Mathematical Explorations in Modern X-ray Crystallography

LAURA NICOLA MIDGLEY

Mathematical Explorations in Modern X-ray Crystallography

Laura Midgley

A Thesis presented for the degree of
Doctor of Philosophy

Pure Mathematics
Department of Mathematical Sciences
Durham University
United Kingdom

December 2022

Mathematical Explorations in Modern X-ray Crystallography

Laura Midgley

Submitted for the degree of Doctor of Philosophy

December 2022

Abstract: In this thesis, we explore crystallography through a mathematical lens. We review the basics of crystallography with a mathematical focus, and expand into contributions on two specific areas of the crystallographic refinement process. The first of these is the detection of twin components within crystals using the effect of twinning on detected diffraction peaks. We focus on using the information of particularly underestimated peaks along with the lattice structure to intelligently search for the most viable twin laws. The second contribution concerns the use of non-spherical form factors in crystallographic refinement, and testing of the impact of setting the form factor derivative to zero within the least-squares refinement process. We utilise numerical differentiation to approximate this derivative more exactly, and evaluate the impact of these choices for modelling the derivative through three test molecules to find that, within the current bounds of uncertainty, modelling the form factor derivative as zero has insignificant impact on the results of refinement. Additional curiosities encountered within our investigations of crystallography are also documented, such as the implementation of extinction parameters.

Declaration

The work in this thesis is based on research carried out in the Department of Mathematical Sciences at Durham University. No part of this thesis has been submitted elsewhere for any degree or qualification.

Some work in this thesis is drawn from papers published alongside others. These are clearly stated and referenced where used.

This work is supported by the European Regional Development Fund - Intensive Industrial Innovation Programme, Grant No. 25R17P01847

Acknowledgements

First, I would like to thank Norbert Peyerimhoff for his excellent supervision. His curiosity and kindness kept me focused and confident and travelling this journey of discovery has been an absolute delight.

I would like to thank the team at OlexSys, in particular Horst Puschmann and Oleg Dolomanov, for the great opportunity to work alongside them and their software and endless support both on the technological side and in travel to conferences and crystallographic connections across the world. I owe a lot to Horst's open-minded enthusiasm in early days of crystallographic exploration.

I would like to thank Florian Kleemiss as one similarly expanding Olex2's code, and who was delightful to bounce ideas off during the development of NoSpherA2.

I would like to thank my secondary supervisor Anna Felikson for insight in supervisory meetings and encouragement throughout.

I would like to thank Djoko Wirosoetisno for engaging with my computational curiosity through my undergraduate degree and directing me towards this uniquely perfect opportunity.

I would like to thank my examiners David Bourne and Wilhelm Klingenberg for their insights and delight. It was wonderful to have you as my examiners.

I would like to thank the European Regional Development Fund, and Julie McLoughlin as the manager in Durham, both for my funding and continual support and connections.

Through this PhD I have been given many incredible opportunities to travel and meet others across Europe.

I wish to thank Michael Bodensteiner for the invitation to visit Regensburg as a guest early on in my PhD, allowing me to present my mathematical thoughts to a group of crystallographers and try out more of the chemical side of crystallography. It was a delightful visit and one I hope to return to in the future!

I would like to thank Simon Grabowsky and Lorraine Malaspina as the organisers of the Bremen workshop on Tools for Chemical Bonding, for creating a workshop which gave me a large basis of knowledge on chemical tools available for crystallographers.

I would like to thank the organisers of the European Crystallographic Meeting in Vienna in 2019 for their generous support of my attendance. It was wonderful to meet many crystallographers and get to explain my mathematics.

I would like to thank the European Study Group with Industry, ESGI 2021, in which we posed the twinning problem to other mathematicians and had some beautiful insights. In particular, I would like to thank Bernard Piette for organising the study group and encouraging and enabling our participation, and Cameron Hall who moderated our group and followed the twinning problem with great enthusiasm far past the study group itself. I would like to thank our other team members, in particular Christian Böhmer, Luke Corcoran, Amit Einav, James P. Harris and Michael J. Negus for great discussions and a beautiful write-up of our discoveries.

I wish to thank my mathematical peers, in particular Clare Wallace, Scott Stirling and Connor Armstrong, for support over the course of the PhD, for the little chats, comparisons and complaints.

I would like to thank all my friends in Durham and afar, for the socials and D&D to take me out of maths-brain and keep me balanced. Especially through the pandemic, the value of human connection cannot be overstated.

Finally, I thank Adam Parnaby for his continual support, for all the little compliments and reassurances that helped me keep on through to the end of this all.

If at first you don't succeed, that's one data point.

— from *xkcd 1154* by Randall Munroe

Dedicated to

My parents, Nicola & Ian Midgley,
whose eternal encouragement and
mathematical enthusiasm pushed
my curiosity ever forward.

Contents

List of Figures

List of Tables

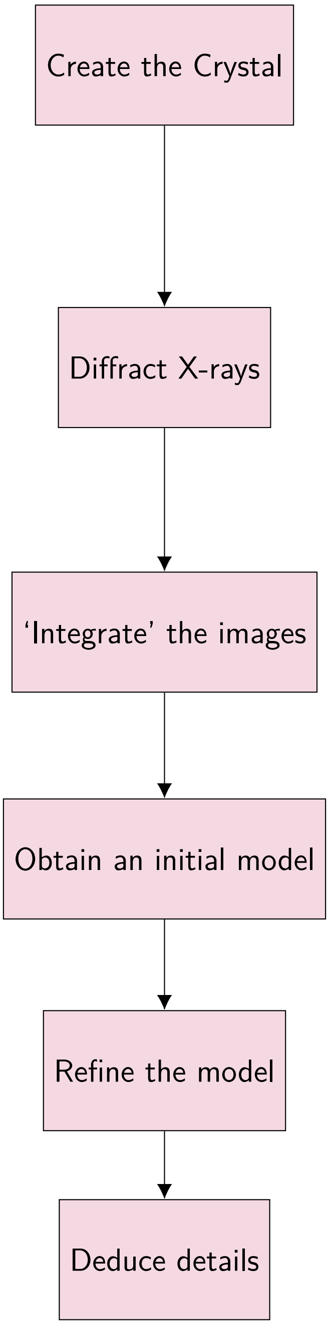
Introduction

Crystallography is a field focused on the determination of crystal structure (the layouts of atoms), typically using diffraction properties of the crystal. This requires a deep mathematical analysis of the physical processes, leading to a backwards deduction process to deduce the crystal structure which gave rise to the diffraction pattern. It is this process which has been my focus over the PhD, and in particular I have delved into two aspects of the refinement process - detection of rotated components within a single crystal (twinning), and improvement of the diffraction model based on ad hoc quantum mechanical calculations (nonspherical refinement).

The process of crystal determination itself has many more factors, outlined in Figure 1. In this thesis, I focus on the refinement aspect of the crystallographic process.

The basics of crystallography - details on atomic structures and how these cause diffraction, as well as the process by which we trace back the diffraction results to return these atomic structures - are described in Part I. This part was written with an intention to describe crystallography in a very precise mathematical way, to allow future mathematicians to understand the fundamentals quickly without the chemical details.

In Part II, we cover the theory behind the effect of twinned crystals on the diffraction pattern, and focus on the detection of these twins *after* initial refinement. We discuss the current available literature and algorithms, and introduce our own algorithms focusing on using the information available to guide our search. The conclusion for this part is in Section 10.9.



A chemist will make a crystal through crystallisation from solution, or a variety of other methods. The chemical ingredients and method of crystallisation will affect the end product, and different crystals can arise in the same experiment. They will need to carefully inspect through a microscope to select their crystal, and may have to try multiple crystals before they get one which is the intended structure.

The crystal is placed on the diffractometer, where X-rays are fired at the crystal, diffract, and are registered by a detector. Typically, the X-ray source and crystal can be rotated around the crystal centre to allow all angles of diffraction to be observed.

The collection of images at known angles are inspected for bright spots which are used to determine basic details about the crystal *lattice*, labelled, and the brightness of each spot is recorded. Integration is the term used for both this whole process and specifically the integration of the spot brightness in the images.

The details of the spot intensities is used to create an initial model of the crystal structure. This process is known as initial structure solution, and there are a number of algorithms for this.

The structure is refined such that the theoretical results of diffraction match as closely as possible with the recorded diffraction images. This is done via least-squares refinement. My work lies within this part of the crystallographic process.

A crystallographer will analyse the crystal structure and record interesting details and likely traits, or deduce reasoning behind traits the crystal is known to have.

Figure 1: The process of crystal determination

In Part III, we explore the use of ad-hoc quantum mechanical calculations for the adjustment of the *form factors* associated to each atom during the refinement process, rather than taking a pre-tabulated model of each atom independent of the structure it lies within. In particular, we explore the impact of approximating a form factor derivative, which is particularly costly to calculate, as zero, on the refinement process. The conclusion for this part is in Section 13.7.

My python files can be found at

`https://github.com/Lycanic/crystal-investigations`.

These files take the form of the *olex_twinning* file provided with Olex2 at the time of thesis submission and a *NonSpherTests.py* plugin which can be added to Olex2 to perform numerical nonspherical refinement (see Part III). It additionally contains my *ns_analyse* python script used to analyse nonspherical data and provide it in a LaTeX- compatible form.

Part I

Basics of X-ray Crystallography

Chapter 1

Fundamental Mathematical Notions

In this chapter, we introduce some convention of notation I use in this thesis as well as some basic mathematical notions used later in this thesis.

1.1 Notational Conventions

A boldface $\mathbf{x}$ denotes a column vector $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, the dimension n of which should be clear from context. $\mathbf{x}$ has a magnitude $x = \|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$, where $\mathbf{x} \cdot \mathbf{y}$ represents the *dot product*, $\sum_{i=1}^{n} x_i y_i$. Additionally, $\mathbf{x} \cdot \mathbf{y} = xy \cos\theta$ where θ is the angle between the two vectors. In three dimensions, the *cross product* $\mathbf{x} \times \mathbf{y} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}$, and is perpendicular to both $\mathbf{x}$ and $\mathbf{y}$. The zero-vector (with 0 in all entries) is denoted $\mathbf{0}$ and additionally is used to represent the origin. The *Standard Basis* is a set of vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$, where $\mathbf{e_i} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$, with the 1 in the i-th row and 0 in every other row. These satisfy $||\mathbf{e}_i|| = 1, \mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$, where δ_{ij} is the *Kronecker Delta*, equal to 1 if $i = j$ and 0 otherwise. The Kronecker Delta is also used in a more general context, for example $\delta_{x \in X}$ would be 1 if the element x were in the set X.

Matrices are represented by capital letters $A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}$, given a matrix with m rows and n columns. A particularly important matrix is the identity matrix I which has 1 along its diagonal ($a_{ii} = 1, a_{ij} = 0$ for $i \neq j$). I is always square, and its dimensions should be clear from context.

We consider subsets of matrices with specific traits, which have a group structure via matrix multiplication (details of groups will be outlined in the next section). A common one is the Orthogonal Group

$$\mathrm{O}(n) = \{A | AA^T = I, A \text{ is an } n \text{ by } n \text{ matrix}\}.$$

A key trait of such matrices is having a determinant of ±1. Such an element with determinant 1 is referred to as an *proper* rotation, with determinant -1 is an *improper* rotation. In 3 dimensions, improper rotations are also known as *rotoinversions* (and change the handedness of the system) and proper rotations simply as rotations. The linear map represented by the matrix -I is called inversion.

1.2 Group actions, orbits, equivalence relations and quotients

In this section, we recall some basic definitions related to groups.

Definition 1.2.1 (Group)**.** A group is a set G together with an operation given by $G \times G \to G$ and denoted by $(g, h) \mapsto g \cdot h$ satisfying the following properties:

(i) Existence of a neutral element $e \in G$ satisfying $e \cdot g = g \cdot e = g$ for all $g \in G$

(ii) Existence of an inverse element g^{-1} for every $g \in G$ such that $g^{-1} \cdot g = g \cdot g^{-1} = e$

(iii) Associativity: For all $g_1, g_2, g_3 \in G : (g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.

A group G is called *commutative* if for all $g_1, g_2 \in G : g_1 g_2 = g_2 g_1$.

Definition 1.2.2 (Equivalence relation and quotient)**.** Let X be a set. A *equivalence relation* on X, denoted by $x \sim y$ meaning $x \in X$ is equivalent to $y \in X$, satisfies the following three properties:

(i) Reflexivity: $x \sim x$ for all $x \in X$,

(ii) Symmetry: $x \sim y$ implies $y \sim x$ for all $x, y \in X$,

(iii) Transitivity: $x \sim y$ and $y \sim z$ implies $x \sim z$ for all $x, y, z \in X$.

Given a set X with an equivalence relation $\sim$, this equivalence relation partitions X into *equivalence classes*, denoted by $[x]$ for $x \in X$ and given by

$$[x] := \{y \in X \mid y \sim x\}.$$

Any $z \in [x]$ is called a representative of the equivalence class $[x]$. The *quotient space* $X/\sim$ is the set of all equivalence classes.

A specific example is $\mathbb{R}/\sim$ where $x, y \in \mathbb{R}$ are equivalent if and only if $x - y \in \mathbb{Z}$. This can also be written as $\mathbb{R}/\mathbb{Z}$, since both $\mathbb{R}$ and $\mathbb{Z}$ are groups with respect to addition.

Definition 1.2.3 (Group action)**.** Let G be a group and X be a general set. A *group action* of G on X is given via a map $G \times X \to X$, $(g, x) \mapsto g \cdot x$ with the following properties:

$$\begin{aligned} e\cdot x &= x \quad \text{for all } x\in X, e \text{ the identity element,}\\ g\cdot(h\cdot x) &= (gh)\cdot x \quad \text{for all } g,h\in G,\ x\in X. \end{aligned}$$

For a given $x \in X$, the *orbit* of x is given by

$$G\cdot x = \{g\cdot x \mid g\in G\},$$

and is a subset of X containing all elements which are results of any g acting on x.

For a given $x \in X$, the *stabiliser* of x is given by

$$G_x = \{g\in G | g\cdot x = x\},$$

and is a subgroup of G preserving x.

A group action of G on X gives rise to a specific equivalence relation: $x, y \in X$ are equivalent if and only if there exists $g \in G$ such that $y = g \cdot x$. In this case the quotient $X/\sim$ of orbits $G\cdot x = [x]$ can also be written as X/G.

Definition 1.2.4 (Fundamental domain)**.** Let $G\times X \to X$ be a group action on X. A *fundamental domain* of this group action is a subset $\mathcal{D}\subset X$ which contains precisely one representative of each orbit in X (which is an element of X/G).

If X has additional topological structure like $\mathbb{R}^n$, one often chooses fundamental domains to be connected and of special shape like polytopes.

Chapter 2

Mathematical Concepts Related to Crystal Structure

In this chapter, we set up the structures needed for the description of a crystal lattice - that is, how the structure repeats in space. Whilst the definition of a crystal has been updated to simply *'a material which exhibits essentially a sharp diffraction pattern'* [27] in order to accommodate quasicrystal structures discovered by Shechtman [51][1], in this thesis we consider only crystals with a regular repeating structure in 3 dimensions. Such regular structures have associated symmetry groups (point groups and space groups) which are outlined in detail. We describe the idealised form of a crystal as one which is infinite in size.

2.1 Atomic Locations and the Lattice

We define a *Crystal Structure* as a set of atoms, subject to repetition conditions.

Definition 2.1.1 (Crystal Structure)**.** Mathematically, we choose to represent a *Crystal Structure* as a discrete set $S \subset \mathbb{R}^3$ (representing atomic locations, or *sites*)

[1] Quasicrystals form an interesting story - even Shechtman was bemused by the existence of a crystal with a tenfold rotational pattern, and tried all sorts of alternate explanations before deducing that it must not follow any crystallographic lattice. He finally received the Nobel prize for this discovery in 2011.

along with a function $p : S \to \mathbb{N}$ (representing the electron/proton number of the atom). We then refer to a crystal structure by the set (S, p).

For a simple periodic crystal, we can choose three linearly independent vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$ such that:

- $\forall \mathbf{s} \in S : \mathbf{s} + \mathbf{a}, \mathbf{s} + \mathbf{b}, \mathbf{s} + \mathbf{c} \in S$ - that is, translations by these vectors map S to itself.
- $\forall \mathbf{s} \in S : p(\mathbf{s} + \mathbf{a}) = p(\mathbf{s} + \mathbf{b}) = p(\mathbf{s} + \mathbf{c}) = p(\mathbf{s})$ - that is, p is periodic with respect to each vector.

We have a similar concept to p of the *electron density function* $\rho : \mathbb{R}^3 \to \mathbb{R}_{>0}$. This also obeys periodicity in $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and is expected to have peaks at locations in S, but is instead a smooth function.

We let these vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ define a *Lattice* of points, or equivalently of translations by those points.

Definition 2.1.2 (Lattice)**.** A lattice $\Gamma = \mathbb{Z}\mathbf{a_1} + \mathbb{Z}\mathbf{a_2} + \cdots + \mathbb{Z}\mathbf{a_n} \subset \mathbb{R}^n$ is a set of *lattice points* $\mathbf{x}$ such that $\mathbf{x} = u_1\mathbf{a_1} + u_2\mathbf{a_2} + \cdots + u_n\mathbf{a_n}, u_i \in \mathbb{Z}$ and $\{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}\}$ the lattice basis. $\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}$ must be linearly independent.

A lattice can also be viewed as a set of translations which carries the structure of a commutative group via composition.

Definition 2.1.3 (Primitive Lattice)**.** Given a crystal structure (S, p), the associated primitive lattice Γ is given by

$$\Gamma := \{\mathbf{t} \in \mathbb{R}^3 | \mathbf{t} + S = S, p(\mathbf{s} + \mathbf{t}) = p(\mathbf{s}) \text{ for all } \mathbf{s} \in S\}.$$

That is, Γ contains every translation which preserves the crystal structure.

In general, for a given periodic crystal structure (S, p) we take the *Direct Lattice* [28, /Direct_lattice] defined by the three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ introduced at the beginning of this section, that is, $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$. We define additionally the angle between $\mathbf{a}$ and $\mathbf{b}$ to be γ, the angle between $\mathbf{b}$ and $\mathbf{c}$ to be α, and the angle between $\mathbf{a}$ and $\mathbf{c}$ to be β. Whilst $\mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$ must be a sublattice of the primitive lattice of the crystal structure, it might not be the entirety of the primitive lattice. Situations in which a larger lattice is chosen are clearly noted, and are frequently done to gain 'nicer' lattices - ones with 90° angles, or $\|\mathbf{a}\| = \|\mathbf{b}\|$. More on non-primitive lattice types will be discussed in subsection 2.4.

Given a periodic crystal, we can note that we need only list elements in a small finite subset of S along with their $p(s)$ to be able to deduce the entirety of S and $p(s)$ over the crystal.

Definition 2.1.4 (Unit cell of lattice)**.** A *unit cell* of a lattice $\Gamma = \sum_{i=1}^{n} \mathbb{Z}\mathbf{a}_i$ is given by $\mathcal{D} = \sum_{i=1}^{n} [0, 1)\mathbf{a}_i \subset \mathbb{R}^n$, that is, a subset of minimal volume in the shape of a parallelepiped with a vertex at $0 \in \mathbb{R}^n$ whose Γ-orbit covers all of $\mathbb{R}^n$. The volume of $\mathcal{D}$ is independent of the choice of lattice basis $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, and is called the *covolume* of Γ.

In other words, a unit cell is a specifically shaped fundamental domain of the lattice action on $\mathbb{R}^n$.

2.2 Relative and Cartesian Coordinates

It is common in crystallography to write the locations of atoms (and other locations relative to a lattice) in *Relative Coordinates*.

Definition 2.2.1 (Relative Coordinates)**.** Given a lattice Γ with basis $\mathbf{a}, \mathbf{b}, \mathbf{c}$ (such that $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$), and a location $\mathbf{x} \in \mathbb{R}^3$, we can write $\mathbf{x} = x_a\mathbf{a} + x_b\mathbf{b} + x_c\mathbf{c}$.

Then, the $\mathbf{x}^{\text{rel}}$ written in relative coordinates is $\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}$.

The typical basis for vectors in $\mathbb{R}^3$,with three perpendicular axes $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, is known as the *Cartesian Basis.* To refresh, this is defined such that $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

Vectors written in relative coordinates can be easily transformed into Cartesian coordinates. This is done through the use of an *orthogonalisation matrix.*

Take the vector $\mathbf{x}^{\text{cart}} = x_a\mathbf{a} + x_b\mathbf{b} + x_c\mathbf{c}$ in the Cartesian basis. Then $\mathbf{x}^{\text{cart}} = \begin{bmatrix} x_a a_1 + x_b b_1 + x_c c_1 \\ x_a a_2 + x_b b_2 + x_c c_2 \\ x_a a_3 + x_b b_3 + x_c c_3 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} =: A\mathbf{x}^{\text{rel}}$, where $\mathbf{x}^{\text{rel}}$ represents $\mathbf{x}^{\text{cart}}$ as written in relative coordinates, and A is the orthogonalisation matrix. A *relative* vector is with respect to the basis of its associated lattice, which should always be obvious from context.

Definition 2.2.2 (Orthogonalisation Matrix)**.** The *Orthogonalisation Matrix* $A = \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{bmatrix}$, transforms vectors $\mathbf{x}^{\text{rel}}$ written in *relative coordinates* with respect to the basis $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ of $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$ into *Cartesian coordinates* via the action $\mathbf{x}^{\text{cart}} = A\mathbf{x}^{\text{rel}}$.

By a suitable choice of Cartesian orthonormal basis (obtained via an orientation-preserving orthogonal transformation), we can always express our orthogonalisation matrix in terms of the cell parameters $a, b, c, \alpha, \beta, \gamma$ as follows (see [53, Section 3.3.1.1.1])

$$A = \begin{pmatrix} a & b\cos\gamma & c\cos\beta \\ 0 & b\sin\gamma & \frac{c(\cos\alpha - \cos\beta\cos\gamma)}{\sin\gamma} \\ 0 & 0 & \frac{V}{ab\sin\gamma} \end{pmatrix}.$$

Here V denotes the volume of the unit cell and is given by (see [53, formula (1.1.3.17)]

$$V = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = abc\sqrt{1 - \cos^2\alpha - \cos^2\beta - \cos^2\gamma + 2\cos\alpha\cos\beta\cos\gamma}.$$

We can consider the dot product of two vectors $\mathbf{x}^{\text{cart}}, \mathbf{y}^{\text{cart}}$ written in relative coordinates to the same lattice, $\mathbf{x}^{\text{rel}}$ and $\mathbf{y}^{\text{rel}}$. Their dot product would be

$$\begin{aligned}
\mathbf{x}^{\text{cart}} \cdot \mathbf{y}^{\text{cart}} &= (A\mathbf{x}^{\text{rel}}) \cdot (A\mathbf{y}^{\text{rel}}) \\
&= (A\mathbf{x}^{\text{rel}})^\top A\mathbf{y}^{\text{rel}} \\
&= \mathbf{x}^{\text{rel},\top} A^\top A\mathbf{y}^{\text{rel}} \\
&= \mathbf{x}^{\text{rel}} \cdot A^\top A\mathbf{y}^{\text{rel}}.
\end{aligned}$$

This central matrix $A^\top A$ is known as the metric matrix, M:

Definition 2.2.3 (Metric Matrix)**.** Given an orthogonalisation matrix A, the *metric matrix* is $A^\top A$. If $A = \begin{bmatrix}\mathbf{a} & \mathbf{b} & \mathbf{c}\end{bmatrix}$, then

$$M = \begin{bmatrix} \mathbf{a}\cdot\mathbf{a} & \mathbf{a}\cdot\mathbf{b} & \mathbf{a}\cdot\mathbf{c} \\ \mathbf{b}\cdot\mathbf{a} & \mathbf{b}\cdot\mathbf{b} & \mathbf{b}\cdot\mathbf{c} \\ \mathbf{c}\cdot\mathbf{a} & \mathbf{c}\cdot\mathbf{b} & \mathbf{c}\cdot\mathbf{c} \end{bmatrix}. \qquad (2.2.1)$$

The metric matrix has many uses. As seen above, it provides one simple transformation within the dot product for vectors in relative coordinates. The determinant of the metric matrix is equal to the squared volume of the fundamental domain of the lattice Γ - equivalently, the determinant of A is plus or minus the volume of the fundamental domain.

We can also consider the action of a rotation matrix in cartesian and relative space. If a rotation matrix R^{cart} takes $\mathbf{x}^{\text{cart}}$ to $\mathbf{y}^{\text{cart}} = R^{\text{cart}}\mathbf{x}^{\text{cart}}$, $\mathbf{y}^{\text{rel}} = A^{-1}R^{\text{cart}}A\mathbf{x}^{\text{rel}}$, and thus we can state that $R^{\text{rel}} = A^{-1}R^{\text{cart}}A$ (and $R^{\text{cart}} = AR^{\text{rel}}A^{-1}$). This is a map by *conjugation* with A.

2.3 Symmetries

Further to this translation repetition, many crystals also have additional symmetries composed of rotation or rotoinversion, or either combined with a simultaneous translation. Given this, we can further reduce the domain that must be described to describe the entire periodic crystal structure.

Definition 2.3.1 (Rigid Motion)**.** A *rigid motion* is a linear transformation of points within $\mathbb{R}^n$ which preserves distances between points. That is, for a rigid motion $f : \mathbb{R}^n \to \mathbb{R}^n$, $||f(\mathbf{x}) - f(\mathbf{y})|| = ||\mathbf{x} - \mathbf{y}||$. Such transformations can be written as $g(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$, where $R \in \mathrm{O}(n)$ is an operation without translation (rotation or rotoinversion), and $\mathbf{t}$ is a pure translation component.

We can further separate these matrices R in $\mathrm{O}(n)$ into those with determinant -1 or +1. The matrices of determinant +1 form the subgroup $\mathrm{SO}(n)$ known as the special orthogonal group.

The set of all rigid motions preserving a crystals structure gives rise to the notion of a *space group*:

Definition 2.3.2 (Space Group)**.** The *Space Group* G of a crystal (S, p) is the group of all symmetry operations preserving the crystal structure. That is, the space group is defined as

$$G = \{g \text{ a rigid motion} \mid \text{for every } \mathbf{s} \in S, g(\mathbf{s}) \in S \text{ and } p(g(\mathbf{s})) = p(\mathbf{s})\}.$$

The space group is often presented by providing representative elements modulo the lattice translations - that is, crystallographers will write only representative symmetries for each of the equivalence classes in G/Γ, assuming the lattice translations to be inherently present.

Given the space group, we can obtain the *Asymmetric Unit* of the crystal similarly to the unit cell, that is

Definition 2.3.3 (Asymmetric Unit of a crystal)**.** An *asymmetric unit* of a periodic crystal (S, p) with space group G is given by a fundamental domain $\mathcal{D}$ of G. The asymmetric unit will generally be a polyhedron with vertex at the origin.

There are in fact a limited number of possibilities for the Space Group of a crystal. Given two space groups G_1 and G_2 of two crystal structures (S_1, p_1) and (S_2, p_2) with associated lattices, bases, and orthogonalisation matrices, we consider G_1 and G_2 to be the same if the following holds: the corresponding groups in relative coordinates agree up to an orientation preserving operation involving permutations and reversals of axes, and combinations thereof (such an operation is represented by a matrix with a single entry ±1 in each row and column and determinant of +1).

There are 230 possible space groups [53], up to this identification, whose descriptions are well formulated and summarised later in this Section (see e.g. Table. 2.2).

Theorem 2.3.4 (The Crystallographic Restriction)**.** Given a crystal system (S, p), with a space group G, any $g \in G$ represented as $g(\mathbf{s}) = R(\mathbf{s}) + \mathbf{t}$, the matrix $R \in \mathrm{O}(3)$ will be of order $1, 2, 3, 4$, or 6.

For the proof of this, we first need to prove the following:

Proposition 2.3.5. Given a crystal system (S, p), with a space group G, any $g \in G$ represented as $g(\mathbf{s}) = R(\mathbf{s}) + \mathbf{t}$, the matrix R will preserve the primitive lattice Γ associated to (S, g).

Proof. We begin by partitioning S into subsets which are the orbits of each element in the unit cell under the lattice translations,

$$S = \bigcup_{\mathbf{s} \in \mathcal{D} \cap S} (\mathbf{s} + \Gamma)\,,$$

where $\mathcal{D}$ is the fundamental domain of Γ.

Recall that as G is the space group of S, $g(S) = S$ for all $g \in G$. Using that, and taking $g(\mathbf{s}) = R\mathbf{s} + \mathbf{t}$:

$$\begin{aligned} g(S) &= g\left(\bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (\mathbf{s}+\Gamma)\right) \\ &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s}+\Gamma)) \\ &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (R\mathbf{s}+R\Gamma+\mathbf{t}) \\ &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s})+R\Gamma)\,. \end{aligned}$$

The rest of the proof will use the following arguments: Since Γ is the set of all translations which preserve S (and p), we can conclude that if $R\Gamma$ preserves S, then it must be a subset of Γ, and if the intersections of $R\Gamma$ and Γ with every open ball $B_r(0)$ centered at the origin have the same cardinality, then $R\Gamma$ and Γ must be equal.

Take a translation $\mathbf{y} \in R\Gamma, \mathbf{y} = R\mathbf{x}$ for some $\mathbf{x} \in \Gamma$. Then, for $g \in G$,

$$\begin{aligned} S = g(S) &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s})+R\Gamma) \\ \Rightarrow g(S)+\mathbf{y} &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s})+R\Gamma+\mathbf{y}) \\ \Rightarrow g(S)+\mathbf{y} &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s})+R(\Gamma+\mathbf{x})) \\ \Rightarrow g(S)+\mathbf{y} &= \bigcup_{\mathbf{s}\in\mathcal{D}\cap S} (g(\mathbf{s})+R(\Gamma)) \\ \Rightarrow S+\mathbf{y} = S & \qquad\qquad\qquad = g(S) = S. \end{aligned}$$

Thus, any translation $\mathbf{y} \in R\Gamma$ preserves S, and thus $R\Gamma$ is a subset of Γ.

We can show that they are of the same size by restricting them to a ball of size r. That is, take $\overline{B_r(0)} = \{\mathbf{z}| \text{ s.t. } |\mathbf{z}|| \leq r\}$ as a closed ball of radius r around 0. Then we take $\Gamma_r = \Gamma \cap \overline{B_r(0)}$. As the rotation of the ball remains the ball, $(R\Gamma)_r = R\Gamma_r$, and these contain the same number of elements. Thus, $R\Gamma = \Gamma$ and R must preserve the lattice. □

Proof of Theorem 2.3.4. By Proposition 2.3.5 any rotational component R of an element $g \in G$ preserves the lattice Γ.

Initially, we show that R has integer coefficients in its characteristic polynomial. This arises naturally from the fact that $R\mathbf{y} \in \Gamma$ for any lattice point $\mathbf{y}$, and thus the rotation R, expressed as a matrix in relative coordinates, must be integer-filled because of the fact that R maps lattice vectors to lattice vectors. Such an integer-valued matrix gives rise to a characteristic polynomial with integer coefficients, and as the characteristic polynomial is unchanged under conjugation, the characteristic polynomial of R is integer valued.

All eigenvalues of such an $R \in \mathrm{O}(3)$ must be complex numbers of modulus 1 (since R preserves Cartesian length of vectors), and non-real eigenvalues come in pairs which are complex conjugates. Since the determinant of R (which is equal to the product of the eigenvalues) is ± 1, one of the eigenvalues of R must be real, and thus must be plus or minus 1. We can perform a change of bases such that the eigenvector corresponding to this value is the z-axis, and thus reduce the rotation to: $\begin{bmatrix} q_{11} & q_{12} & 0 \\ q_{21} & q_{22} & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$, where $Q \in \mathrm{O}(2)$ is its restriction to $z = 0$. We can now simply consider restrictions on such a plane.

Thus, we have $Q \in \mathrm{O}(2)$. When its determinant is positive 1, it can be represented as a rotation $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ for $\theta \in [0, 2\pi)$, with the same characteristic polynomial $\lambda^2 - 2\cos\theta\lambda + 1$. By the above claim, this must have integer coefficients, and thus $\cos\theta \in \mathbb{Z}/2$, giving rise to $\theta \in \{0, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \pi\}$, corresponding to rotations of order 1, 6, 4, 3, and 2 respectively. In the case where the determinant of Q is -1, Q represents a reflection across a *mirror line* in the restricted plane, and thus must be of order 2.

Thus, in all cases, R may only be of order $1, 2, 3, 4$, or 6.

□

The 230 space groups come under 32 *point groups*:

Definition 2.3.6 (Crystallographic Point Group)**.** A *crystallographic point group* of a lattice $\Gamma \in \mathbb{R}^3$ is the group of operations which map the lattice to itself, preserving the origin. That is, the crystallographic point group is given by

$$G_0 = \{Q \in \mathrm{O}(3) | Q\Gamma = \Gamma\}.$$

There is a limited set of symmetry transformations which satisfy this condition. The requirement to preserve the location of the origin removes the possibility of translation components. Thus, the symmetry component can be represented as $R \in \mathrm{O}(3)$, consisting of only rotoinversions and rotations.

In fact, this transformation is further restricted, as there are a very limited number of rotations which can preserve a lattice structure. As before, a rotation component must be twofold, threefold, fourfold or sixfold, and we can then introduce an inversion component. In the crystallographic community, Hermann-Mauguin notation is commonly used, representing rotations as numerals (such as 1, 2, 3, 4, 6) and rotoinversions with a 'bar' to represent the negation ($\bar{1}$,$\bar{2}$,$\bar{3}$,$\bar{4}$,$\bar{6}$) - for example, $\bar{4}$ means a fourfold rotation followed by an inversion with respect to a centre on the rotation axis. Rotations must be defined around a particular axis, whilst inversions must be defined about a point. A twofold rotation combined with an inversion through some point on the rotation axis results in a mirror plane, which the operation can be described as a reflection through. Hermann-Mauguin notation here deviates by denoting this as 'm' rather than '$\bar{2}$'.

The lattice system will first be declared (see Table 2.1), and this will give the relevant axes to declare the symmetry elements with respect to. Symmetry elements can be partitioned in relation to the axes L as follows:

Let G_0 be a crystallographic point group fixing the origin, and L a line through the origin, with a plane P perpendicular to L, passing through the origin. Consider the subgroup

$$G_L = \{R \in G_0 | RL = L, R \text{ restricted to } P \text{ is a rotation}\},$$

Then G_L is either the identity, cyclic with generators $2, 3, 4, 6, \bar{1}, \bar{2}, \bar{3}, \bar{4}$, or $\bar{6}$, or not cyclic, and generated by a rotation $2, 4$ or 6 and a reflection with respect to the plane P. Such non-cyclic groups are written $\frac{n}{\mathrm{m}}$, where n is the order of the generating rotation, and the 'm' represents the reflection generator.

These 32 point groups are divided between 7 *lattice types*, according to their essential symmetries.

The lattice type of highest symmetry is named a 'cubic' lattice, and has 3-fold symmetry about 4 axes, the diagonal cross-sections of a cube.

In Figure 2.1, we show a demonstration of this on a distorted cube, with the red lines being mapped to each other on the cube through this threefold symmetry, the axis of rotation represented by the dotted line.

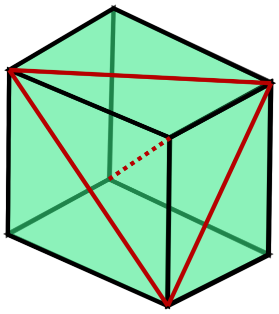


Figure 2.1: A distorted view of a cube, showing three lines which map to each other about the dotted line.

These lattice types are classified via the essential symmetry of the underlying crystal system and described in Table [57]. Each of the 230 space groups comes under one of the above 7 symmetries. Given the assignment of the essential symmetry to a particular axis, each of the 230 space groups can then be described in a unique way via their elements as linear maps with respect to the relative coordinate system given

System	Essential Symmetry	Parameter Restrictions
Cubic	3-fold about 4 axes	$\lvert\mathbf{a}\rvert = \lvert\mathbf{b}\rvert = \lvert\mathbf{c}\rvert$, $\alpha = \beta = \gamma = \frac{\pi}{2}$
Hexagonal	6-fold about **c** axis	$\lvert\mathbf{a}\rvert = \lvert\mathbf{b}\rvert$, $\gamma = \frac{2\pi}{3}$, $\alpha = \beta = \frac{\pi}{2}$
Trigonal	3-fold about **c** axis	$\lvert\mathbf{a}\rvert = \lvert\mathbf{b}\rvert$, $\gamma = \frac{2\pi}{3}$, $\alpha = \beta = \frac{\pi}{2}$
Tetragonal	4-fold about **c** axis	$\lvert\mathbf{a}\rvert = \lvert\mathbf{b}\rvert$, $\alpha = \beta = \gamma = \frac{\pi}{2}$
Orthorhombic	2-fold or mirror about 3 axes	$\alpha = \beta = \gamma = \frac{\pi}{2}$
Monoclinic	2-fold or mirror about **b** axis	$\alpha = \gamma = \frac{\pi}{2}$
Triclinic	No symmetries	None

Table 2.1: The 7 crystal systems with their essential symmetries

by the basis $\mathbf{a}, \mathbf{b}, \mathbf{c}$. The number of space groups associated to each crystal system is given in Table 2.2.

System	Number of Space Groups
Cubic	36
Hexagonal	27
Trigonal	25
Tetragonal	68
Orthorhombic	59
Monoclinic	13
Triclinic	2

Table 2.2: The numbers of space groups present for each crystal system

For example, the *Triclinic* system has no essential symmetries, and all space groups with additional symmetries would come under other systems. Given this, the two space groups under the triclinic system are 1 and $\bar{1}$ - no symmetry and inversion only.

In addition to simple rotation components, symmetry operations can also include translation components. Similarly to the rotations, they are also limited. We denote the *order* of a translational component $\mathbf{x}$ to be the lowest possible multiplier n for which $n\mathbf{x} \in \Gamma$.

A translation paired with a mirror plane is known as a *glide plane.* The translation component must be in a direction $\mathbf{x}$ which lies within the mirror plane, and if repeated twice should be a multiple of a lattice vector (it should be of order 2). If it needs to be repeated more times to reach a lattice vector, then it can equivalently be written with respect to a smaller lattice. A glide plane is written as a symbol by the letter that the translation moves in the direction of. The letters a, b, and c represent a glide plane with translation in the $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ directions respectively, the letter n represents a glide plane with translation in the diagonal direction along the plane (for example, for a plane taking x to $-x$, a letter 'n' would signify an additional shift by $\begin{bmatrix} 0 \\ 1/2 \\ 1/2 \end{bmatrix}$), and the letter 'd' represents a similar glide plane to 'n' within a *Face-centered* cell, where the translation only covers 1/4 of the cell face diagonal (such different larger cell types will be discussed later in Section 2.4).

A translation paired with a rotation is known as a *screw axis*. The translation component must be in the direction of the axis of rotation, and can only be of an order which respects the order of the rotation (for example, for a sixfold rotation, the translation must have order 2, 3, or 6). A screw axis is written as a symbol n_s, where n is the order of the rotation and s represents the fraction of the axis the translation component moves - that is, the translation component is s/n in the direction of the axis. For example, 4_1 represents a rotation by $\pi/2$ along with a translation of 1/4 in the direction of the axis.

2.4 Non-Primitive Lattices

Sometimes, it is beneficial to describe the crystal structure with a lattice which is a sub-lattice of the primitive lattice. In this case, $\mathbf{a}, \mathbf{b}, \mathbf{c}$ will differ and the unit cell will be larger, containing a primitive lattice point within it.

Such a lattice is usually chosen when it results in more preferable lattice parameters - for example, angles $\alpha, \beta, \gamma = \pi/2$, or lengths $a = b$.

Such a lattice is largely dealt with as usual, with an additional consideration to the additional translation component to any lattice points within the lattice.

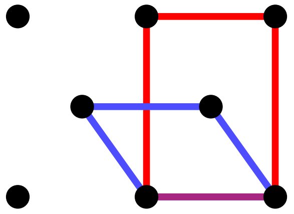


Figure 2.2: A diagram of a primitive unit cell in blue and a non-primitive preferred unit cell in red.

Chapter 3

Diffraction

In X-ray crystallography, a beam of coherent X-rays are fired into the crystal. The interactions within the crystal result in observable 'peaks' of intensity at a detector screen. The process by which these arise is outlined here.

When an atom is hit by an X-ray, its electrons scatter that X-ray in all directions. For a single atom alone, this would simply result in a dispersion of the X-ray, however when multiple atoms are involved interference comes into play.

3.1 Interference

X-rays form a wavelike structure, meaning they fluctuate through space. Our mathematical model of a wave is represented as travelling along a straight line. At any point $\mathbf{x}$ along this line, we describe the wave as $Ae^{2\pi i(|\mathbf{x}|/\lambda - t/c+\varphi)}$, where t is the time parameter, c is the speed of the wave (typically the speed of light) and λ its wavelength. $A \geq 0$ represents the (real) amplitude of the wave, whilst the *complex amplitude* is $Ae^{2\pi i\varphi}$. $\varphi \in R$ is a phase of the wave - these are generally important in relation to other waves.

Definition 3.1.1 (Phase Difference)**.** Given two waves of the same wavelength and of phase φ_1 and φ_2, their *Phase Difference* is the difference of their phases, $\varphi_1 - \varphi_2$. If waves have a phase difference in $\mathbb{Z}$ they are described as *coherent*.

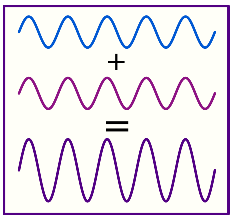
In general we will not be considering the time and space components of the wave, only the other relative variations - that is, we will likely factor out at least $e^{2\pi i(|\mathbf{x}|/\lambda - t/c)}$, resulting in $e^{2\pi i|\mathbf{x}|/\lambda - t/c}Ae^{2\pi i\varphi}$.

When waves travel in the same direction within coherence range, they interfere, creating a combined wave.

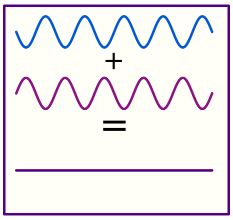
Definition 3.1.2 (Interference)**.** When two waves $A_1e^{2\pi i(\mathbf{x}/\lambda - t/c+\varphi_1)}$ and $A_2e^{2\pi i(\mathbf{x}/\lambda - t/c+\varphi_2)}$ interfere, they create a wave the sum of their parts, $A_1e^{2\pi i(\mathbf{x}/\lambda - t/c+\varphi_1)} + A_2e^{2\pi i(\mathbf{x}/\lambda - t/c+\varphi_2)} = e^{2\pi i(\mathbf{x}/\lambda - t/c)}(A_1e^{2\pi i\varphi_1} + A_2e^{2\pi i\varphi_2})$

If these two waves are of equal amplitude $A_1 = A_2 = A$, the resultant wave $= Ae^{2\pi i(\mathbf{x}/\lambda - t/c)}(e^{2\pi i\varphi_1} + e^{2\pi i\varphi_2})$

Definition 3.1.3 (Constructive interference)**.** In the special case where the phases of the waves perfectly match ($\varphi_1 - \varphi_2 \in \mathbb{Z}$), they undergo *constructive interference*, adding their intensities. The resultant wave can be written as $(A_1+A_2)e^{2\pi i(\mathbf{x}/\lambda - t/c+\varphi)}$



Definition 3.1.4 (Destructive interference)**.** In the case where the waves are of equal amplitude and the phase difference $\varphi_1 - \varphi_2 \in 1/2 + \mathbb{Z}$, they undergo *destructive interference*, cancelling one another to create a wave $= 0$.



Destructive interference can also occur with the interference of many waves, so long as the waves add to 0.

3.2 Laue Cones

The *Laue Cone* is a representation of a set of possible constructive interference around a line of atoms. Here, we will outline how these possibilities arise and are described.

The incident beam of X-rays is denoted by a vector $\mathbf{W}_{\text{in}}$. This vector has a magnitude $W_{\text{in}} = ||\mathbf{W}_{\text{in}}|| = \frac{1}{\lambda}$, proportional to its photon energy.

We begin with an infinite straight line of atoms, spaced by one vector of the lattice basis (for example, $\mathbf{a}$) - that is, we assume the atom sites are given by the vectors $k\mathbf{a}, k \in \mathbb{Z}$. We can consider that each such atom will be 'hit' by the incoming beam $\mathbf{W}_{\text{in}}$ and scatter the wave in all directions. We consider a particular scattering direction and denote it $\mathbf{W}_{\text{out}}$ (this retains the size such that $W_{\text{out}} = ||\mathbf{W}_{\text{out}}|| = \frac{1}{\lambda}$).

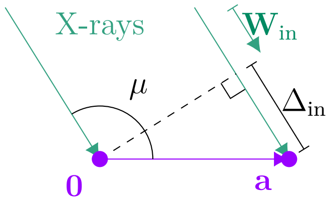


Figure 3.1: The incident beam hitting two adjacent atoms

As the X-rays come to hit these adjacent atoms, they may travel different distances before interaction. We take the angle μ from the incident ray to the direction of $\mathbf{a}$, such that

$$-\mathbf{W}_{\text{in}} \cdot \mathbf{a} = W_{\text{in}} a \cos\mu = \frac{a}{\lambda} \cos\mu. \tag{3.2.1}$$

As can be seen from Figure 3.1, the additional distance travelled by the wave hitting the second atom is $\Delta_{\text{in}} = a \sin(\mu - \pi/2) = -a \cos\mu$.

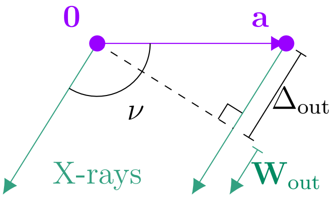


Figure 3.2: The outgoing beam from two adjacent atoms

The X-rays also travel different distances following interaction. We consider a scattering angle ν between the outgoing scattering direction $\mathbf{W}_{\text{out}}$ and the lattice direction $\mathbf{a}$. This satisfies

$$\mathbf{W}_{\text{out}} \cdot \mathbf{a} = W_{\text{out}} a \cos\nu = \frac{a}{\lambda} \cos\nu. \tag{3.2.2}$$

Additionally, the extra distance travelled by the second wave is $\Delta_{\text{out}} = a \sin(\nu - \pi/2) = -a \cos\nu$.

Following on from this, we can determine that the additional distance travelled by the second wave is equal to $\Delta_{\text{in}} + \Delta_{\text{out}} = -a(\cos\mu + \cos\nu)$. This equates to a *phase* *shift* of $\varphi = -\frac{a}{\lambda}(\cos\mu + \cos\nu)$.

We do not merely have two atoms scattering the waves though - we have a near-infinite line of them. Each subsequent atom results in an outward beam of identical intensity A and phase shift $k\varphi$ compared to the X-ray passing through the origin, where k is the number of the atom. When these interfere, they result in a wave of complex amplitude $\sum_k Ae^{2\pi i(k\varphi)} = A\sum_k e^{2\pi ik\varphi}$.

Claim 3.2.1. The only relevant values of φ are integers. For sufficiently large n of atoms along a line, constructive interference dominates for integers in comparison to non-integer φ.

Proof. Take $\varphi \in \mathbb{Z}$. Then, $e^{2\pi ik\varphi} = 1$ for all k. This results in the sum $\sum_{k=0}^{n-1} e^{2\pi ik\varphi} = n$, where n is the number of atoms along a line taken into account. Clearly, this causes a very large amplitude, as n nears infinity.

Else, we can investigate the sum $\sum_{k=0}^{n-1} e^{2\pi ik\varphi}$. Taking a line of n atoms, this becomes

$$\sum_{k=0}^{n-1} e^{2\pi ik\varphi} = \sum_{k=0}^{n-1} \left[e^{2\pi i\varphi}\right]^k$$

Without loss of generality, we take unique $\varphi \in (0,1)$ (modulo 1), and define $q = e^{2\pi i\varphi} \in \mathbb{C}, |q| = 1, q \neq 1$. Then the above is a geometric series

$$= \sum_{k=0}^{n-1} q^k = \frac{1-q^n}{1-q}$$

Considering only $\varphi \notin \mathbb{Z}$, $(q \neq 1)$, we can bound this function:

$$\begin{aligned}\left|\frac{1-q^n}{1-q}\right| &= \frac{|1-q^n|}{|1-q|} \leq \frac{|1+|q^n||}{|1-q|} \\ &= \frac{|1+|q|^n|}{|1-q|} = \frac{|1+1^n|}{|1-q|} = \frac{2}{|1-q|}.\end{aligned}$$

We want this intensity to be negligible in comparison to the intensities rising from integer valued φ. It is clear to see that this has no dependency on n for non-integer

φ. Thus, the ratio is maximally $n : \frac{2}{|1-q|}$. So long as $n >> \frac{2}{|1-q|}$, the direction will be negligible in comparison to integer valued φ.

□

Example 3.2.1 (Sizes of relevant phase shifts)**.** In this example we assume arbitrarily a crystal diameter of 0.5mm and cell lengths of 10Å, giving an approximate $n = 500000$ atoms across the crystal along a lattice direction, and present for curiosity the strength of the $n >> \frac{2}{|1-q|}$ relation.

$$\begin{aligned} n > \frac{2}{|1-q|} \Rightarrow |1-q| > \frac{2}{n} \Rightarrow (1-q)(1-\bar{q}) &> \frac{4}{n^2} \\ \Rightarrow (1-e^{2\pi i\varphi})(1-e^{-2\pi i\varphi}) &> \frac{4}{n^2} \\ \Rightarrow 2-2\cos(2\pi\varphi) &> \frac{4}{n^2} \\ \Rightarrow 2\sin^2(\pi\varphi) &> \frac{4}{n^2} \\ \Rightarrow |\sqrt{2}\sin(\pi\varphi)| &> \frac{2}{n} \\ \Rightarrow |\sin(\pi\varphi)| &> \frac{\sqrt{2}}{n}. \end{aligned}$$

Without loss of generality, we take $\varphi \in (-0.5, 0.5]$. For an n of 500000, the above inequality requires $|\varphi| > 4.5 \times 10^{-7}$. To be ten times as large, we require a $|\varphi| > 4.5 \times 10^{-6}$, a hundred times (well beyond imperceptible compared to background noise), $|\varphi| > 4.5 \times 10^{-5}$. This clearly indicates a very small area of perceivable constructive interference, effectively a point.

Given the knowledge that for significant waves, φ must be in $\mathbb{Z}$, we know that $\frac{a}{\lambda}(\cos\mu + \cos\nu) = -\mathbf{W}_{\text{in}} \cdot \mathbf{a} + \mathbf{W}_{\text{out}} \cdot \mathbf{a} = \mathbf{a} \cdot (\mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}) \in \mathbb{Z}$ for constructive waves. We define the vector $\mathbf{w} = \mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}$ to be the *scattering vector*. Given a specific entry angle μ, a *Laue Cone* is the set of $\mathbf{W}_{\text{out}}$ at the specific exit angle ν deduced from a chosen integer $n \in \mathbb{Z}$ such that $\mathbf{a} \cdot (\mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}) = n$. Differing values of n give rise to different angles and thus different cones about the line $\mathbb{R}\mathbf{a}$.

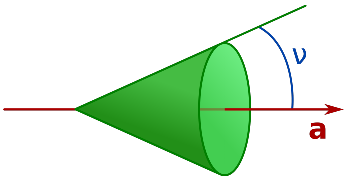


Figure 3.3: A Laue Cone of constructive interference

We take the tip of the Laue cone to be emanating from an atom site.

Definition 3.2.2 (Laue Cone)**.** Given a spacing $\mathbf{a}$ of scatterers in a line and an incident ray $\mathbf{W}_{\text{in}}$ to that line, a *Laue Cone* is the set of directions $\mathbf{W}_{\text{out}}$ where *constructive interference* arises from all the scatterers.

For constructive interference in a lattice of atoms, we require that no direction provides destructive interference. This can be covered by checking only the three lattice directions $\mathbf{a}, \mathbf{b}, \mathbf{c}$. For each lattice direction we provide an integer and an associated Laue cone. We may only have interference where all three cones intersect, with their tips placed at the same atom site. The intersection of the three (if it exists) will then be typically a ray.

The intersection of such cones can be difficult to visualize, but with cones placed with their tips at the origin, they can be considered to intersect with a unit sphere centered at the origin. This represents the cones as circles on this sphere, and their intersection ray (starting at the origin) as an intersection point on that sphere.

Figure 3.4 is a visualization of these cones on a sphere and their representative circles.

We name the integer associated with this line of atoms defined by $\mathbf{a}$ to be h, that

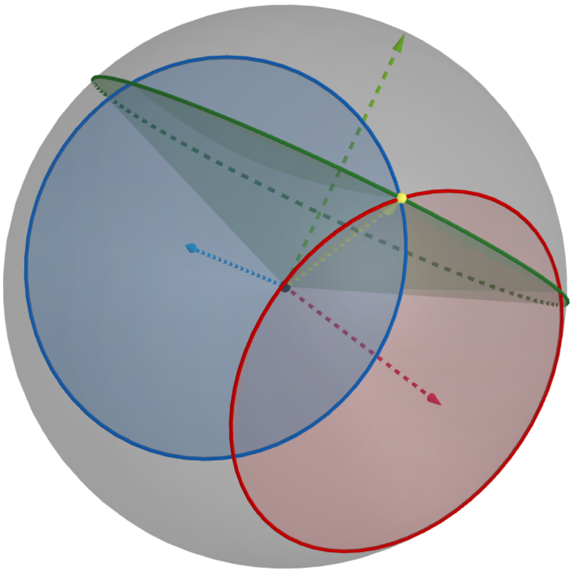


Figure 3.4: A set of 3 cones with a single intersection direction in yellow

is, $\mathbf{a} \cdot \mathbf{w} = h$. However, the lattice is defined by a 3-dimensional basis, and thus we must satisfy this condition for every direction. We can utilise the three lattice directions, naming the integers corresponding to $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ as h, k and l respectively. This means that we require:

$$\mathbf{a} \cdot \mathbf{w} = h, \tag{3.2.3}$$

$$\mathbf{b} \cdot \mathbf{w} = k, \tag{3.2.4}$$

$$\mathbf{c} \cdot \mathbf{w} = l. \tag{3.2.5}$$

It is clear that any other line's stepping vector can be realised as an integer linear sum $n_a\mathbf{a} + n_b\mathbf{b} + n_c\mathbf{c}$ of these three basis vectors , and that $(n_a\mathbf{a} + n_b\mathbf{b} + n_c\mathbf{c}) \cdot \mathbf{w} = n_a h + n_b k + n_c l$, which is also an integer fully defined by the above equations. It is therefore sufficient to ensure the above three equations are satisfied to ensure that no destructive interference has occurred.

3.3 Reciprocal Lattice

As we know $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and choose h, k, l, we then must deduce the vector $\mathbf{w}$ - if we know the direction of $\mathbf{W}_{\text{in}}$, this will provide the full information.

This task is made far easier by defining the concept of the *Reciprocal Lattice.*

Definition 3.3.1 (Dual basis and dual lattice)**.** Let $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ be a basis of $\mathbb{R}^n$. The *dual basis* is the uniquely determined set $\{\mathbf{a}_1^*, \dots, \mathbf{a}_n^*\}$ of vectors satisfying

$$\mathbf{a}_i \cdot \mathbf{a}_j^* = \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ \\ 0, & \text{if } i \neq j. \end{cases}$$

For a given lattice $\Gamma \subset \mathbb{R}^n$ the *dual lattice* is defined by

$$\Gamma^* = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\}.$$

Lemma 3.3.2. In particular, if $\Gamma = \sum_{i=1}^n \mathbb{Z}\mathbf{a}_i$ then

$$\Gamma^* = \sum_{i=1}^n \mathbb{Z}\mathbf{a}_\mathbf{i}^*,$$

where $\{\mathbf{a}_1^*, \dots, \mathbf{a}_n^*\}$ is the dual basis of $\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$. Additionally, $(\Gamma^*)^* = \Gamma$.

Proof. Take $\mathbf{x} = \sum_{i=1}^n x_i \mathbf{a}_i^* \in \sum_{i=1}^n \mathbb{Z}\mathbf{a}_i^*$ $(x_i \in \mathbb{Z})$. Then, for any $\mathbf{y} = \sum_{j=1}^n y_j \mathbf{a}_j \in \Gamma$ $(y_i \in \mathbb{Z})$,

$$\begin{aligned} \mathbf{x} \cdot \mathbf{y} &= \sum_{i=1}^n x_i \mathbf{a}_i^* \cdot \sum_{j=1}^n y_j \mathbf{a}_j \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i y_j \mathbf{a}_i^* \cdot \mathbf{a}_j \\ &= \sum_{i=1}^n x_i y_i \in \mathbb{Z} \end{aligned}$$

As $\mathbf{x} \cdot \mathbf{y} \in \mathbb{Z}$, $\sum_{i=1}^n \mathbb{Z}\mathbf{a}_i^* \subset \Gamma^*$.

Given $\mathbf{x} \in \Gamma^*$, we know that $\mathbf{x} \cdot \mathbf{y} \in \mathbb{Z}$ for all $\mathbf{y} \in \Gamma$. As each $\mathbf{a}_i \in \Gamma$, we know

$\mathbf{x} \cdot \mathbf{a}_i = x_i \in \mathbb{Z}$. Then, we can write $\mathbf{x}$ in the form $\sum_{i=1}^n x_i \mathbf{a}_i^*$, which implies that $\Gamma^* \subset \sum_{i=1}^n \mathbb{Z}\mathbf{a}_i$ (the set of all such possibilities.

Thus, we have $\sum_{i=1}^n \mathbb{Z}\mathbf{a}_i^* \subset \Gamma^*$ and $\Gamma^* \subset \sum_{i=1}^n \mathbb{Z}\mathbf{a}_i$, implying that $\Gamma^* = \sum_{i=1}^n \mathbb{Z}\mathbf{a}_i$.

$(\Gamma^*)^* = \Gamma$ follows from the fact that the dual basis of $\mathbf{a}_1^*, \ldots, \mathbf{a}_n^*$ is again $\mathbf{a}_1, \ldots, \mathbf{a_n}$.

□

Proposition 3.3.3. Let $\Gamma \subset \mathbb{R}^3$ be a lattice, and $R \in \mathrm{O}(3)$ a rotation. Then the dual lattice of the lattice

$$R\Gamma = \{R\mathbf{y} | \mathbf{y} \in \Gamma\}$$

is given by $R\Gamma^*$, that is,

$$(R\Gamma)^* = R\Gamma^*$$

Proof. The dual of the lattice $R\Gamma = \{R\mathbf{y}|\mathbf{y} \in \Gamma\}$ is the set of $\mathbf{x}$ such that $\mathbf{x} \cdot R\mathbf{y} \in \mathbb{Z}$ for all $\mathbf{y} \in \Gamma$. Thus:

$$\begin{aligned}
(R\Gamma)^* &= \{\mathbf{x}|\mathbf{x} \cdot R\mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= \{\mathbf{x}|\mathbf{x}^\top R\mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= \{\mathbf{x}|(R^\top \mathbf{x})^\top \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= \{\mathbf{x}|(R^\top \mathbf{x}) \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= \{(R^\top)^{-1}\mathbf{z}|\mathbf{z} \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \qquad \text{taking } \mathbf{z} = R^\top \mathbf{x} \\
&= \{R\mathbf{z}|\mathbf{z} \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= R\{\mathbf{z}|\mathbf{z} \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma\} \\
&= R(\Gamma^*).
\end{aligned}$$

□

For the lattice $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$, the dual lattice is $\Gamma^* = \mathbb{Z}\mathbf{a}^* + \mathbb{Z}\mathbf{b}^* + \mathbb{Z}\mathbf{c}^*$ with a basis of $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$. In fact, we can take $\mathbf{a}^* = \frac{1}{V}(\mathbf{b} \times \mathbf{c}), \mathbf{b}^* = \frac{1}{V}(\mathbf{c} \times \mathbf{a}), \mathbf{c}^* = \frac{1}{V}(\mathbf{a} \times \mathbf{b})$, with V is the volume of the unit cell, $V = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$.

Given a lattice $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$ with basis $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and its reciprocal lattice $\Gamma^* = \mathbb{Z}\mathbf{a}^* + \mathbb{Z}\mathbf{b}^* + \mathbb{Z}\mathbf{c}^*$ with basis $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$, we take A as the orthogonalisation matrix associated to the basis of Γ and A^* as the orthogonalisation matrix associated to the basis of Γ^*. Then, $(A^*)^\top$ is the inverse of A, and $A^\top$ is the inverse of A^*. We can easily show this as

$$\begin{aligned}
(A^*)^\top A &= \begin{bmatrix} (\mathbf{a}^*)^\top \\ (\mathbf{b}^*)^\top \\ (\mathbf{c}^*)^\top \end{bmatrix} \begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{a}^* \cdot \mathbf{a} & \mathbf{a}^* \cdot \mathbf{b} & \mathbf{a}^* \cdot \mathbf{c} \\ \mathbf{b}^* \cdot \mathbf{a} & \mathbf{b}^* \cdot \mathbf{b} & \mathbf{b}^* \cdot \mathbf{c} \\ \mathbf{c}^* \cdot \mathbf{a} & \mathbf{c}^* \cdot \mathbf{b} & \mathbf{c}^* \cdot \mathbf{c} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

We can use this to our advantage when taking the dot product of two vectors, $\mathbf{x}^{\mathrm{rel}}$ in relative coordinates of a lattice Γ and $\mathbf{h}^{\mathrm{rel}}$ in relative coordinates with respect to the associated reciprocal lattice Γ^*. Then:

$$\begin{aligned}
\mathbf{x}^{\mathrm{cart}} \cdot \mathbf{h}^{\mathrm{cart}} &= (A\mathbf{x}^{\mathrm{rel}}) \cdot A^*\mathbf{h}^{\mathrm{rel}} \\
&= (A\mathbf{x}^{\mathrm{rel}})^\top A^*\mathbf{h}^{\mathrm{rel}} \\
&= \mathbf{x}^{\mathrm{rel},\top} A^\top A^*\mathbf{h}^{\mathrm{rel}} \\
&= \mathbf{x}^{\mathrm{rel},\top}\mathbf{h}^{\mathrm{rel}} \\
&= \mathbf{x}^{\mathrm{rel}} \cdot \mathbf{h}^{\mathrm{rel}},
\end{aligned}$$

that is, the dot product between these two vectors can be simply calculated through the effective dot product of their relative coordinates.

The reciprocal lattice basis $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ form a separate basis of $\mathbb{R}^3$, and thus $\mathbf{w} = \mathbf{W}_{\mathrm{out}} - \mathbf{W}_{\mathrm{in}}$ can be written as a combination of them. The three Equations (3.2.3)-

(3.2.5) and the orthogonality relations between lattice vectors and reciprocal lattice vectors immediately imply $\mathbf{w} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$, a fully defined vector.

Theorem 3.3.4. Given $\lambda > 0, \mathbf{W}_{\text{in}} \in \mathbb{R}^3$ with $||\mathbf{W}_{\text{in}}|| = \frac{1}{\lambda}$ and a lattice $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$, we assume there are scatterers at all $\mathbf{z} + \Gamma$, for some $\mathbf{z} \in \mathbb{R}^3$. Then we have constructive interference in precisely those directions $\mathbf{W}_{\text{out}} \in \mathbb{R}^3, ||\mathbf{W}_{\text{out}}|| = \frac{1}{\lambda}$ for which we have

$$\begin{aligned} \mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}} &\in \Gamma^* = \mathbb{Z}\mathbf{a}^* + \mathbb{Z}\mathbf{b}^* + \mathbb{Z}\mathbf{c}^* \\ \text{and } ||\mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}|| &\leq \frac{2}{\lambda}. \end{aligned}$$

3.4 Miller Planes

Definition 3.4.1 (Primitive Lattice Point)**.** A lattice point $\mathbf{x} \neq 0$ in a lattice Γ is called *primitive* if there is no other lattice point of Γ on the straight line segment $\{\lambda\mathbf{x} | \lambda \in (0,1)\}$.

For any lattice point $\mathbf{h} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^* \in \Gamma^* \setminus \{\mathbf{0}\}$, its corresponding primitive lattice point, pointing in the same direction, is given by $\mathbf{h}_0 = \mathbf{h}/\gcd(h,k,l)$

Theorem 3.4.2. Given a lattice Γ and a vector $\mathbf{h} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^* \in \Gamma^* \setminus \{\mathbf{0}\}$, the plane P perpendicular to $\mathbf{h}$ passing through the origin contains infinitely many points of Γ, and all points in Γ are fully contained in the set of parallel planes

$$r\mathbf{d} + P, r \in \mathbb{Z}$$

where $\mathbf{d} = \frac{\mathbf{d}^*}{||\mathbf{d}^*||^2}$, where for $m = \gcd(h,k,l), \mathbf{d}^* = \frac{\mathbf{h}}{m}$ ($\mathbf{d}^*$ is the primitive lattice point in the direction of $\mathbf{h}$).

The following number theoretical lemma is crucial for the proof of the above theorem.

Lemma 3.4.3. Given a lattice $\Gamma \subset \mathbb{R}^3$ and a primitive lattice point $\mathbf{z} \in \Gamma$. Then $\mathbf{z}$ can be extended to a lattice basis, that is, there exist $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \in \Gamma$ with $\mathbf{z}_1 = \mathbf{z}$ such that

$$\Gamma = \mathbb{Z}\mathbf{z}_1 + \mathbb{Z}\mathbf{z}_2 + \mathbb{Z}\mathbf{z}_3.$$

Proof. First we consider the two-dimensional case - a vector $\mathbf{d} = h\mathbf{a} + k\mathbf{b}$ with $\gcd(h,k) = 1$ and a lattice plane $P = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b}$ to cover. We use the fact that there exist integers x, y such that $xh + ky = 1$, and deduce:

$$\begin{aligned} x\mathbf{d} &= xh\mathbf{a} + xk\mathbf{b} & \qquad y\mathbf{d} &= yh\mathbf{a} + yk\mathbf{b} \\ &= \mathbf{a} + k(x\mathbf{b} - y\mathbf{a}) & &= \mathbf{b} + h(y\mathbf{a} - x\mathbf{b}). \end{aligned}$$

We then define $\mathbf{f} = x\mathbf{b} - y\mathbf{a}$, and retrieve $\mathbf{a} = x\mathbf{d} - k\mathbf{f}$ and $\mathbf{b} = y\mathbf{d} + h\mathbf{f}$. Therefore, $\mathbf{d}$ and $\mathbf{f}$ form a basis of the lattice P.

For the three-dimensional case, we assume $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$ and we begin with a vector $\mathbf{z}_1 = h\mathbf{a} + k\mathbf{b} + l\mathbf{c}$ with $\gcd(h,k,l) = 1$.

We consider 3 cases:

Case 1: $\mathbf{z}_1 = \mathbf{a}, \mathbf{b}$ or $\mathbf{c}$

In this case, we simply take the lattice basis $\mathbf{a}, \mathbf{b}, \mathbf{c}$ as the generated lattice basis.

Case 2: One of h, k, l is 0

Let us assume, without loss of generality, $l = 0$. In this case, we take $\mathbf{z}_3 = \mathbf{c}$ and we choose $\mathbf{d} = \mathbf{z}_1$ in the two-dimensional sub-lattice $\mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b}$ primitive as in the two-dimensional case and find $\mathbf{z}_2 = \mathbf{f}$.

Case 3: None of h, k, l are 0.

Finding a 3-dimensional lattice basis is equivalent to finding a matrix with integer entries with determinant 1 extending the vector $\mathbf{z}_1$. We take

$$\begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 \end{bmatrix} = \begin{bmatrix} h & 0 & q \\ k & -s & -\frac{kp}{t} \\ l & r & -\frac{lp}{t} \end{bmatrix}$$

where we find p, q, r, s such that $ph + q\gcd(k, l) = 1$ (using the fact that $\gcd(h, k, l) = \gcd(h, \gcd(k, l)))$ and $\gcd(k, l) = t = rk + sl$. It is simple to verify that the determinant of this matrix is 1: $h(slp/t + rkp/t) + q(kr + sl) = (hp/t)(sl + rk) + qt = hp + qt = 1$.

□

Proof of Theorem 3.4.2. First, we will prove that the plane P contains infinitely many lattice points. Given $\mathbf{h} \in \Gamma^*$ and its corresponding primitive lattice point $\mathbf{d}^*$, we can extend $\mathbf{d}^*$ to a basis of Γ^*, $\mathbf{d}^*, \mathbf{e}^*, \mathbf{f}^*$ such that $\Gamma^* = \mathbb{Z}\mathbf{d}^* + \mathbb{Z}\mathbf{e}^* + \mathbb{Z}\mathbf{f}^*$, by Lemma 3.4.3. Given this, we can represent Γ as $\mathbb{Z}\mathbf{g} + \mathbb{Z}\mathbf{e} + \mathbb{Z}\mathbf{f}$, where $(\mathbf{g}, \mathbf{e}, \mathbf{f})$ is the dual basis of $(\mathbf{d}^*, \mathbf{e}^*, \mathbf{f}^*)$. Then, via the construction of the dual basis, $\mathbf{e}$ and $\mathbf{f}$ are perpendicular to $\mathbf{h}$ and thus describe the plane P perpendicular to it. Given this, the infinite set of lattice points described by $\mathbb{Z}\mathbf{e} + \mathbb{Z}\mathbf{f}$ is contained within P as required.

Now we wish to show that all points $\mathbf{x} \in \Gamma$ lie within the set of parallel planes $r\mathbf{d} + P, r \in \mathbb{Z}$.

First we take a description of the entirety of $\mathbb{R}^3$ as $r\mathbf{d} + P, r \in \mathbb{R}$, noting that $\mathbf{d}$ is perpendicular to P. Then clearly any element $\mathbf{x} \in \Gamma$ lies within $\mathbb{R}^3$, and thus $\mathbf{x} = r\mathbf{d} + P$, for some $r \in \mathbb{R}$.

Next we make use of the description of $\Gamma = \{\mathbf{x} | \mathbf{x} \cdot \mathbf{y} \in \mathbb{Z} \text{ for all } \mathbf{y} \in \Gamma^*\}$, the knowledge that $\mathbf{d}^* \in \Gamma^*$ and that for any $\mathbf{x} \in \Gamma$:

$$\begin{aligned} \mathbf{x} \cdot \mathbf{d}^* &= r\mathbf{d} \cdot \mathbf{d}^* + P \cdot \mathbf{d}^* \\ &= r\frac{\mathbf{d}^*}{||\mathbf{d}^*||^2} \cdot \mathbf{d}^* + P \cdot \mathbf{d}^* \\ &= r + P \cdot \mathbf{d}^*. \end{aligned}$$

For all $\mathbf{p} \in P, \mathbf{p} \cdot \mathbf{h} = 0$ as $\mathbf{h}$ is perpendicular to the plane. Thus, $\mathbf{x} \cdot \mathbf{d}^* = r$, and as $\mathbf{x} \cdot \mathbf{d}^* \in \mathbb{Z}, r \in \mathbb{Z}$.

□

Definition 3.4.4 (Miller Plane)**.** A *Miller Plane* is a representative of a set of parallel planes, each containing infinitely many points of a crystal lattice $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$. Such a set of parallel planes is perpendicular to a vector $ha^* + kb^* + lc^*$ of the dual lattice Γ^*, and we say that this set of parallel planes corresponds to the Miller Index (h, k, l).

Proposition 3.4.5. For a set of planes corresponding to hkl, the first plane adjacent to the plane at the origin intersects the axes at $\mathbf{a}/h, \mathbf{b}/k$ and $\mathbf{c}/l$.

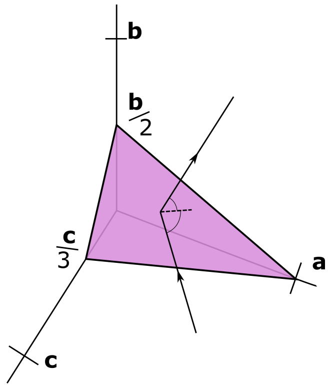


Figure 3.5: The Miller Plane (123) corresponding to the Miller Index (1,2,3)

Proof. We know that for such a set of planes, the planes are perpendicular to the vector $\mathbf{h}$. Further, the first plane adjacent to the origin is $\mathbf{d} + P$ and thus elements $\mathbf{x}$ in the plane satisfy $\mathbf{x} \cdot \mathbf{h} = 1$. Given this, by setting the components of $\mathbf{x}$ in two lattice directions to be 0 (ie finding the axis intercept), we can see that the intercepts lie at $x_a = 1/h$ for the $\mathbf{a}$-axis, $x_b = 1/k$ for the $\mathbf{b}$-axis, and $x_c = 1/l$ for the $\mathbf{c}$-axis. □

3.5 Bragg's Law of Reflection from Planes

We can view the pair of rays $\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}}$ as a reflection from a plane P.

Lemma 3.5.1. Given $\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}} \in \mathbb{R}^3$ linear independent vectors with $||\mathbf{W}_{\text{in}}|| = ||\mathbf{W}_{\text{out}}||$, there exists a unique plane $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$ though the origin such that the plane spanned by $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ is perpendicular to $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$, $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ lie on different sides of the plane, and that the angle between $\mathbf{W}_{\text{in}}$ and $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$, and the angle between $\mathbf{W}_{\text{out}}$ and $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$ are equal.

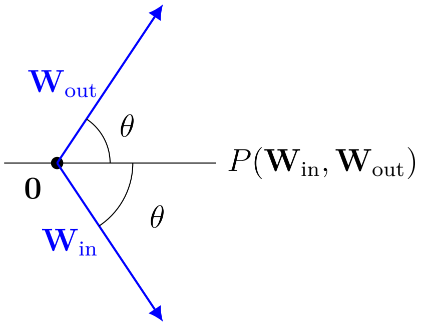

$\mathbf{W}_{\text{out}}$
θ
$P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$
$\mathbf{0}$
θ
$\mathbf{W}_{\text{in}}$

Figure 3.6: A visualisation of the vectors $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ and their associated plane $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$, which comes out of the page perpendicular to the plane of the page.

That is, the rays $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ can be viewed as directions representing a reflection from the plane $P(\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}})$, satisfying the rule that angle of incidence = angle of reflection.

Proof. Let P be a plane satisfying the two properties of the Lemma. Firstly, because it has to be perpendicular to the plane spanned by $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$, it must contain $\mathbf{W}_{\text{in}} \times \mathbf{W}_{\text{out}} \neq \mathbf{0}$. Secondly, $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ must be on both sides of the plane P with the same angle and thus $\mathbf{W}_{\text{in}} + \mathbf{W}_{\text{out}} \neq 0$ must also be contained in this plane. As $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ are linearly independent and $\mathbf{W}_{\text{in}} \times \mathbf{W}_{\text{out}}$ must be perpendicular to $\mathbf{W}_{\text{in}} + \mathbf{W}_{\text{out}}$, these are linearly independent vectors also. Therefore the unique plane in question must be the span of these two vector s$\mathbf{W}_{\text{in}} \times \mathbf{W}_{\text{out}}$ and $\mathbf{W}_{\text{in}} + \mathbf{W}_{\text{out}}$.

□

These descriptions of $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$ lead to the view of *reflections from planes.* We first consider the standard view of reflections between planes, and then show this matches to the Miller planes outlined above.

Given a set of parallel planes at a perpendicular distance d apart, and a ray of light of wavelength λ shone on them, we can consider the interference between the light reflected from parallel planes.

Given rays reflected at an angle θ from planes spaced a distance d apart, the ray travelling to the second plane will travel a further distance $2d\sin\theta$, resulting in a phase difference of $2d\sin\theta/\lambda$ wavelengths. This is demonstrated in Figure 3.7.

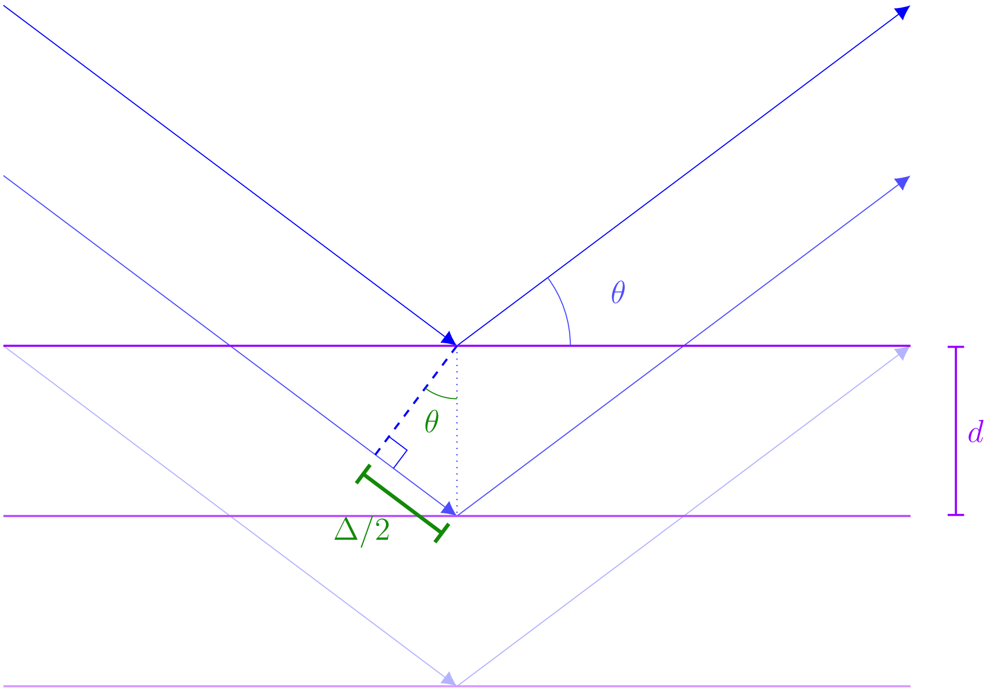


Figure 3.7: Reflection of x-rays from parallel planes

We know from our prior discussion that we want this phase difference to be an integer number of wavelengths, that is, $2d\sin\theta/\lambda = n \in \mathbb{Z}$. This is know as *Bragg's Law.* If we take a plane P as defined in Lemma 3.5.1, related to a reciprocal lattice vector $\mathbf{h}$, and the parallel planes at distance d as defined in Theorem 3.4.2, we find that these are precisely the set of parallel planes satisfying the above Bragg's Law for $n = \gcd(h, k, l)$.

As a side note, it is common in the crystallographic community to view reflections with an $n \neq 1$ as reflections from sets including 'artificial' planes at d/n spacing.

3.6 Ewald Sphere

The final way to look at these reflections, that of the Ewald Sphere, allows a simple viewing of where particular reflections come into view during the rotation of the crystal.

We take $\mathbf{w} = \mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$ given the reciprocal lattice Γ^* as usual. We know that $\mathbf{W}_{\text{out}}$ and $\mathbf{W}_{\text{in}}$ are of length $1/\lambda$, and therefore $||\mathbf{w}|| \leq ||\mathbf{W}_{\text{out}}|| + ||\mathbf{W}_{\text{in}}|| = \frac{2}{\lambda}$. Given this, we can determine that only vectors $\mathbf{w}$ with length less than or equal to $2/\lambda$ can in fact appear as scattering vectors.

This leads naturally to the concept of the *Ewald Sphere.* Assume that we have our crystal centered at the origin. Without loss of generality, we can assume that $\mathbf{W}_{\text{in}} = \frac{1}{\lambda}\mathbf{e}_1$ (fixing our coordinate system appropriately). In the diffraction process, we rotate the crystal through rotations R in O(3). The corresponding lattice under such a rotation is $R\Gamma$ and thus the dual lattice is $R\Gamma^*$ (see Proposition 3.3.3). Taking Γ^* centered at the origin, $R\Gamma^*$ is also centered at the origin. Then, the vector $\mathbf{w}$, which as we know appears as a scattering vector only when equal to a lattice point in $R\Gamma^*$, will lie upon a sphere of radius $1/\lambda$ centered at $-\mathbf{W}_{\text{in}}$. That is, for a particular rotation R, the viable scattering vectors are the intersection of this sphere and the dual lattice - $S_{1/\lambda}(-\mathbf{W}_{\text{in}}) \cap (R\Gamma^*)$, where $S_{1/\lambda}(-\mathbf{W}_{\text{in}}) = \left\{\mathbf{z} \in \mathbb{R}^3 \middle| ||\mathbf{z} + \mathbf{W}_{\text{in}}|| = \frac{1}{\lambda}\right\}$. Such an intersection point $\mathbf{z}$ is the scattering vector, and the direction in which this constructive interference appears is $\mathbf{z} + \mathbf{W}_{\text{in}} = \mathbf{W}_{\text{out}}$

Definition 3.6.1 (Ewald Sphere)**.** The *Ewald Sphere* is a sphere of radius $1/\lambda$ with the origin on the sphere (not its centre), where the incident ray runs from the centre of the sphere to the origin. A reflected direction will satisfy the diffraction condition if its corresponding reciprocal lattice point lies on the *Ewald Sphere.*

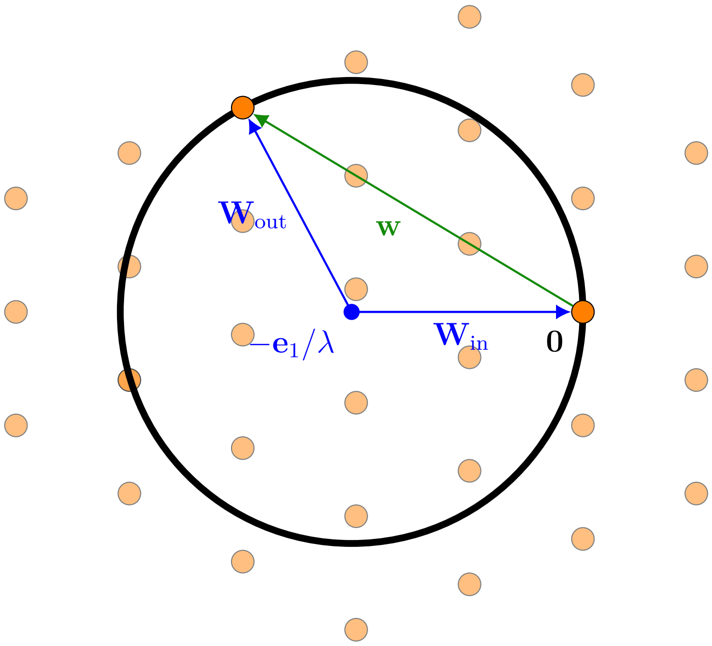


Figure 3.8: The Ewald Sphere

3.7 Indexing the h-Identifiers of Peaks

For a direction $\mathbf{h}$, its peak will show up for a rotation R if $\mathbf{h}$ lies on the Ewald Sphere $\left\{\mathbf{z} \in \mathbb{R}^3 \middle| ||\mathbf{z} + \mathbf{W}_{\text{in}}|| = \frac{1}{\lambda}\right\}$, at a direction $\mathbf{W}_{\text{out}} = \mathbf{h} + \mathbf{e}_1/\lambda$. There is an angle 2θ between $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$, and there is a direct relation between θ and the size of $\mathbf{h}$ ($||\mathbf{h}|| = 2\sin\theta/\lambda$). Taking this into account, it is then possible to take these known directions from the crystal, and known angles and wavelengths to transfer this data into reciprocal space and thus determine the reciprocal lattice and provide labels for each reflection.

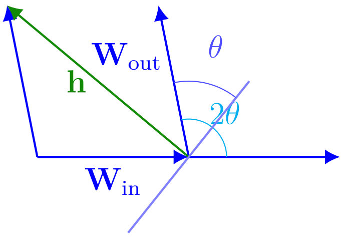


Figure 3.9: Relation between angle of reflection and incident, reflection and scattering vectors

3.8 Intensity of Peaks: the Idealised Case

We have identified directions in which constructive interference appears, but not the intensities. Assume the following idealised case: We have scatterers precisely at the positions given by $s \in S$ with scattering power $p(s)$. In an electron density approximation, these would be represented by the distribution

$$\rho(\mathbf{x}) = \sum_{\mathbf{v}\in\Gamma} \underbrace{\sum_{j=1}^{n} p(\mathbf{z}_j)\delta_{\mathbf{z}_j+\mathbf{v}}(\mathbf{x})}_{\text{scatterers in the translates unit cell}} \quad ,$$

where n is the number of sites in the unit cell of the lattice Γ, with $\delta_{\mathbf{z}_j+\mathbf{v}}(\mathbf{x})$ denotes the classical Dirac delta function.

We consider each family of atoms as scatterers of rays in the directions described by $\mathbf{h}$, with an *amplitude* proportional to the scattering power $p(s)$. We assume that atoms within the plane perpendicular to $\mathbf{h}$ which intersects the origin has a phase of 0. Then, atoms within the next plane ($\mathbf{x}$ such that $\mathbf{x} \cdot \mathbf{h} = 1$) have a phase shift of 1. Any atom will have a phase shift proportional to its distance between adjacent planes, which can be determined by the dot product $\mathbf{z}_j \cdot \mathbf{h}$.

Then, each family of atoms at positions $\mathbf{z}_j + \Gamma$ would lead to a contribution

$$p(\mathbf{z}_j)e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}},$$

to the complex amplitude in the constructive interference occurring via reflection along planes parallel to (h, k, l) (perpendicular to $\mathbf{h}$). So the observed intensity at the screen in the direction $\mathbf{h}$ would be proportional to

$$\left\| \sum_{j=1}^{n} p(\mathbf{z}_j)e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}} \right\|^2 .$$

In reality, the scatterers are not represented by this distribution at the positions s, but are represented by the electron density. A similar approach for the electron density function ρ leads to the concept of form factors and structure factors.

Chapter 4

Structure factors

The peaks detected on a screen in directions of constructive interference as above have varied intensities. We can calculate these by considering the scattering power at each point in the unit cell, alongside the phase differences due to ray distances travelled.

4.1 Derivation of Structure Factor

The following definition is inspired by that given in [27]

Definition 4.1.1 (Structure Factor $F(\mathbf{h})$)**.** The *structure factor* $F(\mathbf{h})$ is the mathematical function describing the amplitude and phase of a beam of coherent waves diffracted from a family of parallel crystal lattice planes perpendicular to the vector $\mathbf{h}$ in the direction of the angle represented by $\mathbf{h}$.

We now present a direct derivation of these structure factors. This is largely based on the derivation presented in [13].

We begin with source S and detector D. These are represented as bounded subsets of planes, with the points $\mathbf{s}$ and $\mathbf{d}$ as the respective points on each plane which are closest to the origin, $\mathbf{0}$. The source emits a stream of parallel X-rays (perpendicular to its surface), whilst the detector receives parallel X-rays (perpendicular to its surface).

We introduce scatters at locations $\mathbf{z}_j$ with scattering powers p_j.

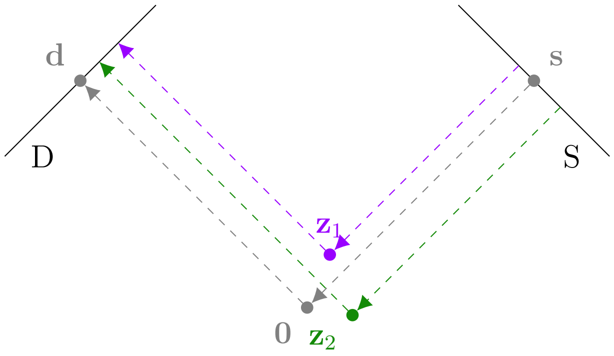


Figure 4.1: A source beam reflecting from a crystal.

The jth scatterer will receive an X-ray from S and transmit it onwards to D. This X-ray travels a distance of $(\mathbf{s} - \mathbf{z}_j) \cdot \mathbf{s}/|\mathbf{s}|$ from the source to the scatterer, and $(\mathbf{d} - \mathbf{z}_j) \cdot \mathbf{d}/|\mathbf{d}|$ from the scatterer to the detector. We set a *phase of 0* to be that of an X-ray travelling from the source to the origin to the detector, that is a phase of 0 corresponds to $|\mathbf{s}| + |\mathbf{d}|$. Any beam which travels a *shorter* distance will have been released from the source at a *later* time-emission point, and thus has a *positive* phase difference. Thus, the phase is the *negative* distance. That is, the jth scatterer has a phase difference of

$$\begin{aligned}\phi_j &= -\left[\left((\mathbf{s} - \mathbf{z}_j) \cdot \mathbf{s}/|\mathbf{s}| + (\mathbf{d} - \mathbf{z}_j) \cdot \mathbf{d}/|\mathbf{d}| - |\mathbf{s}| - |\mathbf{d}|\right)/\lambda\right] \\ &= -\left[\left(|\mathbf{s}| - \mathbf{z}_j \cdot \mathbf{s}/|\mathbf{s}| - |\mathbf{s}| + |\mathbf{d}| - \mathbf{z}_j \cdot \mathbf{d}/|\mathbf{d}| - |\mathbf{d}|\right)/\lambda\right] \\ &= -\left[-\mathbf{z}_j \cdot (\mathbf{s}/|\mathbf{s}| + \mathbf{d}/|\mathbf{d}|)/\lambda\right].\end{aligned}$$

With the nature of our previous definitions, $(\mathbf{s}/|\mathbf{s}| + \mathbf{d}/|\mathbf{d}|)/\lambda = \mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}$, and thus the phase $\phi_j = \mathbf{z}_j \cdot (\mathbf{W}_{\text{out}} - \mathbf{W}_{\text{in}}) =: \mathbf{z}_j \cdot \mathbf{w}$, where $\mathbf{w}$ is the scattering vector.

