePhi, cmap='gray')
   plt.show()
   clear_output()
calculate abs, dpc, and dark-field
In [81]: absorption = -np.log(sampleTransmission / referenceTransmission)
    print [np.min(absorption), np.mean(absorption, flatten()), np.max(absorption)]
        -0.1865720476 8.921181013344 1.55146589957

In [82]: differentialPhase = samplePhi - referencePhi
    print [np.min(differentialPhase), np.mean(differentialPhase, flatten()), np.max(differentialPhase)]
        -6.18534169372 6.61898899392 6.21258528859

In [83]: differentialPhase = funcCorrectLargeOPValue(differentialPhase)
    print [np.min(differentialPhase), np.mean(differentialPhase, flatten()), np.max(differentialPhase)]
        -3.1415985701 -1.44734577255 3.1415915697

In [84]: darkfield = [sampleVisibility / referenceVisibility] / [sampleTransmission / referenceTransmission]
    print [np.min(darkfield), np.mean(darkfield), np.max(darkfield)]
        0.00128907997511 0.63357856373 8.53538310199
In [85]:
plt.imshow(absorption, cmap='gray')
plt.show()
plt.imshow(differentialPhase, cmap='gray', vmax=np.pi, vmin=-np.pi)
plt.show()
plt.imshow(darkfield, cmap='gray', vmax=1, vmin=0)
plt.show()
#clear_output()

Step 5: for all angles, process ref and sample
In [86]: [gMatrix, bVector, aVector, hMatrix, visibility, phi] = funcPrepareAllVectors(gratingPeriodMicrons, listGratingStepsMicrons, rows, columns)
   print([gMatrix.shape, bVector.shape, aVector.shape, hMatrix.shape, visibility.shape, phi.shape])

   (3, 12) (12, 3) (3, 614400) (600, 1024, 3) (600, 1024) (600, 1024)

In [87]: oldlistFilenamesReferenceOneInterferogram = [""
sampleName = sampleFiles[i].split("/")[6]
print('sample name: ', sampleName)
numberOfAngles

  sample name: ExptB_AM_TENS_Horiz_3D_v2

Out[87]: 230
In [88]:
print("reference transmission:")
plt.imshow(referenceTransmission, cmap = "gray")
plt.show()
print("reference phi")
plt.imshow(referencePhi, cmap = "gray")
plt.show()
print("reference visibility")
plt.imshow(referenceVisibility, cmap = "Greys_r")
plt.show()

reference transmission:

reference phi

reference visibility
In [89]: numberofAngles, rows, columns
Out[89]: (230, 698, 1824)

In [90]: for indexListAngle in range(numberofAngles):
    
        angle = listSampleAngles[indexListAngle]
        referenceTransmission = referenceTransmission
        referencePhi = referencePhi
        referenceVisibility = referenceVisibility
        listFilenamesSampleOneInterferogram = funcGetSampleFilenamesForSpecificAngle(numSample, angle)
        print([indexListAngle, ' # of sample images = ', len(listFilenamesSampleOneInterferogram)])
        interferogramSample = funcReadOneInterferogram(listFilenamesSampleOneInterferogram, rows, columns, listGratingStepsMicrons)
        print([indexListAngle, angle]
        [sampleTransmission, sampleVisibility, samplePhi] = funcCalculateTransmissionVisibilityPhi(interferogramSample)
        absorption = -np.log(sampleTransmission / referenceTransmission)
        differentialPhase = samplePhi - referencePhi
        differentialPhase = funcCorrectLargeDPCValue(differentialPhase)
        darkfield = (sampleVisibility / referenceVisibility) / (sampleTransmission / referenceTransmission)
        #darkfield = 1 - darkfield
        #write abs FITS files
        absFilename = os.path.join(fitsFile, sampleName + '_abs_' + np.str(angle/par1) + '.fit')
        try:
            #remove file if exists
            os.remove(absFilename)
            except OSError:
                pass
            absFits = fits.HDUList([fits.PrimaryHDU(absorption)])
            absFits.writeto(absFilename)
            absFits.close()
        #write dpc FITS files
        dpcFilename = os.path.join(fitsFile, sampleName + '_dpc_' + np.str(angle/par1) + '.fit')
        try:
            #remove file if exists
            os.remove(dpcFilename)
            except OSError:
                pass
            dpcFits = fits.HDUList([fits.PrimaryHDU(differentialPhase)])
            dpcFits.writeto(dpcFilename)
            dpcFits.close()
        #write darkfield FITS files
        darkfieldFilename = os.path.join(fitsFile, sampleName + '_darkfield_' + np.str(angle/par1) + '.fit')
        try:
            #remove file if exists
            os.remove(darkfieldFilename)
            except OSError:
                pass
            darkfieldFits = fits.HDUList([fits.PrimaryHDU(darkfield)])
            darkfieldFits.writeto(darkfieldFilename)
            darkfieldFits.close()
            clear_output()
Appendix F: Neutron Bragg Edge Mathematica Code

