



5-2014

Methods for Quantitative Local Structure Analysis of Crystalline Materials Employing High Performance Computing

Tara Marie Michels-Clark

University of Tennessee - Knoxville, tmichels@utk.edu

Follow this and additional works at: https://trace.tennessee.edu/utk_graddiss

 Part of the [Materials Chemistry Commons](#), and the [Physical Chemistry Commons](#)

Recommended Citation

Michels-Clark, Tara Marie, "Methods for Quantitative Local Structure Analysis of Crystalline Materials Employing High Performance Computing. " PhD diss., University of Tennessee, 2014.
https://trace.tennessee.edu/utk_graddiss/2770

This Dissertation is brought to you for free and open access by the Graduate School at TRACE: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of TRACE: Tennessee Research and Creative Exchange. For more information, please contact trace@utk.edu.

To the Graduate Council:

I am submitting herewith a dissertation written by Tara Marie Michels-Clark entitled "Methods for Quantitative Local Structure Analysis of Crystalline Materials Employing High Performance Computing." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Chemistry.

Robert J. Hinde, Major Professor

We have read this dissertation and recommend its acceptance:

Cynthia B. Peterson, Zi-Ling Xue, Gregory D. Peterson

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Methods for Quantitative Local
Structure Analysis of Crystalline
Materials Employing High
Performance Computing

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Tara Marie Michels-Clark

May 2014

© by Tara Marie Michels-Clark, 2014
All Rights Reserved.

This dissertation is dedicated to my loyal, ever supportive and long suffering husband, Clayton Jeremiah Clark and to my parents, Larry and Anita Michels who have always provided steadfast love and support throughout my life and in all of my endeavors.

Acknowledgements

I must start with those at the Spallation Neutron Source at Oak Ridge National Laboratory with whom I have been privileged to work almost daily. Dr. Christina Hoffmann's detailed, fundamental knowledge of single crystal diffraction is unsurpassed. Her input has helped me to grow and improve tremendously as a scientist. Dr. Hoffmann's patient guidance, timely editing of my writing and regular encouragement have been invaluable. Ms. Vickie Lynch has graciously been available with her computing expertise, editing skills, and wise guidance. I will always be deeply grateful for her professional and personal example as a great software specialist and a true friend. I would also like to thank the Neutron Data Analysis and Visualization Division (NDAV) at the Spallation Neutron Source and the division director Dr. Thomas Proffen for hosting me at the Spallation Neutron Source and providing input in my work and career. A special thank you is extended to Dr. Andrei Savici of the NDAV division for his help in making "publication quality" figures in IDL, tips in LaTeX usage and reviewing my presentations and papers.

I would also like to acknowledge my international collaborators; Professor Hans-Beat Bürgi (University of Zürich and University of Bern), Professor Thomas Weber (ETH), Dr. Michal Chodkiewicz (University of Warsaw), Professor Anthony Linden (University of Zürich) and Professor Jürg Hauser (University of Bern). In particular, I am grateful to Professor Bürgi for his patience, scientific input and editing of my work. Professor Bürgi's scientific knowledge and experience is extraordinary and I am truly honored by the time he invested in sharing it with me. I am also grateful

to Dr. Chodkeiwicz for his quick to answers to questions and assistance in using ZODS (Zürich Oak Ridge Disorder Simulations). I would also like to thank Professor Weber for his suggestions in analyzing and developing computational tests and quick responses to all of my questions. Additionally, Professor Tibor Koritsanszky of Middle Tennessee State University has my profound gratitude for recommending me to this international project.

Professor Robert Harrison and my committee members Professors R. J. Hinde, Zi-ling Xue, Cynthia Peterson and Gregory Peterson have graciously made themselves available. In particular, I am grateful to Professor Harrison for taking the role as my advisor. Throughout this time he has provided vital support of my professional development allowing me to travel for my collaborative research and to various conferences. Also, regardless of the circumstances, Professor Harrison is always ready with much needed words of encouragement. I would also like to express my gratitude to Dr. Hinde for serving as my committee chair and teaching one of the most interesting courses I took during graduate school, Statistical Mechanics and Thermodynamics. Professor Gregory Peterson was so kind and generous to answer my call and come on board at the final stage of my work. He also provided me with foundational programming knowledge through his meticulous, detailed teaching of one the most practical classes I took with regard to my research, C++ for Scientists and Engineers.

This project was funded through an international grant from the National Swiss Science Foundation and is an interdisciplinary collaboration between those at the University of Zürich, University of Bern, and ETH in Switzerland and Oak Ridge National Laboratory, managed by UT-Battelle LLC under Contract No. DE-AC05-00OR22725 with the US Department of Energy. Computing resources for the computational methodology study using Tris(bicyclo[2.1.1]hexeno)benzene were provided by the Oak Ridge Institutional Cluster (OIC), Texas Advanced Computing Center (TACC) machine Ranger and Kracken. The β [beta]-NaLaF₄ calculations were done on the National Energy Research Scientific Computing Center machine Hopper

with computing time granted through a larger joint proposal entitled “Computational resources for the interpretation of neutron scattering data from the Spallation Neutron Source and HFIR”.

Finally, I must thank my husband Jeremy for his patience with me when the stress was high, for many hours of cooking, cleaning and holding down the fort at home when I could not do my part, for encouraging me when the going was tough, for making this achievement possible.

*Call to Me, and I will answer you. And I will show you great and wonderful things
which you do not know.*

Jeremiah 33:3

Abstract

A fundamental computational methodology was investigated to extract quantitative local structure information from single crystal diffuse scattering data. The principles of a highly efficient, parallelizable local structure analysis using massively parallel computing resources at Oak Ridge National Laboratory (ORNL) are demonstrated on an organic hydrocarbon compound containing stacking faults, Tris(bicyclo[2.1.1]hexeno)benzene. A probabilistic model of the stacking variations with a five layer interaction depth was developed. The final model structure motif statistics are verified using the steady state distribution of Markov matrix representing the four to five layer transitions. The computations revealed that highly parallelizable “structure-clones” could replace less computationally efficient “structure lots”. Further testing of the method is under way, using a new comprehensive modeling software suite ZODS (Zürich Oak Ridge Disorder Simulations) developed in Zürich, on synchrotron and lab X-Ray data of a highly efficient light-upconversion member of the NaLnF_4 [Sodium Lanthanide tetra fluoride] family. Initially, a synchrotron data set was collected at the high resolution Swiss-Norwegian Beam Line at the European Synchrotron Radiation Facility and is being analyzed. High resolution neutron diffraction data were recently collected at the time-of-flight Laue single crystal diffractometer TOPAZ at the Spallation Neutron Source at ORNL using the newly available event-mode processing. Currently, exploration of the event-mode data treatment and event based corrections for data preparation are under way. Simultaneous massively parallel local structure simulations of NaLaF_4 [Sodium

Lanthanum tetra fluoride] using ZODS on the National Energy Research Scientific Computing Center are in progress. A step-wise modeling approach was adopted. The largest contributors to the X-Ray diffuse scattering, La2 [Lanthanum 2] and Na2 [Sodium 2] column neighbor interactions were modeled first, followed by F1 [Fluorine 1] shift from its average position toward La [Lanthanum] and away from Na [Sodium]. This work provides a basis for streamlining diffuse scattering analysis and yields a quantitative interpretation of the local atomic arrangement of crystalline materials, which may provide valuable information for interpreting their structure property relationships.

Table of Contents

1	Introduction	1
2	Experimental techniques	5
2.1	Single crystal diffraction fundamentals	5
2.1.1	Interference	5
2.1.2	Bragg's Law	8
2.1.3	Reciprocal space	11
2.1.4	Electron density and structure factor	17
2.1.5	Scattering factors and scattering length	21
2.2	Diffraction experiment	22
2.2.1	X-Ray	22
2.2.2	Neutron	25
2.3	Diffuse data	29
3	Computational techniques	32
3.1	Quick overview of average structure refinement	32
3.2	Brief description of ZODS-Zürich Oak Ridge Disorder Simulations . .	33
3.2.1	Measured diffuse data preparation	34
3.2.2	MC crystal building	36
3.2.3	Model calculations and optimization	37
4	Neutron data of β[beta]-NaLaF₄	40

4.1	Introduction	40
4.2	Bragg scattering: neutron TOF data	42
4.3	TOF data correction	43
4.4	Experimental details	44
4.5	TOF data integration methods	46
	4.5.1 Spherical	46
	4.5.2 Elliptical	47
4.6	X-Ray data collection	47
5	New computational method for analysing diffuse scattering	49
5.1	Introduction	49
5.2	Chemical model system, growth modeling, and reference data	51
5.3	Computations	54
	5.3.1 Methods for optimizing model parameters	54
	5.3.2 Clones and parallelization	60
6	Local structure of analysis of β[beta]-NaLaF₄	64
6.1	Introduction	64
6.2	Super cell	65
6.3	Disentanglement	68
6.4	Model parameterization	70
6.5	Crystal building and model refinement	72
7	Results	75
7.1	β [beta]-NaLaF ₄ neutron data analysis results	75
	7.1.1 TOF data integration and refinement comparison	75
	7.1.2 Comparison to X-Ray refinement	78
7.2	New quantitative method verification	82
	7.2.1 Comparison of global optimization methods	83
	7.2.2 Influence of clones	87

7.2.3	Polynomial fitting to eight dimensional fitness surface	90
7.2.4	Sensitivity analysis	93
7.2.5	Principal component analysis	99
7.2.6	Examination of e_2 parameter	99
7.2.7	Motif statistics and comparison with reference model	101
7.3	β [beta]-NaLaF ₄ local structure modeling	104
7.3.1	Parameter convergence and model fitness	104
7.3.2	Direct space analysis	105
7.3.3	Intensity comparison	108
8	Conclusions and outlook	111
8.1	Conclusions	111
	Bibliography	114
	Appendix	125
A. 1	Two dimensional occupational disorder analysis	126
A. 1.1	General concepts	126
A. 1.2	C++ code	133
A. 2	Motif percentage calculation	151
A. 3	CIF files	157
A. 3.1	X-ray CIF	157
A. 3.2	Neutron CIF	163
A. 4	ZODS details	170
A. 4.1	xml input	170
A. 4.2	Plotting tools in ZODS	170
	Vita	175

List of Tables

5.1	The transition matrix \mathbf{T} of probabilities for extending the left-hand column of four-layer motifs into the top row of new four-layer motifs. Once a new motif is formed by adding a new fifth layer, the first layer and thus the first motif are dropped. The meaning of the symbols is described in the text.	53
5.2	Comparison of the reference parameters defining the reference data with mean model parameters and their standard deviations obtained by optimizations with a genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) using 40 individuals. The parameter values at the start of the optimizations are generated for each member in the population using the range of values listed in the third column for each parameter. The minimum and maximum values are the world size or absolute limits of the parameter values during optimization.	58
5.3	The control and run parameters for DE, GA, and PSO.	59
7.1	The fitness values and number of reflections for each integration radius are given for spherical and elliptical integration methods all refinements were done using GSAS. The d-spacing is 0.5.	76
7.2	The fitness values and number of reflections for each d-spacing are given for spherical and elliptical integration methods (all refinements were done in SHELX). The radius of integration was 0.15.	78

7.3	The atomic positions in fractional coordinates of the X-Ray and neutron data structure refinements. For the neutron data the radius of integration was 0.15 and d-spacing was 0.5. The standard uncertainties are given in parenthesis.	79
7.4	ADP values of NaLaF ₄ for spherical integration methods at integration radii 0.13–0.18.	80
7.5	ADP values of NaLaF ₄ for elliptical integration methods at integration radii 0.13–0.18.	81
7.6	ADP values of NaLaF ₄ 100K neutron and X-Ray <i>SHELX</i> refinement; standard uncertainties are given in parenthesis. The neutron data were integrated using spherical integration method and integration radii 0.15.	82
7.7	Summary of results after 150 generations of DE optimization with 40 gene sets that have different numbers of layers, lots per crystal, and numbers of clones. The number of cores used, wall clock time and the cpu hours are also reported for each calculation.	89
7.8	The decrease in the average model fitness (incorporating 20 clones) is given as a percent change from optimized population model fitness. A positive value percent change indicates an increase in <i>R</i> and thus a poorer model fitness when the given parameter is shifted.	94
7.9	Changes in structural motif composition based on changes in the independent model parameters defining the various stacking options; <i>c</i> , Δ and <i>e</i> ₂ are shown below. Column 2 corresponds to the structural statistics of the DE optimized parameters averaged over 20 clones. . .	97
7.10	Motif statistics of the reference crystals compared with those from the Markov steady-state distribution and those of the best individual in the 150 th (final) generation of a DE optimization averaged over 20 and 300 clones (reported as percent of the full crystal). The <i>b</i> without a subscript indicates that either <i>b</i> _{<i>L</i>} or <i>b</i> _{<i>R</i>} stacking is allowed.	103

7.11 The first, second and third conditional neighbor probabilities using the relative shift vectors (x, y, z) relative to the CU with no shift $(0, 0, 0)$. For example $P(CU, (0, 0, 0) | CU, (2, 1, 0))$ means that given a CU in its original position (x, y, z) the probability of the CU at $(x + 2, y + 1, z)$ being in the same or opposite orientation is given. 109

List of Figures

2.1	Examples of interference of waves. The left hand side depicts two waves with different phase shifts and/or amplitudes. The right hand side shows the resultant wave.	6
2.2	Figure (a) depicts diffraction of a row of point scatterers along the \mathbf{x} axis. The incident beam angle is α_0 and the diffracted beam is at angle α_n . The path difference between the incident and diffracted beams is given by: $(AB - CD)$ and a is the length of AB . In (b) the incident beam direction is \mathbf{s}_0 and diffracted beam direction is \mathbf{s} . The path difference between the diffracted beams is shown as the difference between the projection of \mathbf{a} (vector between A and B from figure 2.2) onto \mathbf{s}_0 and \mathbf{a} onto \mathbf{s} , i.e.; $\mathbf{a} \cdot (\mathbf{s} - \mathbf{s}_0)$. Figure taken from [1].	8
2.3	A geometric depiction of Bragg's law; the incident radiation (green) and reflected (red) from adjacent parallel hkl planes with spacing d_{hkl} and a shift of an integer multiple (n) of $\lambda/2$ for constructive interference.	10

2.4	A representation of the Ewald sphere. When the reciprocal lattice point, given by $\mathbf{P}^*(hkl)$, is in contact with the sphere constructive interference occurs. The diffracted beam passes through the origin of the reciprocal lattice, \mathbf{O}^* and coincides with the crystal position and the origin of the diffracted beam. By changing the orientation of the reciprocal lattice all diffraction through reciprocal lattice points can be measured. The radius of the sphere is given by $1/\lambda$ and the phase shift is $(\mathbf{s} - \mathbf{s}_0)/\lambda$. Figure from [2].	12
2.5	In (a) a direct unit cell is shown with blue vertices and in (b) the reciprocal unit cell with a common origin is shown with green vertices. Figure from [3].	13
2.6	In (a) a 2-D direct lattice is shown with point scatterers at the origin and in (b) the corresponding diffraction pattern. Figure from [3]. . . .	14
2.7	Incident radiation in direction \mathbf{s}_0 from the source, represented by large circle, is shown interacting with two point scatterers (electron or nucleus) and is represented by the black filled circles, one positioned at the origin and the other at \mathbf{r} . The secondary diffracted waves travel R distance to the detector (large square) at position \mathbf{s}	17
3.1	An example of a 2-D calculation grid for the calculation of intensity points in reciprocal space, depicting the variation of $\mathbf{r}_{i,j}$ where index \mathbf{i} is in the \mathbf{d}_1 direction, index \mathbf{j} is in direction \mathbf{d}_2 and the origin is at $\mathbf{r}_{0,0}$. The step size is the spacing between consecutive $\mathbf{r}_{i,j}$ points. . . .	35
4.1	The $HK4.5L$ layer of NaLaF_4 collected on single crystal TOF diffractometer TOPAZ reveals the same diffuse “honeycomb” pattern observed in X-Ray experiments [4]. This is a view generated using MANTID with data corrections applied.	45

4.2	Depiction of the ellipsoidal region generated in the peak integration algorithm around each found Bragg peak. The event region is indicated by the dashed line surrounding ellipse. Figure from MANTID [5] project website.	48
5.1	Left: Tris(bicyclo[2.1.1]hexeno)benzene molecule (TBHB), black atoms are carbon, white atoms are hydrogen. Center: A skeletal formula representation of TBHB where all atoms on a given dotted circle were assigned the same isotropic atomic displacement parameter (U_1 , U_2 , U_3). Right: schematic tri-star representation of TBHB (used in Fig. 5.2).	52
5.2	Examples of unique five-layer stackings and associated motifs: layers 1,2,3 in solid black, layer 4 (green) and layer 5 (red); respective transition probabilities from four- to five-layer stackings at the bottom of the motif. The symbols above the motif describe the four- and five-layer stacks. The symbols b_L , b_R , e (bent left, bent right, eclipsed) characterize the three possible three layer stacks. A four-layer stack is defined by two symbols, and a five layer stack by three symbols. . .	55
5.3	Reference intensity and calculated intensity (averaged over 40 clones) for the 01L line are shown overlaid. The difference (calculated - reference) is shown in the plot below. (see Table 7.7, $\bar{R} = 0.0077$) . .	56

6.1	The super cell of β -NaLaF ₄ illustrates occupational structural disorder by showing the neighbor preferences of the Na2 ... La2 and the Na1 ... void columns. The central green La column is fully ordered and the three disordered Na(yellow)/La(red) columns surrounding are not all oriented the same way (negative correlation). The Na/La columns prefer La/Na columns as neighbors. The Na (orange)/void (open blue dashed circles) are also not all oriented the same way. These neighbor preferences are impossible to satisfy over the whole 2-D range and lead to the diffuse “honeycomb” diffraction pattern (shown behind the direct space structure) at $(H, K, 0.5L)$. Figure from ACS COMP image award [6]. (b) Average structure of β -NaLnF ₄ (Ln=Y, La Lu) showing the three different columns of metal sites along c : Na2/La2 is a 1:1 mixture of Na2 and Ln2 (pink), the Ln1 column (blue), both with C_{3h} symmetric, tricapped trigonal prismatic coordination geometry, and the half occupied Na1 site (red) with C_3 symmetric, distorted octahedral coordination geometry. Local distortions owing to the disorder in the crystal are indicated by arrows. Figure taken from [4].	66
6.2	CU1 shown (left) with the alternative (right). Either CU1 or its alternative is repeated along c to produce the F1 coordinated to Na2 and La2 in the 2-D crystal. The question marks indicate the direction of the F1 shift which will be tested in the optimization.	70
6.3	Two dimensional illustration of the symmetric CU first neighbor interactions defined in the model. The $-z$ elements are at $z = -1/2$ below the plane shown.	71
6.4	Plot of the crystal energy as a function of MC cycles. After 100 cycles it has equilibrated to about -5650 in crystal energy.	72

7.1	Linear regression of 100K X-Ray ADPs plotted against the 100K neutron ADPs a general linear correlation is observed.	83
7.2	Evolution of population fitness \bar{R} for three global optimization algorithms: general genetic algorithm (GA) and differential evolution (DE) and particle swarm (PSO) as a function of generation number.	84
7.3	Behavior of the model parameters c , Δ , e_2 , $tilt$, U_1 , U_2 , U_3 (plots a-g), and fitness \bar{R} (plot h) during DE optimization with 40 individuals and 20 clones each. The population mean is indicated by the middle points, and the population standard deviation by the vertical lines. The dots above and below the lines represent the largest and the smallest parameter values in the population.	85
7.4	Decrease of \bar{R} with increasing number of lots. \bar{R} depends linearly on the reciprocal square root of the number of lots ($\#lots$).	88
7.5	Contour plot depicting correlation of e_2 and $tilt$ and fitness (given as χ [chi]) in the third dimension. The parameters remain uncorrelated until close to the global minimum.	101
7.6	Contour plot depicting correlation of e_2 and Δ and fitness (given as χ [chi]) in the third dimension. The parameters remain uncorrelated until close to the global minimum.	102
7.7	The convergence of the fitness function R as a function of generation number.	106
7.8	Behavior of the model parameters, p1, which defines the Na2/La2 CU interaction with another Na2/La2 CU in the neighboring unit cell and the F1 shift (plots a and b)during DE optimization with 24 individuals and 20 clones each. The population mean is indicated by the middle points, and the population standard deviation by the vertical lines. The dots above and below the lines represent the largest and the smallest parameter values in the population.	106

7.9	A 2-D crystal composed of CUs illustrating the preference for oppositely oriented CUs. The Na2...La2 column prefers La2...Na2 columns as neighbors. The top view of the crystal shows Na2 (orange) and La2 (green) atoms, each with 6 neighbors.	107
7.10	2-D 120 degree lattice showing first neighbors (red), second (blue), third (orange) and fourth (green) with their corresponding coordinates relative to the central atom (black) at the origin (0,0).	108
7.11	From left to right, measured, calculated and difference intensities for the $(H, K, 0.5L)$ layer.	110
A.1	2-D sixty degree lattice showing first neighbors (red), second (blue), third (orange) and fourth (green) with their corresponding coordinates relative to the central atom (black) at the origin (0,0).	127

Chapter 1

Introduction

A crystalline material is defined as a 3-D periodic, approximately infinite, ordered array of point scatterers. Bragg diffraction, from crystals, is a consequence of the interference of diffracted radiation waves from the point scatterers (atoms) [7]. The diffracted Bragg intensities of the scattering object (crystal) are used to refine the time and space averaged atomic positions. The resulting crystal structure defines the long range order or governing (Bragg) structure. Advances in X-Ray diffraction experimental techniques and instrumentation and a corresponding increase in computing power and software development over the past few decades of crystallography have given rise to a high level of automation in structure determination for well-ordered structures [8]. However, many technologically interesting materials owe their functionality to local inconsistencies in the crystal structure [4, 9]. Their crystal structures are no longer represented by a regular periodic function but are more delocalized resulting in a diffuse diffraction pattern [10, 11].

Evidence of local disturbances are revealed in direct space as un-chemical features such as partial atoms (occupancy < 1) and elongated anisotropic Atomic Displacement Parameters (ADPs) (magnitude in one or more U_{ij} direction significantly larger than the others). ADPs are an ellipsoidal description of temperature dependent atomic motion on the lattice site. Partial occupation means that a given site is

occupied by one atom type in some unit cells, and other types of atoms or possibly vacant in other unit cells. Highly anisotropic ADPs usually indicate that a given atom occupies a slightly shifted position for each crystallographic site. The distance between these positions is lost in the average structure as the local atomic shifts are smaller than the resolution limit of the Bragg data. To extract the important structure details analysis of the diffuse part of the diffraction pattern becomes necessary. However, the analysis and interpretation of diffuse scattering is far from routine, requiring careful experimental set-up, appropriate data correction and reduction and detailed, structure specific models.

Local structure variations are propagated in real space and the resulting diffuse intensities are often orders of magnitude weaker than Bragg intensities and distributed through extensive volumes of reciprocal space. Protocols to extract diffuse data for local structure modeling and the availability of software that interprets these data is largely lacking [8]. Up to now, the data for only a limited number of such compounds has been qualitatively interpreted. The few that have been interpreted utilized elaborate, often ad hoc structural models, which are tested and optimized with the help of Monte-Carlo simulations, genetic algorithms and numerical least-squares calculations [12, 13].

Qualitative descriptions of the local structure and simple modeling techniques can sometimes yield important structural information [14, 15, 16]. However, this work will show an example of the use of large-scale computational modeling employing new approaches to extract a quantitative description of the local structural and capture chemically interesting details (chapter 5 section 5.1). To quantify the intricate interplay between the local structure and the overall governing structure, new computational methods are needed. Extracting a quantitative description of the local structure from the diffuse part of the diffraction pattern is complex and computationally intensive for two primary reasons; (1) only the local structural disturbances contribute to the diffuse scattering in the diffraction pattern and are often distributed over extensive volumes of reciprocal space making data preparation

for computational modeling challenging and (2) the model relies on building a representative set of virtual crystals to simulate the array of local ordering options and global optimization of the structure specific model parameters. Both of these challenges are addressed in this work.

The dissertation is organized as follows. Background details describing single crystal diffraction and current computational methodologies are contained in chapters 2 and 3 respectively. A full reciprocal scattering volume including Bragg and diffuse neutron data of β -NaLaF₄ from a series of rare earth doped β -NaLnF₄ (Ln= Y-Lu) of highly efficient light up-conversion materials, commonly used in LED display devices and as labels for immunoassays [17] was collected for the first time at 100K. Beam time was awarded on the time of flight Laue single crystal diffractometer TOPAZ at the Spallation Neutron Source. The β phase of NaLaF₄ has a hexagonal average structure [4]. Due to local Na₂/La₂ column neighbor interactions, 2-D “honey comb” diffuse planes between Bragg layers were observed. It was observed by [4] through single crystal absorption spectroscopy that the local site symmetry of one of the optically active sites responsible for the upconversion properties of NaGdF₄ is of lower symmetry than that of single crystal average structure. This means the local ordering options need to be examined [4]. Diffuse X-Ray data has been collected and processed for computational modeling however, the availability of complementary neutron data will permit better modeling as the more similar scattering lengths of F, Na and La allow a better determination of the shifted F1 position. A path for neutron data corrections (Lorentz, absorption, spectrum and background) was explored using MANTID (Manipulation and Analysis Toolkit for Instrument Data) [5] and compared to conventional post integration correction. This is described in chapter 4.

A new computational method providing a quantitative description of the local structure of crystalline materials is described in this work. Tris(bicyclo[2.1.1]hexeno)benzene that crystallizes in three fold, symmetric, coplanar layers, each with multiple stacking options [18] was selected as a test model. The stacking options were represented using a probabilistic model with up to five layer interaction depth. The layer stacking

probabilities along with anisotropic displacement parameters (ADPs) and a layer tilt comprised the local structure model. Three global optimization algorithms were tested in optimizing the model parameters. These optimization algorithms iteratively improved the parameter values until the best fit to the experimental diffuse intensities was achieved. Clones (copies of parameter sets) were found to reduce the statistical modeling noise, serve as an effective mechanism for parallelization and can replace lots (chapter 5). A comparison of the percentages of the possible structural motif types, calculated from a probabilistic layer growth model, showed a close match to the reference data. A C++ code was written to calculate the motif statistics from the final generation of optimized crystals, including their clones (Appendix A. 2). The results were subsequently verified theoretically using the steady-state distribution of the four- to five-layer Markov transition matrix (chapter 7 section 5.1). The parallelization mechanism using clones was incorporated into a new comprehensive modeling tool called ZODS (Zürich Oak Ridge Disorder Simulations). Results and verification are discussed in chapter 7.

Finally, the local structure of NaLaF₄ was modeled using ZODS (Zürich Oak Ridge Disorder Simulations) against lab X-Ray diffuse intensity data using a step-wise approach. The biggest contributors to the diffuse intensity were identified first then more aspects of the disorder were incorporated (chapter 6). Additional details along with ZODS program input and a C++ and python codes used in this work are contained in the Appendix. A code written in C++ (Appendix A. 1) modeling local occupational disorder on a 2-D hexagonal lattice provides a glimpse into the modeling complexity that a seemingly limited example of deviation from average structure requires.

Chapter 2

Experimental techniques

2.1 Single crystal diffraction fundamentals

The fundamentals of single crystal diffraction experiments and experimental setup are briefly detailed in the following sections. Further information can be found in fundamental textbooks on X-Ray and neutron scattering such as [1, 3, 19, 20, 21].

2.1.1 Interference

The interaction of an incident wave with an array of point scatterers generates a spherical wave at each point scatterer. The sum of these diffracted waves produces a diffraction pattern through constructive and destructive interference. The path difference (phase shift) between point scatterers determines if constructive or destructive interference occurs (figure 2.1).

A path difference of odd integer multiple of $\lambda/2$ relative to the first diffracted wave, corresponds to a phase shift of π and results in waves with equal amplitude and opposite phase. The resultant wave is the sum of the diffracted waves from each point scatterer, therefore the waves will cancel each other out resulting in a wave of zero amplitude (destructive interference). A path difference of an even multiple of λ results in a phase shift of 2π . The two waves are in phase with one another and

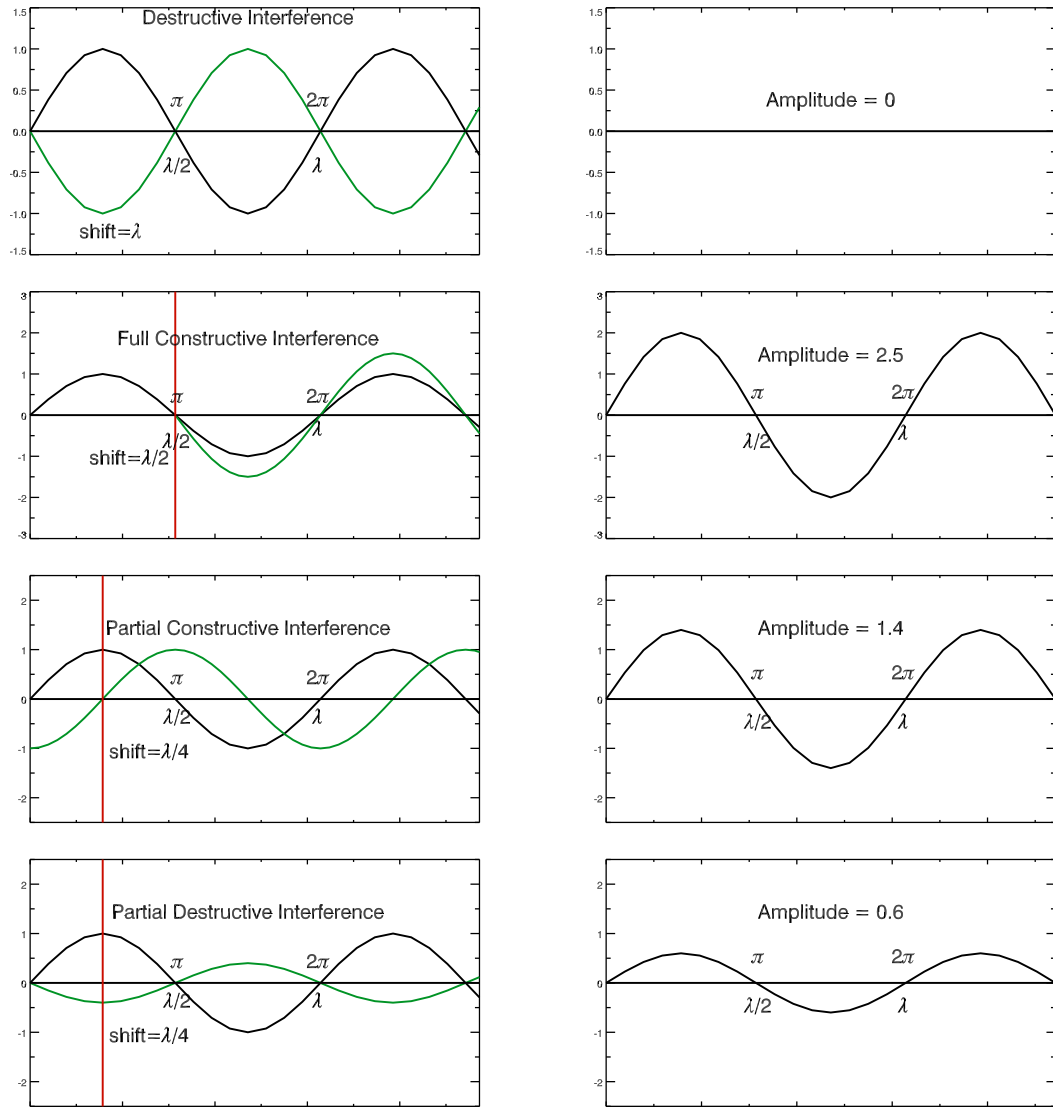


Figure 2.1: Examples of interference of waves. The left hand side depicts two waves with different phase shifts and/or amplitudes. The right hand side shows the resultant wave.

the resultant wave has double the amplitude (sum of both wave's amplitude). This is constructive interference. The magnitude of the resultant wave can be anywhere between these two extremes as illustrated in the bottom two plots of figure 2.1.

Consider each point scatterer as a single atom (scattering center) situated at a lattice point. The 3-D crystal can then be thought of as consisting of rows of atoms with spacing a, b and c along \mathbf{x} , \mathbf{y} and \mathbf{z} respectively. The condition required for constructive interference of a single row of scatterers as shown in figure 2.2(a) along the \mathbf{x} axis is found by examining the path differences between the scattered waves of adjacent atoms (with spacing a). The path difference must be a whole number (integer) multiple of the wavelength, λ , therefore

$$(AB - CD) = a(\cos \alpha_n - \cos \alpha_0) = n_x \lambda \quad (2.1)$$

where n_x is the diffraction order (number of integer multiples) of λ and α_n and α_0 correspond to the incident and diffracted beam angles respectively; relative to the \mathbf{x} axis.

Equation 2.1 can be expressed in vector notation. If \mathbf{s}_0 and \mathbf{s} represent the direction of the incident and diffracted beams respectively and \mathbf{a} is the vector between atoms A and B (each at lattice points) along \mathbf{x} (Figure 2.2(b)) then the path difference $a(\cos \alpha_n - \cos \alpha_0)$ can be expressed as

$$\mathbf{a} \cdot \mathbf{s} - \mathbf{a} \cdot \mathbf{s}_0 = \mathbf{a} \cdot (\mathbf{s} - \mathbf{s}_0) = n_x \lambda. \quad (2.2)$$

All diffracted beams with the same path difference occur at the same angle α_n to the atom row.

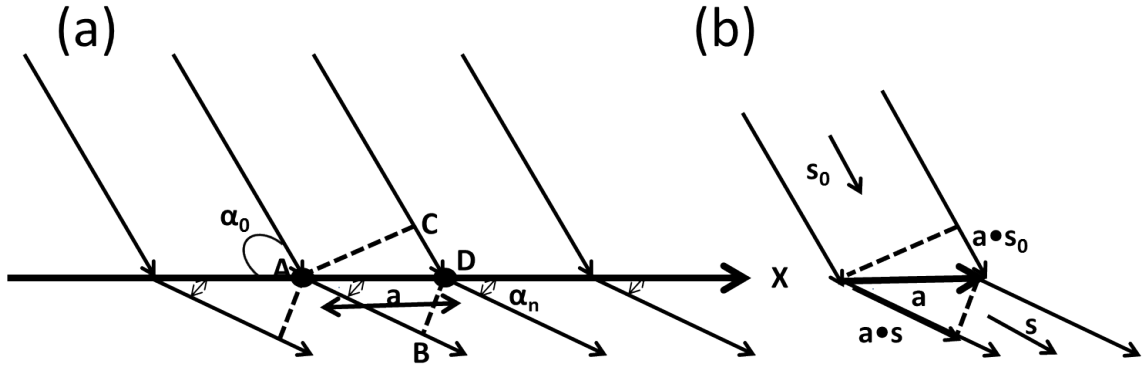


Figure 2.2: Figure (a) depicts diffraction of a row of point scatterers along the x axis. The incident beam angle is α_0 and the diffracted beam is at angle α_n . The path difference between the incident and diffracted beams is given by: $(AB - CD)$ and a is the length of AB . In (b) the incident beam direction is \mathbf{s}_0 and diffracted beam direction is \mathbf{s} . The path difference between the diffracted beams is shown as the difference between the projection of \mathbf{a} (vector between A and B from figure 2.2) onto \mathbf{s}_0 and \mathbf{a} onto \mathbf{s} , i.e.; $\mathbf{a} \cdot (\mathbf{s} - \mathbf{s}_0)$. Figure taken from [1].

Analogously, this same analysis when performed for atoms along y and z at spacing b and c respectively, produces the resulting Laue equations below;

$$b(\cos \beta_n - \cos \beta_0) = \mathbf{b} \cdot (\mathbf{s} - \mathbf{s}_0) = n_y \lambda \quad (2.3)$$

$$c(\cos \gamma_n - \cos \gamma_0) = \mathbf{c} \cdot (\mathbf{s} - \mathbf{s}_0) = n_z \lambda. \quad (2.4)$$

The values of n_x , n_y and n_z correspond to indexes h , k and l described in section 2.1.3. The conditions for the incident beam to produce constructive interference from a row of equally spaced point scatterers is known as Laue diffraction [22]. Laue formulated the diffraction conditions based on the reciprocal lattice (section 2.1.3).

2.1.2 Bragg's Law

The scatterers described in section 2.1 were oriented in a straight line and required six angles; α_n , β_n , γ_n , α_0 , β_0 and γ_0 ; three lattices spacings; a , b and c and three integers; n_x , n_y and n_z to calculate the directions of the diffracted beams. Now

considering the displacement, d_{hkl} , of two point scatterers between planes instead of rows, maximum constructive interference can be achieved at every integer $n\lambda$ wavelength. The distance of the crystal to the detector (cm or m) is orders of magnitude greater than the distance between atoms making up the crystal (nm or Å) that the approximation of parallel paths for each resultant wave can be applied (figure 2.3). The angle between the incident and diffracted beams is 2θ . From figure 2.3 in the triangle the length of side AB is equal to

$$AB = d_{hkl} \sin \theta \quad (2.5)$$

The total path length is $(AB + CD)$ and since $AB = CD$;

$$(AB + CD) = 2AB = n\lambda \quad (2.6)$$

then substituting equation 2.5 into the path length expression, Bragg's law is obtained,

$$2d_{hkl} \sin \theta = n\lambda. \quad (2.7)$$

Assume the distance between lattice planes, d_{hkl} is the magnitude of the vector that connects two point scatterers according to the figure 2.3. The scatterers are in phase if the angle between their incident and scattered waves is $\pi/2 - \theta$. This type of diffraction is also called Bragg diffraction and is named for the father and son team, William Henry Bragg and William Laurence Bragg who developed the diffraction conditions based on lattice planes in direct space [7]. They found that crystals, at certain specific wavelengths and incident angles, produced intense peaks of reflected radiation called Bragg peaks. The concept of Bragg diffraction applies equally to neutron diffraction and electron diffraction processes.

Bragg's Law

$$2d_{hkl}\sin\theta = n\lambda$$

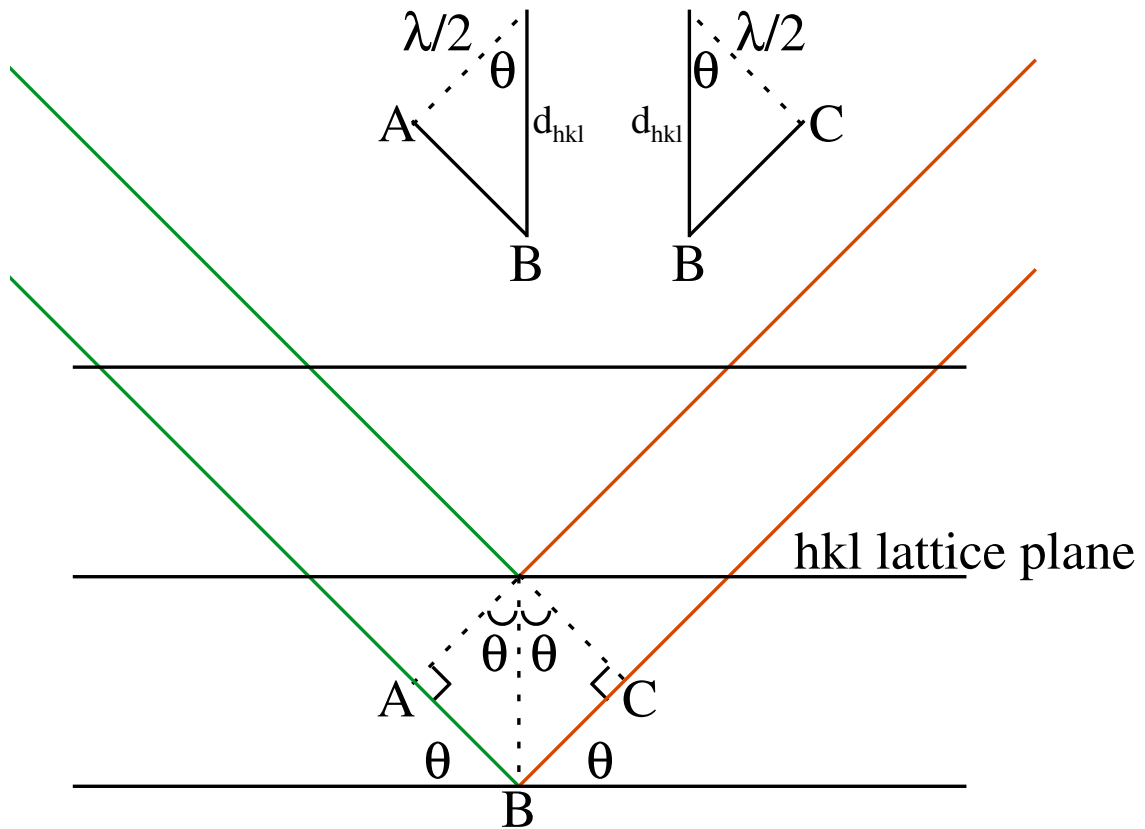


Figure 2.3: A geometric depiction of Bragg's law; the incident radiation (green) and reflected (red) from adjacent parallel hkl planes with spacing d_{hkl} and a shift of an integer multiple (n) of $\lambda/2$ for constructive interference.

2.1.3 Reciprocal space

There is a reciprocal relationship between the location of the diffraction maxima in the diffraction pattern and the location of the point scatterers in the crystal lattice. The occurrence of the intensity maxima is determined by the diffraction angle. The maxima occur every $2 \sin \theta / \lambda = n / d_{hkl}$, where n is an integer. Thus, the maxima are an integral multiple of the reciprocal of the d_{hkl} distance between scattering planes. The distance between scattering planes is given as d_{hkl} in direct space and $1/d_{hkl}$ in reciprocal space. Given a spacing of $1/d_{hkl}$, a maxima is observed for every n/d_{hkl} where $n = 1, 2, 3, \dots$. The lattice that corresponds to $1/d_{hkl}$ spacing is called the reciprocal lattice and is defined in terms of diffraction pattern (reciprocal space).

In order to understand the relationship between real and reciprocal space the construct developed by Paul Peter Ewald called the Ewald sphere [23] in three dimensions is used. This sphere is constructed around a single point scatterer serving as the origin of both the real and reciprocal lattice, with the incident beam through the origin and a radius of $1/\lambda$. The conditions of diffraction are met when the sphere intersects the reciprocal lattice at one of its lattice points (figure 2.4).

Suppose there are two lines of point scatterers, one with spacing a and the other spacing b , perpendicular to the incident beam but at angle γ with respect to each other. As the plane of scatterers is rotated through the Ewald construction diffraction maxima occur that are perpendicular to the line spaced a distance of K/b (K/a) in reciprocal space (figure 2.6). If these two lines are combined into a single two dimensional lattice array, the resulting diffraction pattern has maxima when both lines of scatterers produce waves that constructively interfere. This occurs at the intersection of the diffraction points produced by each line of scatterers and these intersections of mutual constructive interference are called diffraction nodes. The resulting two dimensional lattice of diffraction maxima defines the reciprocal lattice in two dimensions.

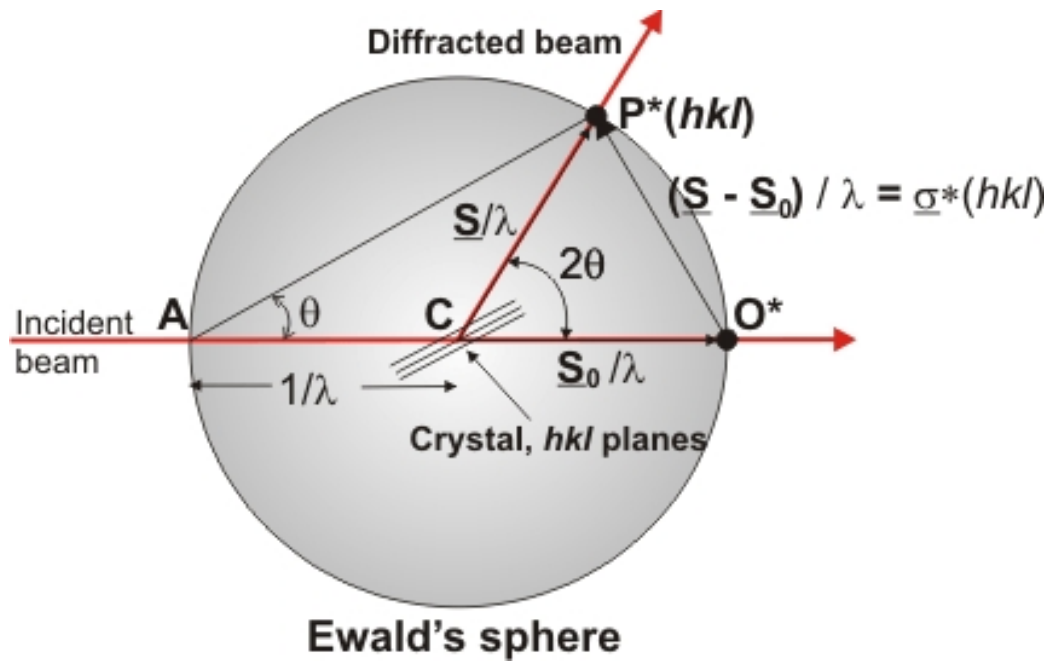


Figure 2.4: A representation of the Ewald sphere. When the reciprocal lattice point, given by $\mathbf{P}^*(hkl)$, is in contact with the sphere constructive interference occurs. The diffracted beam passes through the origin of the reciprocal lattice, \mathbf{O}^* and coincides with the crystal position and the origin of the diffracted beam. By changing the orientation of the reciprocal lattice all diffraction through reciprocal lattice points can be measured. The radius of the sphere is given by $1/\lambda$ and the phase shift is $(\mathbf{s} - \mathbf{s}_0)/\lambda$. Figure from [2].

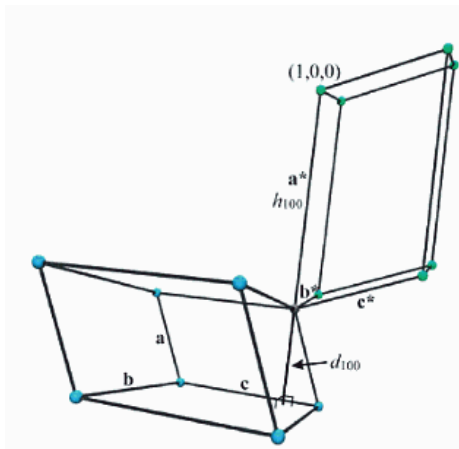


Figure 2.5: In (a) a direct unit cell is shown with blue vertices and in (b) the reciprocal unit cell with a common origin is shown with green vertices. Figure from [3].

The unit cell defined as the smallest unit containing the overall crystal symmetry that can be translated to build the entire crystal lattice. The unit cell in direct space is defined by vectors \mathbf{a} , \mathbf{b} and \mathbf{c} (for two dimensions only \mathbf{a} and \mathbf{b}). In reciprocal space the unit cell is defined by vectors \mathbf{a}^* , \mathbf{b}^* and \mathbf{c}^* (figure 2.5).

The reciprocal unit cell is readily identified in the diffraction pattern and has reciprocal axial lengths a^* and b^* scaled by K and the angle between the reciprocal axis, \mathbf{a}^* and \mathbf{b}^* is γ^* . The $(1, 0)$ in the direct lattice containing the \mathbf{b} axis produces a set of perpendicular lines $(0^*, 1^*)$ containing the \mathbf{a}^* axis and similarly the $(0, 1)$ containing the \mathbf{a} axis produces a set of perpendicular lines $(\bar{1}^*, 0^*)$ containing the \mathbf{b}^* axis. The \mathbf{a}^* axis is perpendicular to the \mathbf{b} axis (and \mathbf{c} axis when extended to three dimensions) in the direct lattice and the \mathbf{b}^* axis is perpendicular to the \mathbf{a} axis in the direct lattice. In general a set of lines, (h, k) in the two dimensional direct lattice will produce a set of perpendicular lines (\bar{k}^*, h^*) in the reciprocal lattice. Upon scaling $K = 1$, the (h, k) parallel lattice lines are separated by a reciprocal distance of $1/d_{hkl}$ with d_{hkl} defining the distance between direct lattice lines.