We take from each scatterer a contribution with phase ϕ_j and relative scattering power p_j, leading to a resultant complex amplitude of $\sum_j p_j e^{2\pi i \phi_j}$.

If, instead of modelling the scatterers individually, we take an electron density function $\rho(\mathbf{x})$, the sum over the scatters transforms into an integral over all space, that is

$$F(\mathbf{w}) = \int_{\mathbb{R}^3} \rho(\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\mathbf{w}} d\mathbf{x}. \tag{4.1.1}$$

In the case of a crystal, we have a finite repetition of the unit cell of the crystal. We represent the electron density associated to the atoms in the unit cell by $\rho_0(\mathbf{x})$ (see 4.1.5 later in this section for more understanding), which fades to 0 approaching infinity, and is much more dense within the unit cell. Then, the full electron density of the crystal is this repeated by translation many times. We take $\mathcal{H}$ as the set of finitely many integer triples (n_1, n_2, n_3) representing the repetitions of this unit cell through space (each beginning at $n_1 x + n_2 y + n_3 z$). We define the structure factor of the unit cell as $F_0(\mathbf{w}) = \int_{\mathbb{R}^3} \rho_0 \mathbf{x} e^{2\pi i \mathbf{x}\cdot\mathbf{w}} d\mathbf{x}$. The total electron density of the crystal $\rho(\mathbf{x}) = \sum_{\mathbf{n}\in\mathcal{H}} \rho_0(\mathbf{x} - A\mathbf{n})$, leading to a total structure factor of

$$\begin{aligned} F(\mathbf{w}) &= \int_{\mathbb{R}^3} \sum_{\mathbf{n}\in\mathcal{H}} \rho_0(\mathbf{x} - A\mathbf{n}) e^{2\pi i \mathbf{x}\cdot\mathbf{w}} d\mathbf{x} \\ &= \sum_{\mathbf{n}\in\mathcal{H}} \int_{\mathbb{R}^3} \rho_0(\mathbf{x} - A\mathbf{n}) e^{2\pi i (\mathbf{x} - A\mathbf{n})\cdot(\mathbf{w})} d\mathbf{x} e^{2\pi i (A\mathbf{n})\cdot\mathbf{w}} \\ &= F_0(\mathbf{w}) \sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i (A\mathbf{n})\cdot\mathbf{w}}. \end{aligned} \tag{4.1.2}$$

For clarity, we rearrange the dot product $(A\mathbf{n})\cdot\mathbf{w}$ to be $\mathbf{n}\cdot(A^\top\mathbf{w})$, which is equivalent to $\mathbf{n}\cdot\mathbf{w}^{\text{rel}}$, where $\mathbf{w}^{\text{rel}}$ is the vector of coefficients of the vector $\mathbf{w}$ expressed with respect to the basis $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ of the dual lattice Γ^*.

We now consider the sum $\sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i \mathbf{n}\cdot\mathbf{w}^{\text{rel}}}$. In the case $\mathbf{w}^{\text{rel}} \in \mathbb{Z}^3$, each element of this sum is 1, and the total sum is $\#\mathcal{H}$ (that is, the number of elements in $\mathcal{H}$). For the case where $\mathbf{w}^{\text{rel}} \notin \mathbb{Z}^3$, we utilise the formulae for geometric series as in Claim 3.2.1.

We first expand the sum to:

$$\sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i \mathbf{n}\cdot\mathbf{w}^{\text{rel}}} = \sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i n_1 w_1^{\text{rel}}} e^{2\pi i n_2 w_2^{\text{rel}}} e^{2\pi i n_3 w_3^{\text{rel}}}. \tag{4.1.3}$$

Then, assuming the crystal takes up a convex space, we can sum each exponent in turn. That is, for a given choice of n_1 and n_2, we take a sum over the range of n_3 permitted with this choice. We refer to the *upper* and *lower* bounds for each n_i as u_i and l_i respectively (with dependence on the prior choice of n), that is:

$$\begin{aligned}&\sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i n_1 w_1^{\text{rel}}} e^{2\pi i n_2 w_2^{\text{rel}}} e^{2\pi i n_3 w_3^{\text{rel}}} \\ &\qquad = \sum_{n_1=l_1}^{u_1} \left(e^{2\pi i n_1 w_1^{\text{rel}}} \sum_{n_2=l_2(n_1)}^{u_2(n_1)} \left(e^{2\pi i n_2 w_2^{\text{rel}}} \sum_{n_3=l_3(n_1,n_2)}^{u_3(n_1,n_2)} \left(e^{2\pi i n_3 w_3^{\text{rel}}} \right)\right)\right).\end{aligned}$$

We take the most inner sum,

$$\begin{aligned}\sum_{n_3=l_3(n_1,n_2)}^{u_3(n_1,n_2)} \left(e^{2\pi i n_3 w_3^{\text{rel}}} \right) &= \sum_{n_3=l_3(n_1,n_2)}^{u_3(n_1,n_2)} \left(e^{2\pi i w_3^{\text{rel}}} \right)^{n_3} \\ &= e^{2\pi i w_3^{\text{rel}} l_3} \sum_{k_3=0}^{u_3(n_1,n_2)-l_3(n_1,n_2)} \left(e^{2\pi i w_3^{\text{rel}}} \right)^{k_3} \\ &= e^{2\pi i w_3^{\text{rel}} l_3} \frac{1-e^{2\pi i w_3^{\text{rel}}(u_3(n_1,n_2)-l_3(n_1,n_2)+1)}}{1-e^{2\pi i w_3^{\text{rel}}}}.\end{aligned}$$

As in Claim 3.2.1, this is in norm less than or equal to

$$\frac{2}{\left|1-e^{2\pi i w_3^{\text{rel}}}\right|},$$

for every choice of l_3 and u_3.

We can repeat this with the other two sums to obtain[1]:

$$\left|\sum_{\mathbf{n}\in\mathcal{H}} e^{2\pi i \mathbf{n}\cdot\mathbf{w}^{\text{rel}}}\right| \leq \frac{8}{\left|1-e^{2\pi i w_1^{\text{rel}}}\right|\left|1-e^{2\pi i w_2^{\text{rel}}}\right|\left|1-e^{2\pi i w_3^{\text{rel}}}\right|}.$$

We then compare this to the sum obtained for $\mathbf{w}^{\text{rel}} \in \mathbb{Z}^3$, $\#\mathcal{H}$, such that the relative structure factor for such an off-peak $\mathbf{w}^{\text{rel}}$ is $\frac{1}{\#\mathcal{H}}\frac{8}{\left|1-e^{2\pi i w_1^{\text{rel}}}\right|\left|1-e^{2\pi i w_2^{\text{rel}}}\right|\left|1-e^{2\pi i w_3^{\text{rel}}}\right|}$.

[1] if any value of $\mathbf{w}^{\text{rel}}$ is integer-valued, its corresponding sum will equate to the total n_i in the line, rather than the fraction shown. This can be bounded above by a greater limit of the largest $u_i - l_i$, and thus points with some (but not all) integer values may get closer to $\#\mathcal{H}$, but will still be significantly smaller than $\#\mathcal{H}$.

As before, so long as $\#\mathcal{H} >> \frac{8}{\left|1-e^{2\pi i w_1^{\text{rel}}}\right|\left|1-e^{2\pi i w_2^{\text{rel}}}\right|\left|1-e^{2\pi i w_3^{\text{rel}}}\right|}$, the direction represented by $\mathbf{w}^{\text{rel}}$ will be negligible in strength.

We then simply note that integer $\mathbf{w}^{\text{rel}}$ corresponds to $\mathbf{w} \in \Gamma^*$. From now on, we return to using $\mathbf{h} \in \Gamma^*$ for these most visible points with respect to the structure factor.

Given this, we can reduce our calculations of $F(\mathbf{h})$ to be simply $F_0(\mathbf{h})$ for any $\mathbf{h} \in \Gamma^*$, with the constant factor $\#\mathcal{H}$ removed. That is:

$$F(\mathbf{h}) = \int_{\mathbb{R}^3} \rho_0(\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\mathbf{h}} d\mathbf{x}. \tag{4.1.4}$$

In structure determination, the number of unit cells is irrelevant as the unit cell repeats periodically. Additionally, there are additional multiplicative factors unknown to us. Therefore, reducing this to a single unit cell does not impact the determination of the structure factors, and we allow the multiplicative factor to be determined within the structure refinement.

Equation (4.1.4) is mathematically a Fourier transform of the electron density, which we will discuss in more details in Section 4.2.

We partition the electron density into associated densities belonging to each atom, assuming each contributes to the overall electron density via its own atomic electron density ρ_j.

We take

$$\rho_0(\mathbf{x}) = \sum_{j=1}^{n} \rho_j(\mathbf{x} - \mathbf{z}_j) \tag{4.1.5}$$

where n is the number of atoms within the unit cell, ρ_j is the electron density assigned to an atom as if its centre, $\mathbf{z}_j$, were at the origin. Then:

$$F(\mathbf{h}) = \int_{\mathbb{R}^3} \sum_{j=1}^{n} \rho_j(\mathbf{x} - \mathbf{z}_j) e^{2\pi i \mathbf{x}\cdot\mathbf{h}} d\mathbf{x}$$

$$
\begin{aligned}
&= \sum_{j=1}^{n} \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i(\mathbf{y}+\mathbf{z}_j)\cdot\mathbf{h}} d\mathbf{y} && \text{via } \mathbf{y} = \mathbf{x} - \mathbf{z}_j \\
&= \sum_{j=1}^{n} e^{2\pi i \mathbf{z}_j\cdot\mathbf{h}} \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i \mathbf{y}\cdot\mathbf{h}} d\mathbf{y}
\end{aligned}
$$

We can then define the *form factor* of the jth atom $f_j(\mathbf{h})$ (again, this is a Fourier transform of the atomic electron density.):

Definition 4.1.2 ((Atomic) Form Factor)**.** The form factor,

$$f_j(\mathbf{h}) = \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i \mathbf{y}\cdot\mathbf{h}} d\mathbf{y}, \tag{4.1.6}$$

is a measure of the scattering power of an individual atom.

Given this, we can write these *structure factors* as:

$$F(\mathbf{h}) = \sum_{j=1}^{n} e^{2\pi i \mathbf{z}_j\cdot\mathbf{h}} f_j(\mathbf{h}). \tag{4.1.7}$$

These form factors have historically been approximated by an assumption of spherically symmetric electron density around an atom, known as the *Independent Atom Model* (IAM). This leads to a spherically symmetric function for $f_j(\mathbf{h})$, which is approximated by a sum of Gaussian functions:

$$f_j(\mathbf{h}) = \sum_{i=1}^{4} a_i e^{-b_i\left(\frac{|\mathbf{h}|}{4\pi}\right)^2} + c,$$

where specific values for the a_i, b_i, c can be found at [58] and [6, Table 6.1.1.4]. These have additionally been verified by experiment, and are considered valid for $||\mathbf{h}||$ below 25Å^{-1}.

4.2 Fourier Theory

To the experienced eye, the Equation (4.1.1) of the structure factor can be identified as a Fourier Transform[2] (we take details from [55] and [36]).

For simplicity, we assume all our functions in this section to be smooth and we restrict to the 3-dimensional case.

Definition 4.2.1 (Fourier Transform)**.** The *Fourier Transform* of a function $f \in L^1(\mathbb{R}^3) \cap L^2(\mathbb{R}^3)$ is $\mathcal{F}(f)(\xi) = \int_{\mathbb{R}^3} f(\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\xi} d\mathbf{x}$.

Recall the well-known *Fourier Inversion Formula* [36, page 132], $\mathcal{F}^{-1}(\mathcal{F}(f)) = f$, where

$$\mathcal{F}^{-1}(g)(\mathbf{x}) = \int_{\mathbb{R}^3} g(\xi) e^{-2\pi i \xi\cdot\mathbf{x}} d\xi$$

This $\mathcal{F}(f)$ is a function of the variable ξ in *Fourier Space* (a separate copy of $\mathbb{R}^3$) and also obeys the above requirements on the function f, that is $\mathcal{F}(f)$ is square integrable and smooth.

Definition 4.2.2 (Fourier Coefficient)**.** The *Fourier Coefficients* are defined on (smooth) functions f periodic with respect to a lattice Γ such that $f(\mathbf{x}+\mathbf{t}) = f(\mathbf{x})$ for $\mathbf{t} \in \Gamma$ and $\mathbf{x} \in \mathbb{R}^3$.

The *Fourier Coefficient* of f for $\mathbf{h} \in \Gamma^*$ is defined as

$$\hat{f}(\mathbf{h}) = \frac{1}{\mathrm{Vol}(\mathcal{D})} \int_{\mathcal{D}} f(\mathbf{x}) e^{2\pi i \mathbf{h}\cdot\mathbf{x}} d\mathbf{x}$$

(with $\mathcal{D}$ the unit cell of the lattice Γ).

With these Fourier Coefficients, the function f can be recovered by the following

[2] Typically in mathematical literature, the Fourier Transform and Fourier Inverse Transform are the reverse of these two (that is, the Fourier Transform contains a negative exponent and the inverse a positive exponent), but taking positive for the forward transform is typical in crystallography, and this remains mathematically consistent.

Fourier Series:

$$f(\mathbf{x}) = \sum_{\mathbf{h}\in\Gamma^*} \hat{f}(\mathbf{h})e^{-2\pi i\mathbf{h}\cdot\mathbf{x}}.$$

In the next section we will require the useful property in Theorem 4.2.4 below of the Fourier Transform. For this, we need an operation called convolution defined on absolutely integrable functions, that is $\int |f| < \infty$. As before, to simplify matters, we restrict to smooth functions.

Definition 4.2.3 (Convolution)**.** A convolution $*$ on two absolutely integrable functions f and g is:

$$\left(f * g\right)(\mathbf{x}) = \int_{\mathbb{R}^3} f(\mathbf{x}-\mathbf{y})g(\mathbf{y})d\mathbf{y}. \tag{4.2.1}$$

This can be seen as a *smearing* of f, using g as weighting (it is a symmetric function, so can also be seen as a smearing of g with f as weighting).

The result we require in the next section is the following fact (see pg 101 of [36]):

Theorem 4.2.4. Let f, g be two absolutely integrable functions. Then $f * g$ is again absolutely integrable and we have:

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g).$$

4.3 Anisotropic Displacement Parameters

The model of a structure is enhanced by the addition of extra modelling factors and parameters. A very common such enhancement is that of Anisotropic Displacement Parameters, or ADPs.

ADPs were initially described as 'Anisotropic Thermal Parameters', and intended to account for the thermal motion of atoms. However, they account for more than the thermal motion and were thus renamed to the more general *Displacement* parameters.

These parameters provide a smearing factor to the atomic electron densities, which represent things such as the atom moving or vibrating in space, the lengthening and contracting of bonds, and even electron density within bonds themselves.

We implement this smearing as a convolution on the electron densities with a Gaussian function $g_j(\mathbf{x})$ for each atom, that is

$$(\rho_j * g_j)(\mathbf{x}).$$

As seen above, $\mathcal{F}(\rho_j * g_j) = \mathcal{F}(\rho_j)\mathcal{F}(g_j)$. The Fourier Transform of a Gaussian is again a Gaussian, and we write

$$\mathcal{F}(g_j) = G_j := e^{-2\pi^2 \mathbf{h}^\top U_j \mathbf{h}}, \tag{4.3.1}$$

thus,

$$F(\mathbf{h}) = \sum_{j=1}^{n} f_j(\mathbf{h}) e^{2\pi i \mathbf{z}_j \cdot \mathbf{h}} e^{-2\pi^2 \mathbf{h}^\top U_j \mathbf{h}}. \tag{4.3.2}$$

U_j is a 3 by 3 symmetric matrix. These ADPs may be written as a Cartesian (U^{cart}) or relative (U^*) format [20], much like the Miller indices and positions. These are related simply by:

$$U^{\text{cart}} = AU^*A^\top.$$

where A is the standard orthogonalisation matrix. It is easy to show that $\mathbf{h}^{\text{rel}} U^* \mathbf{h}^{\text{rel},\top} = \mathbf{h}^{\text{cart}} U^{\text{cart}} \mathbf{h}^{\text{cart}}$.

Additionally, ADP parameters will be displayed as U^{cif} format in crystallographic programs. This is given by

$$U^{\text{cif}} = \begin{bmatrix} a^* & 0 & 0 \\ 0 & b^* & 0 \\ 0 & 0 & c^* \end{bmatrix} U^* \begin{bmatrix} a^* & 0 & 0 \\ 0 & b^* & 0 \\ 0 & 0 & c^* \end{bmatrix},$$

and converts the values into a more understandable, Å^2-scale values whilst preserving the $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ directions as refined.

Finally, they may also be represented as the Debye-Waller factor β, which simply includes the $2\pi^2$ factor - that is, $\beta = 2\pi^2 U^*$.

Chapter 5

Initial Structure Determination

Let us recall from Section 4 Equation (4.1.4) that

$$F(\mathbf{h}) = \int_{\mathbb{R}^3} \rho_0(\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\mathbf{h}} d\mathbf{x}$$

which can be re-written as

$$F(\mathbf{h}) = \int_{\mathcal{D}} \rho(\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\mathbf{h}} d\mathbf{x}, \tag{5.0.1}$$

where we take $\rho(\mathbf{x}) = \sum_{\mathbf{t}\in\Gamma} \rho_0(\mathbf{x} - \mathbf{t})$ with the idealisation of an infinite lattice. It stands to reason that the electron density could be retrieved by the inverse equation, that is:

$$\rho(\mathbf{x}) = \sum_{\mathbf{h}\in\Gamma^*} F(\mathbf{h}) e^{-2\pi i \mathbf{h}\cdot\mathbf{x}}$$

The theoretically derived structure factor $F(\mathbf{h})$ has a direct relation to the observed reflection intensities $I_o(\mathbf{h}) = |F_{\text{obs}}(\mathbf{h})|^2$. In fact, they are considered the *theoretical* and *physical* estimations of the same quality[1]. If we were able to know the physically reflected structure factor $F_{\text{obs}}(\mathbf{h})$ precisely for all Miller indices $\mathbf{h}$, we would be able

[1] We focus around *calculated* and *observed* values of the same quantity, with the calculated values for a specific model being represented with the subscript c or calc, and the observed values from a physical crystal labelled with the subscript o or obs. In discussion, we tend to write the full 'obs' or 'calc' but we shorten this to o and c within more complicated equations. When these quantities are presented without a subscript (eg $F(\mathbf{h})$), we are discussing the theory without a specific crystal or model.

to use the inverse transform above to find the electron density. However, the detected values on the screen do not provide a complex number - the information we have for $F_{\text{obs}}(\mathbf{h})$ contains only its amplitude and no phase information. Thus, we cannot utilise this Fourier connection completely from the experiment - this fact is known as the *phase problem* of crystallography. Additionally, within the Fourier Coefficient it is assumed that $\mathbf{h}$ is taken over the infinite lattice Γ^*, whilst we are physically limited to $||\mathbf{h}|| < \frac{2}{\lambda}$ in size (where λ is the wavelength of the incoming X-ray beam).

However, there is clear information in the directions and intensities that can be used to deduce crystallographic information.

5.1 Space Group and Lattice Determination

Firstly, the indexing process gives information on the reciprocal lattice basis $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$, which can be directly transformed into the crystal lattice basis $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Additionally, noted absences in the diffraction pattern can imply symmetries of the crystal.

Definition 5.1.1 (Systematic Absence)**.** A *systematic absence* of a diffraction pattern is a set of reflections $\mathbf{h}$ with intensity 0 which can be explicitly expressed by certain relations.

For example, a diffraction pattern could have a systematic absence in the $h = 0, k = 0$ line when l is odd.

Example 5.1.1. We will analyse the appearance of the diffraction pattern of a simple crystal with a single non-trivial symmetry element in its space group.

Within these examples, for ease of understanding, all locations and reciprocal lattice points are written in relative coordinates (that is, $\mathbf{s} = \begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix}; s_a, s_b, s_c \in [0, 1)$

represents an atom at Cartesian location $s_a\mathbf{a} + s_b\mathbf{b} + s_c\mathbf{c}$, and $\mathbf{h} = \begin{bmatrix} h \\ k \\ l \end{bmatrix}; h, k, l \in \mathbb{Z}$ represents a reciprocal lattice point with Cartesian representation $h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$. All elements within these examples can be equivalently fully written out within the Cartesian coordinate system.

The general situation:

We take a crystal system with a glide plane, with the reflection plane perpendicular to the **b**-direction and a shift in the **c**-direction (this space group is known as Pc, or P1c1). This symmetry can be represented by a rigid motion $g(\mathbf{s}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{s} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2} \end{bmatrix} =: R\mathbf{s} + \mathbf{t}$.

We partition our set of n atomic sites, S, into two subsets S_1 and S_2 such that $g(S_1) = S_2, g(S_2) = S_1, S_1 \cap S_2 = \varnothing, S_1 \cup S_2 = S$, and we label the elements of S_1 lying within the unit cell from 1 to $\frac{n}{2}$ and the elements of S_2 lying within the unit cell from $\frac{n}{2} + 1$ to n such that for $\mathbf{s}_j \in S_1$, $g(\mathbf{s}_j) = \mathbf{s}_{j+\frac{n}{2}}$. Then additionally, the form factors for these atoms are related by $f_{j+\frac{n}{2}}(\mathbf{h}) = f_j(R\mathbf{h})$. Also, note that $\mathbf{s} \cdot R\mathbf{h} = \mathbf{s}^T R\mathbf{h} = (R^T\mathbf{s})^T\mathbf{h} = R^T\mathbf{s} \cdot \mathbf{h}$, and that our particular choice of R guarantees $R^T = R, R^T R = R^2 = I$.

Then we take the equation for the structure factor, (4.1.7), and partition it according to these subsets:

$$\begin{aligned}
F(\mathbf{h}) &= \sum_{j=1}^{n} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + \sum_{j=\frac{n}{2}+1}^{n} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_{j+\frac{n}{2}} \cdot \mathbf{h}} f_{j+\frac{n}{2}}(\mathbf{h})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + e^{2\pi i g(\mathbf{s}_j)\cdot \mathbf{h}} f_j(R\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + e^{2\pi i \mathbf{t}\cdot \mathbf{h}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + e^{2\pi i \frac{l}{2}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}).
\end{aligned}
$$

a) Now we search for *Systematic Absences*, $\mathbf{h}$ where this structure will have no intensity. If we take $\mathbf{h}$ with $k = 0$, $R\mathbf{h} = \mathbf{h}$ and we can factor out the form factors:

$$
\begin{aligned}
&= \sum_{j=1}^{\frac{n}{2}} f_j(\mathbf{h})(e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} + e^{2\pi i \frac{l}{2}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}}) \\
&= \sum_{j=1}^{\frac{n}{2}} f_j(\mathbf{h})(e^{2\pi i (s_a h + s_c l)} + e^{2\pi i \frac{l}{2}} e^{2\pi i (s_a h + s_c l)}) \\
&= \sum_{j=1}^{\frac{n}{2}} f_j(\mathbf{h}) e^{2\pi i (s_a h + s_c l)} (1 + e^{2\pi i \frac{l}{2}})
\end{aligned}
$$

If we also take l odd, then the final bracket

$$
\begin{aligned}
1 + e^{2\pi i \frac{l}{2}} &= 1 + e^{\pi i l} \\
&= 1 - 1 \\
&= 0,
\end{aligned}
$$

and thus, for $\mathbf{h}$ with $k = 0, l$ odd (h may take any value), there will be no intensity. In turn, observed intensity patterns with these systematic absences imply the existence of this glide plane (or similar symmetries) within the crystal.

b) Now we wish to show that the diffraction pattern has intensities invariant under R.

To do this, we look at $F(R\mathbf{h})$

$$\begin{aligned}
F(R\mathbf{h}) &= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot R\mathbf{h}} f_j(R\mathbf{h}) + e^{2\pi i \mathbf{t} \cdot R\mathbf{h}} e^{2\pi i R\mathbf{s}_j \cdot R\mathbf{h}} f_j(R^2\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}) + e^{2\pi i R\mathbf{t} \cdot \mathbf{h}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) \\
&= \sum_{j=1}^{\frac{n}{2}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}) + e^{2\pi i \mathbf{t} \cdot \mathbf{h}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) \\
&= e^{2\pi i \mathbf{t} \cdot \mathbf{h}} \sum_{j=1}^{\frac{n}{2}} e^{-2\pi i \mathbf{t} \cdot \mathbf{h}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}) + e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}).
\end{aligned}$$

Noting that as $\mathbf{t} \cdot \mathbf{h} \in \mathbb{Z}/2$, $e^{-2\pi i \mathbf{t} \cdot \mathbf{h}} = e^{2\pi i \mathbf{t} \cdot \mathbf{h}}$,

$$F(R\mathbf{h}) = e^{2\pi i \mathbf{t} \cdot \mathbf{h}} \sum_{j=1}^{\frac{n}{2}} e^{2\pi i \mathbf{s}_j \cdot \mathbf{h}} f_j(\mathbf{h}) + e^{2\pi i \mathbf{t} \cdot \mathbf{h}} e^{2\pi i R\mathbf{s}_j \cdot \mathbf{h}} f_j(R\mathbf{h}) = e^{2\pi i \mathbf{t} \cdot \mathbf{h}} F(\mathbf{h}).$$

This is then a multiplication of $F(\mathbf{h})$ by $e^{2\pi i \mathbf{t} \cdot \mathbf{h}}$, and thus $F(R\mathbf{h})$ has the same intensity as $F(\mathbf{h})$.

It is the case that multiple different symmetries can give rise to the same systematic absences, but these have been recorded and all possibilities can be noted for the observed absences, which can be determined through further analysis.

5.2 Initial Atom Placement

Beyond this determination of Γ, and implication of suitable space groups G, the full determination of an initial model of the crystal is typically much more challenging.

One method for identifying an initial model is the *Patterson Function*, a function which is the Fourier Transform of the electron density $\rho_0(\mathbf{x})$ convolved with $\rho_0(-\mathbf{x})$, leading to

$$P(\mathbf{u}) = \frac{1}{V}\sum_{\mathbf{h}} |F(\mathbf{h})|^2 \cos(2\pi\mathbf{h}\cdot\mathbf{u}). \tag{5.2.1}$$

This provides a map of the differences of atom sites within the structure, with intensities proportional to those atoms' density (or atomic number). One would then map out all of these peaks, divide by the intensity of the lowest such peak (which may correlate to a hydrogen-hydrogen distance), and use the strengths of the other peaks to assign atoms and their locations in relation to one another.

The method of *charge flipping* [44] iteratively assigns likely phases to the observed structure factors and uses Fourier inversion to generate an electron density map. Any negative electron density results are either 'flipped' to positive (giving the name) or set to 0, and this is followed by Fourier transforms to generate new likely phases. This iteration continues until reasonably converged, or it is clear that convergence is unlikely. Typically, charge flipping will begin with hundreds or thousands of potential electron density functions which go through this process simultaneously and the best fitting one will be chosen. Due to this, this is a very recent method, as it required much computational power to run these many simulations.

Chapter 6

Refinement

Given an initial estimation for the crystal structure (S, p) from some *initial structure solution* method, we then perform in a second stage *refinement* on the structure to shift it into a structure which best fits the data collected. However, for this we need an idea of how to evaluate what a *best fit* looks like.

Definition 6.0.1 (R-factor)**.** The *R-factor* is a quantity used to express how closely a model's theoretical structure factors fit to the observed data.

There are two popular equations to measure this - colloquially known as 'R1' and 'wR$_2$'.

For R1, the standard quoted R-factor,

$$R = \frac{\sum_{\mathbf{h}} ||F_{\text{calc}}(\mathbf{h})| - |F_{\text{obs}}(\mathbf{h})||}{\sum_{\mathbf{h}} F_{\text{obs}}(\mathbf{h})}, \tag{6.0.1}$$

where the sums are taken over those Miller indices $\mathbf{h} \in \Gamma^*$ which have recorded observed intensities, $F_{\text{calc}}(\mathbf{h})$ is the calculated structure factor and $F_{\text{obs}}(\mathbf{h})$ the observed structure factor. This is the most direct linear evaluation of the closeness of F_{calc} and F_{obs}

The factor wR$_2$ is typically used as the target of refinement, and is as follows:

$$\text{wR}_2 = \sqrt{\frac{\sum_{\mathbf{h}} w(\mathbf{h})(Y_o(\mathbf{h}) - \tilde{K}Y_c(\mathbf{h}))^2}{\sum_{\mathbf{h}} w(\mathbf{h})Y_o(\mathbf{h})^2}}. \tag{6.0.2}$$

Here, $Y_o(\mathbf{h})$ are the observed intensities $|F_{\text{obs}}|^2$, $Y_c(\mathbf{h})$ the calculated intensities $|F_{\text{calc}}|^2$, and $w(\mathbf{h})$ the weight associated to the Miller triple $\mathbf{h}$ (typically related to the confidence in its value). Here we also have the scaling factor $\tilde{K}$, an analytically calculated constant multiplier across all Y_c which scales Y_c to minimise the difference vector $Y_o - \tilde{K}Y_c(\mathbf{h})$. In future, when we are comparing different structures, we will introduce an additional variable $\mathbf{x}$ representing all the parameters of the model, which $\tilde{K}$ and Y_c are dependent on.

The wR$_2$-factor can also be written as a quotient of *weighted norms*

$$\text{wR}_2 = \frac{||\tilde{r}||_w}{||Y_o||_w}$$

of the residual vector $\tilde{r} = Y_o(\mathbf{h}) - \tilde{K}Y_c(\mathbf{h})$ and the observed intensities Y_o.

Here, the weighted norm of a vector $\mathbf{v}$ is given by

$$||\mathbf{v}||_w = \sqrt{\sum_h (w(\mathbf{h})v(\mathbf{h})^2)}. \tag{6.0.3}$$

That is, these R-factors evaluate the difference between the theoretical and observed structure factors or intensities relative to the strength of the observed structure factors or intensities - it provides a *percentage inaccuracy.*

The aim with the refinement process is to minimise the wR$_2$-factor. The process by which we do this is known as *Least Squares Minimisation.* In this section we will discuss the Gauss-Newton method, whilst the Levenberg-Marquadt method utilises gradient descent alongside Gauss-Newton to speed up the convergence far from the minimum and move to more careful refinement steps close to the minimum. For more details on the intricacies of these choices, see [17].

6.1 Least Squares Minimisation

The general setting for least squares minimisation is given as followed:

- a set of N observations $(\mathbf{h}_n, y_n)$ ($\mathbf{h}_n$ the label and $y_n \in \mathbb{R}$ the observed value), $n = 1, 2, \ldots, N$
- a set of corresponding weights $w_1, \ldots, w_n > 0$,
- an initial vector $\mathbf{x^0} = (x_1^0, \ldots, x_M^0) \in \mathbb{R}^M$,
- for every $n \in \{1, \ldots, N\}$, a function $Y(\mathbf{x}, \mathbf{h}_n) \in \mathbb{R}$ which is a linear polynomial in $\mathbf{x} = (x_1, \ldots, x_M) \in \mathbb{R}^M$ - that is,
$$Y(\mathbf{x}^0 + \Delta\mathbf{x}, \mathbf{h}_n) = Y(\mathbf{x}^0, \mathbf{h}_n) + \sum_{i=1}^{M} c_i(\mathbf{h}_n)\Delta x_i,$$
where $\Delta\mathbf{x} = (\Delta x_1, \ldots, \Delta x_M) \in \mathbb{R}^M$ with $c_i(\mathbf{h}_n) \in \mathbb{R}$.
- $N >> M$, an over-determined system.

Given this, we can generate N residuals $\Delta_n(\mathbf{x}) := Y(\mathbf{x}, \mathbf{h}_n) - y_n$, denoting the deviation of Y from y.

Our goal, given a start vector $\mathbf{x^0} = (x_1^0, \ldots, x_M^0) \in \mathbb{R}^M$, is to find the shift vector $\Delta\mathbf{x} = (\Delta x_1, \ldots, \Delta x_M) \in \mathbb{R}^M$ such that the *sum* of the *squares* of these residuals is minimised, that is

$$Z(\Delta\mathbf{x}) := \sum_{n=1}^{N} w_n \left(\Delta_n(\mathbf{x}^0 + \Delta\mathbf{x})\right)^2 \tag{6.1.1}$$

is minimised.

The construction of the function Z guarantees that $Z : \mathbb{R}^M \to \mathbb{R}$ is infinitely many times differentiable, non-negative and strictly convex and has therefore, a unique global minimum.

The minimum of Z is characterised by the following conditions on its derivatives:

$$\frac{\partial Z}{\partial x_j}(\Delta\mathbf{x}) = 0 \text{ for all } j = 1, 2, \ldots, M.$$

Or in short - grad$Z(\Delta\mathbf{x}) = 0$. So, we can seek out the solution by taking these derivatives of Z:

$$\begin{aligned}
\frac{\partial Z}{\partial x_j}(\Delta \mathbf{x}) &= \frac{\partial}{\partial x_j}\sum_{n=1}^{N} w_n \left(\Delta_n(\mathbf{x^0}+\Delta\mathbf{x})\right)^2 \\
&= \sum_{n=1}^{N} w_n \frac{\partial}{\partial x_j}\left(\Delta_n(\mathbf{x^0}+\Delta\mathbf{x})\right)^2 \\
&= \sum_{n=1}^{N} 2w_n \left(\Delta_n(\mathbf{x^0}+\Delta\mathbf{x})\right)\frac{\partial \Delta_n}{\partial x_j}\left(\mathbf{x}^0+\Delta\mathbf{x}\right) \\
&= \sum_{n=1}^{N} 2w_n \left(\Delta_n(\mathbf{x^0}+\Delta\mathbf{x})\right)\left(\frac{\partial Y}{\partial x_j}\left(\mathbf{x^0}+\Delta\mathbf{x},\mathbf{h_n}\right)\right).
\end{aligned}$$

Recalling that we can write Y as $Y(\mathbf{x}+\Delta\mathbf{x},\mathbf{h}) = Y(\mathbf{x},\mathbf{h})+\sum_{i=1}^{M} c_i(\mathbf{h})\Delta x_i$, and using our prior $\mathbf{x}^0$ to define $Y^0(\mathbf{h}) = Y(\mathbf{x}^0,\mathbf{h})$,

$$\frac{\partial Y}{\partial x_j}\left(\mathbf{x^0}+\Delta\mathbf{x},\mathbf{h_n}\right) = c_j(\mathbf{h}).$$

Given this, our above derivative becomes:

$$\begin{aligned}
\frac{\partial Z}{\partial x_j}(\Delta \mathbf{x}) &= \sum_{n=1}^{N} 2w_n \left(\Delta_n(\mathbf{x^0}+\Delta\mathbf{x})\right) c_j(\mathbf{h}) \\
&= \sum_{n=1}^{N} 2w_n \left(Y(\mathbf{x}^0+\Delta\mathbf{x},\mathbf{h}_n) - y_n\right) c_j(\mathbf{h}) \\
&= \sum_{n=1}^{N} 2w_n \left(Y^0(\mathbf{h}) + \sum_{i=1}^{M} c_i(\mathbf{h})\Delta x_i - y_n\right) c_j(\mathbf{h}) \\
&= \sum_{n=1}^{N} 2w_n c_j(\mathbf{h}) \left(Y^0(\mathbf{h}) - y_n\right) + \sum_{n=1}^{N} 2w_n c_j(\mathbf{h}) \sum_{i=1}^{M} c_i(\mathbf{h})\Delta x_i.
\end{aligned}$$

Thus, to find $\frac{\partial Z}{\partial x_j}(\Delta\mathbf{x}) = 0$, we have the following normal equations (after dividing the equations by 2):

$$\begin{aligned}
0 &= \sum_{n=1}^{N} w_n c_j(\mathbf{h}_n)\left(Y^0(\mathbf{h}_n) - y_n\right) + \sum_{i=1}^{M}\sum_{n=1}^{N} w_n c_j(\mathbf{h}_n) c_i(\mathbf{h}_n)\Delta x_i \\
&=: -v_j + \sum_{i=1}^{M} A_{ji}\Delta x_i. \qquad (6.1.2)
\end{aligned}$$

This defines a vector $\mathbf{v}$ (such that $v_j = -\sum_{n=1}^{N} w_n c_j(\mathbf{h}_n)\left(Y^0(\mathbf{h}_n) - y_n\right)$) and a matrix A (such that $A_{ji} = \sum_{n=1}^{N} w_n c_j(\mathbf{h}_n) c_i(\mathbf{h}_n) = A_{ij}$). Therefore, $\mathrm{grad} Z(\Delta\mathbf{x}) = 0$ implies that

$$A \cdot \Delta\mathbf{x} = \mathbf{v}.$$

Assuming invertibility of A, this allows us to determine $\Delta\mathbf{x}$ as

$$\Delta\mathbf{x} = A^{-1}\mathbf{v}.$$

The minimum of Z is then attained as $\mathbf{x}^0 + \Delta\mathbf{x}$. We call $\Delta\mathbf{x}$ the *shift* needed to get from the initial vector $\mathbf{x}$ to the minimum.

Given this solution, then $\mathbf{x}^0 + \Delta\mathbf{x}$ is a minimum of Z.

6.2 Non-Linear Least-Squares

An expansion of least-squares minimisation to non-linear functions is attained via the Gauss-Newton method. For this, we have a function $Y(\mathbf{x}, \mathbf{h}_n)$ which is *not* a linear polynomial in $\mathbf{x}$, but the remainder of the setup remains the same as in the linear least-squares case. For this, we must approximate Y by its linearisation at $\mathbf{x}^0$ which is then minimised. Then, given the adjustment made, a new linearisation at $\mathbf{x}^1 = \mathbf{x}^0 + \Delta\mathbf{x}$ is taken and minimised - this is repeated with $\mathbf{x}^2, \mathbf{x}^3$ until a minimisation is reached where $\Delta\mathbf{x}$ is numerically 0.

To linearise a function $g(\mathbf{x})$ at $\mathbf{x}^s$, we must take its Taylor expansion, that is:

$$\begin{aligned} g(\mathbf{x}^{\mathrm{s}} + \Delta\mathbf{x}) = g(\mathbf{x}^{\mathrm{s}}) &+ \sum_{j=1}^{M} \frac{\partial g}{\partial x_j}(\mathbf{x}^s)\Delta x_i \\ &+ \sum_{j=1}^{M}\sum_{i=1}^{M} \frac{\partial^2 g}{\partial x_i \partial x_j}(\mathbf{x}^s)\Delta x_i \Delta x_j \\ &+ \text{ higher powers of } \Delta x_i \end{aligned}$$

recalling that $\Delta\mathbf{x} = \begin{bmatrix} \Delta x_1 \\ \vdots \\ \Delta x_M \end{bmatrix}$.

We take those components containing products of more than one Δx_j to be negligible, as the Δx_j involved are very small. Then, our linearisation is:

$$g(\mathbf{x^s} + \Delta\mathbf{x}) \simeq g(\mathbf{x^s}) + \sum_{j=1}^{M} \frac{\partial g}{\partial x_j}(\mathbf{x}^s)\Delta x_j.$$

That is, we take $c_i(\mathbf{h}) = \frac{\partial g}{\partial x_i}(\mathbf{x}^s, \mathbf{h})$ and the remainder of the solution for each individual step stays the same.

6.3 Basic Application to Crystallography

In the crystallographic case, $Y(\mathbf{x}, \mathbf{h})$ is a either $|F(\mathbf{x}, \mathbf{h})|$ or $|F(\mathbf{x}, \mathbf{h})|^2$, where the *structure factor* $F(\mathbf{x}, \mathbf{h}) = \sum_{j=1}^{n} f_j(\mathbf{h}) e^{2\pi i \mathbf{z}_j \cdot \mathbf{h}} e^{-2\pi^2 \mathbf{h}^\top U_j \mathbf{h}}$. y_n are the amplitudes recorded from the screen $F_{\text{obs}}(\mathbf{h})$, or their intensities (the squares). The $\mathbf{x}$ represent atomic parameters for atoms within the asymmetric unit - thusfar, the locations ($\mathbf{z}_j$) of atoms and their ADPs (U_j).

Sometimes, because of internal symmetries, constraints, or other reasons, the vector $\mathbf{x}$ is a contraction of the U_j and $\mathbf{z}_j$. One example is a nitrogen atom constrained such that $x = y = z$, in which case there will be a single parameter in $\mathbf{x}$ corresponding to all 3 variables.

We additionally use a vector $\mathbf{y}$ which can be derived from $\mathbf{x}$ and contains the full $\mathbf{z}_j$ and U_j for each atom in order[1]. $\mathbf{x}$ will be of the same or lower dimension than $\mathbf{y}$. The vector $\mathbf{y}$ is useful for calculations related to the atomic parameters specifically, for example calculating cascaded uncertainties using matrices.

[1] that is, $\mathbf{y}^\top = [x_1, y_1, z_1, U_{1,11}, U_{1,22}, U_{1,33}, U_{1,12}, U_{1,13}, U_{1,23}, x_2, y_2, \ldots, U_{n,13}, U_{n,23}]$

Then for our two options:

$$Y(\mathbf{x},\mathbf{h}) = |F(\mathbf{x},\mathbf{h})| \text{ or } |F(\mathbf{x},\mathbf{h})|^2,$$

$c_i(\mathbf{h})$ is a little more difficult. For the case of $|F(\mathbf{x},\mathbf{h})|^2$, we first expand the equation to $F(\mathbf{x},\mathbf{h})\overline{F(\mathbf{x},\mathbf{h})}$, and then the differential is:

$$\begin{aligned}
c_i(\mathbf{h}) &= \frac{\partial}{\partial x_i}(|F(\mathbf{x}^s,\mathbf{h})|^2) \\
&= \frac{\partial}{\partial x_i}(F(\mathbf{x}^s,\mathbf{h})\overline{F(\mathbf{x^s},\mathbf{h})}) \\
&= \frac{\partial F}{\partial x_i}(\mathbf{x}^s,\mathbf{h})\overline{F(\mathbf{x}^s,\mathbf{h})} + F(\mathbf{x}^s,\mathbf{h})\overline{\frac{\partial F}{\partial x_i}(\mathbf{x}^s,\mathbf{h})} \\
&= 2\mathrm{Re}\left(\overline{F(\mathbf{x}^s,\mathbf{h})}\frac{\partial F}{\partial x_i}(\mathbf{x}^s,\mathbf{h})\right).
\end{aligned}$$

For the case of $Y(\mathbf{x},\mathbf{h}) = |F(\mathbf{x},\mathbf{h})|$, we use $|F(\mathbf{x},\mathbf{h})| = \sqrt{|F(\mathbf{x},\mathbf{h})|^2}$, and by the chain rule:

$$\begin{aligned}
c_i(\mathbf{h}) &= \frac{1}{2|F(\mathbf{x}^s,\mathbf{h})|}\frac{\partial}{\partial x_i}\left(|F(\mathbf{x}^s,\mathbf{h})|^2\right) \\
&= \frac{1}{2|F(\mathbf{x}^s,\mathbf{h})|}\left(2\mathrm{Re}\left(\overline{F(\mathbf{x}^s,\mathbf{h})}\frac{\partial F}{\partial x_i}(\mathbf{x}^s,\mathbf{h})\right)\right).
\end{aligned}$$

Of importance in both cases is the derivative,

$$\begin{aligned}
\frac{\partial F}{\partial x_i}(\mathbf{x}^s,\mathbf{h}) =& \sum_{j=1}^{n}\frac{\partial}{\partial x_i}\left(f_j(\mathbf{h})e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}}e^{-2\pi^2\mathbf{h}^\top U_j\mathbf{h}}\right) \\
=& \sum_{j=1}^{n} e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}}e^{-2\pi^2\mathbf{h}^\top U_j\mathbf{h}}\frac{\partial}{\partial x_i}(f_j(\mathbf{h})) \\
& + f_j(\mathbf{h})e^{-2\pi^2\mathbf{h}^\top U_j\mathbf{h}}\frac{\partial}{\partial x_i}(e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}}) \\
& + f_j(\mathbf{h})e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}}\frac{\partial}{\partial x_i}(e^{-2\pi^2\mathbf{h}^\top U_j\mathbf{h}}).
\end{aligned} \tag{6.3.1}$$

It is generally the case that x_i will only be related to to one of U_j or $\mathbf{z}_j$ - that is, one of $\frac{\partial}{\partial x_i}(e^{2\pi i\mathbf{z}_j\cdot\mathbf{h}})$ and $\frac{\partial}{\partial x_i}(e^{-2\pi^2\mathbf{h}^\top U_j\mathbf{h}})$ will be 0. As $f_j(\mathbf{h})$ is independent of the parameters

$\mathbf{x}$ (in the classical case of the Independent Atom Model, IAM), $\frac{\partial}{\partial x_i}(f_j(\mathbf{h}))$ is zero. We will discuss in Part III models in which $f_j(\mathbf{h})$ do depend on the parameters $\mathbf{x}$ in which case this partial derivative is no longer zero.

Where these cases and assumptions hold, only one of the 3 parts of the sum will be nonzero, and it will contain a factor $f_j(\mathbf{h})$.

Thus, in the case of refinement with respect to $Y_c(\mathbf{x}, \mathbf{h}) = |F_{\text{calc}}(\mathbf{x}, \mathbf{h})|^2$, given our current structure $\mathbf{x}^s$, we obtain the matrix A in (6.1.2) as

$$\begin{aligned} A_{ij} &= \sum_{n=1}^{N} w_n c_j(\mathbf{h}_n) c_i(\mathbf{h}_n) \\ &= 4 \sum_{n=1}^{N} w_n \frac{\partial Y_c}{\partial x_i}(\mathbf{x}^s, \mathbf{h}) \frac{\partial Y_c}{\partial x_j}(\mathbf{x}^s, \mathbf{h}) \end{aligned}$$

and the vector $\mathbf{v}$ as

$$v_j = 2 \sum_{n=1}^{N} w_n \mathrm{Re} \frac{\partial Y_c}{\partial x_i}(\mathbf{x}^s, \mathbf{h}) \left(Y(\mathbf{x}^s, \mathbf{h}_n) - y_n \right).$$

6.4 Incorporation of Restraints and Constraints

The use of Restraints and Constraints is a method by which the model is encouraged towards a theoretically justified solution. For example, one could know that a bond should be a particular length, or that two atoms are in related positions, meaning if one moves, the other should equivalently. These can be forced by constraints or encouraged by restraints.

Definition 6.4.1 (Constraint)**.** A crystallographic *Constraint* is a direct restriction on parameters. It will take the form of an equation on parameters to be enforced, and result in a reduction of the number of x_i refined.

A constraint may state that particular values are linked, for example $\mathbf{z}_H = \mathbf{z}_C + \mathbf{l}(C, H)$ states that a Hydrogen atom is precisely $\mathbf{l}(C, H)$ away from the Carbon atom. This will result in their locations being moved by identical vectors.

Definition 6.4.2 (Restraint)**.** A crystallographic *Restraint* provides an additional target to be refined towards. It will take the form of an additional quality T added to the *least squares function.* Incorporating a number of restraints denoted by T_i, the modified least squared function takes the form

$$Z(\Delta\mathbf{x}) := \sum_{n=1}^{N} w_n \left(\Delta_n(\mathbf{x}^0 + \Delta\mathbf{x})\right)^2 + \sum_{\text{restraints } i} w_i' \left(T_i^{\text{model}}(\mathbf{x}^0 + \Delta\mathbf{x}) - T_i^{\text{target}}\right)^2,$$

where the w_i' are weights corresponding to the strength of encouragement of the restraint i.

The additional restraints essentially provide additional data to refine against, though unlinked to $\mathbf{h}$ and with a different function to refine against.

If the above example were presented as a *restraint*, it would appear as

$$T^{\text{model}}(\mathbf{x}^0 + \Delta\mathbf{x}) =((\mathbf{z}_H^0 + \Delta\mathbf{z}_H) - (\mathbf{z}_C^0 + \Delta\mathbf{z}_C)), \qquad T^{\text{target}} =\mathbf{l}(C, H).$$

This would encourage the function to find the $\mathbf{z}_H, \mathbf{z}_C$ which have their difference closest to $\mathbf{l}(C, H)$.

6.5 Scaling factor $\tilde{K}$

Due to various factors, such as the distance to the screen and the relative power of the incoming X-ray beam, the recorded intensities have an unknown multiplier with relation to the calculated intensities which we will denote K.

That is, we *scale* each calculated intensity $Y_c(\mathbf{x}, \mathbf{h})$ to

$$Y_c'(\mathbf{x}, \mathbf{h}) = KY_c(\mathbf{x}, \mathbf{h}),$$

where K is some constant factor applied to all calculated intensities. By choosing some suitable K, we try to reduce the error between Y_{o} and Y_c', rather than against Y_c.

We know that we are trying to minimise the weighted sum of the squares of the residuals,

$$\Delta_n = r(\mathbf{x}, K) = Y_\text{o} - KY_c(\mathbf{x}). \tag{6.5.1}$$

The weighted sum can equivalently be written as a *weighted scalar product* $\langle\cdot,\cdot\rangle_w$ on $\mathbb{R}^N$, defined as

$$\langle X, Y\rangle_w = X^\top WY = \sum_{n=1}^{N} w(\mathbf{h}_n)X_nY_n, \tag{6.5.2}$$

for all $X = (X_1,\dots,X_N), Y = (Y_1,\dots,Y_N) \in \mathbb{R}^N$, where W is a diagonal weight matrix such that $W_{ii} = w_i$. The corresponding *weighted norm* is given by $\|X\|_w = \sqrt{\langle X, X\rangle_w}$.

This scale factor K is adapted in such a way that $\|r(\mathbf{x}, K)\|_w$ becomes minimal. Introducing $L(\mathbf{x}, K) = \|r(\mathbf{x}, K)\|_w^2$, we have

$$\begin{aligned} L(\mathbf{x}, K) &= \|Y_o\|_w^2 - 2K\langle Y_o, Y_c(\mathbf{x})\rangle_w + K^2\|Y_c(\mathbf{x})\|_w^2 \\ &= \|Y_c(\mathbf{x})\|_w^2\left(K - \frac{\langle Y_o, Y_c(\mathbf{x})\rangle_w}{\|Y(\mathbf{x})\|_w^2}\right)^2 + \|Y_o\|_w^2 - \frac{\langle Y_o, Y_c(\mathbf{x})\rangle_w^2}{\|Y_c(\mathbf{x})\|_w^2}, \end{aligned}$$

which immediately implies the following lemma:

Lemma 6.5.1 (see [4, (67) and (68)])**.** The weighted norm of the residual vector $r(\mathbf{x}, K)$, given by (6.5.1), is smallest for the following choice of the scale factor K:

$$\tilde{K} = \tilde{K}(\mathbf{x}) = \frac{\langle Y_c(\mathbf{x}), Y_o\rangle_w}{\|Y(\mathbf{x})\|_w^2}, \tag{6.5.3}$$

and we have

$$L(\mathbf{x}, \tilde{K}) = \|r(\mathbf{x}, \tilde{K})\|_w^2 = \|Y_o\|_w^2 - \frac{\langle Y_o, Y_c(\mathbf{x})\rangle_w^2}{\|Y_c(\mathbf{x})\|_w^2}.$$

We have the following explicit expression for the partial derivatives of the scale factor $\tilde{K}$:

Lemma 6.5.2 (see [4, (75)])**.** Let $\tilde{K}$ be given by (6.5.3). Then

$$\frac{\partial\tilde{K}}{\partial x_j}(\mathbf{x}) = \frac{1}{\|Y_c(\mathbf{x})\|_w^2}\left\langle\frac{\partial Y_c}{\partial x_j}(\mathbf{x}), Y_o - 2\tilde{K}(\mathbf{x})Y_c(\mathbf{x})\right\rangle_w.$$

Proof. This follows straightforwardly from (6.5.3):

$$\begin{aligned}\frac{\partial \tilde{K}}{\partial x_j}(\mathbf{x}) &= \frac{\left\langle \frac{\partial Y_c}{\partial x_j}(\mathbf{x}), Y_o \right\rangle_w}{\|Y_c(\mathbf{x})\|_w^2} - \frac{2\langle Y_c(\mathbf{x}), Y_o\rangle_w \left\langle \frac{\partial Y_c}{\partial x_j}(\mathbf{x}), Y_c(\mathbf{x}) \right\rangle_w}{\langle Y_c(\mathbf{x}), Y_c(\mathbf{x})\rangle_w^2} \\ &= \frac{1}{\|Y_c(\mathbf{x})\|_w^2} \left\langle \frac{\partial Y_c}{\partial x_j}(\mathbf{x}), Y_o - 2\frac{\langle Y_c(\mathbf{x}), Y_o\rangle_w}{\|Y_c(\mathbf{x})\|_w^2} Y \right\rangle_w .\end{aligned}$$

□

Finally, we define the scaled calculated intensity $\tilde{Y}_c = \tilde{K} Y_c$. Typically when working with the calculated intensities we assume that they are scaled in this way unless otherwise stated.

6.5.1 The residual vector $\tilde{r}$

The residual vector corresponding to the scale factor $\tilde{K}$ minimising $\|r(\mathbf{x}, K)\|_w$ is denoted by $\tilde{r}$:

$$\begin{aligned}\tilde{r}(\mathbf{x}) = r(\mathbf{x}, \tilde{K}(\mathbf{x})) &= Y_o - \tilde{K}(\mathbf{x}) Y_c(\mathbf{x}) \qquad (6.5.4) \\ &= Y_o - \frac{\langle Y_c(\mathbf{x}), Y_o\rangle_w}{\|Y_c(\mathbf{x})\|_w^2} Y_c(\mathbf{x}) \in \mathbb{R}^N,\end{aligned}$$

using the explicit expression for $\tilde{K}(\mathbf{x})$, given in (6.5.3).

Note that the residual vector $\tilde{r}(\mathbf{x})$ depends **only** on the weight matrix W and the theoretical and observed intensities $Y_c(\mathbf{x})$ and Y_o. The residual vector does **not** involve the partial derivatives $\frac{\partial Y_c}{\partial x_m}(\mathbf{x})$.

6.6 Solving for the shift

We now take a change in notation and name the shift vector $\mathbf{s} = \Delta\mathbf{x}$. This allows us to more easily denote different inputs to this shift vector, which will be required in Part III. Thus,

$$\mathbf{s} = A^{-1}\mathbf{v}$$

where

$$A_{ij} = \sum_{n=1}^{N} w_n \frac{\partial \tilde{Y}_c}{\partial x_i}(\mathbf{x},\mathbf{h}_n)\frac{\partial \tilde{Y}_c}{\partial x_j}(\mathbf{x},\mathbf{h}_n)$$

$$v_i = \sum_{n=1}^{N} w_n \frac{\partial \tilde{Y}_c}{\partial x_i}(\mathbf{x},\mathbf{h}_n)\left(\tilde{Y}_c(\mathbf{x},\mathbf{h}_n) - Y_o\right).$$

We can further expand this equation in matrix form for further crystallographic insights. For this, we define the *design matrix* D as a $N \times M$ matrix containing the derivatives:

$$(D(\mathbf{x}))_{nj} = \frac{\partial Y_c}{\partial x_j}(\mathbf{x},\mathbf{h}_n).$$

As usual, we will wish to work not with Y_c but $\tilde{Y}_c$, so we have an adjusted design matrix $\tilde{D}$:

$$\begin{aligned}\left(\tilde{D}(\mathbf{x})\right)_{nj} &= \frac{\partial(\tilde{K}Y_c)}{\partial x_j}(\mathbf{x},\mathbf{h}_n)\\ &= \frac{\partial \tilde{K}}{\partial x_j}(\mathbf{x})Y_c(\mathbf{x},\mathbf{h}_n) + \tilde{K}(\mathbf{x})\frac{\partial Y_c}{\partial x_j}(\mathbf{x},\mathbf{h}_n)\\ &= \frac{\partial \tilde{K}}{\partial x_j}(\mathbf{x})Y_c(\mathbf{x},\mathbf{h}_n) + \tilde{K}(\mathbf{x})(D(\mathbf{x}))_{nj}. \qquad (6.6.1)\end{aligned}$$

These two design matrices are related by

$$\tilde{D}(\mathbf{x}) = \tilde{K}(\mathbf{x})D(\mathbf{x}) + Y_c(\mathbf{x})\left(\operatorname{grad}\tilde{K}(\mathbf{x})\right)^{\top}, \qquad (6.6.2)$$

where both $Y_c(\mathbf{x})$ and the gradient are taken as column vectors. The design matrix $\tilde{D}(\mathbf{x})$ is, therefore, a rank-one perturbation of the rescaled design matrix $\tilde{K}(\mathbf{x})D(\mathbf{x})$. The matrix A from least square refinement is named the *normal matrix* $B(\mathbf{x}) \in \mathbb{R}^{N\times N}$, and is the symmetric matrix given by

$$B(\mathbf{x}) = \tilde{D}(\mathbf{x})^{\top}W\tilde{D}(\mathbf{x}), \qquad (6.6.3)$$

whilst

$$\mathbf{v} = \tilde{D}(\mathbf{x})^{\top}W\tilde{r}(\mathbf{x})$$

and the shift vector can then be computed via the formula

$$s(\mathbf{x}) = B(\mathbf{x})^{-1}\tilde{D}(\mathbf{x})^{\top}W\tilde{r}(\mathbf{x}).$$

This agrees with [4, (72)], namely, $Bs = -g$, where g coincides with $-\tilde{D}^{\top}(\mathbf{x})W\tilde{r}(\mathbf{x})$. Setting

$$S(\mathbf{x}) = B(\mathbf{x})^{-1}\tilde{D}(\mathbf{x})^{\top}W = (\tilde{D}(\mathbf{x})^{\top}W\tilde{D}(\mathbf{x}))^{-1}\tilde{D}(\mathbf{x})^{\top}W, \qquad (6.6.4)$$

with $\tilde{D}(\mathbf{x})$ given in (6.6.2), we obtain the shift vector $s(\mathbf{x})$ as the product

$$s(\mathbf{x}) = S(\mathbf{x}) \cdot \tilde{r}(\mathbf{x}).$$

Note that the design matrices $D(\mathbf{x})$ and $\tilde{D}(\mathbf{x})$ and, therefore, also the normal matrix $B(\mathbf{x})$ and the shift matrix $S(\mathbf{x})$ depend not only on the weight matrix W and the observed and theoretical intensities at $\mathbf{x} \in \mathbb{R}^N$ but also on the partial derivatives $\frac{\partial Y}{\partial x_n}(\mathbf{x})$.

6.7 Cholesky Decomposition

When processing least-squares refinement, we require the inverse of the normal matrix in (6.6.4). However, matrix inversion is something which is both time-wise costly and prone to errors when computer precision is involved - it is a process in which small errors can add, invert, and multiply quickly to create large problems.

Thus in many implementations of this kind of formula, matrix inversion is avoided. Instead, we utilise the process of *Cholesky Decomposition* [47, Ch.2.9], in which we split the matrix[2] into the product of an upper and lower triangular matrix, which are conjugate transposes of each other. This allows solution of the equation via forwards and backwards substitution. This too can suffer from precision issues, and a further improvement could be found via QR decomposition [43].

[2] This can only applied to a positive-definite matrix, but the normal matrix satisfies that requirement.

We take the form of the equation $\mathbf{s} = B^{-1}\tilde{D}^\top W\tilde{r}$. $\tilde{D}^\top W\tilde{r}$ can be calculated directly, and we equate it to $B\mathbf{s}$. Recall that $B = \tilde{D}^\top W\tilde{D}$ and is symmetric. We wish to create a lower triangular matrix L such that $B = LL^\top$.

The form of L is easy to deduce by expanding out $LL^\top$. That is (requiring that the elements of L are always written $L_{ij}, i \geq j$),

$$\begin{aligned} LL^\top &= \begin{bmatrix} L_{11} & 0 & 0 & \dots \\ L_{21} & L_{22} & 0 & \dots \\ L_{31} & L_{32} & L_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} L_{11} & L_{21} & L_{31} & \dots \\ 0 & L_{22} & L_{32} & \dots \\ 0 & 0 & L_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \\ &= \begin{bmatrix} L_{11}^2 & L_{11}L_{21} & L_{11}L_{31} & \dots \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & L_{21}L_{31} + L_{22}L_{32} & \dots \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \end{aligned}$$

These form a cascading series of rules. That is,

$$\begin{aligned} L_{11} &= \sqrt{B_{11}} & L_{21} &= B_{21}/L_{11} \\ & & L_{31} &= B_{31}/L_{11} \\ L_{22} &= \sqrt{B_{22} - L_{21}^2} & L_{32} &= \frac{B_{32} - L_{31}L_{21}}{L_{22}} \\ L_{33} &= \sqrt{B_{33} - L_{31}^2 - L_{32}^2} \end{aligned}$$

expanding to:

$$L_{ii} = \sqrt{B_{ii} - \sum_{k=1}^{i-1} L_{i,k}^2} \qquad L_{ij} = \frac{B_{ij} - \sum_{k=1}^{j-1} L_{ik}L_{jk}}{L_{j,j}}.$$

Then, we have $LL^\top\mathbf{s} = \tilde{D}^\top W\tilde{r}$. Taking this first as an equation of the form $L\mathbf{u} = \mathbf{w} = \tilde{D}^\top W\tilde{r}$, we use forward substitution to solve for $\mathbf{u} = L^\top\mathbf{s}$. That is,

$$L\mathbf{u} = \mathbf{w}$$

$$\begin{bmatrix} L_{11} & 0 & 0 & \dots \\ L_{21} & L_{22} & 0 & \dots \\ L_{31} & L_{32} & L_{33} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \end{bmatrix},$$

leading to the deduction that

$$\begin{aligned} u_1 &= w_1/L_{11} \\ u_2 &= \frac{w_2 - L_{21}u_1}{L_{22}} \\ u_3 &= \frac{w_3 - L_{31}u_1 - L_{32}u_2}{L_{33}} \\ u_i &= \frac{w_i - \sum_{k=1}^{i-1} L_{ik}u_k}{L_{ii}}. \end{aligned}$$

We can similarly solve $L^\top \mathbf{s} = \mathbf{u}$ through the similar process of backwards substitution, which first determines the last value s_N in $\mathbf{s}$ (where $\mathbf{s}$ is N in length) and cascades similarly back to the first, that is:

$$u_i = \frac{w_i - \sum_{k=i+1}^{N} L_{ik}u_k}{L_{ii}}.$$

This allows solving of Equation (6.6.4) without the use of matrix inversion, reducing the numerical errors introduced.

6.8 Uncertainties

Following the least squares process, we retrieve the uncertainties associated to the parameters $\mathbf{x}$ through a variance-covariance matrix defined as the following rescaling of the inverse of the normal matrix $B(\mathbf{x})$ (as defined in (6.6.3)):

$$\text{Var}(\mathbf{x}) := \frac{||\tilde{\mathbf{r}}||_w^2}{\#_{obs} - \#_{param}} B(\mathbf{x})^{-1} \tag{6.8.1}$$

(3.1.10.2 in [53]), where $\mathbf{x}$ is the vector of model parameters, $||\tilde{\mathbf{r}}||_w$ is the weighted norm introduced in (6.0.3) of the residual $\tilde{\mathbf{r}}$, $\#_{obs}$ is the number of observations taken into account in the refinement process and $\#_{param}$ is the number of parameters contained in the vector $\mathbf{x}$ (that is, its length).

The diagonal of $\mathrm{Var}(\mathbf{x})$ holds the variance $\sigma^2(x_i)$ for each parameter, whilst the off-diagonals hold the covariance $\mathrm{Cov}(x_i, x_j)$ between two parameters x_i and x_j. The covariance is largely used for further uncertainty calculations, but the variance (and in particular, its square root, the standard deviation) gives us an estimate of the likely position of the atom. The position of the atom can be seen as lying on a normal distribution for each parameter.

We also require uncertainties for derived parameters. Luckily with a normal distribution like this describing our parameters, it is simple to cascade these uncertainties. The variance of a function f is given by (3.1.10.3 in [53]):

$$\sigma^2(f) = \sum_{i,j} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} \mathrm{Cov}(x_i, x_j), \tag{6.8.2}$$

requiring the values of the covariances between all involved parameters as well as the partial derivative of the function with respect to each parameter, which must be derived analytically for each function.

6.9 Ending refinement

Given a non-linear optimisation function like we have for crystallography, we can continue to generate next shift vectors $\mathbf{s}$ infinitely. In fact, these shift vectors do not continue to get smaller but instead reach a limit in which they continue to generate similar-size steps around a point. Typically, these steps are small enough as to be significantly smaller than the uncertainties associated to the parameters.

It is common in crystallography to stop refinement when the ratio of the shift vector to the uncertainties is less than 0.01. This is known as the ‘shift/esd’, and is calculated by taking the maximum over all parameters of s_i/σ_i.

Chapter 7

Storage and Transfer of Data

Crystallography requires a large amount of data processing, often utilising different programs. To enable this, the community has settled on a few standard file types for recording and transferring this data.

7.1 cif files

The *Crystallographic Information File* (*.cif*) is the standardised format for storing all crystallographic data. It takes the form of a raw text file, with well outlined determiners to make it easy for a program to read and interpret. Designed by the IUCr Working Party on Crystallographic Information and adopted in 1990, the *.cif* file format forms a key part of the *Crystallographic Information Framework* [21], and was designed to make crystallographic information transfer viable and consistent between different machines and programs.

A *.cif* contains sets of data partitioned into *data blocks.* Such blocks contain various different elements of the crystallographic data, for example details of the machine used for the diffraction experiment and the temperature of the sample, or the recorded reflections. These blocks typically take a table-like format, with the entries for each data record defined at the start of the block and each data record presented on its own line. See [26] for the full details for the construction or reading of a *.cif* file.

7.2 hklf files

The file format typically used for conveying information about the recorded reflections of crystals is the *hklf4* format file.

The hklf4 file format consists of a row for each indexed reflection, containing in order the h, k and l indices, followed by the observed intensity I_o and the associated uncertainty σ. This file can contain multiple observations of the same reflection, which will simply occur on separate lines. A snippet of a hklf4 file is as follows:

```
2  0  0  400  20
2  0  0  390  30
2  0  1  200   4
2  0  2  600  50
```

All the reflections here are those with $h = 2, k = 0$, and are ordered in ascending l. This is a typical ordering for a *.hkl* file, but it is not necessary - the reflections can appear in any order. A ‘raw’ *.hkl* file will contain details of every indexed reflection, including multiple instances of the same indices. *.hkl* files can also be typically saved as ‘merged’ data, where those of the same indices are combined into a single line. Doing so loses information, but is done with the knowledge that identical indices imply the reflections should be identical, and any difference is measuring error, so taking an average is the best guess.

Such merged data are typically combined with regards to their weights - so for example, in the file above, the value of ‘400’ would be a more significant contributor to the 2 0 0 reflection than the value of ‘390’, having a default weight of 1/20 versus 1/30, leading to a merged value of 396.

7.3 Intermediate File Types

There are additional file types, typically organised in a similar raw text format to *.hkl* and *.cif* files.

Three such files which are typical during the refinement cycle are *.ins*, *.res* and *.fcf* files.

.ins and *.res* follow the same format, and represent *instructions* and *results*. They begin with defining lines with 4-letter signifiers, beginning with 'TITL' as the title of the atom, CELL, ZERR, LATT, SYMM carrying information about the unit cell information and symmetries, along with many optional parameters containing more information including weighting and form factor parameters [52], as well as the current atomic details of the model.

The atoms are written in a specific form, that is

```
Atom   Type x y z Occupancy U11 U22 U33 = U23 U13 U12
```

with all ADPs as U^*, see 4.3. The Type is an integer matching to the order of atoms presented in the SFAC command. The occupancy begins with a number indicating whether it should be refined (1 for not refined, 0 for freely refined, and '2' or '3' linking it to a specific free variable, with '-2' or '-3' linking to 1- that variable).

The *.res* file contains all the information present in the ins file, adjusted for any refinement steps performed and with information on the refinement stored in the case of olex2 behind REM (remark) lines.

The *.fcf* file echoes the format of the *.cif* file with loops of information, but focuses directly on the reflection (hkl) information (along with minor other things like the cell size and symmetries).

All of these file types are automatically interpreted and created by *olex2.refine* and typically by any other program designed to deal with crystallographic information. It is rare for a crystallographer to manually inspect these files.

Part II

Twinning

Chapter 8

Introduction

As I began my PhD, we sought a way for me to learn about the code behind Olex2. We decided that I would take on the implementation of an algorithm used in ROTAX [11]. During the process of implementing this, I developed particular ideas for an algorithm of my own, making more use of the information available for a more sophisticated search. The development of this algorithm is one of the two major projects undertaken over the course of my PhD.

In this part, I will begin with an overview of the twinning problem itself, followed by a discussion of the current available solutions and continuing with my own solution and the necessary mathematics. In the final chapter of this part, I will discuss further ideas which arose during ESGI 165, and leave with a hope for many future discoveries.

8.1 Motivation

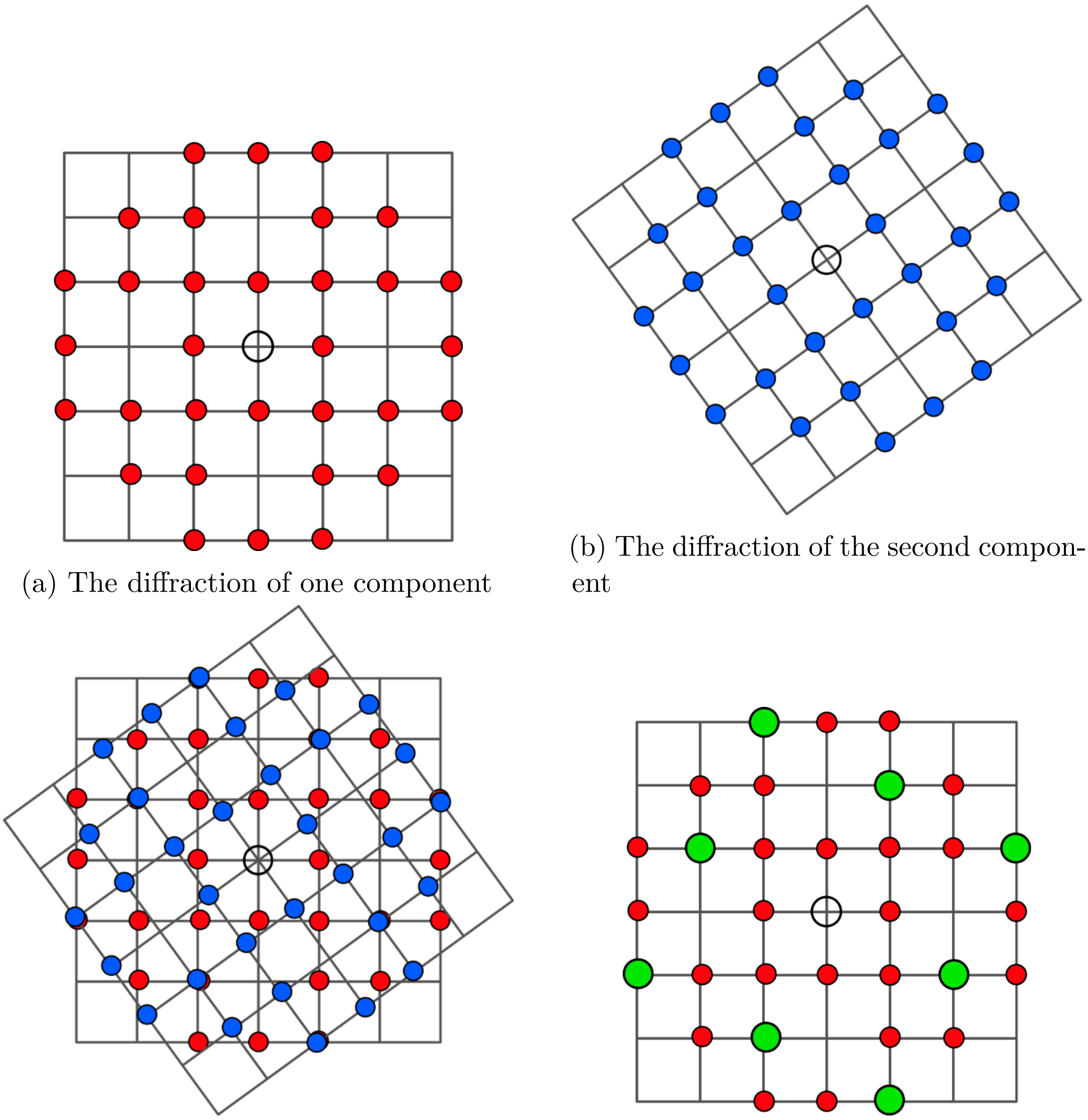
During the physical creation of a crystal, defects can arise. One of these, known as *twinning*, occurs when the crystal is comprised of multiple *components*. Such a crystal can occur in many ways, for example an *intergrowth* twin occurs when two crystals grow into one another, whilst some twins occur within a single crystal along a crystal face, where the new molecules connect in a rotated form. Each of these components has the same basic repeating unit cell, but it is subject to some rotation (or roto-inversion).

When in the path of an X-ray beam, each of these *components* will give rise to its own diffraction pattern (Figures 8.1a and 8.1b), which are overlaid (Figure 8.1c) and both contribute to the final detected diffraction pattern 8.1d. The red dots in Figure 8.1 correspond to the primary component, which makes up a majority of the crystal, whilst the second component's intensities are shown with the blue dots. The software detecting these dots looks only in the locations where the red dots are, but some of the dots (shown in green) are increased in intensity by the blue dots. This adds additional unknowns and complexity to the solving of the crystal structure.

Mathematically, the challenge is to identify the corresponding rotations of the components as well as the proportion of each making up the crystal, which together comprise the 'twin law'. In its simplest form, we hunt for rotations for which the right kind of overlap appears.

Which points are overlapped (green in Figure 8.1) is not an obvious matter either. We take those points where the observed intensities and theoretical intensities differ the most as the likely set of overlapped points, but allow that some may not in fact be overlapped, and may be subject to other artifacts or a particular quirk of the refinement thus far.

One may note that in Figure 8.1c the blue dots are often imperfectly overlapping the red, and yet we still consider them to contribute to the observed diffraction pattern. In this work, we assume that two such diffraction spots may either be *overlapping* or *not overlapping* - there is no such thing as partial overlap (even if the spots do not rest on precisely the same point). My basic understanding of the integration software which identifies and records the intensity of these points supports this, but this is a potential area for further investigation alongside those softwares for more accurate determination of twins.



(a) The diffraction of one component

(b) The diffraction of the second component

(c) The diffraction patterns overlaid

(d) The final observed diffraction

Figure 8.1: The diffraction patterns provided by multiple twin components, and the resulting detected pattern

We utilise a threshold, ϵ, for which if two spots differ by less than ϵ they are treated as overlapping, and otherwise are not overlapping. This ϵ will be utilised throughout this Part, and one typical value for it is 0.002Å.

In this part, we first present background knowledge for the study of twinning, including a brief review of fundamental concepts used in the literature in Subsection 8.2.4. We continue with a chapter on the previous solutions to the twinning problem, and follow with a chapter on my own algorithms, and the additional problems which needed to be solved for them. Finally, we present our views on the future of this investigation, and some promising results from ESGI 165 where we presented the issue of twins to other mathematicians.

8.2 Background

A *twin law* is the collection of information allowing the modelling of the twin component within the crystal. This information comprises the Twin Orientation matrix R_{cart} and the Batch Scale-factor $\mathcal{B}$, outlined below.

Definition 8.2.1 (Twin Orientation)**.** Given two components, A and B with crystal structures (S_A, p_A) and (S_B, p_B) respectively (see Definition 2.1.1 in Part I), within a crystal, their relative orientations can be related by a rotation matrix $R_{\text{cart}} \in$ O(3). That is, $S_B = R_{\text{cart}} S_A + \mathbf{t}$, where the translational difference $\mathbf{t}$ is generally undeterminable, but also has no effect on diffraction and thus is unimportant to the structure determination. Additionally, taking $\mathbf{s}_B = R_{\text{cart}} \mathbf{s}_A + \mathbf{t}$ for each $\mathbf{s}_B \in S_B$, we must have $p_B(\mathbf{s}_B) = p_A(\mathbf{s}_A)$. We call R_{cart} the *twin orientation.* Most commonly, twin orientations are represented as a rotation (or roto-inversion) by some *angle* θ about some axis $\mathbf{k}$.

Mathematically, the orientation of an orthogonal matrix is the sign of its determinant (positive implying orientation preserving and negative implying orientation reversing), but in crystallography the twin orientation is the matrix itself.

The the rotation matrix applied to the crystals in direct space is the same as the rotation applied to the reflections in reciprocal space. This is easily demonstrated. Let us take a rotation R_{cart} such that $S_B = R_{\text{cart}} S_A + \mathbf{t}$, and respective lattices Γ_B and Γ_A along with their reciprocal lattices Γ^*_B and Γ^*_A.

The existence of a twin law can be written mathematically as $S_B = R_{\text{cart}} S_A + \mathbf{t}$. Assuming this situation, we now show that $\Gamma_B = R_{\text{cart}} \Gamma_A$. Recall that Γ preserves S, that is $S + \mathbf{s} = S$ for any $\mathbf{s} \in \Gamma$. We start with an element $\mathbf{s}_A \in \Gamma_A$, and note that $S_A + \mathbf{s}_A = S_A$. We can pre-multiple this equality by R_{cart}:

$$R_{\text{cart}} S_A = R_{\text{cart}} (S_A + \mathbf{s}_A),$$

adding $\mathbf{t}$ to return to S_B we obtain:

$$R_{\text{cart}} S_A + \mathbf{t} = R_{\text{cart}} S_A + \mathbf{t} + R_{\text{cart}} \mathbf{s}_A$$

$$S_B = S_B + R_{\text{cart}} \mathbf{s}_A$$

which implies $R_{\text{cart}} \mathbf{s}_A \in \Gamma_B$, and thus $R_{\text{cart}} \Gamma_A \subset \Gamma_B$. As we can equivalently perform this proof in reverse, $R_{\text{cart}} \Gamma_A = \Gamma_B$. Thus $R_{\text{cart}} \Gamma_A$ is a preserving lattice for S_B. As shown in 3.3.3 in Part I, $\Gamma_B = R_{\text{cart}} \Gamma_A \Rightarrow \Gamma^*_B = R_{\text{cart}} \Gamma^*_A$.

Definition 8.2.2 (Batch Scale-Factor ‘basf’)**.** Given multiple components, A_i, within a crystal, their proportion of the total crystal is denoted by a batch scale factor, or ‘basf’, $\mathcal{B}_i \in [0, 1]$. Typically, these will be listed for every component but the first, with $\mathcal{B}_1 = 1 - \sum_{i=2}^{N} \mathcal{B}_i$.

Twin laws require a modification to the theoretical intensities I_c used in the model [29]. Given a twinned crystal with s components and N measured intensities (we use $n = 1, \ldots, N$ here to enumerate the recorded reflections as they are no longer uniquely given by a particular $\mathbf{h}$), with each component i having an associated basf $\mathcal{B}_i$ such that $\sum_{i=1}^{s} \mathcal{B}_i = 1$. For a given reflection n, $s_n \leq s$ components will contribute to it, enumerated as $1 \leq i_{n,1} < \cdots < i_{n,s_n} \leq s$, giving

$$I_c(n) = \sum_{l=1}^{s_n} \mathcal{B}_{i_{n,l}} |F_{\text{calc}}(\mathbf{h_{n,i_{n,l}}})|^2, \tag{8.2.1}$$

where $\mathbf{h}_{n,i_{n,l}}$ is the associated Miller index for component $i_{n,l}$ for the nth reflection, and I_c is our final theoretical intensity.

8.2.1 Reflections with Likely Overlap

Within the process of searching for twin laws, we require the identification of candidates for reflections which are likely to be overlapped by a second component.

We find these 'bad' hkl as follows: we take the signed difference $I_o - I_c$ between the observed intensity I_o and theoretical intensity I_c, and divide it by the uncertainty σ on the observed intensity, that is we consider $(I_o - I_c)/\sigma$. This gives us the degree to which this particular reflection is underestimated (that is, $I_c < I_o$). Those with the highest value are the most likely to be subject to an overlap from a second component. Looking for twin laws which will overlap many of the most underestimated points are the most likely to mitigate this difference and form a good model of the crystal twin.

Throughout this Part, we will refer to these as *bad hkl*. Whilst we really mean 'the hkl with the largest weighted underestimation on its associated intensity', the shortness of the phrase 'bad hkl' allows a far quicker understanding in the contexts we will be using it, and as they are 'the reflections most in need of correction', it is not inaccurate to name them 'bad'.

8.2.2 The Rodriguez Formula: Relation between R, Axis, and Angle

In this subsection, we outline the formula for obtaining a matrix R_{eucl} from an axis and angle, which is known as the Rodriguez formula. All of our algorithms rely on this method for determining the rotation matrix to use, as axis and angle provide 3 clear parameters (two for a unit axis, one for the angle) which are typically the easiest to determine of any possible rotation representation. Thus, it is vital to have a direct method to move from angle and axis to rotation matrix, which can be utilised directly to transform reflection indices.

Proposition 8.2.3 (Rodriguez Formula)**.** A directed axis $\mathbf{k}$ and angle θ give rise to a unique rotation R_{eucl} in 3d space given by

$$R_{\text{eucl}} = I + \sin\theta K + (1-\cos\theta)K^2, \tag{8.2.2}$$

where K is the cross-product matrix of the unit axis

$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}, K = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix}.$$

Note that $K\mathbf{v} = \mathbf{k} \times \mathbf{v}$ for any vector $\mathbf{v} \in \mathbb{R}^3$, which can be easily shown through expansion of the cross product:

$$\mathbf{k} \times \mathbf{v} = \begin{bmatrix} k_y v_z - k_z v_y \\ k_z v_x - k_x v_z \\ k_x v_y - k_y v_x \end{bmatrix} = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

Another representation of 8.2.2 is also quoted in [53] as

$$R_{\text{eucl},ij} = k_i k_j (1-\cos\theta) + \delta_{ij}\cos\theta + e_{ipj} k_p \sin\theta,$$

where e_{ipj} is the antisymmetric tensor, that is:

$$e_{ipj} = \begin{cases} +1, & \text{if } ipj \text{ is an even permutation of 123} \\ -1, & \text{if } ipj \text{ is an odd permutation of 123} \\ 0, & \text{if } ipj \text{ has two equal values.} \end{cases}$$

Proof of Proposition 8.2.3. We can easily derive Formula 8.2.2 by considering the action of R_{eucl} on a vector:

Take $\mathbf{v} \in \mathbb{R}^3$, and a unit vector $\mathbf{k}$ defining the axis of rotation. Define the cross-product matrix K as above, and note that $K\mathbf{v} = \mathbf{k} \times \mathbf{v}$.

Then note that the component of $\mathbf{v}$ parallel to $\mathbf{k}$ is $\mathbf{v}_{\|} = (\mathbf{v} \cdot \mathbf{k})\mathbf{k}$. The component perpendicular is

$$\begin{aligned}
\mathbf{v}_{\perp} &= \mathbf{v} - \mathbf{v}_{\|}, \\
&= \mathbf{v} - (\mathbf{v} \cdot \mathbf{k})\mathbf{k}, \\
&= (\mathbf{k} \cdot \mathbf{k})\mathbf{v} - (\mathbf{v} \cdot \mathbf{k})\mathbf{k}, \\
&= -\mathbf{k} \times (\mathbf{k} \times \mathbf{v}), \\
&= -K^2\mathbf{v},
\end{aligned}$$

utilising the identity of the vector triple product, $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} + (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$, along with our definition of K.

Then we note that the component $\mathbf{v}_{\|}$ will not be altered by R_{eucl}, whilst the component $\mathbf{v}_{\perp}$ will be rotated. Note that as $\mathbf{v}_{\perp}$ is orthogonal to $\mathbf{k}$, $\mathbf{k} \times \mathbf{v}_{\perp}$ is a rotation of $\mathbf{v}_{\perp}$ around $\mathbf{k}$ by $\pi/2$. Then, the rotation of $\mathbf{v}_{\perp}$ by θ is $\sin\theta \mathbf{k} \times \mathbf{v}_{\perp} + \cos\theta \mathbf{v}_{\perp}$. We note for below that $\mathbf{k} \times \mathbf{v}_{\perp} = \mathbf{k} \times \mathbf{v}$ as the parallel component cross product is 0. Thus:

$$\begin{aligned}
R_{\text{eucl}}\mathbf{v} &= R_{\text{eucl}}\mathbf{v}_{\perp} + \mathbf{v}_{\|}, \\
&= \sin\theta \mathbf{k} \times \mathbf{v}_{\perp} + \cos\theta \mathbf{v}_{\perp} + \mathbf{v}_{\|}, \\
&= \sin\theta \mathbf{k} \times \mathbf{v} - \cos\theta K^2\mathbf{v} + \mathbf{v}_{\|}, \\
&= \sin\theta K\mathbf{v} - \cos\theta K^2\mathbf{v} + \mathbf{v} - \mathbf{v}_{\perp}, \\
&= \sin\theta K\mathbf{v} - \cos\theta K^2\mathbf{v} + \mathbf{v} + K^2\mathbf{v}, \\
&= \sin\theta K\mathbf{v} + \mathbf{v} + (1 - \cos\theta)K^2\mathbf{v}.
\end{aligned}$$

Therefore, we have $R_{\text{eucl}} = \sin\theta K + I + (1 - \cos\theta)K^2$ □

8.2.3 Representation in Relative Coordinates

In our further considerations it will be important to write this twin orientation also as a matrix in relative reciprocal space, that is as acting directly on the integers h, k, l.

To do this, we consider the action of the rotation matrix R_{eucl} on a vector $\mathbf{h} = \begin{bmatrix} h \\ k \\ l \end{bmatrix}$. First, the vector must be transformed into Cartesian coordinates through the use of the reciprocal orthogonalisation matrix A^*. This allows it to be rotated by R_{eucl}, and the result can then, through the application of the inverse reciprocal orthogonalisation matrix, be transformed back into relative space. That is,

$$\mathbf{h}' = (A^*)^{-1} R_{\text{eucl}} A^* \mathbf{h}.$$

Given this, we can define the matrix $R_{\text{rel}} = (A^*)^{-1} R_{\text{eucl}} A^*$ as the relative rotation matrix, such that $\mathbf{h}' = R_{\text{rel}}\mathbf{h}$. Doing so allows much easier identification of the proportion of overlapped points (and whether all points are overlapped, given by matrices containing only integers) n as seen in Subsection 8.2.4, and an intuitive understanding of the impact on the reciprocal basis vectors $\mathbf{a}^*, \mathbf{b}^*$ and $\mathbf{c}^*$.

Example 8.2.1. For example, [23] contains a twin matrix (Example 2) which is

$$R_{\text{eucl}} = \begin{bmatrix} 0.4120 & -0.5422 & 0.7323 \\ 0.5422 & -0.5 & -0.6753 \\ 0.7323 & 0.6753 & 0.0880 \end{bmatrix}$$

in Cartesian space, and

$$R_{\text{rel}} = \begin{bmatrix} 0.5 & -0.5 & 1 \\ 0.5 & -0.5 & -1 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

in relative space [1]. When written in relative space, this R_{rel} brings $h, k, l \in \mathbb{Z}^3$ with $(h+k)$ even into overlap, and thus overlaps half the points in reciprocal space. This is not deductible from the Cartesian matrix, and so both for human understanding and the ease of converting indices, we default to presenting the rotation matrices in relative coordinates.

Throughout the remainder of this Part, R will typically refer to the relative matrix R_{rel} as it is more important for our consideration. We mention this at this point to avoid confusion.

