Application of Modern Techniques in Crystallographic Software Development

Richard J. Gildea

Department of Chemistry

Durham University

A thesis submitted for the degree of

Doctor of Philosophy

March 2011

Declaration

The work described herein was carried out at Durham University between October 2007 and March 2011 under the supervision of Professor Judith A.K. Howard. Unless otherwise stated all the work is my own and has not been submitted previously for a degree at this or any other university.

Richard J. Gildea

Acknowledgements

I would like to start by giving thanks to my supervisor, Professor Judith Howard, for having the trust in allowing me the freedom to explore those areas of the project I found most interesting, and for providing support and giving me the opportunity to visit several international conferences during the course of my PhD.

I am hugely grateful to all of the Olex2 team: Luc Bourhis, Oleg Dolomanov and Horst Puschmann. You are a fantastic group of people to work with, and my PhD would not have been anywhere near as much fun without any one of you.

Luc Bourhis for his patience in spending so many hours explaining mathematical and crystallographic concepts in detail, and for putting up with all my questions about programming. Oleg Dolomanov for the time spent donating his crystallographic and programming expertise and countless hours spent debugging code and compiler errors. Thanks for those beers in Istanbul, and the innumerable "last" pints in the Market Tavern. Horst Puschmann for his constantly positive attitude, for putting up with me sitting next to him for three and a half years, and for counting pixels.

I would like to thank my parents for their support and encouragement throughout my education and for pushing me to reach my full potential. Finally, I must thank all the friends I have made during my time in Durham, especially those with whom I have lived with throughout the years, you have made my stay in Durham all the more enjoyable.

Abstract

This thesis describes contributions made as part of the EPSRC-funded project *Age Concern: Crystallographic Software for the Future*. Work has been done in various areas of small molecule crystallographic software development, both within the smtbx (Small Molecule Toolbox) and the Olex2 software.

Chapter 2 details the work that was done towards the smtbx-based refinement that was developed as part of the "Age Concern" project. A framework was created enabling the inclusion of observations of restraint in the refinement, and new restraints on geometry and anisotropic displacement parameters were added. Refinement of (pseudo-)merohedrally twinned structures was implemented.

In Chapter 3 a description of the determination of absolute structure by various methods is given. The methods of Hooft et al. [2008] and Flack [1983] have been implemented, and a quantitative comparison made between the two methods.

Chapter 4 discusses the method of van der Sluis and Spek [1990] for the refinement of structures containing severely disordered regions. This method has been implemented and a modification designed to give improved results when one or more low angle reflections are missing is proposed and tested, and shown to be beneficial.

Chapter 5 introduces a new module, iotbx.cif, which has been added to the cctbx (Computational Crystallography Toolbox), providing a comprehensive set of tools for the manipulation of Crystallographic Information Files (CIFs).

Contents

List of Figures

Nomenclature

Roman Symbols

$\mathbf{A}^t$	The transpose of the matrix $\mathbf{A}$.
$\mathbf{h}$	A column vector of the Miller indices of a Bragg reflection.
$\mathbf{h}^t$	The transpose of the column vector $\mathbf{h}$ (*i.e.* a row vector).
$F(\mathbf{h})$	The complex structure factor associated with Miller indices $\mathbf{h}$.

Greek Symbols

σ	Standard deviation or uncertainty

Acronyms

ADP	Anisotropic Displacement Parameter
ASCII	American Standard Code for Information Interchange
CBF	Crystallography Binary Format
cctbx	Computational Crystallography Toolbox
CIF	Crystallographic Information File (Framework)
COD	Crystallography Open Database
CSD	Cambridge Structural Database
EPSRC	Engineering and Physical Sciences Research Council

GUI	Graphical User Interface
HTML	HyperText Markup Language
IUCr	International Union of Crystallography
smtbx	Small Molecule Toolbox