Extending the reciprocal lattice relationship to the direct lattice in three dimensions is done by adding equidistant, parallel ab planes along the c axis. The volume of the direct cell is the area of the base, the bc plane times the height,

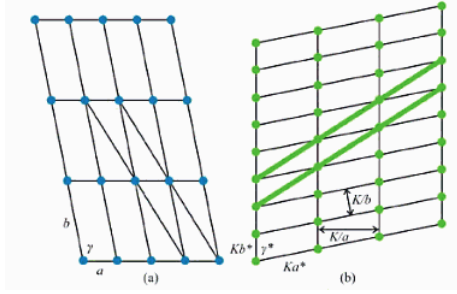


Figure 2.6: In (a) a 2-D direct lattice is shown with point scatterers at the origin and in (b) the corresponding diffraction pattern. Figure from [3].

d_{100} (distance between bc planes). The volume of the unit cell is then given by; $V = A_{bc}d_{100} = |\mathbf{b} \times \mathbf{c}|d_{100}$. The \mathbf{a}^* axis is the \mathbf{h}_{100} vector in the reciprocal lattice and has length $|\mathbf{h}_{100}| = 1/d_{100}$. Solving for the relationship between \mathbf{a}^* and the axial vectors of the direct unit cell:

$$\begin{aligned} V &= A_{bc}d_{100} = |\mathbf{b} \times \mathbf{c}|d_{100} \\ d_{100} &= \frac{V}{|\mathbf{b} \times \mathbf{c}|} = \frac{\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}}{|\mathbf{b} \times \mathbf{c}|} \\ h_{100} &= a^* = \frac{|\mathbf{b} \times \mathbf{c}|}{\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}}. \end{aligned}$$

Since \mathbf{a}^* is perpendicular to the bc plane, it is parallel to $\mathbf{b} \times \mathbf{c}$ and has magnitude h_{100} . Therefore;

$$\mathbf{a}^* = \frac{\mathbf{b} \times \mathbf{c}}{\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}} = \frac{\mathbf{b} \times \mathbf{c}}{V}. \quad (2.8)$$

It then follows that \mathbf{b}^* and \mathbf{c}^* are given by;

$$\mathbf{b}^* = \frac{\mathbf{c} \times \mathbf{a}}{\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}} = \frac{\mathbf{c} \times \mathbf{a}}{V}. \quad (2.9)$$

$$\mathbf{c}^* = \frac{\mathbf{a} \times \mathbf{b}}{\mathbf{a} \cdot \mathbf{b} \times \mathbf{c}} = \frac{\mathbf{a} \times \mathbf{b}}{V}. \quad (2.10)$$

The direct cell in terms of reciprocal lattice vectors is given below;

$$V^* = \mathbf{a}^* \cdot \mathbf{b}^* \times \mathbf{c}^* \quad (2.11)$$

$$= a^* b^* c^* (1 - \cos^2 \alpha^* - \cos^2 \beta^* - \cos^2 \gamma^* + 2 \cos \alpha^* \cos \beta^* \cos \gamma^*)^{\frac{1}{2}}. \quad (2.12)$$

The spacing between the a^*b^* planes is $1/a$ therefore:

$$V^* = A_{b^*c^*}^* \frac{1}{a} = |\mathbf{b}^* \times \mathbf{c}^*| \frac{1}{a}$$

$$a = \frac{|\mathbf{b}^* \times \mathbf{c}^*|}{\mathbf{a}^* \cdot \mathbf{b}^* \times \mathbf{c}^*}.$$

Since \mathbf{a} is perpendicular to the a^*b^* plane and parallel to $\mathbf{b}^* \times \mathbf{c}^*$, the resulting direct lattice vector relations to reciprocal lattice vectors are given by;

$$\mathbf{a} = \frac{\mathbf{b}^* \times \mathbf{c}^*}{\mathbf{a}^* \cdot \mathbf{b}^* \times \mathbf{c}^*} = \frac{\mathbf{b}^* \times \mathbf{c}^*}{V^*}, \quad (2.13)$$

$$\mathbf{b} = \frac{\mathbf{c}^* \times \mathbf{a}^*}{\mathbf{a}^* \cdot \mathbf{b}^* \times \mathbf{c}^*} = \frac{\mathbf{c}^* \times \mathbf{a}^*}{V^*} \quad (2.14)$$

and

$$\mathbf{c} = \frac{\mathbf{a}^* \times \mathbf{b}^*}{\mathbf{a}^* \cdot \mathbf{b}^* \times \mathbf{c}^*} = \frac{\mathbf{a}^* \times \mathbf{b}^*}{V^*}. \quad (2.15)$$

From these relations it follows that;

$$\mathbf{a} \cdot \mathbf{a}^* = 1 \quad (2.16)$$

$$\mathbf{b} \cdot \mathbf{b}^* = 1 \quad (2.17)$$

$$\mathbf{c} \cdot \mathbf{c}^* = 1 \quad (2.18)$$

$$\mathbf{a} \cdot \mathbf{b}^* = 0 \quad (2.19)$$

$$\mathbf{a} \cdot \mathbf{c}^* = 0 \quad (2.20)$$

$$\mathbf{b} \cdot \mathbf{c}^* = 0. \quad (2.21)$$

In the two dimensional lattice, reciprocal lattice vectors to coordinates (h, k) were perpendicular to the lattice lines in the direct lattice. The length of the reciprocal lattice vectors with indexes (h, k) was the reciprocal length of the distance between (h, k) lines. A general vector in a three dimensional reciprocal lattice with fractional coordinates x_f^*, y_f^*, z_f^* is given as:

$$\mathbf{v}^* = x_f^* \mathbf{a}^* + y_f^* \mathbf{b}^* + z_f^* \mathbf{c}^* \quad (2.22)$$

A vector to a lattice node (h, k, l) is given by;

$$\mathbf{h}_{hkl} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^* \quad (2.23)$$

and it holds for the three dimensional case that $h_{hkl} = \frac{1}{d_{hkl}}$ and $\mathbf{h}_{hkl} = \mathbf{d}_{hkl}^*$. The reciprocal lattice vector from a common origin to a reciprocal lattice point (h, k, l) is perpendicular to the hkl planes and has a length (magnitude) equal to the reciprocal of plane spacing (as in figure 2.3).

In a real crystal there are many planes of scatterers that interact with one another, incident radiation will result in the interference of scatterers that are not in the same plane affecting the resulting diffraction maxima. The path difference between two sets of point scatterers will affect the phase shift of the diffracted waves and this effect will be different at each diffraction angle and therefore will not alter each diffraction maxima in the same manner. The relative locations of two sets of point scatterers determine the phase differences and resulting intensities of the waves for each location in the diffraction pattern. Therefore, using the measured intensities which are compared to the calculated phase- shifts from the scatterers placed in the crystal lattice and iteratively refining the calculated to the measured is one path to determine the relative location of the scatterers within the crystal lattice [24].

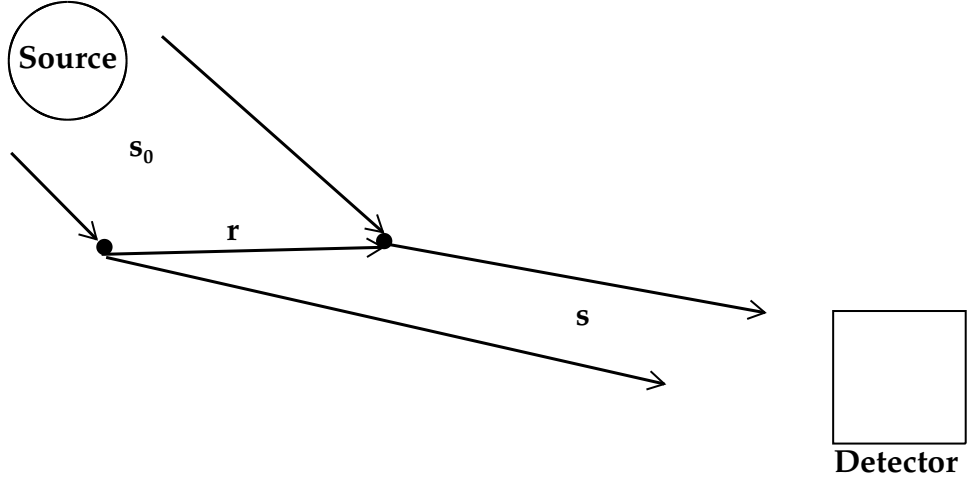


Figure 2.7: Incident radiation in direction \mathbf{s}_0 from the source, represented by large circle, is shown interacting with two point scatterers (electron or nucleus) and is represented by the black filled circles, one positioned at the origin and the other at \mathbf{r} . The secondary diffracted waves travel R distance to the detector (large square) at position \mathbf{s} .

2.1.4 Electron density and structure factor

The two scatterers of figure 2.7, one at the origin and the other at position r and vector \mathbf{r} between each emit a scattered wave upon interaction with radiation. The phase difference between the first and second scatterers is the projection of \mathbf{r} onto \mathbf{s}_0 given by; $\mathbf{s}_0 \cdot \mathbf{r}/\lambda$. The detector is at distance R and position \mathbf{s} and is 2θ relative to \mathbf{s}_0 . The diffracted wave of the first point scatterer is $\mathbf{s} \cdot \mathbf{r}/\lambda$ relative to the second point scatterer. The total phase difference for the second point scatterer is

$$\frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}}{\lambda}. \quad (2.24)$$

The vector representation of the diffracted waves allows a convenient way to keep vector components separate by representing them as a complex number. Thus, the superposition of the two waves resulting from the two point scatterers is given by

$$\varepsilon_r = [1 + \exp 2\pi i (\frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}}{\lambda})]. \quad (2.25)$$

Equation 2.25 gives the phase and amplitude of the superposition wave for the detector position \mathbf{s} . At different detector positions \mathbf{s} the phase and amplitude will change. If we consider n scatterers at different \mathbf{r}_i positions then the superposition wave becomes

$$\varepsilon_r = \sum_{i=1}^n \exp[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}_i}{\lambda}]. \quad (2.26)$$

In the case of X-Ray scattering the distribution of electron density $\rho(\mathbf{r})$ is calculated from the measured intensities. A similar concept applies for neutron scattering experiments except the density measured is not based on the electron distribution but the nuclei distribution.

The electron density of a molecule is approximated in structure analysis as the sum of atomic densities located at the atomic positions \mathbf{r}_i and given by

$$\rho(\mathbf{r}) = \sum_{i=1}^q \rho_i(\mathbf{r} - \mathbf{r}_i) = \sum_{i=1}^q \rho_i(\mathbf{r}) * \delta(\mathbf{r} - \mathbf{r}_i). \quad (2.27)$$

The convolution of the electron density $\rho_i(\mathbf{r})$ with $\delta(\mathbf{r} - \mathbf{r}_i)$ displaces the electron density function from the origin to the position \mathbf{r}_i .

The scattering from a single atom (called scattering factor), $f[(\mathbf{s} - \mathbf{s}_0)/\lambda]$, is given in section 2.1.5. The scattering from atoms with fixed positions and orientations is the Fourier transform of its electron density.

$$\begin{aligned} F[(\mathbf{s} - \mathbf{s}_0)/\lambda] &= \int \sum_{i=1}^q \rho_i(\mathbf{r} - \mathbf{r}_i) \exp[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}_i}{\lambda}] d^3 \mathbf{r} \\ &= \sum_{i=1}^q \int \rho_i(\mathbf{r} - \mathbf{r}_i) \exp[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot (\mathbf{r} - \mathbf{r}_i)}{\lambda}] d^3 \mathbf{r} \cdot \exp[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}_i}{\lambda}] \\ &= \sum_{i=1}^q f_i[(\mathbf{s} - \mathbf{s}_0)/\lambda] \exp[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}_i}{\lambda}] \end{aligned}$$

Given a 3-D periodic arrangement (crystal) of a single atom per unit cell, the electron density and scattering can be derived. The number of unit cells in \mathbf{a} , \mathbf{b} and \mathbf{c} directions are M , N and P , respectively and the atomic positions are $\mathbf{r}_{mnp} =$

$m\mathbf{a} + n\mathbf{b} + p\mathbf{c}$ where m , n and p are integers. The atomic position in one dimension with period a and number of unit cells M , would be $\mathbf{r}_i = a(i - (M - 1)/2)$. Therefore extrapolating to a 3-D arrangement, it is simple to restrict M , N and P to be odd. The electron density is then

$$\rho(\mathbf{r}) = \sum_{m=-(M-1)/2}^{m=(M-1)/2} \sum_{n=-(N-1)/2}^{n=(N-1)/2} \sum_{p=-(P-1)/2}^{p=(P-1)/2} \rho(\mathbf{r} - \mathbf{r}_{mnp}). \quad (2.28)$$

The scattering from this crystal is given by

$$G[(\mathbf{s} - \mathbf{s}_0)/\lambda] = \sum_{m=-(M-1)/2}^{m=(M-1)/2} \sum_{n=-(N-1)/2}^{n=(N-1)/2} \sum_{p=-(P-1)/2}^{p=(P-1)/2} f[(\mathbf{s} - \mathbf{s}_0)/\lambda] \exp[2\pi i(\mathbf{s} - \mathbf{s}_0)/\lambda]. \quad (2.29)$$

The function G is the result of a superposition of waves coming from each atom in the crystal. The conditions for constructive interference described in section 2.1.3 imply that constructive interference will only be observed when $(\mathbf{s} - \mathbf{s}_0)/\lambda$ coincides with the reciprocal lattice vector \mathbf{r}^* . Using Euler's formula for the complex exponential and evaluating the resulting geometric series where $\mathbf{r}^* = (\mathbf{s} - \mathbf{s}_0)/\lambda$ the scattering from a crystal with q atoms per unit cell becomes

$$G(\mathbf{r}^*) = \frac{\sin \pi M \mathbf{a} \cdot \mathbf{r}^*}{\sin \pi \mathbf{a} \cdot \mathbf{r}^*} \frac{\sin \pi N \mathbf{b} \cdot \mathbf{r}^*}{\sin \pi \mathbf{b} \cdot \mathbf{r}^*} \frac{\sin \pi P \mathbf{c} \cdot \mathbf{r}^*}{\sin \pi \mathbf{c} \cdot \mathbf{r}^*} \sum_{i=1}^q f_i(\mathbf{r}^*) \exp 2\pi i(hx_i + ky_i + lz_i), \quad (2.30)$$

where a phase factor, $2\pi i(hx_i + ky_i + lz_i)$ is introduced that describes the position of atoms away from the origin. The structure factor describing the amplitude and phase of a wave diffracted from lattice planes indexed by hkl is

$$F(hkl) = \sum_{i=1}^q f_i(\mathbf{r}^*) \exp 2\pi i(hx_i + ky_i + lz_i). \quad (2.31)$$

Notice that $\mathbf{r}^* = \mathbf{h}_{hkl} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$. From this relationship the lattice function $L(hkl)$ is obtained;

$$L(hkl) = \frac{\sin \pi n_x h}{\sin \pi h} \frac{\sin \pi n_y k}{\sin \pi k} \frac{\sin \pi n_z l}{\sin \pi l}, \quad (2.32)$$

where n_x , n_y and n_z are the number of unit cells in the x , y and z dimensions of the crystal.

The electron density of the crystal can be found from a Fourier back transform of the structure factor if both the modulus, $|F(hkl)|$, and phase, $\Phi(hkl)$, are known;

$$\rho(\mathbf{r}) = \sum_{h=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} |F(hkl)| \exp 2\pi i \Phi(hkl) \exp[-2\pi i(hx + ky + lz)]. \quad (2.33)$$

Given $f_i(\mathbf{r}^*)$ is written in terms of atomic electron density then $F(hkl) = \int \rho(\mathbf{r}) \exp 2\pi i(hx + ky + lz) d^3\mathbf{r}$.

In a real scattering experiment (neutron or X-Ray) the scattered intensities, not the structure factor, are measured. These intensities are proportional to the amplitudes of the secondary waves, $I(hkl) \propto F(hkl)F^*(hkl)$. The complex conjugate of the structure factor (given in equation 2.31) when multiplied by $F(hkl)$ results in loss of the phase information since the exponential factors cancel and the real quantity $|F(hkl)|^2$ is obtained. There are several methods for estimating the phases including Patterson methods [25], direct methods [26, 27, 28, 29], molecular replacement [30] and charge flipping [31, 32, 33]. These are incorporated in general structure analysis software such as *SHELX* [24] and GSAS [34, 35].

Atomic motion, as it depends on temperature weakens the scattered intensities, especially at high angles, and must be taken into account in the structure factor using a correction factor for atomic motion called the Debye-Waller factor. Atomic oscillations are on the order of tens of femtoseconds, therefore the observed diffraction experiment only provides a time-averaged distribution of the atom about its equilibrium position, $p(\mathbf{r})$. The time averaged atomic electron density is the

convolution of the atom's static density with a probability distribution (generally a 3-D Gaussian). The Fourier transform of the convolution is the product of the individual function's Fourier transforms. Therefore the static atomic scattering factor (Fourier transform of atomic density) is multiplied by a temperature dependent factor yielding the thermally averaged scattering factor. The probability distribution function is a Gaussian, if the atomic oscillation is approximated by a free harmonic oscillator and is given by

$$p(\mathbf{r}) = \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{|\mathbf{V}|}} \exp\left(\frac{-\mathbf{r}^T \mathbf{V}^{-1} \mathbf{r}}{2}\right). \quad (2.34)$$

The \mathbf{V} matrix elements are the mean square expectations of the atomic deviations from equilibrium (variance co-variance) along x, y, z, xy, yz and xz in dimensionless fractional coordinates. The Fourier transform of $p(\mathbf{r})$ yields the Debye-Waller factor,

$$q(\mathbf{h}) = \exp(-\mathbf{h}^T \beta \mathbf{h}). \quad (2.35)$$

The Debye-Waller factor multiplies the static atomic scattering factor to give the time-averaged scattering of an oscillating atom. The \mathbf{h} term is the reciprocal lattice vector with hkl indexes and β (atomic displacement parameter matrix) is related to \mathbf{V} by $2\pi^2 \mathbf{G}^* \mathbf{V} \mathbf{G}^*$ where \mathbf{G}^* is the reciprocal metric tensor. Finally the structure factor which now includes the correction for atomic motion is

$$F(\mathbf{h}) = \sum_{i=1}^q f_i(\mathbf{h}) \exp(-\mathbf{h}^T \beta \mathbf{h}) \exp(2\pi \mathbf{h} \cdot \mathbf{r}_i). \quad (2.36)$$

2.1.5 Scattering factors and scattering length

To calculate the X-Ray scattering factor of an atom, the volume is divided into small elements $d^3\mathbf{r}$ and for very small volume elements the summation is replaced by integration;

$$f[(\mathbf{s} - \mathbf{s}_0)/\lambda] = \int \rho(\mathbf{r}) \exp\left[2\pi i \frac{(\mathbf{s} - \mathbf{s}_0) \cdot \mathbf{r}_i}{\lambda}\right] d^3\mathbf{r}. \quad (2.37)$$

The spherical atom approximation is used to estimate the density, $\rho(\mathbf{r})$, and does not vary with θ linearly since $1/d = h = 2 \sin \theta / \lambda$. This is because the electrons are spread out over the atomic diameter. The scattering from the electron density at the top of an atom is slightly out of phase with the electron density at the bottom; this becomes more pronounced as the angle θ increases. The values of the elements have been calculated and are contained in [36]. Structure refinement programs use these values for X-Ray structure analysis. See Appendix section A. 1 for an example and further details.

Neutrons scatter from the nuclei of atoms, therefore there is no $\sin \theta / \lambda$ dependence. The scattering potential of a given atomic nucleus depends on its scattering cross section. The scattering cross section (σ) is the effective area of the nucleus that interacts with the neutron and is measured in barns (1 barn = 10^{-28} square meters). The strength (amplitude) of the scattered wave depends on the nuclear scattering length b (section 2.2.2). This means the scattering length of a given atom depends on the interaction between the neutron and the scattering nucleus. The relationship between b and the cross section is $\sigma = 4\pi b^2$. Therefore, the scattering length is half the radius of the nucleus that interacts with the neutron. The values of b have been determined experimentally for each nuclear isotope [37]. The nuclei of certain elements (e.g. H and Ti) interact with incident neutrons with an attractive rather than repulsive nuclear potential. This means the scattered waves are out of phase with respect to the incident neutron beam, resulting in a negative scattering length.

2.2 Diffraction experiment

2.2.1 X-Ray

X-Rays interact with the charged part of the atom as an electromagnetic wave. The electrons produce scattered waves that are in phase. The interaction of the electromagnetic radiation with the atom is dependent on the number of electrons (Z)

and is represented in the scattering factor. The dimensions of the electron cloud are comparable to the wavelength of X-radiation.

Heavy elements (e.g. Ti, Pb and Po etc.) scatter X-Rays more efficiently than the lighter elements (e.g. H, O and Ni etc.) and as a result dominate the intensities of the diffraction pattern. If a crystal contains a mixture of heavy and light elements the contribution of the lighter elements is relatively small and after deconvolution of the superimposed wave functions is more difficult to determine.

The radius of the Ewald sphere is $1/\lambda$, therefore shorter radiation wavelengths mean more reflections are measured, yielding a better ratio of reflections to refined parameters. For reliable statistics and a representative model the number of reflections should be at least 10 times the number of parameters in the least squares model.

General set-up

An X-ray diffractometer consists of three basic elements; (1) an X-ray source, (2) a sample holder, and (3) a detector. X-rays are generated by a cathode ray tube and filtered to produce monochromatic radiation which is then collimated and directed toward the sample. High intensity X-Rays are also generated at light source facilities using a synchrotron. A synchrotron uses bending magnets and undulators to produce high intensity and broad spectrum beam [28].

In a single-crystal experiment the crystal is mounted on a thin glass fiber (or loop) attached to a brass pin, using oil or glue to attach the crystal to the fiber tip or loop. The mounted crystal is placed on a conventional goniometer. There are 3 (or possibly 4) angles 2θ , χ , ϕ and ω . The angles define the relationship between the crystal lattice, the incident X-Ray beam and detector. The orthogonal X, Y and Z directions are adjusted to allow centering of the crystal within the X-Ray beam. X-Rays leave the collimator in a concentrated beam toward the sample and are either transmitted through the crystal or diffracted. A beam stop is located directly opposite the collimator and blocks the transmitted X-Rays to prevent the

detector from being over radiated and destroyed. Diffracted X-Rays at the correct orientation are measured by the detector. Modern single-crystal diffractometers use 2-D area detectors for fast data collection of multiple reflections simultaneously.

Data integration and corrections

Intensity data for structure solution is collected in a reflection data file called the *hkl* file that contains *hkl* indexes, their associated square structure factors and standard uncertainties for each measured intensity. This file is generated for X-Ray experiments through data reduction software. It is important to understand how the peak integrated intensities are extracted from the measured raw data. Individual diffraction “frames” are collected by rotating the crystal through the diffraction condition (across the Ewald sphere) while being exposed to X-Rays and collecting the diffracted intensity.

In any diffraction experiment it is essential to use a “good” crystal. The 3-D repeating arrangement of atoms acts as a tiny diffraction grating and therefore a crystal without defects, impurities and that is single (not multiple crystals fused together) and has regular faces (smooth without a large degree mosaicity) is important to produce the best possible data. The background is measured and subtracted from the peak. If the background is noisy and the peak is weak (low intensity) it is possible to generate negative peak intensities.

A polarization correction is also applied. Unpolarized X-Ray radiation is a one-to-one mixture of waves that are polarized parallel and perpendicular to the reflection plane. The incident radiation waves polarized parallel to the reflection plane and those perpendicular to the reflection plane produce different diffracted intensities and this effect must be taken into account. The angle between the parallel and perpendicular wave vectors is $\pi/2$ and $(\pi/2-2\theta)$ for the scattered wave. The intensity is proportional to the square magnitude of the scattered wave, therefore, the resulting polarization

factor is

$$\sin^2\left(\frac{\pi}{2}\right) + \sin^2\left(\frac{\pi}{2} - 2\theta\right) = \frac{1 + \cos^2(2\theta)}{2}.$$

If the X-Ray beam is polarized, such as that emanating from a synchrotron source the polarization factor is more complicated and includes the angle between the diffraction planes.

The Lorentz effect also needs to be taken into consideration and relates to the time the reciprocal lattice node takes to rotate in and out of the Ewald sphere. Since the crystal is rotated at a constant rate, the time each lattice node is in contact with the surface of the Ewald sphere depends on the location of the node in reciprocal space and the length of the scattering vector in real space. The following correction factor is applied $L = \frac{1}{\sin 2\theta}$.

The incident beam diminishes as it passes through the crystal. This affects the magnitude of the diffracted intensities in the experiment. The intensity loss is referred to as absorption and must be taken into account. These effects are represented by a linear absorption coefficient, μ (mm^{-1} or cm^{-1}):

$$I = I_o \exp(-\mu x)$$

The intensity I_o is the observed intensity and $\exp(-\mu x)$ is the factor by which the X-Ray beam is weakened as it passes through the crystal. Programs like SADABS [24] calculate a factor for each reflection dependent path length.

2.2.2 Neutron

Neutrons are complimentary to X-Rays. For both radiations the treatment as a wave and wave function is applicable. An advantage mentioned in section 2.1.5 is the neutron cross section is sensitive to isotopes of the same element and independent of the number of electrons. Therefore, mixtures of heavy and light atoms in the same

crystal structures have similar scattering lengths and as a result the atomic position of the lighter elements can be determined with neutron data.

Since neutrons do not carry a charge they are able to penetrate materials far better than charged particles (or waves). Neutrons are weakly interacting or gentle probes as they do not interfere with or destroy the sample. However, large sample volumes are generally required. Despite recent developments, available neutron beams are a flux limited technique. Generally, neutron diffraction is still a specialized technique and it is used when it can provide information on the material's structure that cannot be obtained via other techniques.

General set-up

The scattered neutron wave is isotropic and its wave function can be written as $(-b/r)\exp(ikr)$ when the point scatterer (nucleus) is at the origin. The scattering interaction is assumed to be elastic, therefore, the wave vector remains unchanged, therefore, k_0 , the incident wave vector is the same as k_i , the scattered wave vector. Scattering experiments are done for static structure determination. This means the incident neutron wave interacts with nuclei coherently therefore the nuclei have relative phases and interfere with one another. The same diffraction principals apply to single crystal X-Ray and neutron experiments. The diffraction conditions, structure factor and nuclear number (electron) density functions are equivalent. In a structure analysis the X-Ray scattering factor is replaced by the scattering length and the electron density is calculated from the measured intensities.

In neutron experiments using a monochromatic or polychromatic beam incident to a single-crystal sample the scattered neutrons are collected as a function of neutron time-of-flight (TOF). TOF is the time the neutron takes to reach the detectors from the target and through the sample. In this way, a given reflection and all harmonics of a reflection are collected at the same location on the detector at different TOFs= λs . This is time resolved Laue diffraction, whose individual reflection is separated and collected at a time of flight. The de Broglie equation gives the relationship between

the neutron wavelength, λ , to its momentum, $m_n v$, is

$$\lambda(m) = \frac{h}{m_n v},$$

where h is plank's constant (6.62610^{-34} Js or kg m²/s), m_n (1.67510^{-27} kg) is the mass of a neutron and v is the velocity. The velocity is inversely proportional to the wavelength λ . Replacing the velocity, v (meters per second) with the total path length L divided by time in seconds and relating to Bragg's law yields;

$$\lambda = \frac{ht}{m_n L} = 2d_{hkl} \sin Q.$$

Solving this relationship for time of flight dependence; $t = 505.56 L d \sin Q$ [38] where $505.56 = 2 * m_n / h$ (s/m). The number of Bragg reflections that can be measured with one crystal orientation depends on the characteristics of the source and the detector positions around the sample and unit cell size and symmetry.

Data integration and corrections

Polychromatic, TOF experiments require similar corrections to those required by X-Ray experiments with the exception of the polarization correction. In a TOF single crystal experiment many reflections are being collected simultaneously and are measured at different wavelengths (a white beam) and angles. The measured intensities vary according to

$$I_{hkl} = \frac{i_0(\lambda) V N^2 |F_{hkl}|^2 \lambda_{hkl}^4 \varepsilon(\lambda, \alpha) A_{hkl}(\lambda) E_{hkl}(\lambda)}{2 \sin^2 \theta_{hkl}}, \quad (2.38)$$

where I_{hkl} is the measured intensity of reflection hkl , $i_0(\lambda)$ is the incident flux, V is the volume of the unit cell, N is the number of unit cells, F_{hkl} is the structure factor, λ^4 represents the scattering power of the sample, $\varepsilon(\lambda, \alpha)$ is the detector efficiency as a function of the wavelength and detector coordinate and θ_{hkl} is the Bragg angle of the reflection. The Lorentz ($1/\sin 2\theta_{hkl}$) and absorption corrections are also included.

The Lorentz correction for X-Ray intensity data (section 2.2.1) is the length of time a Bragg reflection node takes to pass through the Ewald sphere. In TOF the crystal is not “moving” but the Ewald sphere contracts (since its radius is $1/\lambda$) through the diffraction condition as the wavelength λ increases during the pulse time frame.

Most TOF Laue data corrections (detector efficiency, Lorentz, background) are applied during the data reduction phase on the value of the integrated Bragg peak with the exception of the extinction correction. As part of Bragg peak integration, the background is subtracted from each Bragg peak. Newly available techniques for integrating TOF Bragg data in reciprocal space are described in section 4.5. The integrated intensities are scaled by the number of monitor counts during each measurement. The isotropic scattering data from Vanadium is used to correct the detector response per λ . The Bragg reflections for each crystal setting are now normalized and can be combined and refined as a single data set.

The absorption correction for neutron Laue TOF data is similar to X-Ray absorption. In the case of neutron experiments there are two factors to consider in the reduction of the primary beam intensity; (1) the decrease in intensity as the incident beam passes through the material (true absorption), which is represented by the cross section σ_a and (2) the reduction of beam intensity by incoherent scattering σ_{inc} (a result of randomly distributed isotopes and nuclear spins). The transmission factor is the ratio of the absorption-affected intensity to the absorption free intensity:

$$T = \frac{I_\mu}{I_o} = A/V \quad (2.39)$$

where V is the crystal volume and $A = \int \exp(-\mu L)dV$ is the absorption factor. For a given reflection μ is wavelength dependent and calculated in cm^{-1} . The mean free path L depends on the size and shape of crystal.

The extinction correction is applied to compensate for the weakening of the diffracted beam that occurs as a result of repeated reflections from multiple crystallite domains. The crystallite domains are not aligned with one another and have a mosaic

spread. There are two main factors that govern this effect; (1) the radii of the crystallite domains and (2) the range of their angular distribution throughout the crystal. The crystal is characterized by one of two types: type I, the particle size is the dominant effect (small particle size, large mosaic spread), type II, the mosaic distribution is the dominant effect (large particle size, small mosaic spread). The angular distributions can be represented by a Gaussian or Lorentzian model and have been incorporated into the least squares structure refinement of *SHELX* and GSAS.

2.3 Diffuse data

Analysis and interpretation of diffuse diffraction data, including its measurement and appropriate application of corrections in order to prepare the data for integration and computational modeling are more complex than for Bragg data. In the case of local structure resulting from static disorder as described in chapter 1 the size of the distortion in real space, L , affects the size of the scattering distribution in reciprocal space, δq , as

$$\delta q \sim \frac{2\pi}{L} \quad (2.40)$$

[39]. The inverse relationship implies that for small local disturbances in direct space the resulting diffuse scattering is distributed over larger regions of reciprocal space. The governing structure is represented by an “average” unit cell that is translated in the three crystallographic dimensions and defines the long range order in direct space. As a consequence, the diffracted intensities of the long range order are bright, sharp, discrete Bragg peaks. The local distortions in real space cannot be described in terms of a single unit cell that represents the structure but are only locally correlated over a finite number of unit cells. The structure variations produce diffracted intensities that are not localized to a single, discrete peak but rather diffuse in reciprocal space with intensities that are generally orders of magnitude weaker than those of Bragg

scattering. The diffuse intensities, which depend on the nature of the local structure, may be distributed in one, two or three dimensions of reciprocal space.

Measuring diffuse intensities with a monochromatic radiation source means that each intensity point is measured using a single Ewald sphere of radius $1/\lambda$. The crystal is continuously rotated in the beam resulting in a 2-D slice of reciprocal space measured per data collection interval. The 2-D slices are reconstructed to form a full 3-D volume of reciprocal space. Diffuse data are often collected using highly efficient 2-D area detectors. The reconstructed data are corrected and the diffuse intensities extracted for modeling.

In neutron TOF Laue experiments, the use of a polychromatic beam means that there are multiple diffraction conditions (Ewald spheres). These intensity points are wave length resolved, therefore each intensity is separated according to the incident flux and the resulting coverage of reciprocal space is a volume defined by the Ewald spheres of radii $1/\lambda_{max} - 1/\lambda_{min}$ per orientation. This means the reciprocal space volumes from each orientation are reconstructed to form the full measured reciprocal space volume.

Each Bragg peak is approximated as measured at a single wavelength, therefore the previously described data correction methods can be applied after integration. In the case of TOF Laue diffuse data, since it occupies large volumes (or larger than that of a single Bragg peak) of reciprocal space it cannot be corrected for incident flux in the same manner as individual Bragg peaks but must be corrected for each intensity point. Using the newly available event mode data processing it is possible to apply the corrections to the neutron events prior to integration. In this work we expand the correction protocol from post-integration to pre-integration correction taking the wavelength variations of TOF into account. A first step in this process of testing and developing the pre-integration corrections is described in Chapter 4.

Modeling diffuse data follows the same principle as modeling a Bragg structure. As in Bragg structure refinement, the data corrections are applied to the diffuse data before modeling, since only the contribution to the structure is modeled. The

examples discussed in this work produced 1-D and 2-D diffuse scattering. Diffuse data in one dimension is integrated as a “rectangle” with a certain width and length based on the reconstructed layers. The intensities are summed along the direction of length of the rectangle. The background is calculated by summing the along the length of the rectangles defined by the width $\pm\Delta$ and subtracting it from the intensities calculated at each point along the length of the rectangle containing the diffuse data. In two dimensions a layer thickness is determined and the intensities are integrated on a two dimensional grid (chapter 3). The background $\pm\Delta$ of the layer thickness is calculated and subtracted from the diffuse intensities at each point of the grid. Separating the background from the diffuse intensity, which may be on the same order of magnitude, presents a challenge that must be addressed on a case by case basis. Determining appropriate integration and correction protocols for diffuse TOF data with each intensity point possessing a different TOF and wavelength still remains a challenge.

Chapter 3

Computational techniques

3.1 Quick overview of average structure refinement

Modeling modulated diffuse scattering of any compound begins with an average structure refined from the Bragg intensities. The Bragg refinement is the optimized time and space averaged structure and describes the long range order that provides the necessary framework for modeling the local structure arrangement. Detailed analysis of the average structure gives first indications that a local structural arrangement might be present. Possible structural alternatives can be represented in the average structure by the presence of one or more of the following; partial occupancies, mean square displacement parameters that contain U_{ij} element(s) that are enlarged (elongated ADPs) and/or unchemical bond distances and intermolecular short contacts (chapter 6).

Single crystal structure refinement is done via a least squares optimization of atomic positions through a comparison of the measured and calculated intensities. The atomic positions are assigned based on electron density peaks and previous knowledge of the chemical composition of the crystal. The R_1 value is given as;

$$R_1 = \frac{\sigma \sum ||F_o| - |F_c||}{\sigma \sum |F_o|}, \quad (3.1)$$

where F_o is the observed and F_c is the calculated structure factor. The R_1 value is minimized during refinement.

The result is the “crystal structure” which is a file containing a list of atomic positions, site symmetry, atomic displacement parameters (ADPs), bond distances and angles called a crystallographic information file (CIF). Shelx [24] and GSAS [35] are commonly used single crystal structure refinement packages.

3.2 Brief description of ZODS-Zürich Oak Ridge Disorder Simulations

The ZODS (Zürich Oak Ridge Disorder Simulations) [40, 41] software package is designed to extract quantitative structural information from the diffuse part of the diffraction pattern [41]. A disorder model is constructed starting with the “disentanglement” of the average structure. This means virtual crystals are built using this local structure model parameters via Monte Carlo (MC). The number of virtual crystals must be large enough to encompass the full range of disorder possibilities building one model crystal per option. The crystal must also contain enough unit cells to contain the full correlation length of the local structure. The computational resources required to build enough crystals of sufficient size to represent the local structure ordering options can be significant.

The diffuse scattering intensities of the simulated crystals are calculated and compared with measured intensities. The model parameters producing simulated crystals that yield the best intensity matches to the measured data are selected and continue to be refined using a generational algorithm, called differential evolution (DE) (section 5.3.1). DE is computationally demanding but efficiently parallelized (chapter 5). ZODS takes advantage of this parallelization and is capable of running on large super computers, smaller clusters and desktop computers.

From the model parameters ZODS builds the crystals using MC (in direct space) and calculates the intensities (reciprocal space) of the model crystals. These results can be visualized and analyzed in freely available external software. For the analysis of crystals in direct space Accelrys Discovery Studio 3.5 [42] can be used and for the intensities ParaView [43] is suggested. ZODS provides statistical analysis of the simulated crystals, convergence plots and statistics of the model parameters, MC crystal energy data, R-value, background and the scale factor for calculated intensities.

3.2.1 Measured diffuse data preparation

In Bragg diffraction the reflections are at integer hkl in reciprocal space. In the case of diffuse scattering this is no longer true. Only the local structure contributes to the diffuse scattering. Since the local structure cannot be represented as a single unit cell but is distributed throughout the governing structure the resulting diffraction pattern shows diffuse intensities distributed throughout large volumes of reciprocal space (chapter 5 section 5.1). The measured diffuse intensities are corrected and integrated over their distributions in reciprocal space (layers, streaks etc.). The diffuse intensities are placed on a regular grid for computational modeling. Depending on how the diffuse scattering is distributed in reciprocal space it is placed in a regular 1- 2- or 3-D array. The model intensities are calculated by stepping along the grid using a step size commensurate with the measured intensity data array.

The calculation grid specifies the location of the measured data points in reciprocal space (figure 3.1) and can be defined in fractional or Cartesian coordinates. The grid consists of an origin vector \mathbf{o} and three directions vectors \mathbf{d}_1 , \mathbf{d}_2 and \mathbf{d}_3 and the dimensions in each direction n_x , n_y and n_z . Each $\mathbf{r}_{i,j,k}$ grid point can be written as $\mathbf{r}_{i,j,k} = \mathbf{o} + i\mathbf{d}_1 + j\mathbf{d}_2 + k\mathbf{d}_3$ where the ijk indices are $(0 \leq i \leq n_x, 0 \leq j \leq n_y, 0 \leq k \leq n_z)$.

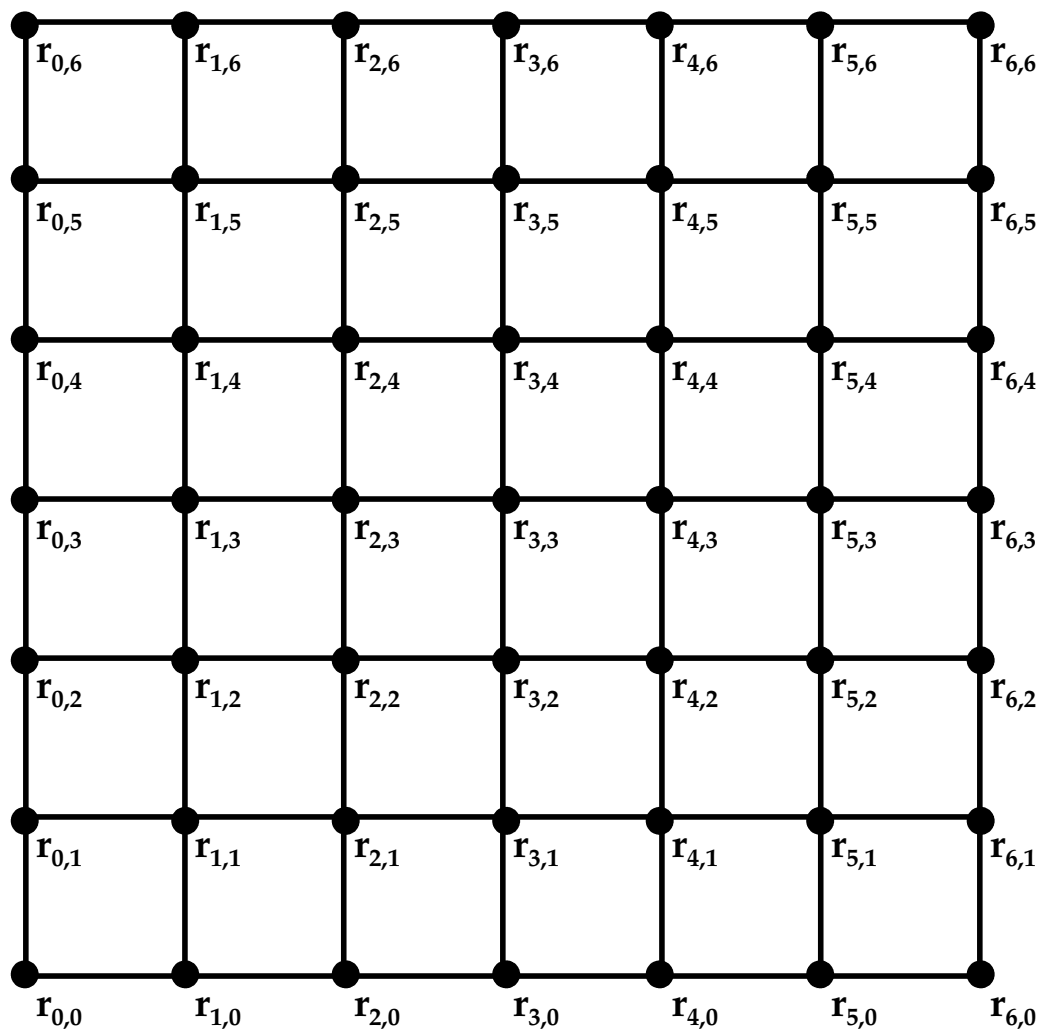


Figure 3.1: An example of a 2-D calculation grid for the calculation of intensity points in reciprocal space, depicting the variation of $\mathbf{r}_{i,j}$ where index \mathbf{i} is in the \mathbf{d}_1 direction, index \mathbf{j} is in direction \mathbf{d}_2 and the origin is at $\mathbf{r}_{0,0}$. The step size is the spacing between consecutive $\mathbf{r}_{i,j}$ points.

If Cartesian coordinates (units of \AA^{-1}) are used in ZODS the coordinate system is defined so that \mathbf{a} is collinear with $(1, 0, 0)$ and \mathbf{c}^* is collinear with $(0, 0, 1)$ and the transformation from fractional coordinates to Cartesian coordinates in direct space is:

$$\mathbf{v}_C = \begin{bmatrix} a & b \cos \gamma & c \cos \beta \\ 0 & b \sin \gamma & c \frac{\cos \alpha - \cos \beta \cos \gamma}{\sin \gamma} \\ 0 & 0 & \frac{V}{ab \sin \gamma} \end{bmatrix} * \mathbf{v}_f$$

where \mathbf{v}_C corresponds to a vector in Cartesian coordinates and \mathbf{v}_f is a vector in fractional coordinates and V is the volume of the unit cell. The step length of the intensity data in \AA^{-1} is multiplied by the normalized reciprocal lattice vector that is collinear with x, y or z in the data. Then the step must be defined according to the diffuse data grid in x, y and z (an example is given in chapter 6).

3.2.2 MC crystal building

Using the long range structure allows the creation of a “super cell”. The “super cell” is a larger unit cell that fully resolves the atomic positions contributing to the local structure, which are only averages of the atomic options in the average structure. In our case, partially occupied positions are separated into fully occupied, alternating positions (chapter 6). This results in a larger unit cell and it becomes necessary to adjust the unit cell parameters accordingly. The unit cell parameters, symmetry operations, atomic positions and occupancy factors are adjusted in the CIF file. ZODS uses the adjusted CIF file as input for the average structure to build the crystal model.

Once the larger “super cell” has been defined the local structure ordering options are disentangled through a chemical unit (CU) representation, each with one or more sets of alternatives based on the total number of options that belong to a given chemical unit. The CUs serve as the building blocks representing one disorder option at a time and are part of the crystal that contributes to the diffuse intensities. Generally many crystals are possible and need to be modeled

simultaneously. Each chemical unit and its associated alternatives are assigned a probability. Interactions between chemical units are defined and a corresponding interaction energy is assigned to each CU interaction. The interaction(s) between the chemical unit(s) are parameterized and then used by the ZODS Monte Carlo crystal builder to construct model crystals. The interaction between CUs is defined as a discrete, Ising interaction and defined between a CU at one site in a crystal with a CU at another site.

Monte Carlo (MC) adjusts the initial guesses by randomly swapping pairs of chemical units between the sites they occupy. This swap results in a crystal with an overall lower MC energy if $\Delta E = E_i - E_{i+1} \leq 0$ and if $\Delta E = E_i - E_{i+1} > 0$ then the swap is accepted with a probability of,

$$P(E) = e^{-\Delta E \frac{k\beta}{T}},$$

where ($T = \frac{1}{k\beta}$). This probability, $P(E)$, is calculated and compared to a random number generated between 0 and 1. If $P(E)$ is less than the random number the swap is accepted otherwise it is not. Each attempted swap is a MC step. A MC cycle means that every site in the crystal has had at least one attempted change, whether or not it was accepted. The MC process is repeated until the crystal energy shows no improvement (equilibration). Appendix A. 1 describes the MC crystal building for a specific example.

3.2.3 Model calculations and optimization

The equilibrated crystal structure intensities are calculated by first dividing the crystal into smaller crystals (lots) [15, 44], Fourier transformed, squared and incoherently summed to obtain the simulated crystal intensity. The lot size needs to be large enough to include all correlated neighbor interactions. The correlation length is indicated by the systematic change in the conditional probabilities of neighbors.

The number of lots is chosen so that the entire simulated crystal is sampled at least once; the number of lots multiplied by the lot dimensions should be greater than or equal to the simulated crystal dimensions.

The model parameters are then optimized through a generational algorithm called differential evolution (DE) [45] that refines the model parameters through an iterative process that minimizes the difference between the observed (measured) and calculated intensities, called the objective function (chapter 5 section 5.3.1) or R value in this case. Multiple crystals are built with the same parameter set (clones) to sample a representative area of the fitness distribution of crystals. The model crystal intensities are calculated by incoherently summing over lots and averaging the individual crystal intensities over clones. DE optimization continues until convergence is achieved with a specified stopping criterion, such as the number of optimization cycles.

The objective function (R value) [15] minimized by DE is given by,

$$R = (A \sum_k \sigma^{-2}(\mathbf{h}_k) [I_{obs}(\mathbf{h}_k) - sI_{calc}(\mathbf{h}_k) - b]^2)^{-\frac{1}{2}} \quad (3.2)$$

where R is summed over all reciprocal data points \mathbf{h}_k with corresponding standard deviation σ_k . The observed and calculated intensities are given by $I_{obs}(\mathbf{h}_k)$ and $I_{calc}(\mathbf{h}_k)$. The s term is the scale factor, b is the background and A given by

$$A = \frac{1}{\sum_k \sigma^{-2}(\mathbf{h}_k) I_{obs}^2(\mathbf{h}_k)}$$

is the normalization constant. The scale factor, s , and background, b , are calculated using linear least squares which solves a system of linear equations

$$\frac{\partial R}{\partial s} = \frac{\partial R}{\partial b} = 0$$

for the optimized objective function at each $I_{calc}(\mathbf{h}_k)$. If the solution of

$$\frac{\partial R}{\partial b} < 0$$

a penalty is assigned by setting the objective function to one. If the solution of

$$\frac{\partial R}{\partial s} < 0, \quad \frac{\partial R}{\partial b} > 0$$

for s and b then b is set to zero and only s is optimized. If solutions of

$$\frac{\partial R}{\partial b} > 0, \quad \frac{\partial R}{\partial s} > 0$$

for s and b , both are retained without change.