8.2.4 Classification of Twins

There are many terms used for how well a twin component matches onto the main component lattice (from now on, we assume that one component is designated the 'main' component). Our main source for the basic descriptions of twinning is found in [2], and for a review of sub-types of twins and their related terms we recommend chapter 4.3 Crystal Twins in [2]. Twin laws are typically be split into 4 categories dependent on

1. how many reflections are overlapped, represented by a *twin index* n
2. how well the reflections overlap in angle, the *twin obliquity* ω

Definition 8.2.4 (Twin Index n)**.** For the lattice Γ^* of reflections from the main component, we take the *twin lattice* Γ^*_{twin} as the sub-lattice containing only points which are overlapped by the twin component. Then, the index n of Γ^*_{twin} in Γ^* is known as the *twin index* and can be understood as the inverse of the proportion of

[1] For ease, the lattice parameters of $a = 10.220\text{Å}, b = 11.083\text{Å}, c = 7.538\text{Å}$, $\beta = 96.85°$ and $\alpha = \gamma = 90°$ result in a direct orthogonalisation matrix of $A = \begin{bmatrix} 10.220 & 0 & -0.899 \\ 0 & 11.083 & 0 \\ 0 & 0 & 7.484 \end{bmatrix}$ leading to a reciprocal orthogonalisation matrix $A^* = \begin{bmatrix} 0.09785 & 0 & 0 \\ 0 & 0.09023 & 0 \\ 0.01174 & 0 & 0.133612 \end{bmatrix}$.

spots in the main lattice present in the twin lattice. For group theorists, the twin index is given by $n = [\Gamma^* : \Gamma^*_{\text{twin}}]$.

Mathematically, $\Gamma^*_{\text{twin}} = \Gamma^* \cap R_{\text{eucl}}\Gamma^*$, where Γ^* is the lattice of the main component, and R_{eucl} the rotation matrix of the twin law. The index can also be interpreted as the quotient of the volume of the supercell (a fundamental domain of Γ^*_{twin}) and the volume of the unit cell of Γ^* (a fundamental domain of Γ^*).

A twin index of 1 indicates that all points overlap, whilst other indices (essentially always natural numbers) indicate partial overlap. The twin lattice and its associated *supercell* can help to identify plausible twin laws as symmetries of the twin lattice[11].

The *twin obliquity* ω is explicitly defined only for twofold twins, requiring a twin plane (hkl) as well as a twin axis $[u, v, w]$, and ω is defined as the angle between the normal to the plane and the twin axis.

More colloquially, it measures the *angular separation* between the two contributors to each overlapping point. Given a rotation matrix R, $R\mathbf{h}_2$ may provide values which are very close to, but not quite, integers. In this case, the point $R\mathbf{h}_2$ will still overlap with some $\mathbf{h}_1$ from the main component, but there will be a slight separation between the centres of the two reflections. Ignoring any separation due to $|A^*R\mathbf{h}_2| \neq |A^*\mathbf{h}_1|$, we take ω as the angle between $R\mathbf{h}_2$ and $\mathbf{h}_1$. This is not an official definition, but appears to match the casual usage of ω. In my opinion, the presented official definition does not provide sufficient clarity, and has no capacity to consider non-twofold twins. However, the casual use of ω to divide between twins with perfect and imperfect overlap (focusing on 0 or non-zero values) does provide a good distinction on types of twins. [40] presents an empirical maximal value of $\omega = 6^\circ$ for a reasonable twin[2].

Notably, there seems to be little consideration for points which do not perfectly

[2] This is notably not a spatial distance but an angle, and whether this is easier or more difficult to deal with remains to be seen. It does mean that higher-index spots are just as likely to overlap as low-index spots.

overlap due to being a slightly different distance from the origin. That is, $|A^*R\mathbf{h}_2| \neq |A^*\mathbf{h}_1|$ for at least one $\mathbf{h}_2 \in \Gamma^*_B$ where $R\mathbf{h}_2$ is almost-integer, and associated $\mathbf{h}_1 \in \Gamma^*_A$). [42] suggests calling this the *twin misfit* δ, and measure it as the separation on the closest overlapping point to the origin, in a plane perpendicular to the twin axis. That is, we take the shortest possible nonzero $\mathbf{h}_1$ matching some $R\mathbf{h}_2$ as defined above, and split each into components parallel ($\mathbf{h}_{1,\|}$ and $\mathbf{h}_{2,\|}$) and perpendicular ($\mathbf{h}_{1,\perp}$ and $R\mathbf{h}_{2,\perp}$) to the axis of rotation. Then the misfit $\delta = |A^*\mathbf{h}_{1,\perp} - A^*R\mathbf{h}_{2,\perp}|$. We note that $\omega \neq 0 \Rightarrow \delta \neq 0$, though the inverse does not apply. Again, I find it strange that this disregards any difference due to differences parallel to the axis of rotation.

Table 8.1 gives the names of twin laws with respect to n and consideration of the overlap (perfect or imperfect), which relates to ω and δ but remains a little more general. The terms related to *merohedral twinning* are the most commonly used, with *TLS/TLQS* (Twin Lattice Symmetry & Twin lattice Quasi-Symmetry) seeming to be a recent but encouraged term, and *integral* mostly a literal description not commonly seen.[3]

	no n constraint	$n = 1$	$n > 1$
no overlap restriction	twin	integral twin	non-integral twin
perfect overlap	TLS	merohedral twin	reticular merohedral twin OR non-merohedral twin
imperfect overlap	TLQS	pseudo-merohedral twin	reticular pseudo-merohedral twin OR non-merohedral twin

Table 8.1: The names dependent on type of twinning, based on the proportion of overlapped points ($1/n$) and degree of overlap[4].

[3] [41] states that the common use of 'merohedral twin' is incorrect, and it should be 'merohedric twin' or 'twin by merohedry'. However, 'merohedral twinning' is a common term, whilst 'merohedric' is rarely if ever used, so here I remain with the majority usage.

[4] The degree of overlap is often described with ω, but I find that the definition and typical use do not perfectly match, so we instead use the more descriptive terms.

In an *integral twin law*, all reflections $\mathbf{h}$ are mapped directly onto other reflections - all entries of the relative twin matrix R are integers (potentially up to a small threshold of error, allowing $R\mathbf{h}_2$ to be a small distance from a lattice point but requiring that all $R\mathbf{h}_2$ are sufficiently close to a lattice point). This typically arises as an element of the point group of the unit cell of the crystal (meaning the unit cell has specific dimensions which enable this). In this case, *all* lattice points will be overlapped, and therefore the twin index $n = 1$.

Merohedral twinning refers to what I might call 'perfect' integral twins [46], where the twin law arises due to additional symmetries in the point group of the crystal which are not present in its space group. As an example, ammonia crystallises in a necessarily cubic unit cell with a threefold rotation about the $x = y = z$ axis. Because of the cubic unit cell, if there were to be a second component rotated 90° about the x-axis, its diffraction pattern would perfectly align with that of the first component. *Pseudo-merohedral* twinning on the other hand occurs from 'accidental' such angles, where for example a unit cell has an angle very close to 90° without any symmetries necessitating that. Another example would be if the a and b axes were very similar in length and perpendicular to the c axis. For our intents and purposes, both merohedral and pseudo-merohedral are integral twin laws.

The detection of Merohedral twin laws seem to be largely a solved process - to find rotations which perfectly overlap the reflections is simple for space groups whose point group contains additional symmetry elements, which are pre-tabulated and (relatively) short lists. Detecting twin laws for Pseudo-merohedral twins may be a little harder, but the process boils down to checking whether a point group with additional symmetry elements can have the same (or similar) unit cell - for example, a cell which happened to have $a \approx b \approx c$ and 90.1° angles would be able to have any twin law which lay in the point group of a cubic cell, even if the crystal itself did not require cubic symmetry.

Non-integral twin laws have relative rotation matrices R which have some non-integer entries. They overlap some reflections, perhaps non-perfectly, and do not overlap others. These can be more easily spotted in the diffraction slides than integral twins as they would appear to have two lattices. However, often these are not spotted and it is indexed as a single crystal. These can also be referred to as reticular merohedral twins (which again are split into normal and pseudo-merohedral dependent on whether they are 'perfect' or not), which refers to the twin operation being a symmetry of the twin lattice sub-lattice. I have found no consideration of twin operations which do not belong to the symmetry of a sub-lattice, and feel certain they must exist, but perhaps come with $n = \infty$ and are not significant enough to be noticed (for example, if $\mathbf{a}^*$ was mapped onto $\mathbf{b}^*$ through an angle of 80° (with $||\mathbf{a}^*||=||\mathbf{b}^*||$, $\mathbf{c}^*$ perpendicular to $\mathbf{a}^*$ and $\mathbf{b}^*$), only those points with $h = 0$ would be overlapped (and would be mapped onto by only points with $k = 0$). Any twin law which is not perfectly merohedral can also be referred to as 'non-merohedral', despite the fact that they can also be referred to as 'pseudo' or 'reticular' merohedral laws.

Some experienced crystallographers can deduce possible non-merohedral twins by inspection of the lattice parameters [46].

Example 8.2.2. We look at Example 2 from [23]. This has a relative matrix of

$$R = \begin{bmatrix} 0.5 & -0.5 & 1 \\ 0.5 & -0.5 & -1 \\ 0.5 & 0.5 & 0 \end{bmatrix},$$

that is

$$\begin{bmatrix} h \\ k \\ l \end{bmatrix} \to \begin{bmatrix} \frac{h-k}{2} + l \\ \frac{h-k}{2} - l \\ \frac{h+k}{2} \end{bmatrix}.$$

This is a non-integral matrix with $n = 2$, overlapping half the points (those with even $h + k$). This is also a threefold twin law (that is, $R^3 = I$). This rotation has an axis $[2, 0, 1]$ (in the relative reciprocal basis) and as a threefold law an angle of 120°.

The twin lattice for this is generated with $\mathbf{a}^*+\mathbf{b}^* = \begin{bmatrix} 0.09785 \\ 0.09023 \\ 0.01174 \end{bmatrix}$, $\mathbf{a}^*-\mathbf{b}^* = \begin{bmatrix} 0.09785 \\ -0.09023 \\ 0.01174 \end{bmatrix}$ and $\mathbf{c}^* = \begin{bmatrix} 0 \\ 0 \\ 0.133612 \end{bmatrix}$, with a matrix with respect to this system of $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & 0 \end{bmatrix}$, with an axis of $[1, 1, 1]$ (in relative coordinates) in this system. We calculate an angular discrepancy of 0.006° and a difference in norm of 6×10^{-6}Å for $\mathbf{a}^* + \mathbf{b}^*$ rotated onto $\mathbf{a}^* - \mathbf{b}^*$, which is sufficiently small as to imply perfect overlap and simple errors due to limited precision. We thus call this a reticular merohedral twin.

8.2.5 hklf5 Format Files

Information about the reflections from twinned crystals is stored in a special variant of hkl-file, adapted to allow the indication of which reflections are overlapped, or belong to different components if multiple diffraction lattices were indexed at the time of slide integration.

The file format typically used for conveying information about the diffraction patterns of crystal twins is the *hklf5* format file. This is an adaptation of the standard *hklf4* format file type used to transfer information about recorded reflections.

The hklf5 file format adapts the hklf4 format (see 7 of Part I) to denote overlapped reflections, and reflections occurring from different crystal components. It has an additional column (highlighted in red in the example below) at the end of each row which serves two purposes. Firstly, this column may be without a sign (positive) or with a '-' symbol. A row with a '-' symbol indicates that it is an overlapped reflection, and is matched to the line below it. Secondly, this column will contain an integer, which (without the potential '-') denotes the component associated to that line. This component number is directly related to the hkl labelling present in that line, which corresponds to the point in reciprocal space with respect to the numbered component.

This is far easier to understand with an example. Consider a two-component crystal with a twofold twin matrix $R_{\text{rel}} = \begin{bmatrix} -1 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ in relative reciprocal space. A snippet of such a hklf5 file is as follows:

```
-2  0  0  400  20  -2
 2  0  0  400  20   1
 2  0  1  200   4   1
-1  0  2  600  50  -2
 2  0  2  600  50   1
 2  0  1  100   5   2
```

The first two lines show that the '2 0 0' reflection of the first component is overlapped with the '-2 0 0' reflection of the second component, which is presented in the first line with a '-' symbol. The third line is a reflection unique to the first component - if we apply the rotation matrix, we get '-1.5 0 1', which contains a non-integer and thus not an overlapped point. The fourth and fifth lines show that the '2 0 2' reflection of the first component is again overlapped by the '-1 0 2' reflection of the second component.

The final line shows a unique reflection of the second component, with the Miller triple labelling as within this component. Such an entry in a hklf5 file can only come about if such reflections are also recorded at the indexing stage, but this is a possibility and one allowed by the hklf5 format file.

One note about the hklf5 format file is that the ordering in the file is very important - a '-' symbol notes that the reflection is the same as the one on the next line, so any change to this order would render these incorrect. The hklf5 format also duplicates the I_o and σ information (in the 4th and 5th columns) on such 'duplicated' reflection lines.

For some twin laws, especially integral ones, hklf5 is not necessary and the reflections can be transformed by refinement software. hklf5 formatted files are most important where multiple components have been directly indexed and thus one needs to clearly indicate overlapped reflections, and which reflections belong to different components. For example, in our example above we had a '2 0 1' reflection for both components, but the reflection caused by the second component was half as strong (with a larger error).

As a first guess, one could assume that the second component is half as large as the first component - however, such an estimate would need to be verified by other information. This could further lead to a deduction that the '2 0 2'/'-1 0 2' reflection has an intensity of 400 from the first component and 200 from the second, in a process known as 'detwinning'. However, this again would need further verification and it is often useful to keep your data as unaltered as possible - allowing the software to actively do these deductions when needed.

Chapter 9

Previous Algorithms

Detecting potential twin laws has been considered many times over, and members of the crystallographic community have made many stand-alone programs. In this section, we will discuss the softwares COSET, ROTAX and Platon, followed by my implementation of ROTAX's method within Olex2.

9.1 Previous Software

We review the softwares COSET, ROTAX and Platon. COSET considers the lattice parameters to deduce (pseudo-)merohedral twin laws, whilst ROTAX and Platon perform more general searches for any viable twin laws.

9.1.1 COSET

COSET uses the theory of *coset decomposition* to find possible (pseudo-)merohedral twin laws. It thus only applies to integral twin laws. Such decomposition provides the possible twin laws up to the symmetry of the crystal itself (that is, without duplicates). [15] provides simple looping algorithms for deducing these representatives. [5] provides a command-line program implementing these algorithms, but still requires much user determination, such as a transformation to a reduced cell, and provides no independent tests for the potential usefulness of a twin law.

9.1.2 ROTAX

ROTAX [11] identifies 30 data points $\mathbf{h}$ with the largest values of $(|F_o(\mathbf{h})|^2 - |F_c(\mathbf{h})|^2)/\sigma(\mathbf{h})$. Then it transforms those $\mathbf{h}$ by choices of rotation matrices to determine which take these to vectors with integer coordinates.

ROTAX focuses in particular on non-merohedral twinning (including reticular merohedral twinning), with the knowledge that (pseudo-)merohedral twins are typically far more easily solved.

The algorithm ROTAX works as shown in the psuedocode in Algorithm 1.

I have implemented a version of ROTAX's algorithm which I refer to as the 'angle-axis' search and this is described in Section 9.2. Additionally, the measure of acceptance, known as the 'figure of merit', is one I also use, and describe in more detail in Subsection 10.4.1.

9.1.3 PLATON

It has been stated that Platon uses the 'd-spacing' to find likely twin laws. This is the same as points of the same distance from the origin, but the methodology is unspecified. Platon [54] describes its method as "Those [underestimated] reflections are overlapped by a strong reflection with (approximately) the same diffraction θ angle from a rotation-related twin-component lattice. Analysis of multiple such cases may lead to proposed candidates for the twinning operation at hand." The diffraction angle θ is what leads to the distance from the origin in the reciprocal lattice, so this works on the same principle as our methods. Unfortunately, there is no published literature including the details of the algorithm used in Platon.

Algorithm 1 Find Viable Rotations

```
Require: 30 $\mathbf{h}$ with largest $(|F_o(\mathbf{h})|^2 - |F_c(\mathbf{h})|^2)/\sigma(\mathbf{h})$
Require: reciprocal lattice vectors $\mathbf{k} = (h, k, l)$ with $h, k, l \in \{-12, \ldots, 12\}$ and
  $\gcd(h, k, l) = 1$
Require: rotation fraction $n$, typically 2
  $\theta \leftarrow 360°/n$
  for each $\mathbf{k}$ do
    $m = 1$
    while $m \leq n/2$ do
      $R \leftarrow$ rotation matrix about $\mathbf{k}$ with angle $m\theta$
      distances $\leftarrow$ NewList()
      for each underestimated $\mathbf{h}$ do
        $\mathbf{h}' \leftarrow R\mathbf{h}$ for each $\mathbf{h}$
        $d \leftarrow$ distance from $\mathbf{h}'$ to nearest lattice point
        AddItem(distances, $d$)
      end for
      average $\leftarrow$ mean(distances)
      omitted $\leftarrow 0$
      sort distances by size
      while average$> 0.002$Å AND omitted$< 15$ do
        remove largest value from distances
        average $\leftarrow$ mean(distances)
        omitted $\leftarrow$ omitted $+1$
      end while
      if average$> 0.002$Å then
        discard R
      else
        fom $\leftarrow$ average
        save R with its fom
      end if
      $m \leftarrow m + 1$
    end while
  end for
  Return all saved R
```

9.2 Implementing the Angle-Axis Search of ROTAX

When I began my PhD, Pascal Parois had begun working on implementing the ROTAX method within Olex2. ROTAX existed as independent software to Olex2 [11], and the task was to bring this algorithm into Olex2 to allow full use of Olex2's integrated systems. Based on his initial work, I continued with the implementation which was determined to be an appropriate task which would help me get an impression of how Olex2 was built.

To give it a more descriptive name, I call ROTAX's method the *angle-axis* search, as it iterates through angles and axes to find twin laws. Whilst ROTAX allows the user to input alternate values for some parameters, I also believe that some more complicated parts of the program could also allow variable inputs. ROTAX allows these variables to be altered:

- **Maximal axis coefficient n:** ROTAX defaults to a maximum axis coefficient for h, k, and l of $n = 12$ (that is, $|h|, |k|, |l| \leq n$).
- **Angle Fraction m:** ROTAX determines what angle $\theta = 360°/m$ to rotate around each axis. ROTAX recommends the choice of $m = 2$, but greater fractions are viable.

I think it is feasible that the following parameter values could be allowed to change with user input and different molecules, however the main ROTAX program does not do so:

- **Bad hkl:** ROTAX utilises a list of the 30 hkl with the most underestimated reflections according to $\left(|F_o(\mathbf{h})|^2 - |F_c(\mathbf{h})|^2\right)/\sigma(\mathbf{h})$. Whilst not implemented, one could employ larger or smaller lists (or ones limited to a particular discrepancy), or a different condition on underestimated reflections.

- **Overlap Threshold:** ROTAX does not have an explicit overlap threshold, but does use a value of 0.002Å in its figure of merit tests[1]. I consider this to be a value which could be altered, and in my implementation allow the user to input their own threshold ϵ.

We comment now on the way in which we store our list of bad hkl. We utilise a numpy array to store this information, which allows us to use matrix operations to act on all our hkl at once. We define a matrix, H_0, which contains in each row one hkl entry, that is:

$$H_0 = \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \vdots \\ \mathbf{h}_n^\top \end{bmatrix} = \begin{bmatrix} h_1 & k_1 & l_1 \\ h_2 & k_2 & l_2 \\ \vdots & \vdots & \vdots \\ h_N & k_N & l_N \end{bmatrix}.$$

In python, this also allows us to take specific rows, for example $H_0[i] = \begin{bmatrix} h_i & k_i & l_i \end{bmatrix}$. If we wish to multiply all entries by some matrix M (which could be R_{rel}, the orthogonalisation matrix A or any other possibility), we can simply post-multiply $H_M = H_0 M^\top$ to obtain the new matrix of transformed vectors

$$H_M = H_0 M^\top = \begin{bmatrix} (M\mathbf{h}_1)^\top \\ (M\mathbf{h}_2)^\top \\ \vdots \\ (M\mathbf{h}_N)^\top \end{bmatrix}.$$

Operating like this typically results in dramatically faster code than transforming each line one-by-one due to the ability to rely on inbuilt C-compiled numpy functions.

In Figure 9.1, we outline our implementation of this algorithm. The implementation takes as input an upper bound n for the integer triples describing the axes of rotation under consideration, an angle fraction m, and the set of 'bad hkl' $H_0 = H_{\text{bad}}$ (it also utilises the overlap threshold ϵ, but this is 'hidden' in the figure of merit and

[1] See 10.4.1 for the full details of the figure of merit. Due to its implementation, I expect that not all of the overlap distances considered in the average are actually within this threshold, meaning that the figure of merit accepts laws which overlap either less points with a tighter threshold, or more points with a looser threshold.

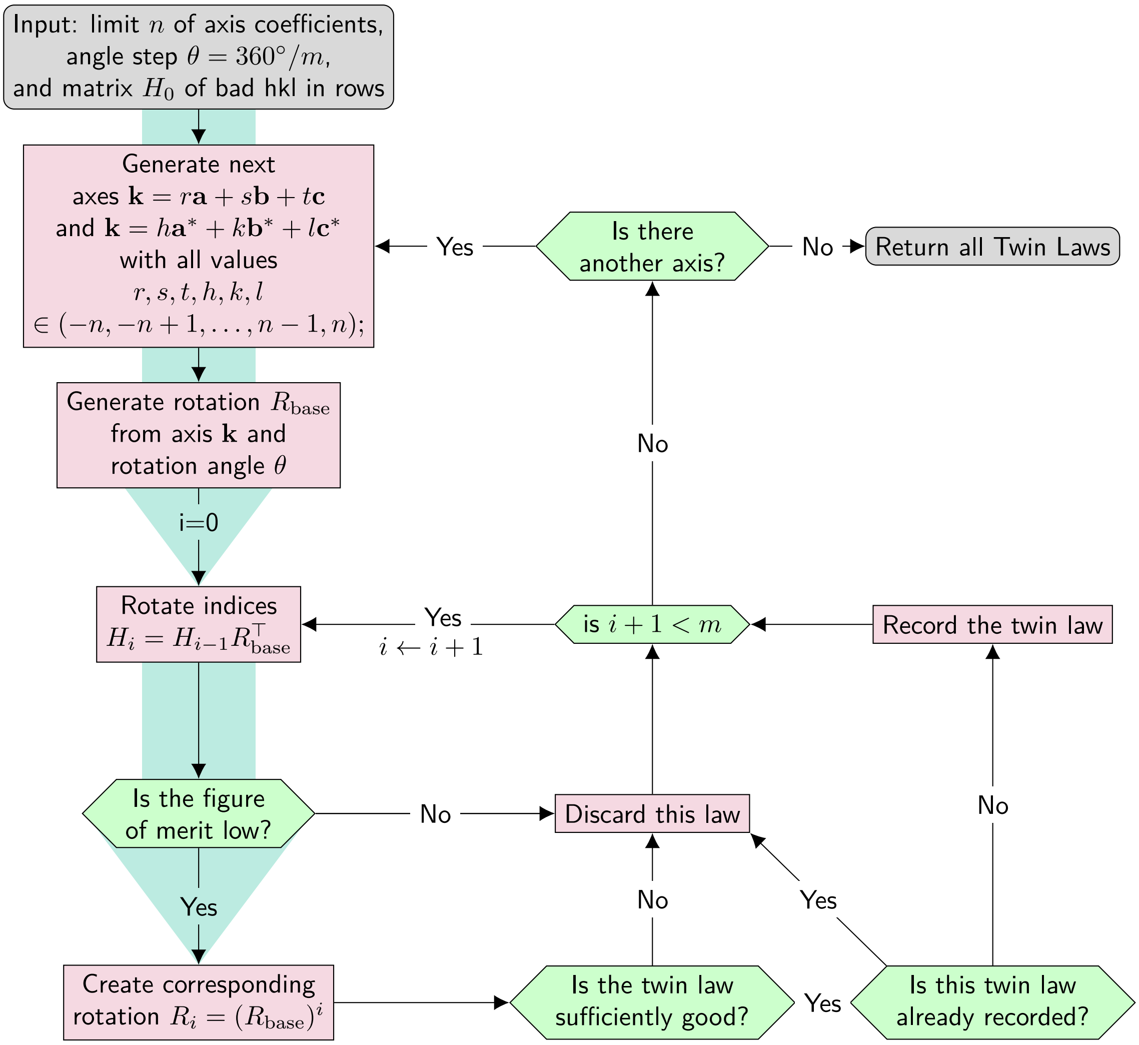

Input: limit n of axis coefficients, angle step $\theta = 360°/m$, and matrix H_0 of bad hkl in rows
Generate next axes $\mathbf{k} = r\mathbf{a} + s\mathbf{b} + t\mathbf{c}$ and $\mathbf{k} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$ with all values $r, s, t, h, k, l \in (-n, -n+1, \dots, n-1, n)$;
Yes
Is there another axis?
No
Return all Twin Laws
Generate rotation R_{base} from axis $\mathbf{k}$ and rotation angle θ
No
i=0
Rotate indices $H_i = H_{i-1} R_{\text{base}}^\top$
Yes
$i \leftarrow i+1$
is $i+1 < m$
Record the twin law
Is the figure of merit low?
No
Discard this law
No
No
Yes
Yes
Create corresponding rotation $R_i = (R_{\text{base}})^i$
Is the twin law sufficiently good?
Yes
Is this twin law already recorded?

Figure 9.1: Flow chart illustration of the angle-axis search

'goodness' tests). The angle fraction is used to generate the angle step $\theta = 360°/m$.

We then create viable axes based on both the direct lattice and reciprocal lattice, running through a list of axes with coefficients up to size n which is restricted by the symmetries of the crystal (as a symmetry equivalent axis gives a symmetry equivalent rotation). Then, we generate a one-step rotation R_{base} for the angle θ via the Rodriguez formula, given this axis and the angle step, for both the direct and reciprocal axes. We work on both axes simultaneously, acting the same on both.

We then repeatedly rotate the bad hkl R_{base}, that is we compute $H_i = H_{i-1}R_{\text{base}}^{\top}$, verify the figure of merit of H_i, and then generate the actual rotation matrix R_{base}^{i} if this figure of merit is sufficiently low.

We then test the twin law by our standard goodness tests as seen in Section 10.4, and if it passes, record it.

When we have iterated through all axes, we return any stored twin laws.

An increase of n or m will search more possibilities for rotations. Most twin laws can already be found by a twofold search $m = 2$. We typically take the list of bad hkl as the worst 30-50, but a more dynamic selection based on the precise values of the underestimation and total number of hkls could provide an improvement to the process.

Chapter 10

My Own Algorithm: Spherical Search

In this chapter, we present two algorithms (twofold twin law detection and general twin law detection) which I created to make most use of the ordered 'bad hkl' list to directly deduce very likely twin laws. We first present the initial idea behind the algorithms, followed by the special case restricted to twofold twins. Then we present the full algorithm with arbitrary rotations, and separately the algorithm for evaluating whether a twin law is sufficiently good, which applies to both. We follow this with a variety of mathematical ingredients required for these algorithms in full detail. Next, we present a discussion of various time optimisations and code improvements, adaptations to the algorithm or implementation to provide the fastest results. Finally, we present step-through examples for these algorithms deducing twin laws for specific datasets.

10.1 Fundamental Ideas Behind the Algorithm

My *spherical search* was inspired by one simple observation - that a point could only be rotated onto a point of the same distance from the origin. Given the list of most underestimated reflections (that is, those most likely to be subject to a twin overlap), I hypothesized that the list of points of similar distance would be relatively short, and thus an algorithm which took advantage of the knowledge of these very likely overlaps would be faster, removing the checking of many irrelevant possibilities.

Given a point $\mathbf{v} \in \Gamma^*$, I find all integer lattice points $\mathbf{v}_k \in \Gamma^*$ of similar distance, and denote their rescalings to the same size as $\mathbf{v}$ as $\mathbf{w}_k = \mathbf{v}_k|\mathbf{v}|/|\mathbf{v_k}|$ with $k = 0, \ldots, k_{\max}$ (see Section 10.6 for this process) (the scaling is necessary for the following mathematics to find the axis/angle to hold, but the 'overlapped' point is still considered the integer lattice point). With the two points $\mathbf{v}$ and $\mathbf{w}_k$, we obtain a *family* of possible rotations whose axes lie on the bisecting plane between $\mathbf{v}$ and $\mathbf{w}_k$. One such rotation is the *twofold* rotation corresponding to a rotation angle of π, around an axis pointing directly between the two points, $\mathbf{n}_1 = \mathbf{v} + \mathbf{w}_k$ (with associated unit vector $\hat{\mathbf{n}}_1 = \mathbf{n}_1/|\mathbf{n}_1|$). As twofold rotations are particularly common in crystallography, this gave rise to the first algorithm - the *twofold sphere search* (see Section 10.2).

This is not the only possible rotation. All axes lying on the plane perpendicular to the difference $\mathbf{v} - \mathbf{w}_k$ provide a rotation which will map $\mathbf{v}$ onto $\mathbf{w}_k$ for some to-be-determined angle θ. If we take a second such pair, which we name $\mathbf{p}$ and $\mathbf{q}_l$ (so long as these difference vectors $\mathbf{v} - \mathbf{w}_k$ and $\mathbf{p} - \mathbf{q}_l$ are not parallel), we find a unique axis $\mathbf{n} = (\mathbf{v} - \mathbf{w}_k) \times (\mathbf{p} - \mathbf{q}_l)$ capable of mapping both $\mathbf{v}$ onto $\mathbf{w}_k$ and $\mathbf{p}$ onto $\mathbf{q}_l$. In this case, we must additionally find the appropriate angle for each such map and only if these are the same (within some threshold) does this present a candidate for a viable twin law. This algorithm is expanded on in Section 10.3.

This allows a dramatically smaller number of possible rotation matrices to be tested whilst fully scrutinising the most relevant possibility space

10.2 Twofold Sphere Search

The twofold sphere search locates viable twofold laws (rotation by π), iterating through the points with underestimated intensity to find all laws which take the chosen point to any other lattice point. This is especially useful as twofold laws are by far the most common laws detected in crystallography, and very likely the most common way that twinned crystals grow. The algorithm is outlined in Figure 10.1.

Importantly, this search forgoes the need for two selected points $\mathbf{v}, \mathbf{p}$ by restricting the angle to 180°. We simply take each point with underestimated intensity $\mathbf{v}$ in turn, then generate all integer lattice points of a similar (up to a threshold, ϵ) distance (see Section 10.6 for details of how these are generated) and scale them to be of the same distance as $\mathbf{v}$[1]. We name these points $\mathbf{w}_k, k = 0, \ldots, k_{\max}$. We then iterate again through these points, taking the sum $\mathbf{n} = \mathbf{v} + \mathbf{w}_k$ as the axis of rotation. We must scale this to a unit vector $\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|}$, at which point we can use the Rodriguez Formula 8.2.2 to generate our rotation, with a slight simplification due to the prior knowledge that $\sin\theta = 0$ and $\cos\theta = -1$ for $\theta = 180°$, such that $R = I + 2K^2$.

We must then test the law, to evaluate whether or not it is effective, and in doing so find the basf which minimises the R-factor. This process is the same for all generation methods, and is expanded on in Section 10.4. If the twin law is effective, it is recorded. We then perform a check as to whether we have found the required amount of twin laws, as there is no need to continue searching if many laws are found quickly as they are most likely to be correct. If we have not found the required amount, the loops continue through other points $\mathbf{w}_k$ and $\mathbf{v}$ until sufficient laws are found or we reach the end of points of underestimated intensity to check, at which point it is likely there are no further 180° twin laws.

[1] This is excluded from the flowchart for simplicity, but scaling these integer points to very close non-integer points of the same size as $\mathbf{v}$ is necessary for the mathematical derivation of the axis to hold correctly and provide the error ϵ - if the scaling is ignored, the error from $R\mathbf{v}$ to $\mathbf{w}_k$ will be larger.

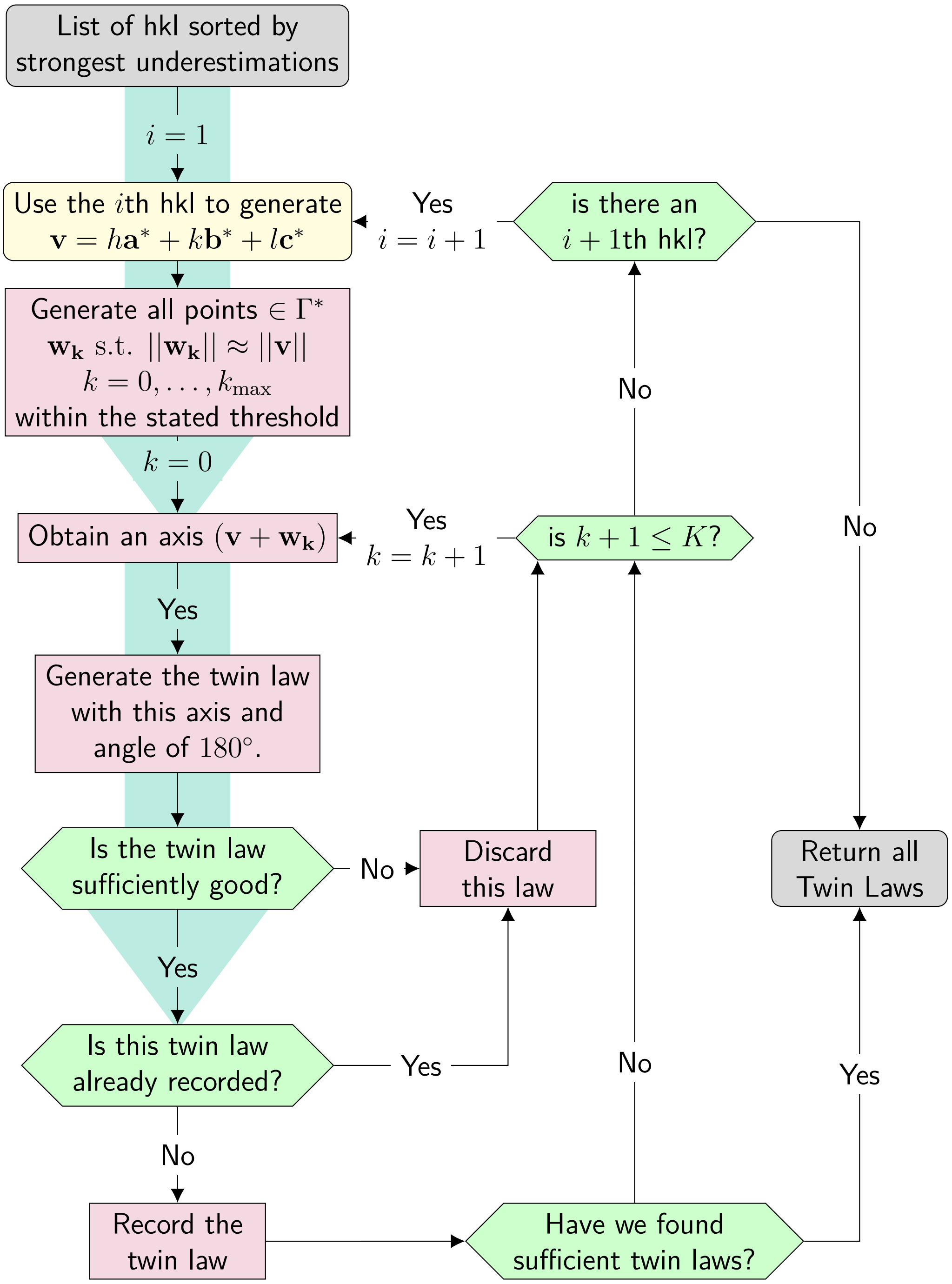

List of hkl sorted by strongest underestimations
$i = 1$
Use the ith hkl to generate $\mathbf{v} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$
Generate all points $\in \Gamma^*$ $\mathbf{w_k}$ s.t. $||\mathbf{w_k}|| \approx ||\mathbf{v}||$ $k = 0, \ldots, k_{\max}$ within the stated threshold
$k = 0$
Obtain an axis $(\mathbf{v} + \mathbf{w_k})$
Yes
Generate the twin law with this axis and angle of $180°$.
Is the twin law sufficiently good?
No
Discard this law
Yes
Is this twin law already recorded?
Yes
No
Record the twin law
Have we found sufficient twin laws?
No
Yes
Return all Twin Laws
is $k + 1 \leq K$?
Yes $k = k + 1$
No
is there an $i + 1$th hkl?
Yes $i = i + 1$
No

Figure 10.1: Flow chart illustration of the twofold spherical search

10.3 Full Sphere Search

The full sphere search locates viable twin laws by iterating over all pairs of points with underestimated intensity, and finding all axes which take both points to other lattice points. The algorithm is outlined in Figure 10.2.

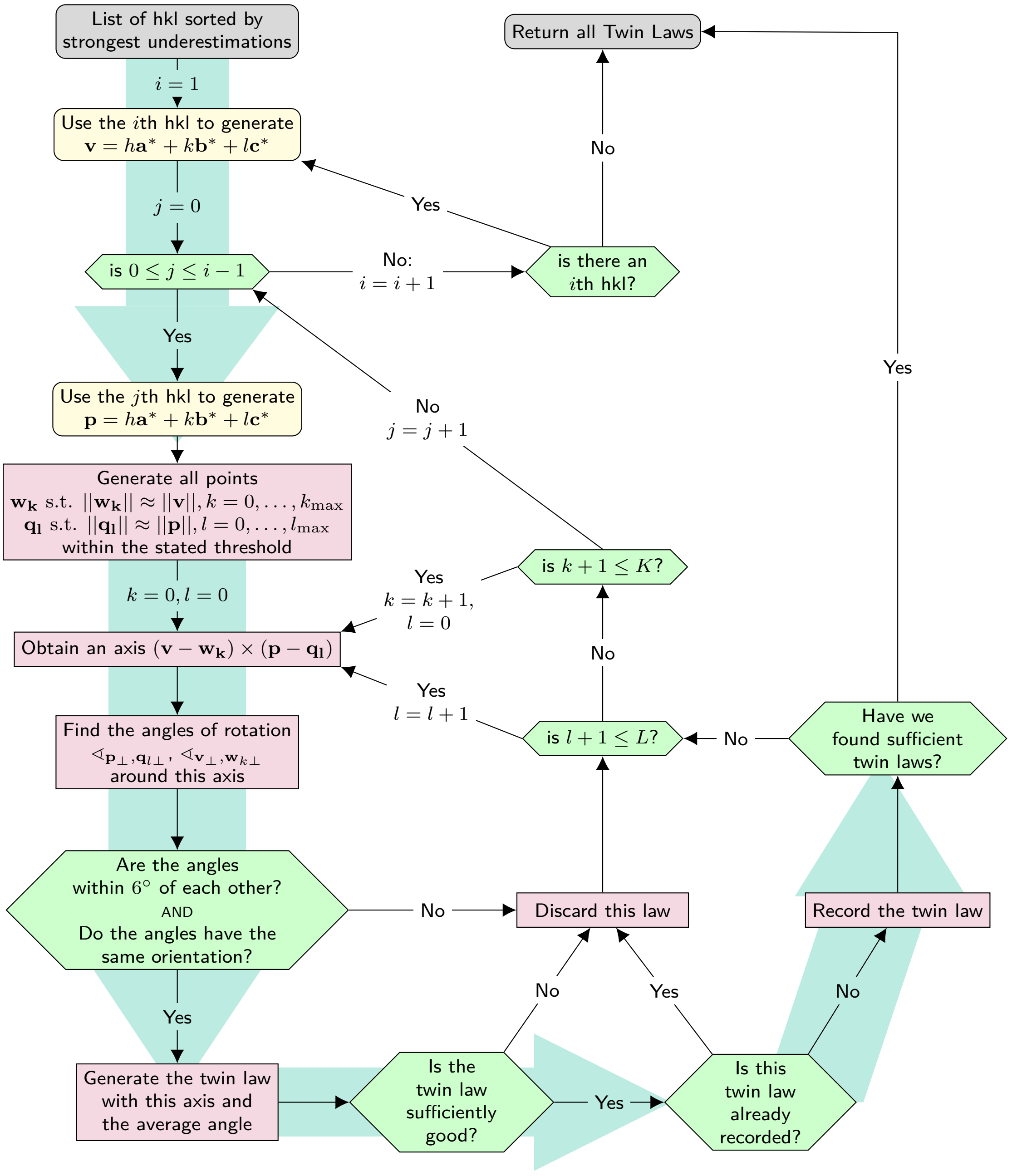

List of hkl sorted by strongest underestimations
$i = 1$
Use the ith hkl to generate $\mathbf{v} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$
$j = 0$
is $0 \leq j \leq i - 1$
No: $i = i + 1$
is there an ith hkl?
Yes
No
Return all Twin Laws
Yes
Use the jth hkl to generate $\mathbf{p} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$
No $j = j + 1$
Generate all points
$\mathbf{w_k}$ s.t. $||\mathbf{w_k}|| \approx ||\mathbf{v}||, k = 0, \ldots, k_{\max}$
$\mathbf{q_l}$ s.t. $||\mathbf{q_l}|| \approx ||\mathbf{p}||, l = 0, \ldots, l_{\max}$
within the stated threshold
$k = 0, l = 0$
is $k + 1 \leq K$?
Yes $k = k + 1,$ $l = 0$
Obtain an axis $(\mathbf{v} - \mathbf{w_k}) \times (\mathbf{p} - \mathbf{q_l})$
No
Yes $l = l + 1$
is $l + 1 \leq L$?
No
Have we found sufficient twin laws?
Yes
Find the angles of rotation $\sphericalangle_{\mathbf{p}_\perp, \mathbf{q}_{l\perp}}$, $\sphericalangle_{\mathbf{v}_\perp, \mathbf{w}_{k\perp}}$ around this axis
Are the angles within 6° of each other? AND Do the angles have the same orientation?
No
Discard this law
Record the twin law
Yes
No
Yes
No
Generate the twin law with this axis and the average angle
Is the twin law sufficiently good?
Yes
Is this twin law already recorded?

Figure 10.2: Flow chart illustration of the spherical search

To expand upon this diagram, we take these points of strongest underestimated intensity and use the indices i and j to enumerate them, with i always being larger than j. This allows us to effectively iterate through these pairs beginning with the pairs with strongest underestimation, which allows us to cut off the search early if it is effective.

We label this pair of points $\mathbf{v}$ and $\mathbf{p}$, and find all integer lattice points $\mathbf{w}_{k,0}, \mathbf{q}_{l,0}$ such that $|\mathbf{v} - \mathbf{w}_{k,0}| < \epsilon$ and $|\mathbf{p} - \mathbf{q}_{l,0}| < \epsilon$ (see Section 10.6 for the generation of these points). Then, we scale these found integer lattice points to be of equal length - that is, $\mathbf{w}_k = \mathbf{w}_{k,0}\frac{|\mathbf{v}|}{|\mathbf{w}_{k,0}|}$ and $\mathbf{q}_l = \mathbf{q}_{l,0}\frac{|\mathbf{p}|}{|\mathbf{q}_{l,0}|}$. This ensures that the generated rotations provide the correct error[2] of ϵ. These are then fully iterated through (described in the following paragraphs), looking for a twin law for each $\mathbf{v}, \mathbf{w}_k$ and $\mathbf{p}, \mathbf{q}_l$ pair.

There are many possibilities here which do not give rise to a twin law. If $\mathbf{v} - \mathbf{w}_k$ and $\mathbf{p} - \mathbf{q}_l$ point in the same direction, they describe the same family of axes and thus do not provide a unique axis and require a third point to find one - which is left to be done with the pairing of each of the two with this third point rather than duplicated for this instance.

The set of axis which supports a rotation from $\mathbf{v}$ to $\mathbf{w}_k$ lie in the plane perpendicular to $\mathbf{v} - \mathbf{w}_k$, whilst those supporting a rotation from from $\mathbf{p}$ to $\mathbf{q}_l$ lie in the plane perpendicular to $\mathbf{p} - \mathbf{q}_l$. Thus, the unique axis of rotation which supports both rotations can be found via $\mathbf{n} = (\mathbf{v} - \mathbf{w}_k) \times (\mathbf{p} - \mathbf{q}_l)$. We then take the unit axis, $\hat{\mathbf{n}} = \mathbf{n}/|\mathbf{n}|$.

Each vector $\mathbf{v}$ can be split into its portion parallel to the axis $\hat{\mathbf{n}}$ and perpendicular to that. We take the parallel portion as $\mathbf{v}_{\|} = (\mathbf{v} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$, and the perpendicular portion as $\mathbf{v}_{\perp} = \mathbf{v} - \mathbf{v}_{\|}$. It is the case that $\mathbf{w}_{k,\|} = \mathbf{v}_{\|}$, whilst $\mathbf{w}_{k,\perp} = \mathbf{w}_k - \mathbf{w}_{k,\|}$. The angle of rotation θ is the angle between the perpendicular components, which can be found via $\mathbf{v}_{\perp} \cdot \mathbf{w}_{k,\perp} = |\mathbf{v}_{\perp}||\mathbf{w}_{k,\perp}| \cos\theta$. Finally, this retrieved angle is without orientation, but one can give it an orientation by looking at the axis given by $\mathbf{v}_{\perp} \times \mathbf{w}_{k,\perp}$ and whether it lies in the same direction as $\hat{\mathbf{n}}$ or the opposite direction.

[2] In fact, using the mathematical derivation presented here on unscaled end points typically provides a greater error, caused by the derived rotation, due to the innacuracies then present in the process.

If the two pairs of points share a unique axis but have different angles, there is no twin law which works for both. If they share the axis and angle (with orientation), then we can generate the twin matrix describing this paired rotation. This can then be tested (see Section 10.4) to determine its usefulness.

10.4 Evaluating the Viability of Twin Laws

It is vitally important to have a quick way to evaluate the viability of twin laws - within the process, one can generate thousands of rotation matrices, and we need to filter that down to the most likely candidates. I use two main filters - the *figure of merit* as introduced in the *ROTAX* paper [11] to confirm that enough 'bad' points overlap, then *find basf* to check if the twin law is capable of lowering the R-factor.

In Figure 10.3, we describe the algorithm for the evaluation of a twin matrix R. Recall that the twin matrix $R = R_{\text{rel}}$ is written with respect to the lattice basis, not in pure Cartesian coordinates. The blue (and black) boxes are expanded on in their own subsections.

We begin with a twin matrix R (typically generated through one of the previous algorithms such as Spherical Search), and require a few additional matrices for our algorithm to function. Due to the nature of programming, we organise lists of n hkls $\mathbf{h}_i = \begin{bmatrix} h_i \\ k_i \\ l_i \end{bmatrix}$ as numpy arrays, which can be equivalently represented as matrices $H = \begin{bmatrix} h_1 & k_1 & l_1 \\ h_2 & k_2 & l_2 \\ \vdots & \vdots & \vdots \\ h_n & k_n & l_n \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1^\top \\ \mathbf{h}_2^\top \\ \vdots \\ \mathbf{h}_n^\top \end{bmatrix}$. This allows us to perform operations on all such hkl simultaneously, reducing the time cost of such operations. As a reminder, $HM = \begin{bmatrix} (M\mathbf{h}_1)^\top \\ (M\mathbf{h}_2)^\top \\ \vdots \\ (M\mathbf{h}_n)^\top \end{bmatrix}$. We begin with one such matrix, H_0, containing all hkl with

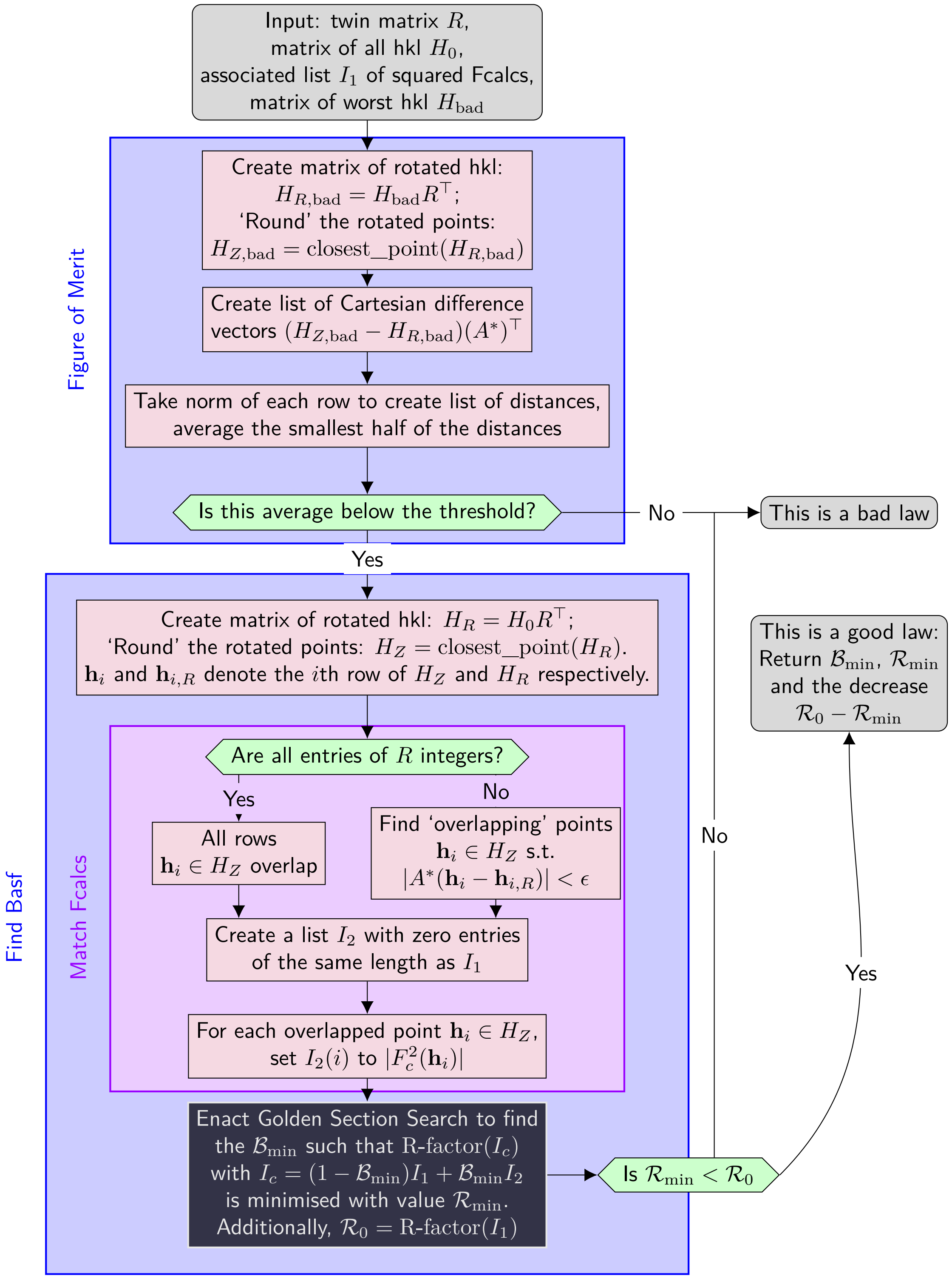

Input: twin matrix R, matrix of all hkl H_0, associated list I_1 of squared Fcalcs, matrix of worst hkl H_{bad}
Figure of Merit
Create matrix of rotated hkl: $H_{R,\text{bad}} = H_{\text{bad}} R^\top$; 'Round' the rotated points: $H_{Z,\text{bad}} = \text{closest\_point}(H_{R,\text{bad}})$
Create list of Cartesian difference vectors $(H_{Z,\text{bad}} - H_{R,\text{bad}})(A^*)^\top$
Take norm of each row to create list of distances, average the smallest half of the distances
Is this average below the threshold?
No
This is a bad law
Yes
Find Basf
Create matrix of rotated hkl: $H_R = H_0 R^\top$; 'Round' the rotated points: $H_Z = \text{closest\_point}(H_R)$. $\mathbf{h}_i$ and $\mathbf{h}_{i,R}$ denote the ith row of H_Z and H_R respectively.
Match Fcalcs
Are all entries of R integers?
Yes
No
All rows $\mathbf{h}_i \in H_Z$ overlap
Find 'overlapping' points $\mathbf{h}_i \in H_Z$ s.t. $|A^*(\mathbf{h}_i - \mathbf{h}_{i,R})| < \epsilon$
Create a list I_2 with zero entries of the same length as I_1
For each overlapped point $\mathbf{h}_i \in H_Z$, set $I_2(i)$ to $|F_c^2(\mathbf{h}_i)|$
Enact Golden Section Search to find the $\mathcal{B}_{\min}$ such that $\text{R-factor}(I_c)$ with $I_c = (1 - \mathcal{B}_{\min})I_1 + \mathcal{B}_{\min} I_2$ is minimised with value $\mathcal{R}_{\min}$. Additionally, $\mathcal{R}_0 = \text{R-factor}(I_1)$
Is $\mathcal{R}_{\min} < \mathcal{R}_0$
No
Yes
This is a good law: Return $\mathcal{B}_{\min}$, $\mathcal{R}_{\min}$ and the decrease $\mathcal{R}_0 - \mathcal{R}_{\min}$

Figure 10.3: Flow chart illustration of the evaluation of twin laws

recorded observations, alongside equal-height lists of the squared Fcalcs and observed intensities (squared fobs) associated to these hkl. We also have a matrix H_{bad} which contains the most underestimated hkls in order.

10.4.1 Figure of Merit

The *Figure of merit* is a value measuring the average distance between a rotated point and its nearest lattice point, outlined in the first blue box of Figure 10.3. The lower the figure of merit, the more likely it is that the twin law provides a viable rotation.

To calculate the figure of merit, we take our list of 'bad' hkl and apply the rotation matrix to them. Then, for each we obtain the distance from the rotated point to its nearest lattice point. Finally, we take the smallest half of these distances and average them - if this average is below the threshold of overlap, then it 'passes' the figure of merit test.

In essence, this test guarantees that a reasonable number of the most underestimated Fcalcs are accounted for by this rotation matrix.

10.4.2 R-factor Minimisation via Batch Scale Factor (basf)

A smaller R-factor indicates a closer fit to the recorded data. Once we have a twin orientation matrix R, there is a single parameter (in our case of only two components) which can impact the R-factor - the batch scale factor, $\mathcal{B}$. When testing a single twin law, we have two components only, so

$$I_c(\mathbf{h}, \mathcal{B}) = (1 - \mathcal{B}) I_{c,1}(\mathbf{h}) + \mathcal{B} I_{c,2}(R\mathbf{h}),$$

where $I_{c,1}$ and $I_{c,2}$ are the theoretical intensities corresponding to each component. We keep the label $\mathbf{h}$ from component 1 as the label for the combined intensity. We take $\mathcal{B} \in [0, 1)$ (where $\mathcal{B} = 0$ corresponds to the twin law not being present). As seen in Equation (8.2.1), $\mathcal{B}$ alters the calculated intensities I_c. We can then either try to optimise the R1-factor,

$$R_1 = \frac{\sum_{\mathbf{h}} \left| \left| \sqrt{\tilde{K}(\mathcal{B}) I_c(\mathbf{h}, \mathcal{B})} \right| - |F_o(\mathbf{h})| \right|}{\sum_{\mathbf{h}} |F_o(\mathbf{h})|}, \tag{10.4.1}$$

or the wR$_2$-factor,

$$\mathrm{wR}_2 = \sqrt{\frac{\sum_{\mathbf{h}} w(\mathbf{h}) |\tilde{K}(\mathcal{B}) I_c(\mathbf{h}, \mathcal{B}) - I_o(\mathbf{h})|^2}{\sum_{\mathbf{h}} w(\mathbf{h}) I_o^2(\mathbf{h})}}, \tag{10.4.2}$$

where $\mathbf{h} \in \Gamma^*$ are the labels of the recorded observations (note that in the case of a twin F_c is not available to use, and we substitute $|F_c| = \sqrt{I_c}$ in R_1). $\tilde{K}$ is defined in Section 6.5, and due to I_c depending on $\mathcal{B}$, so too does $\tilde{K}$. This dependency means that we cannot easily analytically calculate the minimum of wR$_2$ with respect to $\mathcal{B}$.

We then need to minimise the chosen R-factor with respect to the $\mathcal{B}$. As the R-factor is a sum over all reflections n, and in the case of the R_1-factor is a sum over many square roots of linear functions of the $\mathcal{B}$, we find this minimum numerically.

To do this, I employed the *golden section search*, a method which finds the minimum of a function by algorithmically decreasing the search range. Details on the intricacies of the golden section search can be found in [47, Section 10.2]. Other methods such as gradient descent could also work well, but to the precision of 10^{-2} needed for an initial estimate the golden section search takes only 3-6 steps to find a sufficiently precise minimum without the calculation time needed for the derivative.

In preparation for the golden section search, we must have the theoretical intensities $I_{c,2}$ corresponding to the overlapping point for each $\mathbf{h}$. This part of the algorithm underwent much time optimisation development which is expanded on in Section 10.7. We rotate the entire hkl list and find the closest point to each rotated point (see Section 10.5), with this integer list denoted H_Z.

If R is an integral matrix, we know each hkl is overlapped. If not, only some rows overlap, and all other rows have a corresponding $I_{c,2} = 0$. For all overlapping rows, we must take the integer rotated hkl and locate the theoretical intensity corresponding to that Miller index, which is assigned to the appropriate location in the list of intensities I_2.

The golden section search is then utilised to find a value for $\mathcal{B}$ which gives a minimal R-factor, $\mathcal{R}_{\text{min}}$. So long as this provides a reduction compared to the base R-factor $\mathcal{R}_0$, we determine that the twin law is viable, and return all relevant information. At this point, there are no further tests and twin laws will be presented to the user for inspection and potential use and refinement. Refinement can often confirm more strongly whether a twin law is useful.

10.4.3 Golden Section Search

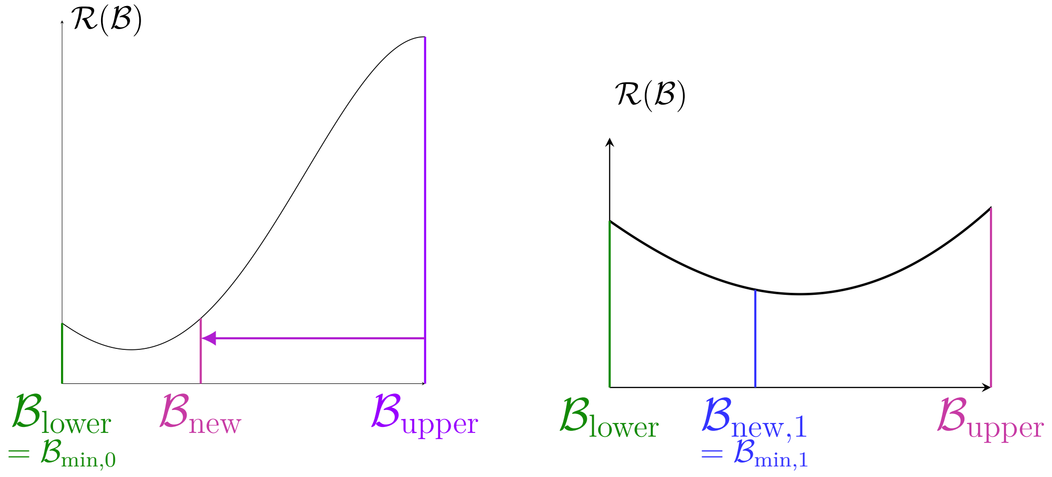
To begin, we will explain the general concept of the golden section search. The search is designed to effectively search for a minimum, narrowing the search space by a factor of approximately 0.618 ($2/(3+\sqrt{5}$ precisely) each step

The golden section search requires a function which is unimodal within the bounds provided to it. In our case, we assume only one local minimum which implies unimodality. If it were to have multiple, the search could enter a false minimum, or simply fail if it finds a point which provides a higher function value than the extremes.

The golden section search begins with two bounds within which to search - in our case, $\mathcal{B}_{\text{lower}} = 0$ and $\mathcal{B}_{\text{upper}} = 1$. These give function values $\mathcal{R}_{\text{lower}} = \text{R-factor}(\mathcal{B}_{\text{lower}})$ and $\mathcal{R}_{\text{upper}} = \text{R-factor}(\mathcal{B}_{\text{upper}})$.

Setup stage

Then it enters the 'setup' stage as shown in Figure 10.4, where we confirm that the function falls below $\mathcal{R}_{\text{lower}}$ and $\mathcal{R}_{\text{upper}}$ at some point in between $\mathcal{B}_{\text{lower}}$ and $\mathcal{B}_{\text{upper}}$. We generate $\mathcal{B}_{\text{new}} = \mathcal{B}_{\text{lower}} + \phi(\mathcal{B}_{\text{upper}} - \mathcal{B}_{\text{lower}})$ ($\phi = \frac{3+\sqrt{5}}{2}$ is a specifically chosen ratio


$\mathcal{R}(\mathcal{B})$
$\mathcal{B}_{\text{lower}}$
$= \mathcal{B}_{\text{min},0}$
$\mathcal{B}_{\text{new}}$
$\mathcal{B}_{\text{upper}}$
$\mathcal{R}(\mathcal{B})$
$\mathcal{B}_{\text{lower}}$
$\mathcal{B}_{\text{new},1}$
$= \mathcal{B}_{\text{min},1}$
$\mathcal{B}_{\text{upper}}$

(a) A 'setup stage' case where the new x is not the minimum of the 3 points, so the region is shrunk with the minimum still at the edge

(b) The end of the setup stage, once the new point is the smallest of the three and we can continue with the main minimum finding process

Figure 10.4: The setup stage of the golden section search.

closely related to the golden ratio to enable easy iteration, which is explained later). If at any point $\mathcal{R}_{\text{new}} := \text{R-factor}(\mathcal{B}_{\text{new}}) < \mathcal{R}_{\text{lower}}$ and $\mathcal{R}_{\text{new}} < \mathcal{R}_{\text{upper}}$, we proceed to the next stage with the triple $\mathcal{B}_{\text{lower}}, \mathcal{B}_{\text{upper}}$ and $\mathcal{B}_{\text{min}} := \mathcal{B}_{\text{new}}$. If not, we shorten the search to where it is currently minimal (that is, if $\mathcal{R}_{\text{lower}} \leq \mathcal{R}_{\text{upper}}$, $\mathcal{B}_{\text{upper}} = \mathcal{B}_{\text{new}}$, and similar for the opposite), and repeat this process.

Main Process

With the main process, illustrated in Figure 10.5, we begin with the triple $\mathcal{B}_{\text{lower}}, \mathcal{B}_{\text{upper}}$ and $\mathcal{B}_{\text{min}}$, which lies between $\mathcal{B}_{\text{lower}}$ and $\mathcal{B}_{\text{upper}}$ and has an associated $\mathcal{R}_{\text{min}} := \text{R-factor}(\mathcal{B}_{\text{min}}) < \mathcal{R}_{\text{lower}}$ and $\mathcal{R}_{\text{upper}}$. We then create a new $\mathcal{B}_{\text{new}} = \mathcal{B}_{\text{lower}} + \mathcal{B}_{\text{upper}} - \mathcal{B}_{\text{min}}$, and evaluate its R-factor, $\mathcal{R}_{\text{new}}$. Then, whichever of $\mathcal{R}_{\text{min}}$ and $\mathcal{R}_{\text{new}}$ is smaller determines the new centre point, which is again denoted $\mathcal{B}_{\text{min}}$, of the new triple, whilst the two points adjacent to it become the new $\mathcal{B}_{\text{lower}}$ and $\mathcal{B}_{\text{upper}}$. We repeat the process, calculating again $\mathcal{B}_{\text{new}} = \mathcal{B}_{\text{lower}} + \mathcal{B}_{\text{upper}} - \mathcal{B}_{\text{min}}$. This continues until we have reaches a sufficiently small difference between $\mathcal{B}_{\text{lower}}$ and $\mathcal{B}_{\text{upper}}$ to declare the minimum as the point between them.

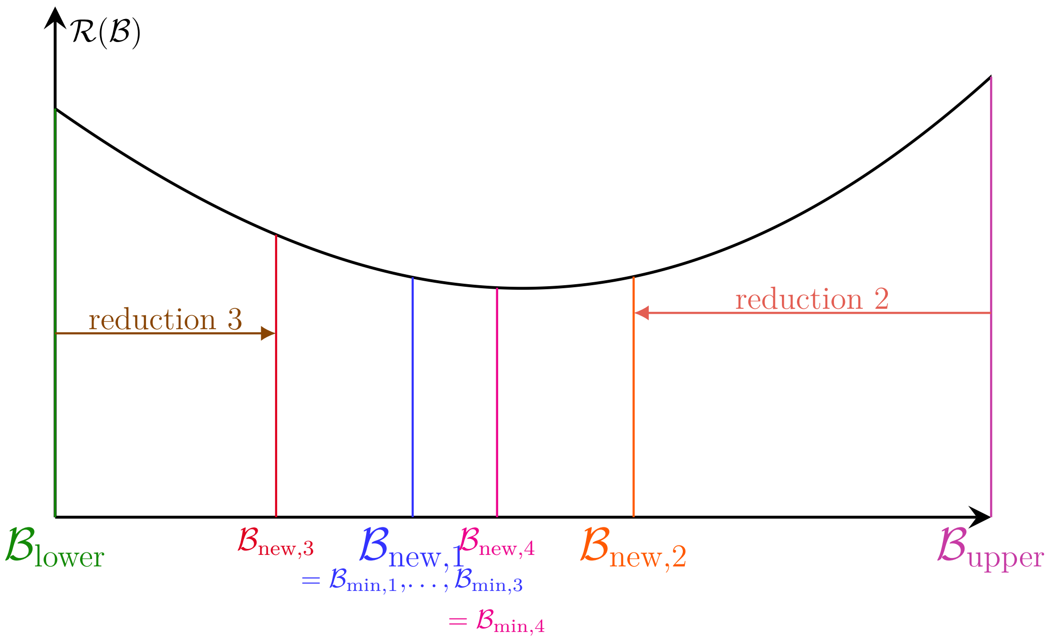

$\mathcal{R}(\mathcal{B})$
reduction 3
reduction 2
$\mathcal{B}_{\text{lower}}$
$\mathcal{B}_{\text{new},3}$
$\mathcal{B}_{\text{new},1}$
$= \mathcal{B}_{\text{min},1}, \ldots, \mathcal{B}_{\text{min},3}$
$\mathcal{B}_{\text{new},4}$
$= \mathcal{B}_{\text{min},4}$
$\mathcal{B}_{\text{new},2}$
$\mathcal{B}_{\text{upper}}$

Figure 10.5: Continuation of the golden section search, gradually reducing the search space to focus in on the minimum

Determining ϕ

The particular ratio ϕ employed in golden section search is specifically chosen to ensure that we descend in a consistent way no matter the results presented. We have two end points (in our case $\mathcal{B}_{\text{lower}}$ and $\mathcal{B}_{\text{upper}}$) and two 'middle' points (in our case $\mathcal{B}_{\text{min}}$ and $\mathcal{B}_{\text{new}}$) at any given time, but require that when the interval is shortened the ratios remain constant.

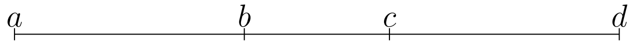


Figure 10.6: The distribution of points in the golden section search

In Figure 10.6 we show a set of points with a being the lowest and d being the largest. We want the first midpoint, b, to be calculated as a ratio of the distance between a and d, thus $b = a + \phi(d - a)$. c is the mirror of b, and can be found either as $c = d - \phi(d - a)$ or $c = a + d - b$. Next, we require that when the interval is shortened it keeps the same distribution of points (regarding their ratios).

In one case, where we shorten around b giving the minimum, we now have $[a, c]$ as the new interval, and b is one of the two midpoints (with the other being at $a + c - b$).

Let us assume that b is the larger midpoint, then we want $b = c - \phi(c-a)$, and this determines the value of ϕ. That is:

$$\begin{aligned}
& a + \phi(d-a) = b = c - \phi(c-a), \\
\Rightarrow\quad & a + \phi(d-a) = d - \phi(d-a) - \phi(d - \phi(d-a) - a) \quad \text{using } c = d - \phi(d-a), \\
\Rightarrow\quad & \phi(d-a) = d - a - \phi(d-a) - \phi(d-a) + \phi^2(d-a), \\
\Rightarrow\quad & \phi^2 - 3\phi + 1 = 0, \\
\Rightarrow\quad & \phi = \frac{3 \pm \sqrt{5}}{2}.
\end{aligned}$$

We can easily deduce that we must choose $\phi = \frac{3-\sqrt{5}}{2}$, as else this ratio would be greater than 1. This is also equal to $\frac{2}{3+\sqrt{5}}$. Additionally, for curiosity, the mirror of this ratio, $1 - \frac{3-\sqrt{5}}{2} = \frac{\sqrt{5}-1}{2} = \frac{2}{1+\sqrt{5}}$, which is the inverse of the golden ratio, giving the search its name.

We note also that during the setup stage, one can ‘jump’ to the next new point by combination of the present points rather than multiplication with ϕ. That is, if our section were to be reduced from $[a, d]$ to $[a, b]$, the new point found between a and b would be found at $e = 3b - a - d$. However, due to the likelihood of rounding errors within code, we recommend using the initial formula $e = a + \phi(b-a)$.

Flowchart 10.7 is a graphical illustration of the golden section search as used in our process.

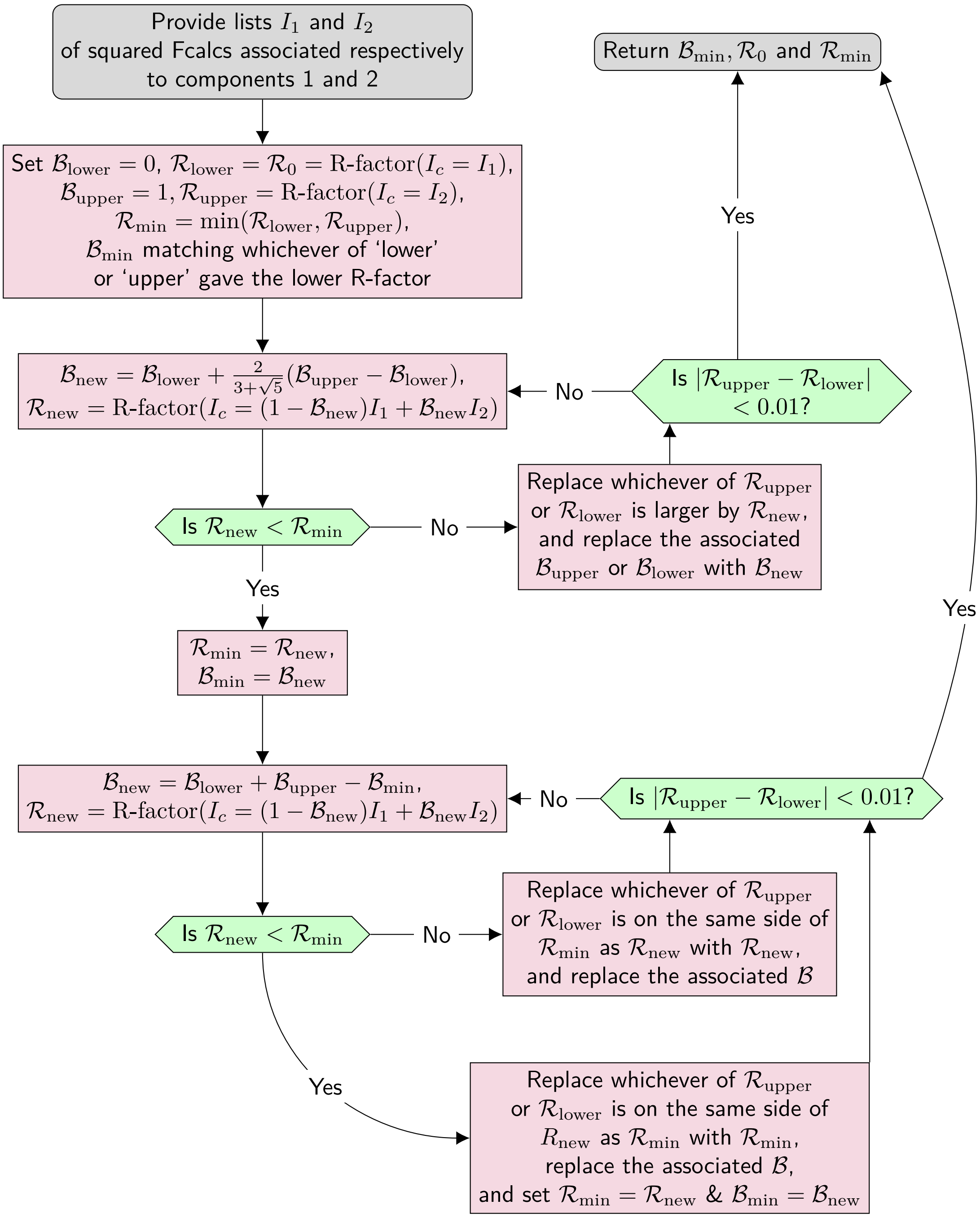

Provide lists I_1 and I_2 of squared Fcalcs associated respectively to components 1 and 2
Set $\mathcal{B}_{\text{lower}} = 0$, $\mathcal{R}_{\text{lower}} = \mathcal{R}_0 = \text{R-factor}(I_c = I_1)$, $\mathcal{B}_{\text{upper}} = 1, \mathcal{R}_{\text{upper}} = \text{R-factor}(I_c = I_2)$, $\mathcal{R}_{\text{min}} = \min(\mathcal{R}_{\text{lower}}, \mathcal{R}_{\text{upper}})$, $\mathcal{B}_{\text{min}}$ matching whichever of 'lower' or 'upper' gave the lower R-factor
$\mathcal{B}_{\text{new}} = \mathcal{B}_{\text{lower}} + \frac{2}{3+\sqrt{5}}(\mathcal{B}_{\text{upper}} - \mathcal{B}_{\text{lower}})$, $\mathcal{R}_{\text{new}} = \text{R-factor}(I_c = (1 - \mathcal{B}_{\text{new}})I_1 + \mathcal{B}_{\text{new}} I_2)$
Is $\mathcal{R}_{\text{new}} < \mathcal{R}_{\text{min}}$
No
Replace whichever of $\mathcal{R}_{\text{upper}}$ or $\mathcal{R}_{\text{lower}}$ is larger by $\mathcal{R}_{\text{new}}$, and replace the associated $\mathcal{B}_{\text{upper}}$ or $\mathcal{B}_{\text{lower}}$ with $\mathcal{B}_{\text{new}}$
Is $|\mathcal{R}_{\text{upper}} - \mathcal{R}_{\text{lower}}| < 0.01$?
No
Yes
Return $\mathcal{B}_{\text{min}}, \mathcal{R}_0$ and $\mathcal{R}_{\text{min}}$
Yes
$\mathcal{R}_{\text{min}} = \mathcal{R}_{\text{new}}$, $\mathcal{B}_{\text{min}} = \mathcal{B}_{\text{new}}$
$\mathcal{B}_{\text{new}} = \mathcal{B}_{\text{lower}} + \mathcal{B}_{\text{upper}} - \mathcal{B}_{\text{min}}$, $\mathcal{R}_{\text{new}} = \text{R-factor}(I_c = (1 - \mathcal{B}_{\text{new}})I_1 + \mathcal{B}_{\text{new}} I_2)$
No
Is $|\mathcal{R}_{\text{upper}} - \mathcal{R}_{\text{lower}}| < 0.01$?
Yes
Is $\mathcal{R}_{\text{new}} < \mathcal{R}_{\text{min}}$
No
Replace whichever of $\mathcal{R}_{\text{upper}}$ or $\mathcal{R}_{\text{lower}}$ is on the same side of $\mathcal{R}_{\text{min}}$ as $\mathcal{R}_{\text{new}}$ with $\mathcal{R}_{\text{new}}$, and replace the associated $\mathcal{B}$
Yes
Replace whichever of $\mathcal{R}_{\text{upper}}$ or $\mathcal{R}_{\text{lower}}$ is on the same side of R_{new} as $\mathcal{R}_{\text{min}}$ with $\mathcal{R}_{\text{min}}$, replace the associated $\mathcal{B}$, and set $\mathcal{R}_{\text{min}} = \mathcal{R}_{\text{new}}$ & $\mathcal{B}_{\text{min}} = \mathcal{B}_{\text{new}}$

Figure 10.7: Flow chart illustration of the golden section search

10.5 Closest Point Determination

Within this algorithm, we frequently must determine whether a point is sufficiently close to an integer lattice point - but to do so, we must know which integer lattice point to compare it to. Naively, we may simply round to the nearest integer lattice point, but this might not be the nearest point.

This rounding issue is discussed in Subsection 10.5.1, followed in Subsection 10.5.2 to find the shortest lattice vector, and expanded to closest lattice point in general in Subsection 10.5.3.

10.5.1 A Threshold for Rounding Providing the Closest Lattice Point

For example, we consider a system in two dimensions with $\mathbf{a}^* = (1, 0)$ and $\mathbf{b}^* = (0.5, 1)$. Each quadrant marked in Figure 10.8 will be rounded to its corresponding edge point. The marked point, $\mathbf{p} = (0.55, 0.25) = \frac{17}{40}\mathbf{a}^* + \frac{1}{4}\mathbf{b}^*$, using the coefficients $\frac{17}{40}, \frac{1}{4} < \frac{1}{2}$ would be rounded to $\mathbf{0} = 0\mathbf{a}^* + 0\mathbf{b}^*$, but has a distance of approximately 0.60 to $\mathbf{0}$ and a distance of approximately 0.51 to $\mathbf{a}^*$.

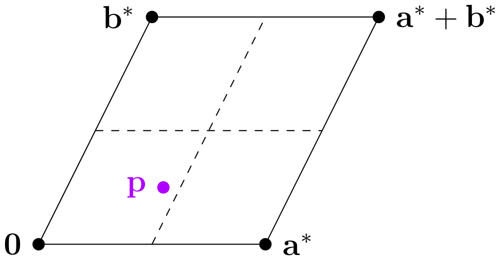


Figure 10.8: A lattice demonstrating that rounding can take you to a non-closest point.

It is simple to note that this specific chosen point $\mathbf{p}$ is quite far from either $\mathbf{0}$ or $\mathbf{a}^*$, and would likely not be considered 'close' to either. The natural question arises - for what Cartesian distance threshold, ϵ_c, can we round without worrying that this will not give us the nearest point?

Theorem 10.5.1. Any point $\mathbf{q}$ is rounded to the closest (measured in Cartesian space) integer lattice point $\mathbf{h}$ when the distance to that lattice point is less than ϵ_c, where[3]:

$$\epsilon_c < \frac{1}{2} \min\left(\frac{1}{|\mathbf{a}|}, \frac{1}{|\mathbf{b}|}, \frac{1}{|\mathbf{c}|}, r\right),$$

given

$$r = \min_{(h,k,l)\in\Gamma^*\setminus\{(0,0,0)\}} (h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*),$$

that is: any point $\mathbf{q}$ with $|\mathbf{q} - \mathbf{h}| \leq \epsilon_c$ will be rounded to $\mathbf{h}$, and all $\mathbf{q}$ rounded away from $\mathbf{h}$ have $|\mathbf{q} - \mathbf{h}| > \epsilon_c$.

Proof. Without loss of generality, we take $\mathbf{h} = \mathbf{0}$; points closest to a different integer lattice point are simply points closest to zero shifted by that integer lattice point.

Let $A = (-1/2, 1/2)\mathbf{a}^* + (-1/2, 1/2)\mathbf{b}^* + (-1/2, 1/2)\mathbf{c}^*$ be the domain of all points which would be rounded to zero. We define the $B_{\epsilon_c}(\mathbf{0})$ to be the open ball of radius ϵ_c around the origin. We wish to find $\epsilon_c > 0$ such that, for all $\mathbf{q} \in B_{\epsilon_c}(\mathbf{0})$:

1. $\mathbf{0}$ is the closest integer lattice point to $\mathbf{q}$
2. $\mathbf{q}$ is not within ϵ_c-distance of a non-zero integer lattice point in Γ^*.
3. $\mathbf{q} \in A$

To satisfy the first point, we simply need ϵ_c to be less than half the shortest distance between lattice points. We name this distance r, and calculate it as follows (the algorithm for its calculation is described in Subsection 10.5.2):

$$r = \min_{h,k,l\in\Gamma^*\setminus\{(0,0,0)\}} (h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*).$$

This radius r creates a separation of $\mathbb{R}^3$: $B_{r/2}(\mathbf{h})$ for all $\mathbf{h} \in \Gamma^*$ are disjoint. Given this, any $\mathbf{q}$ within one of these balls cannot be closer than $r/2$ to any other lattice point, satisfying points 1 and 2.

[3] we name ϵ_c for the closest-point-ϵ. Typically, our overlap threshold ϵ is directly compared to ϵ_c to decide whether we simply round all our indices to their nearest integer or perform a more complicated search.

However, a ball of radius $r/2$ may expand outside the parallelepiped of $\mathbf{q}$ rounded to $\mathbf{0}$ (equivalently, there may be points rounded to $\mathbf{0}$ which lie within other points' balls). In this case, we must reduce our radius such that it fits within this space - that is, we require

$$B_{\epsilon_c}(\mathbf{0}) \subset A \cap B_{r/2}(\mathbf{0}),$$

where as above $A = (-1/2, 1/2)\mathbf{a}^* + (-1/2, 1/2)\mathbf{b}^* + (-1/2, 1/2)\mathbf{c}^*$

This puts an upper limit on ϵ_c of $r/2$, if $B_{r/2}(\mathbf{0}) \subset A$. Otherwise, ϵ_c must equal the shortest distance from $\mathbf{0}$ to the facets of the parallelepiped A, and $B_{\epsilon_c}(\mathbf{0}) \subset B_{r/2}(\mathbf{0})$ trivially.

Thus, our final task is to determine the shortest distance between $\mathbf{0}$ and each facet of the parallelepiped. We note that due to symmetry of the shape, we need only consider half the parallelepiped. We thus consider the planes containing these facets,

$$\begin{aligned} P_a &= \frac{\mathbf{a}^* + \mathbf{b}^* + \mathbf{c}^*}{2} + \mathbb{R}\mathbf{b}^* + \mathbb{R}\mathbf{c}^*, \\ P_b &= \frac{\mathbf{a}^* + \mathbf{b}^* + \mathbf{c}^*}{2} + \mathbb{R}\mathbf{a}^* + \mathbb{R}\mathbf{c}^*, \\ P_c &= \frac{\mathbf{a}^* + \mathbf{b}^* + \mathbf{c}^*}{2} + \mathbb{R}\mathbf{a}^* + \mathbb{R}\mathbf{b}^*, \end{aligned}$$

where $\frac{\mathbf{a}^*+\mathbf{b}^*+\mathbf{c}^*}{2}$ is a vertex of the parallelepiped A.

As the normal of a plane is perpendicular to the vectors in the plane, the vector perpendicular to P_a is in the direction of $\mathbf{b}^* \times \mathbf{c}^*$, that is normalised to $\mathbf{a}/|\mathbf{a}|$, and similarly for P_b we obtain $\mathbf{b}/|\mathbf{b}|$ and P_c we obtain $\mathbf{c}/|\mathbf{c}|$. Then, we obtain the distance to the origin via the dot product with a point in the plane - of which $\frac{\mathbf{a}^*+\mathbf{b}^*+\mathbf{c}^*}{2}$ is for all three planes.

To demonstrate this for P_a (the other two follow in the same way),

$$\frac{\mathbf{a}^* + \mathbf{b}^* + \mathbf{c}^*}{2} \cdot \frac{\mathbf{a}}{|\mathbf{a}|} = \frac{1}{2|\mathbf{a}|}(\mathbf{a}^* \cdot \mathbf{a} + \mathbf{b}^* \cdot \mathbf{a} + \mathbf{c}^* \cdot \mathbf{a}) = \frac{1}{2|\mathbf{a}|}.$$

Thus the shortest distance to the boundary of the parallelepiped is $\frac{1}{2} \min\left(\frac{1}{|\mathbf{a}|}, \frac{1}{|\mathbf{b}|}, \frac{1}{|\mathbf{c}|}\right)$, and thus any $\epsilon \leq \epsilon_c = \min\left(\frac{1}{|\mathbf{a}|}, \frac{1}{|\mathbf{b}|}, \frac{1}{|\mathbf{c}|}, r\right)$ ensures that all points which are within

ϵ of an integer lattice point have all their indices within 0.5 of that integer lattice point.

□

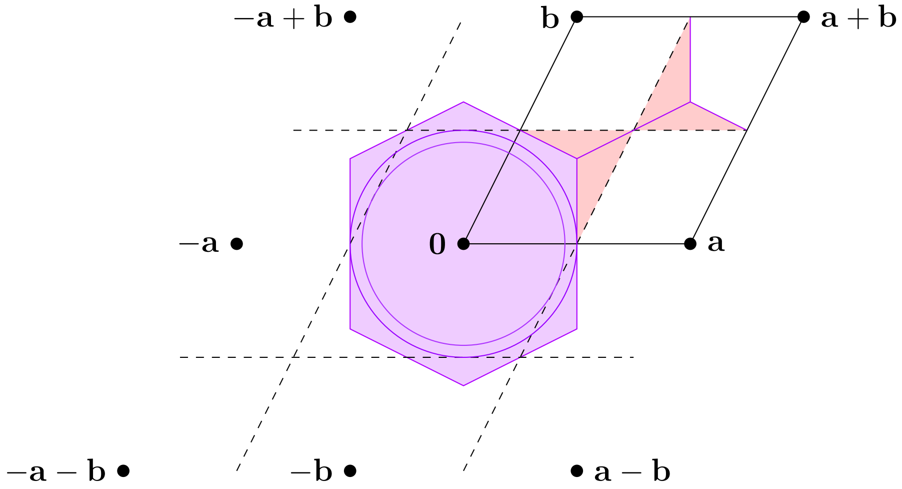


Figure 10.9: Representation of relevant areas within a lattice. The red zones show areas which will be rounded to a non-closest point within the unit cell $[0,1)\mathbf{a}+[0,1)\mathbf{b}$. The pink zone around $\mathbf{0}$ is its Dirichlet domain of all points which are closest to $\mathbf{0}$, the larger circle shows all points within $r/2$ to $\mathbf{0}$, whilst the inner circle shows the 'safe' ϵ_c-distance.

10.5.2 An Algorithm to Find the Shortest Non-trivial Lattice Vector

Recall that the length of the shortest non-trivial lattice vector is the radius r, which is defined over the whole infinite lattice Γ^*, and is not a terminating search if calculated naively brute-force. Outlined in [1] is an algorithm for solution of this 'shortest vector problem', which is conjectured to be an NP-hard problem (that is, it is currently believed that it cannot be solved for an arbitrary lattice in polynomial time). We adapt this algorithm for our specific 3-dimensional case with knowledge of the direct lattice $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$.

To begin, we will carefully define a series of terms which will allow us to efficiently find the closest lattice point. Firstly, we recall some basic facts about our lattice.

Our lattice, $\Gamma^* = \mathbb{Z}\mathbf{a}^* + \mathbb{Z}\mathbf{b}^* + \mathbb{Z}\mathbf{c}^*$. Its dual lattice, $\Gamma = \mathbb{Z}\mathbf{a} + \mathbb{Z}\mathbf{b} + \mathbb{Z}\mathbf{c}$. Additionally,

$$\begin{aligned}
V =&(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}, & V^* =&(\mathbf{a}^* \times \mathbf{b}^*) \cdot \mathbf{c}^*, & V =&\frac{1}{V^*},\\
\mathbf{a} =&\frac{1}{V^*}\mathbf{b}^* \times \mathbf{c}^*, & \mathbf{b} =&\frac{1}{V^*}\mathbf{c}^* \times \mathbf{a}^*, & \mathbf{c} =&\frac{1}{V^*}\mathbf{a}^* \times \mathbf{b}^*,\\
\mathbf{a}^* =&\frac{1}{V}\mathbf{b} \times \mathbf{c}, & \mathbf{b}^* =&\frac{1}{V}\mathbf{c} \times \mathbf{a} & \text{and } \mathbf{c}^* =&\frac{1}{V}\mathbf{a} \times \mathbf{b}.
\end{aligned}$$

In addition, due to the nature of their construction, the vectors $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ form an upper triangular matrix, whilst $\mathbf{a}^*, \mathbf{b}^*$ and $\mathbf{c}^*$ form a lower triangular matrix. That is,

$$\begin{aligned}
\mathbf{a} &= \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{b} &= \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix}, & \mathbf{c} &= \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix},\\
\mathbf{a}^* &= \begin{bmatrix} a_1^* \\ a_2^* \\ a_3^* \end{bmatrix}, & \mathbf{b}^* &= \begin{bmatrix} 0 \\ b_2^* \\ b_3^* \end{bmatrix}, & \text{and } \mathbf{c}^* &= \begin{bmatrix} 0 \\ 0 \\ c_3^* \end{bmatrix}.
\end{aligned}$$

This fact allows some quite convenient simplifications in our process!