The following Mathematica code is used for a 2017 Bragg edge neutron imaging experiment at HZB, Berlin. The samples of interest were 2 and 3 mm thick SS316 AM dogbones that were pulled and bent, respectively. First, functions are defined for the experimental setup and image processing. Next, files are uploaded and cropped. In the code below, some SS316 AM cubes and crystals were cropped out to solely focus on the additively manufactured dogbones. Once a mask is made of the fractured, half-life, and pristine dogbones, the transmission is calculated for each wavelength of the neutron beam. Finally, Bragg edges can be plotted and exported.
Expt14_Bragg_Tension

19 March 2017
This code reads
data from /Raw_Data/BraggEdge_TENS_2A0_4A5_step0A02 17 03 17_10
08/1_Energy_Scan/
flat field from/ Raw_Data/Bragg_OpenBeam 17 03 16_16 16/1_Energy_Scan/
dark from /Raw_Data/AM_TENS_HORIZ_3D_V2 17 03 15_12
28/1_Darkfield/

Step 1: Define functions: 2D imaging

Functions for reading file, finding number of groups of reference files and sample files (Andor NEO, Andor iKon)

funcReadPilatusFile[filename_] This will read data from a Pilatus integer-32 TIFF file.
The standard Mathematica Import[filename] cannot read integer-32 TIFF.
funcFindReferenceTIFFfiles[pathTIFF_] Finds files based on ".white." and orders the
list based on the image sequence number.
funcFindSampleTIFFfiles[pathTIFF_] Finds files based on ".raw." and orders the list
based on the image sequence number.
funcFindReferenceGroups[] Based on sequence number, finds the grouping of the
reference files.
funcFindSampleGroups[] Based on sequence number, finds the grouping of the sample
files.

funcReadAndorTIFFFile[filename_, cropRows_, cropColumns_,
binning_, referenceDarkfield_] := Module[{image, data},
image = Import[filename];
image = ImageTake[image, cropRows, cropColumns];
image = ImageResize[image,
Scaled[1/binning], Resampling -> "Linear"];
data = ImageData[image, "Bit16"];
data = data - referenceDarkfield];

funcFindReferenceTIFFfiles[pathTIFF_] := Module[{},
filenamesReference = FileNames["*.tif", pathTIFF];
numberOfReferenceFiles = Length[filenamesReference];
filenamesReference]

funcFindSampleTIFFfiles[pathTIFF_] := Module[{},
filenamesSample = FileNames["*.tif", pathTIFF];
numberOfSampleFiles = Length[filenamesSample];
filenamesSample]
Functions for interferometry

funcCorrectForZeroValuePixels[dataOriginal_] :=
Module[{coordinatesZeroIntensityPixels,
    dataCorrected, dataMedianFilter, r, c},
  coordinatesZeroIntensityPixels =
    Position[dataOriginal, p_?(# ≤ 0 &)];
  If[Length[coordinatesZeroIntensityPixels] > 0,
    Module[{},
      dataCorrected = dataOriginal;
      dataMedianFilter = MedianFilter[dataOriginal, 1];
      For[index = 1,
        index ≤ Length[coordinatesZeroIntensityPixels], index++,
        {r, c} = coordinatesZeroIntensityPixels[[index]];
        dataCorrected[[r, c]] = dataMedianFilter[[r, c]];
      ];
    ];
  ];
  If[Length[coordinatesZeroIntensityPixels] > 0,
    dataCorrected, dataOriginal]];
Interferometer steps and period, Image cropping