The model results can be analyzed in direct and reciprocal space. The differences between calculated and observed intensities can be calculated using a ZODS utility program, “compare intensities” and can then be visualized in Paraview [43]. The direct space results can be analyzed by calculating chemical unit neighbor pair correlations (see Appendix A. 1 for a short 4 neighbor example). A principal component analysis can be done (section 7.2.5) on parameters [46] using the ZODS “analyse” utility. Parameter convergence statistics can be calculated and graphically analyzed in the plotting utility program “plot tools” (Appendix A. 4.2 and chapter 6). The general concepts of MC crystal building and equilibration, lots, neutron and X-Ray intensity calculation on a grid and neighbor pair correlation statistics are incorporated into an exemplary 2-D crystal modeling code written to demonstrate occupational disorder on a 2-D hexagonal lattice (Appendix A. 1). Details of the use of the ZODS program and utilities are demonstrated in chapter 6.

Chapter 4

Neutron data of β [beta]-NaLaF₄

4.1 Introduction

Er³⁺ doped β (hexagonal phase) NaLaF₄ from a series of β -NaLaF₄ (Ln=Y-Lu) belongs to a family of light emitting Sodium Lanthanide tetra fluorides. Some of the rare earth doped compounds are efficient upconversion phosphors. They are currently heavily studied as luminescence host matrices [17]. Owing to their biological compatibility [47, 48, 49, 50], deep tissue penetration [51] and virtually zero auto fluorescent background [52] upconversion nanocrystals (UCNCs) have been developed for cell-labeling and tracking [53], small animal imaging [48], delivery of drugs [54], photodynamic therapy [55], and photothermal therapy [56]. UCNCs also have a variety of non-biological applications including lasers [57, 58, 59], solar cells [60], wave guides [61, 62] and display devices [63].

As the internal make-up of UCNs is crystalline, the upconversion process is supported by the crystal structure [64]. The host matrix (crystal lattice) combined with a dopant (usually Lanthanide ions) in low concentrations form the upconversion phosphor [4]. The crystal structure of the host matrix provides a framework that brings the luminescent centers, triggered by the dopant, into optimal position [4, 65, 66]. Thus, the inherent properties of the crystalline host matrix and

its interaction with the dopant ions dramatically affect the ability to support an upconversion process. To date the most efficient upconversion phosphor, introduced in 1972 by Menyuk et al. [67] and Kano et al. in 1973 [68] is Yb³⁺ and Er³⁺ doped NaYF₄.

To understand the interplay between the dopant ions and the crystalline host an unambiguous characterization of the local spectroscopic site symmetry is essential. There have been several conflicting crystal structural descriptions of NaLaF₄ [4, 65, 69], from X-Ray single crystal diffraction data [4], polarized Raman, infrared spectroscopic techniques and piezoelectric resonance spectra [9]; however all of these methods confirmed P $\bar{6}$ symmetry, which is a non centrosymmetric space group.

Polarized single-crystal absorption spectroscopy studies revealed that La³⁺ occupies an optically active site lacking mirror symmetry (C_1), whereas the crystallographic structural symmetry indicated a higher C_{3h} symmetry for all La³⁺ [4]. Closer examination of the average structure revealed that local structure must account for this apparent discrepancy in the two results [4]. This is also supported by diffuse scattering observed in the diffraction pattern. Only a qualitative estimate of diffuse X-Ray data was attempted, which described the general correlations of a frustrated local arrangement of the Na₂/La₂ cation columns in β -NaLaF₄ [4].

X-Ray diffraction yields a good average structure and provides adequate data, however the availability of neutron data would provide a complementary data set to verify the predicted local structure. Complementary neutron TOF Laue single crystal diffraction data was collected providing the opportunity to study the structure characteristics and validity of the models from both data sets. The combination of heavy and light elements present in β -NaLaF₄ make the use of neutron diffraction particularly useful due to the more similar scattering powers of Na, La and F than X-Rays. The X-Ray diffraction pattern is dominated by La³⁺(57 electrons) since it has a large scattering factor relative to the other elements, F and Na, which only have 9 and 11 electrons respectively. With the availability of high flux and intensity neutron sources it is feasible to better determine the Na and F positions via neutron

diffraction. The coherent scattering length for F is 5.654 fm, for Na it is 3.63 fm while for La it is 8.24 fm, values that are more similar than the X-Ray scattering factors [37]. In our study we focus first on developing and testing variable wavelength TOF Bragg data correction and integration techniques. For this work, 100K TOF Bragg data was collected from large β -NaLaF₄ single crystals on the single crystal diffractometer TOPAZ at the Spallation Neutron Source at Oak Ridge National Laboratory. Two TOF data integration techniques are tested and compared. Complementary 100K single crystal X-Ray data was collected, integrated and refined as P $\bar{6}$. The anisotropic displacement parameters for both the neutron and X-Ray data were compared and their differences discussed.

4.2 Bragg scattering: neutron TOF data

A large single crystal of β -NaLaF₄ (approximately 1 cubic millimeter) was selected from a batch of crystals grown using the Bridgeman technique by collaborators at the University of Bern. The crystal was mounted on an aluminum pin using super glue and placed on the goniometer equipped with a 100K nitrogen cryostream. Neutron event data were collected and viewed live using ISAWEV [70]. The first Bragg reflections appeared in a matter of minutes. The quality of the single crystal was determined by examining the initial Bragg peaks. After three attempts a single crystal was found with only single Bragg peaks without splitting or multiples.

The **UB** matrix is found using the strongest Bragg peaks to determine the smallest primitive cell (Niggli cell). The Niggli cell and its orientation in the instrument define the **UB** matrix. The Niggli cell and its corresponding **UB** matrix is calculated using a fast Fourier transform (FFT) algorithm and refined using least squares. The Bragg peaks are projected on possible vectors generated from the range of values for each a , b and c . Fast Fourier transforms (FFT) of the projected peaks are calculated and regular patterns are found in the collection of peaks identified. The magnitudes and directions of the **a**, **b** and **c** vectors are optimized through a least squares

algorithm which maximizes the number of peaks indexed. The three shortest, linearly independent vectors are used to form a unit cell with a volume whose corresponding **UB** matrix indexes at minimum of 80% of the maximum number of Bragg peaks indexed by any other three vector combination tested. The best three vectors are used to form a new **UB** matrix that is again optimized using least squares. Finally, starting with the optimized **UB** matrix, a new **UB** matrix corresponding to the Niggli reduced cell is calculated. This is the **UB** matrix that is used to index the Bragg peaks in the measured data within the specified tolerance in this case (0.12 deviation from h,k or l).

The peaks indexed using the **UB** matrix were used to plan the experiment using the Crystal Plan software [71] to maximize the coverage per orientation. Twelve crystal settings were collected for approximately 2 hours per setting. This resulted in a Bragg data set with completeness greater than 90%. The Bragg data were integrated, corrected for Lorentz, spectrum and absorption effects and refined with the single crystal refinement software program GSAS, which allows for Laue time-of-flight extinction correction and scaling. 2125 independent measured reflections were refined to an overall $R = 0.0557$, Goof = 1.025 (CIF file Appendix A. 3.2). Diffuse data was also collected for 12 crystal settings for approximately 11 hours per setting.

4.3 TOF data correction

Background and spectrum corrections for diffuse single crystal neutron TOF data in reciprocal space have not to date been developed. Aspects of a new protocol for correcting the multiple wavelength data on a neutron by neutron basis have been studied and tested for validity on Bragg data in MANTID [5], where the results can be first compared to the common post-integration reduction and then to the diffuse data. A common TOF data correction process of Bragg peaks to subtract the background and divide the isotropic scatterer by Vanadium spectrum is used (equation 4.1)[21]. The sample data is loaded with the prompt pulse range [1500, 15666] counts. The

detector calibration is included for the sample data. Next these data are normalized to the monitor counts. Monitor counts are integrated over the range [1000,12500] and the sample data is divided by this value. The background is measured without a sample and the detector calibration is also normalized to monitor counts and subtracted from the sample. Any resulting values of zero along with detector edges (16 pixels) are masked. After masking the Vanadium (isotropic scatterer) is normalized to monitor counts and the normalized background is subtracted. Any values of this difference resulting in zero are masked along with the edges. Finally the normalized differences are divided as shown in relation 4.1 below.

$$\frac{\frac{S}{\int m_S d_{TOF}}}{\frac{V}{\int m_V d_{TOF}}} = \frac{\frac{B}{\int m_B d_{TOF}}}{\frac{B}{\int m_B d_{TOF}}} \quad (4.1)$$

The correction to adjust for the effects of spectrum, which is not flat, is applied to the sample data. The raw data corrections are done on all intensity data in reciprocal space (Bragg and diffuse). The spectrum correction is simplified for the post-integration Bragg peaks [21]. In the case of Bragg intensities, they are assumed to be at a single point in reciprocal space and not distributed over volumes of reciprocal space as in the case of diffuse scattering, so only individual points on the spectrum are used and not a continuous spectrum. In the case of diffuse scattering the spectrum correction is applied to the diffuse intensities (see figure 4.1) over the entire diffuse pattern. This means that each measured neutron event is corrected for its associated wavelength.

4.4 Experimental details

The crystal structure determination for β -NaLaF₄ is described in the following text. A crystal of globular shape with size of $1.4 \times 1.4 \times 1.0 \text{ mm}^3$ and space group $P\bar{6}$ was refined with unit cell dimensions $a = b = 6.1520(14)\text{\AA}$, $c = 3.8210(8)\text{\AA}$. The unit

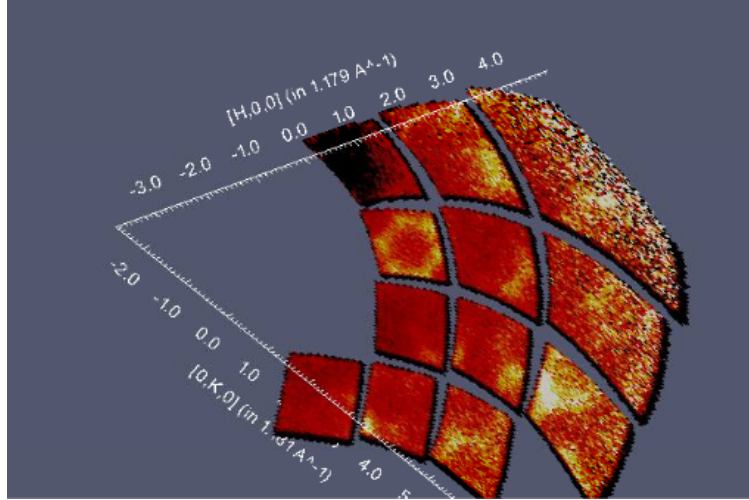


Figure 4.1: The $HK4.5L$ layer of NaLaF_4 collected on single crystal TOF diffractometer TOPAZ reveals the same diffuse “honeycomb” pattern observed in X-Ray experiments [4]. This is a view generated using MANTID with data corrections applied.

cell volume was calculated as $V = 125.43(5)\text{\AA}^3$ and the density, $\rho_{calc} = 4.724\text{ gm}^{-3}$. The number of chemical units per unit cell was $Z = 1$. The data was measured at $T = 100\text{K}$ with incident neutron radiation wavelength range of $\lambda = [0.5\text{--}3.5]\text{\AA}$. A total of 2125 reflections were measured resulting in 2042 unique reflections with angle $2\theta_{max} = 162^\circ$. A spherical absorption correction was done in the TOF data correction and integration software written by A. Schultz called anvred2x [72]. The calculated absorption correction was $\mu = 0.04\text{ cm}^{-1}$ at 1.8\AA and the transmission minimum and maximum were $T_{min} = 0.9631$ and $T_{max} = 0.9788$, respectively. A secondary type 1 Lorentzian extinction correction was modeled in GSAS resulting in an extinction coefficient of 1.3560×10^{-4} . The final refinement to generate the CIF file was done with a full-matrix least squares refinement against F^2 in SHELX97 [24] using 2042 reflections ($I > 2\sigma(I)$) and 20 parameters. The resulting fitness parameters were; $R_1 = 0.0534$ ($I > 2\sigma(I)$), $wR2 = 0.1354$ (all data). The residual neutron number density was $+1.530$ and -1.558 for $d_{min} = 0.4\text{\AA}$ high resolution data.

4.5 TOF data integration methods

Discrete single crystal Bragg peaks were integrated using two different peak integration methods in reciprocal space; (1) spherical integration and (2) elliptical peak fitting (see figure 4.2). Both integration algorithms begin with the same peaks found and indexed using the fast Fourier transform found **UB** matrix. The events are detected in $x - y$ TOF detector space and mapped to reciprocal space. Integration is performed by summing all neutron events inside a chosen radius (in \AA^{-1}) around each peak position. Since the error associated with each event is assumed to be random and independent they are summed in quadrature.

If the chosen radius results in an integration volume that is either partially or entirely off the detector edge, the peak is discarded. The background is estimated by defining a second “shell” of a specified thickness around the peak. This defines a shell around the volume containing the peak, which is used to calculate the background. The background density (intensity) within a given peak radius is calculated by scaling the shell intensity to the volume of the peak as

$$I_{bg} = I_{shell} \frac{V_{peak}}{V_{shell}};$$

and the square error is given by,

$$\sigma I_{bg}^2 = I_{shell} \frac{V_{peak}}{V_{shell}} \sigma I_{shell}^2.$$

The integrated intensity is corrected (background subtracted) as $I_{corr} = I_{peak} - I_{bg}$ and the errors ($\sigma(I)$) are summed in quadrature.

4.5.1 Spherical

The general integration method described above is applied to a sphere defined by a single peak. Neutron events are weighted by their standard uncertainties. Events in

the defined spherical region of the indexed peaks are summed and the event standard uncertainties (errors) are summed in quadrature. The background intensity is scaled and calculated as above and the final integrated peak intensity is also calculated as described above.

4.5.2 Elliptical

The elliptical integration algorithm integrates discrete single crystal Bragg peaks by summing all raw, unweighted events contained in a 3-D elliptical region of reciprocal space and subtracting the background estimated from an ellipsoidal shell that is derived from the peak radius (figure 4.2). The inner and outer background sizes as described above define the ellipsoidal background shell that is subtracted from the peak. The length of the major axis of the ellipse is fixed and kept the same for all peaks; the other two axes defining the 3-D ellipse are adjusted based on the standard deviation of the events in their directions. The major axis is defined to be in the direction of the principal axis, which is the direction whose standard deviation is the largest. The other two ellipsoidal axes are in the direction of other axes and scaled according to the major axes in proportion to their standard deviations. In addition to the ellipsoidal radii a region radius must also be specified. This defines the maximum distance from the peak center that is considered for integration in reciprocal space. An event is assigned to at most one peak with the closest hkl value. The region radius should be slightly larger than the expected peak region in order to ensure that the entire peak is integrated (figure 4.2).

4.6 X-Ray data collection

An X-Ray data set was collected as a basis of comparison to the neutron data using a crystal from the same batch. All data were measured at 100K, using a nitrogen cryostream. The refinement and data collection details of NaLaF₄ are given here.

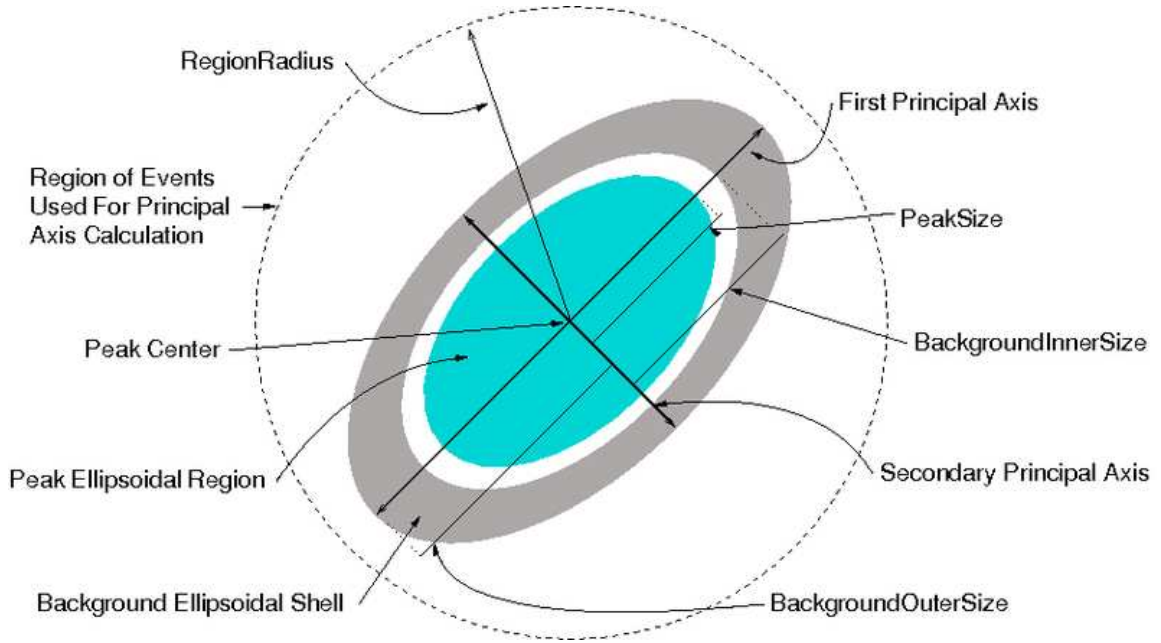


Figure 4.2: Depiction of the ellipsoidal region generated in the peak integration algorithm around each found Bragg peak. The event region is indicated by the dashed line surrounding ellipse. Figure from MANTID [5] project website.

Crystals of globular shape with size $0.12 \times 0.1 \times 0.09 \text{ mm}^3$ were used for data collection. The crystal structure had hexagonal symmetry with space group $P\bar{6}$ and was refined with unit cell dimensions $a = b = 6.1520(14)\text{\AA}$, $c = 3.8191(9)\text{\AA}$ and unit cell volume, $V = 125.18(6)\text{\AA}^3$ with density $\rho_{calc} = 4.734 \text{ gcm}^{-3}$. A Bruker Smart 1 K area detector diffractometer with graphite-monochromated $\text{MoK}\alpha$ radiation and wavelength $\lambda = 0.71013\text{\AA}$ using 0.3° ω -scans and SMART and SAINT software were used. A total of 816 reflections were measured resulting in 239 unique reflections at angle $2\theta_{max} = 56.37^\circ$. An empirical absorption correction with SADABS [24] was done with $\mu = 12.848 \text{ mm}^{-1}$ and the minimum and maximum transmission were $T_{min} = 0.5704$ and $T_{max} = 0.7457$, respectively. A least squares, full-matrix refinement against F^2 with SHELX97 [24] using all 816 reflections ($I > 2\sigma(I)$) and 22 parameters resulted in an $R_1 = 0.0138$ ($I > 2\sigma(I)$) and $wR2 = 0.0346$ (all data). The inversion twinning ratio was 0.43 and refined as a two component inversion twin. The residual electron density was $+0.43 / -0.51\text{\AA}^{-3}$.

Chapter 5

New computational method for analysing diffuse scattering

The results in this chapter are an expanded version of the paper:

Michels-Clark, T.M.*, Lynch, V.E., Hoffmann, C.M., Hauser, J., Weber, T., Harrison, R., Bürgi, H. B. Analyzing diffuse scattering with Supercomputers. *J. Appl. Crystallogr.*, **46**, 1616-1625

I was lead and corresponding author and my primary contributions to this paper include (i) testing and development of method and analysis (ii) running of computations (iii) writing manuscript and preparing all figures for publication (iv) writing of results analysis code (v) corresponding with editor and referees (vi) final submission.

5.1 Introduction

Interesting and exploitable macroscopic material properties of functional single crystals are often related to microscopic local deviations from a periodic average structure. These deviations may take the form of static or dynamic disorder and manifest themselves as diffuse scattering (DS) in one, two, or three dimensions (rods, layers, clouds) concomitant with Bragg scattering [12]. Phonon related

scattering resulting in thermal diffuse scattering (TDS) [73] is not the focus of this investigation. Here we focus on static disorder associated with local structural variations. Qualitative DS analysis can show general aspects of disorder, but only a quantitative analysis can reveal details of the deviations from the average (or Bragg) structure and provide a basis for explaining the origin of the functional properties [4, 74]. Only the disordered atoms or molecules within the overall structure contribute to diffuse scattering, which, being distributed over extensive volumes of reciprocal space, is usually orders of magnitude weaker per unit volume of scattering space than Bragg diffraction. This means that measuring the diffraction of structurally disordered materials requires a careful experimental set-up at powerful neutron and synchrotron sources and careful discrimination of the experimental noise from scattering introduced by the sample environment and not by the crystal itself. Better radiation sources, detectors, and data reduction routines make the acquisition of reliable diffuse scattering data increasingly tractable.

Usually, preliminary knowledge of the structural disorder is vague and consists mainly of chemical or geometric rules that are violated in the average structure. The measured, quantitative information requires quantitative modeling, which necessitates iterative optimization of empirical disorder parameters. Growth or Monte Carlo (MC) models are most effective for estimating an initial disorder model and equilibrating it [15]. Lattice energy minimizations have also been used to qualitatively verify the local structural disorder, and the energy-minimized structures were found to qualitatively reproduce the observed diffuse diffraction pattern well [14]. Intrinsic issues with these techniques are twofold: Firstly, the size of the constructed model crystals needs to be sufficiently large to encompass the observed disorder (short-range-ordered) motifs, but is generally still small compared to the scattering volume of the actual sample. Secondly, a global optimization procedure is needed to find the best empirical parameters for describing the disorder and their numerical values. The substantial computational resources needed to resolve these issues are becoming increasingly available with advances in computer technology.

In this work three aspects of computationally modeling stacking disorder are investigated with the help of growth models [75]. 1) We compare the efficiency of different algorithms for global optimization of model parameters, namely, Differential Evolution (DE) [45], Genetic Algorithm (GA) [76], and Particle Swarm Optimization (PSO) [77]. 2) We analyze the speckle-type intensity variations inherent in all procedures for modeling disorder using the concept of “clones”, *i.e.*, model crystals that are independently built from a single set of model parameters. 3) The concept of clones lends itself to parallelization on super or grid-computers. Here we report on the scalability of such parallelization.

In section 5.2 the chemical model system is described and growth modeling of stacking disorder is sketched. Section 5.3 describes three global optimization methods for the parameters of the growth model and ways to parallelize the computations. Section 7.2 compares the performance of the optimization methods and of the parallelization. It also summarizes the structural results obtained.

5.2 Chemical model system, growth modeling, and reference data

Tris(bicyclo[2.1.1]hexeno)benzene (TBHB) crystallizes in multiple stacking variants of threefold symmetric layers consisting of coplanar, three-pointed star-shaped molecules (Figure 5.1). The polymorph with space group $P6_3/mmc$ and Bragg lattice parameters $a' = b' = 5.2145(5)\text{\AA}$ and $c' = 8.9429(8)\text{\AA}$ shows diffuse streaks of scattering intensity at non-integral values of $(-h' + k')/3$ (with $h', k' = integer$) [18]. The streaks indicate faulted layer stacking [78]. The unit cell of a single layer is $\mathbf{a} = \mathbf{a}' - \mathbf{b}'$, $\mathbf{b} = \mathbf{a}' + 2\mathbf{b}'$ with lengths $a = b = \sqrt{3}a'$. Correspondingly the diffuse streaks are indexed as $h k L$ with $-h + k \neq 3$ and L the continuous variable along the streaks. The unit along L was chosen as $c = 2c'$. The observed diffuse lines are no wider than the Bragg reflections which allow the collapse of the diffuse intensities into

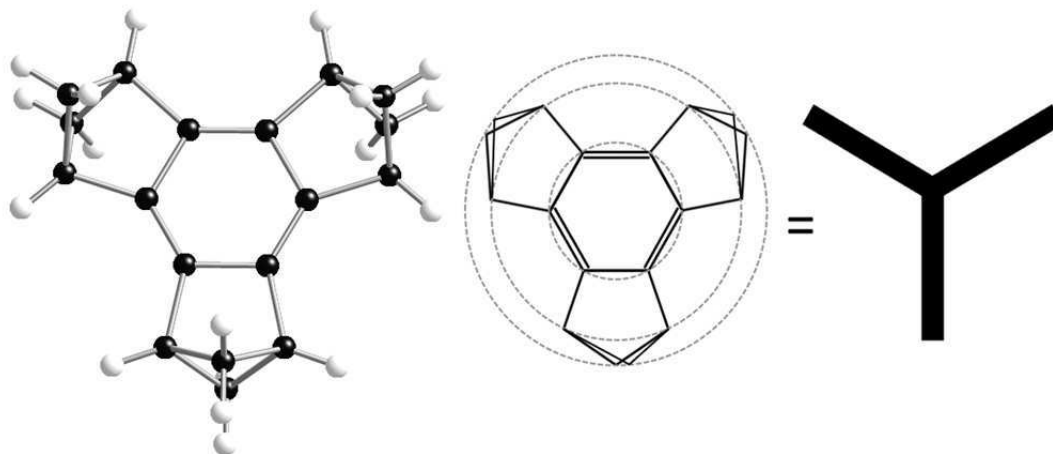


Figure 5.1: Left: Tris(bicyclo[2.1.1]hexeno)benzene molecule (TBHB), black atoms are carbon, white atoms are hydrogen. Center: A skeletal formula representation of TBHB where all atoms on a given dotted circle were assigned the same isotropic atomic displacement parameter (U_1, U_2, U_3). Right: schematic tri-star representation of TBHB (used in Fig. 5.2).

a 1-d profile. A total of 14 $h k L$ lines were extracted by defining a rectangular box over the full length L of the diffuse intensity rod. The box is one pixel deep in l and includes the full extent of diffuse intensity in h . At each value of L, h and k intensities are summed and corrected for background, thus resulting in a line profile along L . The process is described in detail elsewhere [78]. The disorder has been described with growth modeling, a procedure in which a new layer is added onto the preceding layers of a crystal. Addition in different positions is associated with different probabilities. The probability of each added layer depends on the arrangement of the preceding layers [78], four of them in the present case. Selected growth sequences labeled with a shorthand and the symbols of the associated stacking probabilities are shown in Figure 5.2.

The symbols of the shorthand refer to three layers: the symbol e (for eclipsed) implies that layer $n + 2$ sits exactly on top of layer n ; $b_L(b_R)$ means that layers n (lowest), $n + 1$, and $n + 2$ (highest) spiral in a clockwise (anti-clockwise) fashion when looking onto the growing crystal face. Figure 5.2 uses this nomenclature to describe transitions from four- to five-layer stacking sequences. The full Markov matrix of

Table 5.1: The transition matrix \mathbf{T} of probabilities for extending the left-hand column of four-layer motifs into the top row of new four-layer motifs. Once a new motif is formed by adding a new fifth layer, the first layer and thus the first motif are dropped. The meaning of the symbols is described in the text.

from \ to	... eb_L	... eb_R	... ee	... b_Lb_R	... b_Lb_L	... b_Le	... b_Rb_L	... b_Rb_R	... b_Re
eb_L	0	0	0	$t - \frac{\Delta}{2}$	$c - \frac{\Delta}{2}$	$e_1 + \Delta$	0	0	0
eb_R	0	0	0	0	0	0	$t - \frac{\Delta}{2}$	$c - \frac{\Delta}{2}$	$e_1 + \Delta$
ee	$\frac{1-e_2}{2}$	$\frac{1-e_2}{2}$	e_2	0	0	0	0	0	0
b_Lb_R	0	0	0	0	0	0	$t - \Delta$	$c + \Delta$	e_1
b_Lb_L	0	0	0	$t + \Delta$	$c - \Delta$	e_1	0	0	0
b_Le	$\frac{1-e_2}{2} - \Delta$	$\frac{1-e_2}{2} + \Delta$	e_2	0	0	0	0	0	0
b_Rb_L	0	0	0	$t - \Delta$	$c + \Delta$	e_1	0	0	0
b_Rb_R	0	0	0	0	0	0	$t + \Delta$	$c - \Delta$	e_1
b_Re	$\frac{1-e_2}{2} + \Delta$	$\frac{1-e_2}{2} - \Delta$	e_2	0	0	0	0	0	0

transition probabilities is given in Table 5.1. Adding a new layer on the right-hand side of the column vector “from” generates the sequence in the top row vector “to” of the transition matrix; in short $\mathbf{p}' = \mathbf{p}\mathbf{T}$, where \mathbf{p} and \mathbf{p}' are row vectors describing the probabilities of finding a given four-layer sequence before and a five-layer sequence after adding a new layer, respectively; \mathbf{T} is the matrix of transition probabilities. For the sequences b_R and b_L , the molecules in layer $n + 1$ are tilted out of the trigonal plane, but not for the sequence e . Chemically equivalent atoms are assigned the same isotropic mean-square displacement parameter (Figure 5.1).

Nearly noise-free diffuse intensity data were obtained from 1280 clones, generated with a disorder model derived from an experimental study of a crystal of TBHB [79]. The parameters of the model used in this work are the stacking probabilities, a molecular tilt angle, and isotropic atomic displacement parameters. A total of 1280 virtual model crystals (clones), each consisting of $0.96 \cdot 10^5$ layers, were grown from a single set of parameters that best fit the 14 experimentally determined hkL lines [79](Table 5.2). Each clone was divided into 1600 randomly chosen lots [15, 44] encompassing 60 layers. Diffuse intensities were calculated by Fourier transformation of each lot. The calculated $2.048 \cdot 10^6$ sets of intensities were then incoherently averaged to create a reference data set of 14 hkL lines including $0kL$ ($k = 1, 2, 4, 5, 7$),

$1kL$ ($k = 3, 5, 6$), $2kL$ ($k = 3, 4, 6$), $3kL$ ($k = 4, 5$), and $0 < L < 5$ for all lines. As an example the reference and optimized model intensities of the 01L line are compared in Figure 5.3.

5.3 Computations

5.3.1 Methods for optimizing model parameters

In general, the initial values of the parameters chosen for modeling disorder are “educated guesses” at best, usually far from their real value. Therefore a global optimization technique that is not based on sophisticated *a priori* knowledge but is able to optimize sets of random initial model parameters is needed. Population-based, metaheuristic algorithms are well suited for the purpose of optimization without preliminary assumptions of the solution.

We selected three representative algorithms for numerical optimization: a Genetic Algorithm (GA), a Differential Evolution (DE), and a Particle Swarm Optimization (PSO). GA and DE are population-based search algorithms that implement principles of genetics. Each individual gene set k in a population is evaluated according to its fitness R_k (high fitness = low R_k).

$$R_k = \sqrt{\frac{\sum_i [(\sum_j (\frac{I_{i,k}^j}{J})) - I_{i,ref}]^2 \cdot w_i}{\sum_i (I_{i,ref})^2 \cdot w_i}}, \quad (5.1)$$

where the sum over i includes all I data points $I_{i,k}^j$ from 14 diffuse lines ($I = 301 \cdot 14 = 4214$; weight $w_i = 1$), and the sum over j includes all J clones. All intensities are given unit weight in the calculations.

The resulting population of R-values, R_k (corresponding to K model parameter sets),

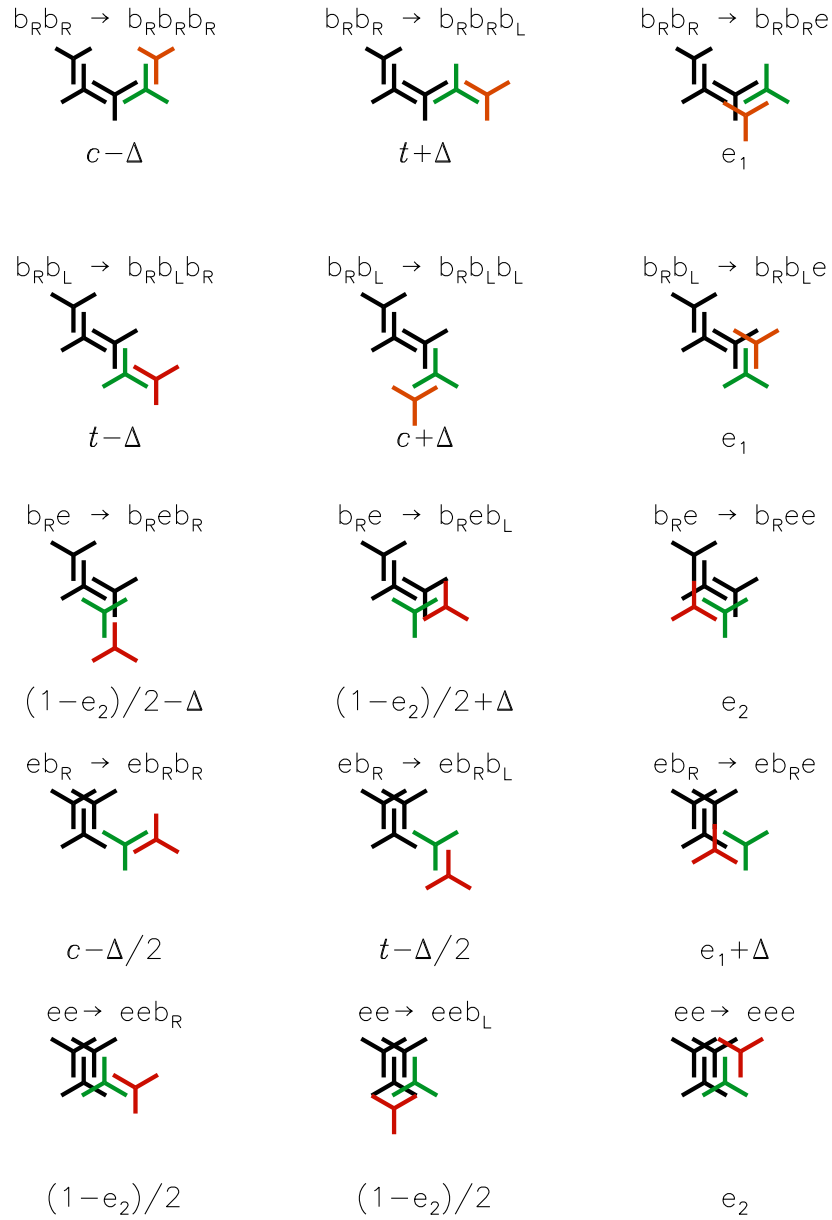


Figure 5.2: Examples of unique five-layer stackings and associated motifs: layers 1,2,3 in solid black, layer 4 (green) and layer 5 (red); respective transition probabilities from four- to five-layer stackings at the bottom of the motif. The symbols above the motif describe the four- and five-layer stacks. The symbols b_L , b_R , e (bent left, bent right, eclipsed) characterize the three possible three layer stacks. A four-layer stack is defined by two symbols, and a five layer stack by three symbols.

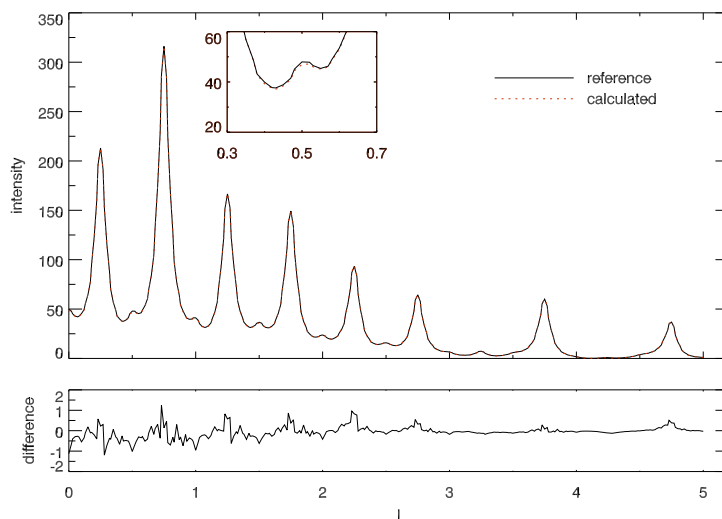


Figure 5.3: Reference intensity and calculated intensity (averaged over 40 clones) for the 01L line are shown overlaid. The difference (calculated - reference) is shown in the plot below. (see Table 7.7, $\bar{R} = 0.0077$)

is characterized by its mean and standard deviation.

$$\bar{R} = \sum_k \frac{R_k}{K}; \quad s = \sqrt{\frac{\sum_k (R_k - \bar{R})^2}{K - 1}}. \quad (5.2)$$

The individuals yielding the lower R_k values in a comparison between parents and children survive and form the parents for the next generation. This process is repeated until a stopping criterion is reached, in our case a set number of generations. Population convergence to a solution is signaled by a low overall \bar{R} accompanied by a low s value, which remains essentially constant over many generations.

Like GA and DE, PSO is a population-based, stochastic search technique; however, it does not use genetic operators. Instead the position and movement of each particle in a swarm is adjusted with respect to the overall trend of velocity and direction for the swarm. Convergence to a solution is achieved when all particles have a small displacement and have clustered together.

Genetic algorithm

The genetic algorithm (GA) is a widely used evolutionary algorithm and is described in detail in Refs. [76, 80, 81, 82, 83, 84]. The initial generation is randomly generated within a set range for each parameter (Table 5.2). Cycling through crossover, mutation, and selection sequentially creates the subsequent generations. The control parameters of the algorithm are crossover rate, and mutation rate (Table 5.3). During a GA optimization, new individuals are generated from two randomly chosen individuals (genotypes) of a generation, crossover is applied by recombining the parameter vectors at a random point, and then a mutation is applied by randomly selecting and changing parameters. Since GA uses a bit-wise representation of the parameters (genes) during numerical optimization, so-called Hamming cliffs occur when flipping a randomly chosen bit in the binary representation. This may change parameter values drastically. In order to avoid Hamming cliffs, the genes are represented by so-called ‘Gray codes’ [85]. Once the genetic procedure for the population is complete, the fitness of the individuals in the new generation and the parent generation are compared with the objective function R_k . The individual with the better fitness, either the parent or the new candidate, survives to serve as parent for the next generation. This process is repeated until a stopping criterion is reached, in our case a maximum number of generations (Table 5.3).

Differential evolution

Differential Evolution (DE) is a vector-based method successfully used for numerical optimization problems and problems that are parameterized with real numbers [45]. Application of DE to disorder modeling and the interpretation of diffuse scattering has been described in detail by Weber and Bürgi [86].

DE forms a child, an individual of the subsequent generation, by picking genes from a target individual (t) with gene vector \mathbf{d}_t in the parent generation and from a mate \mathbf{d}'_c created from three randomly chosen parent individuals a , b , and c . The three

Table 5.2: Comparison of the reference parameters defining the reference data with mean model parameters and their standard deviations obtained by optimizations with a genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) using 40 individuals. The parameter values at the start of the optimizations are generated for each member in the population using the range of values listed in the third column for each parameter. The minimum and maximum values are the world size or absolute limits of the parameter values during optimization.

par.	reference	min.,max.	GA	DE	PSO
c	0.48877	0, 0.5	0.49697(3)	0.4885(4)	0.488(2)
Δ^a	0.49336	-0.5, 0.5	0.45(1)	0.484(6)	0.486(7)
e_2	0.006748	0, 1.0	0.7372(6)	0.008(4)	0.02(3)
$tilt^b$	2.2723	-5.0, 5.0	2.251(8)	2.27(1)	2.3(1)
U_1^c	2.6284	0, 5.0	2.67(1)	2.63(3)	2.7(1)
U_2^c	2.2734	0, 5.0	2.026(5)	2.28(4)	2.2(2)
U_3^c	3.0005	0, 5.0	2.982(9)	2.97(6)	3.0(2)
t^a	0.48877	0, 0.5	0.49697(3)	0.4885(4)	0.488(2)
e_1^a	0.022452	0, 1.0	0.0061(7)	0.0230(8)	0.02(3)
R^d	-	-	0.0338(3)	0.0090(4)	0.02(1)

^aConstrained parameters: $c = t, (c \pm \Delta) + (t \mp \Delta) + e_1 = 1$

^b units of $tilt$ in degrees

^c units of U_i in 10^{-2}\AA^2

^d R-factor measures fitness after 150 generations

Table 5.3: The control and run parameters for DE, GA, and PSO.

Algorithm Control Parameters		
GA	DE	PSO
mutation rate = 0.005	$f_m = 0.70$	$w = 0.95$
crossover = 0.95	$f_r = 0.50$	$c_1 = 1.0$
		$c_2 = 1.0$
		$V_{max} = 0.25 \cdot (\text{limits}^a)$
Run Parameters		
num. generations ^b		150
population size ^c		40
num. of clones ^d		10 – 40

^a see Table 5.2 min. and max.

^b user specified

^c individuals (genotypes) included in the population or swarm

^d copies of each genotype included in the calculation

vectors \mathbf{d}_a , \mathbf{d}_b , and \mathbf{d}_c are combined to create \mathbf{d}'_c , according to $\mathbf{d}'_c = \mathbf{d}_c + f_m(\mathbf{d}_a - \mathbf{d}_b)$, where f_m is a scalar mutation constant, a control parameter of the algorithm. If any gene of a mate \mathbf{d}'_c is outside the set search range, the mate is rejected and a new mate calculated. To create the child, one randomly selected parameter (gene) in t is replaced by the corresponding gene from \mathbf{d}'_c , and the remaining genes of the child are inherited from \mathbf{d}'_c with a probability given by the crossover constant f_r , another control parameter of the algorithm. The control parameters used for DE are listed in Table 5.3.

The survival of either the target individual or the child to the next generation is determined by which of the two has the higher fitness. DE is repeated until a stopping criterion is reached, here a maximum number of generations (Table 5.3).

Particle swarm optimization

Particle Swarm Optimization (PSO) is modeled after the behavior of swarms such as birds or insects in nature [77]. A vector \mathbf{x}_i of parameters defines each individual or particle i within the swarm. Each particle is guided to the optimal solution by

the best solution it has seen plus the best solution seen by the population. The initial population for PSO is randomly generated as for the GA and DE algorithms. Unlike GA and DE, PSO does not operate on the principles of genetics; instead, each particle or candidate solution \mathbf{x}_i is assigned a displacement per unit of time (t), generally referred to as velocity \mathbf{v}_i , by which the particle travels the search space. Each variable in \mathbf{x}_i is updated from one generation to the next with \mathbf{v}'_i according to $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{v}'_i \cdot t$. The velocity \mathbf{v}'_i modifying the current parameter vector \mathbf{x}_i is influenced by the best solutions seen by the particle, \mathbf{b}_i , and the population, \mathbf{b}_P . It is updated according to

$$\mathbf{v}'_i = w \cdot \mathbf{v}_i + c_1 r_1 (\mathbf{b}_P - \mathbf{x}_i) + c_2 r_2 (\mathbf{b}_i - \mathbf{x}_i). \quad (5.3)$$

The algorithm depends on the following control parameters:

- inertia parameter w (generally < 1);
- acceleration constants c_1 and c_2 (indicating how much the particle vector \mathbf{x}_i is directed toward the best solution that is seen by the swarm, \mathbf{b}_P , and the particle, \mathbf{b}_i , respectively);
- random numbers r_1 and r_2 generated within the range $(-V_{max}, V_{max})$.

The control parameters for PSO used in this work are listed in Table 5.3. Any parameter of \mathbf{x}_i that is outside of the search space is reset to its limit.

5.3.2 Clones and parallelization

In previous DE optimizations it was observed that the fitness of certain individuals (intensities) was so high that their genes survived many generations. For reasons of computational efficiency, the disordered crystals, their intensities, and the fitness of such individuals were not recalculated in subsequent generations. On calculating many individuals and their fitness with the same gene set, it was found that the fitness values covered a distribution of R-values. The fitness of individuals surviving many generations was invariably found at the high fitness-end of such distributions. Conversely, the fitness of individuals whose gene sets were frequently replaced was

often found at the low fitness-end even though the average fitness of the distribution might have been quite reasonable. These two phenomena lead to the conclusion that a reliable estimate of fitness requires multiple intensity calculations with the same gene set, subsequent averaging of the intensities, and calculating a fitness from the averaged intensities [87]. Crystals originating from the same parameter or gene set are referred to as clones.

The phenomenon of ultra-stable individuals may be understood in terms of the physical background of a diffuse scattering experiment. Coherent scattering of an object lacking translational symmetry results in a speckle pattern. Locally the scattering intensity in such a pattern may change rapidly. Small differences in the object produce slightly different speckle patterns. The coherence length of X-rays used in diffuse scattering experiments on disordered crystals is typically smaller than the sample size. The experimentally observed signal is thus an incoherent superposition of different speckle patterns originating from slightly offset regions within the sample. Apart from experimental noise, such signals usually look quite smooth. The phenomenon of incoherent superposition is simulated in our disorder modeling with clones, albeit at a much smaller scale. While the real sample may contain on the order of 10^{18} (slightly differing) unit cells, a typical clone consists of a mere $10^4 - 10^6$ unit cells which are divided up into lots of dimensions that are chosen to match the correlation length defining the short range order [74, 88]. The lots are Fourier transformed and averaged incoherently. If the variations of the resulting averaged, simulated pattern, also called MC noise, are of the same order as the experimental noise, MC noise may or may not match the experimental noise, thus giving the false impression of unusually high or unusually low fitness of the model. Simulating diffuse scattering patterns with model crystals of inadequate size and insufficient numbers of lots hence runs the risk of mistaking noise in the experimental pattern as being the result of disorder, thus explaining the phenomenon of the unjustified survival of some of the model crystals as described above. To

reduce this risk, the volume of simulated crystals must be large enough and thus the calculated diffuse pattern must be smooth enough to minimize bias due to MC noise.

Random number bias is another problem. Disorder modeling usually starts from a randomly seeded crystal that is grown into a full-sized model crystal or equilibrated by using a MC process. Such crystals may be biased by the starting configuration. Building several crystals, each starting from a different random seed, minimizes the risk of random number bias and reduces the probability of ultra-stable individuals.

A disadvantage of clones is the increase of computational cost and a corresponding slow-down of the structure determination process. The latter can be compensated effectively by parallel computation of the clones, one per processor of a supercomputer or a grid computing facility, as will be discussed in section 7.2.2. In the present case of nearly noise-free data, the use of clones serves to analyse and control the inherent dispersion of results characteristic for crystal growth and MC models (Section 7.2.7).

There is an additional dimension to parallelization. The global optimization methods discussed above explore parameter space by calculating individuals and their clones with many different gene sets or swarm particles. Thus, the computation of J clones for K gene sets is easily distributed over $J \cdot K$ compute nodes. The efficiency of parallelization is limited only by the amount of communication necessary between the nodes. In the present case this corresponds essentially to the transfer and averaging of the J clone intensities for each of the K individuals. Unless specifically mentioned, optimizations were performed with 40 gene sets, and fitness was calculated from averaging over 20 clones.

Computing resources were provided for the project by the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory (ORNL) and the US National Science Foundation's TeraGrid cyber infrastructure project. The Oak Ridge Institutional Cluster (OIC) at ORNL is a combined 3136 core shared cluster with a 29Tera-flop peak performance. This work was run on the SNS data analysis share of the OIC consisting of unrestricted parallel use of up to 192 cores, grouped in clusters of 8 cores per node. The Extreme Science and Engineering Digital Environment

(XSEDE), previously TeraGrid, is a national shared cluster for US NSF users. It encompasses over 20 different computational resources with over 2686 Tera-flops of combined performance. An allocation of 195,000 core hours with access to 5 different supercomputers within TeraGrid was granted and used for this work.