Within our process, we make use of subspaces of $\mathbb{R}^3$ and the lattice points which lie within them. We define the plane $E(n) = n\mathbf{a}^* + \mathbb{R}\mathbf{b}^* + \mathbb{R}\mathbf{c}^*$ as the nth plane from the origin. Within this plane, we define the line $L(n, m) = n\mathbf{a}^* + m\mathbf{b}^* + \mathbb{R}\mathbf{c}^*$.

We require also the perpendicular vectors between adjacent n-planes and m-lines. For the perpendicular to the plane $E(n)$, we know that it is perpendicular to $\mathbf{b}^*$ and $\mathbf{c}^*$, and thus parallel to $\mathbf{a}$. The distance between the planes can be determined by $d_E := \mathbf{a}/|\mathbf{a}| \cdot \mathbf{a}^* = a_1^* > 0$, and we can name $\mathbf{u}_E := \begin{bmatrix} a_1^* \\ 0 \\ 0 \end{bmatrix}$, as the perpendicular vector between planes, illustrated in Figure 10.10. For the perpendicular to the line $L(n, m)$ within the plane $E(n)$, it must be perpendicular to $\mathbf{a}$ and $\mathbf{c}^*$ - which results in a vector parallel to $\begin{bmatrix}0\\1\\0\end{bmatrix}$. Similarly to before, we can find the distance between consecutive lines within the plane by taking $d_L := \begin{bmatrix}0\\1\\0\end{bmatrix} \cdot \mathbf{b}^* = b_2^*$, leading to our perpendicular vector between lines of $\mathbf{u}_L := \begin{bmatrix}0\\b_2^*\\0\end{bmatrix}$. For note, our third perpendicular vector to $\mathbf{u}_E$ and $\mathbf{u}_L$ is none other than $\mathbf{c}^*$.

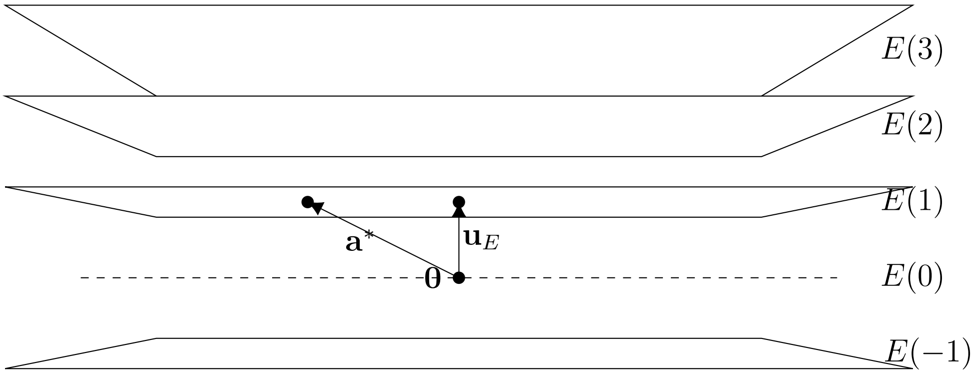


Figure 10.10: A perspective view of adjacent planes $E(n)$, and an emphasised position for $\mathbf{a}^*$ and $\mathbf{u}_E$

The general idea is as follows: We perform nested searches of subspaces of $\mathbb{R}^3$ to determine the shortest vector within those subspaces. Our search is bounded by never looking at a subspace which is, itself, further than the previous shortest distance from the origin.

As we progress, we keep a running record of the closest point to the origin, $\mathbf{h}_{\min} \in \Gamma^*$, along with its distance from the origin, $d_{\min}$.

The plane $E(n)$ closest to the origin is $E(0)$. Following this, the planes at $n = \pm 1$ are at a perpendicular distance a_1^* from the origin, those at $n = \pm 2$ are at a distance $2a_1^*$ from the origin, and so on. If at any point this distance is larger than $d_{\min}$, we need not consider planes of that n or larger-magnitude n as every point in the plane is at least na_1^* from the origin. That is, we need only check planes up to a value of $|n| \leq d_{\min}/a_1^*$, which will reduce each time we find a closer $\mathbf{h}_{\min}$.

Take then a plane $E(n)$. This has a perpendicular distance na_1^* from the origin, and this is found at the point $n\mathbf{u}_E$. We now, similar to before, seek the nearest lower-dimensional subspace to this point - in this case, some $L(n, m_0)$.

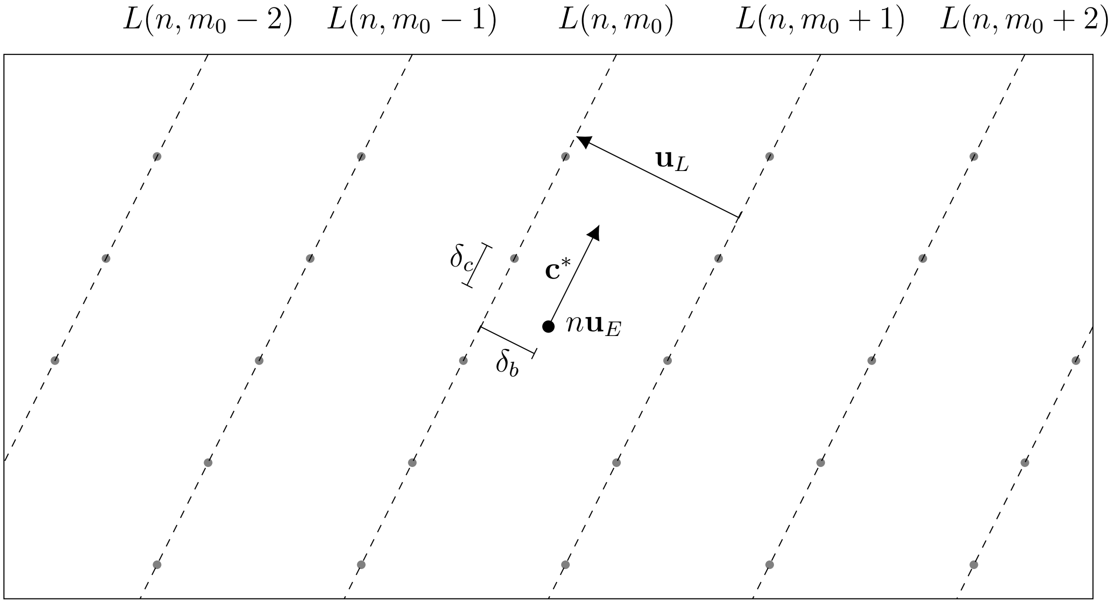


Figure 10.11: Lines within the plane E(n) for determination of shortest vector

We note that such a line must be within $\pm 0.5\mathbf{u}_L$ of $n\mathbf{u}_E$ (as $\mathbf{u}_L$ takes you from one line to the next). An illustration of this situation is presented in Figure 10.11. We take $\delta_b \in (-0.5, 0.5]$, and seek the point $n\mathbf{u}_E + \delta_b \mathbf{u}_L$ which lies on $L(n, m_0)$ for some $m_0 \in \mathbb{Z}$. We can equate these to find this point:

$$\begin{aligned} L(n, m_0) &= n\mathbf{u}_E + \delta_b \mathbf{u}_L + \mathbb{R}\mathbf{c}^*, & \delta_b &\in (-0.5, 0.5] \\ n\left(\mathbf{a}^* - \mathbf{u}_E\right) + m_0 \mathbf{b}^* + R\mathbf{c}^* &= \delta_b \mathbf{u}_L, & m_0 &\in \mathbb{Z}, R \in \mathbb{R} \\ n \begin{bmatrix} 0 \\ a_2^* \\ a_3^* \end{bmatrix} + m_0 \begin{bmatrix} 0 \\ b_2^* \\ b_3^* \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ c_3^* \end{bmatrix} &= \delta_b \begin{bmatrix} 0 \\ b_2^* \\ 0 \end{bmatrix}. \end{aligned}$$

We can immediately solve the second component for m_0 and δ_b:

$$na_2^* + m_0 b_2^* = \delta_b b_2^*,$$

$$\begin{aligned} b_2^*(m_0 - \delta_b) &= -na_2^*, \\ m_0 - \delta_b &= -\frac{na_2^*}{b_2^*}. \end{aligned}$$

As $m \in \mathbb{Z}$ whilst $\delta_b \in (-0.5, 0.5]$, we can solve this as follows:

$$m_0 = \left\lfloor -\frac{na_2^*}{b_2^*} \right\rceil, \qquad \delta_b = m_0 + \frac{na_2^*}{b_2^*},$$

where $\lfloor x \rceil$ is the closest integer to x, rounding halves up.

Thus the closest line to $n\mathbf{u}_E$ is $L(n, m_0)$, which lies at a distance of $\sqrt{(na_1^*)^2 + (\delta_b b_2^*)^2}$ from the origin. To find the next closest line, we take the sign of δ_b as s (if $\delta_b = 0$, we arbitrarily take $s = 1$) and then take $m \in \{m_0 - s, m_0 + s, m_0 - 2s, \dots\}$ in order and $m_u := m - m_0 + \delta_b$, leading to investigating the point $n\mathbf{u}_E + m_u\mathbf{u}_L \in L(n, m)$. As before, when $\sqrt{(na_1^*)^2 + (m_u b_2^*)^2} \geq d_{\min}$, we take no further lines within this plane.

Finally, given our point $n\mathbf{u}_E + m_u\mathbf{u}_L \in L(n, m)$, we need to find the nearest lattice point on that line to it. This is again illustrated in Figure 10.11. As before, we take $\delta_c \in (-0.5, 0.5]$ and solve where $n\mathbf{u}_E + m_u\mathbf{u}_L + \delta_c\mathbf{c}^*$ is an integer lattice point:

$$n\mathbf{a}^* + m\mathbf{b}^* + q\mathbf{c}^* = n\mathbf{u}_E + m_u\mathbf{u}_L + \delta_c\mathbf{c}^*$$

$$n\begin{bmatrix} 0 \\ a_2^* \\ a_3^* \end{bmatrix} + \begin{bmatrix} 0 \\ (m - m_u)b_2^* \\ mb_3^* \end{bmatrix} + (q - \delta_c)\begin{bmatrix} 0 \\ 0 \\ c_3^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Again, to solve for q and δ_c, we may simply take the third line:

$$\begin{aligned} na_3^* + mb_3^* + (q - \delta_c)c_3^* &= 0 \\ q - \delta_c &= -\frac{na_3^* + mb_3^*}{c_3^*}, \end{aligned}$$

and as before:

$$q = \left\lfloor -\frac{na_3^* + mb_3^*}{c_3^*} \right\rceil, \qquad \delta_c = q + \frac{na_3^* + mb_3^*}{c_3^*}.$$

At this point we are left with the closest point to zero within the line $L(n, m)$, $n\mathbf{a}^* + m\mathbf{b}^* + q\mathbf{c}^*$ with a distance of $\sqrt{(na_1^*)^2 + (m_u b_2^*)^2 + (\delta_c c_3^*)^2}$. If this is lower than the present $d_{\min}$, it replaces it and $\mathbf{h}_{\min} = n\mathbf{a}^* + m\mathbf{b}^* + q\mathbf{c}^*$. Either way, we then move on to the next m if viable, or if this would be too large, to the next n. If the next n is too large, we are done and can return our $\mathbf{h}_{\min}$ and $d_{\min}$.

Typically, we begin with $n = 0$, which leads to $m_0 = 0$, and thus also $q = 0$. However, we must exclude this origin point as a possibility (as then the distance would always be zero). Thus, for our first point we do not take this closest point within the line but instead take $q = 1$ and the point $\mathbf{h}_{\min} = \mathbf{c}^*$ as our initial shortest point with $d_{\min} = c_3^*$.

10.5.3 Finding the Closest Lattice Point to an Arbitrary Point

Similarly to this, we may need to find the closest lattice point to an arbitrary point $\mathbf{z}$ in $\mathbb{R}^3$ (if our ϵ_c rule does not apply). The adjustment of the above algorithm for this is very simple, instead of beginning with $n = 0$, we begin by finding the closest plane $E(n_0)$ to $\mathbf{z}$. Similarly to before, we simply need $n_0\mathbf{a}^* + R_1\mathbf{b}^* + R_2\mathbf{c}^* = \mathbf{z} + \delta_a \mathbf{u}_E$, with $\delta_a \in (-0.5, 0.5]$, and by the nature of $\mathbf{u}_E$ we use the first row for this,

$$n_0 - \delta_a = \frac{z_1}{a_1^*}.$$

Additionally, at any point our distance thusfar contains $n_u a_1^*$ rather than na_1^*, with $n_u := n - n_0 + \delta_a$. The calculations of q and m_0 have an additional factor of z_k in the numerator.The formulas for the relevant constants above are adjusted as follows:

$$\begin{aligned} n_0 &= \left\lfloor \frac{z_1}{a_1^*} \right\rceil, & \delta_a &= n_0 + \frac{-z_1}{a_1^*}, \\ m_0 &= \left\lfloor \frac{z_2 - na_2^*}{b_2^*} \right\rceil, & \delta_b &= m_0 + \frac{na_2^* - z_2}{b_2^*}, \\ q &= \left\lfloor \frac{z_3 - na_3^* - mb_3^*}{c_3^*} \right\rceil, & \delta_c &= q + \frac{na_3^* + mb_3^* - z_3}{c_3^*}. \end{aligned}$$

10.6 Finding Reciprocal Lattice Points on the Same Sphere about the Origin within ϵ Threshold

We must find reciprocal lattice points with similar distance to the origin $|A^*\mathbf{h}_{\text{base}}|$ for a given Miller triple $\mathbf{h}_{\text{base}} = [h, k, l]$ (in the above algorithms, both $\mathbf{v}$ and $\mathbf{p}$ are examples of a choice of $A^*\mathbf{h}_{\text{base}}$). We will discuss how to do this given one specific hkl input - with larger numbers of hkls to determine equivalent points for, it may result in more time efficiency to begin by creating a complete sorted list of Cartesian distances associated to Miller indices, and simply retrieving those closest. However, as we typically deal with a maximum of 10 hkls, we did not follow this implementation up and instead went for direct calculation of those within the same annulus of the origin as the input hkl.

We do this by finding points with the same Cartesian distance $|A^*\mathbf{h}|$ from the origin as $\mathbf{h}_{\text{base}}$ up to some threshold ϵ (we typically choose such ϵ to be 0.002Å^{-1}, but this can be varied to allow more or less tolerance of overlaps, but due to the nature of twins this must be smaller than the smallest distance between two lattice points). We denote the upper bound on this, $m_+ = |A^*\mathbf{h}_{\text{base}}|+\epsilon$ (and similarly, $m_- = |A^*\mathbf{h}_{\text{base}}|-\epsilon$). We then seek all lattice points $\mathbf{h}_0 = \begin{bmatrix} h_0 \\ k_0 \\ l_0 \end{bmatrix}$ such that $m_- \leq |A^*\mathbf{h}_0| \leq m_+$.

10.6.1 Outline

Before we begin, it is useful to review some basic notions and facts:

- We are working with $\mathbf{h} \in \mathbb{R}^3$ in relative coordinates in reciprocal space, with lattice vectors $\mathbf{a}^*, \mathbf{b}^*$ and $\mathbf{c}^*$.

- The reciprocal space orthogonalisation matrix $A^* = \begin{bmatrix} | & | & | \\ \mathbf{a}^* & \mathbf{b}^* & \mathbf{c}^* \\ | & | & | \end{bmatrix}$.
- The Cartesian norm of a relative vector $\mathbf{h}$ is $|A^*\mathbf{h}| = \sqrt{\mathbf{h}A^{*\top}A^*\mathbf{h}}$.
- The reciprocal metrical matrix $M^* = A^{*\top}A^* = \begin{bmatrix} \mathbf{a}^* \cdot \mathbf{a}^* & \mathbf{a}^* \cdot \mathbf{b}^* & \mathbf{a}^* \cdot \mathbf{c}^* \\ \mathbf{a}^* \cdot \mathbf{b}^* & \mathbf{b}^* \cdot \mathbf{b}^* & \mathbf{b}^* \cdot \mathbf{c}^* \\ \mathbf{a}^* \cdot \mathbf{c}^* & \mathbf{b}^* \cdot \mathbf{c}^* & \mathbf{c}^* \cdot \mathbf{c}^* \end{bmatrix}$.
- The inverse of the reciprocal orthogonalisation matrix is the transpose of the direct orthogonalisation matrix, $((A^*)^{-1})^\top = A = \begin{bmatrix} | & | & | \\ \mathbf{a} & \mathbf{b} & \mathbf{c} \\ | & | & | \end{bmatrix}$.
- The inverse of the reciprocal metrical matrix is the direct metrical matrix, $(M^*)^{-1} = M = A^\top A = \begin{bmatrix} \mathbf{a} \cdot \mathbf{a} & \mathbf{a} \cdot \mathbf{b} & \mathbf{a} \cdot \mathbf{c} \\ \mathbf{a} \cdot \mathbf{b} & \mathbf{b} \cdot \mathbf{b} & \mathbf{b} \cdot \mathbf{c} \\ \mathbf{a} \cdot \mathbf{c} & \mathbf{b} \cdot \mathbf{c} & \mathbf{c} \cdot \mathbf{c} \end{bmatrix}$.
- For a point $\mathbf{h}$ in reciprocal space, we refer to its elements as h, k and l respectively.

It is common to assume that the maximal possible value for h_0 is m_+/a^*, but this is not necessarily the case, as lattices without 90° angles can allow the decreasing of distance by adding other lattice vector directions. For example, with a 2-dimensional hexagonal basis $\mathbf{a}^* = (1,0), \mathbf{b}^* = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$, and the target function $f(h\mathbf{a}^* + k\mathbf{b}^*) = h$ under the condition $||h\mathbf{a}^* + k\mathbf{b}^*|| = 1$, $h = 1, k = 0$ would be expected to be the maximiser, but $\frac{2}{\sqrt{3}}\mathbf{a}^* - \frac{1}{\sqrt{3}}\mathbf{b}^* = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$ gives a larger maximum $f(h\mathbf{a}^* + k\mathbf{b}^*) = \frac{2}{\sqrt{3}} > 1$ whilst attaining the target distance of 1, due to the non-orthogonality. Thus we must consider the unique point which maximises h, which will not line up with taking the other variables as 0 unless the unit cell is orthogonal.

Our plan continues as follows:

- Every point of the sphere S_{m_+} of radius m_+ can be expressed via the basis $\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*$ as $h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$. Find the maximal value $h_{\max}$ for the coefficient h

which appears in this representation of all the points in S_{m_+}. Due to inversion symmetry (at the simplicity we work when considering twins), the minimal value for the coefficient h will equivalently be at $-h_{\max}$, and importantly we can ignore points where $h < 0$ as these will result in equivalent twin laws to the negation of that point.

- Choose each $h_0 \in [0, h_{\max}] \cap \mathbb{Z}$
- Given h_0, find the values k_- and k_+ which minimise and maximise k for that specific h_0 within the sphere S_{m_+}.
- Choose each $k_0 \in [k_-, k_+] \cap \mathbb{Z}$
- Given h_0, k_0, find the values $l_{-,\text{outer}}, l_{-,\text{inner}}, l_{+,\text{inner}}, l_{+,\text{outer}}$ bounding l inside S_{m_+} and outside S_{m_-}, which geometrically represents an annulus (it may be the case that all points $\begin{bmatrix} h_0 \\ k_0 \\ l \end{bmatrix}$ are outside S_{m_-}, in which case we only obtain $l_{-,\text{outer}}$ and $l_{+,\text{outer}}$) (see Figure 10.12 for a visual representation)
- Choose each $l_0 \in ([l_{-,\text{outer}}, l_{-,\text{inner}}] \cup [l_{+,\text{inner}}, l_{+,\text{outer}}]) \cap \mathbb{Z}$ (or, if $l_{-,\text{inner}}$ and l_L are not present, in $[l_{-,\text{outer}}, l_{+,\text{outer}}] \cap \mathbb{Z}$)
- Denote $\mathbf{h_0} = \begin{bmatrix} h_0 \\ k_0 \\ l_0 \end{bmatrix}$ and store it (along with its distance) into a list of viable lattice points.

10.6.2 Maximising h

Our first target is to find the maximal value for the h coefficient. That is, given the function $f(\mathbf{h}) = f(h, k, l) = h$, and the restriction $|A^*\mathbf{h}| = m_+$, we wish to find

$$h_{\max} = \max_{|A^*\mathbf{h}|=m_+} f(\mathbf{h}).$$

We begin by introducing a new variable $\mathbf{k} = A^*\mathbf{h}$, and expanding $f(\mathbf{h}) = h = \mathbf{h} \cdot \mathbf{e}_1$, which allows us to manipulate the maximisation into real space:

$$h_{\max} = \max_{|\mathbf{k}|=m_+} (A^{*-1}\mathbf{k}) \cdot \mathbf{e}_1.$$

Utilising the fact that $A^\top A^* = I$, we find

$$h_{\max} = \max_{|\mathbf{k}|=m_+} \mathbf{k}^\top A \mathbf{e}_1$$

which simplifies to:

$$h_{\max} = \max_{|\mathbf{k}|=m_+} \mathbf{k} \cdot \mathbf{a}.$$

The maximisation of this dot product is attained when $\mathbf{k}$ is parallel to $\mathbf{a}$ with the same direction. Additionally, recall that $|\mathbf{k}| = m_+$. Thus, this maximum is attained for

$$\mathbf{k} = m_+ \mathbf{a}/a$$

and the associated

$$h_{\max} = m_+ a. \tag{10.6.1}$$

10.6.3 Viable k Choices

Given an integer value $0 \leq h_0 \leq h_{max}$ as our choice for h, we must then find the range of viable k corresponding to points $h_0\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$ lying on the sphere $|A^*\mathbf{h}| = m_+$. The setting of $h = h_0$ defines the plane $P = h_0\mathbf{a}^* + \mathbb{R}\mathbf{b}^* + \mathbb{R}\mathbf{c}^*$, whose intersection with S_{m_+} defines a circle C[4], an illustration of which can be found in Figure 10.12. Our aim is to find $\max_{\mathbf{h} \in C} f(\mathbf{h})$ (and $\min_{\mathbf{h} \in C} f(\mathbf{h})$) where $f(\mathbf{h}) = f(h, k, l) = k$. The level sets of $f : \mathbf{h} \mapsto k$ in P are $f^{-1}(\{k\}) = h_0\mathbf{a}^* + k\mathbf{b}^* + \mathbb{R}\mathbf{c}^*$, that is, parallel lines in direction $\mathbf{c}^*$. We then need to find the intersection of C with the maximal level set, which will give us the maximal k. The intersection of C with a line lying within

[4] Assuming that such an intersection exists. However, as h_0 is chosen to be between zero and the maximal $h_{\max}$ allowing a $\mathbf{h}$ lying on the sphere of radius m_+, it is necessarily true that the intersection $S_{m_+}(\mathbf{0}) \cap P$ is non-empty. In the special case where the intersection is a single point, the point (and thus k and l) is fully determined.

P is either empty, 1 point (when tangent) or 2 points. Our maximal level set is found for one of these tangent lines (and our minimal for the other) - the dashed line to the left in Figure 10.12 shows a minimal level set.

We consider the vector $\mathbf{q} \in P$ to be the closest point of P to the origin, such that $\mathbf{q} \perp P$ and importantly, $\mathbf{q} \perp \mathbf{c}^*$. $\mathbf{q}$ is the centre of the circle C. We consider the radius vector $\mathbf{u}$ of the circle C pointing from $\mathbf{q}$ to $A^*\mathbf{h}$, which is again perpendicular to the tangent vector $\mathbf{c}^*$ - that is, $A^*\mathbf{h} = \mathbf{q}+\mathbf{u}$, where both $\mathbf{q}$ and $\mathbf{u}$ are perpendicular to $\mathbf{c}^*$, thus $A^*\mathbf{h} \perp \mathbf{c}^*$.

We then rely on the idea that, with h_0 fixed and $|A^*\mathbf{h}| = m_+$, we maximise (or minimise) $f(\mathbf{h})$ when our vector $A^*\mathbf{h}$ is perpendicular to $\mathbf{c}^*$.

For the vector $A^*\mathbf{h}$ perpendicular to $\mathbf{c}^*$, we know that $A^*\mathbf{h} \cdot \mathbf{c}^* = 0$:

$$A^*\mathbf{h} \cdot \mathbf{c}^* = h_0\mathbf{a}^* \cdot \mathbf{c}^* + k\mathbf{b}^* \cdot \mathbf{c}^* + l\mathbf{c}^* \cdot \mathbf{c}^* = 0. \tag{10.6.2}$$

This implies that

$$l = -\frac{h_0\mathbf{a}^* \cdot \mathbf{c}^* + k\mathbf{b}^* \cdot \mathbf{c}^*}{c^{*2}}. \tag{10.6.3}$$

We then use the norm equation:

$$|A^*\mathbf{h}|^2 = h_0^2a^{*2} + k^2b^{*2} + l^2c^{*2} + 2h_0k\mathbf{a}^* \cdot \mathbf{b}^* + 2h_0l\mathbf{a}^* \cdot \mathbf{c}^* + 2kl\mathbf{b}^* \cdot \mathbf{c}^* = m_+^2$$

and rewrite it as:

$$h_0^2a^{*2}+k^2b^{*2}+l(h_0\mathbf{a}^*\cdot\mathbf{c}^*+k\mathbf{b}^*\cdot\mathbf{c}^*+l\mathbf{c}^*\cdot\mathbf{c}^*)+2h_0k\mathbf{a}^*\cdot\mathbf{b}^*+l(h_0\mathbf{a}^*\cdot\mathbf{c}^*+k\mathbf{b}^*\cdot\mathbf{c}^*) = m_+^2. \tag{10.6.4}$$

Note that by (10.6.2), the third term vanishes when $\mathbf{h}$ maximises $f(\mathbf{h})$, and replacing l by (10.6.3), we obtain

$$h_0^2a^{*2} + k^2b^{*2} + 2h_0k\mathbf{a}^* \cdot \mathbf{b}^* - \frac{h_0\mathbf{a}^* \cdot \mathbf{c}^* + k\mathbf{b}^* \cdot \mathbf{c}^*}{c^{*2}}(h_0\mathbf{a}^* \cdot \mathbf{c}^* + k\mathbf{b}^* \cdot \mathbf{c}^*) = m_+^2.$$

We then have only the one variable k, in the form of a quadratic equation. First, we substitute in for each dot product its angular equivalent. That is, $\mathbf{a}^*\cdot\mathbf{b}^* = a^*b^*\cos\gamma^*$,

$\mathbf{a}^* \cdot \mathbf{c}^* = a^*c^* \cos\beta^*$ and $\mathbf{b}^* \cdot \mathbf{c}^* = b^*c^* \cos\alpha^*$:

$$h_0^2 a^{*2} + k^2 b^{*2} + 2h_0 k a^* b^* \cos\gamma^* \\ - \frac{h_0 a^* c^* \cos\beta^* + k b^* c^* \cos\alpha^*}{c^{*2}} (h_0 a^* c^* \cos\beta^* + k b^* c^* \cos\alpha^*) = m_+^2.$$

This easily allows a cancellation of c^*:

$$h_0^2 a^{*2} + k^2 b^{*2} + 2h_0 k a^* b^* \cos\gamma^* \\ - (h_0 a^* \cos\beta^* + k b^* \cos\alpha^*)(h_0 a^* \cos\beta^* + k b^* \cos\alpha^*) = m_+^2.$$

We can then expand this and rearrange to find

$$k^2 b^{*2}(1 - \cos^2\alpha^*) + 2k h_0 a^* b^* (\cos\gamma^* - \cos\beta^* \cos\alpha^*) = m_+^2 - h_0^2 a^{*2}(1 - \cos^2\beta^*),$$

which simplifies to

$$k^2 b^{*2} \sin^2\alpha^* + 2k h_0 a^* b^* (\cos\gamma^* - \cos\beta^* \cos\alpha^*) = m_+^2 - h_0^2 a^{*2} \sin^2\beta^*.$$

We can then complete the square to find

$$\left(k b^* + h_0 a^* \frac{(\cos\gamma^* - \cos\beta^* \cos\alpha^*)}{\sin^2\alpha^*} \right)^2 \\ = \frac{m_+^2 - h_0^2 a^{*2} \sin^2\beta^*}{\sin^2\alpha^*} + \frac{h_0^2 a^{*2} (\cos\gamma^* - \cos\beta^* \cos\alpha^*)^2}{\sin^4\alpha^*}$$

leading to

$$k b^* + h_0 a^* \frac{(\cos\gamma^* - \cos\beta^* \cos\alpha^*)}{\sin^2\alpha^*} \\ = \pm \frac{1}{\sin\alpha^*} \sqrt{m_+^2 - h_0^2 a^{*2} \sin^2\beta^* + \frac{h_0^2 a^{*2} (\cos\gamma^* - \cos\beta^* \cos\alpha^*)^2}{\sin^2\alpha^*}}$$

giving us solutions for k of

$$k = -h_0 a^* \frac{(\cos\gamma^* - \cos\beta^* \cos\alpha^*)}{b^* \sin^2\alpha^*} \pm \frac{1}{b^* \sin\alpha^*} \sqrt{\left(m_+^2 + h_0^2 a^{*2} \frac{\Theta}{\sin^2\alpha^*} \right)},$$

where $\Theta = \cos^2\gamma^* + \cos^2\alpha^* + \cos^2\beta^* - 1 - 2\cos\gamma^* \cos\beta^* \cos\alpha^*$.

Whilst this is by no means a pleasant equation, many of the present constants can be pre-calculated (in fact, everything excluding h_0 and m_+ is dependent only on the unit cell, not on the twin law under investigation, whilst m_+ is set by each $\mathbf{h}_{\text{base}}$ we compare to.). Then, the positive and negative values for k are our maximum k_+ and minimum k_- respectively.

That is, we can define

$$k_\pm = h_0 T \pm T' \sqrt{m_+^2 + h_0^2 T''} \tag{10.6.5}$$

given

$$\begin{aligned} T &= -a^* \frac{(\cos\gamma^* - \cos\beta^* \cos\alpha^*)}{b^* \sin^2\alpha^*}, \\ T' &= \frac{1}{b^* \sin\alpha^*}, \\ T'' &= a^{*2} \frac{\Theta}{\sin^2\alpha^*}. \end{aligned}$$

Then, we choose an integer k_0 such that $k_- \leq k_0 \leq k_+$, and are finally left with a direct calculation of l.

10.6.4 Viable l Choices

Having selected $0 \leq h_0 \leq h_{\max}$ and $k_- \leq k_0 \leq k_+$, we finally wish to calculate viable l. It is at this point that we fully introduce the tolerance ϵ. Previously, we determined the extreme values corresponding to $m_+ = |A^*\mathbf{h}_{\text{base}}| + \epsilon$, with the knowledge that any choices of h_0 and k_0 within these extremes would provide a line which certainly intersected with S_{m_+}. However, now we wish for

$$|A^*\mathbf{h}_{\text{base}}| - \epsilon \leq |A^*\mathbf{h}| = |h_0\mathbf{a}^* + k_0\mathbf{b}^* + l\mathbf{c}^*| \leq |A^*\mathbf{h}_{\text{base}}| + \epsilon, \tag{10.6.6}$$

that is, lying in the annulus bounded by S_{m_-} and S_{m_+} (a demonstration of which is presented within the purple circles in Figure 10.12). To determine the integers l satisfying (10.6.6), we first split the vector $\mathbf{h}$ into two vectors, $\mathbf{h} = \begin{bmatrix} h_0 \\ k_0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ l \end{bmatrix} =$

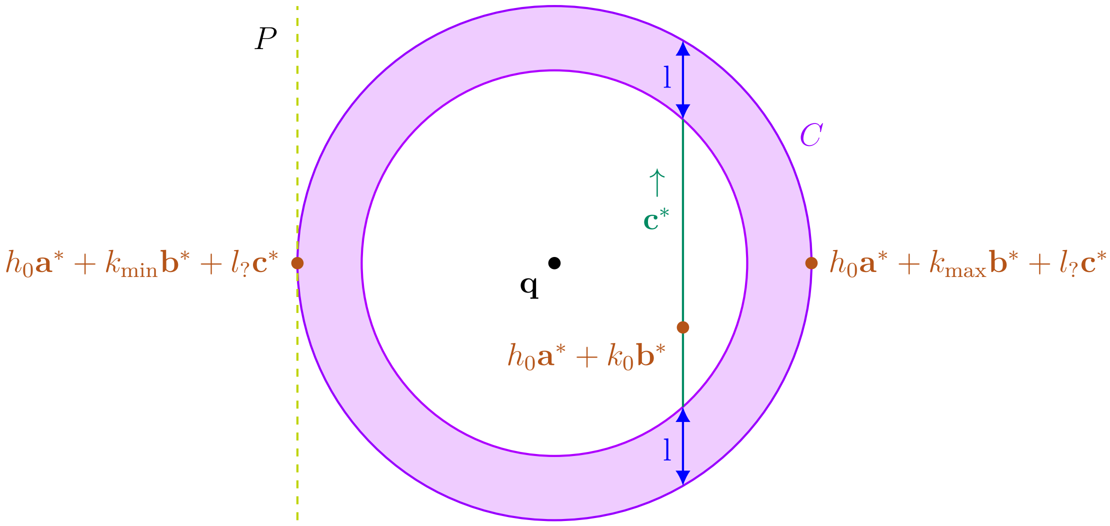

P
C
l
l
$\uparrow$
$\mathbf{c}^*$
$h_0\mathbf{a}^* + k_{\min}\mathbf{b}^* + l_?\mathbf{c}^*$
$h_0\mathbf{a}^* + k_{\max}\mathbf{b}^* + l_?\mathbf{c}^*$
$\mathbf{q}$
$h_0\mathbf{a}^* + k_0\mathbf{b}^*$

Figure 10.12: Diagram of the circle C in the plane P with respect to important deductions made within this algorithm. P is the plane of the page, C is the outer circle. $\mathbf{q}$ is the centre of the circle, and the circle is oriented such that $\mathbf{c}^*$ is vertical in the page. The tangent line to C in the direction of $\mathbf{c}^*$ is shown dashed on the left. Possibilities for l given a particular choice of k_0 are shown as the blue $\leftrightarrow$, and the points attaining the minima and maxima of k (with undetermined l) are shown on the left and right.

$\mathbf{k}_0 + l\mathbf{e}_3$, giving:

$$m_-^2 \leq (A^*(\mathbf{k}_0 + l\mathbf{e}_3)) \cdot (A^*(\mathbf{k}_0 + l\mathbf{e}_3)) \leq m_+^2,$$

which we can expand to obtain

$$m_-^2 \leq |A^*\mathbf{k}_0|^2 + l^2c^{*2} + 2(A^*\mathbf{k}_0) \cdot (l\mathbf{c}^*) \leq m_+^2. \tag{10.6.7}$$

Thus, again, we have a quadratic equation, this time in l. We will have to rearrange to solve this, but we must take special caution with regards to the inequalities.

Firstly, we will simply solve (10.6.7) at the equality points, and then determine whether l is bounded above or below by each. As we expect two solutions for each equality, we expect four bounds on l.

Now we look to solve the quadratic equation in l:

$$|A^*\mathbf{k}_0|^2 + l^2c^{*2} + 2(A^*\mathbf{k}_0) \cdot (l\mathbf{c}^*) = m_\pm^2.$$

As usual, we begin by rearranging and completing the square:

$$(l + (A^*\mathbf{k}_0) \cdot \mathbf{c}^*/c^{*2})^2 = \frac{m_\pm^2}{c^{*2}} - \frac{|A^*\mathbf{k}_0|^2}{c^{*2}} + ((A^*\mathbf{k}_0) \cdot \mathbf{c}^*/c^{*2})^2 =: B_\pm.$$

The right hand side is not necessarily non-negative in the m_- case (though, as before, it is non-negative in the m_+ case due to the bounds on the choices of h_0 and k_0). We define U, L and V as follows:

$$L = \sqrt{\max(0, B_-)} \qquad U = \sqrt{B_+} \qquad V = (A^*\mathbf{k}_0) \cdot \mathbf{c}^*/c^{*2}$$

This allows us to determine bounds on l.

$$l_{\pm,\text{outer}} = -V \pm U \tag{10.6.8}$$

and

$$l_{\pm,\text{inner}} = -V \pm L. \tag{10.6.9}$$

We allow all integers l in $[l_{-,\text{outer}}, l_{-,\text{inner}}] \cup [l_{+,\text{inner}}, l_{+,\text{outer}}]$. If L is 0, this reduces to l in $[l_{-,\text{outer}}, l_{+,\text{outer}}]$.

Unlike for the case of k, much of this is dependent on $\mathbf{k}_0$ and thus must be recalculated in each loop.

Each of these triples h_0, k_0, l_0 give a point of similar distance to our $\mathbf{h}_{\text{base}}$, a list of which is then returned.

10.6.5 Summary

To find lattice points with a similar ($\pm\epsilon$) distance from the origin to some reciprocal lattice point $\mathbf{h}_{\text{base}}$, we first take $m_\pm = |A^*\mathbf{h}_{\text{base}}| \pm \epsilon$.

Then, we iterate over a series of values. Firstly, $h_{\max} = m_+a$, and we iterate over all $h_0 \in [0, h_{\max}] \cap \mathbb{Z}$. Secondly, we find $k_\pm = h_0T \pm T'\sqrt{m_+^2 + h_0^2T''}$ where

$$T = -a^* \frac{(\cos\gamma^* - \cos\beta^* \cos\alpha^*)}{b^* \sin^2\alpha^*} \tag{10.6.10}$$

$$T' = \frac{1}{b^* \sin \alpha^*} \tag{10.6.11}$$

$$T'' = a^{*2} \frac{\cos^2 \gamma^* + \cos^2 \alpha^* + \cos^2 \beta^* - 1 - 2 \cos \gamma^* \cos \beta^* \cos \alpha^*}{\sin^2 \alpha^*}. \tag{10.6.12}$$

We then iterate over all $k_0 \in [k_-, k_+] \cap \mathbb{Z}$. We generate the bounds on l,

$$l_{\pm,\text{outer}} = -V \pm U \qquad l_{\pm,\text{inner}} = -V \pm L \tag{10.6.13}$$

where

$$L = \sqrt{\max \left(0, \frac{m_-^2}{c^{*2}} - \frac{|A^* \mathbf{k}_0|^2}{c^{*2}} + ((A^* \mathbf{k}_0) \cdot \mathbf{c}^* / c^{*2})^2 \right)} \tag{10.6.14}$$

$$U = \sqrt{\frac{m_+^2}{c^{*2}} - \frac{|A^* \mathbf{k}_0|^2}{c^{*2}} + ((A^* \mathbf{k}_0) \cdot \mathbf{c}^* / c^{*2})^2}, \tag{10.6.15}$$

$$V = (A^* \mathbf{k}_0) \cdot \mathbf{c}^* / c^{*2} \tag{10.6.16}$$

and take $l_0 \in ([l_{-,\text{outer}}, l_{-,\text{inner}}] \cup [l_{+,\text{inner}}, l_{+,\text{outer}}]) \cap \mathbb{Z}$.

Finally, we return all generated triples h_0, k_0, l_0.

10.7 Computational Optimisation and Time Analysis

In order to optimise the algorithms described above it became necessary to investigate various aspects of them and optimise them with regards to time cost. This required the careful consideration of the methods by which information was generated, stored and retrieved.

My primary source of information for time cost analysis was using python's cProfiler library [49] and the timeit module [50].

We focus on 3 particular substantial time improvements, followed by some minor improvements, and rejected ideas.

10.7.1 Theoretical Intensities as h, k, l-Array

This optimisation concerns the storage of theoretical intensities for quicker retrieval.

Original Implementation: The theoretical intensities are stored in a list Fsq. The corresponding hkls are stored in a matching list H_0, with the positions in the list directly relating the two. That is, $Fsq[i]$ is the theoretical intensity corresponding to the Miller triple $H_0[i]$.

Values of the theoretical intensity for a known hkl require searching H_0 for the matching entry, before taking that entry from the list Fsq.

Improved Implementation: The theoretical intensities are stored in a 3d array, where the hkl indices directly label each value $Fsq(h, k, l)$.

However, the indices hkl can also be negative integers. We solve this problem by using the fact that in python negative indices technically store data from the end of the array backwards. By using a sufficiently large array, we avoid overlap and negative indices work as expected.

Values of theoretical intensities for a known hkl require then taking the $[h, k, l]$ entry of the array.

Time Analysis: Table 10.1 presents the time taken for retrieval of theoretical intensities. Note that the function will typically be called 100-10,000 times (reflecting the number of overlapping hkls), as well as for multiple R-matrices, and the setup cost will occur once across all tests. Testing was performed on an example structure with only inversion symmetry and 4630 hkl entries, taking the median time over 20 trials of 100 repetitions of the setup and 10000 repetitions of the retrieval.

	Setup Cost/s	Per call/s
Original	0	8.46×10^{-5}
Improved	1.51×10^{-2}	5.55×10^{-6}

Table 10.1: Time taken for retrieval of theoretical intensities.

10.7.2 Expanding Outside the Generating Region

Due to crystal symmetries, $|F(\mathbf{h})| = |F(S\mathbf{h})|$ for all symmetries S of the crystal. Due to this, theoretical intensity data is only created for $\mathbf{h}$ within a specific generating region of the lattice Γ^*. We denote these triples within the standard matrix H_0. When rotated, H_Z (derived from $H_R = H_0R$) will contain many Miller triples $\mathbf{h}$ outside this generating region. Therefore a procedure is required to find the corresponding theoretical intensity inside the generating region. Two main ideas provide a solution - to map the rotated point using symmetry back into the generating region, or to expand the stored information from the generating region to cover more space.

Original Implementations: When searching for a known hkl $\mathbf{h}$, we first see if it exists within H_0. For the case where it does not, we implemented two separate ways of dealing with this:

(a) we call a function within cctbx called 'map_to_asu' which will find a triple in H_0 corresponding to$\mathbf{h}$,

and

(b) we try searching for $S\mathbf{h}$ within H_0 for all S symmetries in turn.

There was little time difference between the two implementations (a) and (b).

Improved Implementation: The storage of the theoretical intensities is altered at the beginning (within the setup of Fsq) to be $\#S$ times as large, containing $H_0S^\top$ for all symmetries S and their associated theoretical intensities. Note: the theoretical intensity associated to the ith entry of $H_0S^\top$ is the same as the theoretical intensity associated to the ith entry of H_0.

Time Analysis:

Table 10.2 presents time costs associated with this retrieval under two test functions for the original implementation b) and the improved implementation. These were performed with randomised data outside of the Olex2 environment[5]. Data presented is in seconds for a single run of the function. The setup function would be called once for the entire search, whilst the retrieval function would be called for each hkl (or each overlapping hkl) whenever needed, likely multiple times per search. The results presented are based on the median time to run the function 100 times for the setup function over 5 trials, whilst the retrieval functions are based on the median time to run the function 1000 times over 20 trials. We choose the median rather than minimum time taken in this case due to the random hkl indices taking a variable amount of retrieval time depending on which symmetry finds them in the b) case, and to keep consistent with likely system variance we use the same measure for the improved case.

The first case is a simple example with the only symmetry being the inversion symmetry. The second case includes one reflection and the inversion symmetry, and thus has 4 symmetry matrices (including the identity). The third case includes a quarter rotation and the inversion symmetry, and has 8 symmetry matrices. The test cases include 9260 hkl indices, including those related by symmetries (-10 to +10 for each index).

A basf determination would require a fraction of the hkls dependant on its overlap. For example, an index $n = 3$ twin law would require over 3000 calculations, each requiring one retrieval. Typically, one would expect 20 similar twin laws to be tested in this way, for an estimated 60,000 such retrievals.

The setup costs in the b) case are for converting the initial hkl list to a grid, whilst in the improved case it also needs multiply by symmetry matrices and add those into the grid. Thus, as the initial hkl lists are reduced by symmetries, b) has a smaller setup cost as we have more symmetries, whilst the improved method has similar times for all, increasing only slightly with more symmetries.

[5] The retrieval functions contain an element which randomises the hkl they are to retrieve, which takes 3.1×10^{-6}s to run, and this time has been removed from the function time.

Case	Function	Original b	Improved
Case 1	Setup	7.06×10^{-3}	1.49×10^{-2}
	Retrieval	4.43×10^{-6}	2.64×10^{-6}
Case 2	Setup	3.52×10^{-3}	1.76×10^{-2}
	Retrieval	5.06×10^{-6}	2.17×10^{-6}
Case 3	Setup	1.71×10^{-3}	1.7×10^{-2}
	Retrieval	5.89×10^{-6}	2.96×10^{-6}

Table 10.2: Time costs (in seconds) associated with retrieval methods b) and the improved method.

There is an additional unsolved issue, that the hkl can theoretically be rotated outside even this symmetry-made shape - it is generally not a sphere, so if we imagine a corner rotated to be in line with the centre of an edge, it would be outside of the region of hkl for which we have values. Ideally, in this case, we would recalculate a theoretical intensity for this hkl. However, at present for efficiency's sake we simply allow these to take an intensity of 0.

10.7.3 Filtering Rotated hkls

This improvement brings a substantial (10×) time improvement with a minor adjustment. We need to match up 'overlapping' points to their correct theoretical intensity, and this improvement concerns determining which points are overlapped.

Original Implementation: Within the *for* loop which assigns the theoretical intensities, we first have a check for if the current point's difference, $H_R[i] - H_Z[i]$, is sufficiently small by directly calculating the size of the difference. That is,

```
for i in range(len(H_R)):
  if np.linalg.norm(np.dot(metrical_inv,(H_Z[i]-H_R[i])))>ϵ:
    continue
  twin_sf[i]=hkl_grid[int(H_Z[i][0]),int(H_Z[i][1]),int(H_Z[i][2])]
```

Improved Implementation: Before the *for* loop which assigns the theoretical intensities, we utilise numpy's array functionality to calculate the size of the difference for each point simultaneously, and loop through only those indices which provide a sufficiently small difference:

```
if np.allclose(R,np.rint(R)):
  small_indices=[range(0,len(H_R)]
else:
  hkl_diff_cartesian=np.dot(metrical_inv,(H_Z-H_R).T).T
  hkl_comparison=np.linalg.norm(hkl_diff_cartesian,axis=1)
  small_indices= np.where(hkl_comparison<=cartesian_threshold)
for i in small_indices[0]:
  twin_sf[i]=hkl_grid[int(H_Z[i][0]),int(H_Z[i][1]),int(H_Z[i][2])]
```

This contains both an improvement via simultaneous calculation through the whole array using numpy's C-coded functions, and an improvement for shortening the *for* loop. The simultaneous calculation accounts for a vast majority of the time improvement, a third function with no shortening of the for loop but using the simultaneous calculation of distances (and then filtering within the for loop) is presented for completeness:

```
hkl_diff_cartesian=np.dot(metrical_inv,(H_Z-H_R).T).T
hkl_comparison=np.linalg.norm(hkl_diff_cartesian,axis=1)
for i in range(len(H_R)):
  if hkl_comparison[i]<cartesian_threshold:
    continue
  twin_sf[i]=hkl_grid[int(H_Z[i][0]),int(H_Z[i][1]),int(H_Z[i][2])]
```

we call this the '**Midway**' function in the following Table 10.3.

Time Analysis: We present tests with four test cases in Table 10.3. Case 1 is a simple twofold law with half of the points overlapping, case 2 a threefold law with half the points overlapping, case 3 is an integral law and case 4 is a twofold law with one in 5 points overlapping.

	Original	Improved	Midway
Case 1	67.84	5.931	6.554
Case 2	65.58	5.978	6.998
Case 3	74.92	8.478	3.542
Case 4	62.91	3.605	8.117

Table 10.3: Tests of the three filtering implementations. Times are presented in seconds to run 1000 tests, and were performed on purely theoretical data sets.

In all but the integral case, the improved implementation is the best of the three functions. It boasts a tenfold improvement for the 1-in-2 laws, and nearly 20-fold for the 1-in-5 law. It still brings an improvement of almost 10× even for the integral law.

10.7.4 Minor Improvements

Other optimisations were not quite as dramatic as those described above, or were quite simple. These are outlined in this subsection.

Size of a 3d Vector: We frequently need to determine the norm of a vector in $\mathbb{R}^3$. Whilst numpy contains a function (np.linalg.norm) for this, I experienced a twofold time reduction by creating a function which directly squared every entry, summed them and square rooted. I anticipate that this allowed such a time improvement due to not having to check or care about the entries passed to it. Of course, one must make sure it is only ever passed a 3d-vector, as due to the importance of time it contains no errors or checks and will simply take the first 3 entries of whatever is passed to it.

Golden Section Search: Initially, the finding of $\mathcal{B}$ was done by evaluating the

R-factor at $\mathcal{B} = 0, 0.01, 0.02, \ldots, 0.99, 1$ and comparing. With golden section search, it takes at most 10 steps to reduce the interval to below 0.01.

Closest Point: Unlike many others, this is a case of correctness rather than time. Initially, we only checked the 8 points ‘surrounding’ the rotated lattice point for which was the closest of them, unaware that it was possible for a ‘non-rounded’ lattice point to actually be the closest. Additionally, we would always do this - there was no check for the circumstance in which if a point was within ϵ of a lattice point, that lattice point was certainly closest to it. The algorithm outlined in Section 10.5 is the improvement, allowing a direct rounding of the points when ϵ is sufficiently small and engaging in the full algorithm when not.

10.7.5 Rejected Ideas

The realm of algorithmic optimisation is not one which always succeeds, but some attempts are particularly notable due to the reasons they failed to give the expected improvements, and these are outlined here.

Sufficient Overlaps: Considered as a replacement for find_fom (as discussed in Subsection 10.4.1), this function takes the size of each row of $H_{Z,\text{bad}} - H_{R,\text{bad}}$ in turn, and for any row which was not sufficiently close to a lattice point it would tick up a counter by one. If this counter passed a threshold of failure (typically being half of the points tested), the function would immediately return False. The principle behind this method was that it would be able to cutoff early compared to find_fom, and thus have a chance of terminating earlier and thus be quicker in general.

However, find_fom benefitted from the use of numpy arrays over for loops - we could calculate all row-norms simultaneously and average them, whilst sufficient_overlaps was implemented using a for loop. Due to this, it was dramatically slower, and was scrapped as a superfluous test.

The 'Short' basf Test: Under the belief that generating an R-factor and associated $\mathcal{B}$ for bad hkls only would:

- take dramatically less time than finding a $\mathcal{B}$ given all hkls,
- have a higher $\mathcal{B}$ due to containing the 'worst' offenders,

this function was intended to be a quicker test which would remove some 'bad' R more quickly than the full $\mathcal{B}$-estimate function.

However, this shorter function consistently underestimated $\mathcal{B}$ compared to the full function, plausibly significantly enough to discard a valid twin law. Additionally, with the speed improvements due to storing Fcalcs as a h, k, l-array, this function no longer had a dramatic improvement on the full function. Considering this, it was not worthwhile to use this test and it was discarded.

10.8 A Walk-Through Example

For this walk-through, we look at example 2 from [23]. We focus on the execution of the full spherical search 10.3, as this is able to find the twin law most directly. We calculate with a threshold of 0.001 Å.

The six points (h, k, l) with the most underestimated intensity are indexed by $(4, -2, -4)$, $(-4, 2, 4)$, $(1, 3, 0)$, $(5, -3, -5), (3, -1, -6)$, and $(0, 0, 3)$.

The first point, $(4, -2, -4)$, has a list of 18 viable rotation points: $(\pm_a 1, \pm_b 7, \mp_a 1)$, $(\pm_a 2, \pm_b 3, \pm_a 4), (\pm_a 3, \pm_b 5, \mp_a 3), (\pm_a 6, \pm_b 3, 0)$ and $(\pm_a 4 \pm_a 2 \mp_a 4)$[6] (note that a 180° rotation is not useful in this method, so $(-4, 2, 4)$ is excluded, as is the rotation to $4, -2, -4$ which would tell us nothing about the angle).

When we select $(4, -2, -4)$ as $\mathbf{v}$ and $(1, 3, 0)$ as $\mathbf{p}$ (as explained in Section 10.3), with $(4, -2, -4)$ mapped to $(6, 3, 0)$ and $(1, 3, 0)$ mapped to $(1, -3, 0)$, we obtain a

[6] The ± are associated if they share the same subscript and otherwise respectively free - that is, $(\pm_a 1, \pm_b 7, \mp_a 1)$ covers the 4 points $(1, 7, -1), (1, -7, -1), (-1, 7, 1)$ and $(-1, -7, 1)$

relative matrix of

$$R = \begin{bmatrix} 0.712 & 0.0935 & -0.833 \\ -0.148 & -0.952 & -0.428 \\ -0.575 & 0.187 & -0.664 \end{bmatrix}.$$

However, this rotation matrix returns a figure of merit of 0.0354Å, far above our 0.001Å threshold. Thus, we reject this matrix.

The first pair to pass the figure of merit test is the rotation mapping $(4, -2, -4)$ to $(-3, -5, 3)$ and $(1, 3, 0)$ to $(2, -2, 1)$, with a relative rotation lattice of

$$R = \begin{bmatrix} 0.5 & 0.5 & 1 \\ -0.5 & -0.5 & 1 \\ 0.5 & -0.5 & 0 \end{bmatrix}$$

and a figure of merit of 0.000047Å<0.0001Å. This has a $\mathcal{B}$ of 0.241 alongside an expected reduction in the R-factor by 0.94% from 8.55% to 7.61%

A pair with a stronger reduction is the rotation mapping $(4, -2, -4)$ to $(-1, 7, 1)$ and $(1, 2, 0)$ to $(-1, -1, 2)$, with a relative matrix of

$$R = \begin{bmatrix} 0.5 & -0.5 & 1 \\ 0.5 & -0.5 & -1 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

and a figure of merit of 0.000025Å<0.0001Å. This returns a basf $\mathcal{B}$ of 0.472, alongside an expected reduction in the R-factor by 2.67%, from 8.55% to 5.88%. Other pairs also return symmetry-equivalent rotation matrices with identical initial $\mathcal{B}$ estimates. When refined with the twin law applied, it reaches an R-factor of 5.20% and refines to a $\mathcal{B}$ of 0.502 (noting that *Olex2.refine* includes $\mathcal{B}$ as a refinement parameter).

10.9 Summary

Algorithms for the post-structure-determination detection of twin laws have typically been reliant on programs external to *olex2.refine* with either poorly described or simple brute-force algorithms. Through investigation of the nature of twinning, I sought to develop an algorithm which would make use of the information present to quickly detect the most likely twin laws without requiring human intervention on prediction of likely twinning angles, or similar details.

Key information for the detection of twin laws is the reciprocal lattice parameters along with a list of those reflections (indexed by h, k, l indices) which are most greatly overestimated by the current structure, that is $\mathbf{h}$ with greatest $(I_o(\mathbf{h}) - I_c(\mathbf{h}))/\sigma(\mathbf{h})$. By taking each of these 'bad' hkl in turn and generating the list of possible *hkl* indices which could have mapped onto them (those of a similar distance from the origin), we dramatically reduce the possibility space of viable rotations. Twofold sphere search 10.2 takes this reduced space and further limits it to 180° rotations (which are the most common kind). Conversely, full sphere search 10.3 takes two such 'bad' hkl together with a single possible *hkl* index (which could have mapped to it) for each which combine to provide a unique rotation.

Simply finding candidates is not all that is required - we must also evaluate their effectiveness, which is done in Section 10.4. Rotations are first discarded if they only result in overlap of very few of the points we expect to be affected, which is a relatively simple test. Following that, the rotation is tested (via estimation of proportions of components in the crystal) to see if it can provide an improvement to the R-factor at the current structure. If so, it represents a potentially viable twin law, and is presented to the user for human judgment and further refinement.

For these algorithms, I required knowledge on nature of three-dimensional lattices, including algorithms to effectively find the closest lattice points to a particular point (Section 10.5), and find points of a particular distance from the origin (Section 10.6).

Chapter 11

ESGI Meeting & Outlook

From 12th-19th April 2021 we presented the twinning problem to other mathematicians at ESGI (European Study Group with Industry) 165 in Durham. This lead to a variety of insights by the participants which I present in this chapter, and which are presented in more detail in our report [8]. We also present my hopes for future developments related to the twinning problem.

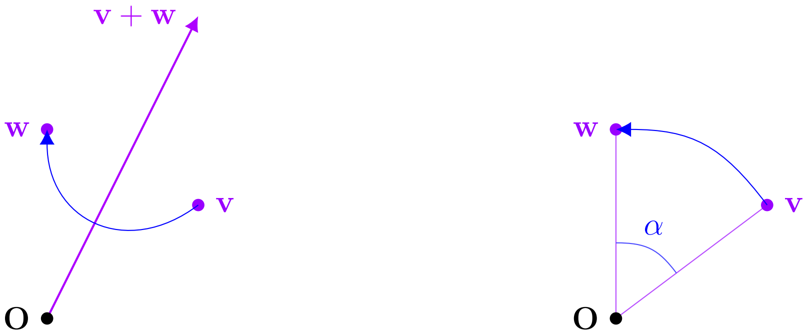
11.1 Representations of Rotations

To begin, we discuss the possible rotations mapping some **v** to some same-size **w** as described in 10.1. In our usage, **w** is matched in norm to **v**, and may no longer be an integer lattice point but would be derived from one. However, these discussions work for general $|\mathbf{v}| = |\mathbf{w}|$.

11.1.1 Relation Between Axis and Angle, Based on Extreme Axis and Angles

To begin, we look at the viable rotations given two points **v** and **w**, and how the angle of rotation is related to the chosen axis when mapping **v** to **w**.

Let us take two points, $\mathbf{v}$ and $\mathbf{w}$, such that $|\mathbf{v}| = |\mathbf{w}|$ with $\mathbf{v} \neq \pm\mathbf{w}$[1]. We first observe the plane containing $\mathbf{v}, \mathbf{w}$, and the origin, illustrated in Figure 11.1a.



(a) An in-plane illustration of the points $\mathbf{v}$ and $\mathbf{w}$ and the 180° rotation provided by the axis $\mathbf{n}_1 = \mathbf{v} + \mathbf{w}$.

(b) An in-plane illustration of the points $\mathbf{v}$ and $\mathbf{w}$ and the α rotation provided by the axis $\mathbf{n}_2 = \mathbf{v} \times \mathbf{w}$ out of the page.

The line equidistant to both $\mathbf{v}$ and $\mathbf{w}$ within this plane is $\mathbf{v} + \mathbf{w}$. Rotating about this line $\mathbf{v} + \mathbf{w}$ in 3-dimensional space by 180° will map $\mathbf{v}$ onto $\mathbf{w}$ (and additionally $\mathbf{w}$ onto $\mathbf{v}$). We label this vector $\mathbf{n}_1 = \mathbf{v} + \mathbf{w}$, and its associated unit axis

$$\hat{\mathbf{n}}_1 = \frac{\mathbf{n}_1}{|\mathbf{n}_1|} = \frac{\mathbf{v}+\mathbf{w}}{|\mathbf{v}+\mathbf{w}|}. \tag{11.1.1}$$

The other extreme is an axis out of the page, perpendicular to $\mathbf{v}$ and $\mathbf{w}$. This allows a rotation with an angle equal to the angle between $\mathbf{v}$ and $\mathbf{w}$, which we name $\alpha = \arccos \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} \in [0, \pi]$, in the plane. We can generate this axis as $\mathbf{n}_2 = \mathbf{v} \times \mathbf{w}$, with its unit axis similarly being

$$\hat{\mathbf{n}}_2 = \frac{\mathbf{n}_2}{|\mathbf{n}_2|} = \frac{\mathbf{v}\times\mathbf{w}}{|\mathbf{v}\times\mathbf{w}|}. \tag{11.1.2}$$

This rotation can be seen in Figure 11.1b.

These are not the only viable axes, though. In fact, any linear combination of the two axes will remain able to rotate $\mathbf{v}$ onto $\mathbf{w}$ (as $\mathbf{v}$ and $\mathbf{w}$ will remain equidistant

[1] Whilst the unique case $\mathbf{w} = -\mathbf{v}$ allows solutions with any axis perpendicular to $\mathbf{v}$ and an angle of π, its circle cannot be described in the same manner as other pairs and so it is excluded from this set.

from these axes). Taking the unit axes, we can generate new unit axes

$$\hat{\mathbf{n}}(t) = \sin t \hat{\mathbf{n}}_1 + \cos t \hat{\mathbf{n}}_2 \tag{11.1.3}$$

dependent on the single parameter $t \in [0, \pi)$.

Such an axis $\hat{\mathbf{n}}$ paired with the expectation of a rotation from $\mathbf{v}$ to $\mathbf{w}$ must give rise to a single θ, which we surmise should thus also be able to be written as a function $\theta(t)$.

Proposition 11.1.1. Given $\mathbf{v}$ and $\mathbf{w}$ linearly independent vectors lying on the same sphere about the origin with α as the angle between them, the relation between an axis $\hat{\mathbf{n}}(t)$ as defined in (11.1.3) and the mathematically positive[2] angle $\theta(t) \in (-\pi, \pi)$ required for that axis to map $\mathbf{v}$ onto $\mathbf{w}$ is:

$$\tan \frac{\theta(t)}{2} = \tan \frac{\alpha}{2} \sec t, \tag{11.1.4}$$

with $\theta(\pi/2) = \pi$.

Proof. Recall that the rotation $R(t)$ acting on $\mathbf{v}$ gives $\mathbf{w}$. Additionally, we can use Equation (8.2.2) (and using $K\mathbf{v}$=$\hat{\mathbf{n}}(t) \times \mathbf{v}$) to expand this:

$$\mathbf{w} = \mathbf{v} + \sin(\theta(t))(\hat{\mathbf{n}} \times \mathbf{v}) + (1 - \cos(\theta(t)))\hat{\mathbf{n}}(t) \times (\hat{\mathbf{n}}(t) \times \mathbf{v}). \tag{11.1.5}$$

We next take the dot product of this with $\mathbf{v}$:

$$\mathbf{w} \cdot \mathbf{v} = |\mathbf{v}|^2 + \sin(\theta(t))(\hat{\mathbf{n}}(t) \times \mathbf{v}) \cdot \mathbf{v} + (1 - \cos(\theta(t)))(\hat{\mathbf{n}}(t) \times (\hat{\mathbf{n}}(t) \times \mathbf{v})) \cdot \mathbf{v}. \tag{11.1.6}$$

Firstly, the second term vanishes as the cross product $\hat{\mathbf{n}}(t) \times \mathbf{v}$ is perpendicular to $\mathbf{v}$. We can utilise the circular shift of the scalar triple product, $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$,

[2] Mathematically positive with respect to the right hand rule - that is, with a generic ‘thumbs up’ motion, the thumb points in the direction of $\hat{\mathbf{n}}$ whilst the curled fingers indicate the positive direction of rotation. This is equivalent to ‘anticlockwise’ around the axis when it points towards you.

to obtain

$$\mathbf{v}\cdot(\hat{\mathbf{n}}(t)\times(\hat{\mathbf{n}}(t)\times\mathbf{v}))=(\hat{\mathbf{n}}(t)\times\mathbf{v})\cdot(\mathbf{v}\times\hat{\mathbf{n}}(t))=-|\hat{\mathbf{n}}(t)\times\mathbf{v}|^2 \tag{11.1.7}$$

We next use the knowledge that $\frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|}=\cos\alpha$ (along with $|\mathbf{v}|=|\mathbf{w}|$):

$$\cos\alpha=1-(1-\cos(\theta(t)))|\hat{\mathbf{n}}(t)\times\mathbf{v}|^2/|\mathbf{v}|^2 \tag{11.1.8}$$

We can further simplify the right hand side by substituting in $\hat{\mathbf{n}}(t)=\cos(t)\hat{\mathbf{n}}_2+\sin(t)\hat{\mathbf{n}}_1$. Then we can distribute the cross product to obtain

$$|\mathbf{v}\times\hat{\mathbf{n}}(t)|^2=|\cos(t)\mathbf{v}\times\hat{\mathbf{n}}_2+\sin(t)\mathbf{v}\times\hat{\mathbf{n}}_1|^2,$$

and as $\hat{\mathbf{n}}_1$ is parallel to $(\mathbf{v}+\mathbf{w})$, $\mathbf{v}\times\hat{\mathbf{n}}_1$ is parallel to $\hat{\mathbf{n}}_2$, and thus the two terms within this norm are perpendicular. This allows us to utilise Pythagoras' Theorem, and separate this norm:

$$\begin{aligned}|\mathbf{v}\times\hat{\mathbf{n}}(t)|^2 &=\cos^2(t)|\mathbf{v}\times\hat{\mathbf{n}}_2|^2+\sin^2(t)|\mathbf{v}\times\hat{\mathbf{n}}_1|^2\\ &=\cos^2(t)|\mathbf{v}|^2+\sin^2(t)\sin^2(\alpha/2)|\mathbf{v}|^2\\ &=\cos^2(t)|\mathbf{v}|^2+\sin^2(t)\frac{1-\cos\alpha}{2}|\mathbf{v}|^2,\end{aligned}$$

where $|\mathbf{v}\times\hat{\mathbf{n}}_2|=|\mathbf{v}||\hat{\mathbf{n}}_2|\sin(\pi/2)=|\mathbf{v}|$ as $\mathbf{v}$ is perpendicular to $\hat{\mathbf{n}}_2$, and $|\mathbf{v}\times\hat{\mathbf{n}}_1|=|\mathbf{v}||\hat{\mathbf{n}}_1|\sin(\alpha/2)$ as the direction of $\hat{\mathbf{n}}_1$ is precisely halfway between $\mathbf{v}$ and $\mathbf{w}$.

Plugging this into (11.1.8) yields

$$\cos\alpha=1-(1-\cos(\theta(t)))(\cos^2(t)+\frac{1}{2}(1-\cos\alpha)\sin^2(t)),$$

which implies

$$1-\cos(\theta(t))=\frac{1-\cos\alpha}{\cos^2(t)+\frac{1}{2}(1-\cos\alpha)\sin^2(t)}. \tag{11.1.9}$$

Therefore $\theta(t)$ is given by

$$\cos(\theta(t))=\frac{\cos\alpha-\frac{1}{2}(1+\cos\alpha)\sin^2(t)}{1-\frac{1}{2}(1+\cos\alpha)\sin^2(t)}. \tag{11.1.10}$$

For $t = 0$ one readily finds $\cos(\theta(0)) = \cos\alpha$ while $t = \pi/2$ gives $\cos(\theta(\pi/2)) = -1$, confirming the setup. Note that this angle only depends on the relative angle between $\mathbf{v}$ and $\mathbf{w}$.

However, this does not distinguish between the positive and negative values of θ. In fact, one sign will correspond to a rotation from $\mathbf{v}$ to $\mathbf{w}$ whilst the other will correspond to the rotation from $\mathbf{w}$ to $\mathbf{v}$. It remains to detect which sign corresponds to which direction.

Right scalar multiplication of the Rodriguez formula (8.2.2) applied to the vector $\mathbf{v}$ by $\hat{\mathbf{n}}(t) \times \mathbf{v}$ yields

$$\begin{aligned}
\mathbf{w} \cdot (\hat{\mathbf{n}}(t) \times \mathbf{v}) &= \sin(\theta(t))|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2, \\
\hat{\mathbf{n}}(t) \cdot (\mathbf{v} \times \mathbf{w}) &= \sin(\theta(t))|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2, \\
\hat{\mathbf{n}}(t) \cdot (|\mathbf{v} \times \mathbf{w}|\hat{\mathbf{n}}_2) &= \sin(\theta(t))|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2, \\
|\mathbf{v} \times \mathbf{w}|(\sin t\hat{\mathbf{n}}_1 + \cos t\hat{\mathbf{n}}_2) \cdot \hat{\mathbf{n}}_2 &= \sin(\theta(t))|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2, \\
|\mathbf{v} \times \mathbf{w}| \cos t &= \sin(\theta(t))|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2, \\
\frac{|\mathbf{v} \times \mathbf{w}|}{|\hat{\mathbf{n}}(t) \times \mathbf{v}|^2} \cos t &= \sin(\theta(t)).
\end{aligned}$$

As the fraction is strictly positive, we can state that the sign of $\theta(t)$ is the same as the sign of $\cos t$.

Equation 4.8 leads to the following identities, using the specific tangent half-angle formula $\tan(\theta/2) = \sqrt{\frac{1-\cos\theta}{1+\cos\theta}}$:

$$\begin{aligned}
\tan\frac{\theta(t)}{2} &= \pm\sqrt{\frac{1-\cos(\theta(t))}{1+\cos(\theta(t))}} \\
&= \pm\sqrt{\frac{1-\cos\alpha}{1+\cos\alpha-(1+\cos\alpha)\sin^2(t)}} \\
&= \pm\sqrt{\frac{1-\cos\alpha}{(1+\cos\alpha)(1-\sin^2(t))}} \\
&= \pm\frac{1}{\cos(t)}\sqrt{\frac{1-\cos\alpha}{1+\cos\alpha}}
\end{aligned}$$

$$= \pm \tan \frac{\alpha}{2} \sec(t).$$

Additionally, from earlier, we know that the sign of θ agrees with the sign of $\cos t$, which in turn agrees with the sign of $\sec t = 1/\cos t$. Since $\tan(\alpha/2)$ is strictly positive, the negative case is ruled out, and we obtain:

$$\tan \frac{\theta(t)}{2} = \tan \frac{\alpha}{2} \sec t.$$

This equation is ill-defined for $t = \pi/2$, but continuous extension yields $\theta(t) = \pi$ for $t = \pi/2$.

□

11.1.2 The Great Circle

The axes $\hat{\mathbf{n}}(t) = \hat{\mathbf{n}}_1 \sin t + \hat{\mathbf{n}}_2 \cos t$ lie on a circle of radius 1 about the origin, and are equidistant from $\mathbf{v}$ and $\mathbf{w}$. Such a circle is known as a *great circle*. Each pair of vectors will provide its own great circle, whilst two such pairs' great circles will intersect at precisely two points (which are equivalently found at $\pm(\mathbf{v} - \mathbf{w}) \times (\mathbf{p} - \mathbf{q})$ for vector pairs $\mathbf{v}, \mathbf{w}$ and $\mathbf{p}, \mathbf{q}$).

One idea which arose from ESGI was to generate these great circles for many pairs, and then to search the unit sphere for points with a high density of circles passing them - thus implying a large amount of pairs had a viable rotation about an axis in the direction of said point. However, this still lacks the information of angle and would thus still need the deduction and comparison of angle.

11.1.3 Quaternion Representation

A main contribution of the ESGI project moderator Cameron Hall was the use of quaternion representation of viable rotation matrices to provide a more simple search space as well as the discovery of Proposition 11.1.3 below.

A quaternion is an extension of complex numbers to 4 dimensions at the expense of commutativity. Introduced by Hamilton in 1843 and expanded in subsequent letters up to 1850[22], a quaternion is written as $\boldsymbol{q} = q_0 + q_i\boldsymbol{i} + q_j\boldsymbol{j} + q_k\boldsymbol{k}$, where $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$. We name the space of all quaternions $\mathbb{H}$ (after Hamilton).

We write a vector $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ in quaternion form as $\boldsymbol{x} = \mathcal{Q}(\mathbf{x}) = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ - that is, $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ correspond to the unit axes, and it has no real part.

Proposition 11.1.2. A rotation about a unit axis $\hat{\mathbf{n}}$ with angle $\theta \in (-\pi, \pi]$ (mathematically positively with respect to $\hat{\mathbf{n}}$) can be represented through the quaternion

$$\boldsymbol{q} = \cos(\theta/2) + \sin(\theta/2)\hat{\boldsymbol{n}}$$

(where $\hat{\boldsymbol{n}} = \mathcal{Q}(\hat{\mathbf{n}})$ as above: note that quaternions are italicised).

The rotation is then performed on a purely imaginary quaternion $\boldsymbol{x} = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$ (representing the point$(x, y, z) \in \mathbb{R}^3$) through the quaternion operation

$$\boldsymbol{x} \to \boldsymbol{qxq}^{-1}.$$

Proof. We compare the Rodriguez formula (8.2.3) with the above operation. Since $\boldsymbol{q}$ is a unit quaternion, we have in this case $\boldsymbol{q}^{-1} = \cos(\theta/2) - \sin(\theta/2)\hat{\boldsymbol{n}}$.

First, we consider the multiplication of two quaternions purely imaginary components:

$$\begin{aligned}&(q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k})(x_1\boldsymbol{i} + x_2\boldsymbol{j} + x_3\boldsymbol{k})\\ &\quad = -(q_1x_1 + q_2x_2 + q_3x_3) + (q_2x_3 - q_3x_2)\boldsymbol{i} + (q_3x_1 - q_1x_3)\boldsymbol{j} + (q_1x_2 - q_2x_1)\boldsymbol{k}\end{aligned}$$

when these are written as vectors $\mathbf{q} = [q_1, q_2, q_3]^\top$ and $\mathbf{x} = [x_1, x_2, x_3]^\top$, this is equivalent to $-\mathbf{q}\cdot\mathbf{x} + \mathbf{q}\times\mathbf{x}$ (though outside quaternions it makes little mathematical sense to add a scalar and vector). That is,

$$\boldsymbol{qx} = \mathcal{Q}(\mathbf{q})\mathcal{Q}(\mathbf{x}) = -\mathbf{q}\cdot\mathbf{x} + \mathcal{Q}(\mathbf{q}\times\mathbf{x})$$

for purely imaginary quaternions $\boldsymbol{q} = \mathcal{Q}(\mathbf{q})$ and $\boldsymbol{x} = \mathcal{Q}(\mathbf{x})$.

We expand our rotation equation into parts dependent on the real components and parts dependent only on the imaginary components $\hat{\boldsymbol{n}} = \mathcal{Q}(\hat{\mathbf{n}})$ and $\boldsymbol{x} = \mathcal{Q}(\mathbf{x})$:

$$\begin{aligned}
\boldsymbol{q}\boldsymbol{x}\boldsymbol{q}^{-1} &= [\cos(\theta/2) + \sin(\theta/2)\hat{\boldsymbol{n}}]\,\boldsymbol{x}\,[\cos(\theta/2) - \sin(\theta/2)\hat{\boldsymbol{n}}] \\
&= \cos(\theta/2)^2\boldsymbol{x} - \cos(\theta/2)\sin(\theta/2)\boldsymbol{x}\hat{\boldsymbol{n}} + \cos(\theta/2)\sin(\theta/2)\hat{\boldsymbol{n}}\boldsymbol{x} - \sin(\theta/2)^2\hat{\boldsymbol{n}}\boldsymbol{x}\hat{\boldsymbol{n}} \\
&= \cos(\theta/2)^2\boldsymbol{x} + \cos(\theta/2)\sin(\theta/2)\,(\hat{\boldsymbol{n}}\boldsymbol{x} - \boldsymbol{x}\hat{\boldsymbol{n}}) \\
&\qquad\qquad - \sin(\theta/2)^2\,(-\hat{\mathbf{n}}\cdot\mathbf{x} + \mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x}))\,\hat{\boldsymbol{n}} \\
&= \cos(\theta/2)^2\boldsymbol{x} + \cos(\theta/2)\sin(\theta/2)\,(2\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x})) \\
&\qquad\qquad - \sin(\theta/2)^2\,(-(\hat{\mathbf{n}}\cdot\mathbf{x})\hat{\boldsymbol{n}} + \mathcal{Q}((\hat{\mathbf{n}}\times\mathbf{x})\times\hat{\mathbf{n}})) \\
&= \frac{1}{2}(\cos\theta + 1)\boldsymbol{x} + \sin(\theta)\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x}) \\
&\qquad\qquad + \frac{1}{2}(1 - \cos(\theta))\,((\hat{\mathbf{n}}\cdot\mathbf{x})\hat{\boldsymbol{n}} + \mathcal{Q}(\hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x})))
\end{aligned}$$

We can easily see at this point there is no real component. Additionally, the term $\sin\theta\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x})$ maps to the term $\sin\theta K$ from the Rodriguez formula: because of $\mathcal{Q}(K\mathbf{v}) = \mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{v})$.

We now use the fact that $\mathcal{Q}(\hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x})) = (\hat{\mathbf{n}}\cdot\mathbf{x})\hat{\boldsymbol{n}} - \boldsymbol{x}$ to bring the cosine together:

$$\begin{aligned}
\boldsymbol{q}\boldsymbol{x}\boldsymbol{q}^{-1} &= \frac{1}{2}(\cos\theta - 1 + 2)\boldsymbol{x} + \sin(\theta)\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x}) \\
&\qquad\qquad + \frac{1}{2}(1 - \cos(\theta))\,((\hat{\mathbf{n}}\cdot\mathbf{x})\hat{\boldsymbol{n}} + \mathcal{Q}(\hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x}))) \\
&= \frac{1}{2}\cdot 2\boldsymbol{x} + \sin(\theta)\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x}) \\
&\qquad\qquad + \frac{1}{2}(1 - \cos(\theta))\,((\hat{\mathbf{n}}\cdot\mathbf{x})\hat{\boldsymbol{n}} + \mathcal{Q}(\hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x})) - \boldsymbol{x}) \\
&= \boldsymbol{x} + \sin(\theta)\mathcal{Q}(\hat{\mathbf{n}}\times\mathbf{x}) + (1 - \cos(\theta))\,(\mathcal{Q}(\hat{\mathbf{n}}\times(\hat{\mathbf{n}}\times\mathbf{x}))). \\
&= \boldsymbol{x} + \sin(\theta)\mathcal{Q}(K\mathbf{x}) + (1 - \cos(\theta))\mathcal{Q}(K^2\mathbf{x}) \\
&= \mathcal{Q}(R\mathbf{x})
\end{aligned}$$

This maps perfectly onto Formula 8.2.3.

□

By the angle restriction, the unit quaternion $\boldsymbol{q}$ in Proposition 11.1.2 is uniquely determined by its purely imaginary part. So, we can represent a rotation by $\sin(\theta/2)\hat{\mathbf{n}}$. Now we apply this to our target vectors $\mathbf{v}$ and $\mathbf{w}$.

Proposition 11.1.3. Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$ be linearly independent vectors with $|\mathbf{v}| = |\mathbf{w}|$. Then the set $\sin(\theta/2)\hat{\mathbf{n}}$ of all rotations mapping $\mathbf{v}$ to $\mathbf{w}$ traces an ellipse in the bisecting plane.

Proof. We can show this is an ellipse if we map $\hat{\mathbf{n}}_1$ to the x-axis at $(1, 0)$ and $\hat{\mathbf{n}}_2$ to the y-axis at $(0, 1)$. This implies the unit vector $\hat{\mathbf{n}}$ is mapped to $(\sin t, \cos t)$. The equation of the ellipse is then $x^2 + \frac{y^2}{\sin^2(\alpha/2)} = 1$.

To verify this, recall that $x = \sin(\theta(t)/2)\sin t$ and $y = \sin(\theta(t)/2)\cos t$. Then,

$$\begin{aligned} x^2 + \frac{y^2}{\sin(\alpha/2)} &= \sin^2(\theta(t)/2)\sin^2 t + \frac{\sin^2(\theta(t)/2)\cos^2 t}{\sin^2(\alpha/2)} \\ &= \sin^2(\theta(t)/2)\left(\sin^2 t + \frac{\cos^2 t}{\sin^2(\alpha/2)}\right) \end{aligned}$$

First we confirm the case for $t = \pi/2$, such that we can use the simplified tangent formula for the remaining cases:

$$\begin{aligned} x^2 + \frac{y^2}{\sin(\alpha/2)} &= \sin^2(\pi/2)\left(\sin^2(\pi/2) + \frac{\cos^2(\pi/2)}{\sin^2(\alpha/2)}\right) \\ &= 1(1+0) \\ &= 1. \end{aligned}$$

With this satisfied, we utilise (11.1.4) alongside $\sin^2\phi = \frac{\tan^2\phi}{1+\tan^2\phi}$

$$\begin{aligned} x^2 + \frac{y^2}{\sin(\alpha/2)} &= \frac{\tan^2(\theta(t)/2)}{1+\tan^2(\theta(t)/2)}\left(\sin^2 t + \frac{\cos^2 t}{\sin^2(\alpha/2)}\right) \\ &= \frac{\tan^2(\alpha/2)\sec^2 t}{1+\tan^2(\alpha/2)\sec^2 t}\left(\sin^2 t + \frac{(1+\tan^2(\alpha/2))\cos^2 t}{\tan^2(\alpha/2)}\right) \\ &= \frac{1}{\cos^2 t + \tan^2(\alpha/2)}\left(\tan^2(\alpha/2)\sin^2 t + (1+\tan^2(\alpha/2))\cos^2 t\right) \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{\cos^2 t + \tan^2(\alpha/2)} \Big(\tan^2(\alpha/2)(1 - \cos^2 t) + (1 + \tan^2(\alpha/2)) \cos^2 t \Big) \\
&= \frac{1}{\cos^2 t + \tan^2(\alpha/2)} \left(\begin{matrix} \tan^2(\alpha/2) - \tan^2(\alpha/2)\cos^2 t \\ \qquad + \cos^2 t + \cos^2 t \tan^2(\alpha/2) \end{matrix} \right) \\
&= \frac{1}{\cos^2 t + \tan^2(\alpha/2)} \Big(\tan^2(\alpha/2) + \cos^2 t \Big) \\
&= 1
\end{aligned}
$$

With this equation satisfied, we have shown that $\sin(\theta(t)/2)\hat{\mathbf{n}}(t)$ forms an ellipse. □

Given such a representation, we can theoretically draw the families of rotations $\sin(\theta(t)/2)\hat{\mathbf{n}}(t)$ (see Equation (11.1.3)) for many pairs of points. Finding locations in 3d space where these curves cluster again suggests very likely twin laws, but with an added benefit over the great circle of having the angle pre-described.

The location of clusters of points is an ongoing issue, and I expect some investigation into methods utilised in other fields or in computer gaming (which has a particular focus on fast processing, see eg [16],[60]) could lead to dramatic time improvements.

Part III

Analysis of Nonspherical Refinement

Chapter 12

Introduction

12.1 Motivation

Within the *Independent Atom Model*, form factors for atoms have been tabulated (see [6], formula 6.1.1.15 and table 6.1.1.4) as if they were independent in space - that is, ignoring any dependency on other atoms in the molecule. In actuality, the electron density is likely to alter its shape based on the molecule - showing bonds and other interatomic effects.

Historically (1965), Robert Stewart derived non-spherical form factors for bonded hydrogen atoms and commented [56]:

> By necessity, if not by choice, crystallographers have treated bonded atoms as point nuclei with a spherically symmetrical distribution of electron charge.

Technology (with regards to computational power) has now advanced sufficiently that we can now directly calculate quantum mechanically the electron density function ρ of the entire molecule, which provides us with new *non-spherical* form factors $f_j(\mathbf{x}, \mathbf{h})$ for every atom.

With these new form factors, the refinement process must be re-evaluated, to check where changes need to be made. Whilst we find that much of the process requires only simple or no modifications, one aspect of the process - the form factor derivatives (presented in Subsection 12.2.4) - requires a much more in depth investigation which is undertaken in Chapter 13 of this part, and published in [38].

12.2 Theory

We will explain the mathematics behind the use of non-spherical form factors and how *olex2.refine* has been adapted to enable their use. We keep the notation close to the one used in [4]. The non-spherical case remains very similar to the spherical case, with some critical differences which we will summarise at the end of this discussion.

12.2.1 Quantum Theory

The development of computational methods to derive a molecular wavefunction ρ_{mol} has been vital for the implementation of non-spherical theory. We have made use of ORCA [39] as a well-maintained such program which has been interfaced with olex2 via NoSpherA2 [31]. In the discussion ahead, we omit the consideration of spin for simplicity of the presentation, but it is an important component for quantum mechanical programs.

In its most fundamental form, we seek an electronic wavefunction $\Psi_{el}(\mathbf{r})$, where $\mathbf{r}$ consists of the three-dimensional locations $\mathbf{r}_i = (x_i, y_i, z_i)$ (i in $\{1, \ldots, n\}$) of the electrons which make up the molecule (to describe the electrons as 'being located' at a single point is somewhat of a false statement, nevertheless the modelling of an electron as a probability distribution of a point charge is an effective one). To present it somewhat more intuitively, the integral

$$\int_R \Psi^*_{el}(\mathbf{r})\Psi_{el}(\mathbf{r})d\mathbf{r}$$

over a $3n$-dimensional set $R \subset \mathbb{R}^{3n}$ gives the probability of electrons lying in R, and $\int_{\mathbb{R}^{3n}} \Psi^*_{el}(\mathbf{r})\Psi_{el}(\mathbf{r})d\mathbf{r} = 1$.

Finding this Ψ_{el} is motivated by the *Schrödinger equation*:

$$\mathscr{H}\Psi = E\Psi,$$

where Ψ is the total wavefunction (which involves the nuclei as well as the electrons), the energy E is an eigenvalue, and the hamiltonian $\mathscr{H}$ is an operator on Ψ relating to the kinetic and potential *energies*, and can be expressed as follows:

$$\mathscr{H} = \underbrace{-\frac{\hbar^2}{2m_e}\sum_{i=1}^{n}\nabla_i^2 - \frac{\hbar}{2}\sum_{K=1}^{N}\nabla_K^2}_{\text{Kinetic Energies}} + \underbrace{\sum_{K=1}^{N}\sum_{L>K}^{N}\frac{Z_K Z_L e^2}{4\pi\varepsilon_0 R_{KL}} + \sum_{i=1}^{n}\sum_{j>i}^{n}\frac{e^2}{4\pi\varepsilon_0 r_{ij}}}_{\text{Repulsion}} \underbrace{-\sum_{K=1}^{N}\sum_{i=1}^{n}\frac{Z_K e^2}{4\pi\varepsilon_0 R_{Ki}}}_{\text{Attraction}}.$$

Here, the *kinetic energy* terms relate to the movement of the electrons and nuclei respectively, and contain the Laplace operator $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$. The repulsion terms apply nucleus to nucleus and electron to electron respectively, with the attraction term acting between nuclei and electrons, with n being the number of electrons and N being the number of atoms. The Z_K correspond to the number of protons in the nucleus of the Kth atom, $e = 1.602 \times 10^{-19}C$ to the charge of an electron, $m_e = 9.109 \times 10^{-28}g$ the mass of an electron, R and r to the distances between bodies, and $\hbar = 1.055 \times 10^{-34}Js$ Planck's constant with $\varepsilon_0 = 8.854 \times 10^{-12}C^2N^{-1}m^{-2}$ the permittivity constant of vacuum.

In our case, we seek the Ψ which matches to the minimal eigenvalue E_0, known as the 'ground state' of the molecule. Excitement of electrons can lead to higher energies and infinitely many possible energies E_i, but we seek the wavefunction for the basic state of the molecule which matches to the minimal possible energy E_0.

Solving for Ψ is an impossible prospect, and thus various simplifications are made. The Born-Oppenheimer approximation neglects the movement of nuclei, allowing the nuclear kinetic energy function to be removed to an external factor which can be reintroduced later. As a result of this approximation, we obtain an electronic wavefunction Ψ_{el} depending only on the $3n$ coordinates of the electrons (and their spins which we omit for simplicity as mentioned above).

A vital simplification is to allow a separation of variables for the wavefunction.

One such separation (as used in the Hartree-Fock method) is given by the Slater determinant,

$$\Psi_{el} = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_1(\mathbf{r}_1) & \phi_2(\mathbf{r}_1) & \dots & \phi_n(\mathbf{r}_1) \\ \phi_1(\mathbf{r}_2) & \phi_2(\mathbf{r}_2) & \dots & \phi_n(\mathbf{r}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{r}_n) & \phi_2(\mathbf{r}_n) & \dots & \phi_n(\mathbf{r}_n) \end{vmatrix},$$

where the ϕ_i represent molecular orbitals of the molecules. This is motivated by the consideration of the wavefunction as a product of individual molecular orbitals for the electrons, alongside the requirement of the Pauli Principle that interchanging two $\mathbf{r}_i$ parameters should change the sign of the electronic wavefunction, which therefore becomes an antisymmetric function. A final approximation is to take these molecular orbitals as sums of atomic orbitals χ_μ, $\phi_i = \sum_\mu c_{\mu i}\chi_\mu$. These atomic orbitals are pre-calculated, and with this the computation needed is only to calculate the coefficients $c_{\mu i}$ which minimise the energy E and thus satisfy our hamiltonian. It is typically the case that one will not know this minimum E in advance, and thus it is again something which can only be refined towards, not calculated analytically.