Paths, filenames

```plaintext
pathRaw = "'/Volumes/LaCie/HZB2017/Raw_Data/BraggEdge_TENS_2A0_4A5_step0A02 17 03 17_10 08/1_Energy_Scan/
pathFlatField =　　"'/Volumes/LaCie/HZB2017/Raw_Data/Bragg_OpenBeam
　　17 03 16_16 16/1_Energy_Scan/
pathDark = "'/Volumes/LaCie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
　　17 03 15_12 28/1_Darkfield/
pathRootProcessed =　　"'/Users/tomo/Dropbox/LSU_AMFatiguePaper/Mathematica/Bragg_edge
　　/Expt14_Bragg_Tension/
pathFITS = pathRootProcessed <> "FITS_new/";
pathMask = pathRootProcessed;
pixelSize = 30; (* microns *)

/Volumes/LaCie/HZB2017/Raw_Data/BraggEdge_TENS_2A0_4A5_step0A02
　　17 03 17_10 08/1_Energy_Scan/

/Volumes/LaCie/HZB2017/Raw_Data/Bragg_OpenBeam
　　17 03 16_16 16/1_Energy_Scan/

/Volumes/LaCie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
　　17 03 15_12 28/1_Darkfield/

/Users/tomo/Dropbox/LSU_AMFatiguePaper/Mathematica/Bragg_edge/
　　Expt14_Bragg_Tension/

listReferenceFilenames =
　funcFindReferenceTIFFfiles[pathFlatField];
listReferenceFilenames[[1]]
numberOfReferenceFiles

listSampleFilenames = funcFindSampleTIFFfiles[pathRaw];
listSampleFilenames[[1]]
numberOfSampleFiles
testNameStr = "Expt14_Bragg_Tension";
/Volumes/LaCie/HZB2017/Raw_Data/Bragg_OpenBeam
　　17 03 16_16 16/1_Energy_Scan/Energy_Scan_0001.tif
126

/Volumes/LaCie/HZB2017/Raw_Data/BraggEdge_TENS_2A0_4A5_step0A02
　　17 03 17_10 08/1_Energy_Scan/Energy_Scan_0001.tif
126

254
```
TableForm[{{"sample name", testNameStr},
    {"# of reference images", numberOfReferenceFiles},
    {"# of sample images", numberOfSampleFiles}
}]
sample name Expt14_Bragg_Tension
# of reference images 126
# of sample images 126

Image cropping and binning: Crop image with ImageTake[ ]. Binning with ImageResize[___,
Scaled[1/binning]]
{columns,rows}=ImageDimensions[image] and {NY,NX}=Dimensions[data]

For titanite cubes in Prog Add Manu: cropRows={2048-202,2045}; cropColumns={1,2048};
For Sr crystal, cropRows={200,550}; cropColumns={251,1750};
ImageAdjust[image = Import[listSampleFileNames[[1]]],
{0, 0, 1}, {0.0, 0.05}]
{columnsOriginal, rowsOriginal} = ImageDimensions[image]
image =
ImageAdjust[ImageResize[image, 200], {0, 0, 1}, {0.0, 0.20}]
cropRows =  rowsOriginal - {750, 200};
cropColumns = {100, 550};
(*cropRows= {200, rowsOriginal};
cropColumns = {100, columnsOriginal} ; *)
binning = 1;

image = Import[listSampleFilenames[[1]]];
image = ImageTake[image, cropRows, cropColumns];
image = 
    ImageResize[image, Scaled[1/binning], Resampling -> "Linear"];
{columns, rows} = ImageDimensions[image]
data = ImageData[image, "Bit16"];
{NY, NX} = Dimensions[data]
image = 
    ImageAdjust[ImageResize[image, 200], {0, 0, 1}, {0.0, 0.05}];
GraphicsRow[{{image, ArrayPlot[data],
     ListPlot[data[[Round[NY/2], All]]]}, ImageSize -> 500}]