Chapter 6

Local structure of analysis of

β [beta]-NaLaF₄

6.1 Introduction

In this chapter the general computational modeling protocol described in chapter 5 for quantitative modeling is applied using ZODS to analyze the diffuse scattering of NaLaF₄, a highly efficient light upconversion material (chapter 4). A detailed quantitative description of the local structure of the spectroscopic sites is essential in order to understand the mechanism of the energy-transfer processes which underpin the high light upconversion efficiency of this family of materials [4, 9].

Aebischer et al. performed a polarized single crystal absorption spectra of the highly ordered LaCl₃:Er³⁺ and β -NaGdF₄:Er³⁺ (both of structure type UCl₃). The study revealed a doubling of the Er³⁺ absorption lines for in β -NaGdF₄ relative to the ordered LaCl₃. The allowed transitions indicated by the C_{3h} selection rules are violated in the β -NaGdF₄:Er³⁺ spectra indicating a lower site symmetry of the Ln³⁺ ion than in the single crystal average structure. This discrepancy is resolved by examining the local ordering options of the structure.

The variations in the local structure show in the average structure as partial occupancies of the Na2/La2 and Na1/void positions (figure 6.1(a)) and as an elongated anisotropic displacement parameter in the c direction for the F1 and Na1 atoms (figure 6.1(b)). Translational order is preserved within a column, meaning that the Na2...La2 or La2...Na2 strictly alternate in each column [4]. The third column is fully occupied by La1 and is centered along the column between the Na2/La2 and Na1/void positions. However, the three neighboring Na2/La2 columns surrounding the La1 column do not alternate in phase. This produces the “honeycomb” diffuse 2-D diffraction pattern at half integer L as shown in the background of figure 6.1(a). Based on previous simulations it was found that La2...Na2 prefers three negatively correlated neighbors of Na2...La2, which cannot be fully satisfied resulting in occupational disorder around the ordered La1 column [4]. This occupational disorder implies that the alternation (Na2/La2 or La2/Na2) and correlation of the cation columns causes the F1 atoms to be shifted out of plane breaking the C_{3h} symmetry of the La1 column, due to the greater charge attraction of the F^- to La^{3+} than Na^+ ; thereby providing a structural basis for resolving the discrepancy between the average single crystal structure and polarized single crystal absorption [4].

In this work a quantitative description of two aspects of the structural disorder is provided; (1) the shift of the planar F atoms coordinated to the ordered high symmetry spectroscopic site and (2) the neighbor correlations of the Na2/La2 columns.

6.2 Super cell

Modeling the diffuse scattering of any compound begins with an average structure refined from the Bragg intensities only. The Bragg refinement is the best possible time and space averaged real structure. Detailed analysis of the average structure gives indicators of the structural disorder not represented in the Bragg intensities.

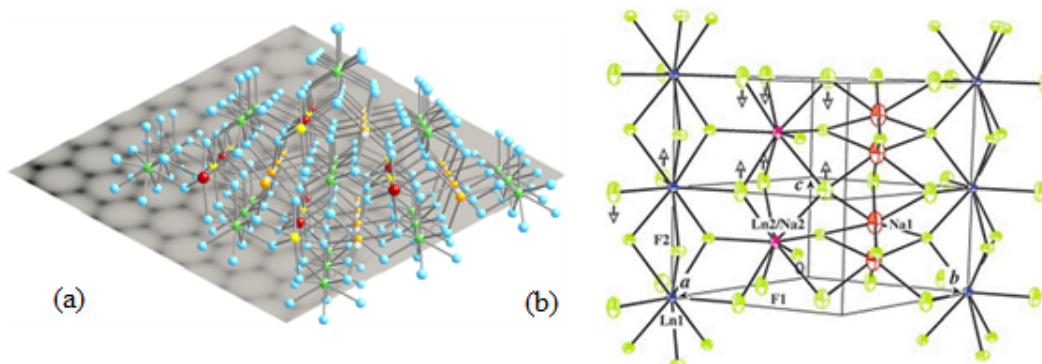


Figure 6.1: The super cell of β -NaLaF₄ illustrates occupational structural disorder by showing the neighbor preferences of the Na2 ... La2 and the Na1 ... void columns. The central green La column is fully ordered and the three disordered Na(yellow)/La(red) columns surrounding are not all oriented the same way (negative correlation). The Na/La columns prefer La/Na columns as neighbors. The Na (orange)/void (open blue dashed circles) are also not all oriented the same way. These neighbor preferences are impossible to satisfy over the whole 2-D range and lead to the diffuse “honeycomb” diffraction pattern (shown behind the direct space structure) at $(H, K, 0.5L)$. Figure from ACS COMP image award [6]. (b) Average structure of β -NaLnF₄ (Ln=Y, La Lu) showing the three different columns of metal sites along **c**: Na2/La2 is a 1:1 mixture of Na2 and Ln2 (pink), the Ln1 column (blue), both with C_{3h} symmetric, tricapped trigonal prismatic coordination geometry, and the half occupied Na1 site (red) with C_3 symmetric, distorted octahedral coordination geometry. Local distortions owing to the disorder in the crystal are indicated by arrows. Figure taken from [4].

A Bragg refinement of NaLnF_4 yields an average structure with space group $\text{P}\bar{6}$ and contains the following symmetry equivalent positions: $(x, y, z), (-y, x-y, z), (-x+y, -x, z), (x, y, -z), (-y, x-y, -z), (-x+y, -x, -z)$. The average structure of NaLnF_4 contains columns of half occupied Na2/La2 cations at $(2/3, 1/3, 1/2)$, half occupied Na1/void at $(1/3, 2/3, 0.3365)$, fully occupied La1 at $(0, 0, 0)$, fully occupied F1 and F2 at $(0.62900, 0.03670, 0.0)$ and $(0.73220, 0.75310, 1/2)$, respectively. The mean square displacement amplitudes of F1 contain a U_{33} , $(0.00483(24))$ that is almost twice as large as the next largest mean square displacement parameter U_{22} (0.002804) . The elongated ADPs in the U_{33} direction correspond to a positional disorder of F1 along \mathbf{c} . F1 was refined as a single rather than a split position yielding the elongated ADP for each of its respective positions. It is possible to refine F1 each with two half occupied positions. This would result in mean square displacement amplitudes with more similar U_{ii} s for each of the two partially occupied F1 positions. The partial occupancies and elongated ADPs are indicators of occupational and positional disorder, respectively.

Extraction of the full real structure requires, in addition to the Bragg refinement, modeling the diffuse intensities. The first step in constructing an appropriate disorder model is to modify its average structure CIF so that the unit cell is redefined as a “super cell”, which contains all structural alternatives found in the Bragg refinement. The modifications necessary to define the super cell of NaLnF_4 are based on the experimental observations of the diffuse intensities that can be ascribed to elements of the disorder in the average structure CIF.

Aebischer et. al [4] conclude based on the sharpness of the honeycomb layers along \mathbf{c}^* there is translational order along \mathbf{c} , which implies that the Na2 and La2 must be ordered in the column along \mathbf{c} meaning that they strictly alternate Na2...La2 or La2...Na2. However, the Na2...La2 in its three neighboring unit cells does not necessarily alternate in the same fashion. Based on simulations of $100 \times 100 \times 1$ unit cells Aebischer et. al [4] found that La2...Na2 prefers neighbors of Na2...La2 (negative correlation). However, it is impossible for all neighbors of any given column

to alternate in opposite phase relative to one another over the whole 2-D range. This aspect of the local structure creates the continuous diffuse honeycomb pattern.

In order to describe the disordered part of the structure, the Na2/La2 columns are used to recreate the structural disorder in **a,b** by modeling the possible neighbors of Na2/La2. The average structure CIF lists Na2, La2 as the same position with occupancy of 0.5 and Na1 position with occupancy of 0.5. Based on the alternation of these atoms Na2...La2 and Na1...void along **c** then doubling the length of **c** would give fully occupied positions for each atom. For example if Na2 is at the average position in the CIF file of $(2/3, 1/3, 1/2)$ and **c** is doubled, then this position becomes $(2/3, 1/3, 1/4)$ and La2 is then at $(2/3, 1/3, 3/4)$ (figure 6.1(b)). This means that doubling the unit cell in the **c** direction adds extra symmetry operations. The new “super cell” CIF file should have the adjusted atomic positions (*z* coordinate is divided by 2) for the doubled **c** and the new symmetry operations for the $z + 1/2$ shift are; $[(x, y, z + 1/2), (-y, x - y, z + 1/2), (-x + y, -x, z + 1/2), (-y, x - y, -z + 1/2), (-x + y, -x, -z + 1/2)]$.

6.3 Disentanglement

The new super cell contains the fully occupied positions for Na2, La2 and Na1. This disentanglement of atomic positions allows the modeling of structural disorder composed of complete structural units each representing a local ordering option called a chemical unit (CU). These CUs are then used to construct the disordered **a,b** plane of the crystal.

To model the occupational disorder of Na2/La2 the Na2...La2 columns, have to be specified. The chemical unit is defined using the same atom labels as the CIF file. This CU is specified using the positions in the CIF file, (X, Y, Z) for Na2 and a La2 at $(X, Y, Z + 1/2)$. This defines a La2/Na2 column.

All possible orientations of the chemical unit need to be described. The orientations of the CU that are not symmetrically equivalent are called “sets of

alternatives” (soas) and are defined in the ZODS input file. In the case of modeling the occupational disorder of Na2/La2, the possibilities are La2...Na2 (defined as the original CU) or Na2...La2. The alternative is defined in the input using a mirror operation on the original CU as CU1, $(X, Y, -Z + 1)$. Since each orientation of the CU is equally likely the probability is set to 0.5.

The next local structure element is the F1 out-of-plane shift. The U_{33} element of the F1 ADP was nearly twice as large as any other U_{ii} component, an indication of positional disorder along \mathbf{c} . This corresponds to a shift in the \mathbf{z} direction in the direct space structure. The previous work of [4] estimated a 0.07\AA shift toward the La^{3+} in direct space, although no quantitative modeling of this shift has been done.

To model this positional disorder the CU needs to contain the F1 atoms with the modified z coordinate. The estimated 0.07\AA shift in z position of F1 is converted to the hexagonal fractional coordinates of the CIF file (see 3.1). Since only the z coordinate is changed, converting from hexagonal fractional coordinates means dividing the z value of 0.07\AA by $|\mathbf{c}|$ (7.667\AA) and corresponds to $+0.00923$ shift of the z coordinate in fractional coordinates. This shifts F1 from its special position at $z = 0$ and produces a split position when the $\bar{6}$ symmetry operation is applied. The value of this shift will be refined and the proposed direction of the shift toward La^{3+} and away from Na^+ will be tested during optimization.

The CU incorporates the F1 positional disorder, incorporating the F1 atoms that are coordinated to La2 and Na2. The CU is the repeatable structural unit along the \mathbf{c} axis such that Na2 and La2 strictly alternate. There are six F1 atoms, three above and three below the La2. The CU is repeated within the column so it is only necessary to define either the three F1 atoms above or below the La2. This gives the three F1 atom positions as $(X, Y, Z + 1/2)$, $(-Y, X - Y, Z + 1/2)$ and $(-X + Y, -X, Z + 1/2)$ (figure 6.2).

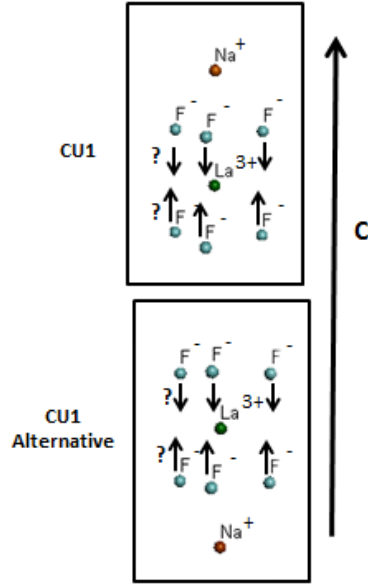


Figure 6.2: CU1 shown (left) with the alternative (right). Either CU1 or its alternative is repeated along c to produce the F1 coordinated to Na2 and La2 in the 2-D crystal. The question marks indicate the direction of the F1 shift which will be tested in the optimization.

6.4 Model parameterization

The interaction between the CU and the out-of-plane shift of F1 are parameterized and used by the ZODS MC crystal builder to construct model crystals. The interaction between CUs is defined as a discrete, Ising type interaction and defined between a CU at one site in the crystal with a CU at another site.

In the Na2/La2 column model only non-symmetry equivalent first neighbor interactions are parameterized. For the Na2/La2 neighbor interaction it is only necessary to define one parameter ($p1$) as the interaction between CU1 and CU1. Since MC operates on relative energy differences in the parameters to build and equilibrate the crystal, not defining the alternative interaction (CU1 with its soa) is equivalent to setting the interaction energy to zero. In the construction of this model the interactions of the CUs were defined so that chemical units of opposite orientations preferred to be neighbors. This assumption was based on the negative

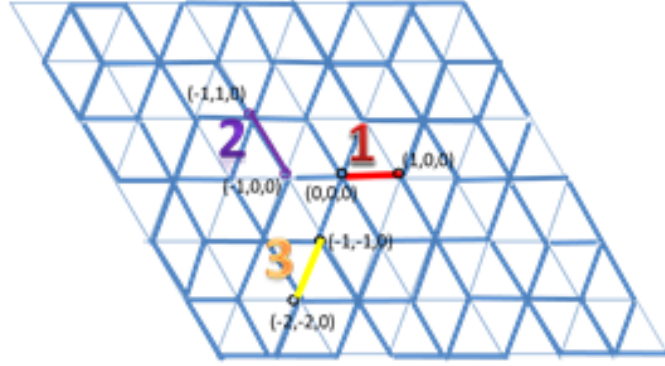


Figure 6.3: Two dimensional illustration of the symmetric CU first neighbor interactions defined in the model. The $-z$ elements are at $z = -1/2$ below the plane shown.

correlation calculated in the simulation of the Na₂/La₂ columns done in [4]. This assumption is tested by adjusting the limits of the search space for global optimization which allow the model to build crystals with positive correlation and calculate their fitness.

The parameter p_1 is defined as the interaction of CU1 with another CU in the same orientation. If this parameter is independent then the interaction of the CU with its alternative is defined as a dependent parameter. The model parameters are randomly generated from defined search limits.

In order to check that the model has the correct symmetry, ZODS generates a file called model.xml which lists all of the symmetry equivalent interactions for the model parameters (figure 6.3). This file should be checked for the correct symmetry and interactions. The NaLnF₄ model parameter p_1 shows the expected $\bar{6}$ symmetry. The interactions are between CUs in neighboring unit cells. The first interaction is between a CU at (X, Y, Z) and another CU at $(X + 1, Y, Z)$. The second interaction is described between CUs at $(X - 1, Y, Z)$ and $(X - 1, Y + 1, Z)$ and the third interaction is between CUs at $(X - 1, Y - 1, Z)$ and $(X - 2, Y - 2, Z)$. The same interactions are listed in model.xml for the $-Z$ component of these three positions but not shown in figure 6.3.

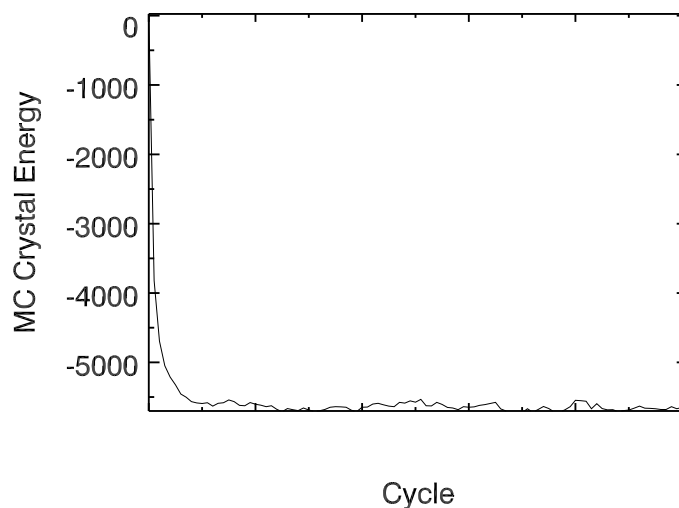


Figure 6.4: Plot of the crystal energy as a function of MC cycles. After 100 cycles it has equilibrated to about -5650 in crystal energy.

The z shift of the F1 atoms is an element of the average structure CIF and is not an Ising parameter. It is a structural parameter and is defined under the `average_structure` node in the XML input file (see ZODS input A. 4.1).

6.5 Crystal building and model refinement

The 2-D crystal is constructed from the chemical units using the model parameters that define the interactions between them. Crystals of sufficient size (10,000 unit cells) are built from the disorder model parameters and their energy equilibrated using MC (described in chapter 3 section 3.2.2). The crystal energy can be visualized as a function of MC cycles using the ZODS plot tools (Appendix A. 4.2). The crystal energy as a function of the number of cycles for the 2-D model crystal containing CUs modeling the F1 shift and the Na2/La2 column correlation is shown in figure 6.4. The number of MC cycles was set to 100. The crystal energy is between -5688 and -5649.93 in crystal energy for the last 10 cycles.

The model parameters are then optimized through a generational algorithm called differential evolution (DE) [45] that refines the model parameters to best fit the experimental diffuse intensities (section 3.2.3). Multiple crystals are built with the same parameter set (clones, see chapter 5 section 5.3.2) to sample a greater representative area of the fitness distribution of crystals. The model crystal intensities are calculated by incoherently summing over lots and averaging the individual crystal intensities over clones. Optimization continues until convergence is achieved with a specified stopping criterion, such as the number of optimization cycles, 36 in this case. Crystals of 100×100 unit cells and 25 lots of size 20×20 and 48 gene sets with 20 clones each in the calculations (see ZODS input in A. 4.1).

The model fitness is based on comparison of the calculated intensities to the measured intensities. This is done in a systematic way (on a grid) for each layer of the 2-D diffuse “honey comb” (as described in 3.1). The diffuse X-Ray intensities for $HK0.5L$, $HK1.5L$, $HK2.5L$, $HK3.5L$ and $HK4.5L$ layers are used in these calculations. The step size is 0.004 and the step directions were converted to Cartesian coordinates from fractional coordinates on a $\gamma = 120^\circ$ lattice. Since \mathbf{a} is collinear with \mathbf{x} , the resulting transformation (from the transformation matrix in section 3.2.1) to Cartesian coordinates in direct space is given by;

$$\mathbf{a} = [a, 0, 0] \tag{6.1}$$

$$\mathbf{b} = [b \cos \gamma, b \sin \gamma, 0] \tag{6.2}$$

Now, the reciprocal space direction \mathbf{x} is along \mathbf{a}^* (for this data) and normalized is $\frac{\mathbf{a}^*}{|\mathbf{a}^*|} = \frac{1}{2}[\sqrt{3}, 1, 0]$ and along \mathbf{y} is $\frac{1}{2}[-1, \sqrt{3}, 0]$. Note in chapter 3 the ZODS convention defines \mathbf{x} to be along \mathbf{a} but in this case the data is defined with \mathbf{x} along \mathbf{a}^* . The intensities are calculated on each 2-D layer so the z coordinate is fixed at 0. The step size is multiplied by \mathbf{x} and \mathbf{y} to give the point on the grid to calculate the intensity and compare to the measured data. In this example \mathbf{x} is the \mathbf{d}_1 direction and \mathbf{y} is

in the \mathbf{d}_2 direction. The points where intensity is calculated must be commensurate with the measured diffuse intensity data points.

Chapter 7

Results

7.1 β [beta]-NaLaF₄ neutron data analysis results

The following are the preliminary results of analysis of the 100K NaLaF₄ X-Ray and Neutron data.

7.1.1 TOF data integration and refinement comparison

The neutron Bragg data were integrated using both spherical and ellipsoidal integration methods and compared systematically with increasing integration radius. The peak radius is given in table 7.1 where the inside radius is 0.002+ peak radius and outside radius is peak radius +0.02. The region radius for the ellipsoidal integration is 0.20 for all radii tested.

The data were integrated the absorption and spectrum corrections were applied and *hkl* file generated using Anvred [72]. The structure was refined using GSAS [34] and EXPGUI [35] (user friendly GUI environment) for each radius starting at 0.13 to 0.18 Å⁻¹. The refinement figure of merit (R -factor) for single crystal data refinement is given in GSAS as;

$$R = \frac{\sum |F_o^2 - SF_{calc}^2|}{\sum |F_o^2|} \quad (7.1)$$

Table 7.1: The fitness values and number of reflections for each integration radius are given for spherical and elliptical integration methods all refinements were done using GSAS. The d-spacing is 0.5.

Spherical Integration			
Int. Radius	$R(F_o)^a$	$R(F_o)^b$	# Observations
0.13	0.058	0.053	1856
0.14	0.058	0.053	1799
0.15	0.059	0.055	1756
0.16	0.061	0.055	1705
0.17	0.062	0.055	1659
0.18	0.063	0.055	1475
Elliptical Integration			
Int. Radius	$R(F_o)^a$	$R(F_o)^b$	# Observations
0.13	0.061	0.54	2196
0.14	0.060	0.54	2191
0.15	0.058	0.54	2179
0.16	0.058	0.052	2181
0.17	0.060	0.052	2171
0.18	0.062	0.054	2164

^a for $I > 0$

^b for $I > 3\sigma(I)$

where S is the scale factor. The fitness values $R(F_o)$ remained essentially equivalent, within 1% , over the whole radii range. The fitness of the spherical and elliptical integration methods were very similar (approximately 5%), however refinements using the spherical integration method with radius 0.15 \AA^{-1} yielded ADPs that were in best agreement with the X-Ray refinement (see section 7.1.2). The number of observed reflections generally decreases with increasing integration radii, however the decrease is not substantial and the number of observed reflections is sufficient to yield good statistics relative to the number of refinement parameters (greater than 10 times the number of parameters), 20 in this model. If the integration radius is too small to encompass the entire peak the intensity is underestimated; if the integration radius is too large more peaks are outside of the detector region and rejected leading to the rejection of quality data. The choice of appropriate integration method and radius is important for the final quality of the structure. After integration reflections that were most disagreeable:

$$\frac{|F_o^2 - F_{calc}^2|}{\sigma(F_o^2)} \geq 10$$

were removed; about (30–40) reflections in each case. The refinement results in terms of R value between the two methods are virtually equivalent. The best choice of radii seems to be between 0.15 and 0.17.

Different d-spacings (between 0.4–0.7 in increments of 0.1) were compared for both methods to determine the optimal spacing for the TOF neutron Bragg data. In general, there is an inverse relationship between the number of reflections and the size of the d-spacing. The d-spacing that yielded the best refinement with enough reflections to satisfy the requirement for a good model was 0.5 (see table 7.2). These refinements were done in *SHELX* [24]. The R_1 figure of merit in *SHELX* is given by;

$$R_1 = \frac{\sigma \left| |F_o| - |F_{calc}| \right|}{\sigma |F_o|}. \quad (7.2)$$

Table 7.2: The fitness values and number of reflections for each d-spacing are given for spherical and elliptical integration methods (all refinements were done in SHELX). The radius of integration was 0.15.

Spherical Integration						
d-spacing	$R_1(F_o)^a$	$R_1(F_o)^b$	merged R_1	all data	data $> 4\sigma(F_o)$	merged data
0.4	0.0681	0.0721	0.0534	3255	3036	719
0.5	0.0531	0.0547	0.0381	1831	1768	376
0.6	0.0450	0.0457	0.0317	1076	1059	221
0.7	0.0428	0.0429	0.0304	697	694	145
Elliptical Integration						
d-spacing	$R_1(F_o)^a$	$R(F_o)^b$	merged R_1	all data	data $> 4\sigma(F_o)$	merged data
0.4	0.0668	0.0718	0.0532	3601	3315	722
0.5	0.0534	0.0556	0.0384	2127	2044	380
0.6	0.0478	0.0491	0.0384	1313	1289	225
0.7	0.0454	0.0461	0.0315	873	864	147

^a for $F_o > 4\sigma(F_o)$

^b for all data

7.1.2 Comparison to X-Ray refinement

Since the results had similar model fitness over all radii tested for both integration methods, the structural results should be investigated. The atomic positions for the neutron and X-Ray refinements are given in table 7.3. The neutron refinement positions are for a d-spacing 0.5 and a spherical integration radius of 0.15Å. The positions are the same within standard uncertainties for all atomic positions.

Given the very close values of the atomic positions from the neutron and X-Ray data structure refinements, a next important comparison is the anisotropic displacement parameters (ADPs). The ADPs reflect the uncertainty of the atomic position on the crystallographic site and are important characteristics of the model, as the model fitness can be improved by adjusting the U_{ij} values. The results in tables 7.4 and 7.5 reveal that the ADP values generally tend to decrease as a function of integration radii. In the elliptical integration method the biggest decrease occurs between 0.14 and 0.15 integration radii. The trend is not quite as clear for the spherical integration method, however the 0.15 integration radii gives reasonable

Table 7.3: The atomic positions in fractional coordinates of the X-Ray and neutron data structure refinements. For the neutron data the radius of integration was 0.15 and d-spacing was 0.5. The standard uncertainties are given in parenthesis.

X-Ray data structure refinement atomic positions					
atom label	x	y	z	occupancy	site symmetry multiplicity
La2	0.6667	0.3333	0.5000	0.5001	6
Na2	0.6667	0.3333	0.5000	0.5001	6
La1	0.0000	0.0000	0.0000	1	6
Na1	0.3333	0.6667	0.313(3)	0.5	3
F1	0.4103(11)	0.3706(9)	0.0000	1	2
F2	0.0239(8)	0.7307(8)	0.5000	1	2
Neutron data structure refinement atomic positions					
atom label	x	y	z	occupancy	site symmetry multiplicity
La2	0.6667	0.3333	0.5000	0.5001	6
Na2	0.6667	0.3333	0.5000	0.5001	6
La1	0.0000	0.0000	0.0000	1	6
Na1	0.3333	0.6667	0.325(3)	0.5	3
F1	0.4092(3)	0.3718(4)	0.0000	1	2
F2	0.0234(3)	0.7306(3)	0.5000	1	2

refinement results for both methods. The spherical integration method with d-spacing 0.5 and integration radii was refined in *SHELX* and the ADP values are compared to the X-Ray refinement results in table 7.6 (X-Ray CIF file in Appendix A. 3.1 and neutron refinement CIF Appendix A. 3.2).

The ADP results of the 100K TOF neutron data refinement produced similar values to those of the X-Ray refinement (see table 7.6). The relative scattering lengths of F and La allow the position of the F atoms to be better distinguished with neutrons than with X-Ray intensity data dominated by the heavy La. Therefore, the disordered F1 position has more variation in the U_{33} neutron APD than in the X-Ray and the ordered F2 position has more similar U_{ij} values (more spherical ADP) than the X-Ray F2 ADP.

If a simple linear correlation is assumed between the neutron and X-Ray ADPs a regression calculation reveals a correlation of about 91% with an intercept of approximately 0.002 and slope of approximately 0.74 (see figure 7.1 below). Removing the F ADPs that are less well determined in the X-Ray refinement improves the

Table 7.4: ADP values of NaLaF₄ for spherical integration methods at integration radii 0.13–0.18.

Spherical Integration							
int. radii	atom	U_{11}	U_{22}	U_{33}	U_{12}	U_{13}	U_{23}
0.13	La1	0.005269	0.005269	0.002609	0.002635	0	0
0.14	La1	0.005026	0.005026	0.002523	0.002513	0	0
0.15	La1	0.004740	0.004740	0.002434	0.002369	0	0
0.16	La1	0.004890	0.004890	0.002126	0.002445	0	0
0.17	La1	0.004652	0.004652	0.001757	0.002325	0	0
0.18	La1	0.004357	0.004357	0.002061	0.002179	0	0
<hr/>							
0.13	La2/Na2	0.006352	0.006352	0.013258	0.003176	0	0
0.14	La2/Na2	0.006336	0.006336	0.013454	0.003167	0	0
0.15	La2/Na2	0.006558	0.006558	0.013317	0.003279	0	0
0.16	La2/Na2	0.006264	0.006264	0.013357	0.003131	0	0
0.17	La2/Na2	0.006119	0.006119	0.013559	0.003060	0	0
0.18	La2/Na2	0.006452	0.006452	0.012737	0.003226	0	0
<hr/>							
0.13	Na1	0.015796	0.015796	0.033635	0.007898	0	0
0.14	Na1	0.015745	0.015745	0.033691	0.007872	0	0
0.15	Na1	0.015716	0.015716	0.033931	0.007857	0	0
0.16	Na1	0.015525	0.015525	0.034877	0.007762	0	0
0.17	Na1	0.015722	0.015722	0.034448	0.007861	0	0
0.18	Na1	0.015154	0.015154	0.035667	0.007577	0	0
<hr/>							
0.13	F1	0.008703	0.012763	0.043519	0.006977	0	0
0.14	F1	0.008616	0.012624	0.043063	0.006843	0	0
0.15	F1	0.008474	0.012371	0.042911	0.006640	0	0
0.16	F1	0.008348	0.012198	0.042255	0.006435	0	0
0.17	F1	0.008398	0.012180	0.041872	0.006520	0	0
0.18	F1	0.008267	0.012058	0.042812	0.006680	0	0
<hr/>							
0.13	F2	0.008141	0.007083	0.006349	0.001317	0	0
0.14	F2	0.008212	0.006857	0.006311	0.001276	0	0
0.15	F2	0.008339	0.006834	0.006297	0.001365	0	0
0.16	F2	0.008162	0.006812	0.006520	0.001353	0	0
0.17	F2	0.007989	0.006758	0.006813	0.001320	0	0
0.18	F2	0.008049	0.006563	0.006285	0.001264	0	0

Table 7.5: ADP values of NaLaF₄ for elliptical integration methods at integration radii 0.13–0.18.

Elliptical Integration							
int. radii	atom	U_{11}	U_{22}	U_{33}	U_{12}	U_{13}	U_{23}
0.13	La1	0.005283	0.005283	0.002543	0.002643	0	0
0.14	La1	0.005236	0.005236	0.002403	0.002617	0	0
0.15	La1	0.004991	0.004991	0.002348	0.002496	0	0
0.16	La1	0.004822	0.004822	0.002055	0.002410	0	0
0.17	La1	0.004680	0.004680	0.001886	0.002341	0	0
0.18	La1	0.004662	0.004662	0.001658	0.002331	0	0
<hr/>							
0.13	La2/Na2	0.006500	0.006500	0.002543	0.002643	0	0
0.14	La2/Na2	0.006465	0.006465	0.002403	0.002617	0	0
0.15	La2/Na2	0.006394	0.006394	0.002348	0.002496	0	0
0.16	La2/Na2	0.006227	0.006227	0.002055	0.002410	0	0
0.17	La2/Na2	0.006301	0.006301	0.001886	0.002341	0	0
0.18	La2/Na2	0.006291	0.006291	0.001658	0.002331	0	0
<hr/>							
0.13	Na1	0.015437	0.015437	0.033626	0.007718	0	0
0.14	Na1	0.015752	0.015752	0.033777	0.007875	0	0
0.15	Na1	0.015496	0.015496	0.032849	0.007748	0	0
0.16	Na1	0.015564	0.015564	0.032228	0.007782	0	0
0.17	Na1	0.015441	0.015441	0.032574	0.007721	0	0
0.18	Na1	0.015462	0.015462	0.032483	0.007732	0	0
<hr/>							
0.13	F1	0.008491	0.012418	0.042991	0.006648	0	0
0.14	F1	0.008430	0.012458	0.043344	0.006648	0	0
0.15	F1	0.008180	0.012200	0.043815	0.006471	0	0
0.16	F1	0.008098	0.011984	0.043539	0.006372	0	0
0.17	F1	0.007907	0.011859	0.043312	0.006244	0	0
0.18	F1	0.007873	0.011786	0.043344	0.006219	0	0
<hr/>							
0.13	F2	0.008560	0.007136	0.006461	0.001395	0	0
0.14	F2	0.008531	0.006979	0.006345	0.001354	0	0
0.15	F2	0.008288	0.006820	0.006115	0.001255	0	0
0.16	F2	0.008285	0.006784	0.006183	0.001213	0	0
0.17	F2	0.008187	0.006718	0.006132	0.001206	0	0
0.18	F2	0.007924	0.006628	0.006216	0.001187	0	0

Table 7.6: ADP values of NaLaF₄ 100K neutron and X-Ray *SHELX* refinement; standard uncertainties are given in parenthesis. The neutron data were integrated using spherical integration method and integration radii 0.15.

radiation type	atom	U_{11}	U_{22}	U_{33}	U_{12}	U_{13}	U_{23}
X-Ray	La1	0.0065(3)	0.0065(3)	0.0027(4)	0.00326(17)	0	0
Neutron	La1	0.0053(3)	0.0053(3)	0.0024(4)	0.00263(15)	0	0
X-Ray	La2/Na2	0.0046(5)	0.0046(5)	0.0142(9)	0.0023(3)	0	0
Neutron	La2/Na2	0.0062(5)	0.0062(5)	0.0137(9)	0.0031(2)	0	0
X-Ray	Na1	0.014(2)	0.014(2)	0.021(4)	0.0070(12)	0	0
Neutron	Na1	0.0156(13)	0.0156(13)	0.034(4)	0.0078(7)	0	0
X-Ray	F1	0.008(2)	0.007(2)	0.037(3)	0.0045(16)	0	0
Neutron	F1	0.0083(5)	0.0125(6)	0.0429(13)	0.0067(5)	0	0
X-Ray	F2	0.009(2)	0.012(2)	0.0046(19)	0.003(2)	0	0
Neutron	F2	0.0084(5)	0.0070(4)	0.0066(4)	0.0014(4)	0	0

correlation slightly. This result is a verification of the neutron TOF Bragg data correction and integration protocol and provides a basis for development of an event mode data correction and integration protocol necessary for diffuse scattering.

7.2 New quantitative method verification

The results in this section are an expanded version of the paper:

Michels-Clark, T.M.*, Lynch, V.E., Hoffmann, C.M., Hauser, J., Weber, T., Harrison, R., Bürgi, H. B. Analyzing diffuse scattering with Supercomputers. *J. Appl. Crystallogr.*, **46**, 1616-1625

I was lead and corresponding author and my primary contributions to this paper include (i) testing and development of method and analysis (ii) running of computations (iii) writing manuscript and preparing all figures for publication (iv) writing of results analysis code (v) corresponding with editor and referees (vi) final submission.

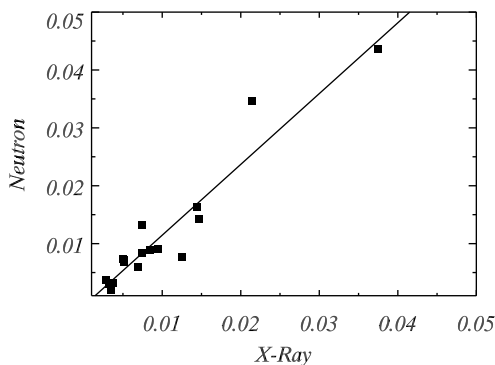


Figure 7.1: Linear regression of 100K X-Ray ADPs plotted against the 100K neutron ADPs a general linear correlation is observed.

7.2.1 Comparison of global optimization methods

Comparison of global parameter optimization by GA, DE, and PSO was performed using 20 clones for each calculation. All three methods show a rapid decrease of \bar{R} , within the first 60 generations. Further decrease of \bar{R} in following generations is gradual. GA, DE, and PSO start with an average fitness \bar{R} of 0.75, 0.75, and 0.69, respectively, in generation zero. \bar{R} of PSO drops fastest, followed by GA. While PSO and GA are seemingly leveled, \bar{R} of DE keeps reducing. At generation 63, GA is surpassed by DE, which shows an $\bar{R}(s)$ of 0.03(1). Finally, DE outperforms PSO in generation 82 with an $\bar{R}(s)$ of 0.019(4). The log scale in Figure 7.2 emphasizes the differences in convergence. The uncertainty $s(\bar{R})$ also decreases. The distributions of \bar{R} become narrower by a factor of ~ 2 between generations 64 and 82. DE converges to a population with $\bar{R} = 0.0090(4)$, compared to the starting range of $\bar{R} = 0.76(23)$. The distribution of uncertainties of \bar{R} -values for GA (0.0338(6)) and PSO (0.02 (1)) are more than an order of magnitude greater for PSO than for GA.

Both PSO and DE converged to essentially the same parameter values, but for PSO the uncertainties in the final generation are larger than for DE and accompanied by inferior overall population fitness. GA converged to similar parameter values as DE and PSO with the exception of e_2 (Table 5.2). We hypothesized that GA

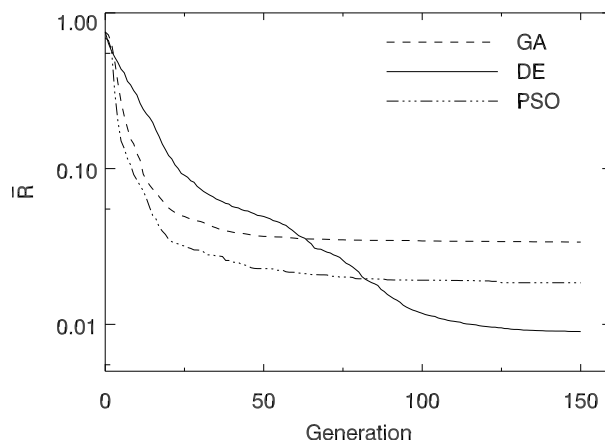


Figure 7.2: Evolution of population fitness \bar{R} for three global optimization algorithms: general genetic algorithm (GA) and differential evolution (DE) and particle swarm (PSO) as a function of generation number.

converged into a local rather than the global minimum, which is supported by a higher \bar{R} -value in the final generation. This hypothesis was tested by changing the pathological parameter in small steps while keeping the remaining parameters fixed at their refined values. The \bar{R} -value was expected to cross a fitness barrier to arrive at the real solution; however, \bar{R} decreased continuously without going through a maximum. Thus, GA possesses similar global optimization power to DE and PSO, but the local optimization power of GA seems inherently weaker than that of PSO or DE.

As illustrated in Figures 7.3a-g for DE optimization, the convergence behavior of the different model parameters varies considerably.

Figures 7.3a-h show the maximum, average, and minimum parameter values (filled circles) and the standard deviation of the population (vertical lines) for every parameter in every generation. The average \bar{R} is cut in half after 5 generations, then again after 7, 9, and ~ 20 generations, showing the decrease in efficiency as the optimization progresses (Figure 7.3h). Parameter c converges at the same rate as \bar{R} (Figure 7.3a), while the other parameters trail behind. Parameters c and t

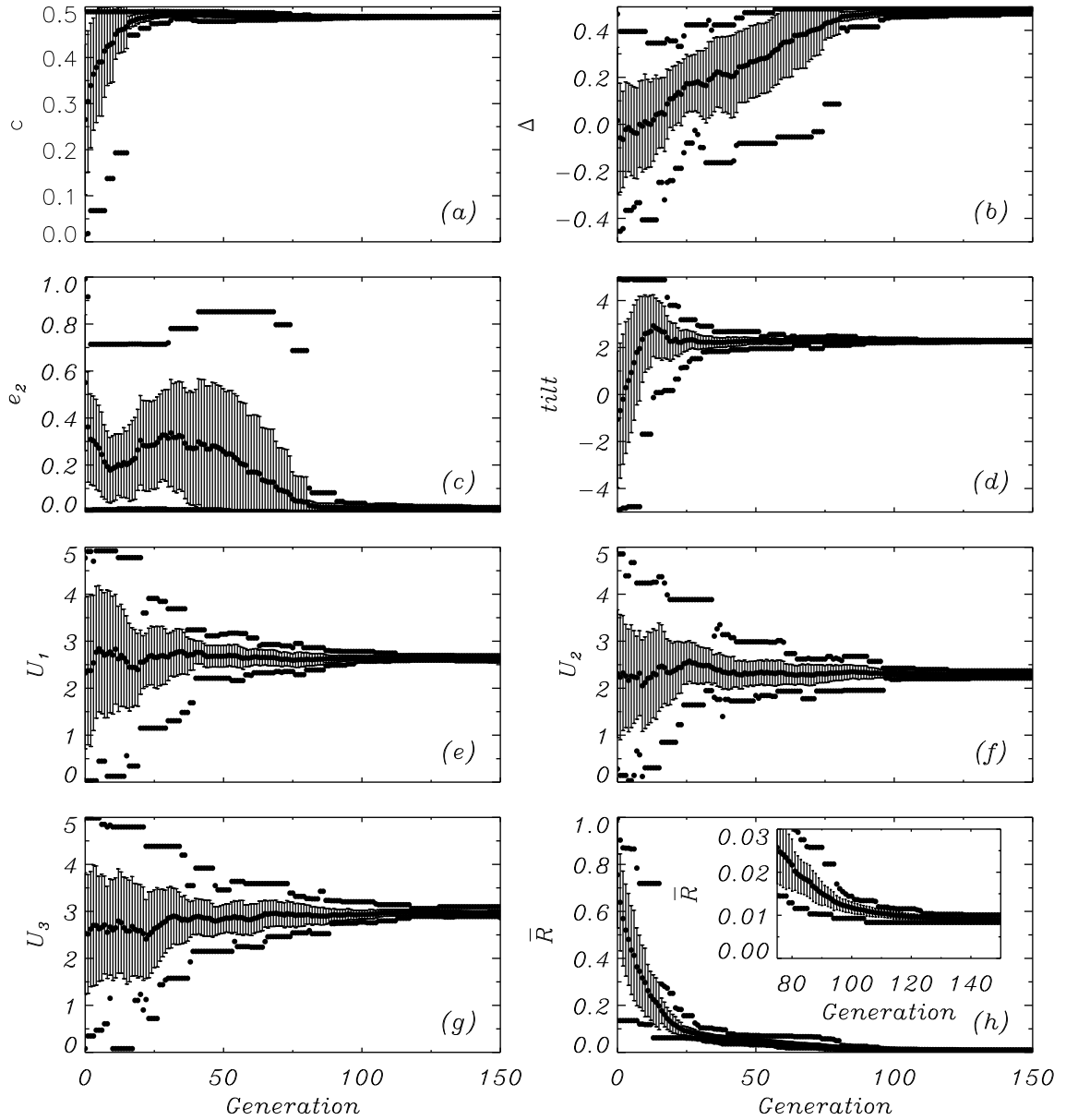


Figure 7.3: Behavior of the model parameters c , Δ , e_2 , $tilt$, U_1 , U_2 , U_3 (plots a-g), and fitness \bar{R} (plot h) during DE optimization with 40 individuals and 20 clones each. The population mean is indicated by the middle points, and the population standard deviation by the vertical lines. The dots above and below the lines represent the largest and the smallest parameter values in the population.

(constrained to be numerically equal) represent the probability of the bent stacking, which is energetically favorable compared to eclipsed stacking (b_R , b_L vs. e), the former being prevalent in the structure. Thus parameters c and t contribute more than the other model parameters toward modeling the reference intensity and the overall fit.

In contrast to c , e_2 is slower to converge (Figure 7.3c). The e_2 parameter is defined as the probability of continuing an energetically unfavorable eclipsed arrangement. Since this stacking option has a low probability, it is infrequently present in the local structure and therefore contributes modestly toward fitting the reference intensity.

The Δ parameter (Figure 7.3b), which distinguishes between the layer stacking to either continue in the same direction ($+\Delta$) or to change direction ($-\Delta$), begins to converge approximately at generation 60 and continues to converge within a small uncertainty in the final generation (Table 5.2, column 4). Since Δ determines the details of bent motifs, it is associated with c and represents a significant determinant of the model intensities.

The *tilt* parameter, which defines the degree of molecular out-of-plane tilt allowed in layer $n + 1$ of a b_R or b_L (but not an e) motif, refines to its optimal value in approximately 80 generations (Figure 7.3d and Table 5.2, column 4). The U'_i s representing the atomic displacement parameters of TBHB all converge at the same rate, settling to an optimal value in approximately 100 generations (Figure 7.3e-g and Table 5.2, column 4).

Generally, small standard deviations of the parameters indicate that the originally quite different 40 gene sets have converged to a single solution. The ratio between the standard deviations and the mean parameter values after 150 generations of DE optimization is in the range of 0.003% for c and 2% for U_3 (Table 5.2), indicating convergence of the optimization to a single solution.

To summarize this section, we tentatively conclude that PSO can initially navigate the search space most efficiently since all of the variables change simultaneously toward the best solution seen by \mathbf{b}_i and \mathbf{b}_p : PSO outperforms DE and GA in

the first 20 cycles of optimization (figure 7.3). However, towards the end of the optimization process, the collective fitness of the models obtained by DE is better than that from GA and PSO. Initializing an optimization with several generations of PSO followed by DE will most likely make the best use of computational resources. When the parameters are fairly well clustered, convergence tends to slow down; it may then be advantageous to conclude the optimization by a numerical least-squares calculation[12]. In the present case, least-squares optimization could start after 50 generations when considering the rapidly converging parameters, or after 100 generations with regard to the slowly converging parameters.

7.2.2 Influence of clones

The dependence of the model fitness on speckle-type intensity variations has been tested for DE only. The results after 150 generations of DE optimization with different crystal sizes and different numbers of clones are reported in Table 7.7. As expected \bar{R} decreases with either increasing the crystal size or increasing the number of clones. The decrease in \bar{R} shows a linear trend with the square root of the reciprocal product of the number of clones and the number of lots per clone, *i.e.*, the total number of lots included in the calculation (Figure 7.4).

This behavior indicates that the speckled nature associated with the intensity of individual lots behaves analogously to noise in an experiment [89]. The incorporation of clones reduces the statistical noise inherent in the MC process. On average the standard deviations of the simulated parameters also decrease roughly with the inverse square root of the number of clones (sample size) as expected if the mean parameter values are normally distributed (Table 7.7). The averages of the parameters over the different gene sets are mostly within one standard deviation of the reference values used to construct the data set (see Section 5.2, Chemical Model System, Growth Modeling, and Reference Data, and Table 7.7).

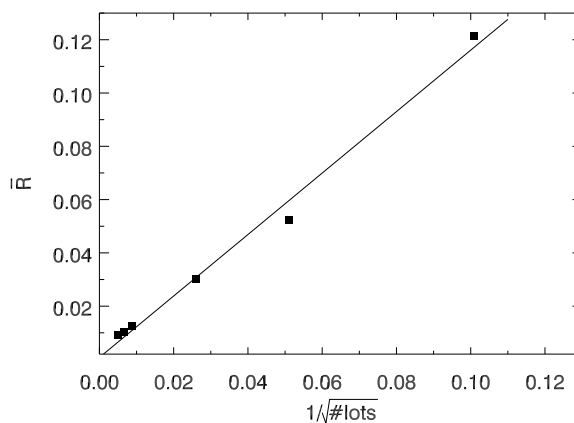


Figure 7.4: Decrease of \bar{R} with increasing number of lots. \bar{R} depends linearly on the reciprocal square root of the number of lots ($\#lots$).

Two additional conclusions considering computational efficiency are worth mentioning. Firstly, in columns 6-8 of Table 7.7, \bar{R} remains approximately constant, showing that decreasing the crystal size can be compensated by increasing the number of clones, thereby allowing a higher degree of parallelization. Secondly, 40 clones for each individual consisting of 96'000 layers achieved a good, but not perfect, agreement with the reference data with $\bar{R} = 0.0077(3)$ in the 150th generation (Table 7.7, column 3). In contrast, the \bar{R} -values that can be achieved from state-of-the-art experiments by state-of-the-art structure refinement of disorder models against diffuse scattering data are about 0.1 in favorable cases. As columns 9-10 of Table 7.7 indicate, a trustworthy estimate of the model parameters may be obtained from crystals of modest size, albeit with relatively large uncertainties.