Chapter 1

Introduction

1.1 Age Concern: Crystallographic Software for the Future

The work described in this thesis is part of a larger software project between groups in Durham and Oxford funded by the EPSRC of the UK[1], with the title *Age Concern: Crystallographic Software for the Future*. The background to this project, and also the aims and objectives as outlined in the grant proposal, are described in detail by Howard and Watkin [2009] and Dolomanov et al. [2009b]. They highlight that whilst in previous decades (1960s, 1970s, 1980s) there was healthy competition amongst a wide variety of actively developed crystallographic systems, in recent years only relatively few are still under active development and commonly used within the small molecule community. They noted that many of the authors of significant programs are approaching retirement, with no clear indication of who would take their place, either through continued development of the existing programs, or by development of a new generation of crystallographic software.

Much of the computer code currently used in small molecule crystallography has its foundations in code written up to 40 years ago, using older programming languages and techniques. Consequently, it is frequently difficult for such code to be extended significantly, or reused in a different context, in particular by

[1]EPSRC Grant EP/C536274/1

developers other than the original authors. Nonetheless, there is a huge amount of knowledge and experience that is coded within these programs, which any new software should strive to incorporate.

In contrast to the small molecule crystallographic community, there currently exist two substantial multi-author efforts within macromolecular crystallography that coordinate the software developments of multiple groups of programmers, namely CCP4 [Potterton et al., 2004] and PHENIX [Adams et al., 2010].

In view of the massive advances both in computer hardware and programming techniques since those long-standing programs were first conceived, it was proposed to provide a new crystallographic software framework implemented in modern programming languages and written in a style designed to maximise extensibility and reusability of code. In addition to providing much of the functionality of the software in common use, this new framework should ensure that new ideas and algorithms in crystallographic computing can be developed rapidly and effectively, and made available to the wider crystallographic community with minimal effort.

A reference application would be developed which would at the same time serve as a test-application for the development of the newly created framework, whilst also providing the crystallographic community with a *fully functional, single crystal refinement application with unprecedented functionality, flexibility, customisability and extensibility.*

It was decided that the crystallographic software framework would be based upon the pre-existing cctbx (Computational Crystallography Toolbox) which is described in §1.3. A new small molecule toolbox, the smtbx, would provide a set of algorithms dedicated to small molecule crystallography, whilst tools developed in the course of the project that are more generally applicable to the whole of crystallography would be added to the cctbx itself, thus contributing back to the wider crystallographic software community as a whole.

The reference application mentioned in the proposal became the software Olex2, which would provide access to the new tools developed within the smtbx as they became available.

1.2 Olex2

A more comprehensive overview of the Olex2 software and its design and implementation is given by Dolomanov et al. [2009a]. The core of the program is written using the C++ programming language and is highly optimised for excellent graphical performance. The program is designed as a set of libraries which can be re-used to build applications with minimal dependencies. Separate libraries are concerned with core functionality, crystallographic operations, input/output and graphical display. As a result, a command line version of the Olex2 executable exists in addition to the graphical user interface (GUI). The graphical display of the model uses the OpenGL [Khronos Group] library. Extended functionality of the Olex2 core is achieved in two ways: through the use of a built-in macro language; or through the provision of an embedded Python interpreter. The *control panel* section of the GUI is written using extended HTML which is displayed using wxWidgets. This provides a set of GUI controls which support event-driven execution, allowing the creation of a clearly laid out and easy-to-follow workflow path.

Much of the overall workflow (especially with regard to structure solution, refinement, and report preparation stages) is written in Python/HTML. Functions or macros provided by the Olex2 core can be accessed either through the command console, which is part of the OpenGL window, or through functionality provided by the GUI.

The Python layer allows the integration of the cctbx (Computational Crystallography Toolbox) and its subpackage, the smtbx (Small Molecule Toolbox). This vastly extends the functionality available through Olex2, including tools for structure solution and refinement.