{451, 551}

{551, 451}
Based on cropping and binning, calculate a dark-field.

```plaintext
listFileNames = FileNames["*" <> "Darkfield" <> "]", pathDark]
  [Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0001.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0002.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0003.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0004.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0005.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0006.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0007.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0008.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0009.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0010.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0011.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0012.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0013.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0014.tif,
   /Volumes/Lacie/HZB2017/Raw_Data/AM_TENS_HORIZ_3D_V2
   17 03 15_12 28/1_Darkfield/Darkfield_0015.tif]

referenceDarkfield = Table[Module[{},
   image = Import[listFileNames[[index]]];
   image = ImageTake[image, cropRows, cropColumns];
   image = ImageResize[image,
      Scaled[1/binning], Resampling -> "Linear"];
   ImageData[image, "Bit16"], {index,
      Length[listFileNames]}];
Dimensions[referenceDarkfield]
referenceDarkfield = Map[Median, referenceDarkfield, {0}] ;
Dimensions[referenceDarkfield]
{15, 551, 451}
{551, 451}
```

258
{Min[referenceDarkfield], Mean[Flatten[referenceDarkfield]] // N, 
Max[referenceDarkfield]}
ArrayPlot[referenceDarkfield] 
(795, 800.092, 1278)

Based on cropping and binning, calculate masks

funcCropMask[filename_] := Module[{},
dataMask = Import[filename, "Data"][[1]];  
(*{Dimensions[dataMask],Min[dataMask], Mean[Flatten[dataMask]]//N,Max[dataMask]}*)
image = Image[dataMask, "Byte"]; 
image = ImageReflect[image, Top \[\rightarrow\] Bottom];
image = ImageTake[image, cropRows, cropColumns];
image = ImageResize[image, 
    Scaled[1/binning], Resampling \[\rightarrow\] "Linear"];
dataMask = ImageData[image, "Byte"] ];
Map[Last, Map[FileNameSplit,
   FileNames["Mask" <> "\*", pathMask]]] // TableForm

Masks_Tension.zip
Mask_Tension_50percent.fits
Mask_Tension_Crystal_Cube.fits
Mask_Tension_frac_long.fits
Mask_Tension_frac_short.fits
Mask_Tension_pristine.fits

dataMaskAll =
   funcCropMask[pathMask <> "Mask_Tension_Crystal_Cube.fits"];
ArrayPlot[dataMaskAll, PlotRange -> All, ImageSize -> 200]

dataMaskTensionPristine =
   funcCropMask[pathMask <> "Mask_Tension_pristine.fits"];
ArrayPlot[dataMaskTensionPristine,
   PlotRange -> All, ImageSize -> 200]
dataMaskTension50percent =
funcCropMask[pathMask <> "Mask_Tension_50percent.fits"]; ArrayPlot[dataMaskTension50percent,
  PlotRange -> All, ImageSize -> 200]


dataMaskTensionFracShort =
funcCropMask[pathMask <> "Mask_Tension_frac_short.fits"]; ArrayPlot[dataMaskTensionFracShort,
  PlotRange -> All, ImageSize -> 200]
dataMaskTensionFracLong =
  funcCropMask[pathMask <> "Mask_Tension_frac_long.fits"];
ArrayPlot[dataMaskTensionFracLong,
  PlotRange -> All, ImageSize -> 200]