Table 7.7: Summary of results after 150 generations of DE optimization with 40 gene sets that have different numbers of layers, lots per crystal, and numbers of clones. The number of cores used, wall clock time and the cpu hours are also reported for each calculation.

size\layers		96,000	96,000	96,000	96,000	6,000	600	600	600
num. Lots		1600	1600	1600	1600	100	10	10	10
num. clones		40	20	10	1	16	160	40	10
par.	reference								
c	0.48877	0.4884(3)	0.4883(4)	0.4883(5)	0.487(1)	0.488(2)	0.487(2)	0.488(2)	0.487(8)
Δ^a	0.49336	0.486(6)	0.482(7)	0.479(8)	0.45(2)	0.452(22)	0.444(23)	0.41(5)	0.25(11)
e_2	0.006748	0.005(4)	0.008(4)	0.014(10)	0.003(21)	0.036(4)	0.0343(25)	0.057(7)	0.2(2)
$tilt$	2.2723	2.27(1)	2.27(2)	2.27(3)	2.28(13)	2.29(7)	2.27(8)	2.27(8)	2.28(40)
U_1	2.6284	2.651(13)	2.65(2)	2.63(4)	2.61(13)	2.65(10)	2.63(10)	2.6(2)	2.8(5)
U_2	2.2734	2.27(2)	2.27(4)	2.27(6)	2.34(14)	2.31(20)	2.28(13)	2.3(3)	2.5(6)
U_3	3.0005	2.97(3)	2.97(6)	2.99(8)	3.0(3)	3.02(19)	3.01(22)	3.0(4)	3.1(8)
t^a	0.48877	0.4884(3)	0.4883(4)	0.4883(5)	0.487(1)	0.488(2)	0.487(2)	0.488(2)	0.487(8)
e_1^a	0.022452	0.0231(6)	0.0235(7)	0.023(1)	0.026(2)	0.025(3)	0.024(3)	0.024(4)	0.03(2)
R^b	-	0.0077(3)	0.0091(5)	0.011(9)	0.029(3)	0.0297(3)	0.0288(3)	0.051(6)	0.12(3)
Cores	-	160	160	160	40	160	160	160	160
wall clock time (hours)		136	63	27	8	6	57	30	9
CPU hours = Cores * Wall Clock time		21760	10080	4320	320	960	9120	4800	1440

^aConstrained parameters: $c = t, (c \pm \Delta) + (t \mp \Delta) + e_1 = 1$

^bR-factor measures fitness

The most important consequence of replacing large crystals by clones is the possibility to parallelize computations. The calculation time for 40 individuals (without clones) sums up to about 8 hours on 40 cores for 150 generations. This corresponds to a total cpu time of $40 * 8 = 320$ hours. As more clones are added, the total cpu time increases from 320, to 4'320, to 10'080, to 21'760 hours for 1, 10, 20, and 40 clones, respectively (Table 7.7, columns 3-6). However, if enough cores are available to calculate the fitness of one clone per core, the wall clock time stays approximately the same as the calculations are performed in parallel. In our case, the predicted scaled numbers, found by dividing the wall clock time by the total number of gene sets and their clones per processor, expand slightly from 8 to 10.8 to 12.6 and 13.6 hours. This corresponds to an increase in simulation time by a factor of 1.7 for a calculation that is 40 times larger. The modestly expanded wall clock time is due to increased communication between nodes required for averaging the intensities of the clones. The wall clock time in columns 8-10 of Table 7.7 follow the same trend as columns 3-6 as more clones are added.

7.2.3 Polynomial fitting to eight dimensional fitness surface

In order to estimate which parameters of the TBHB test model best determine the model fitness of the diffuse scattering intensity data, a second order polynomial was fit to the fitness and parameters values of the converged population (150th generation) of 40 individuals averaged over 10 clones. These individuals form a constellation of parameter vectors each defining a single point on the eight dimensional fitness surface. The model fitness surface is defined by 7 model parameters plus the surface height (fitness). If these points are clustered about the minimal solution then the hypothesis is that these points should approximate a parabolic function whose coefficients are determined by a least squares fit.

The origin of the function is shifted relative to the constellation of points to the center. The eight dimensional function is defined by the Taylor series expansion of

parameters about their mean values, given by:

$$\Delta_{\mu} = c + \sum_{i=1}^7 g_i (p_{\mu} - \bar{p})_i + \frac{1}{2} \sum_{i,j=1}^7 h_{i,j} (p_{\mu} - \bar{p})_i (p_{\mu} - \bar{p})_j, \quad (7.3)$$

where \bar{p} is the average parameter value over all individuals μ and c is the model fitness of \bar{p} . The difference from \bar{p} for each parameter vector element p_i and p_j are calculated for every individual, p_{μ} in the population. Linear and quadratic coefficients of the second order polynomial are given by g_i and $h_{i,j}$ respectively. The second order coefficient matrix, $h_{i,j}$ is symmetric.

The least squares fitting procedure minimizes the sum of the residuals resulting from the fitness of each individual $F(p_{\mu})$ and the approximation of the fitness surface function (7.3) squared. This function is given by:

$$L = \sum_{\mu=1}^{40} [F(p_{\mu}) - \{c + \sum_{i=1}^7 g_i (p_{\mu} - \bar{p})_i + \frac{1}{2} \sum_{i,j=1}^7 h_{i,j} (p_{\mu} - \bar{p})_i (p_{\mu} - \bar{p})_j\}]^2. \quad (7.4)$$

Since $\frac{1}{2} \sum_{i,j=1}^7 h_{i,j} (p_{\mu} - \bar{p})_i (p_{\mu} - \bar{p})_j$ results in a 7X7 symmetric matrix there are 28 unique second order fitting terms, $\sum_{i=1}^7 g_i (p_{\mu} - \bar{p})_i$ gives 7 linear fitting terms and c corresponds to the constant term of the second order polynomial approximation to the fitness surface resulting in 36 total fitting terms. 7.4 is written in matrix equation form, collecting c and the first and second order polynomial coefficients together:

$$L = \sum_{\mu=1}^{40} [F(p_{\mu}) - \sum_{k=1}^{36} \mathbf{z}_k \mathbf{q}_{k,\mu}^{s_k}]^2. \quad (7.5)$$

$\mathbf{z}_k = \begin{pmatrix} c \\ g_i \\ h_{i,j} \end{pmatrix}$ and $\mathbf{q}_{k,\mu}^{s_k}$ corresponds to the zero, first and second order ($p_\mu - \bar{p}$) terms in 7.4. The index s_k is given by

$$s_k = \begin{cases} 0, & k = 1 \\ 1, & 2 \leq k \leq 8 \\ 2, & 9 \leq k \leq 36 \end{cases}$$

The residual is minimized by calculating the zeros of the first derivative. This provides estimating equations to determine the fitting coefficients.

$$\frac{\partial L}{\partial z_k} = 2 \sum_{\mu=1}^{40} [F(p_\mu) - \sum_{k=1}^{36} \mathbf{z}_k \mathbf{q}_{k,\mu}^{s_k}] (-\sum_{l=1}^{36} \mathbf{q}_{l,\mu}^{s_l}) = 0 \quad (7.6)$$

Simplifying 7.6 yields

$$\sum_{\mu=1}^{40} \sum_{k=1}^{36} \mathbf{z}_k \mathbf{q}_{k,\mu}^{s_k} (\mathbf{q}_{k,\mu}^{s_k})^T = \sum_{\mu=1}^{40} \sum_{k=1}^{36} F(p_\mu) (\mathbf{q}_{k,\mu}^{s_k})^T. \quad (7.7)$$

Solving for the fitness surface height and the first and second order fitting coefficients of \mathbf{z}_k is now a matter of solving the associated normal equations in the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{A} = \sum_{\mu=1}^{40} \sum_{k=1}^{36} \mathbf{q}_{k,\mu}^{s_k} (\mathbf{q}_{k,\mu}^{s_k})^T$, $\mathbf{x} = \sum_{k=1}^{36} \mathbf{z}_k$ and $\mathbf{b} = \sum_{\mu=1}^{40} \sum_{k=1}^{36} F(p_\mu) (\mathbf{q}_{k,\mu}^{s_k})^T$. The second order parameters $h_{i,j}$ correspond to the variance/co-variance of the parameters of forty individuals after a hundred and fifty generations. This gives us valuable information about the contributions of the individual parameters to the fitness. The eigenvalues of the quadratic coefficient matrix indicate the curvature of the fitness surface. Negative eigenvalues indicate negative curvature, which would mean a non-parabolic surface (i.e. a saddle point).

Unfortunately, the condition number of the least squares matrix is quite large, which indicates poor fitting precision and a poorly conditioned problem. The eigenvalues of the coefficient matrix (\mathbf{z}_k) were small with many equivalent to machine

noise (approximately 10^{-14}), yielding a condition number of about 10^{15} . The number of fitting terms is 36 and we are fitting this with 40 data points randomly distributed, so this result is not entirely surprising.

Solving the normal equations using the singular value decomposition was attempted to improve the condition of the problem, this method eliminates the smaller eigenvalues but this did not significantly improve the condition number of the fitting coefficient matrix. Some of the diagonal second order coefficients were negative indicating that the MC noise in the converged population of 40 individuals is too large to determine the expected parabolic dependence of the fitness on the deviations of the parameters from their best value. One possible way to resolve this issue would be to include more fitting points. These points should represent areas that would be further away from the minimum and thus farther from the vertex of the parabolic fitness surface. So, one would need to consider incorporating the discarded children of the previous generations in the least squares fitting procedure.

7.2.4 Sensitivity analysis

The sensitivity of R on an individual parameter was tested by varying each independent parameter separately [90] and calculating the resulting model fitness (table 7.8). As only the sensitivity relative to the point estimates chosen and not the entire parameter distribution was taken into account, this is a local sensitivity analysis. Testing the sensitivity of the converged (150th generation) of DE optimized parameters (table 5.2, column 5) allows examining the effect of each individual parameter on the fitness of the model.

The goal of this study was twofold: (1) to investigate which refinement parameters are most and least influential in the model calculation and optimization with a sensitivity and principal component analysis (section 7.2.5), and (2) to adapt the parameter choice most efficiently for large, parallelized, multi core, super computer usage. The independent model parameters were individually displaced by $\pm 4\sigma$ from

Table 7.8: The decrease in the average model fitness (incorporating 20 clones) is given as a percent change from optimized population model fitness. A positive value percent change indicates an increase in R and thus a poorer model fitness when the given parameter is shifted.

Parameter	shift	
	$+4\sigma$	-4σ
c	0.95	19.51
Δ	3.73 ^a	7.12
e_2	0.70	3.73 ^b
<i>tilt</i>	8.83	3.72
U_1	2.34	1.87
U_2	4.95	4.61
U_3	5.74	5.56

^acorresponds to $+1.82\sigma$ parameter change placing parameter at its maximum limit (0.5) see table 5.2.

^bcorresponds to -1.875σ parameter change placing parameter at its minimum limit (0.0) see table 5.2.

their DE optimized values, when this value was within the parameter limits table 5.2, column 3. The exceptions were the Δ and e_2 parameters; their limits permitted them to be varied $+1.82\sigma$ and -1.875σ at the high and low ends of each of their value ranges, respectively. Table 7.8 shows the largest changes occur at 8.83 for *tilt* in the positive direction ($+4\sigma$) and 19.51 for c in the negative direction (-4σ). The c parameter is more sensitive in the -4σ direction, changing the overall model fitness by 19.51% versus the positive displacement which decreases the model fitness by 0.95. This is not unexpected as c accounts for the greatest percentage of layer stacking preferences which comprise a large portion of the structure. Decreasing c would imply that bent stacking has a diminished probability of occurrence therefore the local structure would then contain a smaller percentage of bent motifs and a larger percentage of eclipsed motifs. This change in the structure would then contribute to the calculated intensity and ultimately the model fitness (see discussion on parameter convergence 7.2).

The $\pm 4\sigma$ shifts in the *Tilt* and Δ parameters result in an asymmetric shift in the model fitness. A greater change in the model fitness (approximately 9%) corresponding to decreasing the Δ parameter value by 4σ than for increasing the parameter value by 4σ . The opposite trend was observed for the *Tilt* parameter. The model fitness shifts are 3.73/7.12 and 8.83/3.72 for *Tilt* and Δ , respectively.

The Δ parameter defines the preference of the stacking direction either to continue stacking in the same direction ($+\Delta$) or change direction ($-\Delta$). The asymmetric shift in the model fitness shift as a result of adjusting the Δ parameter is a consequence of the parameter's limits (table 5.2). The maximum value of Δ is 0.5, which corresponds to a deviation of $+1.82\sigma$ and a resulting smaller shift in model fitness than when Δ is shifted by -4σ .

The asymmetric model fitness changes in the $\pm 4\sigma$ of *tilt* must be explained in terms of the parameter's effect on the structure and the nature of the layer stacking. The greater sensitivity in the positive direction is a result of the implications of a larger tilt as this influences the structure to stack in a more ordered fashion and cause the layers to organize much like shingles on a roof as in the monoclinic ordered polymorph [18]. Thus, the local structure variation is minimized (more bent motifs and fewer eclipsed variations) ultimately effecting the match between the model and reference intensities.

The isotropic displacement parameters are fairly symmetric in their fitness shifts. The shift from 2.34/1.87(+/-) for U_1 , 4.95/4.61 for U_2 to 5.74/5.56 for U_3 . The U_i parameters define the average isotropic atomic displacement parameters of the carbon atoms of TBHB. The atomic displacement of the carbons in the central benzene ring is defined by U_1 . The middle "bridged" atomic displacement of the carbons in the three five member rings surrounding the benzene is defined as U_2 . Finally, U_3 is defined as the atomic displacement of the outermost carbons of the three five member rings (as described in figure 5.1 center). Based on this physical description, understanding the increasing changes in the average model fitness from U_1 to U_2 to U_3 implies the atomic

displacement of the outermost carbons (U_3), which allow the greatest displacement, most effect the model intensities.

The e_2 parameter is restricted to a minimum value of zero (meaning no eclipsed stacking), which corresponds to a parameter value change of -1.875σ and a model fitness shift of 3.73 percent. The smaller change in model fitness (0.70 percent) corresponds to an increase in the percentage of eclipsed motifs in the overall structure and parameter value change of $+4\sigma$. The e_2 parameter is less sensitive in the positive direction ($+4\sigma$), indicating that in spite of an increased probability of stacking in an eclipsed fashion, the eclipsed arrangements still account for only a small portion of the overall structure as compared to the bent arrangements and therefore contribute minimally to the model intensity.

In order to understand the structural parameters effect on the model fitness, a corresponding direct space sensitivity analysis was done to determine the structural changes in the crystal. The C++ code to calculate the percentages of each motif type is given in Appendix A section A. 2. Each structural parameter (c , Δ , e_2) was varied by $\pm 4\sigma$ (table 7.9), as the parameter limits (table 5.2) allow. The motif percent changes are given relative to the percent of the structure motifs calculated from crystals generated with the optimal model parameters (c , Δ , e_2) averaged over 20 clones. The changes in the c parameter effects the structure most in the -4σ direction with new structural statistics of 0.0395%, 45.014%, 44.94% and 5.0% for eclipsed(ee), bent(b_Lb), bent (b_Rb), and mixed eb and be , respectively. The percentage of the ee motifs increases by 0.0075 while the b_Lb and b_Rb percentages decrease by 0.596 and 0.58, respectively, as compared to the optimal model parameter layer stacking statistics for 20 clones table 7.9, column 2. For the mixed eb and be cases a -4σ shift in c corresponded to an increase in the percentage (0.58) of these motifs relative to the 20 clone case. However, adjusting c by $+4\sigma$ shifts the structural statistics to 0.0285%, 46.19%, 46.09% and 3.843% for percentages of ee , b_Lb , b_Rb and both mixed eb and be , respectively. The percentage of ee statistics decreases by 0.0035, b_Lb and b_Rb increase by 0.58 and 0.574, respectively, while the percentage of mixed motifs both decrease

Table 7.9: Changes in structural motif composition based on changes in the independent model parameters defining the various stacking options; c , Δ and e_2 are shown below. Column 2 corresponds to the structural statistics of the DE optimized parameters averaged over 20 clones.

	20 clones	c		Δ		e_2	
		+4 σ	-4 σ	+1.82 σ^a	-4 σ	+4 σ	-1.875 σ^b
eclipsed (ee)	0.032(7)	0.0285(57)	0.0395(08)	0.032(08)	0.032(07)	0.106(08)	0.0(0)
Bent (b_Lb)	45.61(21)	46.19(21)	45.014(20)	45.52(21)	45.83(23)	45.58(21)	45.63(22)
Bent (b_Rb)	45.52(18)	46.09(17)	44.94(18)	45.42(18)	45.74(17)	45.49(17)	45.53((17)
Mixed eclipsed & bent (eb)	4.42(9)	3.843(79)	5.00(10)	4.51(09)	4.20(09)	4.41(08)	4.42(09)
Mixed eclipsed & bent (be)	4.42(9)	3.843(80)	5.00(10)	4.51(09)	4.20(09)	4.41(08)	4.42(09)

^acorresponds to +1.82 σ parameter change placing parameter at its maximum limit (0.5) see table 5.2.

^bcorresponds to -1.875 σ parameter change placing parameter at minimum limit (0.0) see table 5.2.

by 0.577 all relative to the optimal 20 clone case. In general, the percentage of bent stacks increases while the number of eclipsed and mixed stacks decreases when the c parameter is adjusted in the +4 σ direction the converse is true for a -4 σ shift in c . Since c represents stacking in a bent manner the results for -4 σ shift correspond to the greatest structural and model fitness shifts observed. These results are physically reasonable as c describes the greatest percentage of motifs composing the structure.

The shifts in the Δ parameter ($\pm 0.007 * 4$) effected the bent and disordered motifs only. Adjusting Δ by +4 σ (0.028) would place Δ outside of the parameter limit (table 5.2) therefore adjusting Δ to the limit (0.50) corresponded to a shift of +1.82 σ . A shift in Δ of +1.82 σ results in new motif structural statistics; 0.032%, 45.52%, 45.42% and 4.51% for ee , b_Lb , b_Rb and both mixed eb and be , respectively. There is no change in the ee motif percentage from the optimal parameter case and for b_Lb and b_Rb there is an increase of 0.10% and 0.09% respectively. The number of disordered motifs does increase by 0.09% from the optimal case. In contrast when Δ is decreased the percentage of disordered motifs (eb and be) decreases by 0.22%, the bent motifs (b_Lb and b_Rb) both increased by 0.22%. The structural statistics are 0.032%, 45.83%, 45.74% and 4.20% for ee , b_Lb , b_Rb and both mixed eb and be respectively. The Δ parameter most significantly effects the bent arrangements within the structure. The changes for Δ however, are less significant since this parameter

accounts for a smaller amount of the structure description. Similarly this is seen in the sensitivity analysis for the reciprocal space, where shifts in Δ affect the overall model fitness much less than shifts in c .

The shifts in the e_2 parameter ($\pm 0.004 * 4$) effected the eclipsed motif ee most. Adjusting e_2 by -4σ (0.016) would place e_2 outside of the parameter limit (table 5.2, column 3) therefore adjusting e_2 to the limit (0.0) corresponded to a shift of -1.875σ . A shift in e_2 by $+4\sigma$ produced the following structural statistics: 0.106% ee , 45.58% b_Lb , 45.49% b_Rb and 4.41% mixed eb and be . This corresponds to an increase in ee by 0.074% and decreases in b_Lb and b_Rb of 0.03%. The disordered motifs essentially remained the same (within statistical noise). When e_2 is increased this increases the probability of getting more eclipsed layers since e_2 represents the probability of continuing layer stacking sequence in an eclipsed fashion this would correspond to an increase in ee or eclipsed motifs in direct space. Similarly adjusting e_2 by -1.875σ make the e_2 parameter 0.0 and thus the percentage of eclipsed motifs in direct space is 0.0. The bent motifs b_Lb and b_Rb remain the same (within statistical noise) as the optimal parameter structural statistics with values of 45.63% and 45.53% respectively. The percentages of the disordered motifs remain the same as those for the optimal parameters of the 20-clone case.

The direct space sensitivity analysis reveals the effect of varying structural parameters (c , Δ , e_2) on the various motif statistics in the resulting crystal. The resulting structural changes directly correlate with the reciprocal space sensitivity analysis for these same parameters (c , Δ , e_2) on the overall model fitness, meaning bigger changes in structure resulted in larger changes in the model fitness. This is most significantly observed in the c parameter, when c is decreased by -4σ the model fitness decreases by 19.51% and correspondingly the largest changes in the motif statistics are also observed. The Δ parameter revealed the least structural statistical change and model fitness change. Therefore, the results of both the direct and reciprocal space sensitivity analysis are complementary.

7.2.5 Principal component analysis

To determine if parameters are correlated a principal component (PCA) analysis was performed. PCA is a mathematical procedure that uses an orthogonal transformation to determine the directions of maximum variance and reveals if parameters are correlated [46]. Since PCA is sensitive to the scaling of each parameter, the parameters are transformed so that the converged parameter population has a mean of zero and variance of one. The first or principal component of the parameter distribution accounts for the maximum amount of the total population variance and is correlated with some of the parameters and possibly all of them. The second principal component will account for the maximum amount of the total variance not associated with the first component and will be correlated with the parameters that were not strongly correlated with the first. In the TBHB model defined by seven independent fitting parameters there are seven principal components. The principal components of the 0, 10, 20 and 40 clone in the converged population of the 150th generation were calculated. Each principal component is a combination of all seven independent parameters. The eigenvalues of the 7 parameter variance/covariance matrix are the magnitude of the 7 principal components (eigenvectors); these values for the 40 clone results were 1.4118799, 1.2762547, 1.0724140, 0.9237576, 0.8576303, 0.6659907 and 0.4419231 for each of the seven principal components respectively. The direction of principal change (1.412) is not significantly larger than that of the other principal components, indicating that the primary direction in the data distribution is not dissimilar to the less significant directions. Therefore, in the TBHB model each principal component contains significant data relationships.

7.2.6 Examination of e_2 parameter

Based on the results of the sensitivity analysis described in 7.2.4 the interdependence of our model parameters was considered to determine if there exist correlations among them. The e_2 parameter was determined the most insensitive in the model results.

The e_2 was decreased to 0.0 and the model fitness did not change significantly (0.70%); for a $+4\sigma$ increase the model fitness in table 7.8 and the structural motif statistics 7.9 for bent motifs did not change significantly from those of the 20 clone results. The correlation of the e_2 parameter with the other structural parameters was tested.

In the model definition the *tilt* is 0 when the layers are eclipsed. A scan over *tilt* slowly varying e_2 and calculating the fitness in the third dimension was done using twenty clones and keeping the other parameters fixed at their optimal values. *Tilt* was varied in increments of 0.5 between the values of 0 and 3.0 and e_2 was varied in the range of 0.0015 and 0.2575. In general, as e_2 and *tilt* get closer to the optimal values found in the 150th generation, the the fitness gets better as the parameters get closer to their optimal values and worse when the parameters are farther from the optimal values. In the contour plot of e_2 versus *tilt* (figure 7.5) the fitness is better at the optimal parameter values for both e_2 and *tilt*. Between the e_2 parameter values of 0.00150 and 0.05750 and with *tilt* fixed at optimal value of 2.2734, the fitnesses are indistinguishable for 20 clones within a standard deviation in $\chi[\text{chi}]$. When *tilt* = 2.2734 and e_2 is between 0.00150 and 0.05750 the model fitnesses shift $+2\sigma$ from the optimal solution, indicating that the e_2 parameter does not significantly affect the model over this relatively wide range. The contour plot reveals how the fitness changes as a function e_2 and *tilt* and remains uncorrelated until close to the global minimum.

Continuing our investigation a contour plot of e_2 and Δ as a function of fitness was done (figure 7.6). Delta was scanned in increments of 0.1 between the values of 0 and 0.6 and e_2 was varied between the values of 0.0015 and 0.2575. The Δ parameter was chosen because it represents the continuation of layer stacking to the next layer and considering the interaction depth to another layer could affect the value of the e_2 parameter. Since e_2 is so relatively undetermined between values of 0 and 0.0750 we wanted to test if this parameter could possibly be correlated with other structural parameters such as *tilt* and *delta*. The investigation seen in the contour plots of

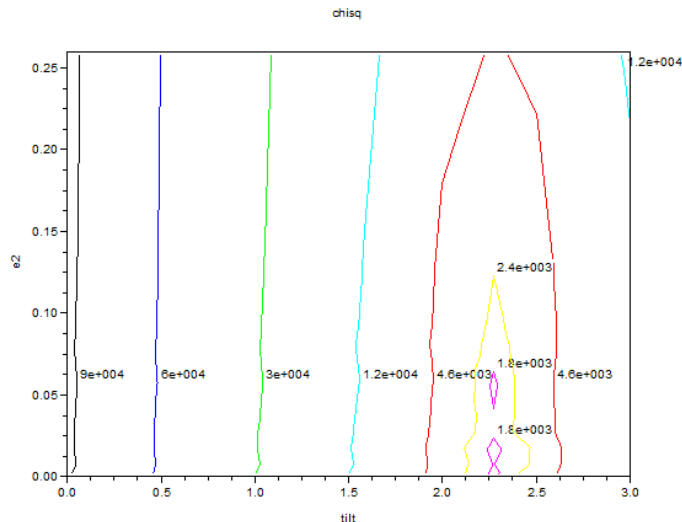


Figure 7.5: Contour plot depicting correlation of e_2 and $tilt$ and fitness (given as $\chi[\text{chi}]$) in the third dimension. The parameters remain uncorrelated until close to the global minimum.

figures 7.5 and 7.6 show no correlation between these parameters and e_2 until close to the global minimum, indicating the model parameters are not correlated.

7.2.7 Motif statistics and comparison with reference model

The purpose of modeling diffuse scattering is to gain insight into the structural motifs composing the crystal, in our case, the types of stacking sequences of TBHB and their lengths. Due to the probabilistic growth or MC procedures for building and equilibrating crystals, a model is not expected to be a one-to-one image of the sample investigated. However, it must show the same statistical distribution of structural motifs. To test this, the occurrence of the nine structural motifs: eb_L , eb_R , ee , b_Lb_R , b_Lb_L , b_Le , b_Rb_L , b_Rb_R , and b_Re in the 1280 virtual reference crystals is compared to that found for the best individual in the 150th (final) generation of a DE optimization and with that calculated from the transition matrix \mathbf{T} .

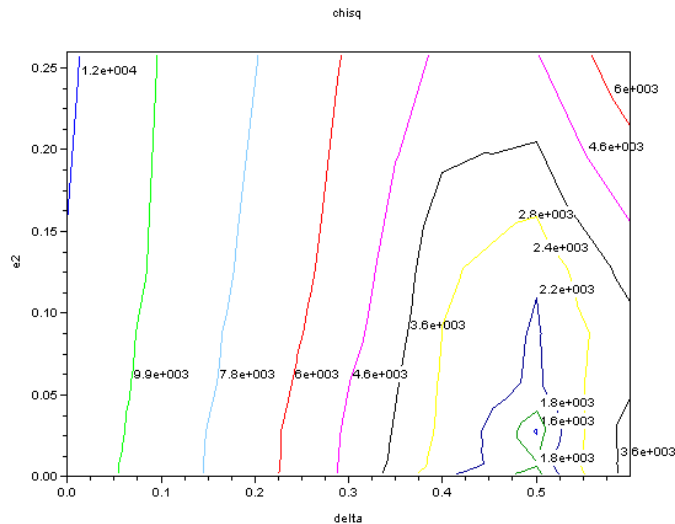


Figure 7.6: Contour plot depicting correlation of e_2 and Δ and fitness (given as $\chi[\text{chi}]$) in the third dimension. The parameters remain uncorrelated until close to the global minimum.

The limiting values of the structural motif probabilities may be found from $\mathbf{p}^n = \mathbf{p}\mathbf{T}^n$, as the number n of layers added approaches infinity. It corresponds to the steady state distribution $\bar{\mathbf{p}}$ where

$$\begin{aligned} \bar{\mathbf{p}} &= \left[\bar{p}(eb_L) \quad \bar{p}(eb_R) \quad \bar{p}(ee) \quad \bar{p}(b_Lb_R) \quad \bar{p}(b_Lb_L) \quad \bar{p}(b_Le) \quad \bar{p}(b_Rb_L) \quad \bar{p}(b_Rb_R) \quad \bar{p}(b_Re) \right] \\ &= \left[0.02194 \quad 0.02194 \quad 0.00033 \quad 0.22797 \quad 0.22797 \quad 0.02194 \quad 0.22797 \quad 0.22797 \quad 0.02194 \right]. \end{aligned}$$

The numerical values of the components of $\bar{\mathbf{p}}$ are obtained from the normalized eigenvector of \mathbf{T} with unit eigenvalue. Table 7.10 gives the four-layer bent left (b_Lb), bent right (b_Rb), mixed eclipsed-bent (eb, be), and eclipsed (ee) motif counts in the crystal as percentages (see Section 5.2, Chemical Modeling System, Growth Modeling, and Reference Data). The b without subscript stands for b_L or b_R .

The statistical distribution of the four-layer motifs in the modeled crystal was also obtained by counting, after 150 generations of DE optimization, the motifs in

Table 7.10: Motif statistics of the reference crystals compared with those from the Markov steady-state distribution and those of the best individual in the 150th (final) generation of a DE optimization averaged over 20 and 300 clones (reported as percent of the full crystal). The b without a subscript indicates that either b_L or b_R stacking is allowed.

Motif Types	Motif Statistics [%]			
	ref.	Markov ^a	20 clones	300 clones
eclipsed (ee)	0.0284(63)	0.033	0.032(7)	0.033(7)
Bent ($b_L b$)	45.75(18)	45.594	45.61(21)	45.61(19)
Bent ($b_R b$)	45.74(18)	45.594	45.52(18)	45.68(20)
Mixed eclipsed & bent (eb)	4.24(11)	4.388	4.42(9)	4.39(10)
Mixed bent & eclipsed (be)	4.24(11)	4.388	4.42(9)	4.39(10)

^a steady-state distribution from Markov transition matrix (table 5.1)

the best individual averaged over 20 and 300 clones. The motif uncertainties were obtained by calculating the standard deviation of each motif count among its clones. Comparing the results for 20 and 300 clones, it is evident that 20 clones are sufficient to obtain reliable values. The four-layer motifs containing only bent arrangements were divided into two categories, $b_L b$ and $b_R b$. For symmetry reasons, the frequency of the two motifs should be the same. For the same reason, the frequency of the motifs eb and be should also be the same. This is shown in Table 7.10. The eclipsed-only motifs, eclipsed in both the second and third layer, accounted for only 0.033% of the total crystal.

The counts from the growth models agree to within standard errors with the counts from the reference crystals and the limiting values from the Markov model (Table 7.10 column 3). This shows that the optimized models truly represent the structural motifs in the crystals from which the reference data set was obtained.

Knowledge of the lengths of repeating motifs (correlation length) is important to ensure that the lot size was chosen appropriately [44]. The correlation length for the eclipsed case is

$$\langle n_e \rangle = \frac{1}{1 - \bar{p}(ee)} = \frac{1}{(1 - 0.00033)} = 1.00033 \quad (7.8)$$

layers; for the bent case $\langle n_b \rangle$ is given by

$$\begin{aligned} \langle n_b \rangle &= \frac{1}{1 - (\bar{p}(b_L b_R) + \bar{p}(b_L b_L) + \bar{p}(b_R b_L) + \bar{p}(b_R b_R))} \\ &= \frac{1}{(1 - 0.9119)} \cong 11.35 \quad (7.9) \end{aligned}$$

layers [78].

Continuing an eclipsed stack is of low probability, with a calculated value of 0.008 ± 0.004 (Table 5.2 column 4). The low probability, e_2 only 2σ above 0, is responsible for the very short average correlation length $\langle n_e \rangle$. In contrast the average length $\langle n_b \rangle$ is about 11.35 layers. Both values indicate that a lot size of 60 layers is sufficient to represent the short-range order in crystals of TBHB.

7.3 β [beta]-NaLaF₄ local structure modeling

The direct and reciprocal space results are described for the Na/La₂ column interactions and the F1 shift model.

7.3.1 Parameter convergence and model fitness

The model parameters are optimized within specified constraints, an initial minimum and maximum and an absolute minimum and maximum that define the world size for the DE optimization. In order to determine the best constraints to use in optimizing the model parameter(s), it is best to try a small test run specifying only an absolute minimum and maximum and examine the parameter convergence to determine if the convergence runs up against either the minimum or maximum value. Then the

model can be rerun with more specific constraints. The constraints for the F1 shift were specified so that shifts both toward and away from La^{3+} were tested in the optimization. The Na2/La2 CU interaction parameter (p1) was defined so that both negative correlation (oppositely oriented neighbors) and positive correlation (same oriented neighbors) were tried during optimization.

Once the appropriate parameter constraints are found the next step is to optimize the model and examine its convergence. In the optimization of this model the limit of the R value is 0.3378(3) (see figure 7.7). The Ising model parameter (p1) has converged to a value of 0.441(18) in 35 generations (see figure 7.8). The positive value for the Na2/La2 interaction parameter with Na2/La2 indicates that like neighbors produce a less favorable MC crystal energy. Continuing to run for more generations would yield little improvement. Since the parameter has such a small standard deviation the model fitness would not improve significantly. The F1 shift parameter has converged to 0.0074(5) in 35 generations (see figure 7.8). Converting 0.007464 shift in $+z$ from fractional coordinates to angstroms corresponds to a 0.06Å shift toward La^{3+} , similar to the 0.07Å shift predicted in [4].

The F1 shift adjusts the bond lengths of Na-F and La-F in the average structure (2.582Å) to 2.654Å and 2.561Å, respectively. The bond valence calculations simply estimate the oxidation state of each ion type within the unit cell. The bond valence of a particular ion is calculated using the atomic positions and distances between its neighbors. Bond valence calculations were done using VaList [91] for the average structure using the F1 original position and the shifted position. The oxidation state of La2 is 2.949 for the shifted F1 bond length versus 2.574 for the average position.

7.3.2 Direct space analysis

The 2-D direct space structure generated by the optimized structural parameters is visualized in Microsoft discovery studio. The direct space structure is built from the CU(s) defined in the model. Looking at the direct space 2-D structure gives an overall

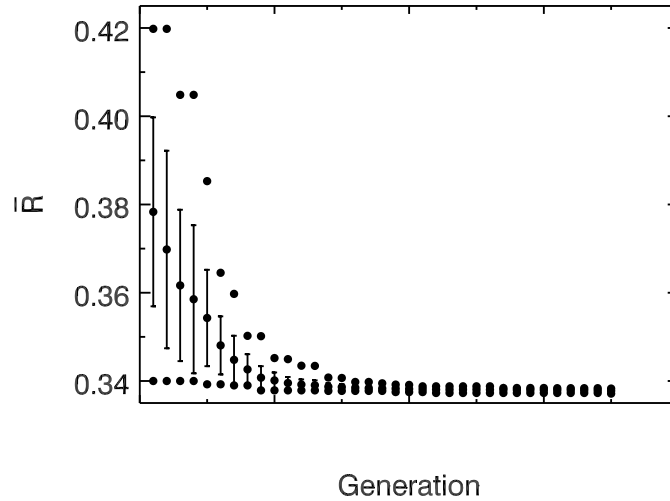


Figure 7.7: The convergence of the fitness function R as a function of generation number.

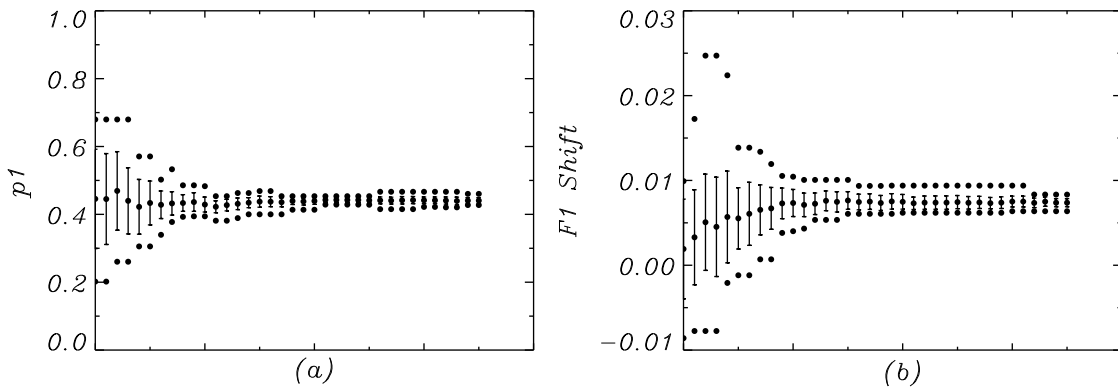


Figure 7.8: Behavior of the model parameters, $p1$, which defines the Na2/La2 CU interaction with another Na2/La2 CU in the neighboring unit cell and the F1 shift (plots a and b) during DE optimization with 24 individuals and 20 clones each. The population mean is indicated by the middle points, and the population standard deviation by the vertical lines. The dots above and below the lines represent the largest and the smallest parameter values in the population.

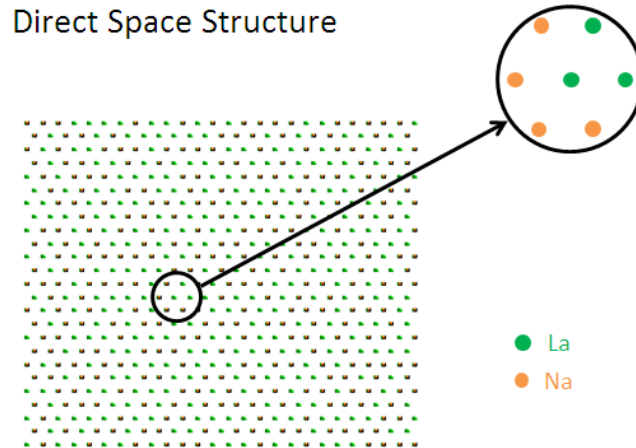


Figure 7.9: A 2-D crystal composed of CUs illustrating the preference for oppositely oriented CUs. The Na₂...La₂ column prefers La₂...Na₂ columns as neighbors. The top view of the crystal shows Na₂ (orange) and La₂ (green) atoms, each with 6 neighbors.

impression of the arrangement of the CUs in the **a,b** plane and assists in determining if the crystal has been built according the structural preferences defined in the model (see figure 7.9).

The structure of the crystal is analyzed by examining the conditional probabilities of the neighbors to determine if they show the preference in the model for oppositely oriented CUs as first neighbors. In the preferences for second neighbors the opposite trend is observed. The neighbor preference begins to fall off by the time the third neighbors are considered (table 7.11). The neighbor coordinates for a 120° lattice are given in figure 7.11.

First neighbors have a probability of CUs in the same orientation of approximately 38–39% and for oppositely oriented CU neighbors the probability was between approximately 61% and 62%. In looking at the preferences for second neighbors this preference the opposite is trend is observed. This indicates that the first neighbor's neighbors (second neighbors) prefer CUs in the same orientation as the CU at the (0, 0, 0) shift but opposite to their first neighbors. This is expected, since the majority of the first neighbors are now oppositely oriented to the CU at (0, 0, 0) then it follows

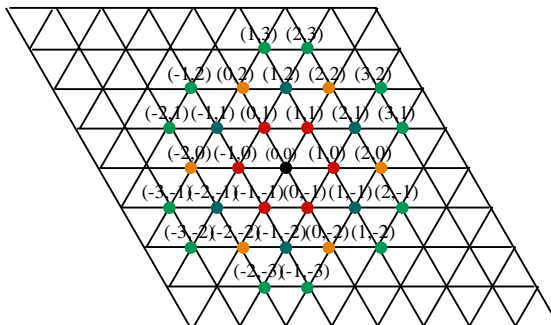


Figure 7.10: 2-D 120 degree lattice showing first neighbors (red), second (blue), third (orange) and fourth (green) with their corresponding coordinates relative to the central atom (black) at the origin (0,0).

that the majority of the second neighbors are now oriented in the same way as the CU at (0,0,0).

The second neighbor preference is then the same orientation as the CU at the (0,0,0) shift with a probability of approximately 56% and the opposite orientation with a probability of around 44%. The correlation for third neighbors is approximately 50% for each orientation, indicating the neighbor correlation does not continue past the third neighbors (table 7.11).

7.3.3 Intensity comparison

The intensities calculated from the fully optimized crystal structures, when compared to the measured intensities reveal how well the local structural disorder has been represented. Moreover, the contribution of a particular element of the structural disorder to the diffuse features in the diffraction pattern is revealed in the optimized structure's calculated intensities.

The systematic differences in the plot of $I_{obs} - I_{calc}$ reveal that aspects of the disorder are not represented in the current model. The intensity modulations not represented in the calculated intensity plot are revealed by the systematic high and low intensities in the difference plot (see figure 7.11). The final average layer R value

Table 7.11: The first, second and third conditional neighbor probabilities using the relative shift vectors (x, y, z) relative to the CU with no shift $(0, 0, 0)$. For example $P(CU, (0, 0, 0)|CU, (2, 1, 0))$ means that given a CU in its original position (x, y, z) the probability of the CU at $(x + 2, y + 1, z)$ being in the same or opposite orientation is given.

Conditional Probabilities		
first neighbor	same	opposite
$P(CU, (0, 0, 0) CU, (1, 0, 0))$	0.380205	0.619795
$P(CU, (0, 0, 0) CU, (0, 1, 0))$	0.393683	0.606317
$P(CU, (0, 0, 0) CU, (1, 1, 0))$	0.392678	0.607322
second neighbor	same	opposite
$P(CU, (0, 0, 0) CU, (1, 2, 0))$	0.559445	0.44055
$P(CU, (0, 0, 0) CU, (2, 1, 0))$	0.561054	0.438946
$P(CU, (0, 0, 0) CU, (-1, 1, 0))$	0.558037	0.441963
third neighbor	same	opposite
$P(CU, (0, 0, 0) CU, (0, 2, 0))$	0.495675	0.504325
$P(CU, (0, 0, 0) CU, (2, 2, 0))$	0.501509	0.498491
$P(CU, (0, 0, 0) CU, (2, 0, 0))$	0.506136	0.493864

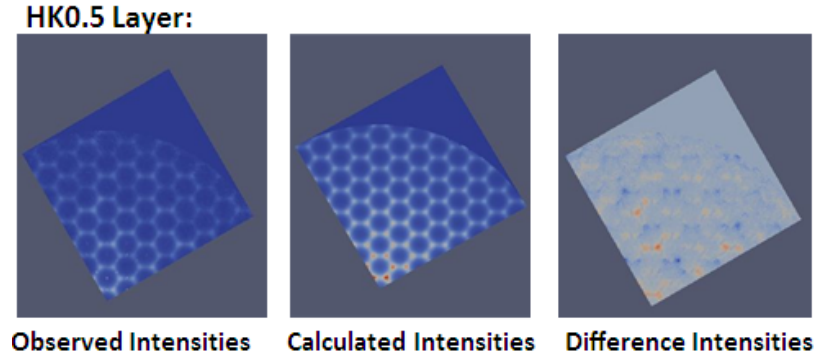


Figure 7.11: From left to right, measured, calculated and difference intensities for the $(H, K, 0.5L)$ layer.

was approximately 0.341, the scale factor was 5.6777×10^{-7} and the background -0.166286 (calculated according to 3.2.3).

The calculated, measured and difference intensities of the $(H, K, 0.5L)$ layer are shown in figure 7.11, the other layers $(H, K, 1.5L)$, $(H, K, 2.5L)$, $(H, K, 3.5L)$ and $(H, K, 4.5L)$ show similar results. In comparing the measured data to the calculated intensities the honey comb pattern of the diffuse layer has clearly been reproduced. The modulation of the diffuse rings does get weaker farther from the origin but is still too high. The bright “nodes” in the hexagonal ring are much sharper in the calculated pattern than the diffuse pattern, however the intensities of the nodes in the hexagonal ring are in the correct position. Examining the difference intensity reveals systematic differences in the calculated and measured intensities. The intensities in the calculated layer are too high at the nodes of the honey comb. The modulation of weaker and stronger intensities is absent in model intensities they are all similar in magnitude. To capture this modulation more correlations need to be taken into account in the model.

Chapter 8

Conclusions and outlook

The results in this section are an expanded version of the paper:

Michels-Clark, T.M.*, Lynch, V.E., Hoffmann, C.M., Hauser, J., Weber, T., Harrison, R., Bürgi, H. B. Analyzing diffuse scattering with Supercomputers. *J. Appl. Crystallogr.*, **46**, 1616-1625

I was lead and corresponding author and my primary contributions to this paper include (i) testing and development of method and analysis (ii) running of computations (iii) writing manuscript and preparing all figures for publication (iv) writing of results analysis code (v) corresponding with editor and referees (vi) final submission.

8.1 Conclusions

A quantitative description of local structure arrangement of single crystals through the interpretation of the diffuse scattering is essential for elucidating the processes that underpin the functionality of many materials with important industrial and biomedical applications. The analysis of diffuse scattering is complex and requires development in two primary areas: (1) diffuse data processing for computations and (2) computational techniques that take full advantage of computational resources. Both aspects were addressed in this work.

Preliminary results analyzing diffuse neutron data through comparison with X-Ray data were described. The initial integration methods were tested for the 100K neutron Bragg structure. The Bragg structure of the neutron and X-Ray data were in agreement.

New ways were presented to quantitatively analyze diffuse scattering and these were applied to the one dimensional stacking disorder described earlier for the organic compound TBHB [18]. Three global optimization algorithms were tested: differential evolution, a general genetic algorithm, and particle swarm optimization. All three algorithms converged to similar parameter values, except for one parameter in the genetic algorithm calculations. Particle swarm optimization was found to be most efficient in the initial stages of optimization. After 150 generations of optimization, the parameter values from differential evolution showed the narrowest range and the best agreement between model and reference diffuse intensities. The testing of many sets of parameters is required in all three global optimization algorithms, but can be significantly accelerated by parallelization: for each parameter set model crystals are calculated on separate compute nodes.

In order to reduce the speckle-type intensity variations inherent in the modeling process, intensities may be calculated either from a single large crystal subdivided into many lots or from several smaller crystals consisting of fewer lots, but constructed from a single set of modeling parameters (clones). Clones are preferred over a large crystal as they allow further parallelization of the calculation. The use of clones also minimizes any bias that might be associated with the random initial layer configuration that seeds the growth of the model crystal in the modeling process. The dependence of the fitness R on the reciprocal square root of the total number of lots (= number of clones times number of lots per clone) was found to be linear, indicating that the behavior of speckle-type intensity variations is analogous to that of experimental noise.

Finally, it was shown that the statistical distribution of four-layer stacking motifs found in the computer simulations was the same within statistical error as that

in the reference crystal. These results were verified theoretically using the steady-state probability distribution resulting from the four- to five-layer Markov transition matrix. The chosen lot size of 60 layers was shown to be sufficient, as the largest correlation length was approximately 11.35 layers for a bent arrangement.

The quantitative analysis protocols reported here for analyzing 1-d diffuse scattering are applicable not only to crystals with stacking disorder but also to 2-d and 3-d types of structural disorder. A more complex disorder might require more model parameters and thus more individuals as well as larger model crystals that encompass the full range of local structure correlations. However, the parallelized global optimization techniques described in this work will also make such structure determinations feasible if the necessary computing resources are available. The calculation of the diffuse intensities could be further optimized by using GPU processors to Fourier transform the scattering density of the disordered crystals as shown by Gutmann [92]. In addition, they may be combined with other modeling techniques, MC modeling [15] and 3D-PDF techniques [93] in particular. The resulting local structure variations may then provide a basis for explaining structure property relationships of disordered materials.

The new computational protocol was tested in the newly developed ZODS modeling and analysis package on β -NaLaF₄ and the results of Aebeischer et al. [4] were quantified.