In addition to the structure solution and refinement methods provided through the smtbx, Olex2 also supports the SHELX suite of solution and refinement programs [Sheldrick, 2008]. Plugins have also been developed by interested users providing access to a range of external programs including PLATON [Spek, 2003], the structure solution program SUPERFLIP [Palatinus and Chapuis, 2007] and the SIR9x-SIR20xx range of structure solution programs [Burla et al., 2007].

The latest installers for Windows, Mac and Linux are included on the DVD

accompanying this thesis. Details regarding their installation can be found in Appendix D.

1.3 Computational Crystallographic Toolbox

The Computational Crystallographic Toolbox (cctbx) is an open source code library originally developed as the open source component of the PHENIX system [Adams et al., 2010] for macromolecular structure determination. It features an object-oriented, highly modular design, which encourages code reuse across many different applications. The cctbx is written using a combination of two modern programming languages, Python [Python Software Foundation] and C++, which provides the flexibility of using an interpreted language (Python) at the same time as the performance benefits gained through using a statically typed, compiled language (C++). Python bindings for C++ code are written using the Boost.Python library. The cctbx code is extremely portable, and is known to compile on a large range of hardware and platforms. The writing of regression tests is actively encouraged, contributing to the stability of the cctbx.

The foundation of the cctbx is the scitbx module, which provides a large number of tools for general scientific computing. Built upon this is the cctbx module, a set of libraries for general crystallographic applications. The iotbx (input/output toolbox) provides libraries for reading and writing most common crystallographic formats. For an in-depth discussion of the design of the cctbx the reader is referred to Grosse-Kunstleve et al. [2002] and the several cctbx articles in the newsletters of the IUCr Commission on Crystallographic Computing, in particular the very first one [Grosse-Kunstleve and Adams, 2003].

The latest cctbx source code bundles are included on the DVD accompanying this thesis. Details regarding their extraction and compilation can be found in Appendix D.

1.4 Small Molecule Toolbox

The Small Molecule Toolbox (smtbx) is an extension of the cctbx with a particular emphasis on the provision of algorithms and tools that are specific to small molecule crystallography. Currently it provides *ab initio* structure solution using the charge flipping algorithm [Oszlányi and Süto, 2008], full matrix least squares refinement of crystal structures with constraints and restraints on parameters, an implementation of the BYPASS algorithm for treating severely disordered solvent in structure refinement [van der Sluis and Spek, 1990], and tools for the determination of absolute structure.

1.4.1 Outline

Chapter 1 describes work carried out as part of the development of the least squares refinement program, smtbx-refine. §2.1 describes the framework that was implemented to allow the inclusion of restraints on anisotropic displacement parameters and geometry in the refinement. The addition of refinement of (pseudo)merohedrally twinned crystal structures is described in §2.2, and §2.3 details the calculation of errors on derived parameters.

§3.1 contains a discussion of the various methods for the determination of absolute structure, along with a description of the implementation of two of those methods within the smtbx. A quantitative comparison is made between the two methods. A description of the various graphs for the analysis of reflection statistics that have been implemented using the cctbx is given in §3.2. The graphs are made available using the new graph plotting tool implemented in Olex2.

Chapter 4 contains a description of the procedure of van der Sluis and Spek [1990] for dealing with severely disordered solvent. The procedure has been implemented within the smtbx and a modification is proposed in §4.2.2 that is intended to give improved results for the procedure when some low angle data are missing. Several test cases and applications of the procedure are given.

A new module has been added to the cctbx providing extensive support for the Crystallographic Information Framework (CIF); a description of the implementation and capabilities of the new module is given in Chapter 5.

Chapter 2

Least Squares Refinement

A crystal structure X-ray diffraction experiment yields a set of intensities of diffracted X-ray beams which contain information about the electron density distribution in the unit cell. The Fourier transform relationship between the electron density, $\rho(\mathbf{x})$, and the structure factors, $F(\mathbf{h})$, is given by:

$$\rho(\mathbf{x}) = V^{-1} \sum_{\mathbf{h}} |F(\mathbf{h})| \exp(i\phi_{\mathbf{h}}) \exp(-2\pi i \mathbf{h} \cdot \mathbf{x}) \tag{2.1}$$

and

$$F(\mathbf{h}) = \int_{cell} \rho(x) \exp(2\pi i \mathbf{h} \cdot \mathbf{x})\, dx, \tag{2.2}$$

where $\mathbf{h}$ is a column vector of the Miller indices for a Bragg reflection.