Create list of file index, wavelength
listSampleIndexWavelength
listReferenceIndexWavelength

numberOfReferenceFiles
numberOfSampleFiles
Length[ listWavelengths = Range[2, 4.5, 0.02] ]
126

126
126
listSampleIndexWavelength = Transpose[{Range[1, numberOfSampleFiles], listWavelengths}]
listReferenceIndexWavelength = listSampleIndexWavelength;

{{1, 2.}, {2, 2.02}, {3, 2.04}, {4, 2.06}, {5, 2.08}, {6, 2.1},
{7, 2.12}, {8, 2.14}, {9, 2.16}, {10, 2.18}, {11, 2.2},
{12, 2.22}, {13, 2.24}, {14, 2.26}, {15, 2.28}, {16, 2.3},
{17, 2.32}, {18, 2.34}, {19, 2.36}, {20, 2.38}, {21, 2.4},
{22, 2.42}, {23, 2.44}, {24, 2.46}, {25, 2.48}, {26, 2.5},
{27, 2.52}, {28, 2.54}, {29, 2.56}, {30, 2.58}, {31, 2.6},
{32, 2.62}, {33, 2.64}, {34, 2.66}, {35, 2.68}, {36, 2.7},
{37, 2.72}, {38, 2.74}, {39, 2.76}, {40, 2.78}, {41, 2.8},
{42, 2.82}, {43, 2.84}, {44, 2.86}, {45, 2.88}, {46, 2.9},
{47, 2.92}, {48, 2.94}, {49, 2.96}, {50, 2.98}, {51, 3.},
{52, 3.02}, {53, 3.04}, {54, 3.06}, {55, 3.08}, {56, 3.1},
{57, 3.12}, {58, 3.14}, {59, 3.16}, {60, 3.18}, {61, 3.2},
{62, 3.22}, {63, 3.24}, {64, 3.26}, {65, 3.28}, {66, 3.3},
{67, 3.32}, {68, 3.34}, {69, 3.36}, {70, 3.38}, {71, 3.4},
{72, 3.42}, {73, 3.44}, {74, 3.46}, {75, 3.48}, {76, 3.5},
{77, 3.52}, {78, 3.54}, {79, 3.56}, {80, 3.58}, {81, 3.6},
{82, 3.62}, {83, 3.64}, {84, 3.66}, {85, 3.68}, {86, 3.7},
{87, 3.72}, {88, 3.74}, {89, 3.76}, {90, 3.78}, {91, 3.8},
{92, 3.82}, {93, 3.84}, {94, 3.86}, {95, 3.88}, {96, 3.9},
{97, 3.92}, {98, 3.94}, {99, 3.96}, {100, 3.98}, {101, 4.},
{102, 4.02}, {103, 4.04}, {104, 4.06}, {105, 4.08}, {106, 4.1},
{107, 4.12}, {108, 4.14}, {109, 4.16}, {110, 4.18}, {111, 4.2},
{112, 4.22}, {113, 4.24}, {114, 4.26}, {115, 4.28}, {116, 4.3},
{117, 4.32}, {118, 4.34}, {119, 4.36}, {120, 4.38}, {121, 4.4},
{122, 4.42}, {123, 4.44}, {124, 4.46}, {125, 4.48}, {126, 4.5}}
Step 3: For all sample and reference images, calculate transmission at each wavelength
Use masks to get average transmission in selected regions.

```plaintext
listTensionBraggData = ConstantArray[0, {Length[listWavelengths], 5}];
For[index = 1, index ≤ Length[listWavelengths], index++,
    wavelength = listWavelengths[[index]];
    indexWavelength = Position[listSampleIndexWavelength[[All, 2]],
        _?(# == wavelength &)][[1, 1]];
    indexFile = listSampleIndexWavelength[[indexWavelength, 1]];
    dataReference = funcReadAndorTIFFFile[filenamesReference[[indexFile]],
        cropRows, cropColumns, binning, referenceDarkfield];
    dataReference = funcCorrectForZeroValuePixels[dataReference];
    dataSample = funcReadAndorTIFFFile[filenamesSample[[indexFile]],
        cropRows, cropColumns, binning, referenceDarkfield];
    transmission = dataSample / dataReference;
    transmission = transmission * dataMaskAll;
    filenameFITS = testNameStr <> "_" <>
        IntegerString[Floor[wavelength], 10, 3] <> "p" <> IntegerString[
            Round[FractionalPart[wavelength] * 1000], 10, 3]
        <> "_Ang.fits";
    Export[pathFITS <> filenameFITS, transmission];
    meanTensionPristine =
        Mean[ Select[Flatten[transmission * dataMaskTensionPristine],
            # > 0 &] ] / N;
    meanTension50percent = Mean[ Select[Flatten[
        transmission * dataMaskTension50percent], # > 0 &] ] / N;
    meanTensionFracLong = Mean[ Select[Flatten[
        transmission * dataMaskTensionFracLong], # > 0 &] ] / N;
    meanTensionFracShort = Mean[ Select[Flatten[
        transmission * dataMaskTensionFracShort], # > 0 &] ] / N;
    listTensionBraggData[[index]] = {wavelength,
        meanTensionPristine, meanTension50percent,
        meanTensionFracLong, meanTensionFracShort}
];
```
transmission =
Import[Last[FileNames["*.fits", pathFITS]], "RawData"][[1]];
gBraggTensionMaskVerify = Show[GraphicsGrid[{
  ArrayPlot[(transmission - dataMaskTensionPristine -
dataMaskTension50percent - dataMaskTensionFracLong -
dataMaskTensionFracShort), PlotRange -> {0, 1}]},
  {ArrayPlot[dataMaskTensionPristine],
   ArrayPlot[dataMaskTension50percent]},
  {ArrayPlot[dataMaskTensionFracLong],
   ArrayPlot[dataMaskTensionFracShort]}],
  ImageSize -> 400]
step 4: plot Bragg edges for tension: AM

```
Export[pathRootProcessed <> testNameStr <> "_transmission.txt",
    listTensionBraggDataData, "Table"]
/Users/tomo/Dropbox/LSU_AMFatiguePaper/Mathematica/Bragg_edge/
    Expt14_Bragg_Tension/Expt14_Bragg_Tension_transmission.txt

Export[pathRootProcessed <> testNameStr <> "_mask_verify.png",
    gBraggTensionMaskVerify, "PNG"]
/Users/tomo/Dropbox/LSU_AMFatiguePaper/Mathematica/Bragg_edge/
    Expt14_Bragg_Tension/Expt14_Bragg_Tension_mask_verify.png