Of course, it is easy to see that with an ever increasing number of electrons, the determinant will get ever more complicated. Additionally, there is no limit on the number of atomic orbital functions used. Thus, any software is presented with a trade-off - to obtain the most perfect result (given these assumptions), one would need to use an infinite number of orbitals, taking infinite time, but the less that are used the less accurate the result. There are far more trade-offs than this present in the determination of wavefunctions, with further approximations allowing faster computation at the cost of accuracy, but the full details of these are outside the scope of this thesis. Practically, these calculations are done by *ORCA* and controlled by different basis sets, as seen in Subsection 13.5.4.

Given such a Ψ_{el}, we can obtain the *electron density function* through a summation of the individual probability densities of each electron, that is:

$$\rho^{\mathrm{mol}}(\mathbf{z}) = \sum_{i=1}^{n} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \cdots \int_{\mathbb{R}^3} \Psi_{el}(\mathbf{r})^* \Psi_{el}(\mathbf{r}) d\mathbf{r}_1 d\mathbf{r}_2 \ldots d\mathbf{r}_{i-1} d\mathbf{r}_{i+1} \ldots d\mathbf{r}_n,$$

where $\mathbf{z} = (x, y, z)$ is substituted for each $\mathbf{r_i}$ in turn in the summation. That is, $\mathbf{r}_i = (x, y, z)$ in the ith term of the sum.

12.2.2 Hirshfeld Partitioning

After using quantum mechanical computations to determine the electron density function ρ^{mol} of the whole molecule, this function is partitioned into individual atoms utilising a weight function - presently, we use

$$w_j(\mathbf{z}) = \frac{\rho_j^{\mathrm{sph}}(\mathbf{z})}{\rho^{\mathrm{pro}}(\mathbf{z})},$$

[24] where $\mathbf{z}$ is a point in space, ρ_j^{sph} is the spherical atomic density associated to the j^{th} atom which is the fourier inverse of the form factors tabulated in the international tables [6, Table 6.1.1.1-5], and $\rho^{\mathrm{pro}}(\mathbf{z}) = \sum_j \rho_j^{\mathrm{sph}}(\mathbf{z})$ is the promolecule density generated from the spherical atomic densities (these are in turn determined by the somewhat simpler single-atom Schrödinger equation). Other partitioning schemes are also viable[9]. Given a molecular electron density ρ^{mol}, we then assign each atom its nonspherical electron density function

$$\rho_i^{\mathrm{nonspher}}(\mathbf{z}) = w_j(\mathbf{z}) \rho^{\mathrm{mol}}(\mathbf{z}).$$

This is commonly known as *Hirshfeld Partitioning* or *Hirshfeld Stockholder Partitioning.* These atomic nonspherical electron density functions are transformed into nonspherical form factors f_j via their Fourier transform

$$f_j(\mathbf{x}, \mathbf{h}) = \int_{\mathbb{R}^3} \rho_j^{\mathrm{nonspher}}(\mathbf{x}, \mathbf{z}) e^{2\pi i \mathbf{z} \cdot \mathbf{h}} d\mathbf{z},$$

where the newly-introduced $\mathbf{x}$ is a parameter containing information about the crystallographic parameters. In the independent atom model, these electron densities (and thus form factors) do not depend on these crystallographic parameters.

As discussed in Part I, modelling the atoms independently using single-atom spherical atomic densities allows a more general solution for the form factors f_j, which has been quantum mechanically solved and can be very well approximated by sums of Gaussians, taking the form:

$$f_j(\mathbf{h}) = \sum_{i=1}^{4} a_i e^{-b_i\left(\frac{|\mathbf{h}|}{4\pi}\right)^2} + c,$$

where the constants a_i, b_i, c are tabulated and openly accessible at [58].

12.2.3 F-calcs and Form Factors

Our mathematical arguments focus on the necessary modifications concerning the treatment of the calculated structure factor, which we denote as usual by $F(\mathbf{x}, \mathbf{h})$. Recall that

$$F(\mathbf{x}, \mathbf{h}) = \sum_{j=1}^{n} f_j(\mathbf{x}, \mathbf{h}) e^{2\pi i \mathbf{z}_j \cdot \mathbf{h}} e^{-2\pi^2 \mathbf{h}^\top U_j \mathbf{h}}.$$

Firstly, we consider the impact of non-spherical form factors on the simplifications present due to *crystal symmetries.*

Recall that the structure factor may be partitioned into portions dependant on each individual *representative atom* A_j, $j = 1, 2, \ldots, N$, in the asymmetric unit:

$$F(\mathbf{x}, \mathbf{h}) = \sum_{j=1}^{N} \underbrace{\sum_{(R|\mathbf{t})\in\mathcal{S}} s_j \underbrace{f_j(\mathbf{x}, R\mathbf{h}) \overbrace{G_j(\mathbf{x}, R\mathbf{h}) e^{2\pi i \mathbf{h}\cdot\mathbf{t}}}^{G_j^{(R|\mathbf{t})}(\mathbf{x},\mathbf{h})}}_{\mathfrak{f}_j^{(R|\mathbf{t})}(\mathbf{x},\mathbf{h})}}_{\text{atoms equivalent by symmetry to } A_j} \tag{12.2.1}$$

with R the rotational part and $\mathbf{t}$ the translational part of the symmetry operation $(R|\mathbf{t}) \in \mathcal{S}$, where S is the set of all symmetries of the unit cell, and s_j the occupancy of the atom A_j[1]. Note that we use $(R|\mathbf{t})(\mathbf{z}) = R\mathbf{z} + t$.

Here the terms $f_j(\mathbf{x}, \mathbf{h})$ are derived from the complex values $f_j(h_i, k_i, l_i)$ given in the tsc-file by an addition of the anomalous scattering factors $f'_j + if''_j$, see [4, eqn. (53)], that is:

$$f_j(\mathbf{x}, \mathbf{h}) := f_j(h_j, k_j, l_j) + f'_j + if''_j.$$

These anomalous scattering factors f'_j and f''_j are fixed real values defined for each atom type, describing additional contributions which arise outside of the classical elastic scattering - for example, via excitation.

The form factor of the atom A_j is calculated in a coordinate system obtained by translating the origin of the crystallographic axes to the centre of atom A_j, with no change in orientation. The form factor $f_j(\mathbf{x}, \mathbf{h})$ is then the Fourier transform of the electron density ρ_j of A_j. In contrast to the case of spherical form factors, this electron density can now depend on the whole model whose information is given in $\mathbf{x}$, as non-spherical form factors take the dependence of the electron density of the surrounding atomic environment into account.

Note that *in the case of spherical form factors*, the functions f_j do not depend on the model information $\mathbf{x}$ and, additionally, we have $f_j(R\mathbf{h}) = f_j(\mathbf{h})$ since $f_j(\mathbf{h})$ does then not depend on the direction of $\mathbf{h}$ but only on $\mathbf{h}^\top M^* \mathbf{h} \in \mathbb{R}$, where M^* is the reciprocal metric matrix. This is generally not true in the case of non-spherical form factors.

For the ADPs, we find that

$$\begin{aligned} G_j(\mathbf{x}, R\mathbf{h}) &= e^{-2\pi^2 (R\mathbf{h})^\top U_j (R\mathbf{h})} \\ &= e^{-2\pi^2 \mathbf{h}^\top R^\top U_j R\mathbf{h}}, \end{aligned}$$

[1] In our paper discussing the introduction of the *.tsc* file [37], $\mathbf{h}$ is a row vector rather than the standard column used in this work, but aside from this transposition of parameters the equations remain the same.

which coincides with a rotation of the U_j by R.

By the symmetry restriction given for crystals, the term $e^{2\pi i \mathbf{h}\cdot\mathbf{t}}$ is always a root of unity of order 1,2,3,4, or 6.

Finally, we consider how rotations affect f_j itself, knowing that it is not preserved as in the case of spherical form factors. We can observe the derivation of $f_j(\mathbf{x}, R\mathbf{h})$ (see (4.1.6) in Part I):

$$\begin{aligned} f_j(\mathbf{x}, R\mathbf{h}) &= \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i \mathbf{y}\cdot(R\mathbf{h})} d\mathbf{y} \\ &= \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i \mathbf{y}^\top R\mathbf{h}} d\mathbf{y} \\ &= \int_{\mathbb{R}^3} \rho_j(\mathbf{y}) e^{2\pi i (R^\top \mathbf{y})^\top \mathbf{h}} d\mathbf{y} && \mathbf{x} = R^\top \mathbf{y} \\ &= \int_{\mathbb{R}^3} \rho_j(R\mathbf{x}) e^{2\pi i \mathbf{x}\cdot\mathbf{h}} d\mathbf{x} && R^{-1} = R^\top, \end{aligned}$$

that is, rotating the atomic density distribution by R results in the form factor of $R\mathbf{h}$.

The least square minimization in the refinement procedure requires derivatives of the structure factor with respect to the components of $\mathbf{x} = (x_1, \ldots, x_n)$. Since the structure factor is the above sum (12.2.1), we only need to consider the derivatives of the individual terms $\mathfrak{f}_j^{(R|\mathbf{t})}(\mathbf{x}, \mathbf{h})$.

Using the product rule, we have for the derivative

$$\frac{\partial \mathfrak{f}_j^{(R|\mathbf{t})}}{\partial x_k}(\mathbf{x}, \mathbf{h}) = \frac{\partial f_j}{\partial x_k}(\mathbf{x}, R\mathbf{h}) G_j^{(R|\mathbf{t})}(\mathbf{x}, \mathbf{h}) + f_j(\mathbf{x}, R\mathbf{h}) \frac{\partial G_j^{(R|\mathbf{t})}}{\partial x_k}(\mathbf{x}, \mathbf{h}). \tag{12.2.2}$$

The differential $\frac{\partial f_j}{\partial x_k}(\mathbf{x}, R\mathbf{h})$ in the first term on the right hand side of (12.2.2) is more difficult to treat due to the complexity of the involved derivations. The derivatives of G_j can be handled easily due to their nature as simple exponential functions.

The set of Miller indices $\mathbf{h}$ required for the non-spherical form factors (given via the *.tsc* file, see 12.3) should correspond to the set of measured Bragg reflections and their symmetry equivalents, as can be seen in Equation (12.2.2).

12.2.4 Derivatives of Structure Factors with Nonspherical Form Factors

Up until (6.3.1) of Chapter 6 of Part I, the derivation required for least-squares refinement remains the same.

This requires the derivatives of this with regards to each parameter, which in turn requires the derivatives of the structure factors $Y_c(\mathbf{h})$ with regards to each parameter. As seen in (6.3.1),

$$\frac{\partial F_c}{\partial x_n}(\mathbf{x},\mathbf{h}) = \sum_{j=1}^{N_{\text{atoms}}} \left(\underbrace{\frac{\partial f_j}{\partial x_n}(\mathbf{x},\mathbf{h}) G_j(\mathbf{x},\mathbf{h})}_{(*)} + f_j(\mathbf{x},\mathbf{h}) \frac{\partial G_j}{\partial x_n}(\mathbf{x},\mathbf{h}) \right). \tag{12.2.3}$$

In fact, we require the partial derivatives of the (theoretical) intensity $Y_c := |F_c|^2$, which can be derived from the partial derivatives of the structure factor via

$$\frac{\partial Y_c}{\partial x_n}(\mathbf{x},\mathbf{h}) = 2\,\mathrm{Re}\left(F_c^*(\mathbf{x},\mathbf{h}) \frac{\partial F_c}{\partial x_n}(\mathbf{x},\mathbf{h}) \right), \tag{12.2.4}$$

where $F^*(\mathbf{x},\mathbf{h}) \in \mathbb{C}$ denotes the complex conjugate of $F(\mathbf{x},\mathbf{h}) \in \mathbb{C}$.

Note that the first term $(*)$ on the right hand side of (12.2.3) vanishes in the case of *spherical* form factors f_j, since they are only functions of the Miller triple $\mathbf{h}$ and not of the model $\mathbf{x} \in \mathbb{R}^N$, and the partial derivatives of G_j have explicit analytical expressions since the functions G_j are products of simple exponential functions. In the case of *nonspherical* form factors, the numerical computation of their partial derivatives $\frac{\partial f_j}{\partial x_n}$ is much more difficult. It involves the change of the form factor f_j under infinitesimally small variation of the parameters corresponding to individual atoms. In the current implementation of nonspherical refinement in *olex2.refine*, the first term on the right hand side of (12.2.3) is taken to be zero and the refinement uses the following approximation of the partial derivatives given in (12.2.3):

$$\left(\frac{\partial F}{\partial x_n}\right)_{\text{appr}}(\mathbf{x},\mathbf{h}) := \sum_{j=1}^{N_{\text{atoms}}} f_j(\mathbf{x},\mathbf{h}) \frac{\partial G_j}{x_n}(\mathbf{x},\mathbf{h}). \tag{12.2.5}$$

It is this approximation which needed to be validated and which became one main part of my PhD investigations.

This approximation gives rise to various questions. Practical experience shows that nonspherical refinement based on this approximation produces significantly better refinement models than spherical refinement (in particular with regards to the atom-to-hydrogen (X-H) distances). Can we achieve a substantial further improvement via more accurate partial derivatives of the nonspherical structure factors using tools like numerical differentiation? Are the collected data via X-ray diffraction precise enough to justify the extra computational effort? Currently the computation of these more accurate partial derivatives is not feasible due to its substantial time effort. However, computer performance may further improve in the years to come. The same may be true with regards to the improvement of precise X-ray diffraction data. To answer these questions it is necessary to carefully investigate the level of inaccuracy in this approximation and the potential advantage of using more accurate partial derivatives of nonspherical structure factors in crystallographic structure refinement. Our method of choice is numerical differentiation, as explained in Subsection 13.1.4. These important questions are the main concern of Chapter 13.

12.2.5 Summary

Let us finally cover the relevant differences to be taken into account when working with non-spherical form factors:

(i) Non-spherical form factors associated to atoms (with the origin at their centre) are no longer real, but are usually complex-valued (as the electron densities are non-spherical).

(ii) It is no longer the case that $f_j(R\mathbf{h}) = f_j(\mathbf{h})$ for rotations R associated to symmetry equivalent atoms in the unit cell.

(iii) Due to the change in the shape of form factors under shifts, there appears an additional term in the derivative of $\mathfrak{f}_j^{(R|\mathbf{t})}$ – the first term on the right hand side of (12.2.2).

(iv) The provided form factors must cover a greater variety of Miller indices than would be needed in the spherical case (due to (ii)). That is, form factors must be provided for all Miller indices $\mathbf{h}$ with recorded reflections and all symmetry equivalents $R\mathbf{h}$, for $(R, \mathbf{t})$ appearing in $\mathcal{S}$ (see (12.2.1)).

12.3 The .tsc Format File

In section 12.2 we found that we require specific values for nonspherical form factors corresponding to specific Miller indices. In this section we will provide information on the *.tsc* file, which is used to transfer this information.

In order to perform non-spherical refinement, *Olex2* requires non-spherical form factors for each *atom* for each *hkl* required. This is provided via an interlacing software (currently, we use *NoSpherA2*) which takes a wavefunction determined by a quantum mechanical program such as ORCA and partitions and Fourier transforms into the individual atomic form factors. This is fed into *Olex2* via the *.tsc* file format[37] (as mentioned at the end of 12.2.3), which can also be used for any alternate externally-sourced method of form factor generation.

Let us now give a brief explanation of the concrete format of this *.tsc* file.

Olex2 expects a file called *[name].tsc* (matching the *.hkl* file name) containing the following information in order to use the external atomic form factors:

The header of the *[name].tsc* file is free-format, as long as it contains the space-separated list of atom names in the 'SCATTERERS:' line and finishes with 'DATA:'. Any identifier must be followed by a colon. The identifiers may start with a space.

$f_j(h_i, k_i, l_i)$ is the form factor (Fourier transform of the electron density) of the atom A_j calculated in a coordinate system obtained by translating the origin of the crystallographic axes to the centre of atom A_j, at h_i, k_i, l_i. Index $j \in (1, \ldots, N)$ should run over all unique atoms of the asymmetric unit, and $i \in (1, \ldots, m)$ should run over at least all reflections defined in the *.hkl* file and any equivalents under symmetry.

[1] In either case, all symmetry equivalent Miller indices must be present in the DATA section. If a list of symmetry operators, expressed as rotation matrices (e.g.: 1 0 0 0 1 0 0 0 1;-1 0 0 0 1 0 0 0 -1) is provided, then the Miller indices must be ordered into corresponding blocks – and each block must have symmetry equivalent indices in the same position in each block and generated by the corresponding matrices. This allows for more efficient calculations during the refinement. Otherwise, if SYMM has the value 'expanded', the indices can be present in any order.

TITLE:	*optional title of the structure*
SYMM:	‘expanded’ *or list of symmetries* [1]
AD:	TRUE or FALSE (anomalous dispersion)
SCATTERERS:	*space-separated list of all atoms*
[ANYTHING] :	*colon must be present*
DATA:	(denotes the end of the header)

h	k	l	A_1	A_2	$\dots$	A_N
h_1	k_1	l_1	$f_1(h_1,k_1,l_1)$	$f_2(h_1,k_1,l_1)$	$\dots$	$f_N(h_1,k_1,l_1)$
h_2	k_2	l_2	$f_1(h_2,k_2,l_2)$	$f_2(h_2,k_2,l_2)$	$\dots$	$f_N(h_2,k_2,l_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
h_m	k_m	l_m	$f_1(h_m,k_m,l_m)$	$f_2(h_m,k_m,l_m)$	$\dots$	$f_N(h_m,k_m,l_m)$

The complex values $f_j(h_i,k_i,l_i)$ must be written as “Re,Im” - their real component followed by a comma followed by the imaginary component, with no spaces.

The start of a tsc file for epoxide, used in the rest of this work, is as follows:

```
TITLE: epoxide.cif

SYMM: expanded

SCATTERERS: O1 C2 H2a H2b C3 H3a H3b

DATA:

-9 0 1 1.42431973e+00,-7.16985709e-03 1.11352758e+00,4.98856433e-04...

-9 0 3 1.43428689e+00,-5.02026979e-03 1.12608844e+00,-3.74646732e-03...

-9 0 5 1.41162731e+00,-2.17215211e-03 1.08918940e+00,-6.49469368e-03...
```

The ’DATA’ rows have been truncated with ‘...’ to prevent the unclear information which would come from them line-wrapping. For example, the first line indicates that the form factor of O1 at -9, 0, 1 is 1.42431973-0.00716985709i. This file was obtained from a model at the spherical minimum.

Chapter 13

Investigating the Approximation

Typically, the partial derivatives $\frac{\partial f_j}{\partial x_i}$ of the form factors have been taken as zero. In this section, we will look deeply into the viability of calculating more accurate values for these derivatives, as well as whether they make a substantial difference to the resulting solution.

13.1 Main Refinement Methods & Background

In this section, I cover the required background knowledge for the testing of this approximation. First, I outline the chemical structures and data used for these tests, followed by a brief overview of the process required for approximate and numerical nonspherical refinements. Spherical refinement is the classical refinement method based on the Independent Atom Model. Approximate nonspherical refinement is that currently implemented in NoSpherA2, whilst numerical nonspherical refinement utilises numerical differentiation to gain more accurate partial derivatives for the form factors, and is our choice for these tests. I cover the process of numerical differentiation in more detail and clarify what is meant by certain 'step sizes', ϵ and δ. Henceforth, the model obtained via different refinement methods will be denoted with a subscript - that is, the minimum obtained by spherical refinement is $\mathbf{x}_{\text{spher}}$, that obtained via approximate nonspherical refinement is $\mathbf{x}_{\text{appr}}$ and that obtained via numerical nonspherical refinement is $\mathbf{x}_{\text{num}}$.

13.1.1 Molecules Used

In our tests, we employ three different molecules - ammonia (NH_3), [3] epoxide (C_2H_4O) [19] and L-alanine ($C_3H_7NO_2$) [34] (see Figure 13.1 for their chemical structures). In the latter two molecules, all atoms are in general positions and the vector $\mathbf{x}$ describing the model has 9 parameters for each atom (3 positional parameters and 6 ADPs). The space group of NH_3 is $P2_13$ and its model is determined by 9 parameters for the H atom, and only 3 parameters for the N atom as it is restricted to the symmetry line $x = y = z$ (meaning also that $U_{11} = U_{22} = U_{33}$ and $U_{12} = U_{13} = U_{23}$). Of course, in the presence of symmetries, the vector $\mathbf{x}$ can be "expanded" to a larger vector $\mathbf{y} = \mathbf{y}(\mathbf{x})$ which provides all 9 parameters for each atom in the molecule.

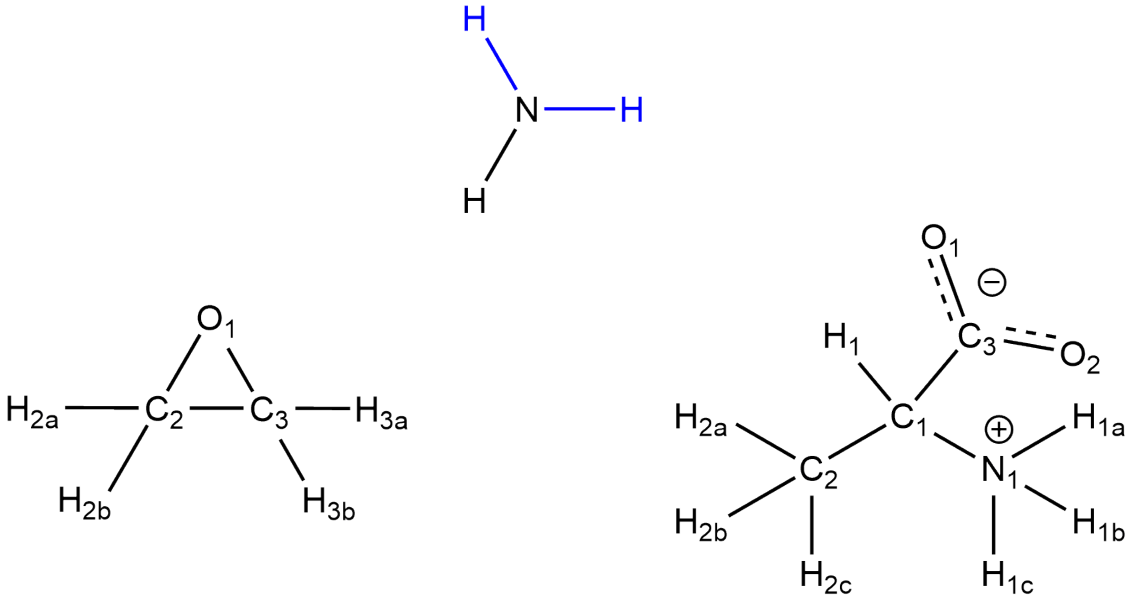


Figure 13.1: The chemical structures of ammonia (top), epoxide (left) and L-alanine (right). Due to crystallographic symmetries in ammonia, two symmetry-equivalent hydrogen atoms are shown in blue.

13.1.2 Approximate Nonspherical Refinement

We refer to the process of refinement with $\frac{\partial f_j}{\partial x_i}(\mathbf{x}, \mathbf{h})$ set to 0 as *approximate nonspherical refinement*. In short, we use molecular quantum mechanical computations and partitioning to obtain better form factors, but we use the simplest approximation of their derivatives by setting them equal to zero. Approximate nonspherical refinement is illustrated in the flow chart in Figure 13.2.

Approximate nonspherical refinement begins with an initial model obtained through classical spherical refinement. Then, by passing this model to a quantum mechanical calculation program, a quantum-mechanical wavefunction is calculated and transferred to the NoSpherA2 software in form of a *wfn* or *wfx* file. NoSpherA2 converts this into an electron density function for the model, partitions it, and computes the nonspherical form factors, which are returned in a *.tsc* file (as discussed in Section 12.3) for further use.

We then utilise least-square minimisation, bringing in other information such as the observed intensities Y_o and weights, calculating F_c and approximating $\partial F_c/\partial x_n$ by the analytical expression (12.2.3) (with $(*) = 0$) and using these to calculate a shift to the model which should result in a better agreement with observed data.

We run this loop repeatedly as in a classical refinement, until the model is converged (by the standard shift/esd ≤ 0.01 rule, where 'esd' is the estimated standard deviation, also referred to as standard uncertainty, of the model parameter), or we have reached a 'limit' $n_{\max}$ of how many times we are comfortable doing a refinement cycle without recalculating the .tsc file, as this will become increasingly inaccurate with increasing changes to the model. As calculation of the .tsc file is time consuming, we prefer to get more 'use' out of the same file before the model changes too significantly.

We then compare this output model to the input model - if they are *sufficiently close* (for example, if the maximal parameter difference/esd ≤ 0.01), then the refinement has converged and we take the output model as our final model. Here we also limit the number of iterations to $m_{\max}$ and we terminate the procedure if this convergence criterion has not been satisfied. In this case, Olex2 displays the message 'Warning: Unconverged Model!'

13.1.3 Numerical Nonspherical Refinement

Our new refinement method uses the same quantum mechanically calculated nonspherical form factors given via a .tsc file. However, a numerical procedure to derive improved values of the derivatives $\frac{\partial F_c}{\partial x_n}$ (in coordinates x_n representing positional parameters) is included as in (12.2.3), without the simplification that $(*)$ is zero. Since differentiation describes the change of F_c in the direction of x_n, we use models which closely neighbour the current model $\mathbf{x}$ - that is, we alter a single positional parameter x_n by a small amount to calculate the derivative $\frac{\partial F_c}{\partial x_n}$. This allows us to perform a mathematically more accurate least-squares minimisation process. We refer to this process as *numerical nonspherical refinement.*

In the case of numerical nonspherical refinement, illustrated by the flow chart Figure 13.3, the initial model can be either obtained by spherical refinement or by approximate nonspherical refinement. We utilise NoSpherA2 in each iteration not only once for the computation of the form factors f_j of the given model but also multiple times for the calculation of the derivatives $\frac{\partial F_c}{\partial x_n}$ of the structure factor. In fact, each such partial derivative computation via numerical differentiation requires two additional quantum mechanical and NoSpherA2 computations and the time complexity of this process grows linearly with the number of atoms in the model (see Subsection 13.3.4). We additionally are cautious and reduce the risk of inaccurate quantum mechanical data by only ever performing one refinement cycle with a given .tsc file. Additionally, we always carry out a full 20 steps whether or not the model has converged, to provide additional information whether a model remains in this status or whether it may still fluctuate. We record the model at each step and return the model with

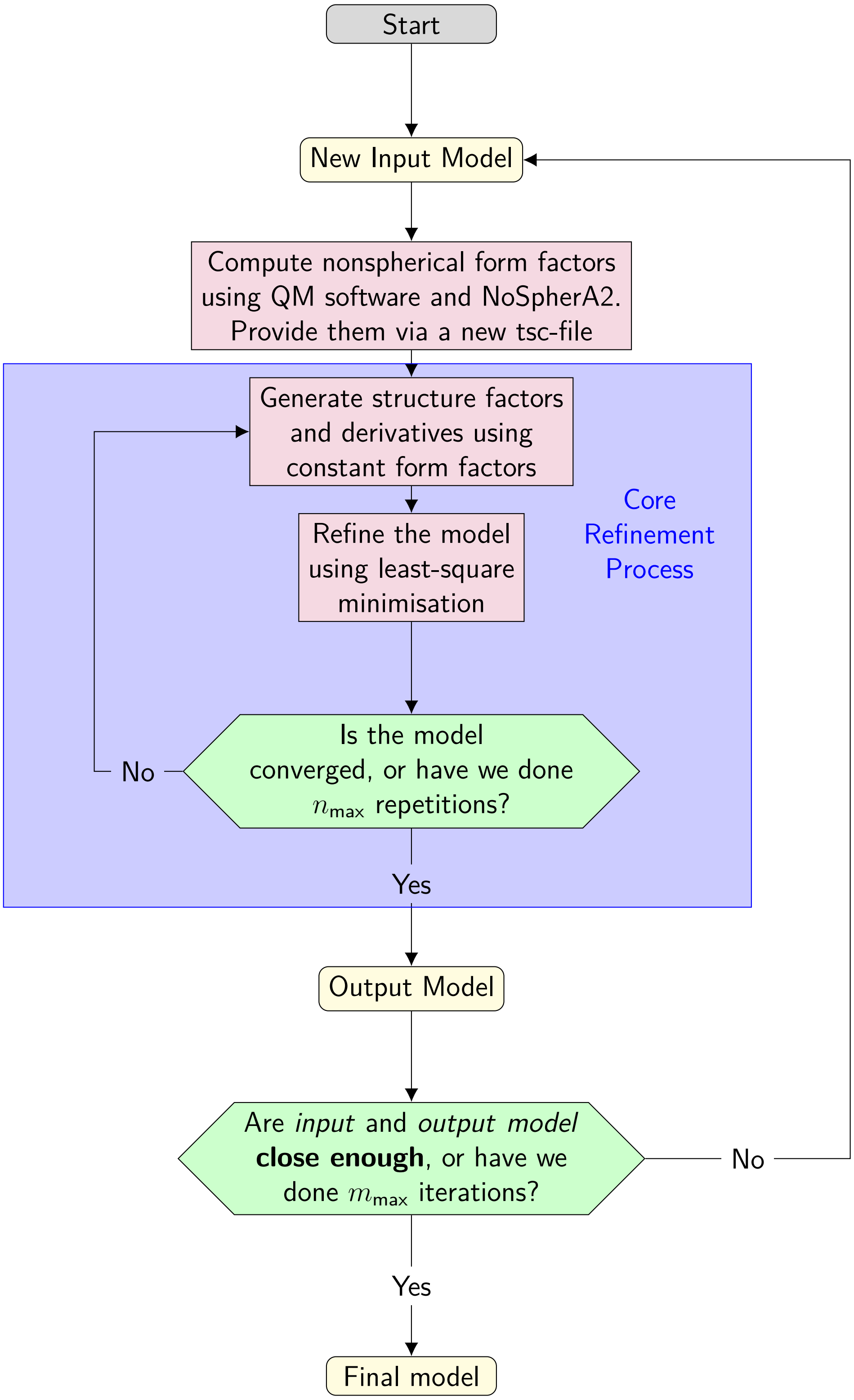

Start
New Input Model
Compute nonspherical form factors using QM software and NoSpherA2. Provide them via a new tsc-file
Generate structure factors and derivatives using constant form factors
Refine the model using least-square minimisation
Core Refinement Process
Is the model converged, or have we done n_{max} repetitions?
No
Yes
Output Model
Are *input* and *output model* **close enough**, or have we done m_{max} iterations?
No
Yes
Final model

Figure 13.2: Flow chart illustration of Approximate Nonspherical Refinement

the lowest wR$_2$-factor as our final model. These settings are currently only used for test purposes, not as defaults in NoSpherA2 for general use.

Typically, there are up to three derivative calculations per atom, corresponding to its x, y, and z coordinates. These can sometimes be reduced by the symmetry of the molecule, as in the case of ammonia. As the ADP parameters have no part in the wavefunction calculation, and no impact on the form factors, all derivatives related to them are simply taken as in the approximate nonspherical case.

13.1.4 Numerical Differentiation

Numerical differentiation of a function $F : \mathbb{R}^N \to \mathbb{R}$ is based on a discretization of its derivatives. For numerical nonspherical refinement, we use the *central difference quotient*

$$\left(\frac{\partial F}{\partial x_n}\right)_\epsilon (\mathbf{x}, \mathbf{h}) := \frac{F(\mathbf{x} + \epsilon \mathbf{e}_n, \mathbf{h}) - F(\mathbf{x} - \epsilon \mathbf{e}_n, \mathbf{h})}{2\epsilon},$$

where $\mathbf{e}_n \in \mathbb{R}^N$ denotes the n-th standard basis vector associated to the parameter x_n[1]. An important aspect in this numerical computation is the right choice of the step size $\epsilon > 0$: This parameter must be small enough to provide a good local fit of the slope describing the derivative (the error appearing here is called the *truncation error*), but it must be also large enough that the subtraction of almost equal numbers in the numerator does not produce a significant error due to the finite digit precision (this error is called the *rounding error*). Theory tells us that we have the following bound for the total error (see [7, p. 181], with thanks to Louis Aslett for pointing us to this specific resource):

$$\left| \frac{\partial F}{\partial x_n}(\mathbf{x}) - \frac{\overline{F}(\mathbf{x} + \epsilon \mathbf{e}_n) - \overline{F}(\mathbf{x} - \epsilon \mathbf{e}_n)}{2\epsilon} \right| \leq a\epsilon^2 + \frac{b}{\epsilon}. \tag{13.1.1}$$

[1] There also exist numerical differentiation of higher orders, making us of 4 or more positions to calculate the derivative. These typically calculate a more accurate derivative (an error related to step size cubed rather than squared), at the cost of needing additional calculations. We tested this, and found that there was no notable improvement worth the time cost.

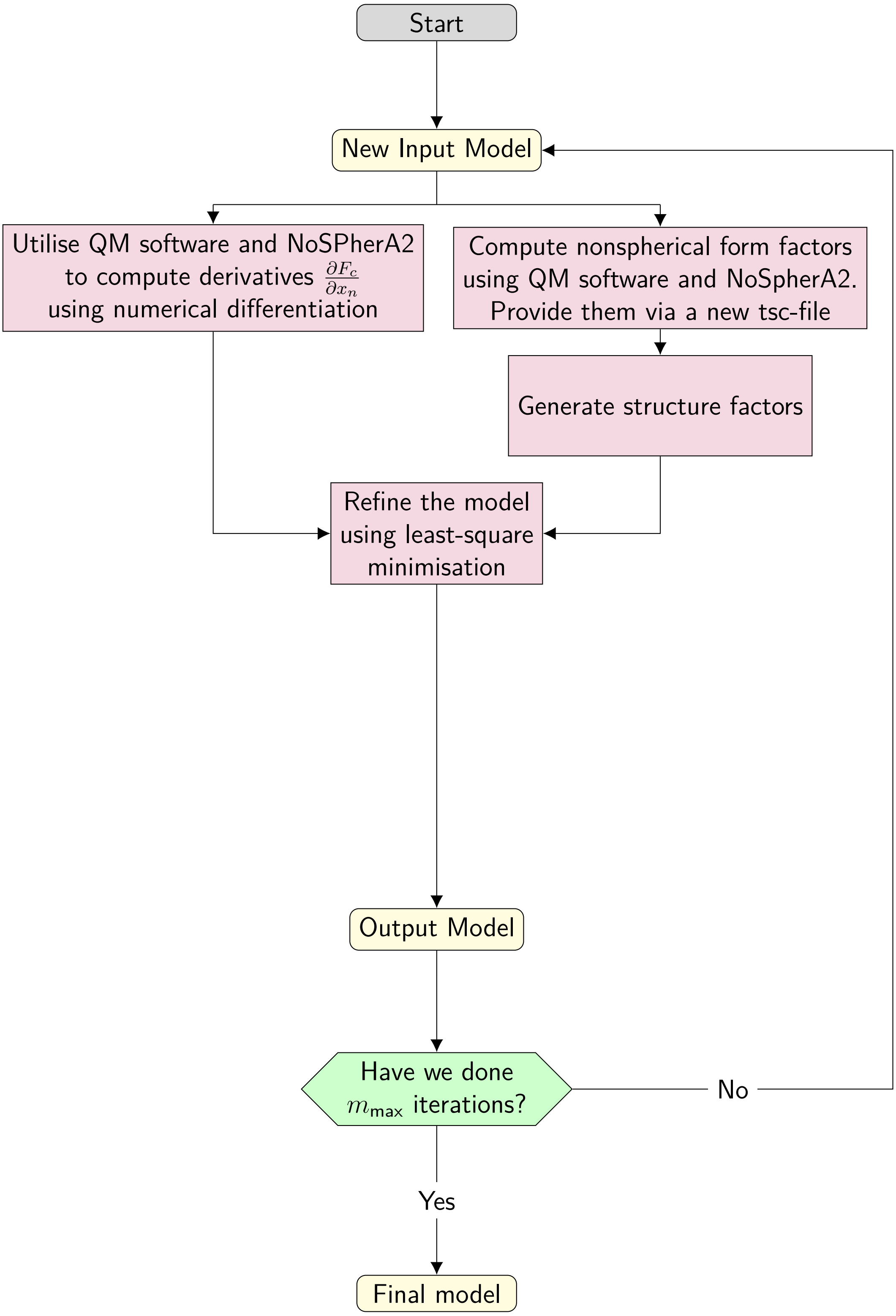

Start
New Input Model
Utilise QM software and NoSPherA2 to compute derivatives $\frac{\partial F_c}{\partial x_n}$ using numerical differentiation
Compute nonspherical form factors using QM software and NoSpherA2. Provide them via a new tsc-file
Generate structure factors
Refine the model using least-square minimisation
Output Model
Have we done m_{max} iterations?
No
Yes
Final model

Figure 13.3: Flow chart illustration of Numerical Nonspherical Refinement

Here, $\overline{F}(\mathbf{x})$ denotes the nearest floating-point number to $F(\mathbf{x})$, $a > 0$ is a constant involving the third derivative of F, and $b > 0$ is a constant involving the function F itself and the maximum relative error between all numbers and their floating-point representations.

As mentioned before, the lack of an exact analytical formula for the partial derivatives $\frac{\partial F}{\partial x_n}$ of the nonspherical structure factor requires us to compute them via numerical differentiation. This chapter is devoted to finding the optimal step size $\epsilon > 0$ for this numerical differentiation and to investigate the reliability of these numerically computed partial derivatives. We refer to them as the numerical partial derivatives of the structure factor.

Before we focus on these numerical derivatives, we first consider the accuracy obtained via numerical differentiation in the case of spherical and approximate refinement, which can be analytically calculated. This is carried out in Section 13.2. We consider this as a first reality check for our numerical differentiation.

Before we start our investigations, let us briefly discuss how we compute $\left(\frac{\partial F}{\partial x_n}\right)_{\text{appr}}$ (given in (12.2.5)) and the numerical partial derivatives $\left(\frac{\partial F}{\partial x_n}\right)_{\text{num},\epsilon}$ using numerical differentiation. For this we introduce the following notation for the structure factor based on a tsc-file:

$$F(\mathbf{x}', \mathbf{h}_k, tsc(\mathbf{x})) := \sum_{j=1}^{N_{\text{atoms}}} f_j(\mathbf{x}, \mathbf{h}_k) G_j(\mathbf{x}', \mathbf{h}_k). \tag{13.1.2}$$

Here $\mathbf{h}_k$ is one of the observed Miller triples and G_j is given in (4.3.1). Note that the terms $f_j(\mathbf{x}, \mathbf{h}_k)$ on the right hand side of (13.1.2) are extracted from the tsc-file $tsc(\mathbf{x})$ on the left hand side of (13.1.2). Using this flexible notation involving two models $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^N$, the computation of (12.2.5) using numerical differentiation with step size $\epsilon > 0$ is

$$\left(\frac{\partial F}{\partial x_n}\right)_{\text{appr},\epsilon} (\mathbf{x}, \mathbf{h}_k) := \frac{F(\mathbf{x} + \epsilon \mathbf{e}_n, \mathbf{h}_k, tsc(\mathbf{x})) - F(\mathbf{x} - \epsilon \mathbf{e}_n, \mathbf{h}_k, tsc(\mathbf{x}))}{2\epsilon}. \tag{13.1.3}$$

Note that the computation of a tsc-file via NoSpherA2 requires quantum-mechanical computations, and here we only need the tsc-file at $\mathbf{x}$, and only the G_j-factors in (13.1.2) are evaluated at the modified models $\mathbf{x} \pm \epsilon\mathbf{e}_n$ corresponding to the n-th coordinate. It is useful to compare this with our numerical differentiation used for the computation of the "numerical" partial derivatives:

$$\left(\frac{\partial F}{\partial x_n}\right)_{\mathrm{num},\epsilon}(\mathbf{x},\mathbf{h}_k) := \frac{F(\mathbf{x}+\epsilon\mathbf{e}_n,\mathbf{h}_k,tsc(\mathbf{x}+\epsilon\mathbf{e}_n)) - F(\mathbf{x}-\epsilon\mathbf{e}_n,\mathbf{h}_k,tsc(\mathbf{x}-\epsilon\mathbf{e}_n))}{2\epsilon}. \quad (13.1.4)$$

Here we need tsc-files at the two models $\mathbf{x} \pm \epsilon\mathbf{e}_n$, and these two models change for every individual partial derivative with $n \in \{1,\ldots,N\}$. This makes every numerical refinement step very time-consuming. One could halve the number of quantum mechanical calculations by replacing 13.1.4 with the forward or backward quotients. This may lead to less accurate derivatives but may still be worthwhile and sufficiently accurate, and is worth investigation in future.

In the following sections, we will measure differences between matrices by using the Frobenius norm

$$\|A\|_F := \left(\sum_{i,j} |A_{ij}|^2\right)^{1/2}$$

for a matrix A with entries A_{ij}. Note that in the special case of vectors $\mathbf{v} \in \mathbb{R}^N$ in Cartesian coordinates, the Frobenius norm simplifies to the Cartesian norm $\|\mathbf{v}\|_F = \|\mathbf{v}\| = (v_i^2)^{1/2}$.

13.1.5 Relative Step Size ϵ and Cartesian Step Size δ

The model and the partial derivatives are given in fractional (or relative) coordinates with respect to the unit cell. This provides a dilemma, as we can choose to take either a consistent step size in fractional space or Cartesian space, but it is frequently the case that consistent values in one corresponds to different values in the other. Due to the use of Cartesian distances in the quantum mechanical optimisation, I decided to keep the Cartesian step size consistent. However, when manipulating values within olex2 (and for the values of the derivatives), I needed to use the fractional step size. In accordance with the description of numerical differentiation, the relative step size is denoted ϵ, with its corresponding Cartesian step size in Å denoted δ. In my work, I choose δ and generate ϵ based on that.

To move from our chosen Cartesian step size δ to our utilised fractional step size ϵ, we must divide by the 'Cartesian length of the unit relative vector in the step direction', that is:

- If the shift is related to only one of $\mathbf{a}, \mathbf{b}$ or $\mathbf{c}$, ϵ is equal to δ divided by the length of the relevant direction. For example, if we are moving by $\delta = 10^{-3}$Å in the $\mathbf{a}$ direction, $\epsilon = 10^{-3}/a$.

- If the shift is related to multiple directions, geometry must be used to determine the relationship between ϵ and δ. For example, the nitrogen atom of ammonia may only move along the $\mathbf{a} + \mathbf{b} + \mathbf{c}$ line. As ammonia has a cubic cell (with angles between $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ at 90° and lengths $a = b = c$), $\delta = 10^{-3}$ Å gives rise to an $\epsilon = 10^{-3}/(\sqrt{3}a)$.

As we do not need to calculate ADP derivatives, we did not need to consider steps outside of this 3-dimensional space.

In order to find the most suitable choice of Cartesian step size δ, we investigate results for numerical differentiation with δ-choices from 10^{-1} Å to 10^{-10} Å.

We must first verify that numerical methods can reasonably emulate the current analytical methods, and then continue to determine the most effective δ choice for numerical refinement itself.

13.2 A Proof of Principle: Design Matrices and Shift Vectors at $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ Compared with Numerically-Calculated Values

We now move on to our tests. At the beginning of our investigations, we evaluate whether numerical refinement is even viable; whether it gives valid and sensible intermediate results before continuing with the far more complicated and time-consuming full refinement process.

Within a given refinement step, the design matrix $\tilde{D}(\mathbf{x})$ can be viewed as the first object derived from the theoretical structure factor and its partial derivatives, and the shift vector $s(\mathbf{x})$ as the end product of one least-square minimisation step. We focus in this subsection on these two objects and their dependence on the step size. The entries of the design matrix are given by

$$(\tilde{D}(\mathbf{x}))_{kn} = \frac{\partial \tilde{K} Y}{\partial x_n}(\mathbf{x}, \mathbf{h}_k),$$

(with Y as the theoretical intensities, $||F_c||^2$) and we denote the positional and ADP components of the Cartesian shift vectors henceforth by $s^{\text{pos,cart}}(\mathbf{x})$ and $s^{\text{ADP,cart}}(\mathbf{x})$.

13.2.1 Comparison of Numerical versus Analytical Results

The partial derivatives of the structure factor in spherical and approximate refinement in *olex2.refine* (using NoSpherA2 in the latter case) are calculated analytically assuming vanishing derivatives of the form factors. This analytic calculation of the partial derivatives can also be done via numerical differentiation. The comparison with analytical allows us to test the accuracy of numerical differentiation (without involving the quantum-mechanical software in derivative calculations) and to have a first look at the error trending. We carry out our comparison at the spherical optimum $\mathbf{x}_{\text{spher}}$ and the approximate optimum $\mathbf{x}_{\text{appr}}$

The design matrices in this subsection obtained via *numerical differentiation* of the structure factor (using step size δ) will be denoted by $\tilde{D}_{\bullet,\delta}$ with $\bullet \in \{\text{spher}, \text{appr}\}$. Similarly, we will use the notation $s_{\bullet,\delta}^{\text{pos,cart}}$ and $s_{\bullet,\delta}^{\text{ADP,cart}}$ for the corresponding Cartesian shift vector components derived from numerically differentiated structure factors.

Design Matrix Analysis

The differences between the internally analytically calculated design matrices $\tilde{D}_{\bullet}$, $\bullet \in \{\text{spher}, \text{appr}\}$, and their analogues $\tilde{D}_{\bullet,\delta}$ computed via numerical differentiation using step sizes

$$\delta \in \left\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}\right\}$$

at $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ are presented in Table 13.1, where we use the following notation: $\max(A)$ and $\max(\mathbf{v})$ denote the maximal entry of a matrix A and vector $\mathbf{v}$, respectively. Similarly, $\text{med}(A)$ and $\text{med}(\mathbf{v})$ denote the median of the entries of A and the vector $\mathbf{v}$. $|A|$ and $|\mathbf{v}|$ denote the matrix/vector whose entries are the absolute values of the corresponding entries of A and $\mathbf{v}$.

Design matrix	$\mathbf{x}_{\text{spher}}$		$\mathbf{x}_{\text{appr}}$	
differences	$\bullet = \text{spher}$	$\bullet = \text{appr}$	$\bullet = \text{spher}$	$\bullet = \text{appr}$
Epoxide				
$\lVert D_\bullet \rVert_F$	1.001e+06	1.030e+06	1.022e+06	1.042e+06
$\max(\lvert D_\bullet \rvert)$	1.393e+05	1.403e+05	1.448e+05	1.456e+05
$\text{med}(\lvert D_\bullet \rvert)$	1.965e+01	2.232e+01	1.607e+01	1.687e+01
$\delta = 10^{-1}$	1.365e+03	1.374e+03	1.413e+03	1.415e+03
$\delta = 10^{-2}$	1.389e+01	1.397e+01	1.439e+01	1.440e+01
$\delta = 10^{-3}$	1.389e-01	1.398e-01	1.439e-01	1.441e-01
$\delta = 10^{-4}$	1.389e-03	1.398e-03	1.439e-03	1.441e-03
$\delta = 10^{-5}$	1.342e-05	1.353e-05	1.424e-05	1.428e-05
$\delta = 10^{-6}$	2.950e-05	3.017e-05	4.487e-05	4.679e-05
$\delta = 10^{-7}$	3.167e-04	3.289e-04	3.807e-04	3.876e-04
$\delta = 10^{-8}$	2.746e-03	3.023e-03	3.120e-03	3.137e-03

Design matrix	$\mathbf{x}_{\mathrm{spher}}$		$\mathbf{x}_{\mathrm{appr}}$	
differences	$\bullet = \mathrm{spher}$	$\bullet = \mathrm{appr}$	$\bullet = \mathrm{spher}$	$\bullet = \mathrm{appr}$
$\delta = 10^{-9}$	3.467e-02	3.884e-02	2.861e-02	2.978e-02
$\delta = 10^{-10}$	3.988e-01	4.224e-01	3.126e-01	3.214e-01

Design matrix	$\mathbf{x}_{\text{spher}}$		$\mathbf{x}_{\text{appr}}$	
differences	$\bullet = \text{spher}$	$\bullet = \text{appr}$	$\bullet = \text{spher}$	$\bullet = \text{appr}$
Ammonia				
$\|D_\bullet\|_F$	1.069e+05	1.209e+05	1.014e+05	1.087e+05
$\max(\|D_\bullet\|)$	4.340e+04	4.941e+04	4.128e+04	4.526e+04
$\text{med}(\|D_\bullet\|)$	2.870e+01	3.286e+01	1.961e+01	2.056e+01
$\delta = 10^{-1}$	9.766e+01	1.095e+02	9.446e+01	9.817e+01
$\delta = 10^{-2}$	9.859e-01	1.105e+00	9.540e-01	9.909e-01
$\delta = 10^{-3}$	9.860e-03	1.105e-02	9.541e-03	9.910e-03
$\delta = 10^{-4}$	9.859e-05	1.105e-04	9.541e-05	9.911e-05
$\delta = 10^{-5}$	9.716e-07	1.009e-06	1.043e-06	1.121e-06
$\delta = 10^{-6}$	2.004e-06	1.588e-06	1.145e-06	1.605e-06
$\delta = 10^{-7}$	9.050e-06	1.517e-05	2.713e-05	2.699e-05
$\delta = 10^{-8}$	1.393e-04	2.363e-04	1.410e-04	1.380e-04
$\delta = 10^{-9}$	1.699e-03	1.965e-03	8.971e-04	1.596e-03
$\delta = 10^{-10}$	3.945e-02	2.464e-02	1.624e-02	1.450e-02
L-Alanine				
$\|D_\bullet\|_F$	1.046e+07	1.047e+07	1.076e+07	1.078e+07
$\max(\|D_\bullet\|)$	9.520e+05	1.027e+06	9.435e+05	1.013e+06
$\text{med}(\|D_\bullet\|)$	9.106e+01	1.000e+02	6.720e+01	7.081e+01
$\delta = 10^{-1}$	1.873e+04	1.861e+04	1.943e+04	1.931e+04
$\delta = 10^{-2}$	1.962e+02	1.949e+02	2.036e+02	2.023e+02
$\delta = 10^{-3}$	1.963e+00	1.950e+00	2.037e+00	2.024e+00
$\delta = 10^{-4}$	1.963e-02	1.950e-02	2.037e-02	2.024e-02
$\delta = 10^{-5}$	1.952e-04	1.940e-04	2.035e-04	2.022e-04
$\delta = 10^{-6}$	9.440e-05	9.668e-05	1.053e-04	1.039e-04
$\delta = 10^{-7}$	8.988e-04	8.977e-04	9.057e-04	9.409e-04
$\delta = 10^{-8}$	8.652e-03	8.756e-03	1.003e-02	1.033e-02
$\delta = 10^{-9}$	1.031e-01	1.068e-01	1.059e-01	1.079e-01
$\delta = 10^{-10}$	8.156e-01	8.209e-01	8.544e-01	9.133e-01

Table 13.1: Comparison of design matrices: The entries in the δ-rows are $\|\tilde{D}_\bullet - \tilde{D}_{\bullet,\delta}\|_F$ with $\bullet \in \{\text{spher}, \text{appr}\}$. Particularly small differences are highlighted in green.

We see that the best agreement is achieved at the step size $\delta = 10^{-5}$. This table also clearly highlights the trending of the errors. For the larger δ-values the *truncation error* dominates which is proportional to δ^2 (decreasing the δ-values by 10 leads to error decay by an order of 10^2 in the upper half of the table). As $\delta > 0$ becomes smaller, the *roundoff error* becomes increasingly significant, and it is proportional to $1/\delta$ (reflected in the fact that the error increases roughly by an order of 10 in the lower half of the table). This matches very well with the error bound given in (13.1.1). Modelling the total error as $a\delta^2 + \frac{b}{\delta}$, then the minimum should be at $\delta = \sqrt[3]{\frac{b}{2a}}$ (see also [7, p. 181]). For ammonia, Choosing $a = 10^4$ and $b = 4 \times 10^{-12}$, we obtain $\delta = 6 \times 10^{-6}$, which lies between the two lowest recorded errors in the table. Similarly, for epoxide we have $a = 1.4 \times 10^5, b = 3.2 \times 10^{-11}$, giving an optimal δ of 5×10^{-6}. For L-alanine, $a = 10^6, b = 7^{-11}$, giving $\delta = 3 \times 10^{-6}$ for the minimum.

Shift Vector Analysis

The impact of the step size on the shift vectors in Cartesian coordinates is presented in Table 13.2. We consider their positional and ADP components separately. These results suggest that the best choice for the step size δ in this case lies somewhere between 10^{-5} and 10^{-6}, and the error pattern is again very clear.

Shift	$\mathbf{x}_{\text{spher}}$				$\mathbf{x}_{\text{appr}}$			
vector	$\bullet =$ spher		$\bullet =$ appr		$\bullet =$ spher		$\bullet =$ appr	
differences	$* =$ pos	$* =$ adp	$* =$ pos	$* =$ adp	$* =$ pos	$* =$ adp	$* =$ pos	$* =$ adp
epoxide								
$\lVert s_\bullet^{*,\text{cart}} \rVert$	2.234e-06	3.970e-06	2.388e-01	9.245e-02	2.485e-01	1.038e-01	3.392e-06	2.884e-06
$\max\left(\lvert s_\bullet^{*,\text{cart}} \rvert\right)$	1.698e-06	1.911e-06	1.006e-01	3.719e-02	9.625e-02	4.617e-02	2.318e-06	1.374e-06
$\operatorname{med}\left(\lvert s_\bullet^{*,\text{cart}} \rvert\right)$	1.645e-07	9.960e-08	2.958e-02	2.799e-03	2.498e-02	2.678e-03	8.843e-08	7.439e-08
$\delta = 10^{-1}$	1.139e-03	5.280e-04	5.932e-03	1.521e-03	5.035e-03	1.364e-03	8.041e-04	3.351e-04
$\delta = 10^{-2}$	1.118e-05	5.135e-06	5.921e-05	1.531e-05	5.031e-05	1.369e-05	7.864e-06	3.299e-06
$\delta = 10^{-3}$	1.118e-07	5.133e-08	5.921e-07	1.531e-07	5.031e-07	1.369e-07	7.862e-08	3.299e-08
$\delta = 10^{-4}$	1.118e-09	5.133e-10	5.921e-09	1.531e-09	5.032e-09	1.369e-09	7.862e-10	3.299e-10
$\delta = 10^{-5}$	1.141e-11	5.338e-12	5.891e-11	1.527e-11	5.128e-11	1.251e-11	8.323e-12	3.503e-12
$\delta = 10^{-6}$	5.484e-11	9.144e-12	1.032e-10	3.929e-11	1.355e-10	3.465e-11	1.963e-11	5.575e-12
$\delta = 10^{-7}$	4.257e-10	1.016e-10	9.985e-10	3.668e-10	1.172e-09	4.993e-10	1.424e-10	3.836e-11
$\delta = 10^{-8}$	2.996e-09	9.323e-10	1.113e-08	1.968e-09	1.101e-08	4.420e-09	1.734e-09	3.308e-10
$\delta = 10^{-9}$	3.355e-08	7.637e-09	6.796e-08	2.472e-08	1.048e-07	3.964e-08	1.394e-08	3.491e-09
$\delta = 10^{-10}$	4.043e-07	7.570e-08	1.168e-06	3.088e-07	1.349e-06	3.973e-07	2.841e-07	8.589e-08
Ammonia								
$\lVert s_\bullet^{*,\text{cart}} \rVert$	2.966e-06	2.244e-06	1.769e-01	5.391e-02	1.679e-01	3.944e-02	3.370e-06	1.862e-06
$\max\left(\lvert s_\bullet^{*,\text{cart}} \rvert\right)$	2.559e-06	1.335e-06	1.285e-01	3.170e-02	1.098e-01	2.825e-02	2.646e-06	1.189e-06
$\operatorname{med}\left(\lvert s_\bullet^{*,\text{cart}} \rvert\right)$	3.024e-07	1.068e-07	3.688e-02	1.183e-03	4.326e-02	1.652e-03	1.030e-07	2.333e-08
$\delta = 10^{-1}$	1.174e-03	5.941e-04	3.349e-03	6.949e-04	5.134e-03	1.059e-03	1.175e-03	4.758e-04
$\delta = 10^{-2}$	1.144e-05	5.870e-06	3.469e-05	6.434e-06	5.111e-05	1.062e-05	1.128e-05	4.681e-06
$\delta = 10^{-3}$	1.144e-07	5.870e-08	3.470e-07	6.429e-08	5.111e-07	1.062e-07	1.127e-07	4.680e-08
$\delta = 10^{-4}$	1.144e-09	5.870e-10	3.470e-09	6.429e-10	5.111e-09	1.062e-09	1.127e-09	4.680e-10
$\delta = 10^{-5}$	1.127e-11	5.864e-12	3.633e-11	6.760e-12	5.117e-11	1.126e-11	1.179e-11	4.938e-12

Shift	$\mathbf{x}_{\text{spher}}$				$\mathbf{x}_{\text{appr}}$			
vector	$\bullet = \text{spher}$		$\bullet = \text{appr}$		$\bullet = \text{spher}$		$\bullet = \text{appr}$	
differences	$* = \text{pos}$	$* = \text{adp}$	$* = \text{pos}$	$* = \text{adp}$	$* = \text{pos}$	$* = \text{adp}$	$* = \text{pos}$	$* = \text{adp}$
$\delta = 10^{-6}$	7.758e-12	2.554e-12	8.126e-12	1.064e-11	4.062e-11	9.602e-12	3.496e-12	1.227e-12
$\delta = 10^{-7}$	2.497e-11	1.479e-11	2.695e-10	1.117e-10	2.124e-10	8.453e-11	1.399e-11	7.536e-12
$\delta = 10^{-8}$	4.127e-10	2.205e-10	2.052e-09	9.564e-10	2.437e-09	1.407e-09	4.251e-10	1.365e-10
$\delta = 10^{-9}$	9.154e-09	3.867e-09	4.026e-08	1.015e-08	1.952e-08	7.339e-09	4.065e-09	9.963e-10
$\delta = 10^{-10}$	2.254e-08	2.835e-09	2.231e-07	6.507e-08	2.228e-07	8.545e-08	4.945e-08	2.088e-08
L-Alanine								
$\lVert s_{\bullet}^{*,\text{cart}} \rVert$	1.152e-05	1.700e-05	2.795e-01	1.359e-01	2.569e-01	1.034e-01	2.346e-06	2.095e-06
$\max\left(\lvert s_{\bullet}^{*,\text{cart}} \rvert\right)$	8.666e-06	1.453e-05	9.919e-02	7.959e-02	1.015e-01	4.121e-02	1.711e-06	1.075e-06
$\text{med}\left(\lvert s_{\bullet}^{*,\text{cart}} \rvert\right)$	6.178e-08	3.168e-08	3.329e-03	2.226e-03	1.191e-02	1.491e-03	3.081e-08	8.054e-09
$\delta = 10^{-1}$	2.692e-03	1.108e-03	8.716e-03	1.187e-03	6.175e-03	1.411e-03	9.882e-04	4.318e-04
$\delta = 10^{-2}$	2.570e-05	1.030e-05	8.833e-05	1.205e-05	6.149e-05	1.378e-05	9.774e-06	4.070e-06
$\delta = 10^{-3}$	2.569e-07	1.030e-07	8.834e-07	1.205e-07	6.149e-07	1.377e-07	9.773e-08	4.068e-08
$\delta = 10^{-4}$	2.570e-09	1.030e-09	8.834e-09	1.205e-09	6.149e-09	1.377e-09	9.775e-10	4.068e-10
$\delta = 10^{-5}$	2.436e-11	9.282e-12	8.894e-11	1.261e-11	5.971e-11	1.495e-11	9.839e-12	3.973e-12
$\delta = 10^{-6}$	2.246e-10	5.155e-11	8.804e-11	3.960e-11	1.696e-10	3.889e-11	4.445e-11	8.016e-12
$\delta = 10^{-7}$	2.332e-09	5.101e-10	7.137e-10	3.593e-10	1.806e-09	4.765e-10	5.005e-10	8.882e-11
$\delta = 10^{-8}$	1.664e-08	3.716e-09	8.170e-09	2.501e-09	2.156e-08	4.806e-09	3.984e-09	7.622e-10
$\delta = 10^{-9}$	9.230e-08	1.766e-08	7.834e-08	3.986e-08	1.659e-07	5.032e-08	3.783e-08	7.372e-09
$\delta = 10^{-10}$	1.476e-06	3.429e-07	1.329e-06	5.219e-07	1.795e-06	4.391e-07	4.801e-07	7.703e-08

Table 13.2: Comparison of shift vectors: The entries in the δ-rows are $\lVert s_{\bullet}^{*,\text{cart}} - s_{\bullet,\delta}^{*,\text{cart}} \rVert$ with $\bullet \in \{\text{spher}, \text{appr}\}$ and $* \in \{\text{pos}, \text{adp}\}$. Particularly small differences are highlighted in green.

13.2.2 Comparison with Different δ-values Within Numerical Nonspherical Approximation

In the previous subsection, we needed to choose $\delta \geq 10^{-6}$ to avoid a significant impact by the *roundoff errors.* In the case of numerical refinement, we do not have analytical values for partial derivatives of the structure factor and we can only compute them via the numerical differentiation given in (13.1.4). Our focus will be again on the derived design matrices and shift vectors. Since an *analytically* computed numerical design matrix is not available, we substitute it by a median design matrix $\tilde{D}_{\text{num,med}}$: each entry of $\tilde{D}_{\text{num,med}}$ is the median of the corresponding entries of the numerically computed matrices $\tilde{D}_{\text{num},\delta}$ where δ runs through the ten step sizes from 10^{-1} to 10^{-10}. As before with the analytical design matrix, we individually compare the differences of each matrix $\tilde{D}_{\text{num},\delta}$ for $\delta \in \{10^{-1}, \ldots, 10^{-10}\}$ with this fixed median matrix $\tilde{D}_{\text{num,med}}$. We also carry out analogous investigations for the positional and ADP components of the Cartesian shift vectors. The results are presented in Table 13.3.

Sadly, this does not give much information on the matrices and in fact it is difficult to evaluate the impact when they have a very large variance in their values and a vast majority of the entries are very close to 0. Especially for the design matrices, differences are dominated by those large values even on the most similar design matrices. Whilst this gives a decent approximation for viable values of δ, our main analysis uses other evaluations such as wR_2drops.

Differences with	$\mathbf{x}_{\text{spher}}$			$\mathbf{x}_{\text{appr}}$		
medians at x_{spher}	$X = \tilde{D}$	$X = s^{\text{pos,cart}}$	$X = s^{\text{ADP,cart}}$	$X = \tilde{D}$	$X = s^{\text{pos,cart}}$	$X = s^{\text{ADP,cart}}$
epoxide						
$\lVert X_{\text{num,med}} \rVert_F$	1.030e+06	2.078e-01	6.463e-02	1.042e+06	5.261e-03	1.471e-03
$\max(\lvert X_{\text{num,med}} \rvert)$	1.403e+05	8.718e-02	2.763e-02	1.456e+05	2.965e-03	7.808e-04
$\text{med}(\lvert X_{\text{num,med}} \rvert)$	2.349e+01	2.205e-02	1.734e-03	1.754e+01	1.781e-04	6.015e-05
$\delta = 10^{-1}$	1.427e+03	5.976e-03	2.076e-03	1.412e+03	1.025e-03	3.930e-04
$\delta = 10^{-2}$	3.895e+02	1.787e-03	1.267e-03	1.730e+02	9.239e-04	1.342e-04
$\delta = 10^{-3}$	3.703e+02	2.183e-03	1.188e-03	1.804e+02	9.027e-04	1.286e-04
$\delta = 10^{-4}$	3.221e+02	2.132e-03	1.001e-03	1.994e+02	8.618e-04	1.280e-04
$\delta = 10^{-5}$	6.546e+02	3.377e-03	1.579e-03	6.100e+02	1.674e-03	3.108e-04
$\delta = 10^{-6}$	2.484e+03	7.032e-03	4.069e-03	7.579e+02	1.966e-03	6.021e-04
$\delta = 10^{-7}$	1.568e+03	8.172e-03	4.438e-03	2.253e+03	7.326e-03	1.963e-03
$\delta = 10^{-8}$	6.851e+03	6.752e-02	1.932e-02	1.183e+04	2.152e-02	4.847e-03
$\delta = 10^{-9}$	5.213e+04	1.761e-01	4.085e-02	8.080e+04	1.362e-02	1.250e-02
$\delta = 10^{-10}$	2.481e+05	2.062e-01	5.228e-02	5.076e+05	5.317e-03	1.027e-02
Ammonia						
$\lVert X_{\text{num,med}} \rVert_F$	1.210e+05	1.630e-01	4.186e-02	1.088e+05	9.822e-03	3.692e-03
$\max(\lvert X_{\text{num,med}} \rvert)$	4.941e+04	1.199e-01	2.585e-02	4.526e+04	7.160e-03	2.492e-03
$\text{med}(\lvert X_{\text{num,med}} \rvert)$	3.353e+01	3.279e-02	1.424e-03	2.202e+01	2.410e-04	2.906e-04
$\delta = 10^{-1}$	1.166e+02	7.482e-03	1.414e-03	1.108e+02	4.171e-04	3.043e-04
$\delta = 10^{-2}$	3.704e+01	2.041e-03	9.190e-04	2.243e+01	1.903e-04	4.329e-05
$\delta = 10^{-3}$	3.491e+01	1.972e-03	8.703e-04	2.221e+01	2.273e-04	6.535e-05
$\delta = 10^{-4}$	5.591e+01	5.273e-04	9.711e-04	2.831e+01	2.869e-04	1.045e-04
$\delta = 10^{-5}$	7.272e+01	2.351e-03	1.950e-03	7.544e+01	1.721e-04	1.767e-04
$\delta = 10^{-6}$	1.481e+02	4.688e-04	5.895e-04	1.011e+02	5.941e-04	1.382e-04
$\delta = 10^{-7}$	1.286e+02	1.204e-03	2.295e-03	1.555e+02	4.645e-04	2.962e-04

Differences with		$\mathbf{x}_{\text{spher}}$			$\mathbf{x}_{\text{appr}}$	
medians at x_{spher}	$X = \tilde{D}$	$X = s^{\text{pos,cart}}$	$X = s^{\text{ADP,cart}}$	$X = \tilde{D}$	$X = s^{\text{pos,cart}}$	$X = s^{\text{ADP,cart}}$
$\delta = 10^{-8}$	7.850e+02	1.080e-02	1.248e-02	8.210e+02	1.716e-03	6.756e-04
$\delta = 10^{-9}$	6.232e+03	1.230e-01	5.354e-02	7.742e+03	5.007e-03	3.432e-03
$\delta = 10^{-10}$	8.117e+08	1.629e-01	1.019e-01	6.553e+04	9.078e-03	6.399e-03
L-Alanine						
$\lVert X_{\text{num,med}} \rVert_F$	1.047e+07	2.372e-01	1.147e-01	1.078e+07	7.056e-03	1.866e-03
$\max(\lvert X_{\text{num,med}} \rvert)$	1.027e+06	8.597e-02	7.270e-02	1.013e+06	3.149e-03	8.487e-04
$\text{med}(\lvert X_{\text{num,med}} \rvert)$	1.018e+02	2.346e-03	1.275e-03	7.241e+01	3.643e-04	2.049e-05
$\delta = 10^{-1}$	1.850e+04	7.604e-03	2.208e-03	1.919e+04	9.750e-04	4.460e-04
$\delta = 10^{-2}$	1.223e+03	2.975e-03	1.575e-03	1.021e+03	5.067e-04	2.092e-04
$\delta = 10^{-3}$	1.240e+03	3.036e-03	1.558e-03	1.004e+03	4.750e-04	2.081e-04
$\delta = 10^{-4}$	1.323e+03	3.901e-03	1.602e-03	1.072e+03	6.084e-04	2.018e-04
$\delta = 10^{-5}$	2.529e+03	4.207e-03	3.028e-03	1.844e+03	1.790e-03	4.132e-04
$\delta = 10^{-6}$	5.450e+03	1.485e-02	5.648e-03	4.682e+03	4.876e-03	8.102e-04
$\delta = 10^{-7}$	9.797e+03	1.175e-02	8.702e-03	1.975e+04	1.364e-02	4.951e-03
$\delta = 10^{-8}$	2.828e+04	8.296e-02	4.441e-02	1.937e+05	2.885e-02	7.378e-03
$\delta = 10^{-9}$	2.090e+05	2.177e-01	5.696e-02	2.012e+05	1.378e-02	9.480e-03
$\delta = 10^{-10}$	1.416e+06	2.369e-01	6.540e-02	1.229e+06	6.823e-03	1.402e-02

Table 13.3: Comparison with the medians: The entries in the δ-rows are $\lVert X_{\text{num,med}} - X_{\text{num},\delta} \rVert_F$ for $X \in \{D, s^{\text{pos,cart}}, s^{\text{ADP,cart}}\}$. Particularly small differences are highlighted in green.

We first note that the numerical shift vector components at $\mathbf{x}_{\text{appr}}$ are smaller than at $\mathbf{x}_{\text{spher}}$ (by more than a factor of 10). This is not surprising since we expect $\mathbf{x}_{\text{appr}}$ to be a better approximation of the numerical model than $\mathbf{x}_{\text{spher}}$. With regards to the step size $\delta > 0$, the results do not provide a clear indication for the best choice. While the *roundoff errors* significantly increase for decreasing step sizes from 10^{-6} to 10^{-10} and $\delta = 10^{-1}$ is definitely not in the competition, there is no obvious "winner" amongst the other δ-choices between 10^{-2} to 10^{-6}. The picture will become much clearer in the following subsections, where we investigate the numerical refinement journeys from $\mathbf{x}_{\text{appr}}$ to the numerical optimum $\mathbf{x}_{\text{num},\delta}$ via different step sizes $\delta > 0$.

We finally note the following additional consideration; the number of digits stored within Olex2 and transferred to the quantum mechanical software is truncated. The atomic locations are stored as relative values with 8 decimal places, and thus only shifts above 10^{-8}Å to 10^{-6}Å (dependant on the unit cell lengths of the structure) will provide a reasonable estimation in cases where we involve the external quantum mechanical software, as we do for the numerical determination of the derivative.

13.3 The Quality of Numerical Nonspherical Refinement

This section is concerned with the quality of numerical nonspherical refinement. In particular, we test it with regards to the following criteria:

Convergence When numerical nonspherical refinement has reached a 'minimum', the model should no longer fluctuate - future shift vectors should be sufficiently small

Improvement numerical nonspherical refinement should result in a smaller wR$_2$-factor than spherical or approximate nonspherical refinement

Consistency The minimum should be independent of the start model used for the numerical nonspherical refinement.

Henceforth, we will use the following notation: starting with an initial model $\mathbf{x}_0 = \mathbf{x}_{\text{spher}}$ or $\mathbf{x}_0 = \mathbf{x}_{\text{appr}}$, numerical nonspherical refinement iteratively generates a sequence $\mathbf{x}_j$ of 20 models via $\mathbf{x}_{j+1} = \mathbf{x}_j + \mathbf{s}_j$, where s_j is the j-th shift vector. Then $\mathbf{x}_{\text{num}}$ is the model $\mathbf{x}_j$ with the minimal wR$_2$-factor in this sequence.

As we consider the effectiveness of numerical nonspherical refinement with regards to the above criteria, we must also select a viable choice δ for the step size of our numerical differentiation. The following investigations also help us to find such a potentially optimal step size.

13.3.1 Convergence

In Figures 13.4 to 13.6, we see the maximal shift/esd (see Section 6.9 in Part I for more details) for our examples over 20 refinement steps for each choice of $\delta \in \{10^{-1}\text{Å}, 10^{-2}\text{Å}, 10^{-3}\text{Å}, 10^{-4}\text{Å}, 10^{-5}\text{Å}, 10^{-6}\text{Å}\}$. The 20 refinement steps are represented by the horizontal axis. At step j, we divide each parameter in the shift

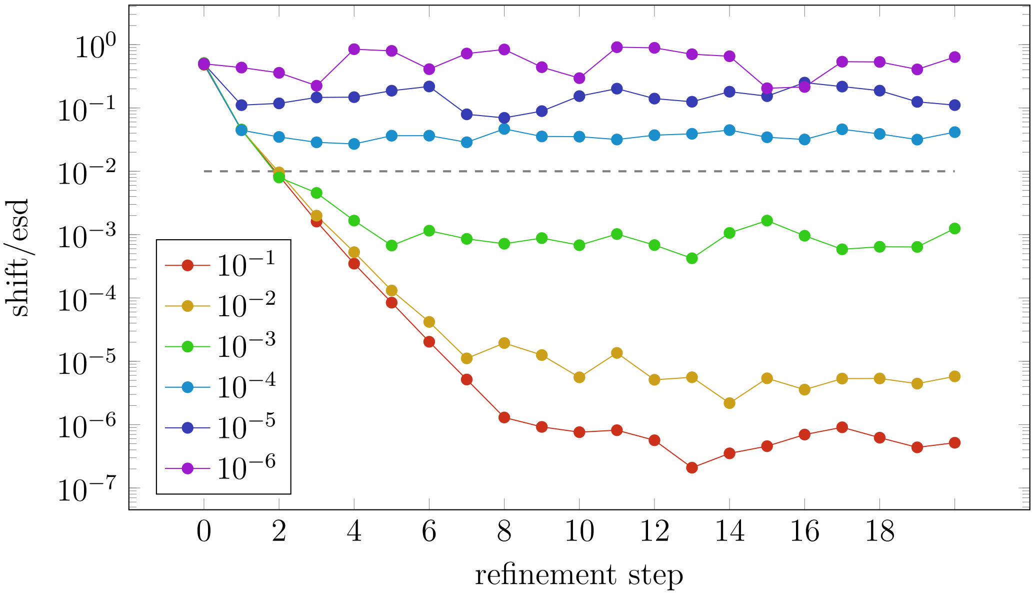


Figure 13.4: Maximal shift/esd for numerical nonspherical refinement starting from $\mathbf{x}_{\text{appr}}$ for $\delta = 10^{-1}$Å, ..., 10^{-6}Å (epoxide)

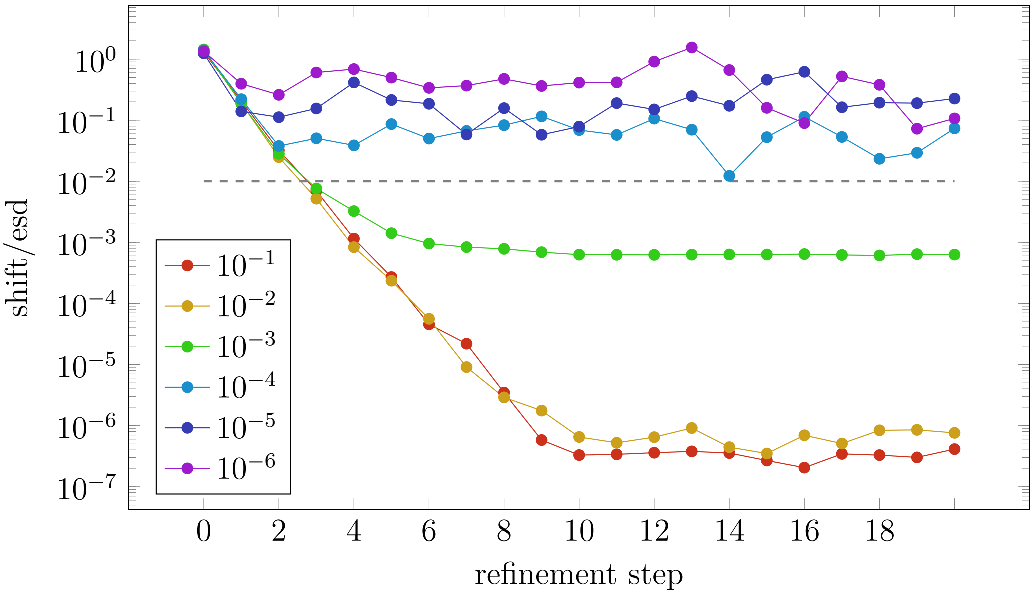


Figure 13.5: Maximal shift/esd for numerical nonspherical refinement starting from $\mathbf{x}_{\text{appr}}$ for $\delta = 10^{-1}$Å, ..., 10^{-6}Å (ammonia)

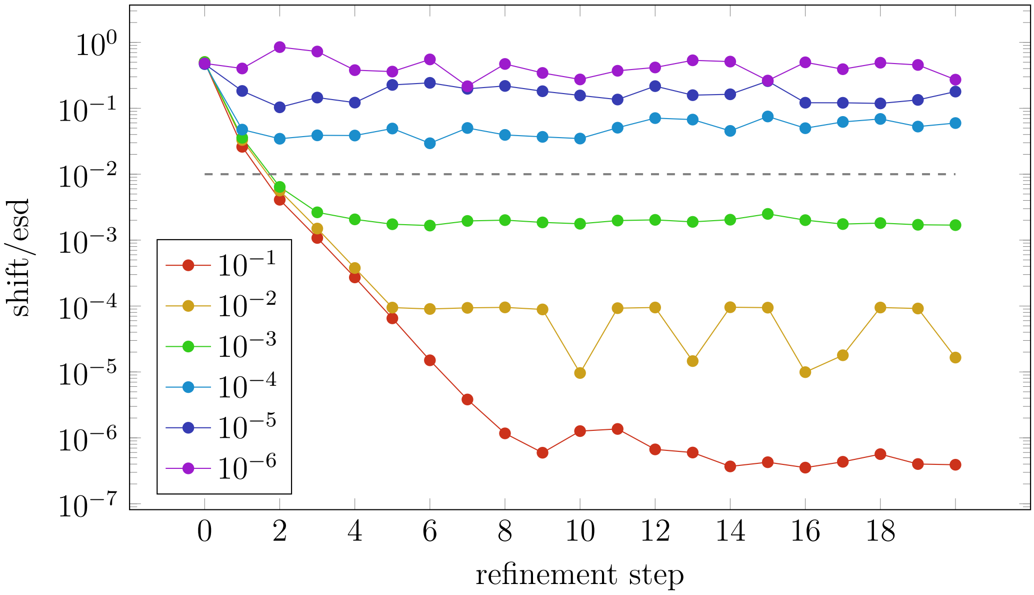


Figure 13.6: Maximal shift/esd for numerical nonspherical refinement starting from $\mathbf{x}_{\text{appr}}$ for $\delta = 10^{-1}$Å, ..., 10^{-6}Å (L-alanine)

vector $\mathbf{s}_j$ by the corresponding uncertainty of the model $\mathbf{x}_j$ and take the maximum over these quotients. This value shift/esd is represented vertically above each step.

If shift/esd is lower than 0.01 the model is considered settled as the shifts cannot cause a significant change to the parameters (compared to their uncertainties). Figures 13.4 to 13.6 show that this convergence threshold (represented by the dashed line) is not reached for $\delta \in \{10^{-4}\text{Å}, 10^{-5}\text{Å}, 10^{-6}\text{Å}\}$. For $\delta \in \{10^{-1}\text{Å}, 10^{-2}\text{Å}, 10^{-3}\text{Å}\}$, convergence is reached from the third step onwards. For such choices of δ, the criterion of **convergence** is achieved. As only these three δ values satisfy convergence, we may only choose one of them as our final δ. However, we continue considering all 6 tested values to allow viewing of the trends across them.