The electron density is usually interpreted in terms of an atomic model and the structure factors can then be calculated according to

$$F(\mathbf{h}) \simeq \sum_{j}^{atoms} f_j T(\mathbf{h}) \exp(2\pi i \mathbf{h} \cdot \mathbf{x}), \tag{2.3}$$

where f_j is the scattering factor calculated for an atom at zero Kelvin and $\mathbf{x} = (x, y, z)$ are the atomic coordinates. The Debye-Waller factor, $T(\mathbf{h})$, is given by

$$T(\mathbf{h}) = \exp(-2\pi^2 \mathbf{h}^t \mathbf{U}^* \mathbf{h}), \tag{2.4}$$

where $\mathbf{U}^*$ is a symmetric second-rank tensor whose elements are dimensionless

mean-square displacements. $\mathbf{U}^*$ is one of several definitions of the anisotropic displacement parameters (ADPs) [Grosse-Kunstleve and Adams, 2002].

Once an atomic model is proposed, the parameters of the model can be varied in order to obtain the best possible model given the experimental data. In small molecule crystallography, this is usually achieved by least squares refinement of the structural parameters.

2.1 Restrained Least Squares Refinement

A small molecule structure refinement typically minimises the weighted least squares function

$$L = \sum_{\mathbf{h}} w_{\mathbf{h}} (Y_{\text{obs}}(\mathbf{h}) - kY_{\text{calc}}(\mathbf{h}))^2 \tag{2.5}$$

where Y_{obs} are the X-ray observations, either F_{obs} or F^2_{obs}, and Y_{calc} are similarly $|F_{\text{calc}}|$ or $|F_{\text{calc}}|^2$ where F_{calc} are the structure factors calculated from the current structure model according to equation 2.3, and k is an overall scale factor that places Y_{calc} on the same scale as Y_{obs}. Each observation is given an appropriate weight, $w_{\mathbf{h}}$, based on the reliability of the measurement. These may be pure statistical weights, $w = 1/\sigma^2(Y_{\text{obs}})$, where σ is the estimated standard deviation of the Y_{obs}, although more complex weighting schemes are usually used.

Since the minimisation function introduced above is not linear, the minimisation is non-linear least squares, which requires that we calculate the gradients of Y_{calc} with respect to each parameter. For a small molecule structure with a high data to parameter ratio, such unconstrained minimisation as defined by equation 2.5 may well be sufficient. However, as the structure becomes larger, or the data to parameter ratio worsens, unconstrained minimisation may not be well-behaved, or result in some questionable parameter values. These X-ray observations can be supplemented with the use of 'observations of restraint', as suggested by Waser [1963], where additional information, such as target values for bond lengths, angles etc. is included in the minimisation. This now gives the minimisation function

$$L = \sum_{h} w_{\mathbf{h}} (Y_{\text{obs}}(\mathbf{h}) - kY_{\text{calc}}(\mathbf{h}))^2 + \sum_{\text{restraints}} w(T_{\text{obs}} - T_{\text{calc}})^2 \tag{2.6}$$

where T_{obs} is the target value for our restraint, and T_{calc} is the value of the target function calculated using the current model (see, for example Giacovazzo et al. [2002]; Watkin [2008]). With the use of appropriate weighting of the restraints the minimisation is gently pushed towards giving a chemically sensible and hopefully correct structure.