```

Data order is:
{wavelength, meanTensionPristine, meanTension50percent, meanTensionFracLong, meanTensionFracShort}

data = Import[pathRootProcessed <>
    "Expt14_Bragg_Tension_transmission.txt", "Table"];
TableForm[data, TableHeadings -> {None, 
    "wavelength", 
    "pristine", "50%", "frac-long", "frac-short")}];

x = data[[All, 1]];
yTensionPristine = data[[All, 2]];
yTension50percent = data[[All, 3]];
yTensionFracLong = data[[All, 4]];
yTensionFracShort = data[[All, 5]];
yMeanFractured =
    Table[Mean[yTensionFracLong[[i]], yTensionFracShort[[i]]],
    {i, Length[listWavelengths]}];
Vita

Adam J. Brooks was born in 1990 in Edison, New Jersey. He finished his undergraduate studies with a Bachelors of Science in Chemistry at Syracuse University in May 2012. His undergraduate research involved the coordination chemistry of alkaline earth tosylates under the guidance of Dr. Karin Ruhlandt-Senge. In August 2012, he came to Louisiana State University to pursue graduate studies in chemistry under Dr. Leslie Butler. His current research interests involve the use of grating-based interferometry for X-ray and neutron imaging of lithium ion batteries, additive manufacturing, and single crystals. He is currently a candidate for the degree of Doctor of Philosophy in Chemistry.