Bibliography

- [1] C. Hammond, *The Basics of Crystallography and Diffraction*. Oxford University Press, third ed., 2009. xvi, 5, 8
- [2] M. M. Ripoll and F. H. Cano, “Cristalografia,” 2014. xvii, 12
- [3] D. W. Bennett, *Understanding single-crystal X-Ray crystallography*. Wiley-VCH Verlag GmbH & Co. KGaA, 2010. xvii, 5, 13, 14
- [4] A. Aebischer, M. Hostettler, J. Hauser, K. Krämer, T. Weber, H. U. Güdel, and H. B. Bürgi, “Structural and spectroscopic characterization of active sites in a family of light-emitting sodium lanthanide tetrafluorides,” *Angew. Chem. Int. Ed.*, vol. 45, no. 17, pp. 2802–2806, 2006. xvii, xix, 1, 3, 40, 41, 45, 50, 64, 65, 66, 67, 69, 71, 105, 113
- [5] J. Taylor, O. Arnold, J. Bilheaux, A. Buts, S. Campbell, M. Doucet, N. Draper, R. Fowler, M. Gigg, V. Lynch, A. Markvardsen, K. Palmen, P. Parker, P. Peterson, S. Ren, M. Reuter, A. Savici, R. Taylor, R. Tolchenov, R. Whitely, W. Zhou, and J. Zikovsky, “Mantid, a high performance framework for reduction and analysis of neutron scattering data,” *Bulletin of the American Physical Society*, vol. 57, 2012. xviii, 3, 43, 48
- [6] T. M. Michels-Clark, “American chemistry society computers in chemistry division image award,” January 2013. xix, 66
- [7] W. L. Bragg, “The structure of some crystals as indicated by their diffraction of X-rays,” *Proc. Roy. Soc. Lond.*, vol. A89, p. 248, 1913. 1, 9
- [8] Trans. Am. Cryst. Assoc., *Beyond Single-Crystal Structure Determination Interpretation of 3D Disorder Diffuse Scattering*, 2010. 1, 2
- [9] D. Tu, Y. Lui, H. Zhu, R. Li, L. Lui, and X. Chen, “Breakdown of crystallographic site symmetry in lanthanide-doped NaYF₄ crystals,” *Angew. Chem. Int. Ed.*, vol. 52, pp. 1128–1133, 2013. 1, 41, 64

- [10] K. Lonsdale, "X-ray study of crystal dynamics: An historical and critical survey of experiment and theory," *Proc. Phys. Soc.*, vol. 54, no. 4, p. 314, 1942. 1
- [11] G. Honjo, S. Kodera, and N. Kitamura, "Diffuse streak diffraction patterns from single crystals i. general discussion and aspects of electron diffraction diffuse streak patterns," *Journal of the Physical Society of Japan*, vol. 19, no. 3, pp. 351–367. 1
- [12] T. R. Welberry, *Diffuse X-Ray Scattering and Models of Disorder*. New York: Oxford University Press, 2004. 2, 49, 87
- [13] R. B. Neder and T. Proffen, *Diffuse scattering and defect structure simulations: a cook book using the program DISCUSS*. Oxford University Press, 2008. 2
- [14] M. U. Schmidt and J. Glinnemann, "Explanation for the stacking disorder in tris(bicyclo[2.1.1]hexeno)benzene using lattice-energy minimisations," *Z. Kristallogr.*, vol. 227, pp. 805–817, 2012. 2, 50
- [15] T. Proffen and T. R. Welberry, "Analysis of diffuse scattering of single crystals using monte carlo methods," *Phase Transit.*, vol. 67, no. 1, pp. 373–397, 1998. 2, 37, 38, 50, 53, 113
- [16] A. K. Wolf, S. Bruhne, J. Glinnemann, C. Hu, M. T. Kirchner, and M. U. Schmidt, "Local atomic order in sodium p-chlorobenzenesulfonate monohydrate studied by pair distribution function analyses and lattice-energy minimisations," *Z. Kristallogr.*, vol. 227, no. 2, pp. 113–121, 2012. 2
- [17] M. Haase and H. Schafer, "Upconverting nanoparticles," *Angew. Chem. Int. Ed.*, vol. 50, no. 26, pp. 5808–5829, 2011. 3, 40
- [18] H. Birkedal, H. B. Bürgi, K. Komatsu, and D. Schwarzenbach, "Polymorphism and stacking disorder in tris(bicyclo 2.1.1 hexeno)benzene," *J. Mol. Struct.*, vol. 647, no. 1-3, pp. 233–242, 2003. 3, 51, 95, 112

- [19] R. Pynn, "Neutron scattering: A primer," 1990. 5
- [20] C. Giacavazzo, H. L. Monaco, D. Viterbo, F. Scordari, G. Gilli, G. Zanotti, and M. Catti, *Fundamentals of Crystallography, IUCr texts on crystallography No. 2*. Oxford: Oxford University Press, 1992. 5
- [21] C. C. Wilson, *Single crystal neutron diffraction from molecular materials*. Series on Neutron Techniques and Applications , Vol 2, London: World Scientific Publishing Co. Ptc. Ltd., 1st ed. ed., 2000. 5, 43, 44
- [22] M. von Laue, "Röntgenstrahlinterferenzen," *Phys. Z.*, vol. 14, no. 22/23, p. 1075, 1913. 8
- [23] P. P. Ewald, "Zur theorie der interferenzen der röntgenstrahlen in kristallen," *Phys. Z.*, vol. 14, p. 465, 1913. 11
- [24] G. M. Sheldrick, "A short history of *SHELX*," *Acta Cryst. A*, vol. 64, no. 1, pp. 112–122, 2008. 16, 20, 25, 33, 45, 48, 77
- [25] A. L. Patterson, "A direct method for the determination of the components of interatomic distances in crystals," *Z. Kristallogr.*, vol. 90, pp. 517–542, 1935. 20
- [26] D. Harker and J. S. Kasper, "Phases of fourier coefficients directly from crystal diffraction data," *Acta Cryst.*, vol. 1, no. 2, pp. 70–75, 1948. 20
- [27] D. Sayre, "The squaring method: a new method for phase determination," *Acta Cryst.*, vol. 5, no. 1, pp. 60–65, 1952. 20
- [28] V. Elser, "Solution of the crystallographic phase problem by iterated projections," *Phys. Rev.*, vol. A59, no. 3, pp. 201–209, 2003. 20, 23
- [29] M. Shiono and M. M. Woolfson, "Direct-space methods in phase extension and phase determination. i. low-density elimination," *Acta Cryst. A*, vol. A48, no. 4, pp. 451–456, 1992. 20

- [30] C. M. Weeks, G. T. DeTitta, R. Miller, and H. A. Hauptman, “Application of the minimal principle to peptide structures,” *Acta Cryst. D*, vol. D49, no. 1, pp. 179–181, 1993. 20
- [31] G. Oszlányi and A. Süto, “*Ab initio* structure solution by charge flipping,” *Acta Cryst. A*, vol. A60, pp. 134–141, 2004. 20
- [32] G. Oszlányi and A. Süto, “*Ab initio* structure solution by charge flipping. ii. use of weak reflections,” *Acta Cryst. A*, vol. A61, no. 1, pp. 147–152, 2005. 20
- [33] G. Oszlányi and A. Süto, “*Ab initio* neutron crystallography by the charge flipping method,” *Acta Cryst. A*, vol. A63, pp. 156–163, 2006. 20
- [34] A. C. Larson and R. B. V. Dreele, “General structure analysis system (GSAS),” Tech. Rep. LAUR 86-748, Los Alamos National Laboratory, 2000. 20, 75
- [35] B. H. Toby, “*EXPGUI*, a graphical user interface for *gsas*,” *J. Appl. Crystallogr.*, vol. 34, no. 2, pp. 210–213, 2001. 20, 33, 75
- [36] *International tables for X-ray crystallography*, vol. IV. Birmingham: Kynoch Press, 1974. 22, 129
- [37] V. F. Sears, “Neutron scattering lengths and cross sections,” *Neutron News*, vol. 3, no. 3, pp. 29–37, 1992. 22, 42
- [38] B. M. T. Willis, “Crystallography with a pulsed neutron source,” *Z. Kristallogr.*, vol. 209, p. 385, 1994. 27
- [39] V. M. Neild and D. A. Keen, *Diffuse Neutron Scattering from Crystalline Materials*. Oxford Series on Neutron Scattering in Condensed Matter, vol. 14, New York: Oxford University Press Inc., 2001. 29
- [40] M. Chodkiewicz, “Zods-Zürich Oak Ridge disorder simulations.” 2014. 33

- [41] M. Chodkiewicz, L. Ahrenberg, T. Michels-Clark, T. Weber, and H. B. Bürgi, “Zods manual,” 2013. 33
- [42] A. S. Inc., “Discovery studio modeling environment, release 3.5,” 2007. 34
- [43] A. Henderson, *The ParaView Guide, A Parallel Visualization Application*. Kitware Inc., 2007. 34, 39
- [44] T. R. Welberry and B. D. Butler, “Interpretation of diffuse-x-ray scattering via models of disorder,” *J. Appl. Crystallogr.*, vol. 27, pp. 205–231, 1994. 37, 53, 104
- [45] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997. 38, 51, 57, 73
- [46] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometr. Intell. Lab.*, vol. 2, no. 1–3, pp. 37–52, 1987. 39, 99
- [47] R. A. Jalil and Y. Zhang, “Biocompatibility of silica coated NaYF₄ upconversion fluorescent nanocrystals,” *Biomaterials*, vol. 29, no. 30, pp. 4122–4128, 2008. 40
- [48] M. Nyk, R. Kumar, T. Y. Ohulchanskyy, E. J. Bergey, and P. N. Prasad, “High contrast in vitro and in vivo photoluminescence bioimaging using near infrared to near infrared up-conversion in Tm³⁺ and Yb³⁺ doped fluoride nanophosphors,” *Nano Lett.*, vol. 8, no. 11, pp. 3834–3838, 2008. 40
- [49] J. Shan, J. Chen, J. Meng, J. Collins, W. Soboyejo, J. S. Friedberg, and Y. Ju, “Biofunctionalization, cytotoxicity, and cell uptake of lanthanide doped hydrophobically ligated NaYF₄ upconversion nanophosphors,” *J. Appl. Phys.*, vol. 104, no. 9, p. 094308, 2008. 40
- [50] F. Wang, D. K. Chatterjee, Z. Li, Y. Zhang, X. Fan, and M. Wang, “Synthesis of polyethylenimine NaYF₄ nanoparticles with upconversion fluorescence,” *Nanotechnology*, vol. 17, no. 23, p. 5786, 2006. 40

- [51] L. C. Ong, M. K. Gnanasammandhan, S. Nagarajan, and Y. Zhang, "Upconversion: road to el dorado of the fluorescence world," *Luminescence*, vol. 25, no. 4, pp. 290–293, 2010. 40
- [52] S. Wu, G. Han, D. J. Milliron, S. Aloni, V. Altoe, D. V. Talapin, B. E. Cohen, P. J. Schuck, and A. P. Alivisatos, "Non-blinking and photostable upconverted luminescence from single lanthanide doped nanocrystals," *P. Natl. Acad. Sci. USA*, vol. 106, no. 27, pp. 10917–10921, 2009. 40
- [53] S. H. Nam, Y. M. Bae, Y. I. Park, J. H. Kim, H. M. Kim, J. S. Choi, K. T. Lee, T. Hyeon, and Y. D. Suh, "Long-term real-time tracking of lanthanide ion doped upconverting nanoparticles in living cells," *Angew. Chem. Int. Ed.*, vol. 50, no. 27, pp. 6093–6097, 2011. 40
- [54] C. Wang, L. Cheng, and Z. Liu, "Drug delivery with upconversion nanoparticles for multi-functional targeted cancer cell imaging and therapy," *Biomaterials*, vol. 32, no. 4, pp. 1110 – 1120, 2011. 40
- [55] J. Shan, S. J. Budijono, G. Hu, N. Yao, Y. Kang, Y. Ju, and R. K. Prud'homme, "Pegylated composite nanoparticles containing upconverting phosphors and meso-tetraphenyl porphine (tpp) for photodynamic therapy," *Adv. Funct. Mater.*, vol. 21, no. 13, pp. 2488–2495, 2011. 40
- [56] L. Cheng, K. Yang, Y. Li, J. Chen, C. Wang, M. Shao, S.-T. Lee, and Z. Liu, "Facile preparation of multifunctional upconversion nanoprobe for multimodal imaging and dual-targeted photothermal therapy," *Angew. Chem. Int. Ed.*, vol. 50, no. 32, 2011. 40
- [57] R. Scheps, "Upconversion laser processes," *Prog. Quant. Electron.*, vol. 20, no. 4, pp. 271 – 358, 1996. 40
- [58] W. Lenth and R. M. Macfarlane, "Excitation mechanisms for upconversion lasers," *J. Lumin.*, vol. 45, no. 1–6, pp. 346 – 350, 1990. 40

- [59] M. F. Joubert, S. Guy, and B. Jacquier, “Model of the photon-avalanche effect,” *Phys. Rev. B*, vol. 48, pp. 10031–10037, 1993. 40
- [60] A. Shalav, B. S. Richards, T. Trupke, K. W. Krämer, and H. U. Güdel, “Application of $\text{NaYF}_4:\text{Er}^{3+}$ up-converting phosphors for enhanced near-infrared silicon solar cell response,” *Appl. Phys. Lett.*, vol. 86, no. 1, p. 013505, 2005. 40
- [61] P. G. Kik and A. Polman, “Cooperative upconversion as the gain-limiting factor in Er doped miniature Al_2O_3 optical waveguide amplifiers,” *J. Appl. Phys.*, vol. 93, no. 9, pp. 5008–5012, 2003. 40
- [62] C. Strohhofer and A. Polman, “Relationship between gain and Yb^{3+} concentration in $\text{Er}^{3+}-\text{Yb}^{3+}$ doped waveguide amplifiers,” *J. Appl. Phys.*, vol. 90, no. 9, pp. 4314–4320, 2001. 40
- [63] E. Downing, L. Hesselink, J. Ralston, and R. Macfarlane, “A three-color, solid-state, three-dimensional display,” *Science*, vol. 273, no. 5279, pp. 1185–1189, 1996. 40
- [64] F. Auzel, “Upconversion and anti-stokes processes with f and d ions in solids,” *Chem. Rev.*, vol. 104, no. 1, pp. 139–174, 2004. 40
- [65] K. W. Krämer, D. Biner, G. Frei, H. U. Güdel, M. P. Hehlen, and S. R. Lüthi, “Hexagonal sodium yttrium fluoride based green and blue emitting upconversion phosphors,” *Chem. Mat.*, vol. 16, no. 7, pp. 1244–1251, 2004. 40, 41
- [66] A. Aebischer, S. Heer, D. Biner, K. Krämer, M. Haase, and H. U. Güdel, “Visible light emission upon near-infrared excitation in a transparent solution of nanocrystalline $\beta\text{-NaGdF}_4:\text{Yb}^{3+}, \text{Er}^{3+}$,” *Chem. Phys. Lett.*, vol. 407, no. 1–3, pp. 124–128, 2005. 40
- [67] N. Menyuk, K. Dwight, and J. Pierce, “ $\text{NaYF}_4:\text{Yb}, \text{Er}$ —an efficient upconversion phosphor,” *Appl. Phys. Lett.*, vol. 21, no. 4, pp. 159–161, 1972. 41

- [68] T. Kano, T. Suzuki, A. Suzuki, and S. Minagawa, "Fabrication of efficient up-conversion type green LED using NaYF₄-Yb,Er phosphor," *J. Electrochem. Soc.*, vol. 120, no. 3, p. C87, 1973. 41
- [69] J. H. Burns, "Crystal structure of hexagonal sodium neodymium fluoride and related compounds," *Inorg. Chem.*, vol. 4, no. 6, pp. 881–886, 1965. 41
- [70] T. Worlton, A. Chatterjee, J. Hammonds, C. Bouzek, D. Mikkelson, R. Mikkelson, M. Miller, B. Serum, and P. Peterson, "Scientific review: New software for neutron scattering data visualization," *Neutron News*, vol. 15, no. 3, pp. 14–15, 2004. 42
- [71] J. Zikovsky, P. F. Peterson, X. P. Wang, M. Frost, and C. Hoffmann, "Crystalplan: an experiment-planning tool for crystallography," *J. Appl. Cryst.*, vol. 44, no. 2, pp. 418–423, 2011. 43
- [72] A. J. Schultz, K. Srinivasan, R. G. Teller, J. M. Williams, and C. M. Lukehart, "Single-crystal, time-of-flight, neutron-diffraction structure of hydrogen cis-diacetyltetracarbonylrhenate, [cis-(OC)₄Re(CH₃CO)₂]H: a metallaacetylacetonone molecule," *J. Am. Chem. Soc.*, vol. 106, no. 4, pp. 999 – 1003, 1984. 45, 75
- [73] B. M. T. Willis and A. W. Pryor, *Thermal vibrations in crystallography*. Cambridge University Press, 1975. 50
- [74] T. R. Welberry and D. J. Goossens, "The interpretation and analysis of diffuse scattering using monte carlo simulation methods," *Acta Crystallogr. A*, vol. 64, pp. 23–32, 2008. 50, 61
- [75] A. J. C. Wilson, *X-ray Optics: the diffraction of X-rays by finite and imperfect crystals*. Methuen's monographs on physical subjects, London: Methuen, 2nd ed. ed., 1962. 51
- [76] J. H. Holland, *Adaptation in natural and artificial systems*. The University of Michigan Press, 1975. 51, 57

- [77] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Net.*, 1995. 51, 59
- [78] H. B. Bürgi, M. Hostettler, H. Birkedal, and D. Schwarzenbach, "Stacking disorder: the hexagonal polymorph of tris(bicyclo[2.1.1]hexeno)benzene and related examples," *Z. Kristallogr.*, vol. 220, no. 12, pp. 1066–1075, 2005. 51, 52, 104
- [79] J. Hauser, T. Weber, and H. B. Bürgi, "private communication." 2009. 53
- [80] D. E. Goldberg, "Genetic and evolutionary algorithms come of age," *Commun. ACM*, vol. 37, no. 3, pp. 113–119, 1994. 57
- [81] C. B. Lucasius and G. Kateman, "Understanding and using genetic algorithms .1. concepts, properties and context," *Chemometr. Intell. Lab.*, vol. 19, no. 1, pp. 1–33, 1993. 57
- [82] M. Srinivas and L. M. Patnaik, "Genetic algorithms - a survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994. 57
- [83] K. Gallagher and M. Sambridge, "Genetic algorithms—a powerful tool for large-scale nonlinear optimization problems," *Comput. Geosci.*, vol. 20, no. 7–8, pp. 1229–1236, 1994. 57
- [84] S. Forrest, "Genetic algorithms—principles of natural-selection applied to computation," *Science*, vol. 261, no. 5123, pp. 872–878, 1993. 57
- [85] F. Gray, "Pulse code communication," 1953. 57
- [86] T. Weber and H. B. Bürgi, "Determination and refinement of disordered crystal structures using evolutionary algorithms in combination with monte carlo methods," *Acta Crystallogr. A*, vol. 58, pp. 526–540, 2002. 57

- [87] H. B. Bürgi, J. Hauser, T. Weber, and R. B. Neder, "Supramolecular architecture in a disordered perhydrotriphenylene inclusion compound from diffuse x-ray diffraction data," *Cryst. Growth Des.*, vol. 5, no. 6, pp. 2073–2083, 2005. 61
- [88] T. Proffen and T. R. Welberry, "Analysis of diffuse scattering via the reverse monte carlo technique: A systematic investigation," *Acta Crystallogr. A*, vol. 53, pp. 202–216, 1997. 61
- [89] M. Sutton, S. G. J. Mochrie, T. Greytak, S. E. Nagler, L. E. Berman, G. A. Held, and G. B. Stephenson, "Observation of speckle by diffraction with coherent x-rays," *Nature*, vol. 352, no. 6336, pp. 608–610, 1991. 87
- [90] H. D. Hamby, "A review of techniques for parameter sensitivity analysis of environmental-models," *Environ. Monit. Assess.*, vol. 32, no. 2, pp. 135–154, 1994. 93
- [91] A. S. Wills, "Valist." 105
- [92] M. J. Gutmann, "Accelerated computation of diffuse scattering patterns and application to magnetic neutron scattering," *J. Appl. Crystallogr.*, vol. 43, pp. 250–255, 2010. 113
- [93] T. Weber and A. Simonov, "The three dimensional pair distribution function analysis of disordered single crystals: basic concepts," *Z. Kristallogr.*, vol. 227, pp. 238–247, 2012. 113

Appendix

A. 1 Two dimensional occupational disorder analysis

A. 1.1 General concepts

A program to simulate occupational disorder of two atoms (type A and B) on a 2 dimensional hexagonal lattice was written in C++. This disorder example provides a valuable understanding for the concepts of disorder simulation. A 2-D crystal of user defined size is randomly generated and then optimized using MC algorithm. The crystal is a randomly generated matrix of A type atoms (0) and B type atoms (1) of user defined dimension $n \times m$. First neighbors defined on a $\gamma = 60^\circ$ lattice given as $x[6, 2] = [(1, -1), (0, 1), (1, 0), (0, -1), (-1, 0), (-1, 1)]$ and the second neighbors; $y[6, 2] = [(1, 1), (-1, 2), (-2, 1), (-1, -1), (1, -2), (2, -1)]$ (figure A.1). The crystal energy is given as the sum of all first and second neighbor interaction energies between the two atoms A and B and $x[6, 2]$ and $y[6, 2]$ are arrays of first and second neighbor relative position coordinates;

$$E = \sum_{k=1}^n \sum_{l=1}^m \sum_{i=1}^6 (a_{k,l} a_{k+x_i,0,l+x_{i,1}} + b_{k,l} b_{k+x_i,0,l+x_{i,1}} + a_{k,l} b_{k+x_i,0,l+x_{i,1}} + b_{k,l} a_{k+x_i,0,l+x_{i,1}}) \\ + \sum_{k=1}^n \sum_{l=1}^m \sum_{i=1}^6 (a_{k,l} a_{k+y_i,0,l+y_{i,1}} + b_{k,l} b_{k+y_i,0,l+y_{i,1}} + a_{k,l} b_{k+y_i,0,l+y_{i,1}} + b_{k,l} a_{k+y_i,0,l+y_{i,1}})$$

The given atom type and the type of its first and second neighbor is determined by a set of conditional statements. The above crystal energy equation gives the formulation for finding the crystal energy given user defined first and second neighbor interaction energies; AA, BB and AB where the energy of $BA = AB$. Boundary conditions are applied so that if a given atom's neighbor is outside the defined crystal (index outside randomly generated crystal matrix) then its position is reflected into the same relative position inside the crystal. The relative position is defined in terms of the first and second neighbor coordinates relative to an atom at the origin for $\gamma = 60^\circ$ as shown in (figure A.1).

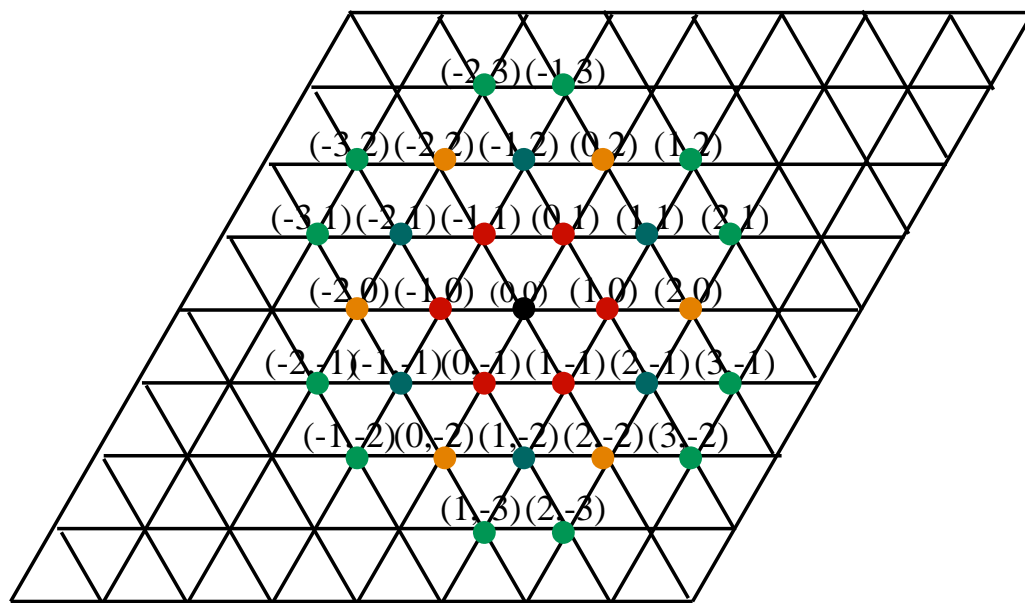


Figure A.1: 2-D sixty degree lattice showing first neighbors (red), second (blue), third (orange) and fourth (green) with their corresponding coordinates relative to the central atom (black) at the origin (0,0).

The crystal energy is calculated and the contents are optimized via a Monte Carlo (MC) process which minimizes the crystal energy through swaps from a randomly generated reservoir (a 1-D vector of A and B type atoms) of user defined size. Each swap is defined as a Monte Carlo step. The change in energy is given by $\Delta E = E_i - E_{i+1}$, where i is the i^{th} MC step and $i + 1$ is the subsequent MC step. The swap is accepted if $\Delta E = E_i - E_{i+1} \leq 0$. If $\Delta E = E_i - E_{i+1} > 0$ then the Boltzmann probability,

$$P(E) = e^{-\Delta E \frac{k\beta}{T}} \quad (1)$$

where ($T = \frac{1}{k\beta}$) is calculated and compared to a random number generated between 0 and 1. If $P(E)$ is less than the random number the swap is accepted; otherwise it is rejected. This process is repeated for a user defined number of MC steps and the final MC optimized 2-D crystal made of A (0) and B (1) atoms and user defined $n \times m$ dimension is printed in a 2-D, 60° grid of 0, 1 (shown in output print out below).

The pair correlation statistics of the crystal before and after optimization are calculated for first, second, third and fourth neighbors using the definition of conditional probability

$$P(A|B) = \frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n (A_i + B_i)}. \quad (2)$$

This means that given an atom of type A (or B) the number of first, second, third or fourth neighbors that are of type A (or B, depending the condition calculated) divided by the total number of first, second, third or fourth neighbors determine the associated pair correlation. For example, $P(A^1|B)$ means that given an atom of type B the total number of first neighbors that are of type A divided by the total number of first neighbors yields the pair correlation of B with first neighbor atoms of type A. All of the conditional probabilities are listed in the output file below calculated on both the optimized and unoptimized crystal. This provides a sanity check for the expected structural output based on the interaction energies. The smaller interaction energies between two atoms should result in a greater probability of atoms of this

type being neighbors. The correlation for this preference should get weaker by the fourth neighbors. For example, in the output file below the interaction energies for $AA = 1$, $BB = 3$ and $AB = 0.5$ and all second neighbor interaction energies are set to 0. The conditional probabilities of the optimized crystal correspond to a greater percentage of $P(A^1|B) \approx 72\%$ and $P(B^1|A) \approx 52.3\%$ as compared to $P(A^1|A) \approx 47.2\%$ and $P(B^1|B) \approx 27\%$. The same condition probabilities before optimization are all approximately equivalent. The probabilities for second, third and fourth neighbors show the interaction is almost nonexistent past the second neighbors and the third and fourth neighbor probabilities are almost the same.

Once the crystal is optimized the optimized crystal file is read and the x-ray or neutron intensities are calculated on a grid of user defined size. Each intensity is calculated at the grid positions determined by the input origin and step size. The neutron scattering lengths are input by the user. The x-ray scattering factors are calculated using normalized scattering curves fitted to a 9 coefficient equation that parameterizes the non-dispersive part of the atomic scattering factor for neutral atoms as a function of $\frac{\sin \theta}{\lambda}$ by Cromer, D. T. and Waber, J. T. using the unpublished wave functions of J. B. Mann described in [36] page 71. This code takes the user entered atoms types for A and B and fits these in the scattering factor function below.

$$f^0\left(\frac{\sin \theta}{\lambda}\right) = \sum_{i=1}^4 a_i * e^{-b_i\left(\frac{\sin \theta}{\lambda}\right)^2} + c \quad (3)$$

given the fitting coefficients a_i , b_i and c with known wavelength λ , the scattering factor for a given atom can be calculated for any $\frac{\sin \theta}{\lambda} = \frac{|x\mathbf{a}^* + y\mathbf{b}^*|}{2}$, where \mathbf{a}^* and \mathbf{b}^* are the 2-D reciprocal hexagonal lattice vectors and x and y define the grid position. The grid position (x, y) is determined by $(x_0 + h * s, y_0 + k * s)$, where s is the step size and (x_0, y_0) defines the grid origin and h, k are integer indices in the x and y dimension of the crystal (or lot). Finally, to calculate the grid position on a hexagonal lattice with $\gamma = 60^\circ$ the (x, y) , the fractional coordinates position is converted to Cartesian

coordinates. The new Cartesian coordinates, (u, v) are given by

$$(u, v) = (x_0 + h * s) \mathbf{a}^* \begin{pmatrix} \cos(\frac{\pi}{6}) \\ \cos(\frac{\pi}{3}) \end{pmatrix} + (y_0 + k * s) \mathbf{b}^* \begin{pmatrix} \cos(\frac{\pi}{2}) \\ \cos(0) \end{pmatrix} \quad (4)$$

using $\mathbf{a}^* = \frac{\mathbf{b} \times \mathbf{c}}{V} = \frac{b \cdot c \sin(\frac{\pi}{2})}{V} \hat{i}$ and $\mathbf{b}^* = \frac{\mathbf{a} \times \mathbf{c}}{V} = \frac{a \cdot c \sin(\frac{\pi}{2})}{V} \hat{j}$, which yields;

$$u = (x_0 + h * s) \frac{b \cdot c}{V} \cos(\frac{\pi}{6}) \quad (5)$$

$$v = (x_0 + h * s) \frac{b \cdot c}{V} \cos(\frac{\pi}{3}) + (y_0 + k * s) \frac{a \cdot c}{V} \quad (6)$$

where a, b and c are the lattice parameters of the crystal. In the case of NaLaF_4 these values are $a = b = 6.18639 \text{ \AA}$ and $c = 3.83388 \text{ \AA}$, when calculated from the ambient X-Ray intensity data.

Using the above equations for (u, v) , $\frac{\sin \theta}{\lambda}$ can easily be calculated as $\frac{\sqrt{u^2 + v^2}}{2}$. This value is then used to calculate the scattering factor for atoms A and B in the function 3 at each grid position for all atoms in the crystal. These X-ray scattering factors are then used to calculate the structure factor at each step on the grid.

The structure factor is given as;

$$F = \sum_i^n \sum_j^n 2\pi * \iota(f_A + f_B) * e^{(x_0 + h * s) * i + (y_0 + k * s) * j} \quad (7)$$

where f_A and f_B are the x-ray scattering factor or neutron scattering length. The neutron scattering lengths are constant and do not depend on $\frac{\sin \theta}{\lambda}$. The structure factor is calculated for each lot and summed over all lots. The user inputs the lot size and number of lots. Each lot is randomly selected from the larger optimized crystal. Boundary conditions are applied to the lots so that the entire lot is contained within the crystal. An example output file is shown below for neutron intensities with and without lots.

Output file: This file contains the original randomly generated crystal, optimized


```
12677.5
3443.86
1510.54
3452.6
23999.8
12677.5
3443.86
```

The neutron intensities of your crystal calculated with 5000 lots of size 10 are:

```
660156
8.20676
385.543
385.543
8.20676
660156
8.20676
385.543
385.543
8.20676
618.75
115.635
1659
3545.69
156.669
618.75
115.635
1659
3545.69
156.669
618.75
1910.56
1121.87
3049.58
3797.25
618.75
1910.56
1121.87
3049.58
3797.25
618.75
3797.25
3049.58
1121.87
1910.56
618.75
3797.25
3049.58
1121.87
1910.56
618.75
156.669
3545.69
1659
115.635
618.75
156.669
3545.69
1659
115.635
660156
8.20676
385.543
385.543
8.20676
660156
8.20676
385.543
385.543
8.20676
618.75
115.635
1659
3545.69
156.669
618.75
115.635
1659
3545.69
156.669
618.75
1910.56
1121.87
3049.58
3797.25
618.75
1910.56
1121.87
3049.58
3797.25
618.75
1910.56
1121.87
3049.58
3797.25
618.75
1910.56
1121.87
3049.58
3797.25
618.75
115.635
1659
115.635
156.669
3545.69
1659
115.635
```

A. 1.2 C++ code

```
/*Created By: Tara Michels-Clark
 * July 25, 2011
 * Name: 2-DCode.cpp
 */

/*****
 * Purpose of the Program:
 Randomly places atoms of type A and B to origin of unit cells of crystal of user defined size.
 The energy of the crystal is then calculated by summing over the user defined for first and
```

second neighbor interaction energies. A randomly chosen unit cell's atom is replaced by an atom of in the reservoir and the new energy is calculated. If the change in energy is less than or equal to 0 then then change is accepted. If the change in energy is greater than 0 then the Boltzman probability $P(E)=\exp(-\text{delta}E)/(T=1/k*\text{Beta})$ is calculated and a random number generated between 0 and 1. If this Boltzman probability is less than the random the change is accepted otherwise it is not. This process is repeated for a user defined number of MC steps and the final "optimized" A and B crystal is printed according to unit cell number in a and b. The neutron intensities are calculated on a 2-D hexagonal grid, where the step size and grid size are defined by the user. Lots can also be calculated, lot size and number of lots are defined by the user.

* INPUT: Number of Monte Carlo Steps; Crystal Size (2-D); first and second interaction energies, neutron scattering lengths of atoms A and B, lot size, number of lots, grid size, grid step size.

* OUTPUT: MC Crystal energy before and after MC cycles; crystal with atom types A and B; first through fourth neighbor conditional probabilities before and after MC; Neutron intensities before and after lot calculations

```

*****/
#include <iostream>
#include<cstdlib>
#include<time.h>
#include <limits>
#include <fstream> //for I/O
#include <string>
#include <vector> //Vector class
#include <cstdlib> //to get Vector from c++ standard Library
#include<cmath> //for the Boltzman probability
#include<complex> //complex number for functions
#define _USE_MATH_DEFINES //to use transcendental pi without having to calculate it

using namespace std;
// matrix vector multiply using vectors from the STL

typedef vector<int> Vec; //define vector type
typedef vector<Vec> Mat; //define Matrix (vector of vectors)
typedef vector<double>V;
typedef vector <complex<double>> CD;
typedef vector <CD> CMat;

/*****
 * Function Name: DisplayCrystal()
 * Input: Matrix
 * output: void
 * Description: Displays the Crystal one row at a time 0 for atom of type A and 1 for atom of
 * type B.
 *****/

void DisplayCrystal (Mat & in )
{
    unsigned int i,j,k;
    cout << "Matrix: " << endl;
    for( i = 0; i < in.size(); i++)
    {
        for( j = 0; j < in[i].size(); j++)
        {
            if(in[i][j]==0){
                cout<< "+" << " ";
            }else{
                cout<< "0" <<" ";
            }
        }
    }
}

```

```

        cout<<endl;
//shifts view to the type of hexagonal crystal the program makes a,b angle is 60 degrees
        for(k=0;k<i+1;k++){

                cout<<" ";

        }

}

/*****
* Function Name: RandomInput()
* Input: Matrix, unsigned int, unsigned int
* output: void
* Description: Generates random atoms of type A (0) and type B (1) in 2-D crystal (array)
*****/
Mat RandomInput( Mat & a, int n, int m )
{
    int i, j;
    int mmin = 0;
    int mmax = 1;
    // Allocate space for the vectors
    Vec row1(n);
    for(i = 0; i <m; i++)
    {
        for(j = 0; j <n; j++){
            row1[j] = mmin + (rand() & mmax);
            a.push_back(row1);
        }
        return(a);
    }
}

void Count( Mat & a, int n, int m, int &A, int &B )
{
    int i, j;

    Vec row1(n);
    for(i = 0; i <m; i++)
    {
        for(j = 0; j <n; j++){

            if(a[i][j]==0){

                B++;
            }else{

                A++;
            }
        }
    }
}

void ReservoirCount(Vec & r, int &Ares, int &Bres){
    unsigned int i;
    for(i=0; i<r.size(); i++){
        if(r[i]==0){
            Bres++;
        }else{
            Ares++;
        }
    }
}
}

```

```

void ReadOptimizedCrystalFile( ifstream &in , Mat &MC)
{
    if( in.good() )
    {
        string temp;
        in >> temp;
        getline(in,temp);
        getline(in,temp);
        in >> temp >> temp >> temp;
        int a,b; //crystal dimensions (axb)
        in>> a >> temp >> b;
        getline(in,temp);
        getline(in,temp);
        int i,j;
        MC.clear();
        Vec row1(b);

        for(i = 0; i < a; i++)
        {
            for(j = 0; j <b; j++){
                in>>temp;
                if(temp==" ")

                    row1[j] = 0;
                else
                    row1[j]=1;
            }
            MC.push_back(row1);
        }
    }
}

/*****
* Function Name: Reservoir()
* Input: Matrix, int
* output: Vec
* Description: Generates a vector of user defined size of random 0s and 1s for B and A
* these are kept in order to swap with atoms in the crystal
*****/

Vec Reservoir(int s)
{
    int row1=0;
    Vec r;
    int i;
    int nmin=0;
    int nmax=1;
    for (i=0;i<s;i++){
        row1=nmin+(rand() & nmax);
        r.push_back(row1);
    }
    return(r);
}

/*****
* Function Name: BoundaryConditions()
* Input: Matrix, int, int, Mat, int, int

```

```

* output: int
* Description: If specified neighbor is outside of crystal then it will shift to the relative
position inside the crystal. This is a general function which will work for any specified
neighbors. Returns the value of the neighbor at the new index.
*****/

int BoundaryConditions( int f, int _g, Mat a, int n, int m){
    int g=_g;
    if(f<0){
        f=m+f;
    }
    if(f>m){
        f=f-m;
    }
    if (f>=0 && f<m){
        f=f;
    }
    if(g<0){
        g=n+g;
    }
    if(g>=n){
        g=g-n;
    }
    if(g>=0 && g<n){
        g=g;
    }
    return(a[f][g]);
}

/*****
* Function Name: CalculateCrystalEnergy()
* Input: Matrix, Mat, int, int, double, double double, double, double, double
* output: double
* Description: Takes the crystal and crystal dimensions and all 6 interaction energies
for the first and second neighbors and calculates the crystal energy. This is returned
as a double.
*****/

double CalculateCrystalEnergy(Mat &a, int n, int m, double AA1,double BB1, double AB1,
double AA2,double BB2,double AB2){

    double IE=0; //initialize interaction energy
    double CE=0; //initialize crystal energy
    int p=0; //First and second neighbor loop iterator
    //relative positions of the first neighbors
    int x[6][2]={{1,-1},{0,1},{1,0},{0,-1},{-1,0},{-1,1}};
    //relative positions of the second neighbors
    int y[6][2]={{1,1},{-1,2},{-2,1},{-1,-1},{1,-2},{2,-1}};
    //loops through crystal and calculates interaction energies for all first and second neighbors
    for( int i = 0; i < m; i++)
    {
        for( int j = 0; j < n; j++)
        {
            for(p=0;p<6;p++)
            {
                if(a[i][j]==1 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
                    IE=AA1;
                } if(a[i][j]==0 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
                    IE=BB1;
                }
            }
        }
    }
}

```



```

    } if (a[i][j]==1&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
        IE=AB1;
    }if(a[i][j]==0&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
        IE=AB1;
    }CE+=IE;
        if(a[i][j]==1 && BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==1){
            IE=AA2;
        } if(a[i][j]==0 && BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==0){
            IE=BB2;
        } if (a[i][j]==1&&BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==0){
            IE=AB2;
        }if(a[i][j]==0&&BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==1){
            IE=AB2;
        }
//Crystal energy is the sum of all first and second neighbor energies
    }
    CE+=IE;
}
}
return(CE/2);
}

/*****
 * Function Name: MakeChangei()
 * Input: int,int
 * output: int
 * Description: randomly generates a new ith index to be selected to swap in the crystal
 *****/

int MakeChangei( int m){
    unsigned int i_index;
    i_index=rand()%m;
    return(i_index);
}

/*****
 * Function Name: MakeChangej()
 * Input: int,int
 * output: int
 * Description: randomly generates a new jth index to be selected to swap in the crystal
 *****/

int MakeChangej(int n){
    unsigned int j_index;
    j_index=rand()%n;
    return(j_index);
}

/*****
 * Function Name: MakeChangeReservoir()
 * Input: int
 * output: int
 * Description: randomly generates a new kth index in the reservoir to be selected for swap.
 *****/

int MakeChangeReservoir(int s){
    unsigned int k;
    k=rand()%s;
    return(k);
}

```

```

}

/*****
* Function Name: swap()
* Input: int, int
* output: void
* Description: Makes the swap with atom at position [i][j] in crystal with atom in reservoir
at position [k]. *note: positions are randomly chosen
*****/

void swap(int &i_index, int &j_index, int &k, Mat &a, Vec &r)
{
    int temp =a[i_index][j_index];
    a[i_index][j_index]=r[k];
    r[k]=temp;
    Mat changed_a=a;

}

double LocalEnergy(Mat &a, int &i, int &j, int n, int m, double AA1,double BB1,
double AB1,double AA2,double BB2, double AB2 ){
    double IE=0; //initialize interaction energy
    double CE=0;//initialize local energy
    int p=0;//First and second neighbor loop iterator
    //relative positions of the first neighbors
    int x[6][2]={{1,-1},{0,1},{1,0},{0,-1},{-1,0},{-1,1}};
    //relative positions of the second neighbors
    int y[6][2]={{1,1},{-1,2},{-2,1},{-1,-1},{1,-2},{2,-1}};
    for (p=0;p<6;p++)
    {
        if(a[i][j]==1 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
            IE=AA1;

            } if(a[i][j]==0 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
                IE=BB1;

            } if (a[i][j]==1&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
                IE=AB1;

            }if(a[i][j]==0&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
                IE=AB1;

            }CE+=IE;

        if(a[i][j]==1 && BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==1){
            IE=AA2;

        } if(a[i][j]==0 && BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==0){
            IE=BB2;

        } if (a[i][j]==1&&BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==0){
            IE=AB2;

        }if(a[i][j]==0&&BoundaryConditions(i+x[p][0],j+y[p][1],a,n,m)==1){
            IE=AB2;

        }

        CE+=IE; //Crystal energy is the sum of all first and second neighbor energies
    }
}

```

```

    return(CE);
}

/*****
* Function Name: MonteCarlo()
* Input: Mat, Vec, int, int, unsigned int, double, double, double, double, double, double
* output: double
* Description: Returns the calculated crystal energy for each MC step.
  Calls MakeChangei(), MakeChangej, MakeChangeReservoir(), swap() and
  CalculateCrystalEnergy().
*****/

void MonteCarlo(Mat &a, Vec &r, int n, int m, unsigned int s, double &E, double AA1, double BB1,
double AB1,double AA2,double BB2,double AB2)
{

    int ic ,jc ,rc; //for random reservoir and Mat element swap
    ic=MakeChangei(m);
    jc=MakeChangej(n);
    rc=MakeChangeReservoir(s);
    //Calculates local energy of A[i][j] before change
    double LE=LocalEnergy(a,ic,jc,n,m,AA1,BB1,AB1,AA2,BB2,AB2);
    swap(ic,jc,rc,a,r);
    //Calculates local energy of A[i][j] after change
    double LEC=LocalEnergy(a,ic,jc,n,m,AA1,BB1,AB1,AA2,BB2,AB2);
    double DE=LEC-LE;
    double prob=exp(-DE);
    double Random=((double) rand() / (RAND_MAX+1)) ; //for random number between (0,1);

    if(DE<=0||prob>Random){ //keeps swaped atoms if delta is favorable
        a=a;
        r=r;
        E=E-LE+LEC;
    }else{
        swap(ic,jc,rc,a,r); //swaps back to original crystal and reservoir
    }
}

void CountNeighbors(Mat &a, int &n, int &m, double &AA1_neigh, double &BB1_neigh, double &AB1_neigh,
double &BA1_neigh, double &AA2_neigh, double &BB2_neigh, double &AB2_neigh, double &BA2_neigh,
double &AA3_neigh, double &BB3_neigh, double &AB3_neigh, double &BA3_neigh, double &AA4_neigh,
double &BB4_neigh, double &AB4_neigh, double &BA4_neigh)
{
    //relative positions of the first neighbors
    int w[6][2]={{1,-1},{0,1},{1,0},{0,-1},{-1,0},{-1,1}};
    //relative positions of the second neighbors
    int x[6][2]={{1,1},{-1,2},{-2,1},{-1,-1},{1,-2},{2,-1}};
    //relative positions of the third neighbors
    int y[6][2]={{-2,0},{2,0},{0,-2},{-2,2},{2,-2},{0,2}};
    //relative positions of the fourth neighbors
    int z[12][2]={{-3,1},{2,-1},{-3,2},{-2,3},{-1,3},{1,2},{2,1},{3,-1},{3,-3},
{2,-3},{1,-3},{-1,-2}};
    int p=0;
    int k=0;
    for( int i = 0; i < m; i++ )
    {
        for( int j = 0; j < n; j++ )
        {

```

```

if(i<j){
    for(p=0;p<6;p++)
    {
if(a[i][j]==1 && BoundaryConditions(i+w[p][0],j+w[p][1],a,n,m)==1){
        AA1_neigh++;

    } if(a[i][j]==0 && BoundaryConditions(i+w[p][0],j+w[p][1],a,n,m)==0){
        BB1_neigh++;

    } if (a[i][j]==1&&BoundaryConditions(i+w[p][0],j+w[p][1],a,n,m)==0){
        AB1_neigh++;

    } if(a[i][j]==0&&BoundaryConditions(i+w[p][0],j+w[p][1],a,n,m)==1){
        BA1_neigh++;
    }
if(a[i][j]==1 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
        AA2_neigh++;

    } if(a[i][j]==0 && BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
        BB2_neigh++;

    } if (a[i][j]==1&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==0){
        AB2_neigh++;

    } if(a[i][j]==0&&BoundaryConditions(i+x[p][0],j+x[p][1],a,n,m)==1){
        BA2_neigh++;
    }
}

if(a[i][j]==1 && BoundaryConditions(i+y[p][0],j+y[p][1],a,n,m)==1){
        AA3_neigh++;

    } if(a[i][j]==0 && BoundaryConditions(i+y[p][0],j+y[p][1],a,n,m)==0){
        BB3_neigh++;

    } if (a[i][j]==1&&BoundaryConditions(i+y[p][0],j+y[p][1],a,n,m)==0){
        AB3_neigh++;

    } if(a[i][j]==0&&BoundaryConditions(i+y[p][0],j+y[p][1],a,n,m)==1){
        BA3_neigh++;
    }
}

for(k=0;k<12;k++){

if(a[i][j]==1 && BoundaryConditions(i+z[k][0],j+z[k][1],a,n,m)==1){
        AA4_neigh++;

    } if(a[i][j]==0 && BoundaryConditions(i+z[k][0],j+z[k][1],a,n,m)==0){
        BB4_neigh++;

    } if (a[i][j]==1&&BoundaryConditions(i+z[k][0],j+z[k][1],a,n,m)==0){
        AB4_neigh++;

    } if(a[i][j]==0&&BoundaryConditions(i+z[k][0],j+z[k][1],a,n,m)==1){
        BA4_neigh++;
    }
}
}
}
}