Using the notation of Watkin [2008], the observational least squares equations can be written

$$\mathbf{W} \cdot \mathbf{A} \cdot \delta\mathbf{x} = \mathbf{W} \cdot \Delta\mathbf{Y}, \tag{2.7}$$

with the weight matrix and the vector of residuals, $\Delta\mathbf{Y}$, where each row is given by $Y_{\mathrm{obs}}(\mathbf{h}) - kY_{\mathrm{calc}}(\mathbf{h})$. The elements of the matrix of derivatives, $\mathbf{A}$, are given by

$$A_{ij} = \frac{\partial Y_c(\mathbf{h}_i)}{\partial x_i}. \tag{2.8}$$

The shifts, $\delta\mathbf{x}$, in the values of the refined parameters are obtained *via* the solution of the normal equations,

$$\mathbf{A^T} \cdot \mathbf{W} \cdot \mathbf{A} \cdot \delta\mathbf{x} = \mathbf{A^T} \cdot \mathbf{W} \cdot \Delta\mathbf{Y}. \tag{2.9}$$

If we allow the repameterisation of the model by use of constraints, the vector of parameters, $\mathbf{x}$ is expressed as a function of a smaller vector of parameters, $\mathbf{y}$, in a non-linear fashion. The linearisation of that relationship reads

$$\delta\mathbf{x} = \mathbf{M}\delta\mathbf{y} \tag{2.10}$$

where $\mathbf{M}$ is the matrix of constraint, usually known to mathematicians as the Jacobian matrix of the transformation $\mathbf{y} \rightarrow \mathbf{x}$.

Since the normal matrix, $\mathbf{A^T} \cdot \mathbf{W} \cdot \mathbf{A}$ is symmetric, it can be inverted using the Cholesky method. A naïve approach to solving these equations would start by first of all constructing the matrix of derivatives, $\mathbf{A}$. This is not feasible, since the design matrix is of size $m \times n_x$, for m observations, and n_x crystallographic parameters. In a typical small molecule crystal structure determination, the data to parameter ratio, m/n_x is typically in the range $10 - 30$. In contrast, the normal matrix, $\mathbf{A^T} \cdot \mathbf{W} \cdot \mathbf{A}$ is symmetric, with dimensions $n_x \times n_x$. With the common use

of constraints, particularly with respect to those on the parameters of hydrogen atoms, the ratio n_x/n_y can be as large as 2, meaning that the most efficient, both in terms of storage and floating point operations, would in fact be to construct directly the normal matrix for the independent parameters, $\mathbf{M^T A^T \cdot W \cdot AM}$.

Whilst the part of the design matrix derived from the observations is relatively dense, that coming from the equations of restraint is sparse, with each restraint typically only involving a few crystallographic parameters. Therefore, it is now feasible to compute and store the design matrix for the restraints independently, and then use sparse matrix techniques to compute the contribution of the restraints to the overall normal equations.

It would be desirable to place the weights of the restraints on the same scale as the typical residual, such that a restraint will have a similar strength for the same weight in different structures. Giacovazzo et al. [2002] suggest the normalization factor

$$w_{\text{restraints}} = \sum_{\mathbf{h}} w_{\mathbf{h}} (Y_{\text{obs}}(\mathbf{h}) - kY_{\text{calc}}(\mathbf{h}))^2 / (m - n_y) , \tag{2.11}$$

where for m observations and n_y independent parameters. This is better known as the square of the *goodness of fit*, χ^2. Ths normalising factor also allows the restraints to have greater influence when the fit of the model to the data is poor (and the goodness of fit is greater than unity), whilst their influence lessens as the fit improves [SHELX manual, Sheldrick, 1997].

2.1.1 Geometry Restraints

Possible restraints on the stereochemistry or geometry of atomic positions include restraints on bond distances, angles and dihedral angles, chiral volume and planarity. These restraints are used extensively in macromolecular crystallography, and hence were already implemented within the cctbx as part of the macromolecular refinement program *phenix.refine* [Adams et al., 2010]. With the exception of the bond distance restraint, these restraints were not able to accept symmetry equivalent atoms. Since this is more frequently required in small molecule crystallography, these restraints have now been extended to allow for symmetry. We have also implemented other restraints commonly used in small molecule struc-

ture refinement, such as a bond similarity restraint, and restraints on anisotropic displacement parameters (ADPs) including restraints based on Hirshfeld's 'rigid-bond' test [Hirshfeld, 1976], similarity restraints and isotropic ADP restraints.