13.3.2 Improvement

Figures 13.7 to 13.15 show the wR$_2$-factors of numerical nonspherical refinement for our 3 molecules, with three figures for each molecule with increasing focus on the lowest wR$_2$-factors achieved (via dropping initial refinement steps and extreme δ-

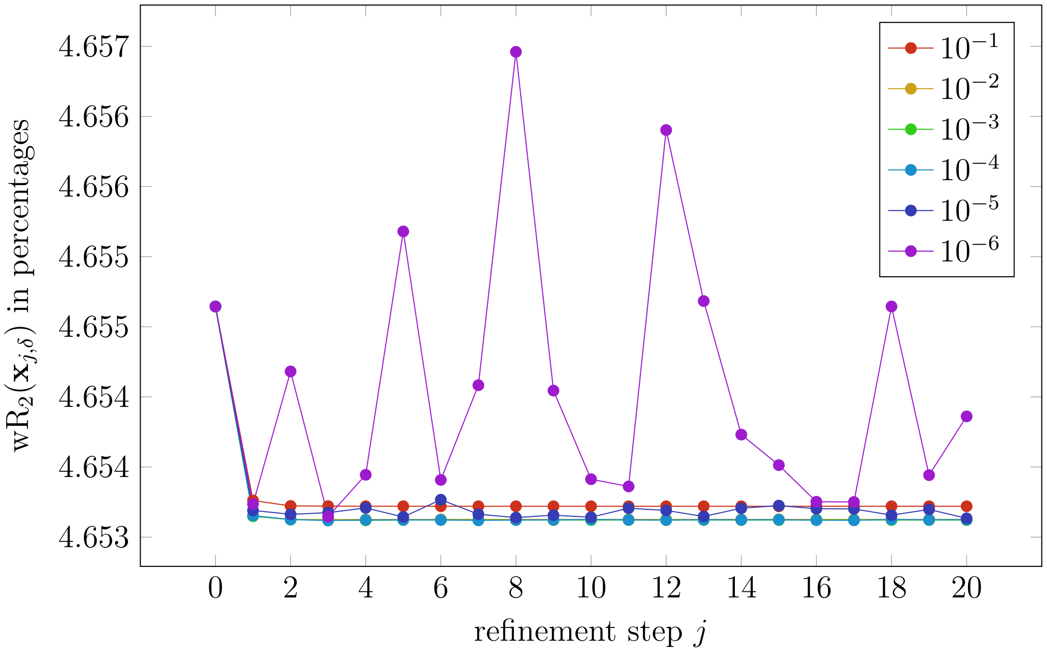


Figure 13.7: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ and for $\delta = 10^{-1}, \ldots, 10^{-6}$ (epoxide)

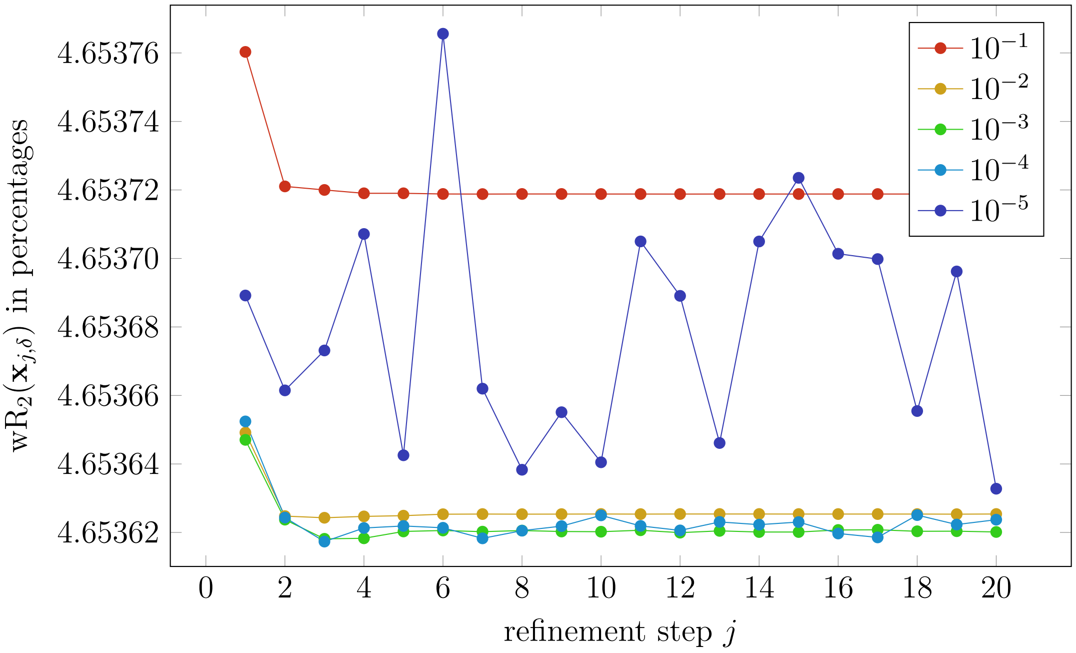


Figure 13.8: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 1, 2, \ldots, 20$ and for $\delta = 10^{-1}, \ldots, 10^{-5}$ (epoxide)

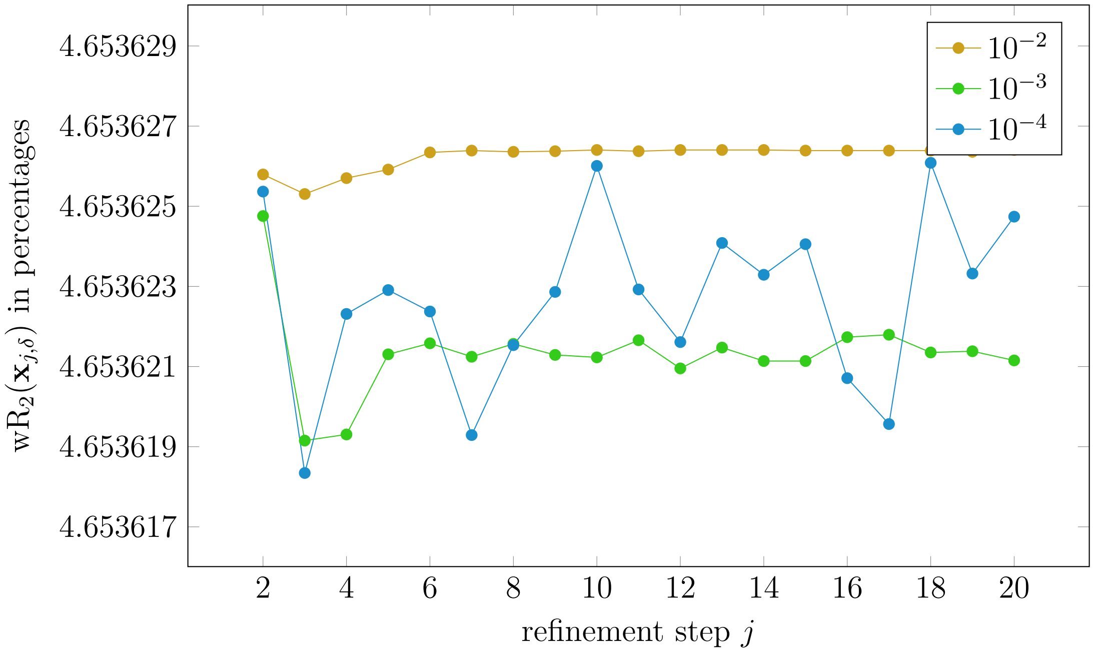


Figure 13.9: The progression of $wR_2(\mathbf{x}_j)$ for $j = 2, 3, \ldots, 20$ and for $\delta = 10^{-2}, \ldots, 10^{-4}$ (epoxide)

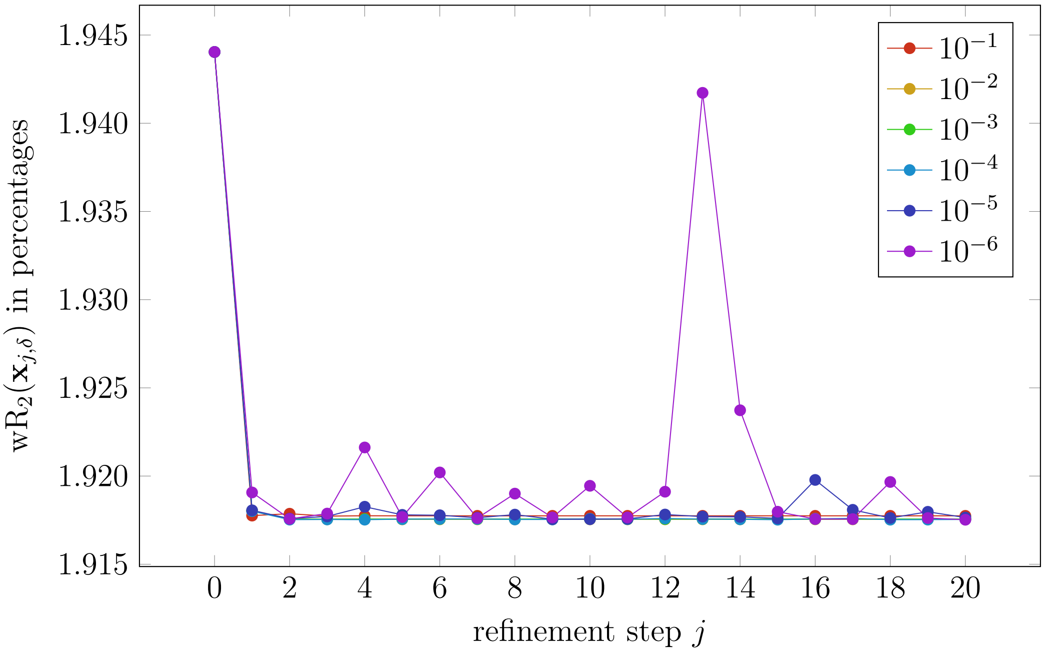


Figure 13.10: The progression of $wR_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ and for $\delta = 10^{-1}, \ldots, 10^{-6}$ (ammonia)

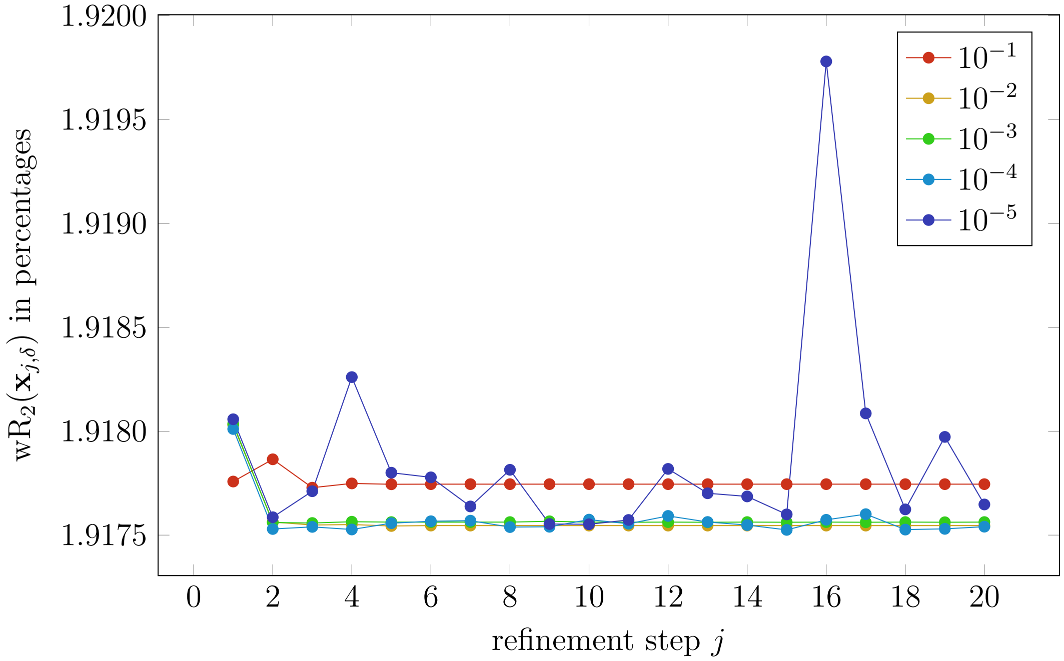


Figure 13.11: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 1, 2, \ldots, 20$ and for $\delta = 10^{-1}, \ldots, 10^{-5}$ (ammonia)

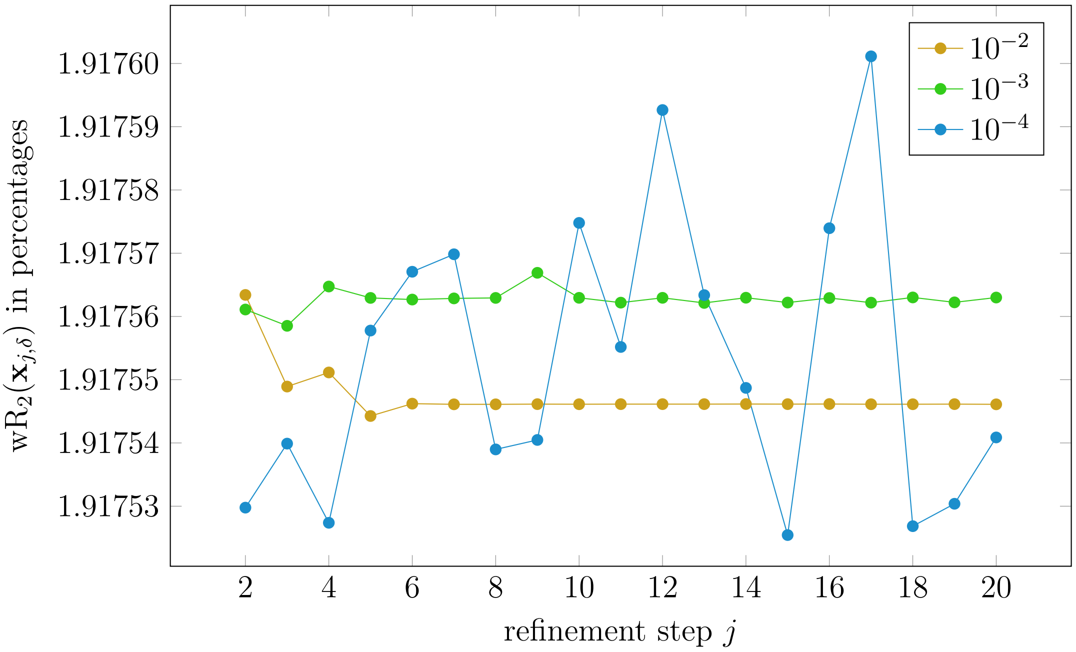


Figure 13.12: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 2, 3, \ldots, 20$ and for $\delta = 10^{-2}, \ldots, 10^{-4}$ (ammonia)

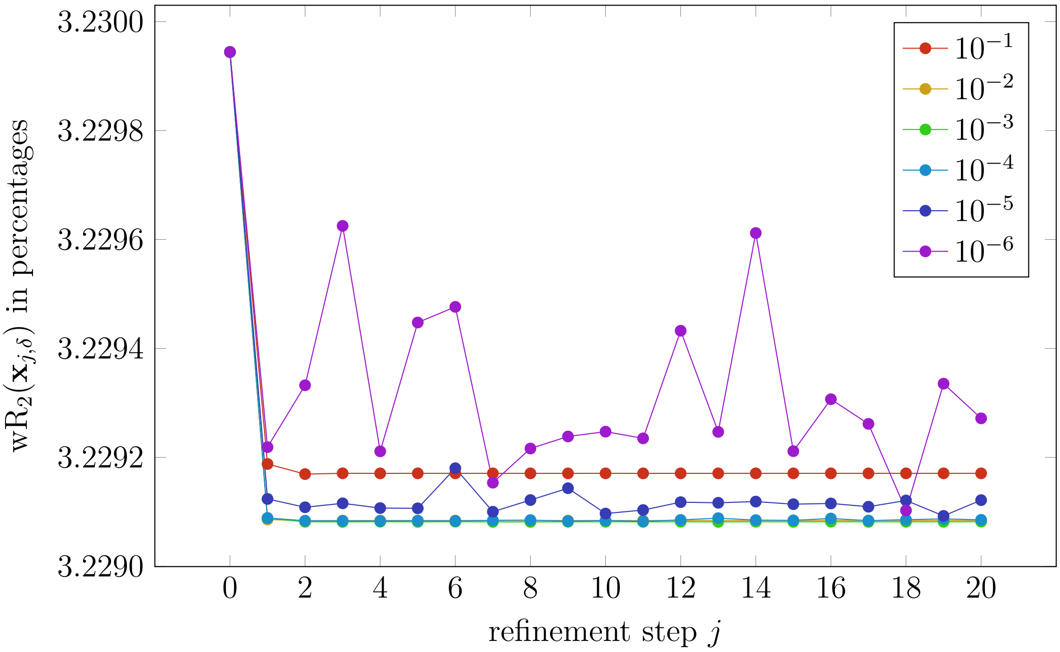


Figure 13.13: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ and for $\delta =$ $10^{-1}, \ldots, 10^{-6}$ (L-Alanine)

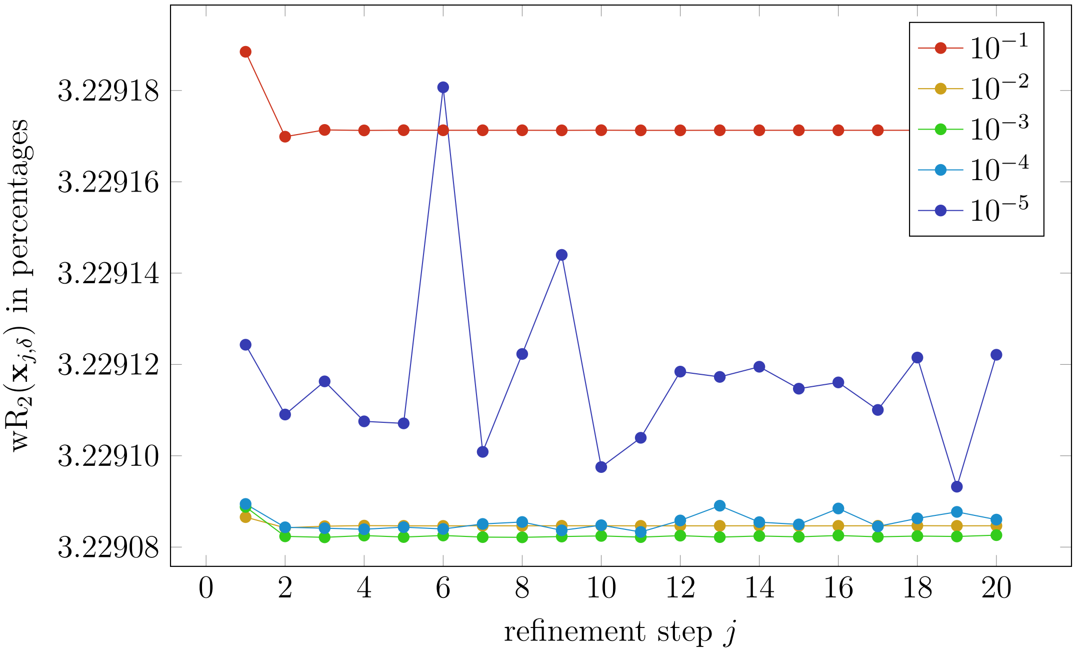


Figure 13.14: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 1, 2, \ldots, 20$ and for $\delta =$ $10^{-1}, \ldots, 10^{-5}$ (L-Alanine)

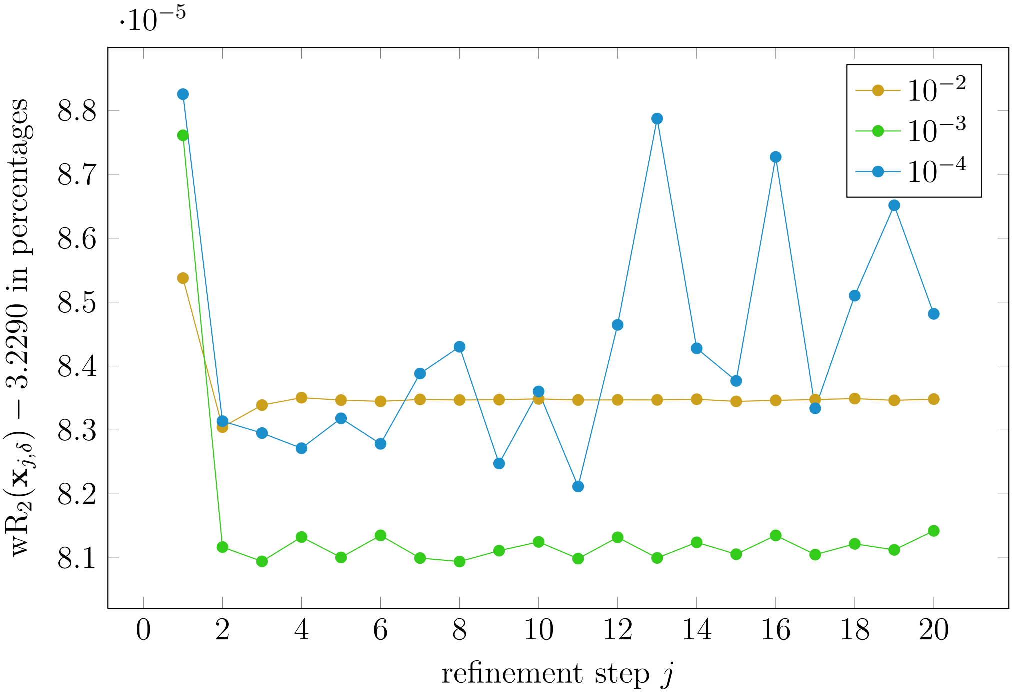


Figure 13.15: The progression of $wR_2(\mathbf{x}_j)$ for $j = 1, 2, \dots, 20$ and for $\delta = 10^{-2}, \dots, 10^{-4}$ (L-Alanine)

values). We note that many of the points overlap, in particular all journeys have the same start point of 4.655% for epoxide, 1.944% for ammonia and 3.230% for alanine. In particular, $\delta = 10^{-2}$Å, 10^{-3}Å, 10^{-4}Å have very similar essentially horizontal wR2 progressions from step 2 onwards. Since the wR_2-factor decreases compared to the start value, we see that numerical nonspherical refinement provides **improvement** (compared to approximate nonspherical refinement). Figures 13.7, 13.10 and 13.13 also indicates that $\delta = 10^{-5}$Å and 10^{-6}Å provide results with larger wR_2-factor shifts later in the journey (outliers) and we consider these δ therefore as worse than the others. Such increased errors at low δ are typically related to errors in the storage of numbers. In our case, this could be related to the way that NoSpherA2 passes information, such as atomic positions, to the quantum mechanical program, and more accurate passing of information could allow smaller δ choices. Presently, we are limited by the 8-digit precision of the wfn file, which can lead to an error of up to 2×10^{-8}Å in the input to the quantum mechanical program.

Tables 13.4 to 13.9 show the differences between the optimal models of numerical

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	6.429e-04	6.586e-04	5.740e-04	7.520e-04	1.307e-03
$\delta_2 = 10^{-2}$	6.429e-04	0	1.060e-04	3.234e-04	6.423e-04	8.825e-04
$\delta_2 = 10^{-3}$	6.586e-04	1.060e-04	0	2.534e-04	6.578e-04	8.245e-04
$\delta_2 = 10^{-4}$	5.740e-04	3.234e-04	2.534e-04	0	6.774e-04	8.579e-04
$\delta_2 = 10^{-5}$	7.520e-04	6.423e-04	6.578e-04	6.774e-04	0	1.132e-03
$\delta_2 = 10^{-6}$	1.307e-03	8.825e-04	8.245e-04	8.579e-04	1.132e-03	0
approximate	6.303e-03	6.259e-03	6.262e-03	6.326e-03	6.560e-03	6.656e-03

Table 13.4: The difference in final model $\left|\mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_2}\right|$ in epoxide in Å.

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	3.541e-04	3.665e-04	3.567e-04	2.894e-04	6.344e-04
$\delta_2 = 10^{-2}$	3.541e-04	0	3.132e-05	8.148e-05	2.352e-04	3.666e-04
$\delta_2 = 10^{-3}$	3.665e-04	3.132e-05	0	7.082e-05	2.377e-04	3.471e-04
$\delta_2 = 10^{-4}$	3.567e-04	8.148e-05	7.082e-05	0	2.394e-04	3.768e-04
$\delta_2 = 10^{-5}$	2.894e-04	2.352e-04	2.377e-04	2.394e-04	0	4.994e-04
$\delta_2 = 10^{-6}$	6.344e-04	3.666e-04	3.471e-04	3.768e-04	4.994e-04	0
approximate	1.939e-03	1.790e-03	1.792e-03	1.815e-03	1.904e-03	1.713e-03

Table 13.5: The difference in final model $\left|\mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_2}\right|$ in Epoxide in Å^2.

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	7.526e-04	7.559e-04	5.684e-04	9.401e-04	8.563e-04
$\delta_2 = 10^{-2}$	7.526e-04	0	8.574e-05	2.515e-04	4.376e-04	1.939e-04
$\delta_2 = 10^{-3}$	7.559e-04	8.574e-05	0	2.772e-04	4.980e-04	1.852e-04
$\delta_2 = 10^{-4}$	5.684e-04	2.515e-04	2.772e-04	0	5.975e-04	3.084e-04
$\delta_2 = 10^{-5}$	9.401e-04	4.376e-04	4.980e-04	5.975e-04	0	5.957e-04
$\delta_2 = 10^{-6}$	8.563e-04	1.939e-04	1.852e-04	3.084e-04	5.957e-04	0
approximate	7.893e-03	8.608e-03	8.598e-03	8.379e-03	8.815e-03	8.641e-03

Table 13.6: The difference in final model $\left|\mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_2}\right|$ in Ammonia in Å.

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	2.888e-04	3.087e-04	2.211e-04	3.265e-04	3.102e-04
$\delta_2 = 10^{-2}$	2.888e-04	0	3.259e-05	9.135e-05	1.335e-04	4.988e-05
$\delta_2 = 10^{-3}$	3.087e-04	3.259e-05	0	1.076e-04	1.539e-04	5.285e-05
$\delta_2 = 10^{-4}$	2.211e-04	9.135e-05	1.076e-04	0	1.978e-04	9.549e-05
$\delta_2 = 10^{-5}$	3.265e-04	1.335e-04	1.539e-04	1.978e-04	0	1.653e-04
$\delta_2 = 10^{-6}$	3.102e-04	4.988e-05	5.285e-05	9.549e-05	1.653e-04	0
approximate	2.646e-03	2.907e-03	2.918e-03	2.820e-03	2.971e-03	2.910e-03

Table 13.7: The difference in final model $\left|\mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_2}\right|$ in Ammonia in Å^2.

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	8.370e-04	8.807e-04	8.877e-04	7.625e-04	1.390e-03
$\delta_2 = 10^{-2}$	8.370e-04	0	1.553e-04	4.561e-04	8.823e-04	9.830e-04
$\delta_2 = 10^{-3}$	8.807e-04	1.553e-04	0	4.321e-04	8.670e-04	1.004e-03
$\delta_2 = 10^{-4}$	8.877e-04	4.561e-04	4.321e-04	0	7.788e-04	1.089e-03
$\delta_2 = 10^{-5}$	7.625e-04	8.823e-04	8.670e-04	7.788e-04	0	1.501e-03
$\delta_2 = 10^{-6}$	1.390e-03	9.830e-04	1.004e-03	1.089e-03	1.501e-03	0
approximate	7.549e-03	7.422e-03	7.468e-03	7.553e-03	7.686e-03	7.288e-03

Table 13.8: The difference in final model $\left|\mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{pos,cart}}_{\text{num},\delta_2}\right|$ in L-Alanine in Å.

	$\delta_1 = 10^{-1}$	$\delta_1 = 10^{-2}$	$\delta_1 = 10^{-3}$	$\delta_1 = 10^{-4}$	$\delta_1 = 10^{-5}$	$\delta_1 = 10^{-6}$
$\delta_2 = 10^{-1}$	0	4.429e-04	4.801e-04	5.091e-04	4.833e-04	5.818e-04
$\delta_2 = 10^{-2}$	4.429e-04	0	8.166e-05	1.372e-04	2.398e-04	2.656e-04
$\delta_2 = 10^{-3}$	4.801e-04	8.166e-05	0	8.823e-05	2.159e-04	2.488e-04
$\delta_2 = 10^{-4}$	5.091e-04	1.372e-04	8.823e-05	0	1.751e-04	2.474e-04
$\delta_2 = 10^{-5}$	4.833e-04	2.398e-04	2.159e-04	1.751e-04	0	3.542e-04
$\delta_2 = 10^{-6}$	5.818e-04	2.656e-04	2.488e-04	2.474e-04	3.542e-04	0
approximate	2.058e-03	1.988e-03	2.043e-03	2.088e-03	2.156e-03	2.048e-03

Table 13.9: The difference in final model $\left|\mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_1} - \mathbf{x}^{\text{ADP,cart}}_{\text{num},\delta_2}\right|$ in Alanine in Å^2.

refinement obtained via different δ, labelled $\mathbf{x}_{\text{num},\delta}$. The smallest distanced are highlighted in green, the worst in red, and those in the middle in yellow. It is clear that $\delta = 10^{-2}$Å and $\delta = 10^{-3}$Å agree most closely across all molecules for both the positional and ADP components. $\delta = 10^{-4}$Å shows some close agreement with these. For Ammonia (13.6 and13.7), the refinement with $\delta = 10^{-6}$Å achieved a low wR2-factor, and this is backed up by it having a strong agreement with $\delta = 10^{-2}$Å and $\delta = 10^{-3}$Å, however typically the inconsistency of $\delta = 10^{-6}$Å makes it unreliable and not a good choice for refinement. Additionally, all choices of δ resulted in models which agreed more strongly with each other than with approximate refinement, that is, all such δ choices provide an improvement to the model.

In view of the above results, we decided to fix the Cartesian step size to be $\delta = 10^{-3}$Å for all numerical differentiation in the remainder of this paper.

13.3.3 Consistency

Finally, to show consistency, we compare the journeys of numerical nonspherical refinement beginning at the two different start models $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$, as shown in Table 13.10.

	shift/esd	shift/esd			maximal atomic
step	($\times 10^5$)	($\times 10^5$)	wR$_2$	wR$_2$	distance
	(from $\mathbf{x}_{\text{spher}}$)	(from $\mathbf{x}_{\text{appr}}$)	(from $\mathbf{x}_{\text{spher}}$)	(from $\mathbf{x}_{\text{appr}}$)	(in 10^{-6}Å)
Ammonia					
k=0	392693	143955	9.2821%	1.9440%	126974(H)
k=1	198189	18797	2.6092%	1.9180%	20665(H)
k=2	47387	2818	1.9216%	1.9176%	2458(H)
k=3	7724	760	1.9176%	1.9176%	380(H)
k=4	1823	324	1.9175%	1.9176%	102(H)
k=5	592	142	1.9176%	1.9176%	36(H)
k=6	140	78	1.9176%	1.9176%	11(H)
k=7...20	≤88	≤119	1.9176%	1.9176%	≤5(H)
Epoxide					
k=0	674626	51117	12.0375%	4.6551%	130527(H3a)
k=1	281762	4568	4.9113%	4.6536%	21095(H2b)
k=2	35810	812	4.6561%	4.6536%	3093(H3a)
k=3	6868	476	4.6537%	4.6536%	461(H3b)
k=4	1365	152	4.6536%	4.6536%	136(H3a)
k=5	292	88	4.6536%	4.6536%	38(H3a)
k=6	138	102	4.6536%	4.6536%	16(H3b)
k=7...20	≤99	≤126	4.6536%	4.6536%	≤11(H3b)

step	shift/esd	shift/esd			maximal atomic
	($\times 10^5$)	($\times 10^5$)	wR$_2$	wR$_2$	distance
	(from $\mathbf{x}_{\text{spher}}$)	(from $\mathbf{x}_{\text{appr}}$)	(from $\mathbf{x}_{\text{spher}}$)	(from $\mathbf{x}_{\text{appr}}$)	(in 10^{-6}Å)
L-Alanine					
k=0	992154	50245	6.4989%	3.2299%	126479(H2b)
k=1	308603	3568	3.3127%	3.2291%	24140(H1a)
k=2	32231	641	3.2298%	3.2291%	1818(H1a)
k=3	7711	244	3.2291%	3.2291%	237(H1c)
k=4	1838	176	3.2291%	3.2291%	59(H1b)
k=5	431	202	3.2291%	3.2291%	20(H1b)
k=6	214	194	3.2291%	3.2291%	16(H1b)
k=7...20	≤190	≤267	3.2291%	3.2291%	≤14(H1a)

Table 13.10: Progression of refinement values from $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ via numerical nonspherical refinement with $\delta = 10^{-3}$Å. In the first two columns, if the shift/esd is below 0.01, it is colored green, else red. The final column represents the maximal distance between sites of corresponding atoms (with the specific atom giving this maximal value in brackets) through numerical nonspherical refinement from these two start points.

The first four columns of Table 13.10 demonstrate the typical convergence measures shift/esd and wR_2-factor for numerical nonspherical refinement processes starting from $\mathbf{x}_{\text{spher}}$ and from $\mathbf{x}_{\text{appr}}$.

Green values in the first two columns mean that the convergence criterion "shift/esd < 0.01" is achieved while red valued mean that it is not.

As expected, the initial wR_2-factor of numerical nonspherical refinement starting from $\mathbf{x}_{\text{spher}}$ is much higher than the initial wR_2-factor of numerical nonspherical refinement starting from $\mathbf{x}_{\text{appr}}$, but the wR_2-factors of both journeys agree already after a few steps in both refinement processes.

Let us, finally, focus on the last column of Table 13.10. Let $\mathbf{x}_{k,s}$ and $\mathbf{x}_k$ be the models obtained via k numerical nonspherical refinement steps starting from $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$, respectively. This column records the maximal distance (in Å) between sites of corresponding atoms in the same refinement step of each journey. For example, the third entry 2458×10^{-6} Å is the distance between the atom site of H of ammonia after three numerical nonspherical refinement steps starting from $\mathbf{x}_{\text{spher}}$ and the same atom site of H after three numerical nonspherical refinement steps starting from $\mathbf{x}_{\text{appr}}$ (since this distance is maximal for H amongst all corresponding atoms). Table 13.11 shows that this maximum is always assumed by one of the hydrogen atoms which is also expected. These maximal distances between sites of corresponding atoms shrink significantly after each refinement step until they reach values around 10^{-6} Å. Since, after about 7 refinement steps, these distances are about the same size as the maximal Cartesian shifts of the individual atoms from the model $\mathbf{x}_k$ to the model $\mathbf{x}_{k+1}$, we can safely conclude that convergence of both refinement processes is essentially obtained after at most 7 refinement steps. Moreover, since, the two refinement processes from different start models $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ lead after about 7 refinement steps to models whose atom sites agree up to 10^{-6} Å, the table proves in particular **consistency** of numerical nonspherical refinement.

Figures 13.16 to 13.19 show the path of numerical refinement from both spherical

		maximal Cartesian	maximal Cartesian
	maximal atomic	atomic shift	atomic shift
step	distance	(in 10^{-6}Å)	(in 10^{-6}Å)
	(in 10^{-6}Å)	(from $\mathbf{x}_{\text{spher}}$)	(from $\mathbf{x}_{\text{appr}}$)
Ammonia			
k=0	126974(H)	-	-
k=1	20665(H)	115659(H)	9355(H)
k=2	2458(H)	21309(H)	981(H)
k=3	380(H)	2457(H)	168(H)
k=4	102(H)	420(H)	44(H)
k=5	36(H)	127(H)	15(H)
k=6	11(H)	35(H)	7(H)
k=7...20	$\leq$5(H)	$\leq$8(H)	$\leq$6(H)
Epoxide			
k=0	130527(H3a)	-	-
k=1	21095(H2b)	124253(H3a)	4694(H3a)
k=2	3093(H3a)	21697(H3b)	321(H3a)
k=3	461(H3b)	2653(H3a)	80(H3a)
k=4	136(H3a)	458(H3b)	45(H3a)
k=5	38(H3a)	93(H3a)	12(H2b)
k=6	16(H3b)	27(H3a)	9(H3b)
k=7...20	$\leq$11(H3b)	$\leq$12(H3a)	$\leq$13(H3b)
L-Alanine			
k=0	126479(H2b)	-	-
k=1	24140(H1a)	110776(H2b)	4802(H1c)
k=2	1818(H1a)	22253(H1a)	172(H1a)
k=3	237(H1c)	1655(H1a)	18(H1b)
k=4	59(H1b)	194(H1c)	19(H1b)
k=5	20(H1b)	50(H1b)	13(H1c)
k=6	16(H1b)	16(H1b)	14(H1c)
k=7...20	$\leq$14(H1a)	$\leq$19(H1b)	$\leq$14(H1a)

Table 13.11: Progression of refinement values from $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ via numerical nonspherical refinement with $\delta = 10^{-3}$Å. The first column represents the maximal distance (together with the atom achieving this distance in brackets) between sites of corresponding atoms through numerical nonspherical refinement from these two start points, whilst the last two columns represent the maximal Cartesian atomic shift within the given refinement path, with the atom achieving this shift named in brackets.

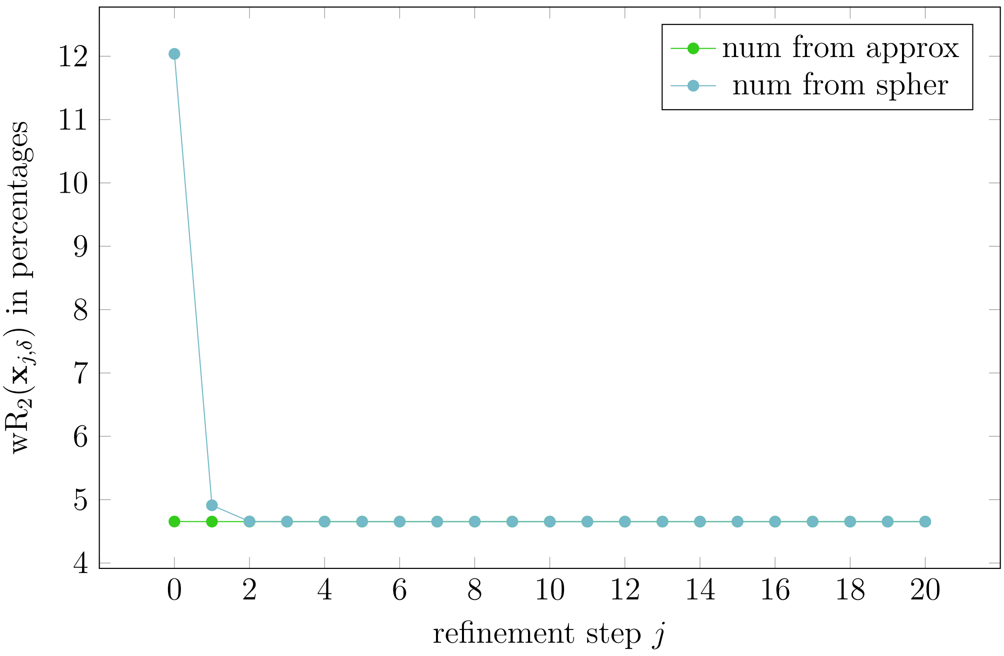


Figure 13.16: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ beginning at $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ (epoxide)

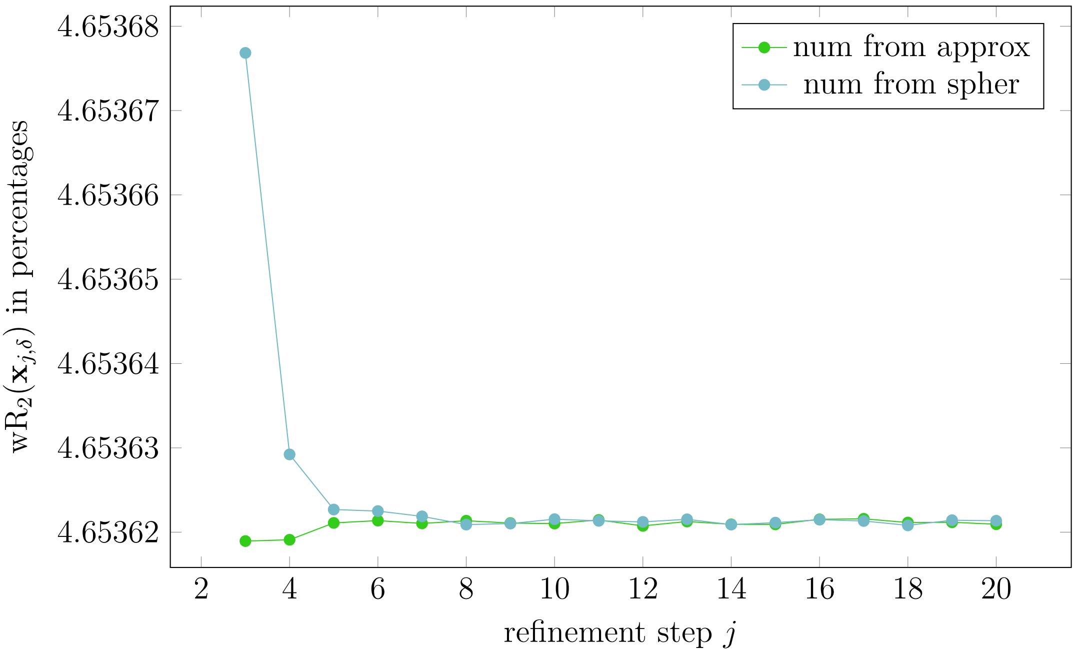


Figure 13.17: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 3, 4, \ldots, 20$ beginning at $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ (epoxide)

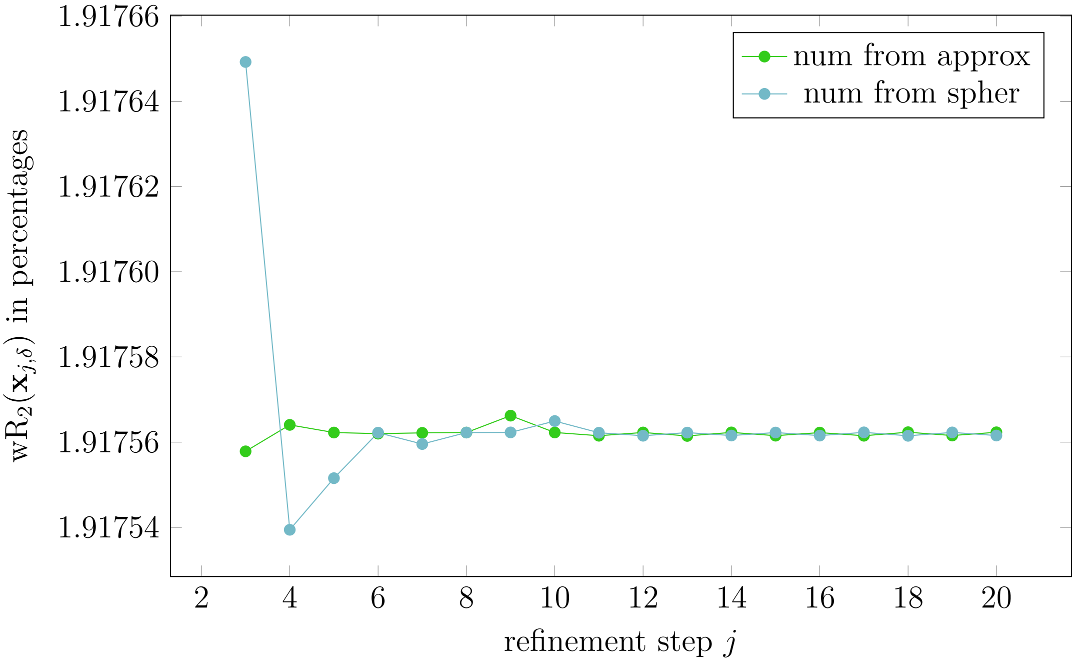


Figure 13.18: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 3, 4, \ldots, 20$ beginning at $\mathbf{x}_{\mathrm{spher}}$ and $\mathbf{x}_{\mathrm{appr}}$ (ammonia)

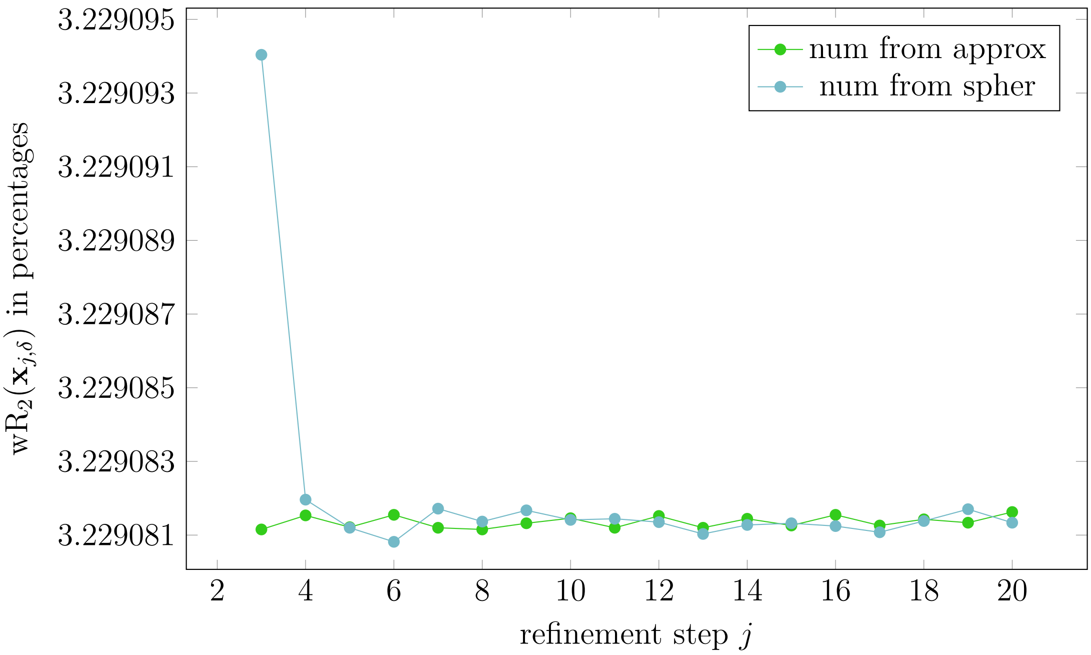


Figure 13.19: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 3, 4, \ldots, 20$ beginning at $\mathbf{x}_{\mathrm{spher}}$ and $\mathbf{x}_{\mathrm{appr}}$ (alanine)

and approximate minima, and clearly shows the two paths converging to the same wR_2.

13.3.4 Time Cost

Time cost comparisons of approximate and numerical non-spherical refinements are presented in Table 13.12. We have also added in hybrid nonspherical refinement which will be defined in Subsection 13.4.1 for completeness. We present the median time cost for one refinement step (that is, the generation and application of one shift vector) amongst 20 random refinement steps in each of the three refinement processes. These were carried out on a computer running Windows 10 with 15GB of RAM and 3 CPU cores.

Molecule	Approximate	Hybrid	Numerical
Ammonia	9.3	63	79
Epoxide	11.8	302	503
L-Alanine	47.2	2137	3783

Table 13.12: Median time costs for one step of each non-spherical refinement process (in seconds)

As numerical non-spherical refinement requires a quantum mechanical calculation twice for each positional parameter in $\mathbf{x}$, as well as one for the central F_{calc}, we expect it to take $(2 \times \#_{\text{parameters}} + 1)$ times as long to run as approximate non-spherical refinement does, which is reflected in this Table. In the case of hybrid non-spherical refinement, we count only those positional parameters relating to hydrogen atoms. For example, in the case of L-alanine, one expects that numerical will take 79 times as long, and hybrid 43 times as long, as approximate non-spherical refinement. This is in good agreement with Table 13.12, where the ratios are $3783/47.2 \approx 80$ and $2137/47.2 \approx 45$ respectively. This makes clear that the major contribution to the time cost is the quantum mechanical calculations.

13.4 Further Refinement Methods

This section focuses on two additional refinement methods, whose methodology lie between numerical and approximate refinement (hybrid refinement) and approximate and spherical refinement (mixed refinement). Hybrid refinement uses numerical information for the hydrogen-related information, and approximate information for the non-hydrogen parts, and provides a result closer to numerical refinement than approximate refinement is with a lower time cost. Mixed refinement uses spherical information for the derivatives and only uses non-spherical information for the structure factors, and yet provides a result much closer to approximate refinement than spherical refinement. These observations are purely academic, as there is no time improvement gained with this method. This section covers their definitions and echoes of the testing completed in Section 13.3 to verify the plausibility of these methods.

13.4.1 Hybrid Refinement

We will see later in Section 13.5 that approximate refinement provides very accurate models for heavier atoms in comparison to numerical refinement. However, the data on hydrogen atoms are less precise. This observation motivates a nonspherical refinement process which focusses specifically on the improvement of the hydrogen data. The idea is to use numerical partial derivatives of the structure factor for directions corresponding to hydrogen atoms and to use approximate partial derivatives for all other directions. We refer to this compromise between approximate and numerical refinement as *hybrid (nonspherical) refinement.* The entries of the design matrix $\tilde{D}_{\text{hybr}}(\mathbf{x})$ of hybrid refinement are therefore given by

$$(\tilde{D}_{\text{hybr}}(\mathbf{x}))_{kn} = \begin{cases} \left(\frac{\partial \tilde{K} Y}{\partial x_n}\right)_{\text{num}}(\mathbf{x}, \mathbf{h}_k) & \text{if } x_n \text{ is a hydrogen coordinate,} \\ \\ \left(\frac{\partial \tilde{K} Y}{\partial x_n}\right)_{\text{appr}}(\mathbf{x}, \mathbf{h}_k) & \text{otherwise.} \end{cases} \tag{13.4.1}$$

The computation of the partial derivatives on the right hand side of (13.4.1) is based on (12.2.4), (13.1.3) and (13.1.4). The time effort to compute the partial derivatives in a hybrid refinement step is a fraction of the time effort for a numerical refinement step. It agrees roughly with the fraction of hydrogen atoms in the molecule.

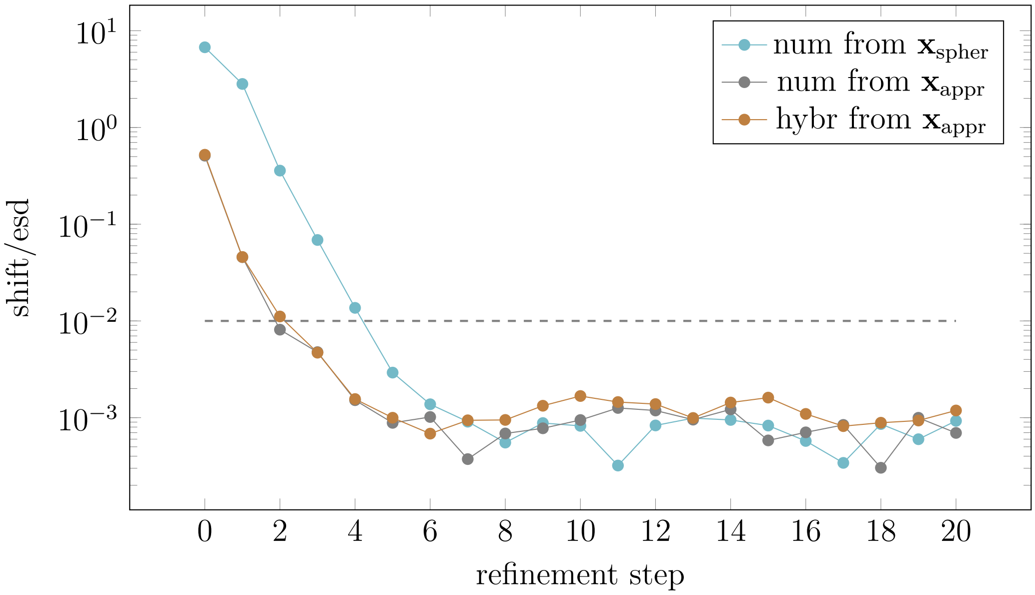


Figure 13.20: Maximal shift/esd for numerical non-spherical refinement starting from $\mathbf{x}_{\text{appr}}$ for hybrid refinement and from $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{spher}}$ for numerical refinement (epoxide)

Figures 13.20 to 13.22 compare the convergence of hybrid refinement starting from $\mathbf{x}_{\text{appr}}$ with that of numerical refinement starting from $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ in terms of the shift/esd. Since the vertical scale is logarithmic and all graphs start in the first refinement steps with the same slope, we have exponential convergence with the same convergence rate (for both hybrid and numerical refinement) which then transitions into a horizontal zigzag behaviour, where the convergence is essentially completed and the models undergo only minor modifications with no apparent further improvement. (We have seen this already earlier in Figures 13.7 to 13.14 in the case of numerical refinement since there is no significant further improvement of the wR_2factor from the 6-th refinement step onwards).

Similarly, we can view the wR_2-factor progression from the approximate minimum

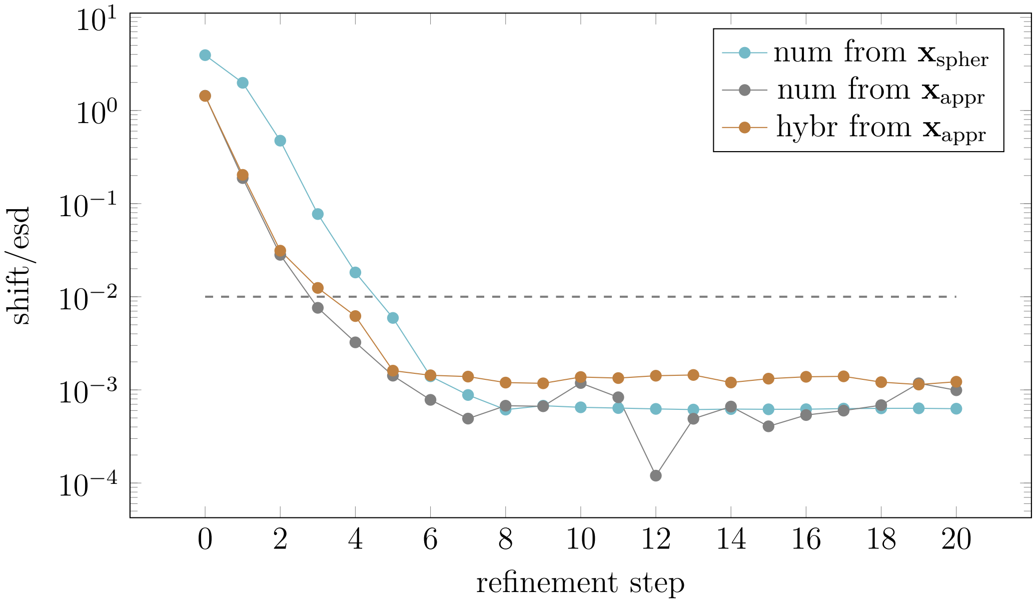


Figure 13.21: Maximal shift/esd for numerical non-spherical refinement starting from $\mathbf{x}_{\text{appr}}$ for hybrid refinement and from $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{spher}}$ for numerical refinement (ammonia)

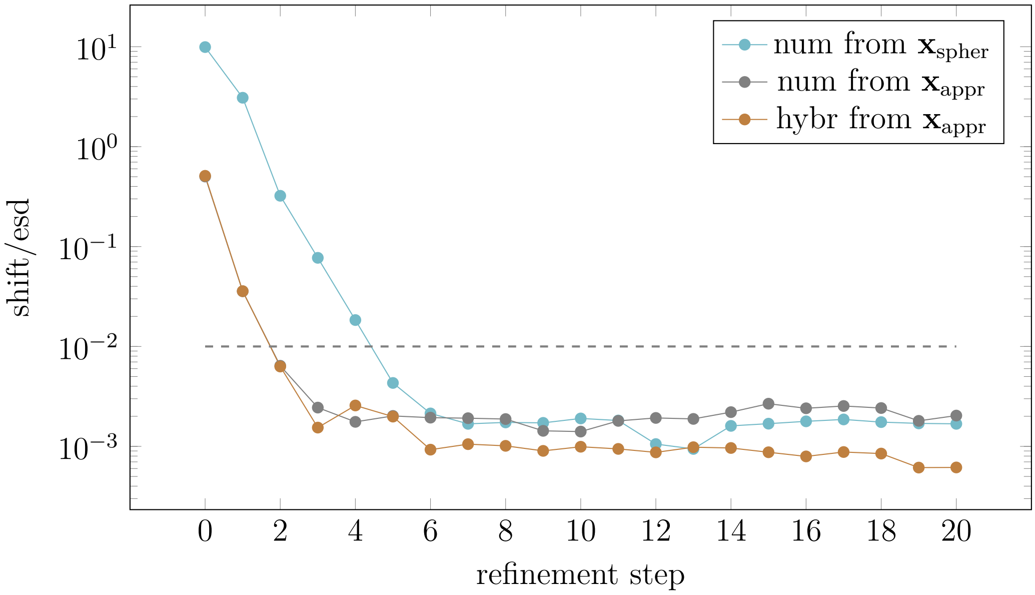


Figure 13.22: Maximal shift/esd for numerical non-spherical refinement starting from $\mathbf{x}_{\text{appr}}$ for hybrid refinement and from $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{spher}}$ for numerical refinement (L-alanine)

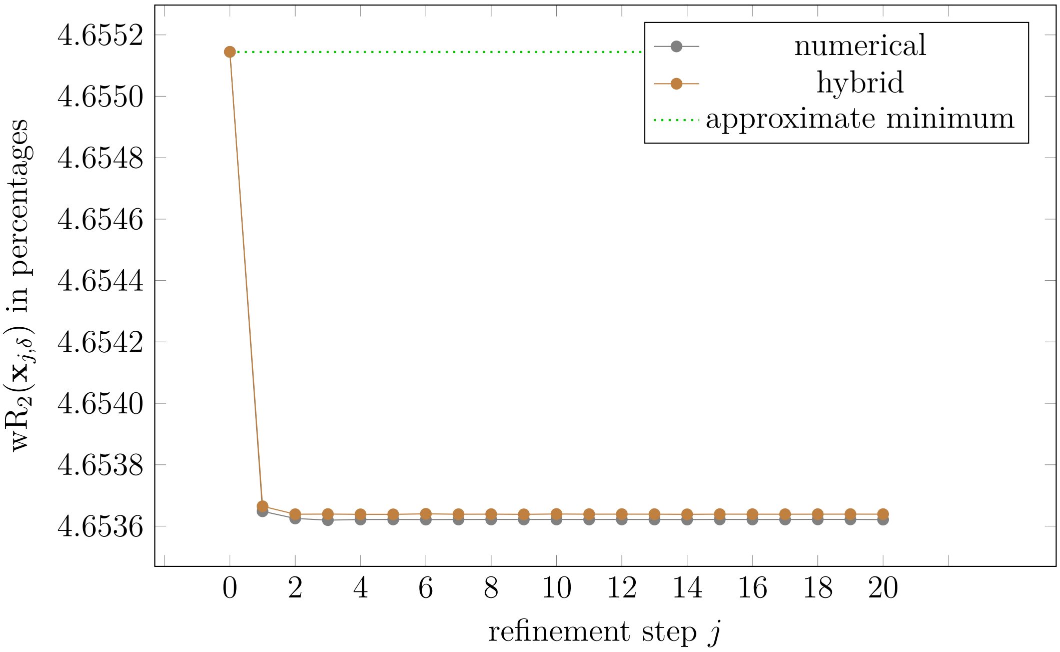


Figure 13.23: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for numerical and hybrid refinement (epoxide)

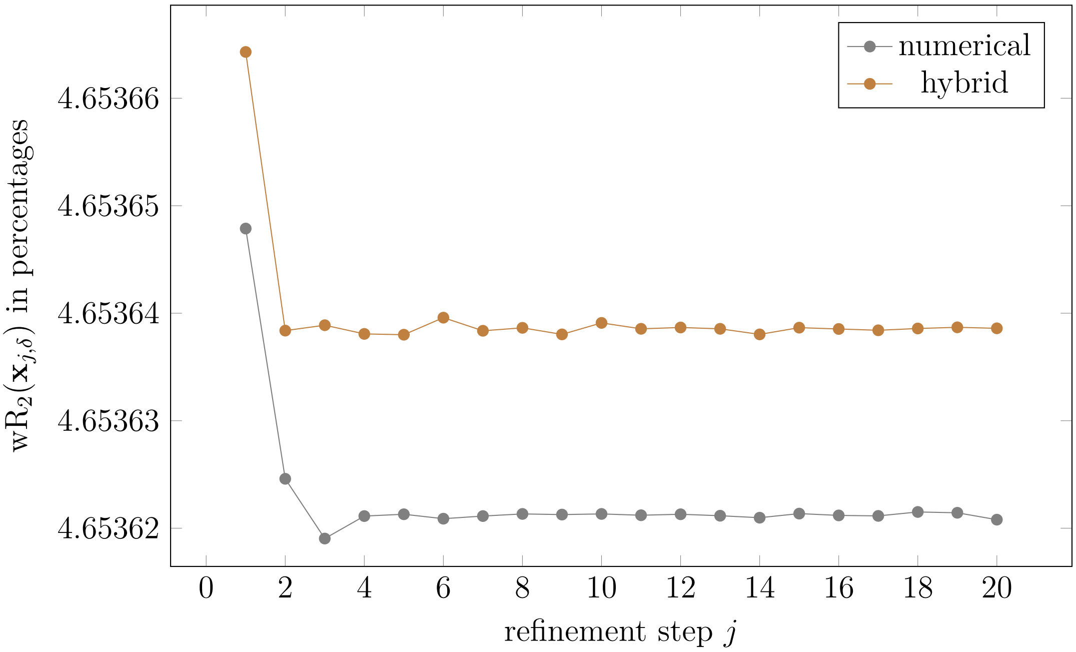


Figure 13.24: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for numerical and hybrid refinement, skipping the first step (epoxide)

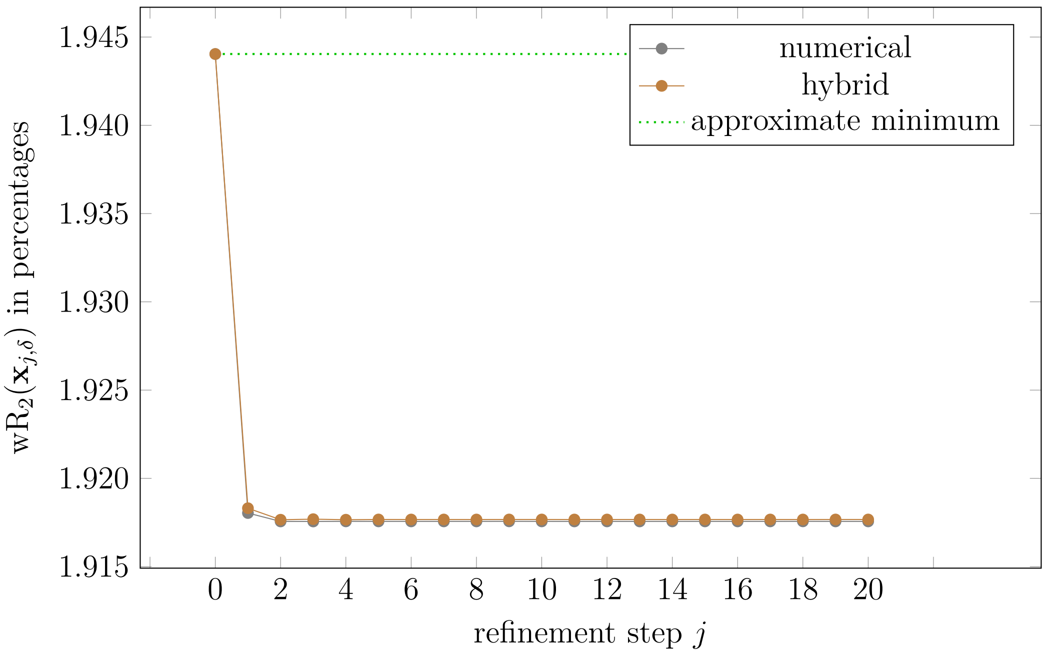


Figure 13.25: The progression of $wR_2(\mathbf{x}_j)$ for numerical and hybrid refinement (ammonia)

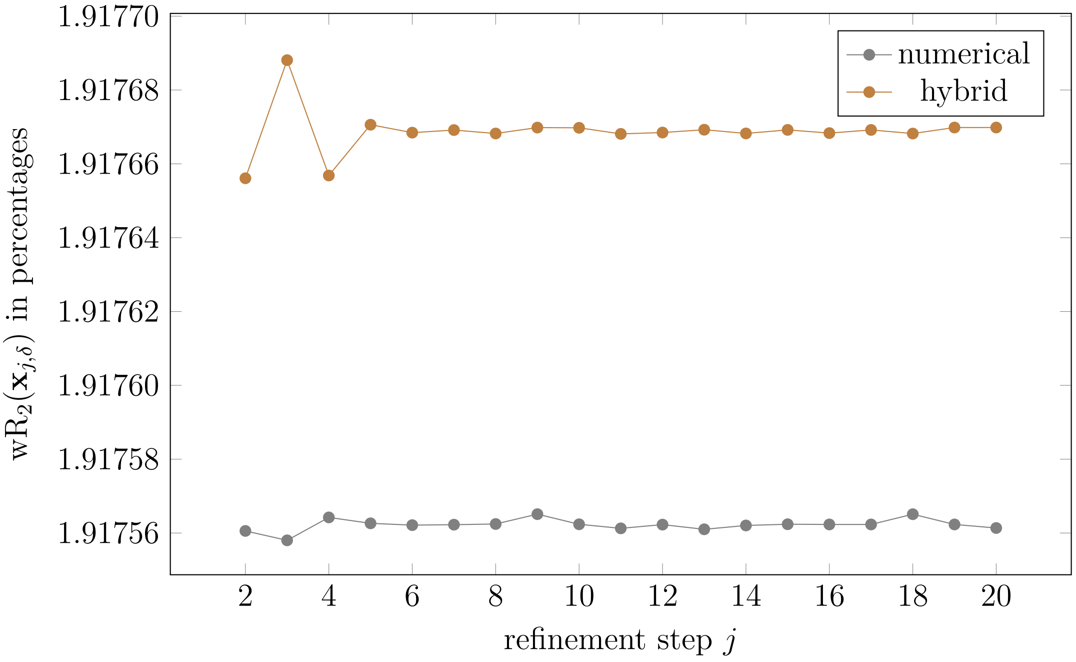


Figure 13.26: The progression of $wR_2(\mathbf{x}_j)$ for numerical and hybrid refinement, skipping the first two steps (ammonia)

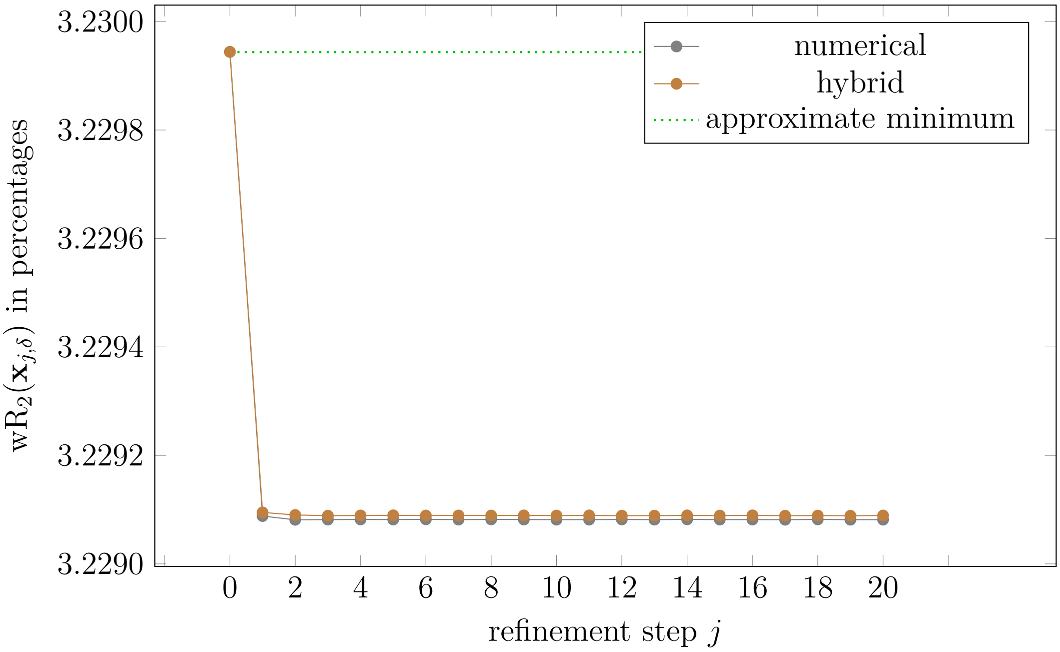


Figure 13.27: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for numerical and hybrid refinement (L-Alanine)

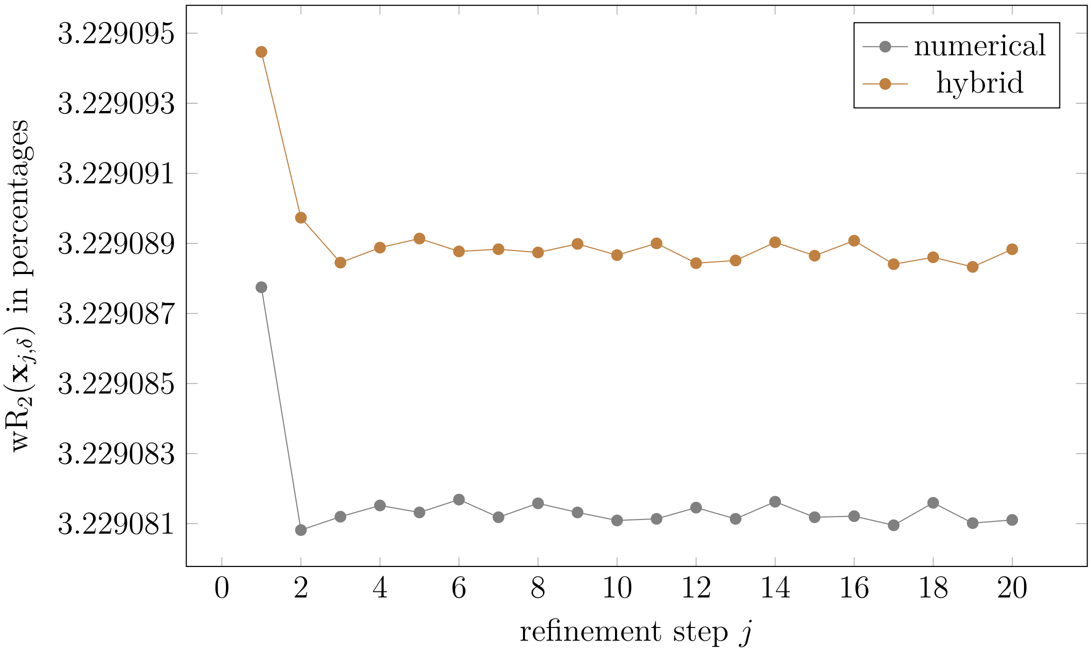


Figure 13.28: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for numerical and hybrid refinement, skipping the first step (L-Alanine)

through both hybrid and numerical refinements in Figures 13.23 to 13.28, which clearly indicate that hybrid refinement provides a huge step towards the results attained through numerical nonspherical refinement.

We will analyse in Section 13.5 whether hybrid refinement provides a more accurate model for all atoms (including the hydrogen atoms) than approximate refinement (in comparison to numerical refinement).

Henceforth we will refer to $\mathbf{x}_{\text{hybr}}$ as the best model (with respect to wR_2) obtained within 20 hybrid refinement steps starting from $\mathbf{x}_{\text{appr}}$. This is in analogy to the earlier definition of $\mathbf{x}_{\text{num}}$.

13.4.2 Mixed Refinement Using $\mathbf{s}_{\text{mix}} = S_{\text{spher}} \cdot \tilde{\mathbf{r}}_{\text{nonspher}}$

Investigations Based on $\mathbf{s} = S \cdot \tilde{\mathbf{r}}$

As discussed in Part I (Subsection 6.6), the shift vector at $\mathbf{x} \in \mathbb{R}^N$ for the refinement step is obtained via the product $\mathbf{s}(\mathbf{x}) = S(\mathbf{x}) \cdot \tilde{\mathbf{r}}(\mathbf{x})$. There are two possibilities for the residuals, namely $\tilde{\mathbf{r}}_{\text{spher}}$ and $\tilde{\mathbf{r}}_{\text{nonspher}}$, and four possibilities for the shift matrices, namely S_{spher}, S_{appr}, S_{hybr} and S_{num} (derived from their corresponding design matrices $\tilde{D}_{\text{spher}}$, $\tilde{D}_{\text{appr}}$, $\tilde{D}_{\text{hybr}}$ and $\tilde{D}_{\text{num}}$ via $S_\bullet(\mathbf{x}) = (\tilde{D}_\bullet(\mathbf{x})^\top W \tilde{D}_\bullet(\mathbf{x}))^{-1} \tilde{D}_\bullet(\mathbf{x})^\top W$ (see (6.6.4) in Section 6.6) with $\bullet \in \{\text{spher}, \text{appr}, \text{hybr}, \text{num}\}$). Recall that the shift vectors of our four refinements under consideration are then given by

$$\begin{aligned}
&\text{spherical refinement:} &\quad \mathbf{s}_{\text{spher}} &= S_{\text{spher}} \cdot \tilde{\mathbf{r}}_{\text{spher}},\\
&\text{approximate refinement:} &\quad \mathbf{s}_{\text{appr}} &= S_{\text{appr}} \cdot \tilde{\mathbf{r}}_{\text{nonspher}},\\
&\text{hybrid refinement:} &\quad \mathbf{s}_{\text{hybr}} &= S_{\text{hybr}} \cdot \tilde{\mathbf{r}}_{\text{nonspher}},\\
&\text{numerical refinement:} &\quad \mathbf{s}_{\text{num}} &= S_{\text{num}} \cdot \tilde{\mathbf{r}}_{\text{nonspher}}.
\end{aligned}$$

However, there are four other possibilities to combine our shift matrices and residuals. In this section, we investigate the contribution of the shift matrices and the residuals to the resulting shift vector and will see that one of these remaining combinations, namely $S_{\mathrm{spher}} \cdot \tilde{\mathbf{r}}_{\mathrm{nonspher}}$, does also provide surprisingly good refinement results. We will refer to this refinement as *mixed (spherical-nonspherical) refinement*, and its corresponding optimum as $\mathbf{x}_{\mathrm{mix}}$.

Shift Vector Analysis at $\mathbf{x}_{\text{spher}}$

Table 13.13 confirms our expectation that the numerical and approximate shift vectors at the spherical optimum $\mathbf{x}_{\text{spher}}$ are roughly in agreement.

shift vector components at $\mathbf{x}_{\text{spher}}$	epoxide	L-alanine	ammonia
$\|\|\mathbf{s}_{\text{num}}^{\text{pos,cart}}\|\|$	2.089e-01	1.157e-01	2.386e-01
$\left\|\left\|\mathbf{s}_{\text{appr}}^{\text{pos,cart}}\right\|\right\|$	2.388e-01	1.250e-01	2.795e-01
$\sphericalangle\left(\mathbf{s}_{\text{num}}^{\text{pos,cart}}, \mathbf{s}_{\text{appr}}^{\text{pos,cart}}\right)$	3.09°	2.36°	5.13°
$\left\|\left\|\mathbf{s}_{\text{num}}^{\text{ADP,cart}}\right\|\right\|$	6.434e-02	3.435e-02	1.145e-01
$\left\|\left\|\mathbf{s}_{\text{appr}}^{\text{ADP,cart}}\right\|\right\|$	9.245e-02	4.560e-02	1.359e-01
$\sphericalangle\left(\mathbf{s}_{\text{num}}^{\text{ADP,cart}}, \mathbf{s}_{\text{appr}}^{\text{ADP,cart}}\right)$	11.30°	8.48°	12.77°

Table 13.13: Comparison of numerical and approximate Cartesian shift vector components at $\mathbf{x}_{\text{spher}}$

Recall that shift vectors are a product of the shift matrix and the residua. Therefore it makes sense to investigate the contributions of the factors S and $\tilde{\mathbf{r}}$ to the length and direction of the shift vectors at $\mathbf{x}_{\text{spher}}$ in Tables 13.14 and 13.15 by considering all 8 possible combinations. The norms $||\cdot||$ are $||\mathbf{s}|| = \sqrt{\mathbf{s}\cdot\mathbf{s}}$, the symbol $\sphericalangle$ in these tables stands for the angle of the vector under consideration to $\mathbf{s}_{\text{num}}^{\blacklozenge,\text{cart}}$ with $\blacklozenge \in \{\text{pos}, \text{ADP}\}$. We provide no angle information in the last rows of the tables, since $\mathbf{s}_{\text{spher}}^{\blacklozenge,\text{cart}}$ is essentially zero at the spherical optimum and the zero vector does not have a well defined angle with the corresponding numerical shift vector component.

The combinations in the Tables 13.14 and 13.15 highlighted in blue are products $S_\bullet \cdot \tilde{\mathbf{r}}_*$ mixing spherical and nonspherical factors. It seems questionable to consider such combinations as meaningful “shift vectors”.

However, all 8 combinations in these tables provide useful information about which of the two factors has more influence into the length and direction of the resulting “shift vector”. The tables reveal that the products undergo a significant change when the nonspherical residual is replaced by the spherical residual and that all products

$(S_\bullet \cdot \tilde{\mathbf{r}}_*)^{\text{pos,cart}}$	epoxide		L-alanine		ammonia	
	$\|\cdot\|$	$\sphericalangle$	$\|\cdot\|$	$\sphericalangle$	$\|\cdot\|$	$\sphericalangle$
$(\bullet, *) = (\text{num}, \text{nonspher})$	2.09e-01	0°	1.16e-01	0°	2.39e-01	0°
$(\bullet, *) = (\text{hybr}, \text{nonspher})$	2.09e-01	0.2°	1.16e-01	0.2°	2.38e-01	0.1°
$(\bullet, *) = (\text{appr}, \text{nonspher})$	2.39e-01	3.1°	1.25e-01	2.4°	2.79e-01	5.1°
$(\bullet, *) = (\text{spher}, \text{nonspher})$	2.51e-01	12.9°	1.33e-01	5.8°	2.98e-01	7.9°
$(\bullet, *) = (\text{num}, \text{spher})$	1.15e-02	83.6°	3.24e-03	81.7°	2.19e-02	91.2°
$(\bullet, *) = (\text{hybr}, \text{spher})$	1.15e-02	80.8°	3.13e-03	77.5°	2.19e-02	91.4°
$(\bullet, *) = (\text{appr}, \text{spher})$	1.24e-02	90.0°	8.54e-03	151.1°	2.71e-02	117.1°
$(\bullet, *) = (\text{spher}, \text{spher})$	2.24e-11	-	2.18e-11	-	2.92e-05	-

Table 13.14: Contributions of the factors S and $\tilde{\mathbf{r}}$ to the positional shift vector component in comparison to the numerical shift vector component at $\mathbf{x}_{\text{spher}}$

$S_\bullet \cdot \tilde{\mathbf{r}}_{\text{nonspher}}$ (with $\bullet \in \{\text{num}, \text{hybr}, \text{appr}, \text{spher}\}$) represent roughly the same vectors. This observation motivates consideration of the product of the spherical shift matrix with the nonspherical residual as another potentially useful shift vector. We will see in the following sections that the resulting mixed refinement process converges for our 3 test molecules and leads to a significantly better optimal model than spherical refinement, measured with regards to the wR_2factor. In support of this deduction of the importance of the choice of $\tilde{\mathbf{r}}$, we can see that shift vectors using $\tilde{\mathbf{r}}_{\text{spher}}$ typically are at 90° (or worse) to $\mathbf{s}_{\text{num}}$, and would thus provide no progress towards the numerical minimum[2].

Investigation of Mixed Refinement

It is natural to be doubtful of a refinement that mixes spherical shift matrices with nonspherical residuals. An immediate question is whether a refinement based on such a combination will converge at all. In fact, Figures 13.29 to 13.31 indicate that this process converges exponentially to an optimum with a similar convergence

[2] the exception to this are the ADP shifts for ammonia, which with a 30° similarity may provide reasonable steps

$(S_\bullet \cdot \tilde{\mathbf{r}}_*)^{\text{ADP,cart}}$	epoxide		L-alanine		ammonia	
	$\Vert\cdot\Vert$	$\sphericalangle$	$\Vert\cdot\Vert$	$\sphericalangle$	$\Vert\cdot\Vert$	$\sphericalangle$
$(\bullet, *) = (\text{num}, \text{nonspher})$	6.43e-02	0°	3.43e-02	0°	1.15e-01	0°
$(\bullet, *) = (\text{hybr}, \text{nonspher})$	6.43e-02	0.6°	3.43e-02	0.3°	1.15e-01	0.2°
$(\bullet, *) = (\text{appr}, \text{nonspher})$	9.24e-02	11.3°	4.56e-02	8.5°	1.36e-01	12.8°
$(\bullet, *) = (\text{spher}, \text{nonspher})$	1.02e-01	22.1°	4.17e-02	20.6°	1.23e-01	30.8°
$(\bullet, *) = (\text{num}, \text{spher})$	1.08e-02	95.1°	4.92e-03	88.1°	5.18e-02	31.1°
$(\bullet, *) = (\text{hybr}, \text{spher})$	1.09e-02	94.2°	4.88e-03	87.5°	5.20e-02	31.2°
$(\bullet, *) = (\text{appr}, \text{spher})$	1.13e-02	95.4°	6.42e-03	104.1°	5.56e-02	32.3°
$(\bullet, *) = (\text{spher}, \text{spher})$	2.09e-13	-	1.42e-11	-	4.14e-06	-

Table 13.15: Contributions of the factors S and $\tilde{\mathbf{r}}$ to the ADP shift vector component in comparison to the numerical shift vector component at $\mathbf{x}_{\text{spher}}$

rate to numerical refinement. Note that the refinement journeys start from $\mathbf{x}_{\text{spher}}$. The graphs in these figures reveal also that this "mixed" process takes a few more refinement steps for the models to finally settle in an optimum. We call this process henceforth *mixed (spherical-nonspherical) refinement*, and we add it to the list of refinement processes that we compare in Section 13.5. Its shift vectors are computed via

$$\mathbf{s}(\mathbf{x}) = S_{\text{spher}}(\mathbf{x}) \cdot \tilde{\mathbf{r}}_{\text{nonspher}}(\mathbf{x}).$$

We do note that presently, mixed refinement requires just as much calculation as approximate nonspherical refinement, and provides no benefit over it, so we do not expect it to provide any practical use, but it remains theoretically interesting.

We can see in Figures 13.32 to 13.36 that mixed refinement provides a significant reduction in the wR_2 factor from the spherical minimum. Our investigations show that it remains worse than approximate refinement, but is far closer to approximate refinement than expected.

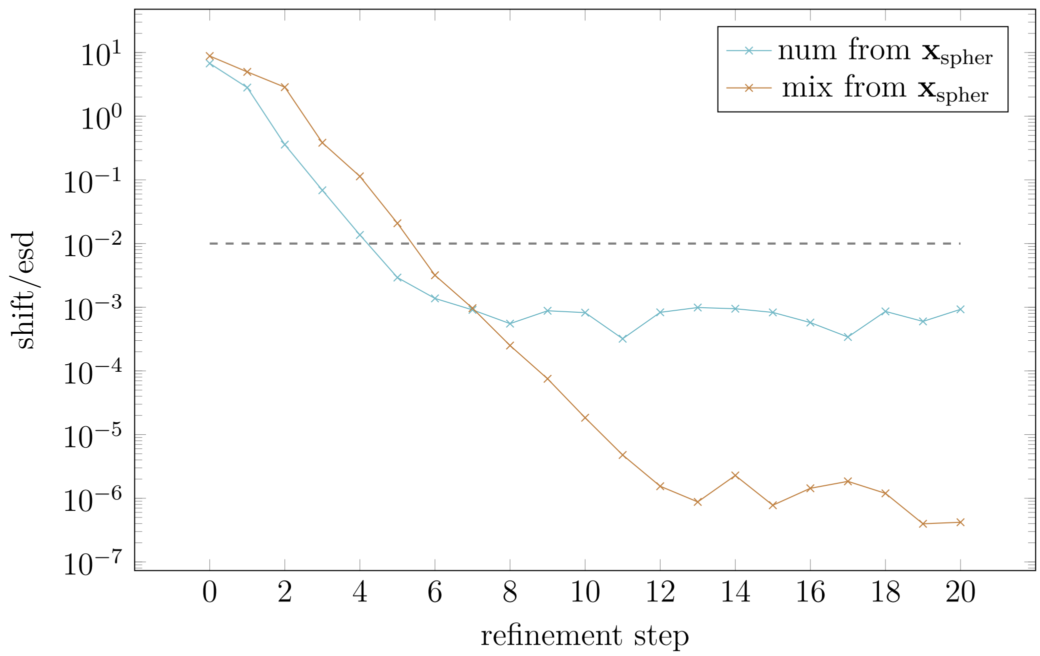


Figure 13.29: maximal shift/esd along numerical refinement journey starting from $\mathbf{x}_{\text{spher}}$ and along mixed refinement journey starting from $\mathbf{x}_{\text{spher}}$ (epoxide)

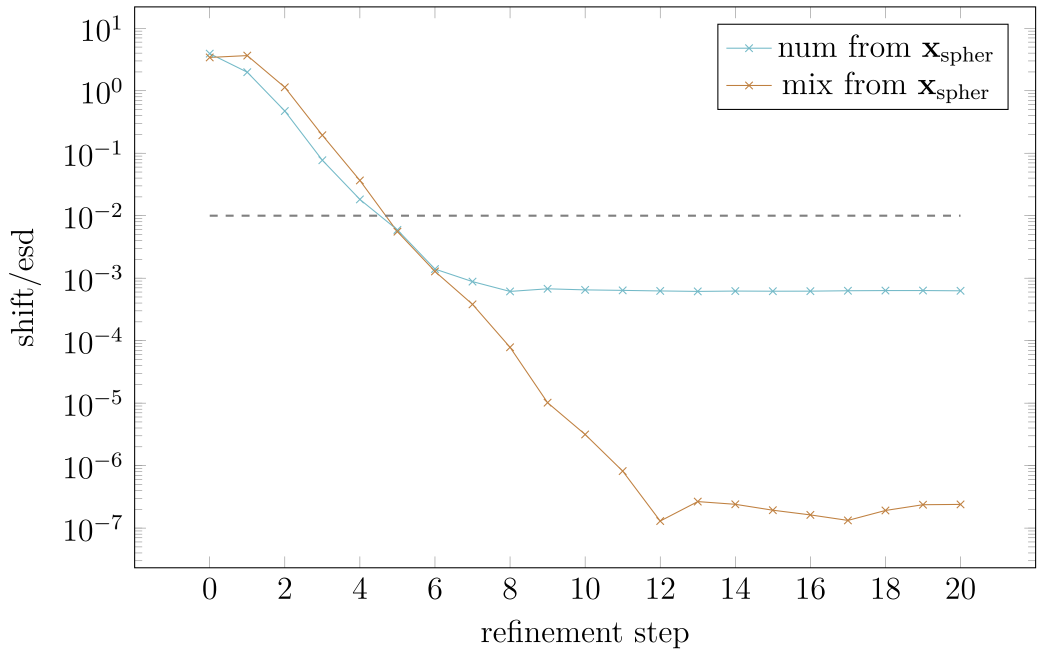


Figure 13.30: maximal shift/esd along numerical refinement journey starting from $\mathbf{x}_{\text{spher}}$ and along mixed refinement journey starting from $\mathbf{x}_{\text{spher}}$ (ammonia)

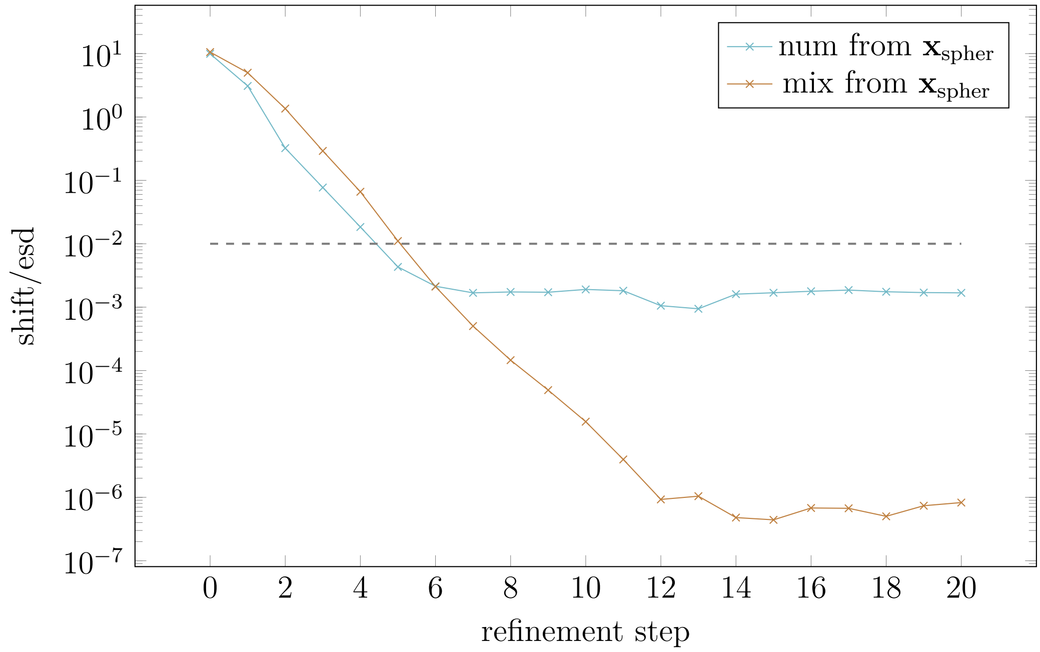


Figure 13.31: maximal shift/esd along numerical refinement journey starting from $\mathbf{x}_{\text{spher}}$ and along mixed refinement journey starting from $\mathbf{x}_{\text{spher}}$ (L-alanine)

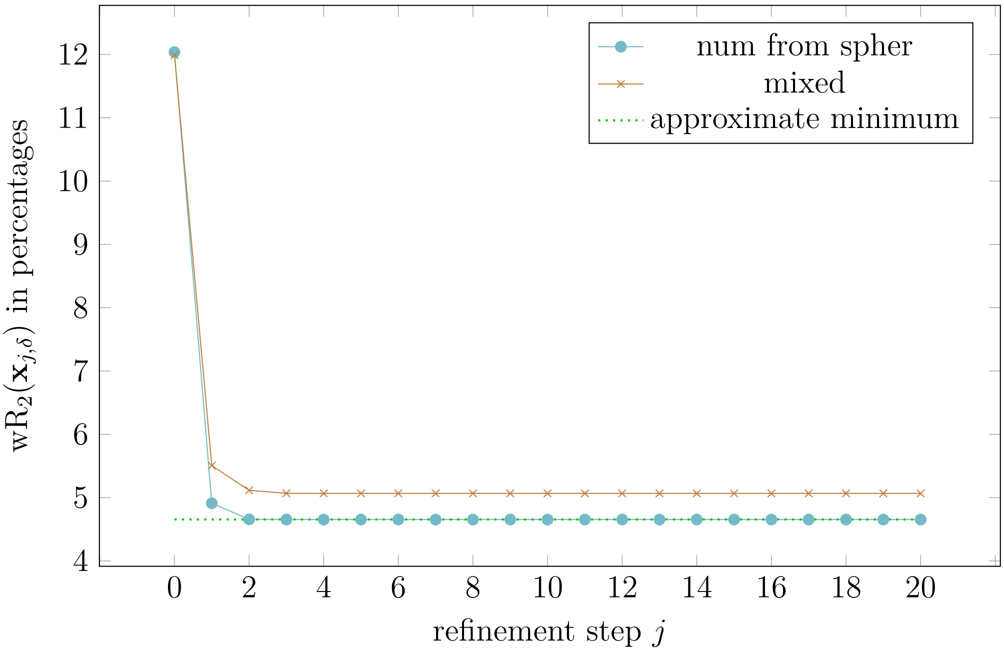


Figure 13.32: The progression of $\text{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ for mixed and numerical refinement (epoxide)

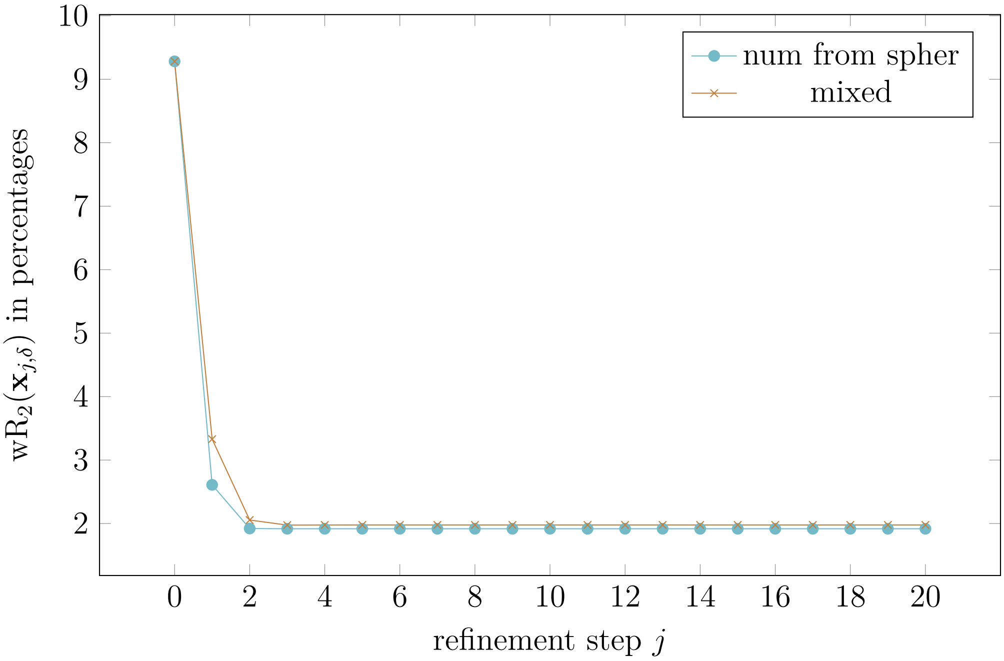


Figure 13.33: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \dots, 20$ for mixed and numerical refinement (ammonia)

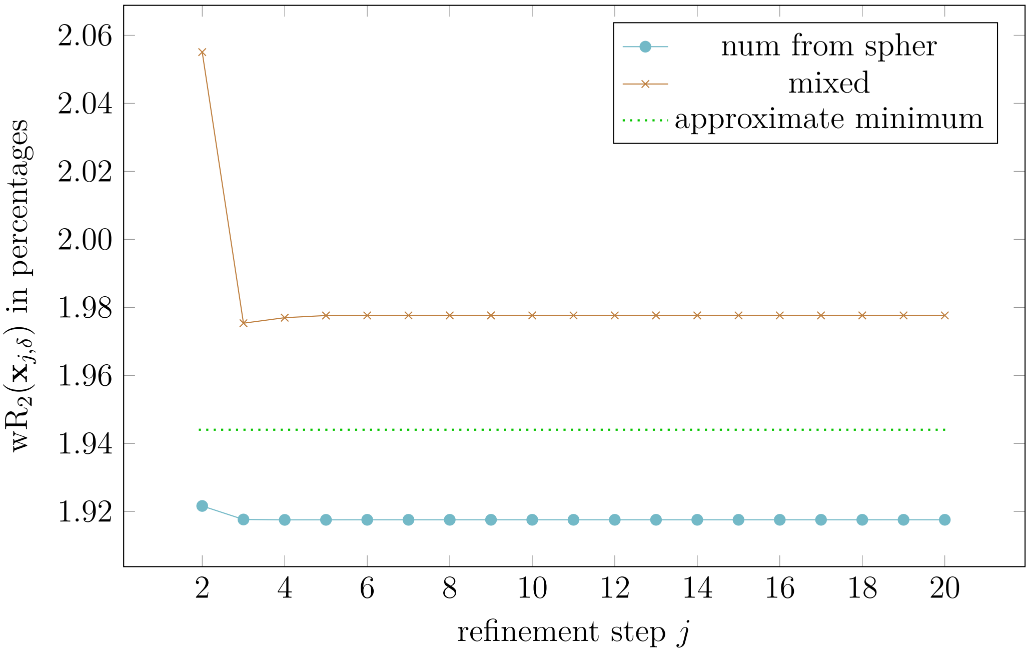


Figure 13.34: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 2, 3, \dots, 20$ for mixed and numerical refinement (ammonia)

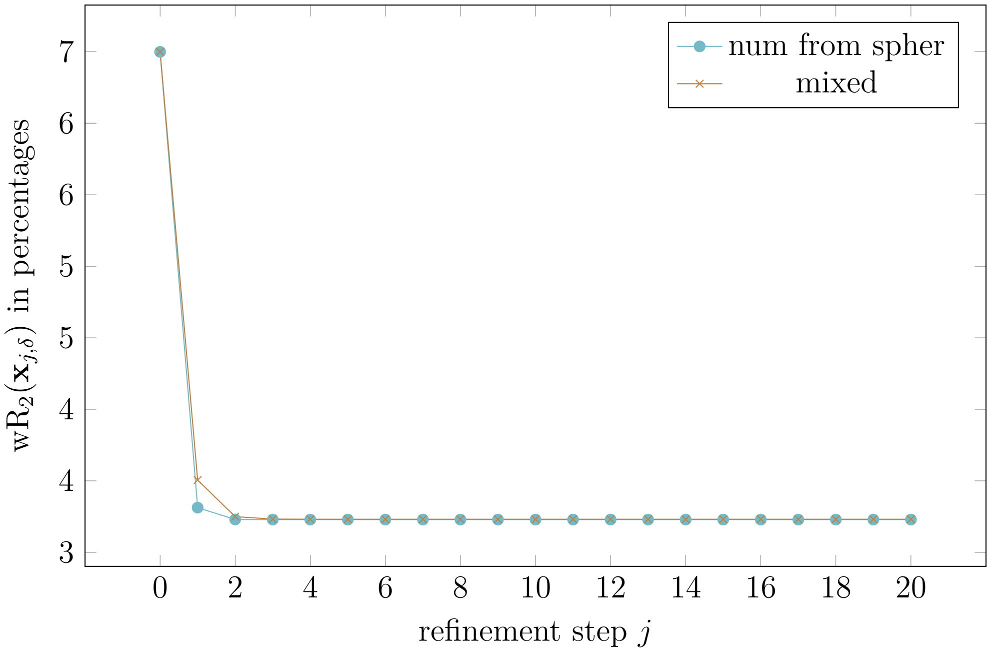


Figure 13.35: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 0, 1, \ldots, 20$ for mixed and numerical refinement (L-alanine)

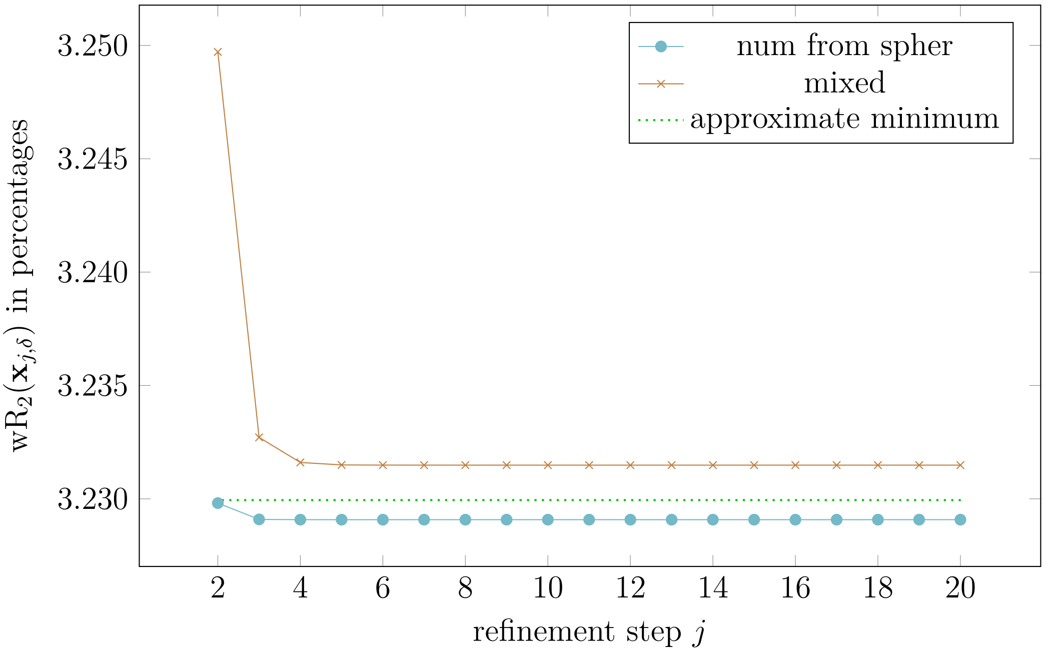


Figure 13.36: The progression of $\mathrm{wR}_2(\mathbf{x}_j)$ for $j = 2, 3, \ldots, 20$ for mixed and numerical refinement (L-alanine)

13.5 Comparison of Refinement Minima

This section is concerned with a direct comparison of the final models obtained via our different refinement methods. Within this, we work with the assumption that numerical nonspherical refinement is the best theoretical refinement method for our X-ray data, and measure the quality of the results obtained with spherical refinement, approximate nonspherical refinement and hybrid nonspherical refinement against that. As mentioned previously, all measurements are based on the Cartesian step size $\delta = 10^{-3}$Å. Our results are in agreement with general consensus that approximate nonspherical refinement provides a significant improvement to classical spherical refinement.