```

```

    }
    void CalcProb(double &aa1, double &bb1, double & ab1, double & ba1, double &aa2, double &bb2,
    double &ab2, double &ba2, double &aa3, double &bb3, double &ab3, double &ba3, double &aa4,
    double &bb4, double & ab4, double & ba4)
    {
        fstream res;
        //output file
        res.open("c:\\temp\\CrystalReport.txt", ios::out|ios::app );

        //assuming A atom total number of first neighbors
        double Tot_A1_neigh=aa1+ba1;
        //assuming B atom total number of first neighbors
        double Tot_B1_neigh=bb1+ab1;
        double Tot_A2_neigh=aa2+ba2;
        double Tot_B2_neigh=bb2+ba2;
        double Tot_A3_neigh=aa3+ba3;
        double Tot_B3_neigh=bb3+ab3;
        double Tot_A4_neigh=aa4+ba4;
        double Tot_B4_neigh=bb4+ab4;

        //Calculates probabilities for first neighbors
        double prob_A1_A=(aa1/Tot_A1_neigh);
        double prob_A1_B=(ab1/Tot_B1_neigh);
        double prob_B1_A=(ba1/Tot_A1_neigh);
        double prob_B1_B=(bb1/Tot_B1_neigh);

        //calculates probabilities for second neighbors
        double prob_A2_A=(aa2/Tot_A2_neigh);
        double prob_A2_B=(ab2/Tot_B2_neigh);
        double prob_B2_A=(ba2/Tot_A2_neigh);
        double prob_B2_B=(bb2/Tot_B2_neigh);

        //calculates probabilities for third neighbors
        double prob_A3_A=(aa3/Tot_A3_neigh);
        double prob_A3_B=(ab3/Tot_B3_neigh);
        double prob_B3_A=(ba3/Tot_A3_neigh);
        double prob_B3_B=(bb3/Tot_B3_neigh);

        //calculates probabilities for fourth neighbors
        double prob_A4_A=(aa4/Tot_A4_neigh);
        double prob_A4_B=(ab4/Tot_B4_neigh);
        double prob_B4_A=(ba4/Tot_A4_neigh);
        double prob_B4_B=(bb4/Tot_B4_neigh);

        res<<"The probability of an A type atom having an A type first neighbor is: "
        <<prob_A1_A<<endl;
        res<<"The probability of an A type atom having a B type first neighbor is: "
        <<prob_B1_A<<endl;
        res<<"The probability of a B type atom having an A type first neighbor is: "
        <<prob_A1_B<<endl;
        res<<"The probability of a B type atom having a B type first neighbor is: "
        <<prob_B1_B<<endl;
        res<<endl;
        res<<"The probability of an A type atom having an A type second neighbor is: "
        <<prob_A2_A<<endl;
        res<<"The probability of an A type atom having a B type second neighbor is: "
        <<prob_B2_A<<endl;
        res<<"The probability of a B type atom having an A type second neighbor is: "
        <<prob_A2_B<<endl;
    }

```

```

    res<<"The probability of a B type atom having a B type second neighbor is: "
    <<prob_B2_B<<endl;
    res<<endl;
    res<<"The probability of an A type atom having an A type third neighbor is: "
    <<prob_A3_A<<endl;
    res<<"The probability of an A type atom having a B type third neighbor is: "
    <<prob_B3_A<<endl;
    res<<"The probability of a B type atom having an A type third neighbor is: "
    <<prob_A3_B<<endl;
    res<<"The probability of a B type atom having a B type third neighbor is: "
    <<prob_B3_B<<endl;
    res<<endl;
    res<<"The probability of an A type atom having an A type fourth neighbor is: "
    <<prob_A4_A<<endl;
    res<<"The probability of an A type atom having a B type fourth neighbor is: "
    <<prob_B4_A<<endl;
    res<<"The probability of a B type atom having an A type fourth neighbor is: "
    <<prob_A4_B<<endl;
    res<<"The probability of a B type atom having a B type fourth neighbor is: "
    <<prob_B4_B<<endl;
    res.close();
}
complex<double> Structure_Factor( Mat &MC, double a1, double b1, int x, int y, double x_0,
double y_0, double s, int h, int k )
{
//define i as square root of -1
    complex<double>complex_i = sqrt(complex<double>(-1));
    complex<double> f;
    complex<double> e;
    complex<double> SF;
    CD row(y);
    for(int i=0; i<x; i++){
        for (int j=0;j<y;j++){
            e=2*M_PI*((x_0+h*s)*i+(y_0+k*s)*j);
            if(MC[i][j]==1)
            {
                f=a1;
            }
            else if (MC[i][j]==0)
            {
                f=b1;
            }
            SF+=(f*exp(e*complex_i));
        }
    }
    return(SF);
}
complex<double> XRay_Structure_Factor( Mat &MC, double a, double b, int x, int y, double x_0,
double y_0, double s, int h, int k, scatFac &atom1, scatFac &atom2, double Ang)
{
    complex<double>complex_i = sqrt(complex<double>(-1)); //define i as square root of -1
    complex<double> f;
    complex<double> e;
    complex<double> SF;
    CD row(y);

    for(int i=0; i<x; i++){
        for (int j=0;j<y;j++){
            e=2*M_PI*((x_0+h*s)*i+(y_0+k*s)*j);

```

```

        if(MC[i][j]==1)
            {
                f=atom1.calcSF(Ang);
            }
        else if (MC[i][j]==0)
            {
                f= atom2.calcSF(Ang);
            }
        SF+=(f*exp(e*complex_i));
    }
}

return(SF);

}

double Calc_rezAng(double x_0, double y_0, double s, int h, int k){
    double A_Cell=6.18639;
    double B_Cell=6.18639;
    double C_Cell=3.83388;
    double V=A_Cell*B_Cell*C_Cell*cos(30.0);
    double p=x_0+h*s;
    double q=y_0*k*s;
    double u=p*(B_Cell*C_Cell*sin(90.0))/V*cos(30.0)+q*(A_Cell*B_Cell*sin(90.0))/V*cos(90.0);
    double v=p*(B_Cell*C_Cell*sin(90.0))/V*cos(60.0)+q*(A_Cell*B_Cell*sin(90.0))/V*cos(0.0);
    double rezAng=sqrt(pow(u,2)+pow(v,2))/2;
    return(rezAng);
}

void Lot(Mat &MC, Mat &Lots, unsigned int size, int x, int y, unsigned int &i_index,
unsigned int &j_index){

    i_index=rand()%x;
    j_index=rand()%y;

    unsigned int i, j;
    Vec row(size);

        for(i=0;i<size;i++){
            for(j=0;j<size;j++){
                Lots[i][j]=BoundaryConditions(i+i_index,j+j_index,MC, x, y);
            }
        }
}

/*****
* Function Name: Main()
* Input:
* output: 0
* Description: Calls RandomInput(), Reservoir(), DisplayCrystal() and MonteCarlo().
Writes optimized crystal after specified number of MC steps to file. prompts user
for input values of crystal size, reservoir size, interaction energies and number of
MC steps.
*****/

```

```

int main(){
    srand ( time(NULL) ); //seeds random number generator to change every time program is run
    fstream res;
    Mat a,CM;
    Vec r;
    int m,n;
    int A=0;//for counting number of A type atoms and B type atoms
    int B=0;
    int Ares=0;
    int Bres=0;
    unsigned int MC,i,j,k;
    unsigned int s; //user defined number of Monte Carlo cycles to calculate
    double AA1_neigh=0;
    double AA2_neigh=0;
    double AA3_neigh=0;
    double AA4_neigh=0;
    double BB1_neigh=0;
    double BB2_neigh=0;
    double BB3_neigh=0;
    double BB4_neigh=0;
    double AB1_neigh=0;
    double AB2_neigh=0;
    double AB3_neigh=0;
    double AB4_neigh=0;
    double BA1_neigh=0;
    double BA2_neigh=0;
    double BA3_neigh=0;
    double BA4_neigh=0;

    cout << "Enter the x dimension of your crystal:" << endl;
    cin >> m;
    cout<<"Enter the y dimension of your crystal:"<<endl;
    cin>>n;
    cout<<"How many elements would you like in your reservoir"<<endl;
    cin>>s;
    cout<<"How many Monte Carlo steps would you like to run?"<<endl;
    cin>>MC;
    double AA1,AA2,AB1,AB2,BB1,BB2;

    cout<<"please enter interaction energies for for first neighbor
interactions: AA, BB, AB respectively"<<endl;
    cin>>AA1;
    cin>>BB1;
    cin>>AB1;

    cout<<"please enter interaction energies for the second neighbor
interactions: AA, BB, AB respectively"<<endl;
    cin>>AA2;
    cin>>BB2;
    cin>>AB2;

    CM=RandomInput( a, n, m );
    Count(CM, n, m, A, B);
    r=Reservoir(s);
    ReservoirCount(r,Ares,Bres);
    V result(MC);

    res.open("c:\\temp\\CrystalReport.txt", ios::out ); //output file

```



```

        if( ! res.good() )
        {
            cout << "ERROR: could not open output file" << endl;
            return( -1 );
        }
res<<"Optimized Crystal for "<<M<<" steps and first neighbor
interaction    energies of: "<<AA<<" "<<BB<<" "<<AB<<endl;
res<<endl;
res<<"second neighbor interaction energies of: "
<<AA2<<" "<<BB2<<" "<<AB2<<endl;
res<<endl;
res<<"crystal dimensions are: "<< m <<" by "<< n <<endl;
res<<endl;
res<<"Reservoir size is: "<<s<<endl;
res<<endl;
res<<"The Starting Crystal is: "<<endl;
res<<endl;
        for( i = 0; i < CM.size(); i++ )
        {
            for( j = 0; j < CM[i].size(); j++ )
            {
                if(CM[i][j]==0){
                    res<< "+" << " ";
                }else{
                    res<< "0" <<" ";
                }
            }
            res<<endl;
        }
//shifts view to the type of hexagonal crystal the program makes a,b angle is 60 degrees
        for(k=0;k<i+1;k++){
            res<<" ";
        }
res<<endl;
double E=CalculateCrystalEnergy(a,n,m,AA1,BB1,AB1,AA2,BB2,AB2);
res<<"Initial crystal energy is: "<<E<<endl;
res<<endl;
res<<"Number of A type atoms in unoptimized crystal is: "
<<A<<
res<<endl;
res<<"The number of A atoms in starting reservoir is: "
<< A
res<<endl;
res<<"Number of B type atoms in unoptimized crystal is: "
<<B
res<<"The number of B atoms in starting reservoir is: "
<< B
res<<endl;
res<<endl;
res<<" The pair correlation statistics of the starting crystal are: "
<<endl;
res<<endl;
res.close();
CountNeighbors(CM, n, m, AA1_neigh, BB1_neigh, AB1_neigh, BA1_neigh, AA2_neigh, BB2_neigh,
AB2_neigh, BA2_neigh, AA3_neigh, BB3_neigh, AB3_neigh, BA3_neigh, AA4_neigh, BB4_neigh,
AB4_neigh, BA4_neigh );

```

```

CalcProb(AA1_neigh, BB1_neigh, AB1_neigh,BA1_neigh,AA2_neigh, BB2_neigh, AB2_neigh,
BA2_neigh, AA3_neigh, BB3_neigh, AB3_neigh, BA3_neigh, AA4_neigh, BB4_neigh, AB4_neigh,
BA4_neigh );
res.open("c:\\temp\\CrystalReport.txt", ios::out|ios::app ); //output file
    if( ! res.good() )
    {
        cout << "ERROR: could not open output file" << endl;
        return( -1 );
    }

for(i=0;i<MC;i++){

    MonteCarlo(CM,r,n,m,s,E,AA1,BB1,AB1,AA2,BB2,AB2);
        cout<<"Crystal Energies are:"<< E <<endl;
    }

int X=0;
int Y=0;
int Xres=0;
int Yres=0;
Count(CM, n, m, X, Y);
ReservoirCount(r,Xres,Yres);
res<<endl;
res<<"The optimized Crystal is: " <<endl;
res<<endl;
    for( i = 0; i < CM.size(); i++ )
    {
        for( j = 0; j < CM[i].size(); j++ )
        {
            if(CM[i][j]==0){
                res<< "+" << " ";
            }else{
                res<< "0" << " ";
            }
        }
    }
res<<endl;

//shifts view to the type of hexagonal crystal the program makes a,b angle is 60 degrees
for(k=0;k<i+1;k++){
    res<<" ";
}

}

res<<endl;
res<<"Final crystal energy is: " <<E<<endl;
res<<endl;
res<<"Number of A type atoms in optimized crystal is: "
<<X<<
res<<"The number of A atoms in the reservoir after Monte Carlo optimization is: "
<< Xres<<endl;
res<<"Number of B type atoms in optimized crystal is: "
<<Y<<
res<<"The number of B atoms in the reservoir after Monte Carlo
optimization is: " << Yres<<endl;
res<<endl;

res.close();
fstream f;
//output file

```

```

f.open("c:\\temp\\CrystalContents.txt", ios::out );

if( ! f.good() )
{
    cout << "ERROR: could not open output file" << endl;
    return( -1 );
}
f<<"Optimized Crystal for "<<MG<<" steps and first neighbor interaction
energies of: "<<AAk<<" "<<BBk<<" "<<ABk<<endl;
f<<"second neighbor interaction energies of: "
<<AA2<<" "<<BB2<<" "<<AB2<<endl;
f<<"crystal dimensions are: "<<m<<" by "<<n<<endl;
f<<"The Optimized Crystal is: "<<endl;
for( i = 0; i < CM.size(); i++)
{
    for( j = 0; j < CM[i].size(); j++)
    {
        if(CM[i][j]==0){
            f<<"+" <<" ";
        }else{
            f<<"0" <<" ";
        }
    }
    f<<endl;
}

//shifts view to the type of hexagonal crystal the program makes a,b angle is 60 degrees
    for(k=0;k<i+1;k++){
        f<<" ";
    }
}
f<<endl;
f.close();
ifstream in("c://temp//CrystalContents.txt");
ReadOptimizedCrystalFile(in ,CM);

int y=0;
int x=CM.size();
if(CM.empty())
    y=0;
else
    y=CM[0].size();

double aa1_neigh=0;
double aa2_neigh=0;
double aa3_neigh=0;
double aa4_neigh=0;
double bb1_neigh=0;
double bb2_neigh=0;
double bb3_neigh=0;
double bb4_neigh=0;
double ab1_neigh=0;
double ab2_neigh=0;
double ab3_neigh=0;
double ab4_neigh=0;
double ba1_neigh=0;
double ba2_neigh=0;
double ba3_neigh=0;
double ba4_neigh=0;

//output file

```

```

res.open("c:\\temp\\CrystalReport.txt", ios::out|ios::app );
if( ! res.good() )
{
    cout << "ERROR: could not open output file" << endl;
    return( -1 );
}
res<<"The optimized Crystal pair correlations Statistics are: "<<endl;
res.close();

    CountNeighbors(CM, y, x, aa1_neigh, bb1_neigh, ab1_neigh, ba1_neigh,
aa2_neigh, bb2_neigh, ab2_neigh, ba2_neigh, aa3_neigh, bb3_neigh,
ab3_neigh, ba3_neigh, aa4_neigh, bb4_neigh, ab4_neigh, ba4_neigh );

    CalcProb(aa1_neigh, bb1_neigh, ab1_neigh, ba1_neigh, aa2_neigh, bb2_neigh,
ab2_neigh, ba2_neigh, aa3_neigh, bb3_neigh, ab3_neigh, ba3_neigh, aa4_neigh,
bb4_neigh, ab4_neigh, ba4_neigh );

    res<<endl;
res.close();

//output file
res.open("c:\\temp\\CrystalReport.txt", ios::out|ios::app );
if( ! res.good() )
{
    cout << "ERROR: could not open output file" << endl;
    return( -1 );
}

Mat Lots;
complex<double> F, I;
complex<double> LF, LI;

double a1,b1; //scattering lengths of A and B atoms
double st, x_0, y_0;
unsigned int g;
unsigned int h;
unsigned int i_index, j_index;

    cout<<endl;
cout<<"Please enter the scattering length of A atom.
"<<endl;
cin>>a1;
cout<<"Please enter the scattering length of B atom.
"<<endl;
cin>>b1;
cout<<"please enter size of the 2-D square grid you would like to calculate the
intensity for the optimized crystal."<<endl;
cin>>g;
cout<<"please enter the 2-D origin (starting point) of the intensity calculation
(i.e x coordinate then y coordinate)."<<endl;
cin>>x_0;
cin>>y_0;
cout<<"Please enter the step size on the 2-D grid which you would like to
calculate the intensities."<<endl;
cin>>st;
res<<"The scattering length of A is "<<a1<<" and the scattering length of B is "
<<b1<<". "<<endl;
res<<"The neutron intensities of your crystal calculated without lots on a square
grid of size " <<g<< " with step size "<<st<< "

```

```

and origin "<<"("<<x_0<<","<<y_0<<") "<<"are: "<<endl;
res<<endl;

    for (h=0;h<g;h++){
        for (k=0;k<g;k++){
            F=Structure_Factor(CM,a1,b1,x,y,x_0,y_0,st,h,k);
            res<<norm(F)<<endl;
        }
    }

    unsigned int size, num_lots;
    cout<<<"Please enter the lot size which you would like to use. "<<endl;
    cin>>size;
    cout<<<"Please enter the number of lots which you would like to use."<<endl;
    cin>>num_lots;

    Lots.resize(size,vector<int>(size));

    res<<endl;
    res<<endl;
    res<<<"The neutron intensities of your crystal calculated with "
    <<num_lots<< " lots of size " <<size<< " are:"<<endl;
res<<endl;

    for (i=0;i<num_lots;i++){

        Lot(CM, Lots, size, x, y, i_index, j_index);

    }
    for (h=0;h<g;h++){
        for (k=0;k<g;k++){
            LF=Structure_Factor(Lots,a1,b1,size,size,x_0,y_0,st,h,k);
            res<<norm(LF)<<endl;
        }
    }

\\For x-ray intensities:

for(m=0;m<num_lots;m++){

    Lot(MC, Lots, size, x, y, i_index, j_index);
    DisplayCrystal(Lots);
}

for (h=0;h<n;h++){
    for (k=0;k<n;k++){
        LF=Structure_Factor(Lots,a,b,size,size,x_0,y_0,s,h,k);
        cout<<norm(LF)<<endl;
    }
}

\\uses scattering factors from file with a table of atoms and their x-ray scattering factors
\\in this example atom A is Na and atom B is La
scatFac atom1("Na");
scatFac atom2("La");

double Ang;
    for (h=0;h<n;h++){
        for (k=0;k<n;k++){
            Ang=Calc_rezAng(x_0,y_0,s,h,k);

```

```

        LF=XRay_Structure_Factor(Lots,a,b,size ,size ,x_0,y_0,s,h,k,atom1, atom2, Ang);
        cout<<"X-Ray intensities are: "<<norm(LF)<<endl;

    }

}

res.close();
cout<<"finished"<<endl;
return(0);

}

```

A. 2 Motif percentage calculation

A C++ code was written (below) to calculate the statistics of each motif type as a percentage of the 96,000 layer crystal. The output of the optimization for the best individual and its clones is parsed and the mean and standard deviation is calculated for each layer motif. This code was used to calculate the results described in chapter 7 section 7.10 and to calculate the structural changes described in the sensitivity analysis of section 7.2.4. The user defines the number of crystals (external variable) as NUMCRYST=individuals*clones.

```

/*****
 * MotiffAve.cpp
 *
 * Created on: Jun 6, 2011
 * Author: Tara Michels-Clark
 *****/

/*****
 * Purpose of the Program: Takes input files of type .txt and parses them to read into a vector.
 * The average of the motif count and motif stats for each generation is calculated for each motif
 * type and output to a results file with the generation number included in file name.
 *
 *****/

//header file with external files for user input of gen # and crystal number = individuals * clones
#include "MotiffAve.h"
#include <iostream>
#include <sstream> //file strings for reading in multiple files with indexed
#include <fstream> //for I/O
#include <string>
#include <vector> //Vector class (probably don't need in this case)
#include <cstdlib> //to get Vector from c++ standard Library
#include <cmath> //for the distance formula

```

```

using namespace std;

/*****
 * CLASS: Particle
 * Data elements: double x, double y, double occupancy, string particle type
 * METHODS: get_particle_type(), get_x(), get_y(), get_occupancy(), Display(), translate()
 * FRIENDS: istream() for input files used with class member functions
 *
 *****/
class Motif
{
private:
    double stats;
    int motifcount;
    string Motif_type;

public:
    string get_Motif_type();
    int get_motifcount();
    double get_stats();
    void Display();
    Motif();
    ~Motif();

    friend istream& operator>>( istream&, Motif& );

};

//for reading in the input file called in.txt.
istream& operator>>( istream& in, Motif& m )
{
    if( in.good() )
    {
        string temp;
        in >> temp;

        if( temp == "statistics:" )
        {
            in >> temp;
            in >> temp;
            // one more read to get the proper value
            in >> temp;
        }

        // Read the first string, if the value is one that expects a
        // second string, read it as well
        if( temp == "disord" )
        {
            in>> m.Motif_type;
            m.Motif_type = temp + " " + m.Motif_type;
        }
        else
        {
            m.Motif_type = temp;
        }

        in>>m.stats;
        in>>m.motifcount;
    }
}

```

```

    }

    return in;
}

/*****
 * Method Name: get_Motif_type()
 * Input: none
 * output: string
 * Description: Returns the String of motif type MDO1, MDO2, MDO3, disord e1, disord e2
 *****/
string Motif::get_Motif_type(){

    return(Motif_type);
}

/*****
 * Method Name: get_motifcount()
 * Input: none
 * output: int
 * Description: Returns the motif count of each motif type.
 *****/
int Motif::get_motifcount(){
    return(motifcount);
}

/*****
 * Method Name: get_stats()
 * Input: none
 * output: double
 * Description: Returns the % of crystals with those motif types
 *****/
double Motif::get_stats()
{
    return(stats);
}

/*****
 * Method Name: Display()
 * Input: none
 * output: void
 * Description: Displays the motif type, motif count and motif stats for the user.
                * Mainly for testing to make sure files are read in correctly)
 *****/
void Motif::Display(){

    cout<<"Motif type "<<Motif_type<<endl;
    cout<<"Motif count = "<<motifcount<<endl;
    cout<<"Motif Statistics = "<<stats<<endl;

}

//Constructor for the Motif class.
Motif::Motif(){

}

//Destructor for the Motif class.
Motif::~Motif(){

```



```

}
typedef vector<Motif> Vec;
typedef vector<double> V;
typedef vector<int> v;
typedef vector<string> sv;

/*****
 * Function Name: Motif_Stats
 * Input: vector
 * output: vector
 * Description: takes an input vector of the motif statistics and loops through
                * and sorts out each motif type and averages them and outputs
                * their average into a vector.
 *****/
void Motif_Stats( V &MS, V &av, V &stdev, sv M)
{
    double MDO1_sum=0;
    double MDO2_sum=0;
    double MDO3_sum=0;
    double disord_e1_sum=0;
    double disord_e2_sum=0;
    double MDO1_ave,MDO2_ave,MDO3_ave,disord_e1_ave,disord_e2_ave;
    double MDO1_std,MDO2_std,MDO3_std,disord_e1_std,disord_e2_std;
    unsigned int j;
    vector<double>result;

    for(j=0;j<MS.size();j++){
        if(M[j]=="MDO1"){
            MDO1_sum+=MS[j];
        }else if (M[j]=="MDO2"){
            MDO2_sum+=MS[j];
        }else if (M[j]=="MDO3"){
            MDO3_sum+=MS[j];
        }else if (M[j]=="disord e1"){
            disord_e1_sum+=MS[j];
        }else if (M[j]=="disord e2"){
            disord_e2_sum+=MS[j];
        }
    }

    MDO1_ave=MDO1_sum/NUMCRYST;
    MDO2_ave=MDO2_sum/NUMCRYST;
    MDO3_ave=MDO3_sum/NUMCRYST;
    disord_e1_ave=disord_e1_sum/NUMCRYST;
    disord_e2_ave=disord_e2_sum/NUMCRYST;
    av.push_back(MDO1_ave);
    av.push_back(MDO2_ave);
    av.push_back(MDO3_ave);
    av.push_back(disord_e1_ave);
    av.push_back(disord_e2_ave);

    double MDO1_sqdiff=0;
    double MDO2_sqdiff=0;
    double MDO3_sqdiff=0;
    double disord_e1_sqdiff=0;
    double disord_e2_sqdiff=0;
    for(j=0;j<MS.size();j++){
        if(M[j]=="MDO1"){
            MDO1_sqdiff+=pow((MS[j]-MDO1_ave),2);

```

```

    }else if (M[j]=="MDO2"){
        MDO2_sqdiff=pow((MS[j]-MDO2_ave),2);
    }else if (M[j]=="MDO3"){
        MDO3_sqdiff=pow((MS[j]-MDO3_ave),2);
    }else if (M[j]=="disord_e1"){
        disord_e1_sqdiff=pow((MS[j]-disord_e1_ave),2);
    }else if (M[j]=="disord_e2"){
        disord_e2_sqdiff=pow((MS[j]-disord_e2_ave),2);
    }
}
MDO1_std=sqrt(MDO1_sqdiff/NUMCRYST);
MDO2_std=sqrt(MDO2_sqdiff/NUMCRYST);
MDO3_std=sqrt(MDO3_sqdiff/NUMCRYST);
disord_e1_std=sqrt(disord_e1_sqdiff/NUMCRYST);
disord_e2_std=sqrt(disord_e2_sqdiff/NUMCRYST);
stdev.push_back(MDO1_std);
stdev.push_back(MDO2_std);
stdev.push_back(MDO3_std);
stdev.push_back(disord_e1_std);
stdev.push_back(disord_e2_std);

}

/*****
* Function Name: main()
* Input: none
* output: none
* Description: opens and reads input files in loop. Calls Motif_Stats() Writes
*              to result file with user input generation number in the file name.
*****/
int main()
{

    ifstream f; //input files f
    Vec m; //motif vector

    stringstream ss,os; //file input and output streams
    string s,rs; //string name for files
    unsigned int x; //index for individual number in file name

    //Vectors of string, int, double and Motif vector type deff
    V Motif_StatResults;
    V MS;
    V stdev,ave;
    V MC;
    V RC;
    sv M;

    //Loops through input files
    for( x = 0; x < 2; x++ )
    {
        ss.clear();
        ss << "c://temp//growth_cal.out_"<< x << ".txt";
        ss >> s;

        f.open( s.c_str() );

        if( f.good() )

```

```

{
    while( !f.eof() )
    {
        Motif ml;
        f >> ml;
        m.push_back(ml);
        MS.push_back(ml.get_stats()); //stat vector
        M.push_back(ml.get_Motif_type()); //Motif type vector
    }

    f.close(); //closes the file after reading
}

else
{
    //error message to user if file cannot be opened
    cout << "ERROR: file could not be opened" << endl;
    return( -1 );
}

f.close();

}

Motif_Stats( MS, ave, stdev, M);

//writing output file
fstream fout;
os.clear(); //output stream for file with user input index
os << "c:/temp/growth_cal.res.out_" << gennum << ".txt";
os >> rs;
fout.open( rs.c_str(),ios::out );
/opening .txt file for output results
//test to see if output file opened
if( ! fout.good() )
{
    cout << "ERROR: could not open output file" << endl;
    return( -1 );
}
//writing to output file
for(unsigned int z=0; z<stdev.size();z++)
{
    fout << M[z]<<"    "<<ave[z]<<"    "<<stdev[z]<<endl;
    cout<<M[z]<<"    "<<ave[z]<<"    "<<stdev[z]<<endl;
}
//closes input and output files
f.close();
fout.close();
return(0);
}

```

A. 3 CIF files

A. 3.1 X-ray CIF

```
data_shelxl
_audit_creation_method      SHELXL-2013
_chemical_name_systematic
;
?
;
_chemical_name_common       ?
_chemical_melting_point     ?
_chemical_formula_moiety    ?
_chemical_formula_sum       'F6 La1.50 Na1.50'
_chemical_formula_weight    356.85

loop_
_atom_type_symbol
_atom_type_description
_atom_type_scatter_dispersion_real
_atom_type_scatter_dispersion_imag
_atom_type_scatter_source
'F' 'F' 0.0171 0.0103
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'
'Na' 'Na' 0.0362 0.0249
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'
'La' 'La' -0.2871 2.4523
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'

_space_group_crystal_system hexagonal
_space_group_IT_number      174
_space_group_name_H-M_alt   'P -6'
_space_group_name_Hall      'P -6'

_shelx_space_group_comment
;
The symmetry employed for this shelxl refinement is uniquely defined
by the following loop, which should always be used as a source of
symmetry information in preference to the above space-group names.
They are only intended as comments.
;

loop_
_space_group_symop_operation_xyz
'x, y, z'
'-y, x-y, z'
'-x+y, -x, z'
'x, y, -z'
'-y, x-y, -z'
'-x+y, -x, -z'

_cell_length_a              6.1520(14)
_cell_length_b              6.1520(14)
_cell_length_c              3.8191(9)
_cell_angle_alpha           90
_cell_angle_beta            90
_cell_angle_gamma           120
_cell_volume                125.18(6)
_cell_formula_units_Z       1
_cell_measurement_temperature 100(2)
_cell_measurement_reflns_used ?
_cell_measurement_theta_min ?
_cell_measurement_theta_max ?

_exptl_crystal_description  plate
_exptl_crystal_colour       colorless
_exptl_crystal_density_meas ?
_exptl_crystal_density_method ?
_exptl_crystal_density_diffrn 4.734
_exptl_crystal_F_000        156
_exptl_transmission_factor_min ?
_exptl_transmission_factor_max ?
_exptl_crystal_size_max     0.120
_exptl_crystal_size_mid     0.100
_exptl_crystal_size_min     0.090
_exptl_absorpt_coefficient_mu 12.848
_shelx_estimated_absorpt_T_min 0.308
_shelx_estimated_absorpt_T_max 0.391
_exptl_absorpt_correction_type multi-scan
_exptl_absorpt_correction_T_min 0.5704
_exptl_absorpt_correction_T_max 0.7457
```

```

_exptl_absorpt_process_details ?
_exptl_special_details
;
?
;

_diffrn_ambient_temperature 100(2)
_diffrn_radiation_wavelength 0.71073
_diffrn_radiation_type MoK\alpha
_diffrn_source ?
_diffrn_measurement_device_type ?
_diffrn_measurement_method ?
_diffrn_detector_area_resol_mean ?
_diffrn_reflns_number 816
_diffrn_reflns_av_unetI/netI 0.0311
_diffrn_reflns_av_R_equivalents ?
_diffrn_reflns_limit_h_min -8
_diffrn_reflns_limit_h_max 8
_diffrn_reflns_limit_k_min -8
_diffrn_reflns_limit_k_max 8
_diffrn_reflns_limit_l_min -5
_diffrn_reflns_limit_l_max 5
_diffrn_reflns_theta_min 3.824
_diffrn_reflns_theta_max 28.183
_diffrn_reflns_theta_full 25.242
_diffrn_measured_fraction_theta_max 1.000
_diffrn_measured_fraction_theta_full 0.989
_diffrn_reflns_Laue_measured_fraction_max 1.000
_diffrn_reflns_Laue_measured_fraction_full 0.989
_diffrn_reflns_point_group_measured_fraction_max 1.000
_diffrn_reflns_point_group_measured_fraction_full 0.994
_reflns_number_total 816
_reflns_number_gt 816
_reflns_threshold_expression 'I > 2\sigma(I)'
_reflns_Friedel_coverage 0.959
_reflns_Friedel_fraction_max 1.000
_reflns_Friedel_fraction_full 1.000

_reflns_special_details
;
_reflns_Friedel_fraction is defined as the number of unique
Friedel pairs measured divided by the number that would be
possible theoretically, ignoring centric projections and
systematic absences.
;

_computing_data_collection ?
_computing_cell_refinement ?
_computing_data_reduction ?
_computing_structure_solution ?
_computing_structure_refinement 'SHELXL-2013 (Sheldrick, 2013)'
_computing_molecular_graphics ?
_computing_publication_material ?

_refine_special_details
;
Refined as a 2-component inversion twin.
;
_refine_ls_structure_factor_coef Fsqd
_refine_ls_matrix_type full
_refine_ls_weighting_scheme calc
_refine_ls_weighting_details
'w=1/[\sigma^2*(Fo^2)] where P=(Fo^2+2Fc^2)/3'
_atom_sites_solution_primary ?
_atom_sites_solution_secondary ?
_atom_sites_solution_hydrogens .
_refine_ls_hydrogen_treatment undef
_refine_ls_extinction_method SHELXL
_refine_ls_extinction_coef 0.33(3)
_refine_ls_extinction_expression
'Fc^*=kFc[1+0.001xFc^2-1^3/sin(2\theta)]^-1/4^'
_refine_ls_abs_structure_details
;
Refined as an inversion twin.
;
_refine_ls_abs_structure_Flack 0.43(6)
_chemical_absolute_configuration ?
_refine_ls_number_reflns 816
_refine_ls_number_parameters 22
_refine_ls_number_restraints 0
_refine_ls_R_factor_all 0.0138
_refine_ls_R_factor_gt 0.0138
_refine_ls_wR_factor_ref 0.0346
_refine_ls_wR_factor_gt 0.0346
_refine_ls_goodness_of_fit_ref 1.107
_refine_ls_restrained_S_all 1.107
_refine_ls_shift/su_max 0.000

```

```

_refine_ls_shift/su_mean      0.000

loop_
  _atom_site_label
  _atom_site_type_symbol
  _atom_site_fract_x
  _atom_site_fract_y
  _atom_site_fract_z
  _atom_site_U_iso_or_equiv
  _atom_site_adp_type
  _atom_site_occupancy
  _atom_site_site_symmetry_order
  _atom_site_calc_flag
  _atom_site_refinement_flags_posn
  _atom_site_refinement_flags_adp
  _atom_site_refinement_flags_occupancy
  _atom_site_disorder_assembly
  _atom_site_disorder_group
La2 La -0.3333 0.3333 -0.5000 0.0078(4) Uani 0.5001 6 d S T P . .
Na2 Na -0.3333 0.3333 -0.5000 0.0078(4) Uani 0.5001 6 d S T P . .
La1 La 0.0000 0.0000 0.0000 0.0052(3) Uani 1 6 d S T P . .
Na1 Na -0.6667 0.6667 -0.687(3) 0.016(2) Uani 0.5 3 d S T P . .
F1 F -0.0397(11) 0.3706(9) 0.0000 0.0171(12) Uani 1 2 d S T P . .
F2 F -0.7546(8) 0.2693(8) -0.5000 0.0091(8) Uani 1 2 d S T P . .

loop_
  _atom_site_aniso_label
  _atom_site_aniso_U_11
  _atom_site_aniso_U_22
  _atom_site_aniso_U_33
  _atom_site_aniso_U_23
  _atom_site_aniso_U_13
  _atom_site_aniso_U_12
La2 0.0046(5) 0.0046(5) 0.0142(9) 0.000 0.000 0.0023(3)
Na2 0.0046(5) 0.0046(5) 0.0142(9) 0.000 0.000 0.0023(3)
La1 0.0065(3) 0.0065(3) 0.0027(5) 0.000 0.000 0.00326(17)
Na1 0.014(2) 0.014(2) 0.021(6) 0.000 0.000 0.0070(12)
F1 0.008(2) 0.007(2) 0.037(3) 0.000 0.000 0.0045(16)
F2 0.009(2) 0.012(2) 0.0046(19) 0.000 0.000 0.003(2)

_geom_special_details
;
All esds (except the esd in the dihedral angle between two l.s. planes)
are estimated using the full covariance matrix. The cell esds are taken
into account individually in the estimation of esds in distances, angles
and torsion angles; correlations between esds in cell parameters are only
used when they are defined by crystal symmetry. An approximate (isotropic)
treatment of cell esds is used for estimating esds involving l.s. planes.
;

loop_
  _geom_bond_atom_site_label_1
  _geom_bond_atom_site_label_2
  _geom_bond_distance
  _geom_bond_site_symmetry_2
  _geom_bond_publ_flag
La2 F2 2.419(5) 2_565 ?
La2 F2 2.419(5) . ?
La2 F2 2.419(5) 3_455 ?
La2 F1 2.559(4) 3_454 ?
La2 F1 2.559(4) 1_554 ?
La2 F1 2.559(4) . ?
La2 F1 2.559(4) 3_455 ?
La2 F1 2.559(4) 2_565 ?
La2 F1 2.559(4) 2_564 ?
La2 Na1 3.623(2) 4_554 ?
La2 Na1 3.623(2) . ?
La2 Na1 3.623(2) 4_654 ?
La1 F1 2.411(5) 2 ?
La1 F1 2.411(5) 3 ?
La1 F1 2.411(5) . ?
La1 F2 2.484(3) 3_445 ?
La1 F2 2.484(3) 3_446 ?
La1 F2 2.484(3) 1_656 ?
La1 F2 2.484(3) 1_655 ?
La1 F2 2.484(3) 2_565 ?
La1 F2 2.484(3) 2_566 ?
La1 Na1 3.748(4) 4_544 ?
La1 Na1 3.748(4) 1_656 ?
La1 Na1 3.748(4) 1_546 ?
Na1 Na1 1.43(2) 4_554 ?
Na1 F2 2.336(6) 2_575 ?
Na1 F2 2.336(6) . ?
Na1 F2 2.336(6) 3_355 ?
Na1 Na1 2.39(2) 4_553 ?
Na1 F1 2.416(8) 1_454 ?
Na1 F1 2.416(8) 3_464 ?
Na1 F1 2.416(8) 2_564 ?

```

```

Na1 Na2 3.623(2) 1_565 ?
Na1 La2 3.623(2) 1_565 ?
F1 Na1 2.416(8) 1_656 ?
F1 Na1 2.416(8) 4_654 ?
F1 La2 2.559(4) 1_556 ?
F1 Na2 2.559(4) 1_556 ?
F2 Na1 2.336(6) 4_554 ?
F2 La1 2.484(3) 1_454 ?
F2 La1 2.484(3) 1_455 ?

loop_
  _geom_angle_atom_site_label_1
  _geom_angle_atom_site_label_2
  _geom_angle_atom_site_label_3
  _geom_angle
  _geom_angle_site_symmetry_1
  _geom_angle_site_symmetry_3
  _geom_angle_publ_flag
F2 La2 F2 120.0 2_565 . ?
F2 La2 F2 120.0 2_565 3_455 ?
F2 La2 F2 120.0 . 3_455 ?
F2 La2 F1 71.43(12) 2_565 3_454 ?
F2 La2 F1 69.69(11) . 3_454 ?
F2 La2 F1 131.72(10) 3_455 3_454 ?
F2 La2 F1 69.69(11) 2_565 1_554 ?
F2 La2 F1 131.72(10) . 1_554 ?
F2 La2 F1 71.43(12) 3_455 1_554 ?
F1 La2 F1 70.42(16) 3_454 1_554 ?
F2 La2 F1 69.69(11) 2_565 . ?
F2 La2 F1 131.72(10) . . ?
F2 La2 F1 71.43(12) 3_455 . ?
F1 La2 F1 141.11(8) 3_454 . ?
F1 La2 F1 96.5(2) 1_554 . ?
F2 La2 F1 71.43(12) 2_565 3_455 ?
F2 La2 F1 69.69(11) . 3_455 ?
F2 La2 F1 131.72(10) 3_455 3_455 ?
F1 La2 F1 96.5(2) 3_454 3_455 ?
F1 La2 F1 141.11(8) 1_554 3_455 ?
F1 La2 F1 70.42(16) . 3_455 ?
F2 La2 F1 131.72(10) 2_565 2_565 ?
F2 La2 F1 71.43(12) . 2_565 ?
F2 La2 F1 69.69(11) 3_455 2_565 ?
F1 La2 F1 141.11(8) 3_454 2_565 ?
F1 La2 F1 141.11(8) 1_554 2_565 ?
F1 La2 F1 70.42(16) . 2_565 ?
F1 La2 F1 70.42(16) 3_455 2_565 ?
F2 La2 F1 131.72(10) 2_565 2_564 ?
F2 La2 F1 71.43(12) . 2_564 ?
F2 La2 F1 69.69(11) 3_455 2_564 ?
F1 La2 F1 70.42(16) 3_454 2_564 ?
F1 La2 F1 70.42(16) 1_554 2_564 ?
F1 La2 F1 141.11(8) . 2_564 ?
F1 La2 F1 141.11(8) 3_455 2_564 ?
F1 La2 F1 96.5(2) 2_565 2_564 ?
F2 La2 Na1 155.47(11) 2_565 4_554 ?
F2 La2 Na1 39.52(11) . 4_554 ?
F2 La2 Na1 82.04(10) 3_455 4_554 ?
F1 La2 Na1 102.88(18) 3_454 4_554 ?
F1 La2 Na1 132.09(17) 1_554 4_554 ?
F1 La2 Na1 112.1(2) . 4_554 ?
F1 La2 Na1 85.94(17) 3_455 4_554 ?
F1 La2 Na1 41.7(2) 2_565 4_554 ?
F1 La2 Na1 63.1(2) 2_564 4_554 ?
F2 La2 Na1 155.47(11) 2_565 . ?
F2 La2 Na1 39.52(11) . . ?
F2 La2 Na1 82.04(10) 3_455 . ?
F1 La2 Na1 85.94(17) 3_454 . ?
F1 La2 Na1 112.1(2) 1_554 . ?
F1 La2 Na1 132.09(17) . . ?
F1 La2 Na1 102.88(18) 3_455 . ?
F1 La2 Na1 63.1(2) 2_565 . ?
F1 La2 Na1 41.7(2) 2_564 . ?
Na1 La2 Na1 22.7(4) 4_554 . ?
F2 La2 Na1 82.05(10) 2_565 4_654 ?
F2 La2 Na1 155.47(11) . 4_654 ?
F2 La2 Na1 39.52(11) 3_455 4_654 ?
F1 La2 Na1 132.09(17) 3_454 4_654 ?
F1 La2 Na1 63.1(2) 1_554 4_654 ?
F1 La2 Na1 41.7(2) . 4_654 ?
F1 La2 Na1 112.1(2) 3_455 4_654 ?
F1 La2 Na1 85.94(17) 2_565 4_654 ?
F1 La2 Na1 102.88(18) 2_564 4_654 ?
Na1 La2 Na1 116.22(12) 4_554 4_654 ?
Na1 La2 Na1 121.29(4) . 4_654 ?
F1 La1 F1 120.0 2 3 ?
F1 La1 F1 120.0 2 . ?
F1 La1 F1 120.0 3 . ?
F1 La1 F2 71.09(10) 2 3_445 ?

```

F1 La1 F2 71.61(11) 3 3_445 ?
 F1 La1 F2 129.75(7) . 3_445 ?
 F1 La1 F2 71.09(10) 2 3_446 ?
 F1 La1 F2 71.61(11) 3 3_446 ?
 F1 La1 F2 129.75(7) . 3_446 ?
 F2 La1 F2 100.50(15) 3_445 3_446 ?
 F1 La1 F2 129.75(7) 2 1_656 ?
 F1 La1 F2 71.09(10) 3 1_656 ?
 F1 La1 F2 71.61(11) . 1_656 ?
 F2 La1 F2 142.71(6) 3_445 1_656 ?
 F2 La1 F2 67.26(12) 3_446 1_656 ?
 F1 La1 F2 129.75(7) 2 1_655 ?
 F1 La1 F2 71.09(10) 3 1_655 ?
 F1 La1 F2 71.61(11) . 1_655 ?
 F2 La1 F2 67.26(12) 3_445 1_655 ?
 F2 La1 F2 142.71(6) 3_446 1_655 ?
 F2 La1 F2 100.50(15) 1_656 1_655 ?
 F1 La1 F2 71.61(11) 2 2_565 ?
 F1 La1 F2 129.75(7) 3 2_565 ?
 F1 La1 F2 71.09(10) . 2_565 ?
 F2 La1 F2 67.26(12) 3_445 2_565 ?
 F2 La1 F2 142.71(6) 3_446 2_565 ?
 F2 La1 F2 142.71(6) 1_656 2_565 ?
 F2 La1 F2 67.26(12) 1_655 2_565 ?
 F1 La1 F2 71.61(11) 2 2_566 ?
 F1 La1 F2 129.75(7) 3 2_566 ?
 F1 La1 F2 71.09(10) . 2_566 ?
 F2 La1 F2 142.71(6) 3_445 2_566 ?
 F2 La1 F2 67.26(12) 3_446 2_566 ?
 F2 La1 F2 67.26(12) 1_656 2_566 ?
 F2 La1 F2 142.71(6) 1_655 2_566 ?
 F2 La1 F2 100.49(15) 2_565 2_566 ?
 F1 La1 Na1 39.10(14) 2 4_544 ?
 F1 La1 Na1 149.21(14) 3 4_544 ?
 F1 La1 Na1 85.24(13) . 4_544 ?
 F2 La1 Na1 78.64(16) 3_445 4_544 ?
 F2 La1 Na1 107.09(17) 3_446 4_544 ?
 F2 La1 Na1 138.10(17) 1_656 4_544 ?
 F2 La1 Na1 104.68(17) 1_655 4_544 ?
 F2 La1 Na1 37.55(18) 2_565 4_544 ?
 F2 La1 Na1 72.43(17) 2_566 4_544 ?
 F1 La1 Na1 149.21(14) 2 1_656 ?
 F1 La1 Na1 85.23(13) 3 1_656 ?
 F1 La1 Na1 39.10(14) . 1_656 ?
 F2 La1 Na1 138.10(17) 3_445 1_656 ?
 F2 La1 Na1 104.68(17) 3_446 1_656 ?
 F2 La1 Na1 37.55(18) 1_656 1_656 ?
 F2 La1 Na1 72.43(17) 1_655 1_656 ?
 F2 La1 Na1 107.09(17) 2_565 1_656 ?
 F2 La1 Na1 78.64(16) 2_566 1_656 ?
 Na1 La1 Na1 123.43(6) 4_544 1_656 ?
 F1 La1 Na1 39.10(14) 2 1_546 ?
 F1 La1 Na1 149.21(14) 3 1_546 ?
 F1 La1 Na1 85.24(13) . 1_546 ?
 F2 La1 Na1 107.09(17) 3_445 1_546 ?
 F2 La1 Na1 78.64(16) 3_446 1_546 ?
 F2 La1 Na1 104.68(17) 1_656 1_546 ?
 F2 La1 Na1 138.10(17) 1_655 1_546 ?
 F2 La1 Na1 72.43(17) 2_565 1_546 ?
 F2 La1 Na1 37.55(18) 2_566 1_546 ?
 Na1 La1 Na1 37.2(3) 4_544 1_546 ?
 Na1 La1 Na1 110.31(16) 1_656 1_546 ?
 Na1 Na1 F2 72.2(3) 4_554 2_575 ?
 Na1 Na1 F2 72.2(3) 4_554 . ?
 F2 Na1 F2 111.1(3) 2_575 . ?
 Na1 Na1 F2 72.2(3) 4_554 3_355 ?
 F2 Na1 F2 111.1(3) 2_575 3_355 ?
 F2 Na1 F2 111.1(3) . 3_355 ?
 Na1 Na1 Na1 180.0 4_554 4_553 ?
 F2 Na1 Na1 107.8(3) 2_575 4_553 ?
 F2 Na1 Na1 107.8(3) . 4_553 ?
 F2 Na1 Na1 107.8(3) 3_355 4_553 ?
 Na1 Na1 F1 119.7(2) 4_554 1_454 ?
 F2 Na1 F1 168.1(5) 2_575 1_454 ?
 F2 Na1 F1 74.12(13) . 1_454 ?
 F2 Na1 F1 75.45(12) 3_355 1_454 ?
 Na1 Na1 F1 60.3(2) 4_553 1_454 ?
 Na1 Na1 F1 119.7(2) 4_554 3_464 ?
 F2 Na1 F1 75.45(12) 2_575 3_464 ?
 F2 Na1 F1 168.1(5) . 3_464 ?
 F2 Na1 F1 74.12(13) 3_355 3_464 ?
 Na1 Na1 F1 60.3(2) 4_553 3_464 ?
 F1 Na1 F1 97.6(3) 1_454 3_464 ?
 Na1 Na1 F1 119.7(2) 4_554 2_564 ?
 F2 Na1 F1 74.12(13) 2_575 2_564 ?
 F2 Na1 F1 75.45(12) . 2_564 ?
 F2 Na1 F1 168.1(5) 3_355 2_564 ?
 Na1 Na1 F1 60.3(2) 4_553 2_564 ?


```

F1 Na1 F1 97.6(3) 1_454 2_564 ?
F1 Na1 F1 97.6(3) 3_464 2_564 ?
Na1 Na1 La2 78.65(18) 4_554 . ?
F2 Na1 La2 75.13(14) 2_575 . ?
F2 Na1 La2 41.22(11) . . ?
F2 Na1 La2 146.0(4) 3_355 . ?
Na1 Na1 La2 101.35(18) 4_553 . ?
F1 Na1 La2 105.30(15) 1_454 . ?
F1 Na1 La2 137.54(19) 3_464 . ?
F1 Na1 La2 44.84(10) 2_564 . ?
Na1 Na1 Na2 78.65(18) 4_554 1_565 ?
F2 Na1 Na2 41.22(11) 2_575 1_565 ?
F2 Na1 Na2 146.0(4) . 1_565 ?
F2 Na1 Na2 75.13(14) 3_355 1_565 ?
Na1 Na1 Na2 101.35(18) 4_553 1_565 ?
F1 Na1 Na2 137.54(19) 1_454 1_565 ?
F1 Na1 Na2 44.84(10) 3_464 1_565 ?
F1 Na1 Na2 105.30(15) 2_564 1_565 ?
La2 Na1 Na2 116.22(12) . 1_565 ?
Na1 Na1 La2 78.65(18) 4_554 1_565 ?
F2 Na1 La2 41.22(11) 2_575 1_565 ?
F2 Na1 La2 146.0(4) . 1_565 ?
F2 Na1 La2 75.13(14) 3_355 1_565 ?
Na1 Na1 La2 101.35(18) 4_553 1_565 ?
F1 Na1 La2 137.54(19) 1_454 1_565 ?
F1 Na1 La2 44.84(10) 3_464 1_565 ?
F1 Na1 La2 105.30(15) 2_564 1_565 ?
La2 Na1 La2 116.22(12) . 1_565 ?
Na2 Na1 La2 0.0 1_565 1_565 ?
Na1 F1 Na1 59.4(5) 1_656 4_654 ?
Na1 F1 La1 101.88(19) 1_656 . ?
Na1 F1 La1 101.88(19) 4_654 . ?
Na1 F1 La2 143.0(2) 1_656 . ?
Na1 F1 La2 93.4(2) 4_654 . ?
La1 F1 La2 108.42(15) . . ?
Na1 F1 La2 93.4(2) 1_656 1_556 ?
Na1 F1 La2 143.0(2) 4_654 1_556 ?
La1 F1 La2 108.42(15) . 1_556 ?
La2 F1 La2 96.5(2) . 1_556 ?
Na1 F1 Na2 93.4(2) 1_656 1_556 ?
Na1 F1 Na2 143.0(2) 4_654 1_556 ?
La1 F1 Na2 108.42(15) . 1_556 ?
La2 F1 Na2 96.5(2) . 1_556 ?
La2 F1 Na2 0.0 1_556 1_556 ?
Na1 F2 Na1 35.5(5) 4_554 . ?
Na1 F2 La2 99.26(15) 4_554 . ?
Na1 F2 La2 99.26(15) . . ?
Na1 F2 La1 132.7(3) 4_554 1_454 ?
Na1 F2 La1 102.0(3) . 1_454 ?
La2 F2 La1 110.69(13) . 1_454 ?
Na1 F2 La1 102.0(3) 4_554 1_455 ?
Na1 F2 La1 132.7(3) . 1_455 ?
La2 F2 La1 110.69(13) . 1_455 ?
La1 F2 La1 100.50(15) 1_454 1_455 ?