2.1.1.1 Restraints involving symmetry

Given a restraint, $f(x)$, involving a site x which is outside the asymmetric unit and which is related to the site y within the asymmetric unit by some symmetry transformation M, such that $x = My$, the gradient is transformed as

$$\begin{aligned} \nabla_y \left(f\left(x \right) \right) &= M^T \nabla_x \left(f\left(x \right) \right) \\ &= M^{-1} \nabla_x \left(f\left(My \right) \right) \end{aligned} \tag{2.12}$$

since M is a space group symmetry operation and is therefore an orthogonal transformation (*i.e.* one which preserves distances and angles), which means that, $M^T = M^{-1}$.

2.1.1.2 Bond similarity restraint

The distances between two or more atom pairs are restrained to be equal by minimising the weighted variance of the distances, where the least squares residual, R, is defined as the population variance biased estimator

$$R(r_1, ..., r_n) = \frac{\sum_{i=1}^{n} w_i (r_i - \langle r \rangle)^2}{\sum_{i=1}^{n} w_i}. \tag{2.13}$$

As discussed above, since our minimisation is non-linear, we need the derivatives of the residuals with respect to the least squares parameters. It is easier to compute the derivatives by using the alternative form of the residual

$$\begin{aligned} R &= \left\langle r^2 \right\rangle - \langle r \rangle^2 \\ &= \frac{\sum_{i=1}^{n} w_i r_i^2}{\sum_{i=1}^{n} w_i} - \left(\frac{\sum_{i=1}^{n} w_i r_i}{\sum_{i=1}^{n} w_i} \right)^2. \end{aligned} \tag{2.14}$$

The derivative of the residual with respect to a distance r_j is then

$$\begin{aligned}\frac{\partial R}{\partial r_j} &= \frac{2w_j r_j}{\sum_{i=1}^{n} w_i} - \frac{2w_j \sum_{i=1}^{n} w_i r_i}{(\sum_{i=1}^{n} w_i)^2} \\ &= \frac{2w_j}{\sum_{i=1}^{n} w_i}(r_j - \langle r \rangle).\end{aligned} \tag{2.15}$$

Given that

$$r_j = u^{\frac{1}{2}},$$

where for a pair of atoms, a and b,

$$u = (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2,$$

the derivative of r_j with respect to the Cartesian coordinate x_a is then

$$\frac{\partial r_j}{\partial x_a} = \frac{\partial r_j}{\partial u}\frac{\partial u}{\partial x_a} = \frac{(x_a - x_b)}{r_j}. \tag{2.16}$$

Therefore, the derivative of the residual with respect to x_a is

$$\begin{aligned}\frac{\partial R}{\partial x_a} &= \frac{\partial R}{\partial r_j}\frac{\partial r_j}{\partial x_a} \\ &= \frac{2w_j(r_j - \langle r \rangle)(x_a - x_b)}{r_j \sum_{i=1}^{n} w_i}.\end{aligned} \tag{2.17}$$

2.1.2 Restraints on Atomic Displacement Parameters

There appears to be very little in the literature with regard to restraints on ADPs, and in particular the details of their implementation in refinement programs. It was therefore necessary to devise our formulae for the equations of restraints and derive their gradients with respect to the least squares parameters. The analytical gradients were confirmed to be correct by testing against gradients determined by the finite differences method. The residuals were also tested for frame invariance (*i.e.* for a given $\mathbf{U}_{\text{cart}}$, the least squares residual should be unchanged after transformation of $\mathbf{U}_{\text{cart}}$ by an arbitrary rotation matrix).