Whilst in the previous section we focused on the wR$_2$-factor, we will now be comparing more specific values such as the model parameters themselves and the hydrogen distances. The results of this section focus on the exact computational results. Considerations of statistical uncertainties will be investigated in the next section.

Approximate nonspherical and spherical refinement are obtained by Olex2, whilst hybrid and numerical nonspherical refinement are obtained by the process described in the flowchart in Figure 13.3 of Section 2.

13.5.1 wR2-Factors and X-H Distances

This section is devoted to the comparison of the optimal models obtained via the different refinement processes. Our main focus is on the precise numerical data, and the standard uncertainties will be considered in detail in Section 13.6. Recall that $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ are the optimal models obtained via classical spherical refinement in *olex2.refine*, and via nonspherical refinement using *olex2.refine* and NoSpherA2, respectively. Starting from $\mathbf{x}_{\text{appr}}$, we obtain $\mathbf{x}_{\text{num}}$ as the model with the lowest wR2-factor within 20 numerical refinement steps and, similarly, we obtain $\mathbf{x}_{\text{hybr}}$ as the model with the lowest wR2-factor within 20 hybrid refinement steps from $\mathbf{x}_{\text{appr}}$.

Finally, we also introduced mixed refinement in the previous section, and its optimal model $\mathbf{x}_{\text{mix}}$ is obtained as the model with the lowest wR2-factor within 20 mixed refinement steps from $\mathbf{x}_{\text{spher}}$.

In Table 13.16 we present the wR$_2$-factor and X-H distances of the models. We note that the wR$_2$-factor calculated with spherical form factors at $\mathbf{x}_{\text{spher}}$ is presented in the final column in parentheses.

	$\mathbf{x}_{\text{num}}$	$\mathbf{x}_{\text{hybr}}$	$\mathbf{x}_{\text{appr}}$	$\mathbf{x}_{\text{mix}}$	$\mathbf{x}_{\text{spher}}$
Ammonia					
R_1	0.836370%	0.835761%	0.823750%	0.838573%	4.06%(1.17%)
wR_2	1.917558%	1.917656%	1.944039%	1.977639%	9.28%(2.39%)
N − H	0.979(5)	0.979(5)	0.973(5)	0.970(5)	0.855(6)
Epoxide					
R_1	2.335272%	2.335347%	2.336363%	2.728423%	4.78%(3.21%)
wR_2	4.653619%	4.653638%	4.655144%	5.067212%	12.04%(6.71%)
C2 − H2a	1.095(6)	1.095(6)	1.095(7)	1.096(7)	0.995(8)
C2 − H2b	1.080(5)	1.080(5)	1.079(6)	1.076(6)	0.975(8)
C3 − H3a	1.106(7)	1.106(7)	1.102(8)	1.101(8)	0.975(11)
C3 − H3b	1.090(7)	1.090(7)	1.090(7)	1.090(8)	0.968(10)
L-Alanine					
R_1	1.906181%	1.906060%	1.906024%	1.908899%	3.03%(2.73%)
wR_2	3.229081%	3.229088%	3.229943%	3.231487%	6.50%(6.04%)
N1 − H1a	1.007(6)	1.007(6)	1.006(6)	1.007(6)	0.923(13)
N1 − H1b	1.019(6)	1.019(6)	1.019(6)	1.020(6)	0.936(12)
N1 − H1c	1.050(6)	1.050(6)	1.045(8)	1.047(7)	1.085(17)
C1 − H1	1.091(4)	1.091(4)	1.090(5)	1.091(5)	1.002(9)
C2 − H2a	1.101(6)	1.101(6)	1.100(7)	1.101(7)	0.993(13)
C2 − H2b	1.095(6)	1.095(6)	1.094(7)	1.096(7)	0.978(11)
C2 − H2c	1.090(5)	1.090(5)	1.089(6)	1.090(6)	1.005(10)

Table 13.16: wR$_2$-factors and X-H distances.

One can clearly see the dramatic difference between $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ with regards to a drop in the R-factors and increase of the X-H distances. The R factors of all four non-spherical optima $\mathbf{x}_{\text{mix}}, \mathbf{x}_{\text{appr}}, \mathbf{x}_{\text{hybr}}$ and $\mathbf{x}_{\text{num}}$ are essentially in agreement. The wR$_2$factor, which is a precise measure for the fit of the theoretical intensities (up to a suitable scaling factor $\tilde{K}$) with the observed intensities, shows a slight decrease from $\mathbf{x}_{\text{mix}}$ to $\mathbf{x}_{\text{appr}}$ to $\mathbf{x}_{\text{hybr}}$ to $\mathbf{x}_{\text{num}}$. Moreover, the wR$_2$factor of $\mathbf{x}_{\text{hybr}}$ tells us that hybrid refinement performs almost as well as numerical refinement.

The X-H distances of $\mathbf{x}_{\text{spher}}$ is typically about 0.1Å lower than that obtained at the other minima. The standard uncertainties (s.u.'s) of the X-H distances are also recorded via the usual notation using parentheses. More detailed information about these s.u.'s can be found in Tables 13.31 and 13.33 of Subsection 13.6.4. These larger X-H distances are in better agreement with results obtained via neutron diffraction experiments for L-alanine [34]. For epoxide and L-alanine, within the margins of their standard uncertainties, there is essential agreement of the X-H distances in all three nonspherical optima as well as the optimum obtained via mixed refinement. Due to its tighter uncertainties and larger differences, ammonia does not support this level of agreement, but it can be seen that $\mathbf{x}_{\text{num}}$ and $\mathbf{x}_{\text{hybr}}$ are in agreement, whilst $\mathbf{x}_{\text{mix}}$ and $\mathbf{x}_{\text{appr}}$ are only twice the standard uncertainty away.

13.5.2 Distances Between Optimal Models

Before we provide information about the distances between the five refinement optima, we first present the positional and ADP distances between $\mathbf{x}_{\text{num}}$ (the best with regard to wR$_2$within 20 numerical refinement steps starting from $\mathbf{x}_{\text{appr}}$) and $\mathbf{x}_{\text{num},s}$ (the best with regard to wR$_2$within 20 numerical refinement steps starting from $\mathbf{x}_{\text{spher}}$) in (13.5.1) for epoxide, in (13.5.2) for ammonia, and in (13.5.3) for L-alanine. The very small numbers in these entries bring the following ones in Tables 13.17 to 13.20 into perspective, and they confirm that numerical refinement is very robust and highly independent of the start models after sufficiently many refinement steps.

$$\begin{aligned} \|\mathbf{x}_{\text{num}}^{\text{pos,cart}} - \mathbf{x}_{\text{num},s}^{\text{pos,cart}}\| &= 6.408e - 05\,\text{Å}, \\ \|\mathbf{x}_{\text{num}}^{\text{ADP,cart}} - \mathbf{x}_{\text{num},s}^{\text{ADP,cart}}\| &= 2.720e - 05\,\text{Å}^2, \end{aligned} \tag{13.5.1}$$

$$\begin{aligned} \|\mathbf{x}_{\text{num}}^{\text{pos,cart}} - \mathbf{x}_{\text{num},s}^{\text{pos,cart}}\| &= 7.068e - 05\,\text{Å}, \\ \|\mathbf{x}_{\text{num}}^{\text{ADP,cart}} - \mathbf{x}_{\text{num},s}^{\text{ADP,cart}}\| &= 4.683e - 05\,\text{Å}^2, \end{aligned} \tag{13.5.2}$$

$$\begin{aligned} \|\mathbf{x}_{\text{num}}^{\text{pos,cart}} - \mathbf{x}_{\text{num},s}^{\text{pos,cart}}\| &= 3.808e - 05\,\text{Å}, \\ \|\mathbf{x}_{\text{num}}^{\text{ADP,cart}} - \mathbf{x}_{\text{num},s}^{\text{ADP,cart}}\| &= 6.297e - 05\,\text{Å}^2, \end{aligned} \tag{13.5.3}$$

`epoxide`	$\mathbf{x}_{\text{num}}^{\text{pos,cart}}$	$\mathbf{x}_{\text{hybr}}^{\text{pos,cart}}$	$\mathbf{x}_{\text{appr}}^{\text{pos,cart}}$	$\mathbf{x}_{\text{mix}}^{\text{pos,cart}}$	$\mathbf{x}_{\text{spher}}^{\text{pos,cart}}$
$\mathbf{x}_{\text{num}}^{\text{pos,cart}}$	0.000e+00	2.256e-04	6.261e-03	8.286e-03	2.376e-01
$\mathbf{x}_{\text{hybr}}^{\text{pos,cart}}$	2.256e-04	0.000e+00	6.416e-03	8.339e-03	2.375e-01
$\mathbf{x}_{\text{appr}}^{\text{pos,cart}}$	6.261e-03	6.416e-03	0.000e+00	5.946e-03	2.348e-01
$\mathbf{x}_{\text{mix}}^{\text{pos,cart}}$	8.286e-03	8.339e-03	5.946e-03	0.000e+00	2.333e-01
$\mathbf{x}_{\text{spher}}^{\text{pos,cart}}$	2.376e-01	2.375e-01	2.348e-01	2.333e-01	0.000e+00

Table 13.17: Distances between optimal models (in Å) for epoxide

Tables 13.17 and 13.18 provide information about the positional and ADP differences of our five optimal models for epoxide. These tables tell us the following for both the positional and the ADP components of the five models: The model $\mathbf{x}_{\text{appr}}$ is more than 10 times closer to $\mathbf{x}_{\text{num}}$ than $\mathbf{x}_{\text{spher}}$ is. Similarly, $\mathbf{x}_{\text{hybr}}$ is more than 10 times closer to $\mathbf{x}_{\text{num}}$ than $\mathbf{x}_{\text{appr}}$ is. Finally, the model $\mathbf{x}_{\text{mix}}$ is also significantly closer to $\mathbf{x}_{\text{num}}$ than $\mathbf{x}_{\text{spher}}$ is (but not as close as $\mathbf{x}_{\text{appr}}$). The corresponding results for L-alanine in Tables 13.19 and 13.20 provide a similar picture. If we think of $\mathbf{x}_{\text{num}}$ as the the "best" theoretical fit to the observed data (which is confirmed by the wR$_2$-factor), these values give insight into the "precision" of the different refinement processes.

epoxide	$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$
$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	0.000e+00	1.544e-04	1.791e-03	7.713e-03	8.345e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	1.544e-04	0.000e+00	1.844e-03	7.667e-03	8.345e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	1.791e-03	1.844e-03	0.000e+00	7.788e-03	8.410e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	7.713e-03	7.667e-03	7.788e-03	0.000e+00	8.291e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$	8.345e-02	8.345e-02	8.410e-02	8.291e-02	0.000e+00

Table 13.18: Norms of ADP differences between optimal models (in Å^2) for epoxide

L-alanine	$\mathbf{x}^{\text{pos,cart}}_{\text{num}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{hybr}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{appr}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{mix}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{spher}}$
$\mathbf{x}^{\text{pos,cart}}_{\text{num}}$	0.000e+00	3.553e-04	8.598e-03	1.146e-02	1.344e-01
$\mathbf{x}^{\text{pos,cart}}_{\text{hybr}}$	3.553e-04	0.000e+00	8.505e-03	1.139e-02	1.345e-01
$\mathbf{x}^{\text{pos,cart}}_{\text{appr}}$	8.598e-03	8.505e-03	0.000e+00	3.011e-03	1.270e-01
$\mathbf{x}^{\text{pos,cart}}_{\text{mix}}$	1.146e-02	1.139e-02	3.011e-03	0.000e+00	1.244e-01
$\mathbf{x}^{\text{pos,cart}}_{\text{spher}}$	1.344e-01	1.345e-01	1.270e-01	1.244e-01	0.000e+00

Table 13.19: Distances between optimal models (in Å) for L-alanine

L-alanine	$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$
$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	0.000e+00	1.870e-04	2.918e-03	5.154e-03	2.543e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	1.870e-04	0.000e+00	2.938e-03	5.151e-03	2.542e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	2.918e-03	2.938e-03	0.000e+00	2.944e-03	2.493e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	5.154e-03	5.151e-03	2.944e-03	0.000e+00	2.334e-02
$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$	2.543e-02	2.542e-02	2.493e-02	2.334e-02	0.000e+00

Table 13.20: Norms of ADP differences between optimal models (in Å^2) for L-alanine

`ammonia`	$\mathbf{x}^{\text{pos,cart}}_{\text{num}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{hybr}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{appr}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{mix}}$	$\mathbf{x}^{\text{pos,cart}}_{\text{spher}}$
$\mathbf{x}^{\text{pos,cart}}_{\text{num}}$	`0.000e+00`	`1.694e-04`	`7.461e-03`	`4.296e-04`	`2.659e-01`
$\mathbf{x}^{\text{pos,cart}}_{\text{hybr}}$	`1.694e-04`	`0.000e+00`	`7.529e-03`	`4.191e-04`	`2.659e-01`
$\mathbf{x}^{\text{pos,cart}}_{\text{appr}}$	`7.461e-03`	`7.529e-03`	`0.000e+00`	`7.553e-03`	`2.655e-01`
$\mathbf{x}^{\text{pos,cart}}_{\text{mix}}$	`4.296e-04`	`4.191e-04`	`7.553e-03`	`0.000e+00`	`2.660e-01`
$\mathbf{x}^{\text{pos,cart}}_{\text{spher}}$	`2.659e-01`	`2.659e-01`	`2.655e-01`	`2.660e-01`	`0.000e+00`

Table 13.21: Distances between optimal models (in Å) for ammonia

`ammonia`	$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$
$\mathbf{x}^{\text{ADP,cart}}_{\text{num}}$	`0.000e+00`	`9.749e-05`	`2.016e-03`	`1.167e-04`	`1.331e-01`
$\mathbf{x}^{\text{ADP,cart}}_{\text{hybr}}$	`9.749e-05`	`0.000e+00`	`2.018e-03`	`1.407e-04`	`1.331e-01`
$\mathbf{x}^{\text{ADP,cart}}_{\text{appr}}$	`2.016e-03`	`2.018e-03`	`0.000e+00`	`2.088e-03`	`1.340e-01`
$\mathbf{x}^{\text{ADP,cart}}_{\text{mix}}$	`1.167e-04`	`1.407e-04`	`2.088e-03`	`0.000e+00`	`1.330e-01`
$\mathbf{x}^{\text{ADP,cart}}_{\text{spher}}$	`1.331e-01`	`1.331e-01`	`1.340e-01`	`1.330e-01`	`0.000e+00`

Table 13.22: Norms of ADP differences between optimal models (in Å^2) for ammonia

13.5.3 Atomic Positions

Finally, we investigate the differences of the Cartesian positions and ADP components of individual atoms in the different optima. We also provide information about the reliability of the numerical optimum and its independence of the initial model, by comparing atomic data of the optimum $\mathbf{x}_{\text{num}}$ with the corresponding atomic data of the optimum $\mathbf{x}_{\text{num},s}$.

In Table 13.23, we compare the individual atom positions in Cartesian coordinates where $\mathbf{z}_*^{\text{cart}}$ denotes the 3 positional Cartesian coordinates of the chosen atom, at the final model obtained through the corresponding refinement with $* \in \{\text{num}, \text{appr}, \text{hybr}, \text{spher}\}$. We compare each final model against the final model $\mathbf{x}_{\text{num}}$. The first column shows the difference between final models of numerical nonspherical refinement processes obtained from different start models $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$, with the subscript 's' indicating that from $\mathbf{x}_{\text{spher}}$.

In both spherical and approximate nonspherical refinement, the hydrogen atoms are significantly further than heavier atoms from the positions at $\mathbf{x}_{\text{num}}$. This difference is dramatically reduced in the case of hybrid nonspherical refinement.

Table 13.23 demonstrates that approximate nonspherical refinement provides a 10- to 60-fold (median at 50-fold) improvement for hydrogen atoms, and a 10- to 300-fold (median at 50-fold) improvement for non-hydrogen atoms versus spherical refinement, when compared to numerical nonspherical refinement. Hybrid refinement then further provides a 40-fold (hydrogen) and 2-fold (non-hydrogen) improvement, typically in approximately a 50% time reduction versus numerical nonspherical refinement (since the proportion of hydrogen atoms is 4/7 in epoxide and 7/13 in L-alanine).

Mixed refinement shows an improvement also, with median improvements for hydrogen atoms being 30-fold and for non-hydrogen atoms being 15-fold.

Table 13.24 shows similar improvements (which are stronger for non-hydrogen atoms), with a median factor of 60 for hydrogen atoms and 150 for non-hydrogen atoms via approximate refinement, 15 for both hydrogen atoms and non-hydrogen atoms via further hybrid refinement, and 15 for hydrogen atoms and 40 for non-hydrogen atoms via mixed refinement.

	$\|\mathbf{z}^{\text{cart}}_{\text{num}} - \mathbf{z}^{\text{cart}}_{\text{num},s}\|$	$\|\mathbf{z}^{\text{cart}}_{\text{num}} - \mathbf{z}^{\text{cart}}_{\text{hybr}}\|$	$\|\mathbf{z}^{\text{cart}}_{\text{num}} - \mathbf{z}^{\text{cart}}_{\text{appr}}\|$	$\|\mathbf{z}^{\text{cart}}_{\text{num}} - \mathbf{z}^{\text{cart}}_{\text{mix}}\|$	$\|\mathbf{z}^{\text{cart}}_{\text{num}} - \mathbf{z}^{\text{cart}}_{\text{spher}}\|$
Ammonia					
N	4.734e-06	4.754e-05	9.513e-05	1.841e-04	3.592e-03
H	7.052e-05	3.521e-04	8.598e-03	1.146e-02	1.343e-01
Epoxide					
O1	5.997e-07	1.198e-05	3.014e-05	1.190e-04	5.112e-03
C2	1.434e-07	1.525e-05	3.207e-05	1.974e-04	1.829e-03
H2a	1.318e-05	3.642e-05	1.722e-03	2.301e-03	1.011e-01
H2b	2.944e-05	8.945e-05	1.799e-03	4.157e-03	1.131e-01
C3	7.023e-07	2.908e-05	1.154e-04	2.450e-04	3.340e-03
H3a	5.194e-05	1.813e-04	5.085e-03	5.140e-03	1.344e-01
H3b	1.917e-05	8.638e-05	2.668e-03	4.423e-03	1.238e-01
L-Alanine					
N1	5.023e-07	1.275e-05	3.137e-05	1.251e-06	8.932e-04
H1a	1.513e-05	8.696e-05	3.723e-03	9.037e-05	1.076e-01
H1b	1.495e-05	1.927e-05	1.736e-03	2.182e-04	9.296e-02
H1c	1.664e-05	1.116e-04	4.864e-03	6.021e-05	7.742e-02
C1	2.286e-07	7.776e-06	7.121e-06	3.403e-06	2.624e-03
H1	1.709e-05	6.874e-05	2.242e-03	2.194e-04	9.049e-02
C2	1.018e-07	6.226e-06	1.933e-05	6.774e-07	2.788e-04
H2a	1.443e-05	2.361e-05	1.990e-03	1.637e-04	1.114e-01
H2b	6.426e-06	1.461e-05	2.074e-03	1.832e-04	1.281e-01
H2c	1.337e-05	4.576e-05	1.359e-03	1.288e-04	8.655e-02
C3	1.687e-07	1.188e-05	7.515e-06	2.392e-06	7.201e-04
O1	1.548e-07	1.038e-05	2.556e-05	1.421e-06	2.022e-03
O2	2.601e-07	1.469e-05	2.130e-05	1.086e-06	8.635e-04

Table 13.23: Distances between the locations of atoms of $\mathbf{x}_{\text{num}}$ versus other models (in Å)

This shows that there is a significant improvement from spherical to approximate nonspherical refinement, and a further improvement from approximate to hybrid

	$\|\mathbf{z}_{\text{num}}^{\text{cart,ADP}} - \mathbf{z}_{\text{num,s}}^{\text{cart,ADP}}\|$	$\|\mathbf{z}_{\text{num}}^{\text{cart,ADP}} - \mathbf{z}_{\text{hybr}}^{\text{cart,ADP}}\|$	$\|\mathbf{z}_{\text{num}}^{\text{cart,ADP}} - \mathbf{z}_{\text{appr}}^{\text{cart,ADP}}\|$	$\|\mathbf{z}_{\text{num}}^{\text{cart,ADP}} - \mathbf{z}_{\text{mix}}^{\text{cart,ADP}}\|$	$\|\mathbf{z}_{\text{num}}^{\text{cart,ADP}} - \mathbf{z}_{\text{spher}}^{\text{cart,ADP}}\|$
Ammonia					
N	2.114e-06	6.037e-06	3.572e-04	7.250e-04	1.972e-03
H	4.678e-05	1.869e-04	2.896e-03	5.103e-03	2.536e-02
Epoxide					
O1	1.285e-07	9.252e-07	1.271e-05	1.493e-04	2.679e-03
C2	1.403e-07	8.177e-07	1.430e-05	2.019e-04	1.853e-03
H2a	8.951e-06	2.433e-05	5.745e-04	5.361e-03	4.276e-02
H2b	1.565e-05	6.760e-05	7.845e-04	2.208e-03	4.525e-02
C3	2.378e-07	1.633e-06	2.415e-05	1.953e-04	1.583e-03
H3a	1.739e-05	8.065e-05	8.633e-04	4.322e-03	4.159e-02
H3b	1.060e-05	1.104e-04	1.232e-03	2.663e-03	3.668e-02
L-Alanine					
N1	1.118e-07	6.927e-07	1.197e-05	4.431e-07	6.211e-04
H1a	1.068e-05	5.539e-05	4.618e-04	5.228e-05	5.186e-02
H1b	4.863e-06	2.068e-05	3.718e-04	3.281e-05	2.991e-02
H1c	5.864e-05	6.180e-05	8.752e-04	6.942e-05	1.041e-01
C1	2.701e-08	5.264e-07	6.396e-06	1.670e-07	9.630e-04
H1	3.988e-06	2.313e-05	3.791e-04	2.193e-05	2.593e-02
C2	5.387e-08	3.628e-07	4.975e-06	6.768e-07	7.760e-04
H2a	1.652e-05	2.087e-05	1.276e-03	4.567e-05	2.999e-02
H2b	6.132e-06	2.078e-05	3.321e-04	3.574e-05	2.103e-02
H2c	7.854e-06	2.801e-05	1.033e-03	3.379e-05	3.582e-02
C3	3.433e-08	2.221e-07	6.221e-06	2.301e-07	1.002e-03
O1	4.128e-08	2.483e-07	5.503e-06	1.679e-07	9.886e-04
O2	2.533e-08	3.740e-07	4.666e-06	2.701e-07	7.778e-04

Table 13.24: Distances between the ADPs of atoms of $\mathbf{x}_{\text{num}}$ versus other models (in Å^2)

nonspherical refinement with regards to the hydrogen atom positions.

13.5.4 Basis Set Dependence

When utilising quantum mechanical calculations, the user needs to select a method, a basis set, and also needs to utilise an accuracy for the integration grids, which can be selected from the pre-defined settings “High", “Normal" and “Low" in *NoSpherA2*.

We carried out refinement procedures with the method of PBE with five basis sets for ammonia - 3-21G, def2-SVP, def2-TZVP, cc-pVTZ, cc-pVQZ (for details on these basis sets, see [48]) - and our results are shown in the following figures.

In Figure 13.37 we compare the hydrogen positions of the results using various options for the basis sets in 3d space.

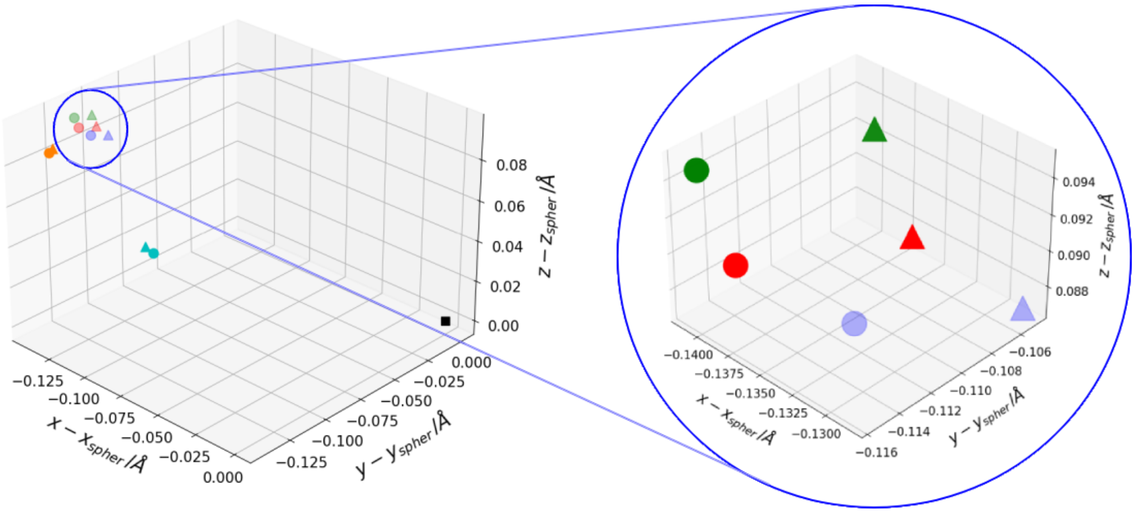


Figure 13.37: The position of the hydrogen atom (in Å), obtained through classical spherical refinement (■) and non-spherical refinement processes (approximate ▲ and numerical ●) using various basis sets, where the hydrogen atom in the classical spherical model is shifted to the origin (0,0,0). The integration accuracy used is ‘Normal’. (ammonia)

Figure 13.37 clearly indicates that the difference between smaller basis sets (3-21G (lower cluster) & def2-SVP (left cluster)) and larger ones (def2-TZVP, cc-pVTZ and cc-pVQZ, emphasised in the zoomed in image) is far more significant than the difference between approximate non-spherical and numerical non-spherical refinements. Additionally, the difference between larger basis sets is roughly of the same magnitude as the change introduced by using numerical rather than approximate non-spherical refinement. We finally note here that every larger basis set results in an additional movement in the $-y$ direction when numerical non-spherical refinement is used instead of approximate non-spherical refinement.

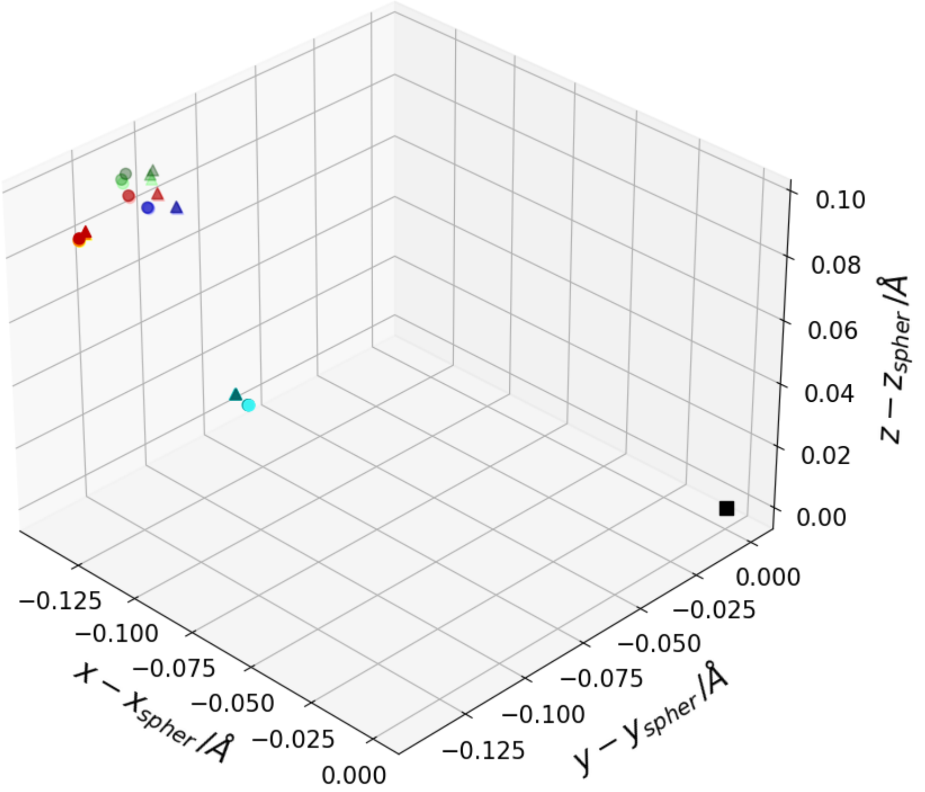


Figure 13.38: The position of the hydrogen atom (in Å) at the final model, obtained through classical spherical refinement (■) and non-spherical refinement processes (approximate ▲ and numerical ●) using various basis sets, where the hydrogen atom in the classical spherical model is shifted to the origin (0,0,0).
The colours are as follows: cc-pVQZ (top), cc-pVTZ (second-top), def2-TZVP (third top), 3-21G (lower cluster), def2-SVP (left cluster), with darkened and lightened versions of these colours corresponding to 'low' or 'high' rather than 'normal' integration accuracy, respectively. (ammonia)

Figures 13.38 and 13.39 add in the 'High' integration grid (shown as a lightened colour) and the 'Low' integration grid (shown as a darkened colour). It is clear that the impact of these is minimal compared to the other differences. It also indicates that the choice of integration accuracy has a far smaller impact than these other

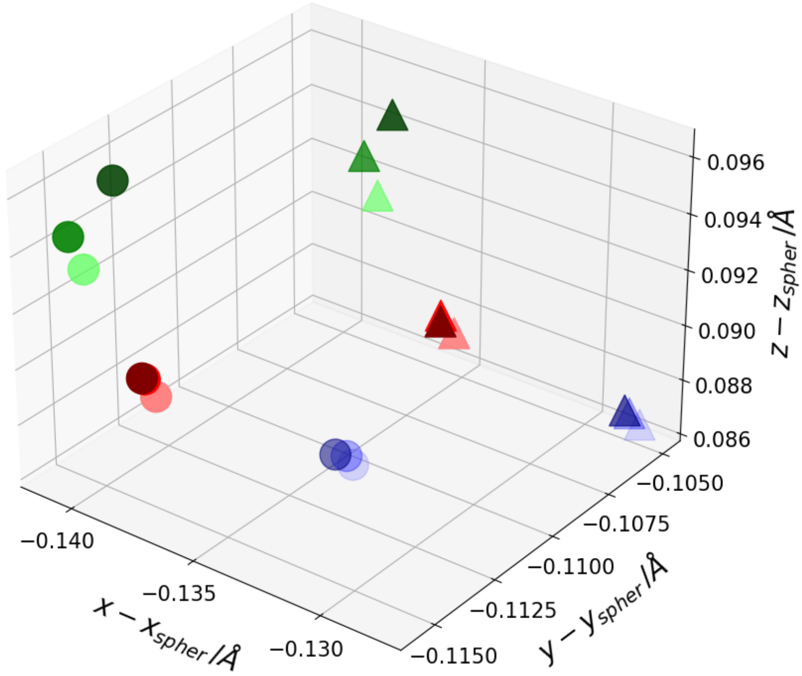


Figure 13.39: A small area of Figure 13.37: The position of the hydrogen atom at the final model, obtained with the larger three basis sets, cc-pVQZ (top), cc-pVTZ (middle), def2-TZVP (bottom), relative to the position of the hydrogen in the classical spherical model (which would appear at (0,0,0)). (ammonia)

choices, with even low accuracy of the integration grid remaining very close to high - however, this might only be a symptom of the small molecule we are using.

Note, however, that all figures in this section are presented without taking into account that all observed and derived data come with their standard uncertainties. Uncertainty considerations are the focus of the following Section 13.6.

13.6 Uncertainties

In crystallography, each parameter (such as atomic location) has an associated uncertainty which is related to both the uncertainty on the recorded data (for example, the σ recorded in the hkl file) and the least squares process itself. To provide a final evaluation of numerical and approximate nonspherical refinement, I investigate their parameters with regards to their uncertainties to find that, up to this uncertainty, approximate nonspherical refinement provides as accurate a model as we can currently produce. This section covers the background of these uncertainties as well as a mathematical visualisation which allows us to better see these similarities.

13.6.1 The Variance-Covariance Matrix and transformations

Background

Recall from Part I Section 6.8 that the variance-covariance matrix arising from the least-squares process is given by the following rescaling of the inverse of the normal matrix $B(\mathbf{x})$:

$$\mathrm{Var}(\mathbf{x}) = \frac{||\tilde{\mathbf{r}}||_w^2}{\#_{obs} - \#_{param}} B(\mathbf{x})^{-1} \tag{13.6.1}$$

(3.1.10.2 in [53]), where $\mathbf{x}$ is the vector of model parameters, $||\tilde{\mathbf{r}}||_w$ is the weighted norm introduced in (6.0.3) of the residual $\tilde{\mathbf{r}}$, $\#_{obs}$ is the number of observations taken into account in the refinement process and $\#_{param}$ is the number of parameters contained in the vectors $\mathbf{x}_{\mathrm{num}}$, $\mathbf{x}_{\mathrm{appr}}$ and $\mathbf{x}_{\mathrm{spher}}$ (that is, their length).

The diagonals of the variance-covariance matrix $\mathrm{Var}(\mathbf{x})$ are the variance $\sigma^2(\mathbf{x}, x_i) =: \sigma_i^2(\mathbf{x})$ of their corresponding parameter x_i whilst the off-diagonals are the covariance of the two corresponding parameters $\mathrm{cov}(\mathbf{x}, x_i, x_j) =: \mathrm{cov}_{i,j}(\mathbf{x})$. $\mathrm{Var}(\mathbf{x})$ can additionally be split into smaller variance-covariance matrices corresponding to each atom's positional or ADP parameters separately.

In our derivations we will make repeated use of the following fundamental rule (see, e.g., [14, III.3],[53, formula (3.1.10.5)]): If random vectors $\mathbf{v}$ and $\mathbf{w}$ are transformed via $\mathbf{w} = Q\mathbf{v}+\mathbf{b}$ (with a fixed matrix Q and a fixed vector $\mathbf{b}$) then their corresponding variance-covariance matrices are related by

$$\mathrm{Var}(\mathbf{w}) = Q\,\mathrm{Var}(\mathbf{v})\,Q^{\top}. \tag{13.6.2}$$

The first transformation from the vector $\mathbf{x}$ (representing a model) into the crystallographic parameters $\mathbf{y}$ is often the identity (in the case of epoxide and L-alanine), but in other cases with constraints (like ammonia) the expansion may be given by some matrix J, that is $\mathbf{y} = J\mathbf{x}$ (the choice of J for ammonia is

$$J = \begin{bmatrix} J_0 & 0_{3,9} \\ 0_{9,9} & \mathrm{Id}_9 \end{bmatrix} \tag{13.6.3}$$

with $0_{p,q}$ the $p \times q$ matrix with all entries equal to 0, Id_p the identity matrix of size p and

$$J_0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}).$$

Application of the above rule yields $\mathrm{Var}(\mathbf{y}) = J\,\mathrm{Var}(\mathbf{x})\,J^{\top}$ (see also (35) in [4]).

The variance-covariance matrix of the fractional positional coordinates $\mathbf{z}^A$ of an atom is a specific 3x3 submatrix of $\mathrm{Var}(\mathbf{y})$. The transformation in Cartesian coordinates is given through multiplication with the orthogonalisation matrix A, that is, $\mathbf{z}^{\mathrm{cart}} = A\mathbf{z}^A$. Consequently, the above rule yields

$$\mathrm{Var}(\mathbf{z}^{\mathrm{cart}}) = A\,\mathrm{Var}(\mathbf{z}^A)\,A^{\top}. \tag{13.6.4}$$

For the ADPs, we utilise the matrix $U^* = \begin{bmatrix} U^*_{11} & U^*_{12} & U^*_{13} \\ U^*_{12} & U^*_{22} & U^*_{23} \\ U^*_{13} & U^*_{23} & U^*_{33} \end{bmatrix}$ and transform it with the help of the orthogonalisation matrix A into a Cartesian matrix

$$U^{\text{cart}} = \begin{bmatrix} U^{\text{cart}}_{11} & U^{\text{cart}}_{12} & U^{\text{cart}}_{13} \\ U^{\text{cart}}_{12} & U^{\text{cart}}_{22} & U^{\text{cart}}_{23} \\ U^{\text{cart}}_{13} & U^{\text{cart}}_{23} & U^{\text{cart}}_{33} \end{bmatrix} = A \begin{bmatrix} U^*_{11} & U^*_{12} & U^*_{13} \\ U^*_{12} & U^*_{22} & U^*_{23} \\ U^*_{13} & U^*_{23} & U^*_{33} \end{bmatrix} A^\top.$$

This transformation can be equivalently written as

$$\begin{bmatrix} U^{\text{cart}}_{11} & U^{\text{cart}}_{22} & U^{\text{cart}}_{33} & U^{\text{cart}}_{12} & U^{\text{cart}}_{13} & U^{\text{cart}}_{23} \end{bmatrix}^\top = B \begin{bmatrix} U^*_{11} & U^*_{22} & U^*_{33} & U^*_{12} & U^*_{13} & U^*_{23} \end{bmatrix}^\top \tag{13.6.5}$$

with a suitably chosen 6x6 matrix B, which can then be used with the above rule to obtain $\mathrm{Var}(U^{\text{cart}})$ (see [45] for further details).

13.6.2 Uncertainty Domains

We now outline the shape of the *domain* of uncertainty of an atomic position. This shape is in Cartesian space, so we first must convert the variance-covariance matrix as outlined in Equation (13.6.4).

We give our atom the label $\mathcal{A}$, and the variance-covariance matrix associated to its location we label $V^{\mathcal{A}\mathcal{A}}$[3], with the transformation to Cartesian being:

$$V^{\mathcal{A}\mathcal{A},\text{cart}} := AV^{\mathcal{A}\mathcal{A}}A^\top,$$

where A is the standard orthogonalisation matrix taking $\mathbf{z}^{\mathcal{A}}$ to $\mathbf{z}^{\mathcal{A},\text{cart}}$.

Since $V^{\mathcal{A}\mathcal{A},\text{cart}}$ is positive semidefinite, it has a natural square root[4], which we denote by $(V^{\mathcal{A}\mathcal{A},\text{cart}})^{1/2}$.

[3] Later, we will introduce $V^{\mathcal{A},\mathcal{C}}$, the covariance between two different atoms.

[4] The square root of a positive semidefinite 3×3 matrix M can be obtained as follows: Choose an orthogonal matrix P (that is, $P^\top = P^{-1}$) such that $D = PMP^\top$ is diagonal with entries $d_1, d_2, d_3 \geq 0$, the eigenvalues of M. Let $D^{1/2}$ be the corresponding diagonal matrix with entries $\sqrt{d_1}, \sqrt{d_2}, \sqrt{d_3} \geq 0$. Then the square root of M is given by $P^\top D^{1/2} P$.

We introduce the *uncertainty domain* $\mathcal{U}(\mathcal{A})$ of the atom $\mathcal{A}$ as the image of the unit ball $B_1(0) = \{\mathbf{z} \mid \|\mathbf{z}\| \leq 1\}$ under $(V^{\mathcal{AA},\text{cart}})^{1/2}$, shifted to $\mathbf{z}^{\mathcal{A}} \in \mathbb{R}^3$, that is

$$\mathcal{U}(\mathcal{A}) := \mathbf{z}^{\mathcal{A},\text{cart}} + (V^{\mathcal{AA},\text{cart}})^{1/2} B_1(0) = \left\{\mathbf{z}^{\mathcal{A},\text{cart}} + (V^{\mathcal{AA},\text{cart}})^{1/2}\mathbf{z} \mid \mathbf{z} \in B_1(0)\right\}.$$

Note that, in the special case $V^{\mathcal{AA},\text{cart}} = \sigma_0^2 \cdot \text{Id}_3$, $c > 0$ (where σ_0 is simply a constant uncertainty shared across all parameters) the Cartesian position of atom $\mathcal{A}$ is $\mathcal{N}(\mathbf{z}^{\mathcal{A},\text{cart}}, \sigma_0^2 \cdot \text{Id}_3)$–normal distributed and the uncertainty domain $\mathcal{U}(\mathcal{A})$ is simply the ball of radius σ_0 about $\mathbf{z}^{\mathcal{A}} \in \mathbb{R}^3$. In relation to the uncertainty domain $\mathcal{U}(\mathcal{A})$, we also introduce two uncertainty radii $r(\mathcal{A})$ and $R(\mathcal{A})$ which are, geometrically, the radii of two balls centered at $\mathcal{A}$, one of them being contained in $\mathcal{U}(\mathcal{A})$ and the other one containing $\mathcal{U}(\mathcal{A})$. The radii $r(\mathcal{A})$ and $R(\mathcal{A})$ are the square roots of the smallest and the largest eigenvalue of the matrix $V^{\mathcal{AA},\text{cart}}$, and they are called the *inner* and *outer uncertainty radius* of the atom $\mathcal{A}$, respectively. In the special case $V^{\mathcal{AA},\text{cart}} = \sigma_0^2 \cdot \text{Id}_3$ we have $r(\mathcal{A}) = R(\mathcal{A}) = \sigma_0$.

Physical Justification

We now cover the justification and meaning of these uncertainty domains with a more intuitive derivation which nevertheless matches to the above definitions. Starting from the distance of an atom $\mathcal{A}$ to a reference point $\mathbf{z}^{\text{ref}}$ (in relative coordinates), we consider the limiting process moving the reference point $\mathbf{z}^{\text{ref}}$ towards the Cartesian position of the atom until both points coincide.

Lemma 13.6.1 (Uncertainty on Distance between Two Points)**.** Given these two points $\mathbf{z}^{\mathcal{A}}$ and $\mathbf{z}^{\text{ref}}$, the Cartesian distance d between them is given by

$$d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\text{ref}}) = \left((\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\text{ref}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\text{ref}})\right)^{1/2} = \left(\sum_{i,j=1}^{3} (z_i^{\mathcal{A}} - z_i^{\text{ref}})(z_j^{\mathcal{A}} - z_j^{\text{ref}}) M_{ij}\right)^{1/2}, \tag{13.6.6}$$

where $M = A^\top A$ denotes the metrical metrix, which leads to partial derivatives

$$\frac{\partial d}{\partial z_i^{\mathcal{A}}}(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}}) = \frac{1}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}})} \sum_{j=1}^{3} (z_j^{\mathcal{A}} - z_j^{\mathrm{ref}}) M_{ij}$$

and, therefore (see Part I Section 6.8 for details),

$$\begin{aligned} \sigma^2(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}})) &= \sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{A}}}(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}}) \frac{\partial d}{\partial z_j^{\mathcal{A}}}(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}}) V_{ij}^{\mathcal{A}\mathcal{A}} \\ &= \frac{1}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}})^2} \sum_{i,j=1}^{3} \left(\sum_{l=1}^{3} (z_l^{\mathcal{A}} - z_l^{\mathrm{ref}}) M_{il} \right) \left(\sum_{m=1}^{3} (z_m^{\mathcal{A}} - z_m^{\mathrm{ref}}) M_{jm} \right) V_{ij}^{\mathcal{A}\mathcal{A}} \\ &= \frac{1}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}})^2} \sum_{i,j,l,m=1}^{3} (z_l^{\mathcal{A}} - z_l^{\mathrm{ref}}) M_{li} V_{ij}^{\mathcal{A}\mathcal{A}} M_{jm} (z_m^{\mathcal{A}} - z_m^{\mathrm{ref}}) \\ &= \frac{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathrm{ref}})^\top M V^{\mathcal{A}\mathcal{A}} M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathrm{ref}})}{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathrm{ref}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathrm{ref}})}. \end{aligned} \tag{13.6.7}$$

We notice the following useful fact about this uncertainty $\delta = \sigma(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathrm{ref}}))$:

The uncertainty δ does not change if we replace $\mathbf{z}^{\mathrm{ref}} - \mathbf{z}^{A}$ by $\mu(\mathbf{z}^{\mathrm{ref}} - \mathbf{z}^{A})$ in (13.6.7) with any scaling factor $\mu \neq 0$. Thus this uncertainty is the same for every reference point $\mathbf{z}'$ on the line

$$\mathbf{z}^{\mathcal{A}} + \mu(\mathbf{z}^{\mathrm{ref}} - \mathbf{z}^{\mathcal{A}}). \tag{13.6.8}$$

Given $\mathbf{z}^{\mathrm{ref}}$ a fixed reference point different from the position of the atom $\mathcal{A}$ and $\mathbf{z}'$ be another reference point moving towards the atom $\mathcal{A}$ along the line given by (13.6.8), that is, we assume $\mu \to 0$ as well as $d \to 0$. We conclude from the above fact that the uncertainty δ of the distance of atom $\mathcal{A}$ to $\mathbf{z}'$ does not change and we introduce, in the limit $\mu = 0$,the uncertainty radius of the atom $\mathcal{A}$ in direction $\mathbf{z}^{\mathrm{ref}}$ by $u(\mathcal{A}, \mathbf{z}^{\mathrm{ref}})$. Hence, $u(\mathcal{A}, \mathbf{z}^{\mathrm{ref}})$ and the s.u. $\sigma_{xyz}(d(\mathcal{A}, \mathbf{z}^{\mathrm{ref}}))$ of the distance of atom $\mathcal{A}$ to a reference point $\mathbf{z}^{\mathrm{ref}} \neq \mathcal{A}$ coincide.

We recover the boundary of the uncertainty domain $\mathcal{U}(\mathcal{A})$ via taking this uncertainty radius in every direction around $\mathcal{A}$:

$$\partial \mathcal{U}(\mathcal{A}) = \left\{ \mathbf{z}^{\mathcal{A}} + u(\mathcal{A}, \mathbf{z}^{\mathrm{ref}}) \cdot \frac{\mathbf{z}^{\mathrm{ref}} - \mathbf{z}^{\mathcal{A}}}{\|\mathbf{z}^{\mathrm{ref}} - \mathbf{z}^{\mathcal{A}}\|} \middle| \mathbf{z}^{\mathrm{ref}} \neq \mathbf{z}^{\mathcal{A}} \right\}.$$

Lemma 13.6.2. While $u(\mathcal{A},\mathbf{z}^{\text{ref}})$ depends on the direction $\mathbf{z}^{\text{ref}}-\mathbf{z}^{\mathcal{A}}$ (but not on its length), the inner and outer uncertainty radii $r(\mathcal{A})$ and $R(\mathcal{A})$ are independent of any direction and coincide with the extrema of the *directed* uncertainty radii:

$$r(\mathcal{A}) = \min_{\mathbf{z}^{\text{ref}}\neq\mathbf{z}^{\mathcal{A}}} u(\mathcal{A},\mathbf{z}^{\text{ref}}) = \left(\min_{\mathbf{v}\neq 0} \frac{\mathbf{v}^{\top}MV^{\mathcal{A}\mathcal{A}}M\mathbf{v}}{\mathbf{v}^{\top}M\mathbf{v}}\right)^{1/2} > 0, \tag{13.6.9}$$

and

$$R(\mathcal{A}) = \max_{\mathbf{z}^{\text{ref}}\neq\mathbf{z}^{\mathcal{A}}} u(\mathcal{A},\mathbf{z}^{\text{ref}}) = \left(\max_{\mathbf{v}\neq 0} \frac{\mathbf{v}^{\top}MV^{\mathcal{A}\mathcal{A}}M\mathbf{v}}{\mathbf{v}^{\top}M\mathbf{v}}\right)^{1/2} > 0, \tag{13.6.10}$$

where for simplicity we have taken $\mathbf{v} = \mathbf{z}^{\text{ref}}-\mathbf{z}^{\mathcal{A}}$. In particular, we have $r(\mathcal{A}) \leq u(\mathcal{A}) \leq R(\mathcal{A})$ for all $\mathbf{z}^{\text{ref}} \neq \mathbf{z}^{\mathcal{A}}$.

Proof. We begin with the knowledge that $r(\mathcal{A})$ and $R(\mathcal{A})$ are the square roots of the smallest and largest eigenvalues of $V^{\mathcal{A}\mathcal{A},\text{cart}}$ respectively. We use the variational characterisation of eigenvalues via the *Rayleigh Quotient* (see eg. page 176 in [25]):

$$r^2(\mathcal{A}) = \lambda_{\min}(V^{\mathcal{A}\mathcal{A},\text{cart}}) = \min_{\mathbf{w}\neq 0}\left(\frac{\mathbf{w}^{\top}V^{\mathcal{A}\mathcal{A},\text{cart}}\mathbf{w}}{\mathbf{w}^{\top}\mathbf{w}}\right). \tag{13.6.11}$$

Then it follows from the expansion of $V^{\mathcal{A}\mathcal{A}}$:

$$r^2(\mathcal{A}) = \min_{\mathbf{w}\neq 0}\left(\frac{\mathbf{w}^{\top}AV^{\mathcal{A}\mathcal{A}}A^{\top}\mathbf{w}}{\mathbf{w}^{\top}\mathbf{w}}\right).$$

Finally, we take $\mathbf{w} = A\mathbf{v}$ such that:

$$r^2(\mathcal{A}) = \min_{\mathbf{v}\neq 0}\left(\frac{\mathbf{v}^{\top}A^{\top}AV^{\mathcal{A}\mathcal{A}}A^{\top}A\mathbf{v}}{\mathbf{v}^{\top}A^{\top}A\mathbf{v}}\right)$$

which, given $A^{\top}A = M$, matches perfectly to Equation (13.6.9). Trading the min for max gives the proof for $R(\mathcal{A})$. □

Whilst we have here used the positional parameters $\mathbf{z}$, a similar approach may be taken for the ADPs U^{*}, in the vector form outlined above.

All above uncertainty radii (inner r, outer R and directed uncertainty radius u) depend on the variance-covariance matrix $V^{\mathcal{A}\mathcal{A}}(\mathbf{x})$ which, in turn, depends on the refinement process whose optimal model $\mathbf{x}$ we investigate via (13.6.1). Denoting the atom $\mathcal{A}$ of the optimal model $\mathbf{x}_{\text{spher}}$ by $\mathcal{A}_{\text{spher}}$, we refer to the corresponding inner and outer uncertainty radii as $r(\mathcal{A}_{\text{spher}})$ and $R(\mathcal{A}_{\text{spher}})$. The uncertainty radii $r(\mathcal{A}_\bullet)$ and $R(\mathcal{A}_\bullet)$ for $\bullet \in \{\text{mix}, \text{appr}, \text{hybr}, \text{num}\}$ are defined similarly.

13.6.3 Uncertainty Balls

These inner and outer uncertainty radii can be represented by balls centered at the atom, and these balls can be plotted to allow us to compare corresponding atoms of the different final models with respect to their uncertainties.

Let us briefly discuss our method of illustrating uncertainties. In any dimension, three points always lie in a single plane (not necessarily uniquely). We apply this fact to the atomic positions of corresponding atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{spher}}$ across our 3 molecules. The atoms are represented by a white diamond for $\mathbf{x}_{\text{num}}$, a black dot for $\mathbf{x}_{\text{appr}}$, and a red square for $\mathbf{x}_{\text{spher}}$. With regards to the balls we use the following colour code: green balls represent inner uncertainty balls $B_{r(\mathcal{A})}(\mathbf{z}^{\mathcal{A}})$ and red balls represent outer uncertainty balls $B_{R(\mathcal{A})}(\mathbf{z}^{\mathcal{A}})$. We plot each atom's relative location horizontally with the spherical location on the right and the nonspherical approximate and numerical locations on the left. The plots are collated by atom type (hydrogen or non-hydrogen) for each molecule, but the separate atoms in each row are plotted independently.

Epoxide

Figures 13.40, 13.41 and Table 13.25 confirm that each individual atom of $\mathbf{x}_{\text{num}}$ and $\mathbf{x}_{\text{appr}}$ lie well within each others' uncertainty bounds, whilst the corresponding atom of $\mathbf{x}_{\text{spher}}$ lies much further away. We conclude that there is such a strong agreement between approximate and numerical nonspherical refinement that the simplification of setting $(*)$ in (12.2.3) to zero is fully justified. Moreover, these figures reveal also that approximate nonspherical refinement is a significant improvement against

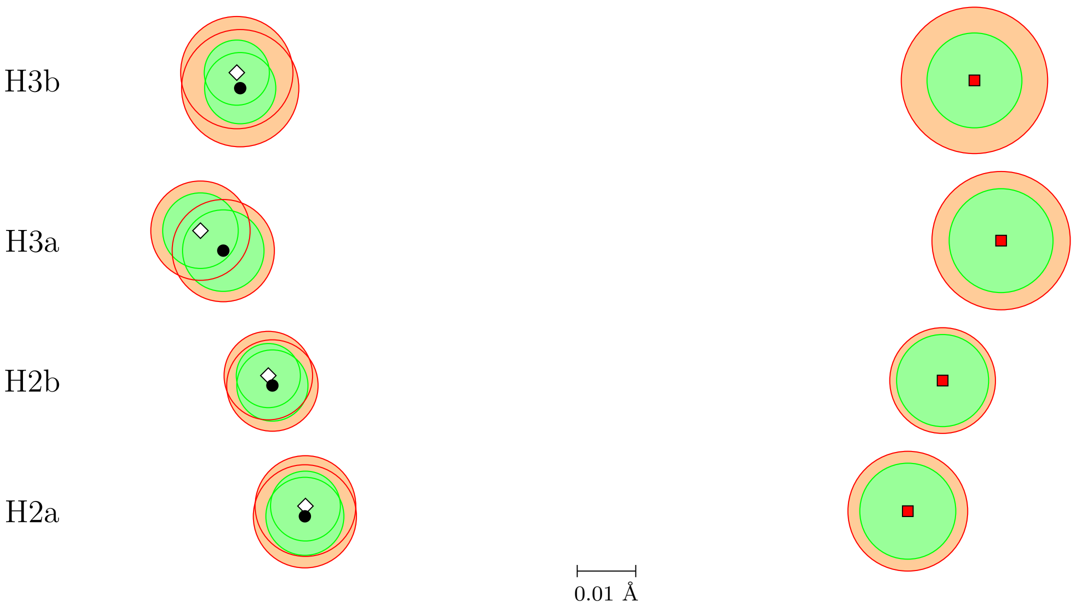


Figure 13.40: Positions of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (epoxide)

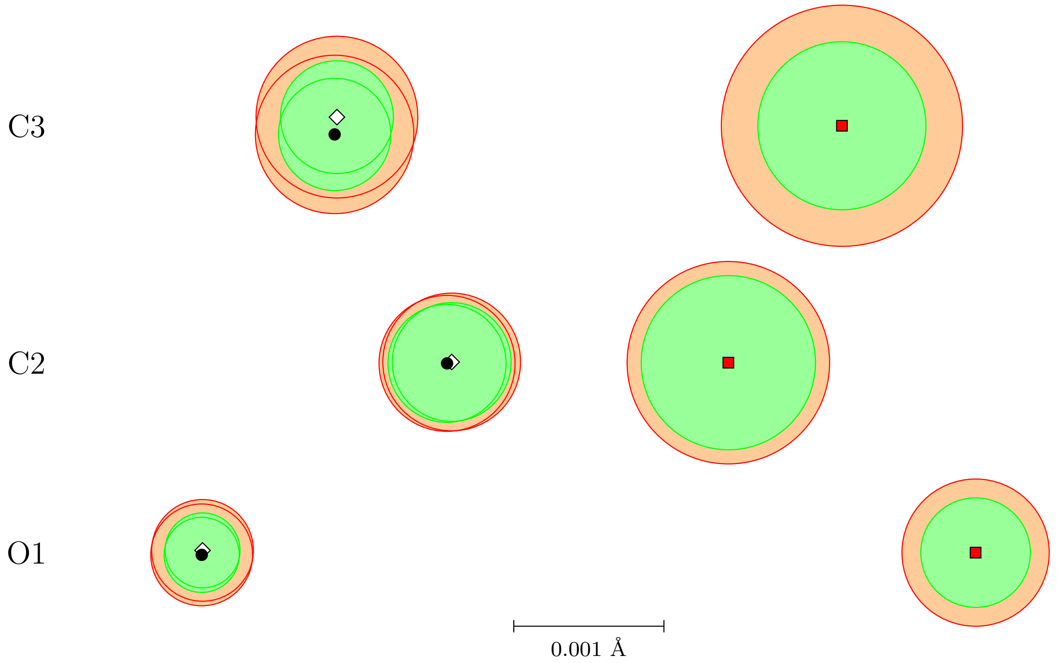


Figure 13.41: Positions of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (epoxide)

	$\lVert \mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}} \rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A}$ =O1	5.112e-03	2.479e-04	3.363e-04	3.624e-04	4.868e-04
$\mathcal{A}$ =C2	1.829e-03	3.928e-04	4.554e-04	5.763e-04	6.695e-04
$\mathcal{A}$ =H2a	1.011e-01	5.859e-03	8.449e-03	8.055e-03	1.005e-02
$\mathcal{A}$ =H2b	1.131e-01	5.380e-03	7.425e-03	7.726e-03	8.868e-03
$\mathcal{A}$ =C3	3.340e-03	3.731e-04	5.349e-04	5.559e-04	7.975e-04
$\mathcal{A}$ =H3a	1.344e-01	6.341e-03	8.333e-03	8.716e-03	1.160e-02
$\mathcal{A}$ =H3b	1.238e-01	5.464e-03	9.420e-03	7.960e-03	1.228e-02
	$\lVert \mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}} \rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A}$ =O1	3.014e-05	2.479e-04	3.363e-04	2.485e-04	3.368e-04
$\mathcal{A}$ =C2	3.207e-05	3.928e-04	4.554e-04	3.903e-04	4.497e-04
$\mathcal{A}$ =H2a	1.722e-03	5.859e-03	8.449e-03	6.546e-03	8.652e-03
$\mathcal{A}$ =H2b	1.799e-03	5.380e-03	7.425e-03	5.975e-03	7.668e-03
$\mathcal{A}$ =C3	1.154e-04	3.731e-04	5.349e-04	3.715e-04	5.238e-04
$\mathcal{A}$ =H3a	5.085e-03	6.341e-03	8.333e-03	6.837e-03	8.576e-03
$\mathcal{A}$ =H3b	2.668e-03	5.464e-03	9.420e-03	5.976e-03	9.838e-03

Table 13.25: Positional difference (in Å) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (epoxide).

classical spherical refinement.

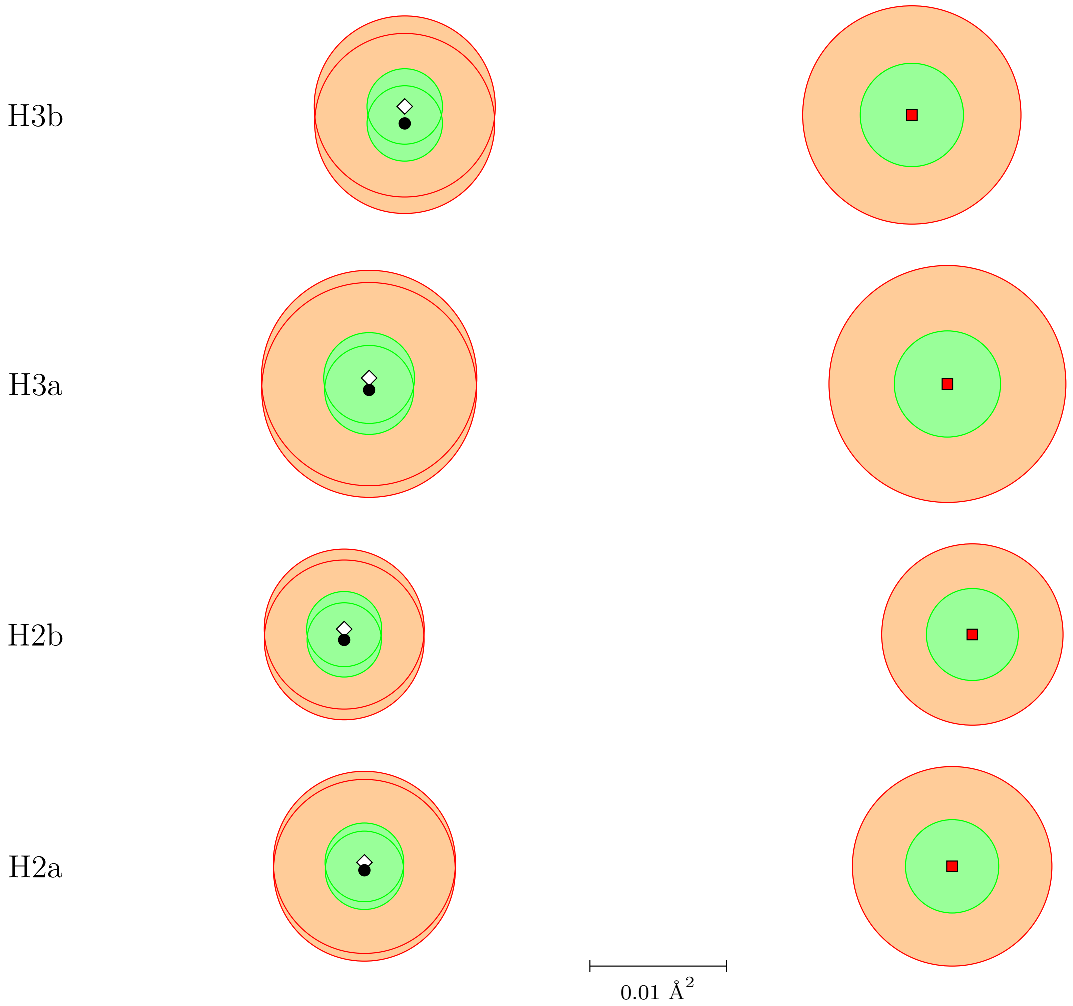


Figure 13.42: ADPs of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (epoxide)

The ADPs for epoxide 13.42, 13.43 and Table 13.26 show a similar story, with numerical and approximate atom locations within each others' inner uncertainties and far from spherical atom's location.

	$\|\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}}\|$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A}$ =O1	2.679e-03	9.246e-05	1.535e-04	1.338e-04	2.259e-04
$\mathcal{A}$ =C2	1.853e-03	1.222e-04	1.985e-04	1.769e-04	2.942e-04
$\mathcal{A}$ =H2a	4.276e-02	2.854e-03	6.604e-03	3.384e-03	7.235e-03
$\mathcal{A}$ =H2b	4.525e-02	2.723e-03	5.802e-03	3.333e-03	6.576e-03
$\mathcal{A}$ =C3	1.583e-03	1.282e-04	2.364e-04	1.862e-04	3.480e-04
$\mathcal{A}$ =H3a	4.159e-02	3.291e-03	7.811e-03	3.851e-03	8.591e-03
$\mathcal{A}$ =H3b	3.668e-02	2.736e-03	6.566e-03	3.746e-03	7.917e-03

	$\|\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}}\|$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A}$ =O1	1.271e-05	9.246e-05	1.535e-04	9.236e-05	1.527e-04
$\mathcal{A}$ =C2	1.430e-05	1.222e-04	1.985e-04	1.207e-04	1.960e-04
$\mathcal{A}$ =H2a	5.745e-04	2.854e-03	6.604e-03	2.841e-03	6.589e-03
$\mathcal{A}$ =H2b	7.845e-04	2.723e-03	5.802e-03	2.692e-03	5.793e-03
$\mathcal{A}$ =C3	2.415e-05	1.282e-04	2.364e-04	1.272e-04	2.347e-04
$\mathcal{A}$ =H3a	8.633e-04	3.291e-03	7.811e-03	3.225e-03	7.790e-03
$\mathcal{A}$ =H3b	1.232e-03	2.736e-03	6.566e-03	2.733e-03	6.528e-03

Table 13.26: ADP difference (in Å^2) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (epoxide).

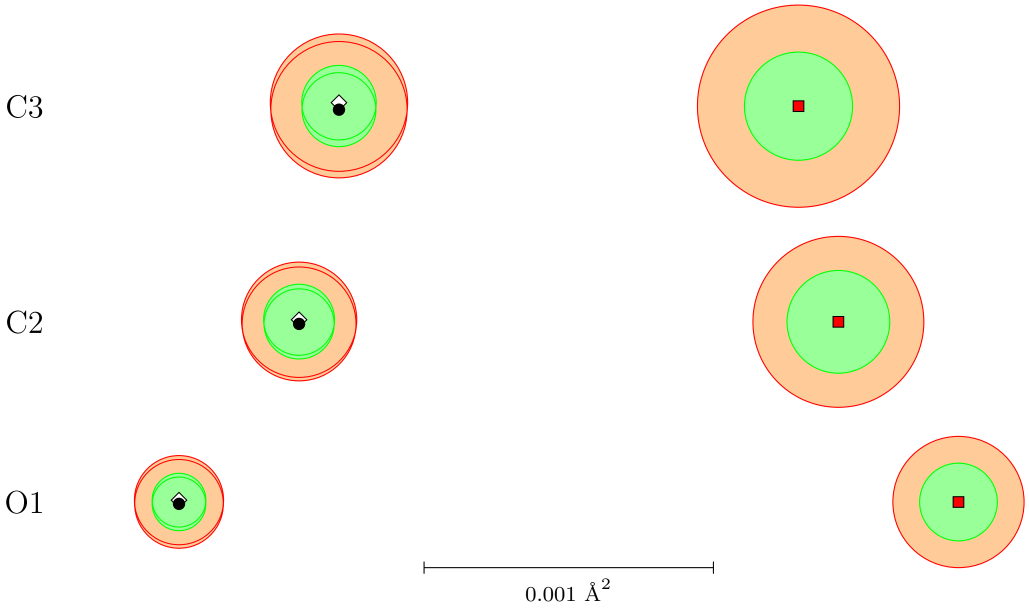


Figure 13.43: ADPs of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (epoxide)

Ammonia

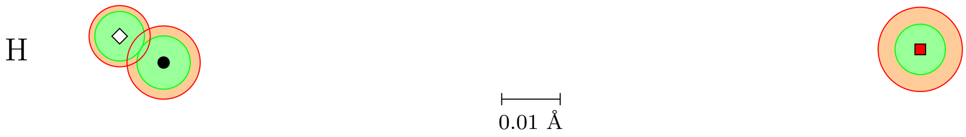


Figure 13.44: Positions of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red ■), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (ammonia)

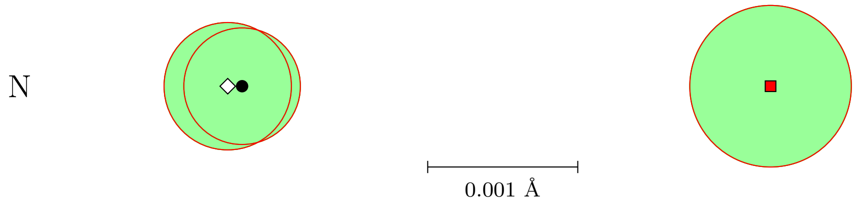


Figure 13.45: Positions of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red ■), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (ammonia)

	$\|\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}}\|$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A}$ =N	3.592e-03	4.214e-04	4.214e-04	5.344e-04	5.344e-04
$\mathcal{A}$ =H	1.343e-01	4.160e-03	5.131e-03	4.225e-03	7.067e-03
	$\|\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}}\|$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A}$ =N	9.513e-05	4.214e-04	4.214e-04	3.856e-04	3.856e-04
$\mathcal{A}$ =H	8.598e-03	4.160e-03	5.131e-03	4.468e-03	6.141e-03

Table 13.27: Positional difference (in Å) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (ammonia).

For the hydrogen atom of ammonia shown in Figure 13.44, we do find that the numerical and approximate positions lie just outside their respective uncertainty radii. However, they remain far further from the spherical minimum, implying that whilst the approximation is not completely fine in this case, it is far preferable to using only the spherical method. For the nitrogen atom they remain in agreement as desired, shown in Figure 13.45. These results are shown numerically in Table 13.27.

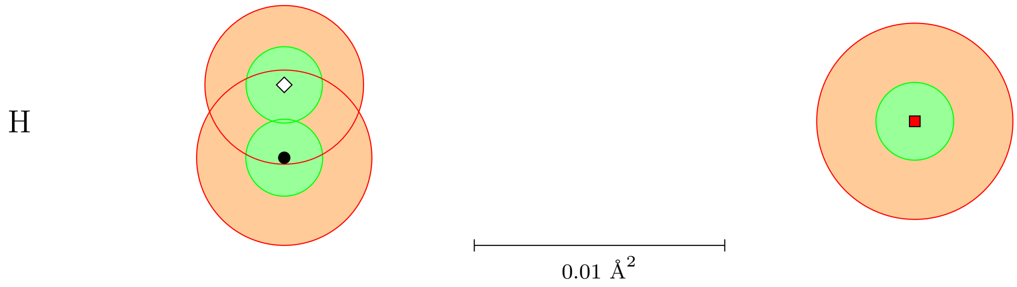


Figure 13.46: ADPs of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (ammonia)

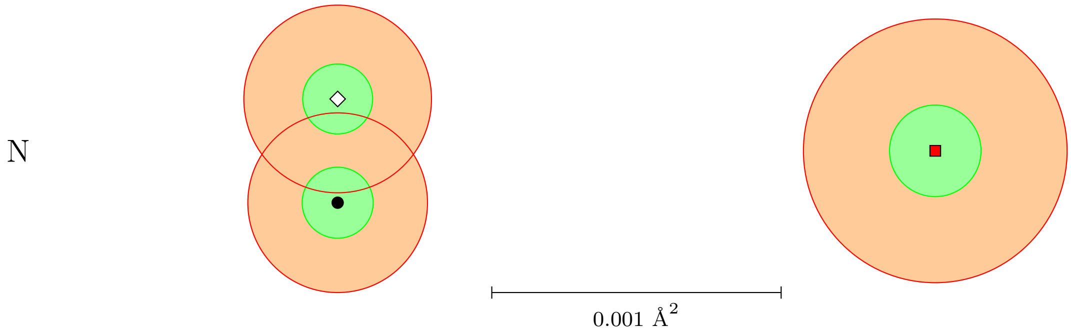


Figure 13.47: ADPs of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (ammonia)

Similarly, all of ammonia's ADPs in Figures 13.46 & 13.47 and Table 13.28 are outside of the inner uncertainty thresholds, though the hydrogen now lie within each other's outer uncertainty. Again, this implies that the approximation does provide a more significant error for ammonia than it has for the other examples.

	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A}$ =N	1.972e-03	1.205e-04	3.230e-04	1.576e-04	4.539e-04
$\mathcal{A}$ =H	2.536e-02	1.517e-03	3.153e-03	1.546e-03	3.899e-03
	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A}$ =N	3.572e-04	1.205e-04	3.230e-04	1.223e-04	3.093e-04
$\mathcal{A}$ =H	2.896e-03	1.517e-03	3.153e-03	1.529e-03	3.483e-03

Table 13.28: ADP difference (in Å^2) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (ammonia).

L-Alanine

L-Alanine returns to agreement between the atomic position in the numerical and approximate cases within their inner uncertainties in Figures 13.48& 13.49 and Table 13.30. For the case of the carbon atom C2, these also lie within the spherical case's inner uncertainty (and it within theirs), but largely this figure emphasises the points made previously.

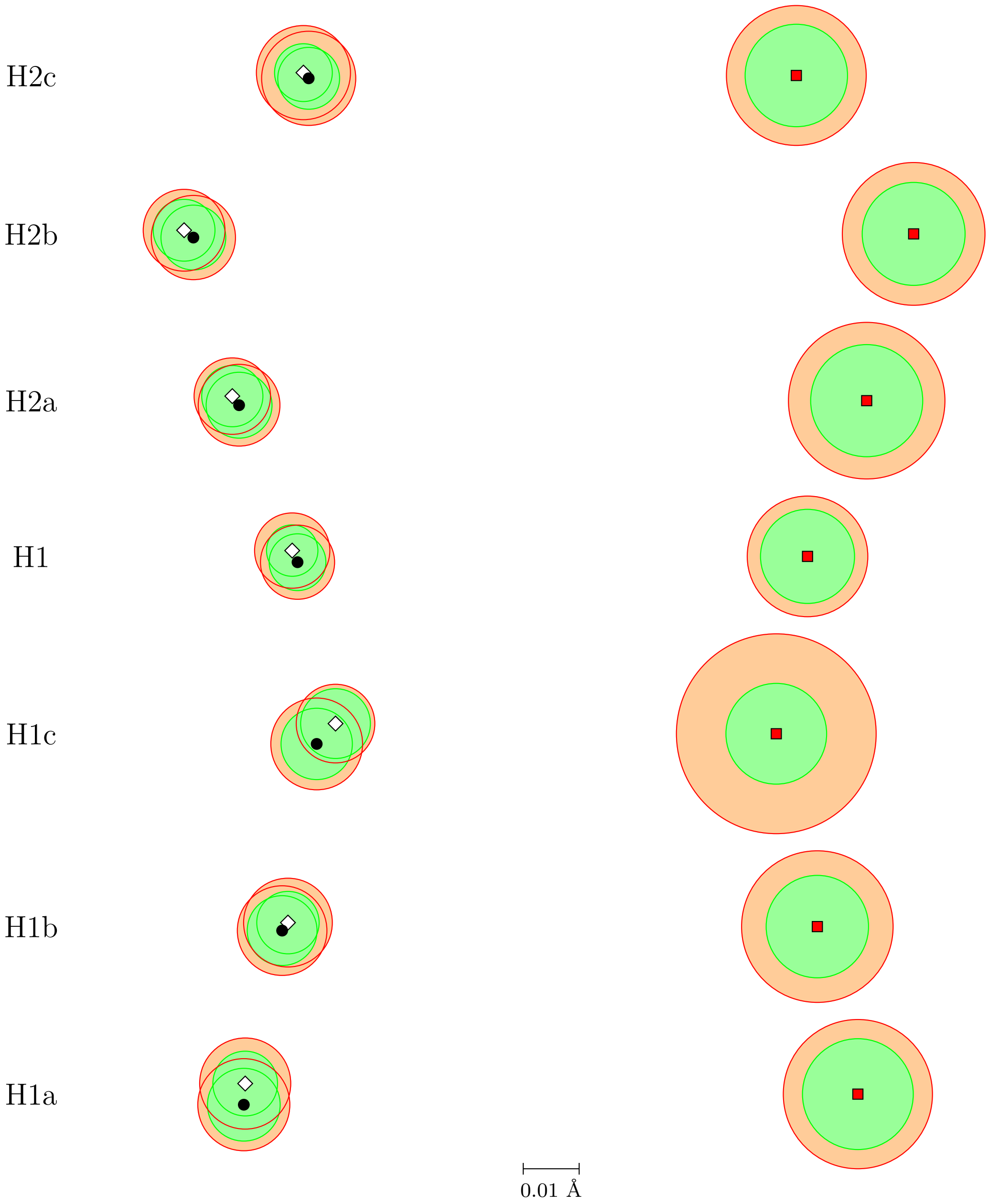


Figure 13.48: Positions of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white $\diamond$), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

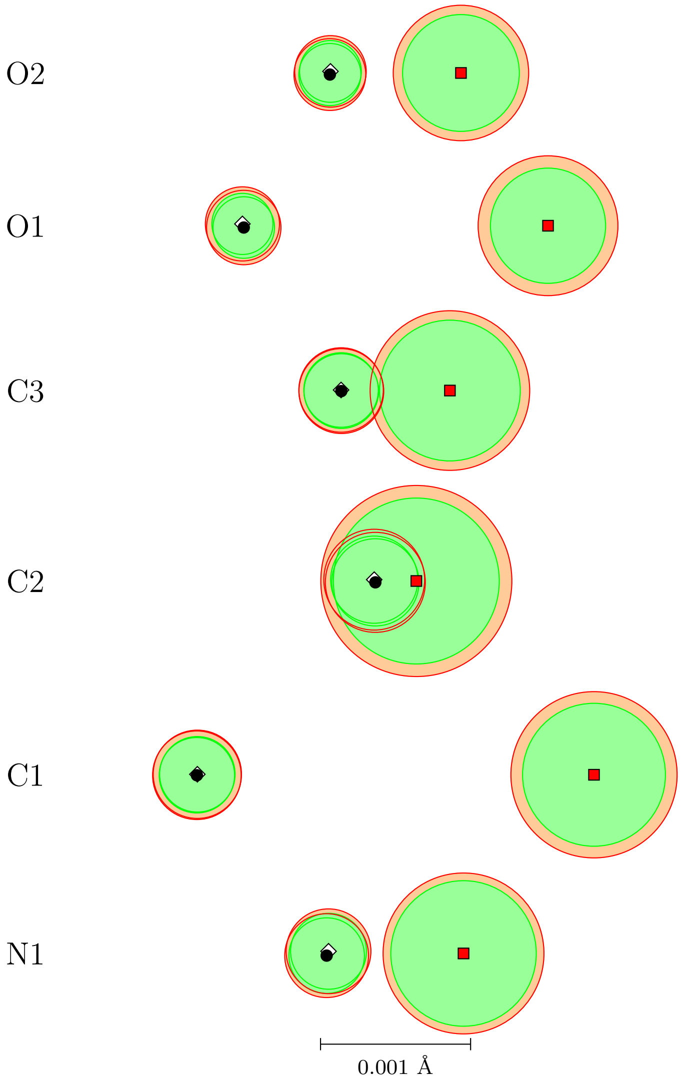


Figure 13.49: Positions of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A}=$N1	8.932e-04	2.510e-04	2.803e-04	4.813e-04	5.322e-04
$\mathcal{A}=$H1a	1.076e-01	5.681e-03	7.998e-03	9.731e-03	1.310e-02
$\mathcal{A}=$H1b	9.296e-02	5.480e-03	7.790e-03	8.991e-03	1.330e-02
$\mathcal{A}=$H1c	7.742e-02	6.125e-03	6.901e-03	8.844e-03	1.754e-02
$\mathcal{A}=$C1	2.624e-03	2.497e-04	2.920e-04	4.727e-04	5.498e-04
$\mathcal{A}=$H1	9.049e-02	4.513e-03	6.608e-03	8.260e-03	1.059e-02
$\mathcal{A}=$C2	2.788e-04	2.884e-04	3.335e-04	5.488e-04	6.320e-04
$\mathcal{A}=$H2a	1.114e-01	5.375e-03	6.711e-03	9.842e-03	1.374e-02
$\mathcal{A}=$H2b	1.281e-01	5.426e-03	7.179e-03	9.036e-03	1.253e-02
$\mathcal{A}=$H2c	8.655e-02	5.073e-03	8.260e-03	8.993e-03	1.228e-02
$\mathcal{A}=$C3	7.201e-04	2.465e-04	2.792e-04	4.656e-04	5.279e-04
$\mathcal{A}=$O1	2.022e-03	2.022e-04	2.452e-04	3.811e-04	4.624e-04
$\mathcal{A}=$O2	8.635e-04	2.043e-04	2.365e-04	3.863e-04	4.480e-04

	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A}=$N1	3.137e-05	2.510e-04	2.803e-04	2.499e-04	2.772e-04
$\mathcal{A}=$H1a	3.723e-03	5.681e-03	7.998e-03	6.385e-03	8.082e-03
$\mathcal{A}=$H1b	1.736e-03	5.480e-03	7.790e-03	6.129e-03	7.868e-03
$\mathcal{A}=$H1c	4.864e-03	6.125e-03	6.901e-03	6.264e-03	8.056e-03
$\mathcal{A}=$C1	7.121e-06	2.497e-04	2.920e-04	2.494e-04	2.930e-04
$\mathcal{A}=$H1	2.242e-03	4.513e-03	6.608e-03	4.981e-03	6.509e-03
$\mathcal{A}=$C2	1.933e-05	2.884e-04	3.335e-04	2.884e-04	3.308e-04
$\mathcal{A}=$H2a	1.990e-03	5.375e-03	6.711e-03	5.785e-03	7.175e-03
$\mathcal{A}=$H2b	2.074e-03	5.426e-03	7.179e-03	5.692e-03	7.394e-03
$\mathcal{A}=$H2c	1.359e-03	5.073e-03	8.260e-03	5.429e-03	8.263e-03
$\mathcal{A}=$C3	7.515e-06	2.465e-04	2.792e-04	2.470e-04	2.807e-04
$\mathcal{A}=$O1	2.556e-05	2.022e-04	2.452e-04	2.036e-04	2.461e-04
$\mathcal{A}=$O2	2.130e-05	2.043e-04	2.365e-04	2.055e-04	2.375e-04

Table 13.29: Positional difference (in Å) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (L-alanine).

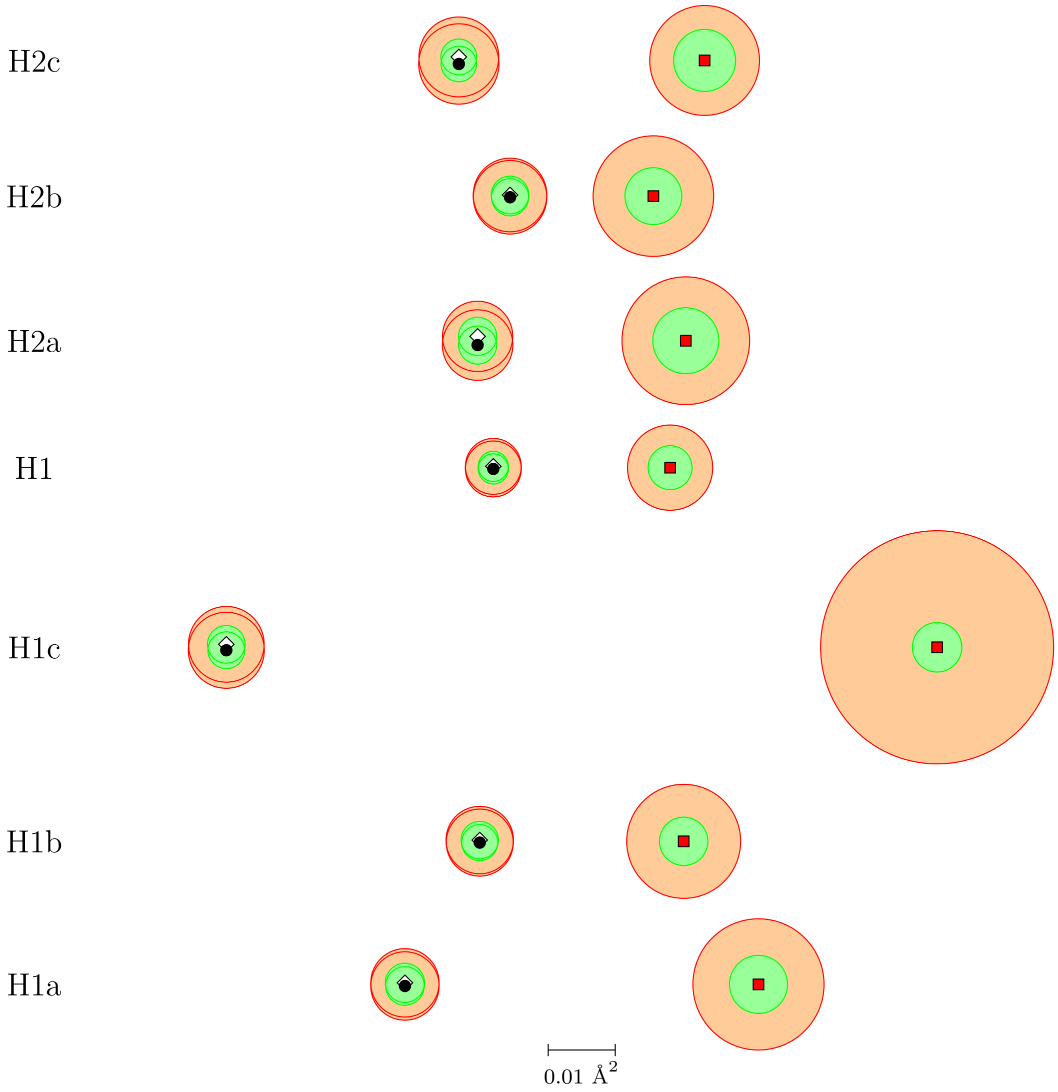


Figure 13.50: ADPs of the hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white ◇), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

Alanine's ADPs emphasise more soundly that approximate provides a suitable substitute to numerical refinement, and a significant improvement on spherical in Figures 13.50 & 13.51 and Table 13.30.