```

```

_refine_diff_density_max 0.430
_refine_diff_density_min -0.512
_refine_diff_density_rms 0.099

```

```

_shelxl_version_number 2013-3

```

```

_shelx_res_file

```

```

;
TITL 558s in P-6
CELL 0.71073 6.1520 6.1520 3.8191 90.000 90.000 120.000
ZERR 1.00 0.0014 0.0014 0.0009 0.000 0.000 0.000
LATT -1
SYMM -Y, X-Y, Z
SYMM -X+Y, -X, Z
SYMM X, Y, -Z
SYMM -Y, X-Y, -Z
SYMM -X+Y, -X, -Z
SFAC F NA LA
UNIT 6 1.5 1.5
TEMP -173
SIZE 0.09 0.10 0.12
MERG 0
TWIN -1 0 0 0 -1 0 0 0 -1
EADP LA2 NA2
EXYZ LA2 NA2
FMAP 2
ACTA
PLAN 20
L.S. 15
WGHT 0.000000
EXTI 0.326174
BASf 0.42846
FVAR 0.47897

```

```

LA2 3  -0.333333  0.333333  -0.500000  10.08335  0.00457  0.00457 =
      0.01421  0.00000  0.00000  0.00228
NA2 2  -0.333333  0.333333  -0.500000  10.08335  0.00457  0.00457 =
      0.01421  0.00000  0.00000  0.00228
LA1 3  0.000000  0.000000  0.000000  10.16667  0.00652  0.00652 =
      0.00270  0.00000  0.00000  0.00326
NA1 2  -0.666667  0.666667  -0.686706  10.16667  0.01404  0.01404 =
      0.02134  0.00000  0.00000  0.00702
F1  1  -0.039665  0.370575  0.000000  10.50000  0.00787  0.00696 =
      0.03742  0.00000  0.00000  0.00447
F2  1  -0.754562  0.269261  -0.500000  10.50000  0.00851  0.01157 =
      0.00464  0.00000  0.00000  0.00317

```

HKLF 4

REM 558s in P-6

REM R1 = 0.0138 for 816 Fo > 4sig(Fo) and 0.0138 for all 816 data
 REM 22 parameters refined using 0 restraints

END

WGHT 0.0000 0.0000

REM Highest difference peak 0.430, deepest hole -0.512, 1-sigma level 0.099

```

Q1  1  -0.0381 -0.1750  0.0000  10.50000  0.05  0.43
Q2  1  0.0346  0.1800  0.0000  10.50000  0.05  0.37
Q3  1  0.2056  0.3326  0.0000  10.50000  0.05  0.30
Q4  1  -0.0647  0.2485  0.0000  10.50000  0.05  0.29
Q5  1  -0.5485  0.4276 -0.5000  10.50000  0.05  0.24
Q6  1  -0.6667  0.6667 -1.0000  10.16667  0.05  0.23
Q7  1  -0.1487  0.2479  0.3202  11.00000  0.05  0.21
Q8  1  -0.2872  0.2460  0.0000  10.50000  0.05  0.21
Q9  1  -0.5228  0.1826 -0.5000  10.50000  0.05  0.20
Q10 1  -0.8804  0.0635 -0.5000  10.50000  0.05  0.20
Q11 1  -0.0169  0.5042  0.3484  11.00000  0.05  0.20
Q12 1  -0.4611  0.5421 -0.5000  10.50000  0.05  0.19
Q13 1  -0.7353  0.4046 -0.7047  11.00000  0.05  0.19
Q14 1  -0.6667  0.6667 -0.5000  10.16667  0.05  0.19
Q15 1  -0.1726  0.1521  0.0000  10.50000  0.05  0.19
Q16 1  -0.9078  0.1105 -0.5000  10.50000  0.05  0.19
Q17 1  -0.2704  0.1044  0.0000  10.50000  0.05  0.18
Q18 1  -0.7213  0.1943 -0.8454  11.00000  0.05  0.18
Q19 1  -0.3455  0.1305 -0.5000  10.50000  0.05  0.15
Q20 1  -0.6474  0.2698 -0.5000  10.50000  0.05  0.14
;

```

A. 3.2 Neutron CIF

data_shelxl

```

_audit_creation_method      SHELXL-2013
_chemical_name_systematic
;
?
;
_chemical_name_common       ?
_chemical_melting_point     ?
_chemical_formula_moiety    ?
_chemical_formula_sum
'F6 La1.5 Na1.5'
_chemical_formula_weight    356.83

loop_
  _atom_type_symbol
  _atom_type_description
  _atom_type_scatter_dispersion_real
  _atom_type_scatter_dispersion_imag
  _atom_type_scatter_source
'F' 'F' 0.0000 0.0000
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'
'La' 'La' 0.0000 0.0000
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'
'Na' 'Na' 0.0000 0.0000
'International Tables Vol C Tables 4.2.6.8 and 6.1.1.4'

_space_group_crystal_system hexagonal
_space_group_IT_number      174
_space_group_name_H-M_alt   'P -6'
_space_group_name_Hall      'P -6'

_shelx_space_group_comment
;
The symmetry employed for this shelxl refinement is uniquely defined

```

by the following loop, which should always be used as a source of symmetry information in preference to the above space-group names. They are only intended as comments.

```

loop_
  _space_group_symop_operation_xyz
    'x, y, z'
    '-y, x-y, z'
    '-x+y, -x, z'
    'x, y, -z'
    '-y, x-y, -z'
    '-x+y, -x, -z'

  _cell_length_a          6.1568(9)
  _cell_length_b          6.1568(9)
  _cell_length_c          3.8210(8)
  _cell_angle_alpha       90
  _cell_angle_beta        90
  _cell_angle_gamma       120
  _cell_volume            125.43(5)
  _cell_formula_units_Z   1
  _cell_measurement_temperature 100(2)
  _cell_measurement_reflns_used 2127
  _cell_measurement_theta_min ?
  _cell_measurement_theta_max ?
  _exptl_crystal_description irregular
  _exptl_crystal_colour   colorless
  _exptl_crystal_density_meas ?
  _exptl_crystal_density_method none
  _exptl_crystal_density_diffrn 4.724
  _exptl_crystal_F_000    34
  _exptl_transmission_factor_min ?
  _exptl_transmission_factor_max ?
  _exptl_crystal_size_max  1.4
  _exptl_crystal_size_mid  1.4
  _exptl_crystal_size_min  1.0
  _exptl_absorpt_coefficient_mu '0.04, at 1.8 Angstrom'
  _shelx_estimated_absorpt_T_min ?
  _shelx_estimated_absorpt_T_max ?
  _exptl_absorpt_correction_type Sphere
  _exptl_absorpt_correction_T_min 0.9631
  _exptl_absorpt_correction_T_max 0.9788
  _exptl_absorpt_process_details anvred2x

  _exptl_special_details
;
?
;

  _diffrn_ambient_temperature 100(2)
  _diffrn_radiation_type      'neutron'
  _diffrn_radiation_wavelength 0.4-3.5
  _diffrn_measurement_device  'SNS single crystal diffractometer, TOPAZ'
  _diffrn_measurement_method  'time of flight laue'
  _diffrn_source              ?
  _diffrn_measurement_device_type ?
  _diffrn_detector_area_resol_mean ?
  _diffrn_reflns_number       2125
  _diffrn_reflns_av_unetI/netI 0.0624
  _diffrn_reflns_av_R_equivalents ?
  _diffrn_reflns_limit_h_min  -12
  _diffrn_reflns_limit_h_max   12
  _diffrn_reflns_limit_k_min  -11
  _diffrn_reflns_limit_k_max   12
  _diffrn_reflns_limit_l_min  -7
  _diffrn_reflns_limit_l_max   7
  _diffrn_reflns_theta_min     2.768
  _diffrn_reflns_theta_max     17.409
  _diffrn_reflns_theta_full    10.370
  _diffrn_measured_fraction_theta_max 0.967
  _diffrn_measured_fraction_theta_full 0.957
  _diffrn_reflns_Laue_measured_fraction_max 0.967
  _diffrn_reflns_Laue_measured_fraction_full 0.957
  _diffrn_reflns_point_group_measured_fraction_max 0.913
  _diffrn_reflns_point_group_measured_fraction_full 0.894
  _reflns_number_total        2125
  _reflns_number_gt           2042
  _reflns_threshold_expression 'I > 2\sigma(I)'
  _reflns_Friedel_coverage    0.871
  _reflns_Friedel_fraction_max 0.858
  _reflns_Friedel_fraction_full 0.830

  _reflns_special_details
;
  _reflns_Friedel_fraction is defined as the number of unique
  Friedel pairs measured divided by the number that would be
  possible theoretically, ignoring centric projections and

```

```

systematic absences.
;

_computing_data_collection      ?
_computing_cell_refinement     ?
_computing_data_reduction      ?
_computing_structure_solution  ?
_computing_structure_refinement 'SHELXL-2013 (Sheldrick, 2013)'
_computing_molecular_graphics  ?
_computing_publication_material ?

_refine_special_details
;
?
;
_refine_ls_structure_factor_coef  Fsqd
_refine_ls_matrix_type           full
_refine_ls_weighting_scheme      calc
_refine_ls_weighting_details     'w=1/[s^2+(Fo^2)+(0.1000P)^2] where P=(Fo^2+2Fc^2)/3'
_refine_ls_extinction_coef       '1.3560 x 10^-04'
_refine_ls_extinction_method     'GSAS Secondary Type I Lorentzian model'
_atom_sites_solution_primary     ?
_atom_sites_solution_secondary  ?
_atom_sites_solution_hydrogens  .
_refine_ls_hydrogen_treatment    undef
_refine_ls_abs_structure_details
;
All f" are zero, so absolute structure could not be determined
;
_refine_ls_abs_structure_Flack   ?
_chemical_absolute_configuration ?
_refine_ls_number_reflns        2125
_refine_ls_number_parameters    20
_refine_ls_number_restraints    0
_refine_ls_R_factor_all         0.0557
_refine_ls_R_factor_gt          0.0534
_refine_ls_wR_factor_ref        0.1354
_refine_ls_wR_factor_gt        0.1335
_refine_ls_goodness_of_fit_ref  1.025
_refine_ls_restrained_S_all     1.025
_refine_ls_shift/su_max         0.000
_refine_ls_shift/su_mean        0.000

loop_
_atom_site_label
_atom_site_type_symbol
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_U_iso_or_equiv
_atom_site_adp_type
_atom_site_occupancy
_atom_site_site_symmetry_order
_atom_site_calc_flag
_atom_site_refinement_flags_posn
_atom_site_refinement_flags_adp
_atom_site_refinement_flags_occupancy
_atom_site_disorder_assembly
_atom_site_disorder_group
La1 La 0.0000 0.0000 0.0000 0.0043(2) Uani 1 6 d S T P . .
La2 La 0.6667 0.3333 0.5000 0.0087(4) Uani 0.5001 6 d S T P . .
Na2 Na 0.6667 0.3333 0.5000 0.0087(4) Uani 0.5001 6 d S T P . .
Na1 Na 0.3333 0.6667 0.675(3) 0.0215(14) Uani 0.5 3 d S T P . .
F1 F 0.6282(3) 0.0374(4) 0.0000 0.0205(5) Uani 1 2 d S T P . .
F2 F 0.7306(3) 0.7540(3) 0.5000 0.0084(2) Uani 1 2 d S T P . .

loop_
_atom_site_aniso_label
_atom_site_aniso_U_11
_atom_site_aniso_U_22
_atom_site_aniso_U_33
_atom_site_aniso_U_23
_atom_site_aniso_U_13
_atom_site_aniso_U_12
La1 0.0053(3) 0.0053(3) 0.0024(4) 0.000 0.000 0.00263(15)
La2 0.0062(5) 0.0062(5) 0.0137(9) 0.000 0.000 0.0031(2)
Na2 0.0062(5) 0.0062(5) 0.0137(9) 0.000 0.000 0.0031(2)
Na1 0.0156(13) 0.0156(13) 0.034(4) 0.000 0.000 0.0078(7)
F1 0.0083(5) 0.0125(6) 0.0427(13) 0.000 0.000 0.0067(5)
F2 0.0084(5) 0.0070(4) 0.0066(4) 0.000 0.000 0.0014(4)

_geom_special_details
;
All esds (except the esd in the dihedral angle between two l.s. planes)
are estimated using the full covariance matrix. The cell esds are taken
into account individually in the estimation of esds in distances, angles
and torsion angles; correlations between esds in cell parameters are only

```

used when they are defined by crystal symmetry. An approximate (isotropic) treatment of cell esds is used for estimating esds involving l.s. planes.

```
;  
  
loop_  
_geom_bond_atom_site_label_1  
_geom_bond_atom_site_label_2  
_geom_bond_distance  
_geom_bond_site_symmetry_2  
_geom_bond_publ_flag  
La1 F1 2.4126(18) 3_665 ?  
La1 F1 2.4126(18) 2_545 ?  
La1 F1 2.4126(18) 1_455 ?  
La1 F2 2.4866(10) 3_565 ?  
La1 F2 2.4866(10) 3_564 ?  
La1 F2 2.4866(10) 1_444 ?  
La1 F2 2.4866(10) 1_445 ?  
La1 F2 2.4866(10) 2_655 ?  
La1 F2 2.4866(10) 2_654 ?  
La1 Na1 3.766(3) 4_556 ?  
La1 Na1 3.766(3) 1_444 ?  
La1 Na1 3.766(3) 4_446 ?  
La2 F2 2.4174(15) 2_655 ?  
La2 F2 2.4174(15) . ?  
La2 F2 2.4175(15) 3_665 ?  
La2 F1 2.5678(13) . ?  
La2 F1 2.5678(13) 2_656 ?  
La2 F1 2.5678(13) 1_556 ?  
La2 F1 2.5678(13) 2_655 ?  
La2 F1 2.5678(13) 3_665 ?  
La2 F1 2.5678(13) 3_666 ?  
La2 Na1 3.6167(19) 4_556 ?  
La2 Na1 3.6167(19) . ?  
La2 Na1 3.6167(19) 4_546 ?  
Na1 Na1 1.33(2) 4_556 ?  
Na1 F2 2.324(3) 2_665 ?  
Na1 F2 2.324(3) 3_565 ?  
Na1 F2 2.324(3) . ?  
Na1 F1 2.431(5) 1_566 ?  
Na1 F1 2.431(5) 2_556 ?  
Na1 F1 2.431(5) 3_666 ?  
Na1 Na1 2.49(2) 4_557 ?  
Na1 Na2 3.6167(19) 1_455 ?  
Na1 La2 3.6167(19) 1_455 ?  
F1 La1 2.4126(18) 1_655 ?  
F1 Na1 2.431(5) 1_544 ?  
F1 Na1 2.431(5) 4_546 ?  
F1 La2 2.5678(13) 1_554 ?  
F1 Na2 2.5678(13) 1_554 ?  
F2 Na1 2.324(3) 4_556 ?  
F2 La1 2.4866(10) 1_666 ?  
F2 La1 2.4866(10) 1_665 ?  
  
loop_  
_geom_angle_atom_site_label_1  
_geom_angle_atom_site_label_2  
_geom_angle_atom_site_label_3  
_geom_angle  
_geom_angle_site_symmetry_1  
_geom_angle_site_symmetry_3  
_geom_angle_publ_flag  
F1 La1 F1 120.0 3_665 2_545 ?  
F1 La1 F1 120.0 3_665 1_455 ?  
F1 La1 F1 120.0 2_545 1_455 ?  
F1 La1 F2 71.48(4) 3_665 3_565 ?  
F1 La1 F2 129.80(3) 2_545 3_565 ?  
F1 La1 F2 71.19(4) 1_455 3_565 ?  
F1 La1 F2 71.48(4) 3_665 3_564 ?  
F1 La1 F2 129.80(3) 2_545 3_564 ?  
F1 La1 F2 71.19(4) 1_455 3_564 ?  
F2 La1 F2 100.41(5) 3_565 3_564 ?  
F1 La1 F2 129.80(3) 3_665 1_444 ?  
F1 La1 F2 71.19(4) 2_545 1_444 ?  
F1 La1 F2 71.48(4) 1_455 1_444 ?  
F2 La1 F2 142.67(2) 3_565 1_444 ?  
F2 La1 F2 67.33(4) 3_564 1_444 ?  
F1 La1 F2 129.80(3) 3_665 1_445 ?  
F1 La1 F2 71.19(4) 2_545 1_445 ?  
F1 La1 F2 71.48(4) 1_455 1_445 ?  
F2 La1 F2 67.33(4) 3_565 1_445 ?  
F2 La1 F2 142.67(2) 3_564 1_445 ?  
F2 La1 F2 100.41(5) 1_444 1_445 ?  
F1 La1 F2 71.19(4) 3_665 2_655 ?  
F1 La1 F2 71.48(4) 2_545 2_655 ?  
F1 La1 F2 129.80(3) 1_455 2_655 ?  
F2 La1 F2 67.33(4) 3_565 2_655 ?  
F2 La1 F2 142.67(2) 3_564 2_655 ?  
F2 La1 F2 142.67(2) 1_444 2_655 ?
```

F2 La1 F2 67.33(4) 1_445 2_655 ?
F1 La1 F2 71.19(4) 3_665 2_654 ?
F1 La1 F2 71.48(4) 2_545 2_654 ?
F1 La1 F2 129.80(3) 1_455 2_654 ?
F2 La1 F2 142.67(2) 3_565 2_654 ?
F2 La1 F2 67.33(4) 3_564 2_654 ?
F2 La1 F2 67.33(4) 1_444 2_654 ?
F2 La1 F2 142.67(2) 1_445 2_654 ?
F2 La1 F2 100.41(5) 2_655 2_654 ?
F1 La1 Na1 39.14(7) 3_665 4_556 ?
F1 La1 Na1 148.62(9) 2_545 4_556 ?
F1 La1 Na1 85.52(4) 1_455 4_556 ?
F2 La1 Na1 36.97(13) 3_565 4_556 ?
F2 La1 Na1 73.04(14) 3_564 4_556 ?
F2 La1 Na1 138.73(12) 1_444 4_556 ?
F2 La1 Na1 104.14(14) 1_445 4_556 ?
F2 La1 Na1 78.10(11) 2_655 4_556 ?
F2 La1 Na1 107.52(11) 2_654 4_556 ?
F1 La1 Na1 148.62(9) 3_665 1_444 ?
F1 La1 Na1 85.52(4) 2_545 1_444 ?
F1 La1 Na1 39.14(7) 1_455 1_444 ?
F2 La1 Na1 107.52(11) 3_565 1_444 ?
F2 La1 Na1 78.10(11) 3_564 1_444 ?
F2 La1 Na1 36.97(13) 1_444 1_444 ?
F2 La1 Na1 73.04(14) 1_445 1_444 ?
F2 La1 Na1 138.73(12) 2_655 1_444 ?
F2 La1 Na1 104.14(14) 2_654 1_444 ?
Na1 La1 Na1 123.68(5) 4_556 1_444 ?
F1 La1 Na1 148.62(9) 3_665 4_446 ?
F1 La1 Na1 85.52(4) 2_545 4_446 ?
F1 La1 Na1 39.14(7) 1_455 4_446 ?
F2 La1 Na1 78.10(11) 3_565 4_446 ?
F2 La1 Na1 107.52(11) 3_564 4_446 ?
F2 La1 Na1 73.04(14) 1_444 4_446 ?
F2 La1 Na1 36.97(13) 1_445 4_446 ?
F2 La1 Na1 104.14(14) 2_655 4_446 ?
F2 La1 Na1 138.72(12) 2_654 4_446 ?
Na1 La1 Na1 109.66(14) 4_556 4_446 ?
Na1 La1 Na1 38.6(3) 1_444 4_446 ?
F2 La2 F2 120.0 2_655 . ?
F2 La2 F2 120.0 2_655 3_665 ?
F2 La2 F2 120.0 . 3_665 ?
F2 La2 F1 71.25(4) 2_655 . ?
F2 La2 F1 131.91(3) . . ?
F2 La2 F1 69.73(4) 3_665 . ?
F2 La2 F1 131.91(3) 2_655 2_656 ?
F2 La2 F1 69.73(4) . 2_656 ?
F2 La2 F1 71.25(4) 3_665 2_656 ?
F1 La2 F1 140.97(3) . 2_656 ?
F2 La2 F1 71.25(4) 2_655 1_556 ?
F2 La2 F1 131.91(3) . 1_556 ?
F2 La2 F1 69.73(4) 3_665 1_556 ?
F1 La2 F1 96.15(7) . 1_556 ?
F1 La2 F1 70.71(5) 2_656 1_556 ?
F2 La2 F1 131.91(3) 2_655 2_655 ?
F2 La2 F1 69.73(4) . 2_655 ?
F2 La2 F1 71.25(4) 3_665 2_655 ?
F1 La2 F1 70.71(5) . 2_655 ?
F1 La2 F1 96.15(7) 2_656 2_655 ?
F1 La2 F1 140.97(3) 1_556 2_655 ?
F2 La2 F1 69.73(4) 2_655 3_665 ?
F2 La2 F1 71.25(4) . 3_665 ?
F2 La2 F1 131.91(3) 3_665 3_665 ?
F1 La2 F1 70.71(5) . 3_665 ?
F1 La2 F1 140.97(3) 2_656 3_665 ?
F1 La2 F1 140.97(3) 1_556 3_665 ?
F1 La2 F1 70.71(5) 2_655 3_665 ?
F2 La2 F1 69.73(4) 2_655 3_666 ?
F2 La2 F1 71.25(4) . 3_666 ?
F2 La2 F1 131.91(3) 3_665 3_666 ?
F1 La2 F1 140.97(3) . 3_666 ?
F1 La2 F1 70.71(5) 2_656 3_666 ?
F1 La2 F1 70.71(5) 1_556 3_666 ?
F1 La2 F1 140.97(3) 2_655 3_666 ?
F1 La2 F1 96.15(7) 3_665 3_666 ?
F2 La2 Na1 82.04(3) 2_655 4_556 ?
F2 La2 Na1 39.34(5) . 4_556 ?
F2 La2 Na1 155.77(7) 3_665 4_556 ?
F1 La2 Na1 112.84(14) . 4_556 ?
F1 La2 Na1 102.46(12) 2_656 4_556 ?
F1 La2 Na1 131.50(14) 1_556 4_556 ?
F1 La2 Na1 86.63(12) 2_655 4_556 ?
F1 La2 Na1 42.17(15) 3_665 4_556 ?
F1 La2 Na1 62.18(15) 3_666 4_556 ?
F2 La2 Na1 82.04(3) 2_655 . ?
F2 La2 Na1 39.34(5) . . ?
F2 La2 Na1 155.77(7) 3_665 . ?
F1 La2 Na1 131.50(14) . . ?

F1 La2 Na1 86.63(12) 2_656 . ?
 F1 La2 Na1 112.84(14) 1_556 . ?
 F1 La2 Na1 102.46(12) 2_655 . ?
 F1 La2 Na1 62.18(15) 3_665 . ?
 F1 La2 Na1 42.17(15) 3_666 . ?
 Na1 La2 Na1 21.3(3) 4_556 . ?
 F2 La2 Na1 39.34(5) 2_655 4_546 ?
 F2 La2 Na1 155.77(7) . 4_546 ?
 F2 La2 Na1 82.04(3) 3_665 4_546 ?
 F1 La2 Na1 42.17(15) . 4_546 ?
 F1 La2 Na1 131.50(14) 2_656 4_546 ?
 F1 La2 Na1 62.18(15) 1_556 4_546 ?
 F1 La2 Na1 112.84(14) 2_655 4_546 ?
 F1 La2 Na1 86.63(12) 3_665 4_546 ?
 F1 La2 Na1 102.46(12) 3_666 4_546 ?
 Na1 La2 Na1 116.68(10) 4_556 4_546 ?
 Na1 La2 Na1 121.13(4) . 4_546 ?
 Na1 Na1 F2 73.3(2) 4_556 2_665 ?
 Na1 Na1 F2 73.3(2) 4_556 3_565 ?
 F2 Na1 F2 112.1(2) 2_665 3_565 ?
 Na1 Na1 F2 73.3(2) 4_556 . ?
 F2 Na1 F2 112.1(2) 2_665 . ?
 F2 Na1 F2 112.1(2) 3_565 . ?
 Na1 Na1 F1 120.8(2) 4_556 1_566 ?
 F2 Na1 F1 75.32(7) 2_665 1_566 ?
 F2 Na1 F1 165.9(4) 3_565 1_566 ?
 F2 Na1 F1 73.98(7) . 1_566 ?
 Na1 Na1 F1 120.8(2) 4_556 2_556 ?
 F2 Na1 F1 73.98(7) 2_665 2_556 ?
 F2 Na1 F1 75.32(7) 3_565 2_556 ?
 F2 Na1 F1 165.9(4) . 2_556 ?
 F1 Na1 F1 96.2(3) 1_566 2_556 ?
 Na1 Na1 F1 120.8(2) 4_556 3_666 ?
 F2 Na1 F1 165.9(4) 2_665 3_666 ?
 F2 Na1 F1 73.98(7) 3_565 3_666 ?
 F2 Na1 F1 75.32(7) . 3_666 ?
 F1 Na1 F1 96.2(3) 1_566 3_666 ?
 F1 Na1 F1 96.2(3) 2_556 3_666 ?
 Na1 Na1 Na1 180.0 4_556 4_557 ?
 F2 Na1 Na1 106.7(2) 2_665 4_557 ?
 F2 Na1 Na1 106.7(2) 3_565 4_557 ?
 F2 Na1 Na1 106.7(2) . 4_557 ?
 F1 Na1 Na1 59.2(2) 1_566 4_557 ?
 F1 Na1 Na1 59.2(2) 2_556 4_557 ?
 F1 Na1 Na1 59.2(2) 3_666 4_557 ?
 Na1 Na1 La2 79.37(15) 4_556 . ?
 F2 Na1 La2 147.4(3) 2_665 . ?
 F2 Na1 La2 75.53(8) 3_565 . ?
 F2 Na1 La2 41.26(4) . . ?
 F1 Na1 La2 104.95(8) 1_566 . ?
 F1 Na1 La2 136.89(16) 2_556 . ?
 F1 Na1 La2 45.17(5) 3_666 . ?
 Na1 Na1 La2 100.63(15) 4_557 . ?
 Na1 Na1 Na2 79.37(16) 4_556 1_455 ?
 F2 Na1 Na2 75.53(8) 2_665 1_455 ?
 F2 Na1 Na2 41.26(4) 3_565 1_455 ?
 F2 Na1 Na2 147.4(3) . 1_455 ?
 F1 Na1 Na2 136.89(16) 1_566 1_455 ?
 F1 Na1 Na2 45.17(5) 2_556 1_455 ?
 F1 Na1 Na2 104.95(8) 3_666 1_455 ?
 Na1 Na1 Na2 100.63(15) 4_557 1_455 ?
 La2 Na1 Na2 116.68(9) . 1_455 ?
 Na1 Na1 La2 79.37(16) 4_556 1_455 ?
 F2 Na1 La2 75.53(8) 2_665 1_455 ?
 F2 Na1 La2 41.26(4) 3_565 1_455 ?
 F2 Na1 La2 147.4(3) . 1_455 ?
 F1 Na1 La2 136.89(16) 1_566 1_455 ?
 F1 Na1 La2 45.17(5) 2_556 1_455 ?
 F1 Na1 La2 104.95(8) 3_666 1_455 ?
 Na1 Na1 La2 100.63(15) 4_557 1_455 ?
 La2 Na1 La2 116.68(9) . 1_455 ?
 Na2 Na1 La2 0.0 1_455 1_455 ?
 La1 F1 Na1 102.07(7) 1_655 1_544 ?
 La1 F1 Na1 102.07(7) 1_655 4_546 ?
 Na1 F1 Na1 61.5(4) 1_544 4_546 ?
 La1 F1 La2 108.21(5) 1_655 . ?
 Na1 F1 La2 143.85(15) 1_544 . ?
 Na1 F1 La2 92.66(18) 4_546 . ?
 La1 F1 La2 108.21(5) 1_655 1_554 ?
 Na1 F1 La2 92.66(18) 1_544 1_554 ?
 Na1 F1 La2 143.85(15) 4_546 1_554 ?
 La2 F1 La2 96.15(7) . 1_554 ?
 La1 F1 Na2 108.21(5) 1_655 1_554 ?
 Na1 F1 Na2 92.66(18) 1_544 1_554 ?
 Na1 F1 Na2 143.85(15) 4_546 1_554 ?
 La2 F1 Na2 96.15(7) . 1_554 ?
 La2 F1 Na2 0.0 1_554 1_554 ?
 Na1 F2 Na1 33.4(5) 4_556 . ?

```

Na1 F2 La2 99.40(5) 4_556 . ?
Na1 F2 La2 99.40(5) . . ?
Na1 F2 La1 131.8(2) 4_556 1_666 ?
Na1 F2 La1 103.0(2) . 1_666 ?
La2 F2 La1 110.74(4) . 1_666 ?
Na1 F2 La1 103.0(2) 4_556 1_665 ?
Na1 F2 La1 131.8(2) . 1_665 ?
La2 F2 La1 110.74(4) . 1_665 ?
La1 F2 La1 100.41(5) 1_666 1_665 ?

_refine_diff_density_max 1.530
_refine_diff_density_min -1.558
_refine_diff_density_rms 0.277

_shelxl_version_number 2013-3

_shelx_res_file
;
TITL turned in space group P-6
CELL 0.3000 6.1568 6.1568 3.8210 90.000 90.000 120.000
ZERR 1 0.0009 0.0009 0.0008 0.000 0.000 0.000
LATT -1
SYMM - Y, + X - Y, + Z
SYMM - X + Y, - X, + Z
SYMM + X, + Y, - Z
SYMM - Y, + X - Y, - Z
SYMM - X + Y, - X, - Z
SFAC F 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 =
0.00000 0.00000 5.65400 0.000 0.000 0.005 0.640 18.998
SFAC La 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 =
0.00000 0.00000 8.24000 0.000 0.000 4.676 1.880 138.906
SFAC Na 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 =
0.00000 0.00000 3.63000 0.000 0.000 1.830 1.860 22.990
UNIT 4.0 1.0 1.0
TEMP -173
EADP La2 Na2
EXYZ La2 Na2
merg 0
REM sump 12 0.001 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 9 1 10 1 11 1 12 1 13
L.S. 25
ACTA
FMAP 2
PLAN 10
OMIT -2 2 -1
OMIT 7 -8 0
REM EXTI 40.107571
REM BASF 1.07008 1.04449 0.95501 0.96325 1.02096 1.04133 1.01059
REM BASF 0.98323 0.87672 0.93149 0.93786 0.97558 0.97210
WGHT 0.100000
FVAR 0.10118
LA1 2 0.000000 0.000000 0.000000 10.16667 0.00525 0.00525 =
0.00241 0.00000 0.00000 0.00263
LA2 2 0.666667 0.333333 0.500000 10.08335 0.00615 0.00615 =
0.01372 0.00000 0.00000 0.00308
NA2 3 0.666667 0.333333 0.500000 10.08335 0.00615 0.00615 =
0.01372 0.00000 0.00000 0.00308
NA1 3 0.333333 0.666667 0.674616 10.16667 0.01556 0.01556 =
0.03351 0.00000 0.00000 0.00778
F1 1 0.628204 0.037435 0.000000 10.50000 0.00833 0.01246 =
0.04267 0.00000 0.00000 0.00667
F2 1 0.730573 0.754013 0.500000 10.50000 0.00842 0.00698 =
0.00658 0.00000 0.00000 0.00135
HKLF 4

REM turned in space group P-6
REM R1 = 0.0534 for 2042 Fo > 4sig(Fo) and 0.0557 for all 2125 data
REM 20 parameters refined using 0 restraints

END

WGHT 0.0796 0.2175

REM Highest difference peak 1.530, deepest hole -1.558, 1-sigma level 0.277
Q1 1 0.6667 0.3333 0.3495 10.33333 0.05 1.12
Q2 1 0.6453 0.7238 0.4100 11.00000 0.05 1.11
Q3 1 -0.0968 0.0120 0.0900 11.00000 0.05 1.03
Q4 1 0.4697 0.5381 0.5000 10.50000 0.05 0.94
Q5 1 0.0949 0.0074 0.0829 11.00000 0.05 0.86
Q6 1 0.6470 0.8174 0.5000 10.50000 0.05 0.81
Q7 1 0.6836 0.1415 0.1243 11.00000 0.05 0.77
Q8 1 0.6894 0.5320 0.5000 10.50000 0.05 0.76
Q9 1 0.0000 0.0000 0.1532 10.33333 0.05 0.73
Q10 1 0.4015 0.7207 0.5000 10.50000 0.05 0.70
;

```


A. 4 ZODS details

A. 4.1 xml input

ZODS xml input file for local structure model of NaLaF₄ below:

```
<?xml version="1.0" encoding="UTF-8"?>
<input xsi:noNamespaceSchemaLocation="simulator.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" print_level="3">
  <reference_intensities file_format="xml" file_name="nalaf4_small_mirrored_noneg.xml"/>
  <average_structure>
    <file name="nalaf4_F(splitpositions).cif" format="cif"/>
    <refinable_parameters>
      <coordinate atom="F1" coordinate="c" parameterized_with="z_shift"/>
    </refinable_parameters>
  </average_structure>
  <disentanglement translational_symmetry_directions="c">
    <user_defined>
      <chemical_units_list>
        <chemical_unit label="cu1" probability="0.5" atom_list="Na2,(X,Y,Z+1/2);F1,(1-X+Y,1-X,Z);F1;F1,(X,Y,1/2-Z);
        F1,(1-Y,X-Y,Z);F1,(1-Y,X-Y,1/2-Z);F1,(1-X+Y,1-X,1/2-Z);La2"/>
      </chemical_units_list>
      <sets_of_alternatives_list>
        <set_of_alternatives label="soa" chemical_units_list="cu1;cu1,(X,Y,-Z+1)"/>
      </sets_of_alternatives_list>
    </user_defined>
  </disentanglement>
  <independent_parameters>
    <parameter label="p1" value="0.273"/>
    <parameter label="z_shift" value="0.008078"/>
  </independent_parameters>
  <interactions>
    <interaction>
      <discrete_parameters="p1">
        <chemical_units>cu1;cu1,(1,0,0)</chemical_units>
      </discrete>
    </interaction>
  </interactions>
  <disordered_crystal_simulation>
    <default_crystal_builder crystal_dimensions="100,100,1" n_MC_cycles="100"/>
  </disordered_crystal_simulation>
  <intensity_calculation form_factor_type="x-ray" lot_dimensions="20,20,1" n_lots="25"/>
  <run_type>
    <optimize_model n_clones="20">
      <output>
        <crystals output_file_formats="xml,pdb,xyz" only_winner="true"/>
        <intensities output_file_formats="xml,vtk" print_clone_intensities="false" only_winner="false"/>
      </output>
      <optimization_method>
        <differential_evolution n_gene_sets="48" n_generations="36">
          <constraints>
            <parameter name="z_shift" min="-0.02" max="0.025" initial_max="0.02" initial_min="0.007"/>
            <parameter name="p1" min="-0.1" max="1.4" initial_max="1.0" initial_min="0.4"/>
          </constraints>
        </differential_evolution>
      </optimization_method>
    </optimize_model>
  </run_type>
</input>
```

A. 4.2 Plotting tools in ZODS

A short python 3.2.3 code was written to parse ZODS output (diff_ev) containing the model fitness and parameter results averaged over clones for each individual for every generation and calculate the statistics for the objective function and each parameter. The mean, minimum, maximum and standard deviation over all individuals for each generation are output into individual text files corresponding to each parameter.

A gnu plot script plots the statistics from the text file(s) as a function of generation number and outputs the plot as a postscript file. A shell script is available that

generates the appropriate gnu script to plot the individual parameters. The gnu script to plot the parameters then outputs plots of each parameter's convergence as a function of generation number. The postscript plots can be converted by the user to pdf files (half the size) by using a pdf shell script converter program.

General python statistics code, reads in diff_ev file and calculates the objective function and parameter convergence statistics:

```
#!/usr/bin/env python3

#Program purpose: To parse diff_ev file output from ZODS and calculate the mean,
#standard deviation of each parameter and objective function over individuals for
#each generation. Final goal is to plot the convergence of each parameter and the
#objective function as a function of generation.

#File must be in the format of ZODS_01

import sys
import math
import numpy
import statistics
#takes user defined file name (designed to read diff_ev (txt) from ZODS_01)
print( "Reading file %s" % (sys.argv[1] ) )

f = open(sys.argv[1])

## Data struct for holding information
de={}
gennum = "unknown"
for rline in f:
    ## remove extra newline at the end of the raw line
    line = rline[:-1]
    cols = line.split() #splits reading line at first white space

    if line!="" and cols[0] == "Generation":
        # print( "%s" % ( line ) )
        gennum = int( cols[1] )
        de[ gennum ] = {}
        parse=True
    #gets all the generation numbers to index the dictionary

    elif parse and line != "" and cols[0][0] == "#":
        ## we have an individual (starts getting information for each individual)
        #gets objective function values
        res1=line.split( "objective function = " )
        #splits again to start getting each parameter
        res2 = res1[1].split( "parameters = " )

        value = float(res2[0])

        res3 = res2[1]
```

```

    params = res3.split() #start parsing out the parameters
    total = len(params) #find out how many parameters are in the file
    #create list for parameter values
    p=[]

    for i in range(total): # goes over all parameters
        if i % 2 != 0: #Skips strings and gets parameter values only
            p.append(float(params[i]))
#puts however many parameters are in the file into the list and converts them to a float

    value = (value, p) #builds the parameter list

    de[ gennum ][int(cols[0][1:])] = value
    #dictionary constructed with generation
    # as first type and individual number as the second type
    #containing objective function and parameters

    #stops parsing at "WINNER" (don't need the intermediate trial individuals)
    elif line != "" and line[:6] == "WINNER":
        parse=False

f.close()
f_out=open("Objective_function_stats.txt", "w")
ob_min=[]
ob_max=[]
ob_mean=[]
ob_stddev=[]
ob_stats={}

## iterate over the generations
for i in de:
    sum = 0
    count = 0

    obj_values=[]
    ## iterate over the individuals
    for j in de[i]:
        ## count the number of iterations
        obj_values.append(de[i][j][0])
        count += 1

    ob_max.append(max(obj_values))
    ob_stddev.append(numpy.std(obj_values))
    ob_min.append(min(obj_values))
    ob_mean.append(statistics.mean(obj_values))
    ob_stats[i]="
    ob_stats[i] += "%s %f %f %f %f\n" % (str(i), ob_min[i], ob_max[i], ob_mean[i], ob_stddev[i])
    f_out.write(ob_stats[i])
f_out.write("%s" % "#   Min   Max   Mean\tstandard deviation")
f_out.close()

#next we need to do generate the same statistics for each parameter contained in diff_ev

params=len(p)
param={}
p_minimum={}
p_maximum={}
p_mean={}

```

```

p_values={}
p_stddev={}
P_values={}
p_info=[]
output={}
for i in de:
    sum2 = 0
    count2 = 0
    ## iterate over the iterations
    p_values={}
    for j in de[i]:
        count2+=1
        for k in range( len(de[i][j][1] ) ):
            if k not in p_values:
                p_values[k] = []

            p_values[k].append(de[i][j][1][k])

    ## processing for one generation is complete
    for q in p_values:

        p_mean = statistics.mean( p_values[q] )
        p_min = min( p_values[q] )
        p_max = max( p_values[q] )
        p_stddev = numpy.std(p_values[q])
        if q not in output:
            output[q]=" "
        output[q] += "%s %f %f %f %f \n" % (str(i),p_min, p_max,p_mean,p_stddev)

for i in output:
    with open('parameter_stats%i.txt' %i, 'w') as f_out_p:
        ## iterate over the generations
        f_out_p.write("##For parameter p %d: \n" % (i+1))
        f_out_p.write("%s" % "# Min\tMax      Mean\tstandard deviation\n")
        f_out_p.write(output[i])
        f_out_p.close()

```

The output file containing the statistics of the objective function:

```

0 0.306293 0.313222 0.308943 0.001775
1 0.306293 0.311586 0.308124 0.001157
2 0.306293 0.308700 0.307631 0.000595
3 0.306212 0.308700 0.307291 0.000548
4 0.306212 0.308694 0.307268 0.000522
5 0.306212 0.308435 0.307099 0.000481
6 0.306212 0.308435 0.307061 0.000464
7 0.306212 0.308435 0.307054 0.000462
8 0.306212 0.308435 0.306976 0.000419
9 0.306212 0.307838 0.306936 0.000355
10 0.306212 0.307838 0.306912 0.000361
11 0.306212 0.307733 0.306881 0.000349
12 0.306212 0.307733 0.306845 0.000337
13 0.306212 0.307733 0.306839 0.000337
14 0.306212 0.307733 0.306825 0.000334
15 0.306212 0.307733 0.306816 0.000333
16 0.306212 0.307466 0.306796 0.000316
17 0.306212 0.307398 0.306772 0.000295
18 0.306212 0.307310 0.306729 0.000271
19 0.306212 0.307310 0.306711 0.000273
20 0.306212 0.307310 0.306706 0.000273
# Min Max Mean standard deviation

```

Parameter 0 (p1 parameter defining the ising interaction parameter between chemical units)

```

#For parameter p 1:
# Min      Max      Mean      standard deviation
0 0.871592 0.947877 0.914785 0.020877
1 0.817536 0.976510 0.916859 0.024779
2 0.876745 0.988153 0.919943 0.023589
3 0.866409 0.978720 0.915913 0.023552
4 0.867344 0.978720 0.916835 0.023709
5 0.867344 0.978720 0.919562 0.024088
6 0.867344 0.978720 0.920443 0.023773
7 0.867344 0.978720 0.919482 0.025263
8 0.867124 0.976510 0.915579 0.023266
9 0.867124 0.976510 0.914189 0.022796
10 0.867124 0.976510 0.913743 0.023257
11 0.867124 0.976510 0.914067 0.022737
12 0.867124 0.957225 0.912683 0.020367
13 0.867124 0.957225 0.912736 0.020304
14 0.867124 0.956781 0.911299 0.019552
15 0.867124 0.956781 0.911940 0.019799
16 0.867124 0.956781 0.909032 0.019600
17 0.867124 0.956781 0.911238 0.019751
18 0.867124 0.956781 0.912057 0.019264
19 0.867124 0.956781 0.912943 0.018349
20 0.867124 0.956781 0.914691 0.018208

```

Parameter 1 (F1 shift):

```

#For parameter p 2:
# Min      Max      Mean      standard deviation
0 0.007708 0.008944 0.008302 0.000372
1 0.007444 0.009397 0.008381 0.000460
2 0.007070 0.009688 0.008411 0.000557
3 0.007444 0.009707 0.008475 0.000586
4 0.007444 0.009688 0.008496 0.000558
5 0.007184 0.009778 0.008436 0.000600
6 0.007184 0.009778 0.008421 0.000594
7 0.007184 0.009778 0.008476 0.000591
8 0.007184 0.009778 0.008428 0.000603
9 0.007202 0.009688 0.008453 0.000596
10 0.007202 0.009786 0.008541 0.000610
11 0.007030 0.009786 0.008483 0.000652
12 0.007030 0.010445 0.008662 0.000695
13 0.007030 0.010445 0.008658 0.000681
14 0.007030 0.010445 0.008656 0.000687
15 0.007030 0.010445 0.008621 0.000674
16 0.007030 0.010445 0.008688 0.000650
17 0.007030 0.010551 0.008680 0.000708
18 0.007030 0.010551 0.008743 0.000688
19 0.007030 0.010551 0.008778 0.000716
20 0.007030 0.010551 0.008790 0.000750

```

Vita

Tara Michels-Clark was born November 23, 1979 in San Antonio, TX. She received her B.S. in Mathematics and a minor in Physics from King College in May, 1999. From there, she went on to earn a M.S. in Mathematics from East Tennessee State University in 2003. After teaching undergraduate mathematics for five years as full time faculty at Nashville State Community College and Middle Tennessee State University, she received funding from the Swiss National Science Foundation to pursue a Ph.D. in Theoretical Physical Chemistry along with an interdisciplinary graduate minor in computational sciences at the University of Tennessee, Knoxville (UTK) in 2009 and work on a multinational, inter-discipline research project to develop methods for elucidating the local structure of disordered crystalline materials. She was mentored by Christina Hoffmann (Spallation Neutron Source, ORNL) and collaborating with the University of Zürich, University of Bern and ETH Zürich. Her advisor at UTK was Robert Harrison followed by R. J. Hinde. Tara's research interests include mathematical modeling of structural disorder in crystalline materials, diffuse scattering, convergence and statistical analysis for numerical modeling processes, genetic algorithms, Monte Carlo modeling and high performance computing for scientific applications. Her graduate work was funded by Oak Ridge National Laboratory, supported by the Division of Scientific User Facilities, Office of Basic Energy Sciences, US Department of Energy, under contract DE-AC05-00OR22725 with UT Battelle, LLC. and the Swiss National Science Foundation. She completed her doctoral studies in May, 2014.