In conclusion, for all 3 molecules considered here the approximate and numerical refinement provide atomic locations which lie within each other's uncertainty radii in the vast majority of cases, and those which don't, still lie very close. These two minima are in far better agreement than either with spherical refinement, leading to

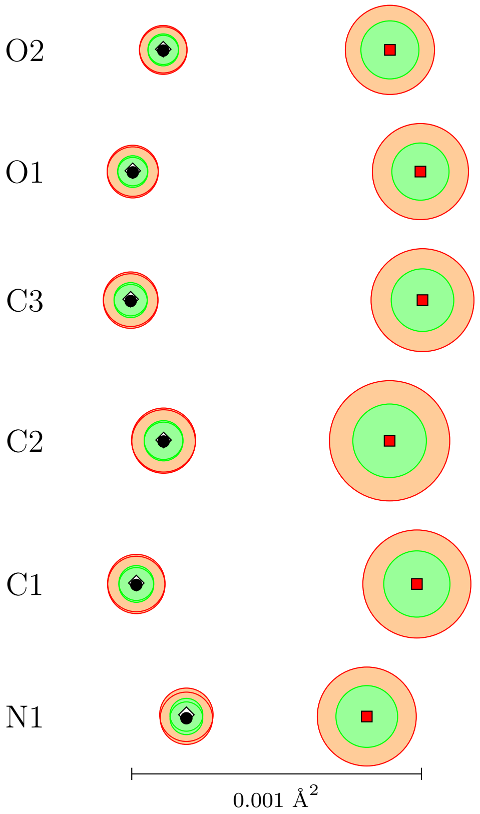


Figure 13.51: ADPs of the non-hydrogen atoms in $\mathbf{x}_{\text{num}}$ (white $\diamond$), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{spher}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{spher}})$	$R(\mathcal{A}_{\text{spher}})$
$\mathcal{A} =$N1	6.211e-04	5.645e-05	9.182e-05	1.061e-04	1.702e-04
$\mathcal{A} =$H1a	5.186e-02	2.876e-03	5.015e-03	4.265e-03	9.627e-03
$\mathcal{A} =$H1b	2.991e-02	2.741e-03	4.949e-03	3.551e-03	8.364e-03
$\mathcal{A} =$H1c	1.041e-01	2.765e-03	5.553e-03	3.631e-03	1.711e-02
$\mathcal{A} =$C1	9.630e-04	6.001e-05	9.864e-05	1.139e-04	1.858e-04
$\mathcal{A} =$H1	2.593e-02	2.216e-03	4.100e-03	3.224e-03	6.255e-03
$\mathcal{A} =$C2	7.760e-04	6.677e-05	1.101e-04	1.265e-04	2.068e-04
$\mathcal{A} =$H2a	2.999e-02	2.800e-03	5.158e-03	4.847e-03	9.370e-03
$\mathcal{A} =$H2b	2.103e-02	2.766e-03	5.404e-03	4.173e-03	8.846e-03
$\mathcal{A} =$H2c	3.582e-02	2.624e-03	5.858e-03	4.554e-03	8.060e-03
$\mathcal{A} =$C3	1.002e-03	5.717e-05	9.384e-05	1.076e-04	1.770e-04
$\mathcal{A} =$O1	9.886e-04	5.171e-05	8.777e-05	9.836e-05	1.653e-04
$\mathcal{A} =$O2	7.778e-04	5.263e-05	8.164e-05	9.986e-05	1.533e-04

	$\lVert\mathcal{A}_{\text{num}} - \mathcal{A}_{\text{appr}}\rVert$	$r(\mathcal{A}_{\text{num}})$	$R(\mathcal{A}_{\text{num}})$	$r(\mathcal{A}_{\text{appr}})$	$R(\mathcal{A}_{\text{appr}})$
$\mathcal{A} =$N1	1.197e-05	5.645e-05	9.182e-05	5.629e-05	8.993e-05
$\mathcal{A} =$H1a	4.618e-04	2.876e-03	5.015e-03	2.833e-03	5.017e-03
$\mathcal{A} =$H1b	3.718e-04	2.741e-03	4.949e-03	2.649e-03	4.917e-03
$\mathcal{A} =$H1c	8.752e-04	2.765e-03	5.553e-03	2.701e-03	5.576e-03
$\mathcal{A} =$C1	6.396e-06	6.001e-05	9.864e-05	5.981e-05	9.860e-05
$\mathcal{A} =$H1	3.791e-04	2.216e-03	4.100e-03	2.219e-03	4.101e-03
$\mathcal{A} =$C2	4.975e-06	6.677e-05	1.101e-04	6.665e-05	1.092e-04
$\mathcal{A} =$H2a	1.276e-03	2.800e-03	5.158e-03	2.797e-03	5.173e-03
$\mathcal{A} =$H2b	3.321e-04	2.766e-03	5.404e-03	2.728e-03	5.396e-03
$\mathcal{A} =$H2c	1.033e-03	2.624e-03	5.858e-03	2.608e-03	5.892e-03
$\mathcal{A} =$C3	6.221e-06	5.717e-05	9.384e-05	5.707e-05	9.384e-05
$\mathcal{A} =$O1	5.503e-06	5.171e-05	8.777e-05	5.164e-05	8.777e-05
$\mathcal{A} =$O2	4.666e-06	5.263e-05	8.164e-05	5.255e-05	8.167e-05

Table 13.30: ADP difference (in Å^2) between atoms in $\mathbf{x}_{\text{num}}$, $\mathbf{x}_{\text{appr}}$, and $\mathbf{x}_{\text{spher}}$ and their comparison with uncertainty bounds. Red indicates a bound smaller than the difference, whilst green represents a bound larger than the difference (L-alanine).

the conclusion that approximate nonspherical refinement is sufficiently precise for practical purposes. This is particularly notable for the positions of hydrogen atoms.

Neutron Data

For the sake of curiosity, we also employed the above ‘uncertainty ball’ plotting of the relative location of atoms with regards to neutron diffraction in Figures 13.52 to 13.55. The purple star now represents the atom in the neutron model, and we compare with the approximate and spherical models. We focus on L-Alanine, the neutron data for which is obtained from [34].

Neutron diffraction is currently regarded as a very good source of atomic information, the ‘gold standard’ to compare to.

For hydrogen both the position 13.52 and ADP diagram 13.54 indicate an improvement via approximate refinement, but an imperfect one. The non-hydrogen atoms 13.55 sadly show a far more scattered distribution.

The implication of this, if we accept neutron diffraction as near-perfect, is that there remains much within X-ray crystallography which is not yet accounted for, and emphasises the insignificance of the particular form-factor derivative approximation we have discussed in this Part.

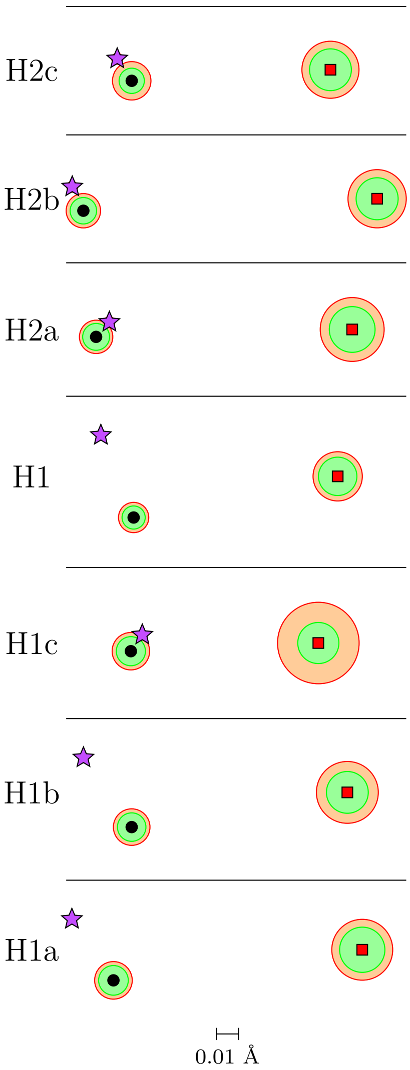


Figure 13.52: Positions of the hydrogen atoms in $\mathbf{x}_{\text{neutron}}$ (purple ★), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red ■), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

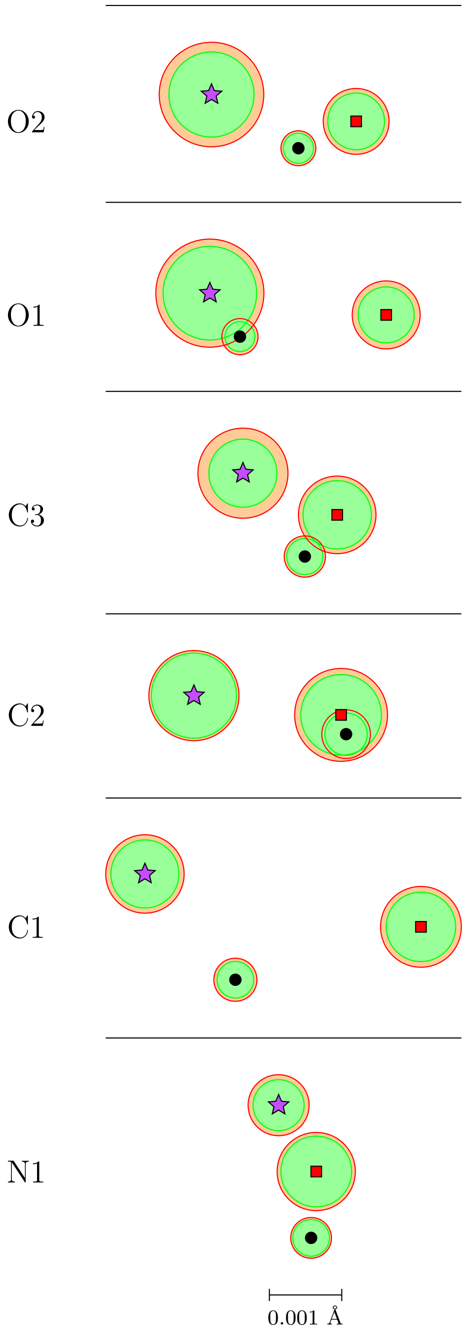


Figure 13.53: Positions of the non-hydrogen atoms in $\mathbf{x}_{\text{neutron}}$ (purple ★), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red ■), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

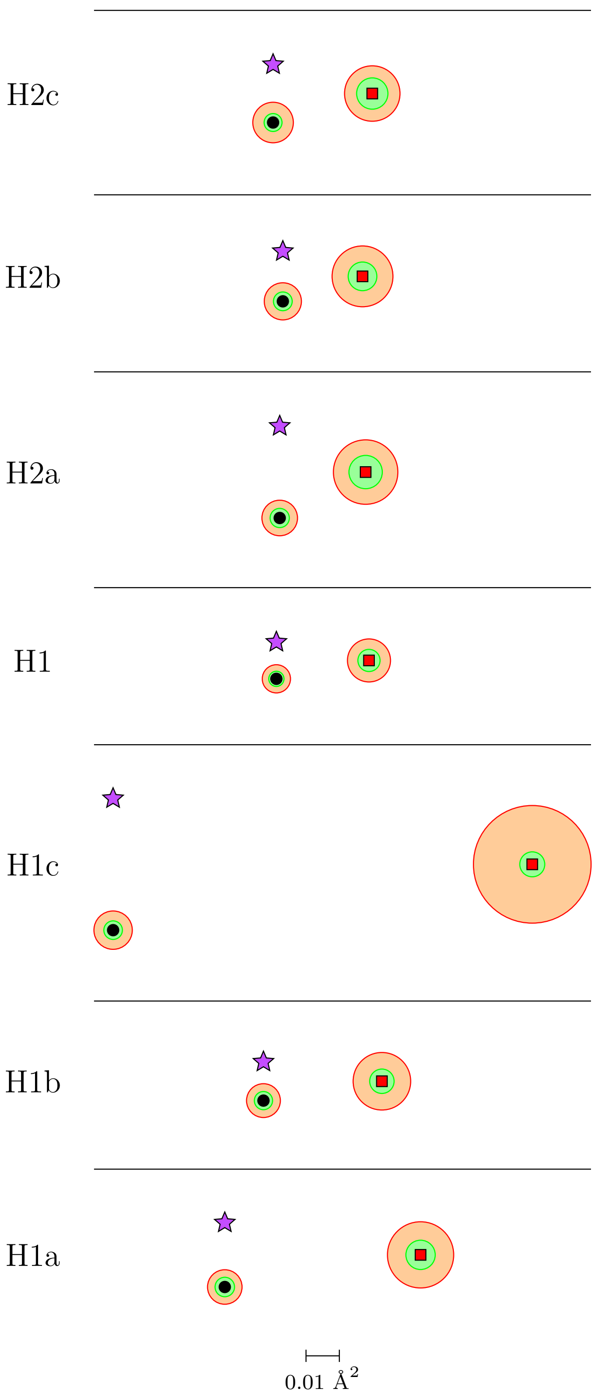


Figure 13.54: ADPs of the hydrogen atoms in $\mathbf{x}_{\text{neutron}}$ (purple ★), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red 🟥), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

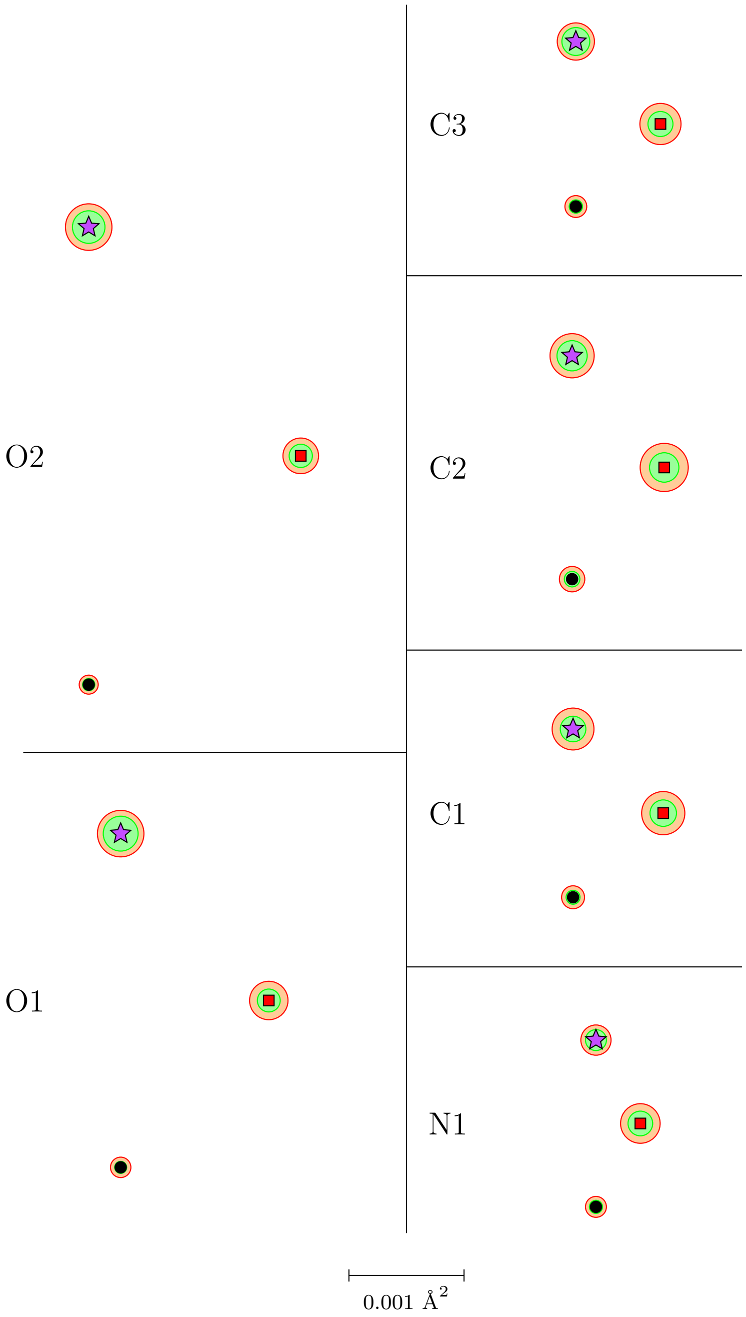


Figure 13.55: ADPs of the non-hydrogen atoms in $\mathbf{x}_{\text{neutron}}$ (purple ★), $\mathbf{x}_{\text{appr}}$ (black ●) and $\mathbf{x}_{\text{spher}}$ (red ■), and their uncertainty balls. Green balls represent inner uncertainty balls and red balls represent outer uncertainty balls. (L-alanine)

13.6.4 Cell Uncertainties

This subsection serves as a review of two contributors to uncertainties on derived values (such as interatomic distances) - the uncertainties on the cell parameters and those on the atomic parameters (which are in turn obtained via the least-squares minimisation process).

Assuming that both types of s.u.'s are uncorrelated, the total s.u. of a function f describing the derived parameter is then given by (see [4, (37)]):

$$\sigma(f) = \sqrt{\sigma^2_{\text{cell}}(f) + \sigma^2_{xyz}(f)}. \tag{13.6.12}$$

We will discuss here the explicit computation of both types of s.u.'s.

We primarily focus on uncertainties of atomic distances, the most common derived value. In this case, the s.u.'s derived from cell parameters are roughly proportional to the distance between the atoms. In contrast, the s.u.'s derived from the atomic parameters are independent of the distance and depend only on the direction between the atoms. This means that the s.u.'s derived from cell parameters become less and less significant for smaller and smaller distances. More precisely, we discuss s.u.'s of interatomic distances and s.u.'s of distances between the location of the same atom in two different models.

Interatomic Distance

The calculations in this section are based on [4, Section 6]. Let $\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}} \in \mathbb{R}^3$ be the relative positional coordinates of the atoms $\mathcal{A}$ and $\mathcal{C}$ of the model $\mathbf{x} \in \mathbb{R}^N$. Let $d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})$ be the Cartesian distance (in Å) between the atoms $\mathcal{A}$ and $\mathcal{C}$, given by

$$d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}) = \left((\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})\right)^{1/2} = \left(\sum_{i,j=1}^{3} (z_i^{\mathcal{A}} - z_i^{\mathcal{C}})(z_j^{\mathcal{A}} - z_j^{\mathcal{C}}) M_{ij}\right)^{1/2}, \tag{13.6.13}$$

where we view $\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}$ as column vectors and $M = (M_{ij})$ is the 3×3 metrical matrix from (2.2.1) with entries with units of Å^2.

Here, the cell parameters contribute to this distance through the metrical matrix M whilst the atomic parameters contribute through the vectors $\mathbf{z}^{\mathcal{A}}$ and $\mathbf{z}^{\mathcal{C}}$. In contrast to Subsection 13.6.2, the atomic parameter-dependent component of the uncertainty is dependent on the uncertainty of both atomic locations (rather than the second being a fixed point without uncertainty).

Thus, for the atomic parameters, we obtain from (13.6.13)

$$\frac{\partial d}{\partial z_i^{\mathcal{A}}}(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}) = -\frac{\partial d}{\partial z_i^{\mathcal{C}}}(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}) = \frac{1}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})}\sum_{j=1}^{3}(z_j^{\mathcal{A}} - z_j^{\mathcal{C}})M_{ij}.$$

In the following derivation, we alternatively use $(z_1^{\mathcal{A}}, z_2^{\mathcal{A}}, z_3^{\mathcal{A}}, z_1^{\mathcal{C}}, z_2^{\mathcal{C}}, z_3^{\mathcal{C}}) = (z_1, z_2, z_3, z_4, z_5, z_6) := \mathbf{z}$ The square of the s.u. of the distance between atoms $\mathcal{A}$ and $\mathcal{C}$ is then given by (see [53, (3.1.10.3)] or [4, (36)]:

$$\begin{aligned}
\sigma_{xyz}^2(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})) &= \sum_{i,j=1}^{6} \frac{\partial d}{\partial z_i}\frac{\partial d}{\partial z_j}\operatorname{cov}(z_i, z_j) \\
&= \sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{A}}}\frac{\partial d}{\partial z_j^{\mathcal{A}}}(V^{\mathcal{A}\mathcal{A}})_{ij} + \sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{A}}}\frac{\partial d}{\partial z_j^{\mathcal{C}}}(V^{\mathcal{A}\mathcal{C}})_{ij} \\
&\quad + \sum_{i,j=1}^{3} \frac{\partial d}{\partial z_j^{\mathcal{A}}}\frac{\partial d}{\partial z_j^{\mathcal{C}}}(V^{\mathcal{C}\mathcal{A}})_{ij} + \sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{C}}}\frac{\partial d}{\partial z_j^{\mathcal{C}}}(V^{\mathcal{C}\mathcal{C}})_{ij},
\end{aligned}$$

where, for example, $V^{\mathcal{A}\mathcal{C}}$ is the 3×3 submatrix of $\operatorname{Var}(\mathbf{x})$ obtained by choosing the positional coordinates of atom $\mathcal{A}$ as rows and of atom $\mathcal{C}$ as columns. We have dropped the arguments ($\mathbf{z}, \mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}$ as appropriate) of the derivatives for simplicity. Now we have

$$\begin{aligned}
\sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{A}}}\frac{\partial d}{\partial z_j^{\mathcal{A}}}(V^{\mathcal{A}\mathcal{A}})_{ij} &= \frac{1}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})^2}\sum_{i,j,l,m=1}^{3}(z_l^{\mathcal{A}} - z_l^{\mathcal{C}})M_{li}(V^{\mathcal{A}\mathcal{A}})_{ij}M_{jm}(z_m^{\mathcal{A}} - z_m^{\mathcal{C}}) \\
&= \frac{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M V^{\mathcal{A}\mathcal{A}} M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}.
\end{aligned}$$

Similarly, using $\frac{\partial d}{\partial z_i^{\mathcal{C}}} = -\frac{\partial d}{\partial z_i^{\mathcal{A}}}$, we obtain

$$\sum_{i,j=1}^{3} \frac{\partial d}{\partial z_i^{\mathcal{A}}}\frac{\partial d}{\partial z_j^{\mathcal{C}}}(V^{\mathcal{A}\mathcal{C}})_{ij} = -\frac{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M V^{\mathcal{A}\mathcal{C}} M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}.$$

Combining all four terms yields

$$\sigma^2_{xyz}(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})) = \frac{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M (V^{\mathcal{A}\mathcal{A}} - V^{\mathcal{A}\mathcal{C}} - V^{\mathcal{C}\mathcal{A}} + V^{\mathcal{C}\mathcal{C}}) M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}{(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})^\top M (\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})}. \quad (13.6.14)$$

For the cell parameters, we follow the arguments in [4, Section 6.2.1]. Let $C = \{a, b, c, \alpha, \beta, \gamma\}$ be the set of cell parameters and $\operatorname{cov}(i, j)$ with $i, j \in C$ be the corresponding covariances of their uncertainties. Then we have

$$\sigma^2_{\text{cell}}(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})) = \sum_{i,j \in C} \frac{\partial d}{\partial i} \frac{\partial d}{\partial j} \operatorname{cov}(i, j) \quad (13.6.15)$$

with

$$\frac{\partial d}{\partial i} = \sum_{l,m=1}^{3} \frac{\partial d}{\partial M_{lm}} \frac{\partial M_{lm}}{\partial i}$$

by the chain rule. We obtain from (13.6.13)

$$\frac{\partial d}{\partial M_{lm}}(\mathbf{z}^{A}, \mathbf{z}^{B}) = \frac{1}{2d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})} (z_l^{\mathcal{A}} - z_l^{\mathcal{C}})(z_m^{\mathcal{A}} - z_m^{\mathcal{C}}).$$

For the derivation of the values $\frac{\partial M_{lm}}{\partial i}$ recall that

$$M = \begin{bmatrix} \mathbf{a} \cdot \mathbf{a} & \mathbf{a} \cdot \mathbf{b} & \mathbf{a} \cdot \mathbf{c} \\ \mathbf{b} \cdot \mathbf{a} & \mathbf{b} \cdot \mathbf{b} & \mathbf{b} \cdot \mathbf{c} \\ \mathbf{c} \cdot \mathbf{a} & \mathbf{c} \cdot \mathbf{b} & \mathbf{c} \cdot \mathbf{c} \end{bmatrix} = \begin{bmatrix} a^2 & ab\cos\gamma & ac\cos\beta \\ ab\cos\gamma & b^2 & bc\cos\alpha \\ ac\cos\beta & bc\cos\alpha & c^2 \end{bmatrix}$$

and thus, for example,

$$\frac{\partial M}{\partial a} = \begin{bmatrix} 2a & b\cos\gamma & c\cos\beta \\ b\cos\gamma & 0 & 0 \\ c\cos\beta & 0 & 0 \end{bmatrix},$$

$$\frac{\partial M}{\partial \alpha} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -bc\sin\alpha \\ 0 & -bc\sin\alpha & 0 \end{bmatrix},$$

with similarly derived formulae applying for b, c, β, γ.

Thus the partial derivatives $\frac{\partial d}{\partial i}$ are given by

$$\frac{\partial d}{\partial a} = \frac{z_1^{\mathcal{A}} - z_1^{\mathcal{C}}}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})} \left(a(z_1^{\mathcal{A}} - z_1^{\mathcal{C}}) + b\cos\gamma\,(z_2^{\mathcal{A}} - z_2^{\mathcal{C}}) + c\cos\beta\,(z_3^{\mathcal{A}} - z_3^{\mathcal{C}}) \right),$$

$$
\begin{aligned}
\frac{\partial d}{\partial b} &= \frac{z_2^{\mathcal{A}} - z_2^{\mathcal{C}}}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})} \Big(b(z_2^{\mathcal{A}} - z_2^{\mathcal{C}}) + a\cos\gamma\,(z_1^{\mathcal{A}} - z_1^{\mathcal{C}}) + c\cos\alpha\,(z_3^{\mathcal{A}} - z_3^{\mathcal{C}}) \Big)\,, \\
\frac{\partial d}{\partial c} &= \frac{z_3^{\mathcal{A}} - z_3^{\mathcal{C}}}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})} \Big(c(z_3^{\mathcal{A}} - z_3^{\mathcal{C}}) + a\cos\beta\,(z_1^{\mathcal{A}} - z_1^{\mathcal{C}}) + b\cos\alpha\,(z_2^{\mathcal{A}} - z_2^{\mathcal{C}}) \Big)\,, \\
\frac{\partial d}{\partial \alpha} &= -\frac{bc\sin\alpha\,(z_2^{\mathcal{A}} - z_2^{\mathcal{C}})(z_3^{\mathcal{A}} - z_3^{\mathcal{C}})}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})}, \\
\frac{\partial d}{\partial \beta} &= -\frac{ac\sin\beta\,(z_1^{\mathcal{A}} - z_1^{\mathcal{C}})(z_3^{\mathcal{A}} - z_3^{\mathcal{C}})}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})}, \\
\frac{\partial d}{\partial \gamma} &= -\frac{ab\sin\gamma\,(z_1^{\mathcal{A}} - z_1^{\mathcal{C}})(z_2^{\mathcal{A}} - z_2^{\mathcal{C}})}{d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})}.
\end{aligned}
$$

Note the following property of the partial derivatives $\frac{\partial d}{\partial i}$, $i \in C$: A replacement of $\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}}$ by $\lambda(\mathbf{z}^{\mathcal{A}} - \mathbf{z}^{\mathcal{C}})$ leads to a multiplication of these partial derivatives by the factor λ. Consequently, these partial derivatives are roughly proportional to the distance $d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}})$. We conclude from formula 13.6.15 that this distance proportionality holds also for $\sigma_{\text{cell}}(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}))$. This is in significant contrast to the uncertainty $\sigma_{xyz}(d(\mathbf{z}^{\mathcal{A}}, \mathbf{z}^{\mathcal{C}}))$, given in (13.6.14), which does not change at all under such a λ-rescaling. As a consequence, the influence of the s.u. coming from the cell parameters becomes less and less significant for the total s.u. for smaller and smaller interatomic atomic distances.

Example 13.6.1 (X-H Distances)**.** In this example, we present details about the uncertainties of the X-H distances of all five optimal models in Tables 13.31 to 13.33. We use the formulas (13.6.14) and (13.6.15) for the s.u.'s derived from atomic parameters and from cell parameters, respectively. Both types of uncertainties are assumed to be uncorrelated and the total uncertainties are therefore given by (13.6.12). The s.u.'s of the cell parameters are provided in Appendix A.

Since we have no information about covariances $\mathrm{cov}(i, j)$ of cell parameters $i \neq j$, we ignore them by assuming that they are zero - that is, we assume they are uncorrelated. Moreover, note that the s.u. of any angles need to be translated from degrees into radians. In some cases, we know due to the crystal structure that an angle is precisely 90° and thus has an s.u. of zero.

Recall the fact that $\sigma_{\text{cell}}(d(\mathbf{z}^A, \mathbf{z}^B))$ is roughly proportional to $d(\mathbf{z}^A, \mathbf{z}^B)$. In the case of a X-H distance, $d(\mathbf{z}^A, \mathbf{z}^B)$ is about 1 Å . This distance is small enough that the derived s.u.'s of cell parameters have essentially no influence on the first two significant digits of the total s.u.'s of the X-H distances.

Epoxide	$\mathbf{x}_{\text{num}}$	$\mathbf{x}_{\text{hybr}}$	$\mathbf{x}_{\text{appr}}$	$\mathbf{x}_{\text{mix}}$	$\mathbf{x}_{\text{spher}}$
	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}
	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}
	σ	σ	σ	σ	σ
C2-H2a	0.000681	0.000681	0.000680	0.000681	0.000610
	0.005860	0.005848	0.006649	0.006610	0.008438
	0.005900	0.005887	0.006684	0.006645	0.008460
C2-H2b	0.000127	0.000127	0.000127	0.000127	0.000115
	0.005403	0.005388	0.006202	0.006214	0.008485
	0.005405	0.005389	0.006204	0.006215	0.008486
C3-H3a	0.000751	0.000751	0.000746	0.000750	0.000689
	0.006655	0.006632	0.007782	0.007879	0.011053
	0.006698	0.006674	0.007817	0.007915	0.011074
C3-H3b	0.000271	0.000271	0.000273	0.000273	0.000237
	0.006544	0.006536	0.007471	0.007548	0.009692
	0.006549	0.006541	0.007476	0.007553	0.009695

Table 13.31: Detailed s.u.'s of X-H distances for epoxide

Example 13.6.2 (Uncertainties of Corresponding Atoms)**.** In this example we provide information about the uncertainties of corresponding atoms in the spherical and approximate refinement optima $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$. More precisely, we choose the Cartesian position of an atom in one of the optima as reference point and present the s.u. of the distance of the corresponding atom in the other optimum. This information is given in Tables 13.34 to 13.36, where the s.u.'s σ_{xyz} derived from atomic parameters and σ_{cell} derived from cell parameters are provided separately.

Since each σ_{xyz} is at least 65 times larger than the corresponding σ_{cell} in the case of

Ammonia	$\mathbf{x}_{\text{num}}$	$\mathbf{x}_{\text{hybr}}$	$\mathbf{x}_{\text{appr}}$	$\mathbf{x}_{\text{mix}}$	$\mathbf{x}_{\text{spher}}$
	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}
	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}
	σ	σ	σ	σ	σ
N-H	0.000108	0.000108	0.000108	0.000108	0.000098
	0.000513	0.000514	0.000539	0.000544	0.000641
	0.000524	0.000525	0.000550	0.000555	0.000648

Table 13.32: Detailled s.u.'s of X-H distances for ammonia

epoxide, at least 22 times larger in the case of ammonia and at least 281 times larger in the case of L-alanine, the total s.u., given by $\sigma = (\sigma_{xyz}^2 + \sigma_{\text{cell}}^2)^{1/2}$, is at most 1.001 larger than σ_{xyz}. This increase is so minimal that we can safely use the figures given in σ_{xyz} as a description of the total uncertainties.

We also see that the σ_{xyz} of corresponding atoms in the tables are significantly smaller than the actual distance between their positions in $\mathbf{x}_{\text{spher}}$ and $\mathbf{x}_{\text{appr}}$ (with the exception of one heavier atom of L-alanine which are already well placed in the spherical optimum – this is highlighted in green in Table 13.36). This confirms again that the optimum of approximate refinement differs substantially from the optimum of spherical refinement.

L-Alanine	$\mathbf{x}_{\text{num}}$	$\mathbf{x}_{\text{hybr}}$	$\mathbf{x}_{\text{appr}}$	$\mathbf{x}_{\text{mix}}$	$\mathbf{x}_{\text{spher}}$
	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}	σ_{cell}
	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}	σ_{xyz}
	σ	σ	σ	σ	σ
$N1 - H1a$	0.000124	0.000124	0.000124	0.000124	0.000117
	0.005701	0.005696	0.006411	0.005798	0.012714
	0.005702	0.005697	0.006412	0.005800	0.012715
$N1 - H1b$	0.000115	0.000115	0.000114	0.000114	0.000107
	0.005519	0.005509	0.006393	0.005819	0.012044
	0.005520	0.005510	0.006394	0.005820	0.012044
$N1 - H1c$	0.000141	0.000141	0.000140	0.000140	0.000141
	0.006309	0.006289	0.007882	0.007011	0.017197
	0.006310	0.006291	0.007883	0.007012	0.017197
$C1 - H1$	0.000143	0.000143	0.000142	0.000143	0.000130
	0.004497	0.004494	0.005012	0.005223	0.008885
	0.004499	0.004496	0.005014	0.005225	0.008886
$C2 - H2a$	0.000142	0.000142	0.000142	0.000142	0.000131
	0.005667	0.005657	0.006657	0.006904	0.013122
	0.005669	0.005659	0.006658	0.006905	0.013122
$C2 - H2b$	0.000123	0.000123	0.000123	0.000123	0.000107
	0.005694	0.005688	0.006530	0.006510	0.011482
	0.005696	0.005689	0.006531	0.006511	0.011482
$C2 - H2c$	0.000143	0.000143	0.000143	0.000143	0.000132
	0.005124	0.005119	0.005928	0.006030	0.010437
	0.005126	0.005121	0.005930	0.006031	0.010438

Table 13.33: Detailled s.u.'s of X-H distances for L-alanine

Epoxide	$\lVert\mathcal{A}_{\text{appr}} - \mathcal{A}_{\text{spher}}\rVert$	$\sigma_{xyz}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$ $\sigma_{xyz}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$	$\sigma_{cell}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$ $\sigma_{cell}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$
$\mathcal{A}$ =O1	5.117e-03	2.499e-04	2.701e-06
		3.652e-04	2.701e-06
$\mathcal{A}$ =C2	1.860e-03	3.980e-04	1.227e-06
		5.905e-04	1.227e-06
$\mathcal{A}$ =H2a	1.012e-01	6.786e-03	6.975e-05
		8.490e-03	6.975e-05
$\mathcal{A}$ =H2b	1.125e-01	6.757e-03	1.956e-05
		8.700e-03	1.956e-05
$\mathcal{A}$ =C3	3.355e-03	4.339e-04	6.037e-07
		6.360e-04	6.037e-07
$\mathcal{A}$ =H3a	1.305e-01	7.675e-03	6.143e-05
		1.082e-02	6.143e-05
$\mathcal{A}$ =H3b	1.232e-01	7.303e-03	3.475e-05
		9.525e-03	3.475e-05

Table 13.34: Distances between corresponding atoms of spherical and approximate optima for epoxide and their derived s.u.'s w.r.t. atomic parameters and cell parameters (in Å)

Ammonia	$\lVert\mathcal{A}_{\text{appr}} - \mathcal{A}_{\text{spher}}\rVert$	$\sigma_{xyz}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$ $\sigma_{xyz}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$	$\sigma_{cell}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$ $\sigma_{cell}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$
$\mathcal{A}$ =N	3.497e-03	1.262e-04	3.148e-07
		1.713e-04	3.148e-07
$\mathcal{A}$ =H	1.270e-01	4.495e-04	1.189e-05
		4.943e-04	1.189e-05

Table 13.35: Distances between corresponding atoms of spherical and approximate optima for epoxide and their derived s.u.'s w.r.t. atomic parameters and cell parameters (in Å)

L-Alanine	$\|\mathcal{A}_{\text{appr}} - \mathcal{A}_{\text{spher}}\|$	$\sigma_{xyz}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$	$\sigma_{cell}(d_{\mathcal{A}_{\text{appr}}}(\mathcal{A}_{\text{spher}}))$
		$\sigma_{xyz}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$	$\sigma_{cell}(d_{\mathcal{A}_{\text{spher}}}(\mathcal{A}_{\text{appr}}))$
$\mathcal{A}$ =N1	9.057e-04	2.564e-04	1.270e-07
		4.973e-04	1.270e-07
$\mathcal{A}$ =H1a	1.078e-01	7.265e-03	9.651e-06
		1.186e-02	9.651e-06
$\mathcal{A}$ =H1b	9.398e-02	7.032e-03	8.385e-06
		1.306e-02	8.385e-06
$\mathcal{A}$ =H1c	8.072e-02	7.102e-03	1.274e-05
		1.369e-02	1.274e-05
$\mathcal{A}$ =C1	2.627e-03	2.891e-04	3.526e-07
		5.424e-04	3.526e-07
$\mathcal{A}$ =H1	8.956e-02	4.990e-03	1.212e-05
		8.873e-03	1.212e-05
$\mathcal{A}$ =C2	2.712e-04	3.088e-04	2.549e-08
		5.901e-04	2.549e-08
$\mathcal{A}$ =H2a	1.102e-01	6.383e-03	1.213e-05
		1.241e-02	1.213e-05
$\mathcal{A}$ =H2b	1.265e-01	6.242e-03	1.499e-05
		1.081e-02	1.499e-05
$\mathcal{A}$ =H2c	8.563e-02	6.004e-03	1.069e-05
		1.061e-02	1.069e-05
$\mathcal{A}$ =C3	7.183e-04	2.704e-04	7.738e-08
		5.072e-04	7.738e-08
$\mathcal{A}$ =O1	2.013e-03	2.158e-04	3.041e-07
		4.043e-04	3.041e-07
$\mathcal{A}$ =O2	8.674e-04	2.133e-04	9.409e-08
		4.021e-04	9.409e-08

Table 13.36: Distances between corresponding atoms of spherical and approximate optima for L-alanine and their derived s.u.'s w.r.t. atomic parameters and cell parameters (in Å); Uncertainties larger than the distances between the atoms are highlighted in green.

13.7 Summary

The implementation of nonspherical structure refinement in *olex2.refine*-NoSpherA2 is based on an approximation of the partial derivatives of nonspherical structure factors (see formula (12.2.5) in the Introduction). We refer to this implementation as *approximate (nonspherical) refinement*. The motivation for this part was to investigate the validity of this approximation. In our investigations we focused on three specific molecules: epoxide, ammonia and L-alanine.

To be able to compare the nonspherical refinement results based on this approximation, we needed to obtain the partial derivatives of the nonspherical structure factors as accurately as possible. Our method to do this is numerical differentiation, which depends on a suitable choice of a step size $\delta > 0$. We refer to the so computed partial derivatives as the numerical partial derivatives and the corresponding refinement as the *numerical (nonspherical) refinement*. The investigations in Section 13.3 show that the best step sizes for numerical differentiation is 10^{-3}Å, with little difference in result from $\delta = 10^{-2}$Å to 10^{-4}Å. This confirms the reliability and stability of the method. Moreover, refinement using numerical differentiation with step size 10^{-3}Å provides optimal models which are – after sufficiently many numerical refinement steps – highly independent of the initial model solutions (see (13.5.1) and (13.5.3) for the Cartesian distance between refined models obtained from two different initial models). Therefore, we use the fixed step size $\delta = 10^{-3}$Å for all further computations involving numerical differentiation.

The optimal model obtained in *olex2.refine* via classical spherical refinement is denoted by $\mathbf{x}_{\text{spher}}$ and the nonspherical optimum obtained via *olex2.refine*-NoSphera2 is denoted by $\mathbf{x}_{\text{appr}}$. Note that the partial derivatives of structure factors to obtain $\mathbf{x}_{\text{appr}}$ are based on the above mentioned approximation. Starting from $\mathbf{x}_{\text{appr}}$, we obtain our numerical nonspherical optimum $\mathbf{x}_{\text{num}}$ as the best model through 20 numerical refinement steps, where each numerical refinement step uses numerical differentiation to compute the partial derivatives of the structure factors as accurately as possible. Therefore, we consider $\mathbf{x}_{\text{num}}$ to be a very accurate representation of the best nonspherical optimum with respect to the wR_2 factor. However, the process to obtain $\mathbf{x}_{\text{num}}$ comes at a price: the computation of $\mathbf{x}_{\text{num}}$ is very time-consuming, since each partial derivative of the structure factor via numerical differentiation requires two additional quantum mechanical calculations.

In Section 13.5 of this part, we compare the optimal models $\mathbf{x}_{\text{spher}}$, $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{num}}$. The X-H distances of $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{num}}$ are significantly larger than the corresponding X-H distances of $\mathbf{x}_{\text{spher}}$. Moreover, the X-H distances of $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{num}}$ agree within the margins of their standard uncertainties (see Table 13.16) while the X-H distances of $\mathbf{x}_{\text{spher}}$ are far outside the standard uncertainties of $\mathbf{x}_{\text{appr}}$ and $\mathbf{x}_{\text{num}}$. In conclusion, the current nonspherical refinement implementation in *olex2.refine*-NoSpherA2 provides significantly better results with regards to X-H distances.

However, disregarding any uncertainties, the precise coordinates of the heavier atoms of $\mathbf{x}_{\text{num}}$ are in better agreement with the corresponding heavier atoms of $\mathbf{x}_{\text{appr}}$ than the coordinates of the hydrogen atoms (roughly by a factor of at least 10). This can be clearly seen in Table 13.23 and it is the motivation to introduce another nonspherical refinement process with particular focus on the hydrogen atoms: *hybrid (nonspherical) refinement*. Hybrid refinement can be viewed as a compromise between the currently implemented approximate nonspherical refinement and the time-consuming numerical refinement. In hybrid refinement, only partial derivatives corresponding to the hydrogen atoms are computed as accurately as possible via numerical differentiation and the partial derivatives corresponding to all other atoms are computed using the approximation given in (12.2.5). This leads to a reduction of computing time: in our examples of epoxide and L-alanine, hybrid refinement is about twice as fast as numerical refinement[5]. Moreover, it turns out that the corresponding optimum $\mathbf{x}_{\text{hybr}}$ – obtained as the best result from $\mathbf{x}_{\text{appr}}$ within 20 hybrid refinement steps – is in high agreement with $\mathbf{x}_{\text{num}}$ also in all of the hydrogen

[5] ammonia take a little longer, as this only removes one of the 4 parameters

atoms.

In every refinement process, the shift vector is computed as the product of a shift matrix S and a residual vector $\tilde{r}$. While the shift matrix involves the partial derivatives of the structure factor, the residual requires only the structure factor itself. Surprisingly, a refinement process using the spherical shift matrix and the nonspherical residual provides results which are much closer to the ones of numerical refinement than of spherical refinement. We refer to this refinement as the *mixed (spherical-nonspherical) refinement.* Its optimum, obtained as the best result within 20 mixed refinement steps starting from $\mathbf{x}_{\mathrm{spher}}$, is denoted by $\mathbf{x}_{\mathrm{mix}}$. Tables 13.17 and 13.19 show that the component of $\mathbf{x}_{\mathrm{mix}}$ containing all positional coordinates is almost as close to $\mathbf{x}_{\mathrm{num}}$ as the corresponding positional component of $\mathbf{x}_{\mathrm{appr}}$, and it is more than 10 times closer than the corresponding component of $\mathbf{x}_{\mathrm{spher}}$. These findings suggest that mixed refinement might perform almost as good as approximate refinement also for other molecules. However, since both refinement processes require a quantum mechanical calculation in each refinement step to obtain the nonspherical residual, there is no gain in computing time by choosing mixed refinement instead of approximate refinement.

The ultimate question is whether in certain cases it is worth using numerical refinement instead of the currently implemented approximate refinement despite the significantly higher time effort. For this decision it is useful to investigate the standard uncertainties of their optimal models. For that purpose, we introduced the notions of atomic uncertainty radii in Subsection 13.6.2. Our results in Section 13.6 show that the atomic positions of the model $\mathbf{x}_{\mathrm{appr}}$ lie well within the uncertainty domains of the atomic positions of $\mathbf{x}_{\mathrm{num}}$. This suggests that the results obtained via approximate refinement are sufficiently accurate for all usual practical purposes.

Chapter 14

Investigations into Extinction

14.1 Motivation

During my non-spherical tests, we encountered a model of glycine ($C_2H_5NO_2$ [12], structure presented in Figure 14.1) which would not refine to a stable minimum if the weighting scheme parameters[1] were set to 0. This is a highly concerning trait, as the weighting scheme is intended to have only a small effect on the refinement process. However, we found that when extinction was used instead, the extinction parameter refined to a high value (0.3 to 0.4) and led to convergence of the model.

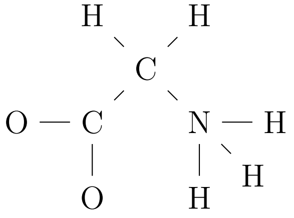


Figure 14.1: The chemical structure of glycine

[1] Typically, a crystallographic weighting scheme takes into account more than simply the recorded uncertainties on the intensity data. One such weight adjustment is that used by SHELXL [52], and adjusts the weights from $1/\sigma^2$, including in the denominator extra factors of $(aP)^2 + bP + d + e \sin\theta$ (where P is $f|F_o|^2 + (1-f)|F_c|^2$), and replacing the numerator with $e^{c\sin^2\theta/\lambda^2}$, 1, or $1 - e^{c\sin^2\theta/\lambda^2}$ depending on if c is positive, zero or negative respectively, where a, b, c, d, e, f are weight parameters. Often in the first course of using a weighting scheme many of these factors are set to 0, usually b, c, d, e (that is, only a and f are refined).

When I attempted numerical non-spherical refinement, I found the following issue regarding the design matrices: using the same method as in Subsection 13.2.1, I compared analytically derived approximate (and spherical) design matrices with numerically derived approximate (and spherical) design matrices, which should be identical or near-identical. I found that some values had very dramatic differences.

Through investigation, I found that the implementation of extinction in *olex2.refine* (as described in [4]) did not cascade the correction fully (equation (62)) into the design matrices, only added the derivatives corresponding to the extinction parameter whilst keeping the other derivatives unchanged. In other words, the theory has not been fully implemented. As the model structure factors are altered with extinction, it follows that the model derivatives also will be changed. The necessary changes are described in this chapter.

14.2 Background

Olex2's current *Extinction Correction* provides a consistent adjustment to all structure factor $F_c(\mathbf{h})$ values to emulate the effects of extinction within a crystal. This includes a new refinable parameter x_e when extinction correction is switched on. The adjustment is (see (62) in [4])

$$F_c'(\mathbf{h}) = F_c(\mathbf{h}) \left(1 + 0.001 x_e \frac{|F_c(\mathbf{h})|^2 \lambda^3}{\sin(2\theta(\mathbf{h}))}\right)^{-1/4}, \tag{14.2.1}$$

where x_e is the extinction parameter, λ is the wavelength of the incoming X-ray beam (in Amstrongs), and $\theta(\mathbf{h})$ is the angle of reflection. Additionally, note that $Y_c(\mathbf{h}) = |F_c(\mathbf{h})|^2 = F_c(\mathbf{h}) F_c^*(\mathbf{h})$.

For simplicity, we introduce some additional equations and terminology:

$$X_e(\mathbf{h}) := 0.001 x_e \frac{\lambda^3}{\sin(2\theta(\mathbf{h}))} \tag{14.2.2}$$

collates many terms in the adjustment factor into one for ease of readability, whilst

$$Y_c'(\mathbf{h}) = Y_c(\mathbf{h})\left(1 + X_e(\mathbf{h})Y_c(\mathbf{h})\right)^{-1/2} \tag{14.2.3}$$

is the equivalent adjustment to (14.2.1) made instead to the intensities Y_c, incorporating $X_e(\mathbf{h})$ as defined in (14.2.2). (Note that Y_c' does not denote a derivative, but just a modification of Y_c.)

We can then further collate our terms and define the ‘multiplier’

$$M(\mathbf{h}) := \left(1 + X_e(\mathbf{h})Y_c(\mathbf{h})\right)^{-1/2}. \tag{14.2.4}$$

14.3 Derivative Derivation

The design matrix $D(\mathbf{x})$ is a matrix with rows corresponding to Miller indices $\mathbf{h}$ and columns corresponding to parameters x_n (including the extinction parameter, x_e). That is,

$$(D(\mathbf{x}))_{kn} = \frac{\partial Y_c}{\partial x_n}(\mathbf{x}, \mathbf{h}_k).$$

With the adjustment as above altering Y_c to Y_c', we should also adjust our design matrix to $\frac{\partial Y_c'}{\partial x_n}(\mathbf{x}, \mathbf{h}_k)$.

Firstly, the design matrix must be expanded with the derivatives of Y_c' with respect to the extinction parameter x_e. This is simple to calculate, and

$$\frac{\partial Y_c'}{\partial x_e}(\mathbf{h}) = -0.001\frac{\lambda^3}{2\sin(2\theta(\mathbf{h}))}Y_c(\mathbf{h})M(\mathbf{h})^3. \tag{14.3.1}$$

Next, we consider only $x_n \neq x_e$, typically the positional and ADP parameters of each atom. This adjustment was not implemented in *olex2.refine* prior to my investigation.

Lemma 14.3.1. Let $Y_c'(\mathbf{h})$ be given as in (14.2.3). Then $\partial Y_c'(\mathbf{h})/\partial x_n$ for $x_n \neq x_e$ is given by:

$$\frac{\partial Y_c'(\mathbf{h})}{\partial x_n} = \frac{\partial Y_c(\mathbf{h})}{\partial x_n}\frac{1}{2}\left(M(\mathbf{h})^3 + M(\mathbf{h})\right). \tag{14.3.2}$$

Proof.

$$\begin{aligned}
\frac{\partial Y_c'(\mathbf{h})}{\partial x_n} &= \frac{\partial}{\partial x_n}\left(Y_c(\mathbf{h})M(\mathbf{h})\right)\\
&= Y_c(\mathbf{h})\frac{\partial}{\partial x_n}\left(M(\mathbf{h})\right) + \frac{\partial Y_c(\mathbf{h})}{\partial x_n}M(\mathbf{h})\\
&= -\frac{Y_c(\mathbf{h})}{2}M(\mathbf{h})^3\frac{\partial}{\partial x_n}\left(1 + X_e(\mathbf{h})Y_c(\mathbf{h})\right) + \frac{\partial Y_c(\mathbf{h})}{\partial x_n}M(\mathbf{h})\\
&= -\frac{Y_c(\mathbf{h})}{2}M(\mathbf{h})^3X_e(\mathbf{h})\frac{\partial Y_c(\mathbf{h})}{\partial x_n} + \frac{\partial Y_c(\mathbf{h})}{\partial x_n}M(\mathbf{h})\\
&= \frac{\partial Y_c(\mathbf{h})}{\partial x_n}M(\mathbf{h})\left(1 - M(\mathbf{h})^2Y_c(\mathbf{h})X_e(\mathbf{h})/2\right)\\
&= \frac{\partial Y_c(\mathbf{h})}{\partial x_n}M(\mathbf{h})\left(\frac{2 + X_e(\mathbf{h})Y_c(\mathbf{h})}{2(1 + X_e(\mathbf{h})Y_c(\mathbf{h}))}\right)\\
&= \frac{\partial Y_c(\mathbf{h})}{\partial x_n}\frac{1}{2}\left(M(\mathbf{h})^3 + M(\mathbf{h})\right).
\end{aligned}$$

□

An interesting result of this is that the multiplier effect to the design matrix given in (14.3.2) appears *more* significant than its effect to the calculated intensities as given in (14.2.3) and (14.2.4), since

$$0 \leq \frac{1}{2}\left(M(\mathbf{h})^3 + M(\mathbf{h})\right) < M(\mathbf{h}) \leq 1.$$

However, it is difficult to predict its impact on the whole refinement process, as the calculation of the inverse normal matrix compresses many of these details, and many $M(\mathbf{h})$ are not far from 1, as seen in the next section.

Let us summarise how to include these extinction adjustments in the refinement process: to move from $D(\mathbf{x})$ without extinction to $D'(\mathbf{x})$ including extinction, each row must be multiplied by its corresponding multiplier $\frac{1}{2}\left(M(\mathbf{h})^3 + M(\mathbf{h})\right)$, where $M(\mathbf{h}) = \left(1 + 0.001x_e\frac{Y_c(\mathbf{h})\lambda^3}{\sin(2\theta(\mathbf{h}))}\right)^{-1/2}$. Additionally, an additional column dependent on the extinction parameter x_e as described in Equation (14.3.1) must be added.

14.4 Example

To evaluate the potential severity of this missing factor, I calculated information on the multiplier at a single pre-refined point.

For these investigations, I utilised a refined solution of glycine as given in [12], with an extinction parameter x_e of 0.396108. Using this data, the minimum multiplier $M(\mathbf{h})$ was 0.4874, which occurred at the **h** index (0 -2 0). The maximal (that is, least adjusting) multiplier was barely below 1 (0.99999999878), occurring at the **h** index (-8 0 2). Whilst half of entries had multipliers very close to 1 (the median value being 0.9994), 1% had a value 0.915 or less (in this case, this is 16 indices out of 1672), with 7 rows having a multiplier of 0.8 or less. Whether or not this would have a significant impact under further refinement remains to be seen, but it does indicate that the F_c for these specific **h** are likely to have been under-adjusted with each refinement step, when not including this multiplier.

I also tried to start a numerical nonspherical refinement process from this data, and encountered a problem that the inverse of the normal matrix could not be calculated (nor would Cholesky decomposition work); the matrix was near-singular with an eigenvalue of 10^{-6} where others were 10^5. Thus I could not investigate where the 'corrected' refinement may lie.

Appendix A

Tables of refinement minima

The final models for each minima are presented in this appendix.

A.1 Ammonia

	x	y	z	σ(x)	σ(y)	σ(z)
N	-0.21023684	-0.21023684	-0.21023684	0.00006014	0.00006014	0.00006014
H	-0.34812626	-0.25292423	-0.12709205	0.00115619	0.00134492	0.00095370

Table A.1: Positions and their uncertainties in $\mathbf{x}_{\text{spher}}$ (ammonia)

	U11	U22	U33	U12	U13	U23
N	0.03724112	0.03724112	0.03724112	-0.00051598	-0.00051598	-0.00051598
H	0.05217024	0.05626107	0.05311731	-0.00519528	0.00523647	-0.00442689

Table A.2: ADPs in $\mathbf{x}_{\text{spher}}$ (ammonia)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N	0.00025718	0.00025718	0.00025718	0.00010405	0.00010405	0.00010405
H	0.00274417	0.00282723	0.00268566	0.00252421	0.00262085	0.00328818

Table A.3: ADP Uncertainties in $\mathbf{x}_{\text{spher}}$ (ammonia)

	x	y	z	σ(x)	σ(y)	σ(z)
N	-0.21063037	-0.21063037	-0.21063037	0.00004339	0.00004339	0.00004339
H	-0.36459300	-0.26748918	-0.11572486	0.00099515	0.00118253	0.00092224

Table A.4: Positions and their uncertainties in $\mathbf{x}_{\text{appr}}$ (ammonia)

	U11	U22	U33	U12	U13	U23
N	0.03633643	0.03633643	0.03633643	-0.00137075	-0.00137075	-0.00137075
H	0.06684603	0.06992138	0.06471864	-0.00941038	0.00829451	-0.01173399

Table A.5: ADPs in $\mathbf{x}_{\text{appr}}$ (ammonia)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N	0.00017600	0.00017600	0.00017600	0.00007683	0.00007683	0.00007683
H	0.00262302	0.00281359	0.00253973	0.00240753	0.00234663	0.00282272

Table A.6: ADP Uncertainties in $\mathbf{x}_{\text{appr}}$ (ammonia)

	x	y	z	σ(x)	σ(y)	σ(z)
N	-0.21064108	-0.21064108	-0.21064108	0.00004742	0.00004742	0.00004742
H	-0.36553288	-0.26887599	-0.11576606	0.00088698	0.00099388	0.00085872

Table A.7: Positions and their uncertainties in $\mathbf{x}_{\text{num}}$ (ammonia)

	U11	U22	U33	U12	U13	U23
N	0.03653857	0.03653857	0.03653857	-0.00141164	-0.00141164	-0.00141164
H	0.06567149	0.07057512	0.06583168	-0.00813844	0.01015144	-0.01225835

Table A.8: ADPs in $\mathbf{x}_{\text{num}}$ (ammonia)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N	0.00018335	0.00018335	0.00018335	0.00007749	0.00007749	0.00007749
H	0.00263048	0.00250930	0.00242584	0.00234009	0.00221070	0.00264031

Table A.9: ADP Uncertainties in $\mathbf{x}_{\text{num}}$ (ammonia)

A.2 Epoxide

	x	y	z	σ(x)	σ(y)	σ(z)
O1	0.11682278	0.83081319	0.12407298	0.00009630	0.00004849	0.00006842
C2	0.14837961	0.93866025	0.29620979	0.00014065	0.00007154	0.00009783
H2a	0.26882585	0.89255774	0.42309445	0.00217402	0.00104309	0.00125185
H2b	0.17257700	1.05105150	0.26632456	0.00178043	0.00101427	0.00127526
C3	-0.13433715	0.86488511	0.21842066	0.00015494	0.00008203	0.00010677
H3a	-0.29343656	0.92326174	0.13303997	0.00240558	0.00106792	0.00155870
H3b	-0.19416073	0.77581343	0.29441725	0.00221466	0.00118997	0.00167968

Table A.10: Positions and their uncertainties in $\mathbf{x}_{\text{spher}}$ (epoxide)

	U11	U22	U33	U12	U13	U23
O1	0.03517212	0.02728050	0.03058483	-0.00520461	0.01112360	0.00258685
C2	0.03404153	0.02557660	0.02758400	-0.00265580	0.00598523	0.00070565
H2a	0.05634256	0.04908961	0.01918077	-0.00836162	-0.01604999	0.00632139
H2b	0.04119990	0.03406674	0.03068180	-0.00335647	0.00701081	-0.01166467
C3	0.03141294	0.03457965	0.03505740	-0.00018481	0.01288547	-0.00280110
H3a	0.03823368	0.08009990	0.04256266	0.00309522	0.00972083	-0.00022733
H3b	0.06008372	0.04038094	0.06253344	-0.00869169	0.02500175	-0.01785658

Table A.11: ADPs in $\mathbf{x}_{\text{spher}}$ (epoxide)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
O1	0.00022982	0.00018214	0.00021036	0.00016005	0.00016980	0.00016897
C2	0.00029279	0.00024122	0.00025566	0.00019796	0.00021759	0.00021077
H2a	0.00630831	0.00599915	0.00423080	0.00486364	0.00421960	0.00571211
H2b	0.00611800	0.00569123	0.00495894	0.00410443	0.00439119	0.00398904
C3	0.00029955	0.00030082	0.00031624	0.00023095	0.00025055	0.00023292
H3a	0.00571220	0.00833418	0.00563862	0.00619847	0.00482044	0.00652402
H3b	0.00734428	0.00560223	0.00667028	0.00627241	0.00640804	0.00573051

Table A.12: ADP Uncertainties in $\mathbf{x}_{\text{spher}}$ (epoxide)

	x	y	z	σ(x)	σ(y)	σ(z)
O1	0.11629706	0.83105629	0.12461968	0.00006669	0.00003312	0.00004704
C2	0.14860645	0.93867319	0.29600402	0.00009426	0.00004844	0.00006599
H2a	0.28180832	0.88998569	0.43677840	0.00184317	0.00084975	0.00103281
H2b	0.17378376	1.06191834	0.25652393	0.00153409	0.00071760	0.00116802
C3	-0.13420169	0.86455269	0.21817113	0.00010271	0.00005529	0.00007059
H3a	-0.30870926	0.93325202	0.12019171	0.00181904	0.00086307	0.00117706
H3b	-0.20345561	0.76473066	0.30368497	0.00177855	0.00103186	0.00130594

Table A.13: Positions and their uncertainties in $\mathbf{x}_{\text{appr}}$ (epoxide)

	U11	U22	U33	U12	U13	U23
O1	0.03539775	0.02561570	0.02938774	-0.00354788	0.01060380	0.00304139
C2	0.03241572	0.02498453	0.02735948	-0.00208911	0.00568577	0.00038726
H2a	0.08133475	0.07166194	0.03146493	0.00882681	-0.01522649	0.02242921
H2b	0.06381217	0.02571240	0.07051370	-0.00284714	0.01845210	-0.00834418
C3	0.03064605	0.03378709	0.03434746	-0.00000406	0.01196525	-0.00343559
H3a	0.04871461	0.10417533	0.05120784	0.01349921	0.00126842	0.02801827
H3b	0.07197904	0.06268409	0.07685965	0.00440319	0.02775164	-0.03385757

Table A.14: ADPs in $\mathbf{x}_{\text{appr}}$ (epoxide)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
O1	0.00015575	0.00012207	0.00014131	0.00010942	0.00011543	0.00011633
C2	0.00019448	0.00016386	0.00017561	0.00013391	0.00014787	0.00014232
H2a	0.00582552	0.00545698	0.00358524	0.00449760	0.00382175	0.00542634
H2b	0.00550721	0.00366361	0.00518875	0.00350873	0.00430389	0.00313875
C3	0.00020145	0.00020449	0.00021365	0.00015697	0.00016893	0.00015608
H3a	0.00454199	0.00742064	0.00451299	0.00481251	0.00364336	0.00527837
H3b	0.00582954	0.00475939	0.00538370	0.00543243	0.00521835	0.00492548

Table A.15: ADP Uncertainties in $\mathbf{x}_{\text{appr}}$ (epoxide)

	x	y	z	σ(x)	σ(y)	σ(z)
O1	0.11630200	0.83105456	0.12462240	0.00006670	0.00003283	0.00004683
C2	0.14860388	0.93867370	0.29600818	0.00009482	0.00004933	0.00006666
H2a	0.28188303	0.89018655	0.43679749	0.00175602	0.00083973	0.00096101
H2b	0.17415410	1.06198945	0.25654421	0.00152052	0.00064049	0.00112451
C3	-0.13422380	0.86455610	0.21816096	0.00010404	0.00005554	0.00007162
H3a	-0.30965599	0.93325279	0.11966245	0.00171279	0.00082679	0.00111782
H3b	-0.20327746	0.76447829	0.30349427	0.00177844	0.00097593	0.00123570

Table A.16: Positions and their uncertainties in $\mathbf{x}_{\text{num}}$ (epoxide)

	U11	U22	U33	U12	U13	U23
O1	0.03540553	0.02561929	0.02939355	-0.00355484	0.01060544	0.00304290
C2	0.03241766	0.02499714	0.02735873	-0.00209336	0.00568212	0.00038483
H2a	0.08161116	0.07198549	0.03125303	0.00869537	-0.01506367	0.02267296
H2b	0.06381938	0.02571043	0.06976910	-0.00281568	0.01813303	-0.00845670
C3	0.03065347	0.03378731	0.03434924	-0.00001576	0.01196974	-0.00345708
H3a	0.04850887	0.10404794	0.05112049	0.01298037	0.00163072	0.02750554
H3b	0.07119344	0.06306339	0.07627523	0.00435705	0.02676631	-0.03375260

Table A.17: ADPs in $\mathbf{x}_{\mathrm{num}}$ (epoxide)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
O1	0.00015637	0.00012241	0.00014181	0.00010984	0.00011567	0.00011629
C2	0.00019661	0.00016611	0.00017776	0.00013683	0.00014951	0.00014380
H2a	0.00589093	0.00550590	0.00359529	0.00453185	0.00379739	0.00546308
H2b	0.00546325	0.00364392	0.00523232	0.00348838	0.00427868	0.00313333
C3	0.00020251	0.00020568	0.00021486	0.00015898	0.00017052	0.00015932
H3a	0.00454027	0.00745359	0.00457971	0.00484397	0.00364728	0.00524844
H3b	0.00596198	0.00482120	0.00545628	0.00534942	0.00518044	0.00485359

Table A.18: ADP Uncertainties in $\mathbf{x}_{\mathrm{num}}$ (epoxide)

A.3 L-Alanine

	x	y	z	σ(x)	σ(y)	σ(z)
N1	0.64715199	0.13748440	0.18297222	0.00008333	0.00004001	0.00009123
H1a	0.69577155	0.06593927	0.18639576	0.00166174	0.00105007	0.00207293
H1b	0.76922956	0.18496216	0.20244209	0.00169047	0.00092367	0.00217118
H1c	0.57059761	0.14785070	0.01423038	0.00228660	0.00079584	0.00268883
C1	0.46618367	0.16103190	0.35483921	0.00008463	0.00003859	0.00009485
H1	0.42487718	0.24008423	0.34287846	0.00145129	0.00073197	0.00179664
C2	0.25985016	0.09070526	0.30330006	0.00009476	0.00004601	0.00010894
H2a	0.20273129	0.10811255	0.14640631	0.00170590	0.00081632	0.00236560
H2b	0.14478156	0.10752585	0.41890998	0.00164680	0.00076871	0.00215600
H2c	0.29956521	0.01107545	0.30956638	0.00167021	0.00082082	0.00211673
C3	0.55405621	0.14076416	0.59994977	0.00008767	0.00003798	0.00009104
O1	0.72707533	0.08372921	0.62429091	0.00007038	0.00003174	0.00007978
O2	0.44092666	0.18410196	0.76119868	0.00007124	0.00003179	0.00007707

Table A.19: Positions and their uncertainties in $\mathbf{x}_{\text{spher}}$ (L-alanine)

	U11	U22	U33	U12	U13	U23
N1	0.00577389	0.00662454	0.00446784	-0.00018451	0.00063649	0.00018035
H1a	0.00595765	0.06825635	0.00201168	0.01589340	0.00717879	-0.00920244
H1b	0.00929117	0.03928127	0.01510088	0.00998003	0.01194376	0.02679480
H1c	0.06714575	0.00019180	0.10205770	-0.02801725	0.07920933	-0.03568289
C1	0.00551139	0.00538394	0.00390322	-0.00008583	0.00002376	0.00045229
H1	0.00429880	0.01510641	0.00173544	-0.00380405	0.01115591	0.00957543
C2	0.00614238	0.00938561	0.00660525	-0.00042254	-0.00064816	-0.00130107
H2a	0.00767116	0.02449755	0.02692839	0.00146072	-0.00374702	-0.00652462
H2b	0.00246485	0.01561439	0.02767608	0.00396895	-0.00303481	-0.00455455
H2c	0.02186048	0.01840313	0.01413230	-0.00171879	0.00246347	-0.01076307
C3	0.00547546	0.00431003	0.00408438	0.00007177	-0.00039891	-0.00032423
O1	0.00674007	0.00691055	0.00684896	0.00066567	-0.00057545	0.00217667
O2	0.00796247	0.00783480	0.00417166	-0.00069984	0.00099902	0.00171969

Table A.20: ADPs in $\mathbf{x}_{\text{spher}}$ (L-alanine)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N1	0.00013661	0.00013525	0.00015990	0.00012349	0.00012269	0.00012067
H1a	0.00578714	0.00917431	0.00679987	0.00668690	0.00510409	0.00625569
H1b	0.00542117	0.00688834	0.00751670	0.00619230	0.00569097	0.00560942
H1c	0.00935418	0.00544910	0.01512111	0.00730861	0.01057508	0.00645620
C1	0.00015323	0.00013529	0.00017331	0.00011890	0.00013025	0.00012223
H1	0.00484772	0.00500877	0.00565425	0.00442049	0.00456052	0.00441289
C2	0.00016169	0.00016472	0.00019390	0.00015027	0.00014509	0.00014619
H2a	0.00592584	0.00685744	0.00887026	0.00609390	0.00596207	0.00514189
H2b	0.00500791	0.00575037	0.00855371	0.00553423	0.00593660	0.00472603
H2c	0.00736634	0.00552057	0.00745904	0.00530167	0.00615965	0.00548926
C3	0.00014167	0.00013028	0.00016655	0.00011973	0.00013186	0.00012872
O1	0.00012834	0.00012154	0.00015767	0.00011532	0.00011713	0.00010903
O2	0.00012914	0.00012152	0.00014305	0.00010691	0.00011883	0.00011449

Table A.21: ADP Uncertainties in $\mathbf{x}_{\text{spher}}$ (L-alanine)

	x	y	z	σ(x)	σ(y)	σ(z)
N1	0.64729053	0.13751227	0.18294275	0.00004322	0.00002054	0.00004758
H1a	0.70445311	0.06058376	0.19819088	0.00112703	0.00052660	0.00138648
H1b	0.77847125	0.18951189	0.21144384	0.00104111	0.00051495	0.00135294
H1c	0.58376013	0.14917634	0.01643847	0.00119424	0.00052738	0.00129350
C1	0.46629967	0.16108988	0.35441909	0.00004487	0.00002035	0.00005055
H1	0.42255275	0.24730043	0.34322177	0.00093090	0.00040879	0.00111761
C2	0.25986975	0.09068834	0.30332258	0.00005001	0.00002409	0.00005699
H2a	0.19209852	0.11035097	0.13153931	0.00100174	0.00050468	0.00122179
H2b	0.12782867	0.10557622	0.43150634	0.00100750	0.00048330	0.00127323
H2c	0.30412208	0.00447976	0.30818369	0.00100187	0.00046814	0.00142565
C3	0.55403078	0.14079781	0.59985168	0.00004638	0.00002015	0.00004844
O1	0.72675475	0.08378246	0.62426987	0.00003759	0.00001697	0.00004244
O2	0.44090952	0.18404169	0.76112224	0.00003754	0.00001689	0.00004089

Table A.22: Positions and their uncertainties in $\mathbf{x}_{\mathrm{appr}}$ (L-alanine)

	U11	U22	U33	U12	U13	U23
N1	0.00555336	0.00609744	0.00424070	-0.00015653	0.00067527	0.00011983
H1a	0.02522476	0.02735933	0.01783128	0.00267880	0.01280222	0.00489290
H1b	0.01340261	0.02796687	0.02170759	-0.00007891	0.00691040	0.00265475
H1c	0.03650293	0.02164220	0.02576888	-0.01170160	0.02671652	-0.00913759
C1	0.00518915	0.00472578	0.00351801	0.00009128	-0.00044247	0.00043513
H1	0.02281192	0.01236980	0.01804812	-0.00116477	0.00412339	0.00810289
C2	0.00581523	0.00886036	0.00613851	-0.00046798	-0.00059124	-0.00131908
H2a	0.01958430	0.04810270	0.01567394	0.01052002	-0.01089222	-0.00988225
H2b	0.01071599	0.03065489	0.03273262	0.00392778	0.00788701	-0.00250176
H2c	0.03190898	0.01552745	0.04808608	-0.00463110	0.00064944	-0.00383372
C3	0.00486618	0.00425965	0.00355403	-0.00003353	-0.00019280	0.00022600
O1	0.00668075	0.00705206	0.00592130	0.00052406	-0.00063649	0.00244631
O2	0.00754769	0.00761713	0.00392052	-0.00056017	0.00059908	0.00210478

Table A.23: ADPs in $\mathbf{x}_{\text{appr}}$ (L-alanine)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N1	0.00007240	0.00007072	0.00008459	0.00006466	0.00006471	0.00006338
H1a	0.00410776	0.00363241	0.00471554	0.00346826	0.00377478	0.00342399
H1b	0.00322159	0.00360407	0.00469599	0.00357162	0.00356594	0.00321163
H1c	0.00440654	0.00384202	0.00498110	0.00367668	0.00400057	0.00386312
C1	0.00008098	0.00007114	0.00009207	0.00006324	0.00006883	0.00006460
H1	0.00344488	0.00271591	0.00387724	0.00275681	0.00333956	0.00281442
C2	0.00008586	0.00008667	0.00010250	0.00007961	0.00007646	0.00007716
H2a	0.00392880	0.00469574	0.00440916	0.00362803	0.00328712	0.00347132
H2b	0.00315943	0.00387597	0.00514589	0.00359278	0.00367685	0.00313768
H2c	0.00484259	0.00295715	0.00567382	0.00349171	0.00446842	0.00331167
C3	0.00007465	0.00006965	0.00008829	0.00006400	0.00006994	0.00006785
O1	0.00006799	0.00006531	0.00008338	0.00006132	0.00006187	0.00005818
O2	0.00006847	0.00006418	0.00007608	0.00005688	0.00006300	0.00006073

Table A.24: ADP Uncertainties in $\mathbf{x}_{\text{appr}}$ (L-alanine)

	x	y	z	σ(x)	σ(y)	σ(z)
N1	0.64728890	0.13751052	0.18293918	0.00004348	0.00002075	0.00004797
H1a	0.70487739	0.06055461	0.19772123	0.00110091	0.00047813	0.00137278
H1b	0.77853649	0.18945748	0.21117536	0.00097618	0.00048447	0.00134424
H1c	0.58317835	0.14933638	0.01595286	0.00114259	0.00051320	0.00108228
C1	0.46629879	0.16109020	0.35441960	0.00004475	0.00002038	0.00005038
H1	0.42273200	0.24738602	0.34351040	0.00092953	0.00037176	0.00113593
C2	0.25987203	0.09068765	0.30332069	0.00005005	0.00002421	0.00005746
H2a	0.19217152	0.11046128	0.13129869	0.00098673	0.00049667	0.00107981
H2b	0.12766701	0.10544931	0.43167567	0.00095739	0.00047716	0.00120445
H2c	0.30407268	0.00438720	0.30830256	0.00097128	0.00041446	0.00142544
C3	0.55403140	0.14079754	0.59985070	0.00004620	0.00002011	0.00004819
O1	0.72675395	0.08378319	0.62426581	0.00003740	0.00001689	0.00004230
O2	0.44091166	0.18404259	0.76111998	0.00003740	0.00001682	0.00004069

Table A.25: Positions and their uncertainties in $\mathbf{x}_{\text{num}}$ (L-alanine)

	U11	U22	U33	U12	U13	U23
N1	0.00554510	0.00609623	0.00424828	-0.00015867	0.00067863	0.00011948
H1a	0.02549738	0.02737437	0.01759412	0.00283709	0.01273538	0.00466275
H1b	0.01335627	0.02819776	0.02173433	0.00006543	0.00666907	0.00259999
H1c	0.03626280	0.02106064	0.02613164	-0.01198870	0.02710476	-0.00906416
C1	0.00519359	0.00472992	0.00351623	0.00009210	-0.00044202	0.00043533
H1	0.02294134	0.01220274	0.01776506	-0.00104112	0.00410680	0.00804487
C2	0.00581726	0.00885965	0.00614227	-0.00046614	-0.00059180	-0.00132057
H2a	0.01876877	0.04731291	0.01609856	0.01033291	-0.01059506	-0.01006855
H2b	0.01052491	0.03069420	0.03288284	0.00396335	0.00789242	-0.00272177
H2c	0.03106142	0.01597713	0.04776419	-0.00472250	0.00079920	-0.00372660
C3	0.00487231	0.00426046	0.00355412	-0.00003286	-0.00019280	0.00022583
O1	0.00668279	0.00705602	0.00592401	0.00052419	-0.00063756	0.00244771
O2	0.00754966	0.00761951	0.00392352	-0.00055902	0.00060035	0.00210424

Table A.26: ADPs in $\mathbf{x}_{\mathrm{num}}$ (L-alanine)

	σ(U11)	σ(U22)	σ(U33)	σ(U12)	σ(U13)	σ(U23)
N1	0.00007289	0.00007206	0.00008590	0.00006465	0.00006494	0.00006436
H1a	0.00415352	0.00362832	0.00471804	0.00350406	0.00378231	0.00340582
H1b	0.00322269	0.00363505	0.00474835	0.00360738	0.00359986	0.00324674
H1c	0.00443657	0.00386887	0.00497369	0.00364313	0.00401598	0.00384358
C1	0.00008116	0.00007160	0.00009216	0.00006323	0.00006881	0.00006468
H1	0.00347419	0.00271630	0.00388888	0.00274262	0.00333884	0.00282823
C2	0.00008626	0.00008784	0.00010308	0.00007983	0.00007697	0.00007731
H2a	0.00391382	0.00470047	0.00440718	0.00362412	0.00329899	0.00342659
H2b	0.00315935	0.00391422	0.00516643	0.00361023	0.00368548	0.00313224
H2c	0.00485168	0.00299532	0.00569547	0.00348751	0.00443944	0.00327954
C3	0.00007477	0.00006967	0.00008829	0.00006398	0.00006993	0.00006785
O1	0.00006801	0.00006535	0.00008344	0.00006131	0.00006185	0.00005817
O2	0.00006849	0.00006421	0.00007608	0.00005687	0.00006302	0.00006078

Table A.27: ADP Uncertainties in $\mathbf{x}_{\mathrm{num}}$ (L-alanine)

Bibliography

[1] Erik Agrell, Thomas Eriksson, Alexander Vardy, and Kenneth Zeger. Closest point search in lattices. *IEEE Transactions on Information Theory*, 48(8):2201–2214, 2002.

[2] G Artioli, Hugo L Monaco, Davide Viterbo, G Ferraris, Gastone Gilli, G Zanotti, M Catti, M Milanesio, and C Giacovazzo. *Fundamentals of crystallography*. Oxford university press, USA, 2011.

[3] Roland Boese, Norbert Niederprüm, Dieter Bläser, Andreas Maulitz, Mikhael Yu Antipin, and Paul R Mallinson. Single-crystal structure and electron density distribution of ammonia at 160 k on the basis of x-ray diffraction data. *J. Phys. Chem. B*, 101(30):5794–5799, 1997.

[4] Luc J Bourhis, Oleg V Dolomanov, Richard J Gildea, Judith AK Howard, and Horst Puschmann. The anatomy of a comprehensive constrained, restrained refinement program for the modern computing environment–olex2 dissected. *Acta Crystallographica Section A: Foundations and Advances*, 71(1):59–75, 2015.

[5] Paul D. Boyle. *COSET*: a program for deriving and testing merohedral and pseudo-merohedral twin laws. *Journal of Applied Crystallography*, 47(1):467–470, Feb 2014.

[6] RF Bryan. *International tables for crystallography. Vol. C. Mathematical, physical and chemical tables edited by AJC Wilson.* International Union of Crystallography, 1993.

[7] Richard L Burden and J Douglas Faires. *Numerical analysis*, volume 9. Cengage learning, 2011. `https://fac.ksu.edu.sa/sites/default/files/numerical_analysis_9th.pdf`.

[8] Christian Böhmer, Luke Corcoran, Amit Einav, Cameron L. Hall, James P. Harris, Laura Midgley, Michael J. Negus, and Norbert Peyerimhoff. The twinning problem. *Mathematics in Industry Reports*, 2021. DOI=10.33774/miir-2021-64667-v2.

[9] Michał Leszek Chodkiewicz, Magdalena Woińska, and Krzysztof Woźniak. Hirshfeld atom like refinement with alternative electron density partitions. *IUCrJ*, 7(6):1199–1215, 2020.

[10] Jeremy Karl Cockcroft. A hypertext book of crystallographic space group diagrams and tables. `http://img.chem.ucl.ac.uk/sgp/`.

[11] Richard I. Cooper, Robert O. Gould, Simon Parsons, and David J. Watkin. The derivation of non-merohedral twin laws during refinement by analysis of poorly fitting intensity data and the refinement of non-merohedrally twinned crystal structures in the program *CRYSTALS*. *Journal of Applied Crystallography*, 35(2):168–174, Apr 2002.

[12] Tat'yana N. Drebushchak, Elena V. Boldyreva, and Elena S. Shutova. β-Glycine. *Acta Crystallographica Section E*, 58(6):o634–o636, Jun 2002.

[13] Jack D Dunitz. *X-ray Analysis and the Structure of Organic Molecules.* Cornell University Press, 1979.

[14] William Feller. An introduction to probability theory and its applications. *1957*, 1971.

[15] H. D. Flack. The derivation of twin laws for (pseudo-)merohedry by coset decomposition. *Acta Crystallographica Section A*, 43(4):564–568, Jul 1987.

[16] GameLudere. Euler angles, hamilton's quaternions and video games. https://www.gameludere.com/2020/03/12/euler-angles-hamilton-quaternions-and-video-games/. Accessed: 2022-01-31.

[17] Henri P. Gavin. The levenberg-marquardt algorithm for nonlinear least squares curve-fitting problems. https://people.duke.edu/~hpgavin/ce281/lm.pdf. Accessed: 2022-07-18.

[18] Carmelo Giacovazzo, Hugo L Monaco, Gilberto Artioli, Davide Viterbo, Giovanni Ferraris, and C Giacovazzo. *Fundamentals of crystallography*, volume 7. Oxford University Press Oxford, 2002.

[19] Simon Grabowsky, Dylan Jayatilaka, Stefan Mebs, and Peter Luger. The electron localizability indicator from x-ray diffraction data—a first application to a series of epoxide derivatives. *Chem. Eur. J.*, 16(43):12818–12821, 2010.

[20] Ralf W Grosse-Kunstleve and Paul D Adams. On the handling of atomic anisotropic displacement parameters. *Journal of applied crystallography*, 35(4):477–480, 2002.

[21] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Crystallographica Section A*, 47(6):655–685, Nov 1991.

[22] William Rowan Hamilton. Ii. on quaternions; or on a new system of imaginaries in algebra. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(163):10–13, 1844.

[23] R. Herbst-Irmer and G. M. Sheldrick. Refinement of Twinned Structures with *SHELXL*97. *Acta Crystallographica Section B*, 54(4):443–449, Aug 1998.

[24] Fred L Hirshfeld. Bonded-atom fragments for describing molecular charge densities. *Theoretica chimica acta*, 44(2):129–138, 1977.

[25] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2013.

[26] IUCr. Crystallographic information framework. `https://www.iucr.org/resources/cif`. Accessed: 2022-06-14.

[27] IUCr. Iucr online dictionary. `https://dictionary.iucr.org`. Accessed: 2021-07-29.

[28] IUCr. Online dictionary of crystallography. `reference.iucr.org/dictionary/`.

[29] G. B. Jameson. On structure refinement using data from a twinned crystal. *Acta Crystallographica Section A*, 38(6):817–820, Nov 1982.

[30] Dylan Jayatilaka and Birger Dittrich. X-ray structure refinement using aspherical atomic density functions obtained from quantum-mechanical calculations. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(3):383–393, 2008.

[31] Florian Kleemiss, Oleg V Dolomanov, Michael Bodensteiner, Norbert Peyerimhoff, Laura Midgley, Luc J Bourhis, Alessandro Genoni, Lorraine A Malaspina, Dylan Jayatilaka, John L Spencer, Franser White, Bernhard Grundkötter-stock, Simon Steinhauer, Dieter Lentz, Horst Puschmann, and Simon Grabowsky. Accurate crystal structures and chemical properties from NoSpherA2. *Chemical Science*, 12(5):1675–1692, 2021.

[32] M. Lutz. General features of twinning. `http://www.cryst.chem.uu.nl/lutz/twin/gen_twin.html`. Accessed: 2021-07-24.

[33] Lorraine A Malaspina, Anna A Hoser, Alison J Edwards, Magdalena Woińska, Michael J Turner, Jason R Price, Kunihisa Sugimoto, Eiji Nishibori, Hans-Beat Bürgi, Dylan Jayatilaka, and Simon Grabowsky. Hydrogen atoms in

bridging positions from quantum crystallographic refinements: influence of hydrogen atom displacement parameters on geometry and electron density. *CrystEngComm*, 22(28):4778–4789, 2020.

[34] Lorraine A. Malaspina, Erna K. Wieduwilt, Justin Bergmann, Florian Kleemiss, Benjamin Meyer, Manuel F. Ruiz-López, Rumpa Pal, Emanuel Hupf, Jens Beckmann, Ross O. Piltz, Alison J. Edwards, Simon Grabowsky, and Alessandro Genoni. Fast and accurate quantum crystallography: From small to large, from light to heavy. *J. Phys. Chem. Lett.*, 10(22):6973–6982, November 2019.

[35] Werner Massa. *Crystal structure determination.* Springer Science & Business Media, 2016.

[36] HP McKean and H Dym. *Fourier series and integrals.* Academic Press, 1972.

[37] Laura Midgley, Luc J Bourhis, Oleg Dolomanov, Norbert Peyerimhoff, and Horst Puschmann. Crystallographic refinement using non-spherical form factors in olex2. refine. *arXiv preprint arXiv:1911.08847*, 2019.

[38] Laura Midgley, Luc J Bourhis, Oleg V Dolomanov, Simon Grabowsky, Florian Kleemiss, Horst Puschmann, and Norbert Peyerimhoff. Vanishing of the atomic form factor derivatives in non-spherical structural refinement–a key approximation scrutinized in the case of hirshfeld atom refinement. *Acta Crystallographica Section A: Foundations and Advances*, 77(6):519–533, 2021.

[39] Frank Neese. Software update: the orca program system, version 4.0. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(1):e1327, 2018.

[40] Massimo Nespolo. Research themes: Crystal twinning. `https://www.crystallography.fr/mathcryst/twins.htm`. Accessed: 2021-07-27.

[41] Massimo Nespolo. Tips and traps on crystal twinning: how to fully describe your twin. *Crystal Research and Technology*, 50(5):362–371, 2015.

[42] Massimo Nespolo and Giovanni Ferraris. Overlooked problems in manifold twins: Twin misfit in zero-obliquity tlqs twinning and twin index calculation. *Acta crystallographica. Section A, Foundations of crystallography*, 63:278–86, 06 2007.

[43] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2nd edition, 1999. `https://www.csie.ntu.edu.tw/~r97002/temp/num_optimization.pdf`.

[44] Gábor Oszlányi and András Sütő. The charge flipping algorithm. *Acta Crystallographica Section A: Foundations of Crystallography*, 64(1):123–134, 2008.

[45] Pascal Parois and Martin Lutz. Linear transformations of variance/covariance matrices. *Acta Crystallographica Section A: Foundations of Crystallography*, 67(4):383–390, 2011.

[46] Simon Parsons. Introduction to twinning. *Acta Crystallographica Section D: Biological Crystallography*, 59(11):1995–2003, 2003.

[47] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes : the art of scientific computing*. Cambridge Univ Press, New York, 3rd ed. edition, 2007.

[48] Benjamin P. Pritchard, Doaa Altarawy, Brett Didier, Tara D. Gibson, and Theresa L. Windus. New basis set exchange: An open, up-to-date resource for the molecular sciences community. *J. Chem. Inf. Model.*, 59(11):4814–4820, 2019.

[49] Python profilers. `https://docs.python.org/3/library/profile.html`. Accessed: 2022-01-19.

[50] Python timeit module. `https://docs.python.org/3/library/timeit.html#module-timeit`. Accessed: 2022-01-19.

[51] D. Shechtman, I. Blech, D. Gratias, and J. W. Cahn. Metallic phase with long-range orientational order and no translational symmetry. *Phys. Rev. Lett.*, 53:1951–1953, Nov 1984.

[52] George M. Sheldrick. Shelxl program details. `http://xray.chem.ualberta.ca/xray/shelxl`. Accessed: 2022-06-24.

[53] Uri Shmueli. *International Tables for Crystallography, Vol. B: Reciprocal Space.* Springer Science & Business Media, 2008.

[54] Anthony L. Spek. *checkCIF* validation ALERTS: what they mean and how to respond. *Acta Crystallographica Section E*, 76(1):1–11, Jan 2020.

[55] Elias M Stein and Rami Shakarchi. *Fourier analysis: an introduction*, volume 1. Princeton University Press, 2011.

[56] Robert F Stewart, Ernest R Davidson, and William T Simpson. Coherent x-ray scattering for the hydrogen atom in the hydrogen molecule. *The Journal of Chemical Physics*, 42(9):3175–3187, 1965.

[57] 17th intensive teaching school in x-ray structure analysis, 2019.

[58] Tugraz. Atomic form factors. `lampx.tugraz.at/~hadley/ss1/crystaldiffraction/atomicformfactors/formfactors.php`.

[59] University of Oklahoma. Symmetry in crystallography. `xrayweb.chem.ou.edu/notes/symmetry.html`. Accessed: 2017-10-10.

[60] James M Van Verth and Lars M Bishop. *Essential mathematics for games and interactive applications.* CRC Press, 2015.

[61] VK Wadhawan. A tensor classification of twinning in crystals. *Acta Crystallographica Section A: Foundations of Crystallography*, 53(5):546–555, 1997.