2.1.2.1 Rigid-bond restraint

In a 'rigid-bond' restraint the components of the anisotropic displacement parameters of two atoms in the direction of the vector connecting those two atoms are restrained to be equal. This corresponds to Hirshfeld's 'rigid-bond' test [Hirshfeld, 1976] for testing whether anisotropic displacement parameters are physically reasonable [see SHELX manual, DELU restraint, Sheldrick, 1997] and is in general appropriate for bonded and 1,3-separated pairs of atoms and should hold true for most covalently bonded systems.

We therefore minimise the mean square displacement of the atoms in the direction of the bond. The weighted least squares residual is then

$$R = w(z^2_{A,B} - z^2_{B,A})^2, \tag{2.18}$$

where in the Cartesian coordinate system the mean square displacement of atom A along the vector $\overrightarrow{AB}$, $z^2_{A,B}$, is given by

$$z^2_{A,B} = \frac{\mathbf{r}^T \mathbf{U}_{cart,A} \mathbf{r}}{\|\mathbf{r}\|^2}, \tag{2.19}$$

where

$$\mathbf{r} = \begin{pmatrix} x_A - x_B \\ y_A - y_B \\ z_A - z_B \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{2.20}$$

$\mathbf{r}^T$ is the transpose of $\mathbf{r}$ (*i.e.* a row vector) and $\|\mathbf{r}\|$ is the length of the vector $\overrightarrow{AB}$.

The derivative of the residual with respect to an element of $\mathbf{U}_{cart,A}$, $U_{A,ij}$ is given by (using the chain rule)

$$\frac{\partial R}{\partial U_{A,ij}} = \frac{\partial R}{\partial z^2_{A,B}} \frac{\partial z^2_{A,B}}{\partial U_{A,ij}} \tag{2.21}$$

$$= 2w(z^2_{A,B} - z^2_{B,A}) \frac{\partial z^2_{A,B}}{\partial U_{A,ij}} \tag{2.22}$$

The matrix multiplication in obtaining $z^2_{A,B}$ can be evaluated as follows (re-

membering $\mathbf{U}_{cart}$ is symmetric):

$$\mathbf{r}^T\mathbf{U}_{cart,A}\mathbf{r} = \begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{12} & U_{22} & U_{23} \\ U_{13} & U_{23} & U_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2.23}$$

$$= U_{11}\,x^2 + U_{22}\,y^2 + U_{33}\,z^2 + 2U_{12}\,xy + 2U_{13}\,xz + 2U_{23}\,yz \tag{2.24}$$

It then follows that

$$\frac{\partial z_{A,B}^2}{\partial U_{11}} = \frac{x^2}{\|\mathbf{r}\|^2}, \qquad \frac{\partial z_{A,B}^2}{\partial U_{22}} = \frac{y^2}{\|\mathbf{r}\|^2}, \qquad \frac{\partial z_{A,B}^2}{\partial U_{33}} = \frac{z^2}{\|\mathbf{r}\|^2}, \tag{2.25}$$

and

$$\frac{\partial z_{A,B}^2}{\partial U_{12}} = \frac{2xy}{\|\mathbf{r}\|^2}, \qquad \frac{\partial z_{A,B}^2}{\partial U_{13}} = \frac{2xz}{\|\mathbf{r}\|^2}, \qquad \frac{\partial z_{A,B}^2}{\partial U_{23}} = \frac{2yz}{\|\mathbf{r}\|^2}. \tag{2.26}$$

These can be combined with eqn (2.22) to give us the derivatives with respect to each U_{ij} component.

2.1.2.2 ADP similarity restraint

The anisotropic displacement parameters of two atoms are restrained to have the same U_{ij} components. Since this is only a rough approximation to reality, this restraint should be given a smaller weight in the least squares minimisation than for a rigid-bond restraint and is suitable for use in larger structures with a poor data to parameter ratio. Applied correctly, this restraint permits a gradual increase and change in direction of the anisotropic displacement parameters along a side-chain [Sheldrick, 1997]. This is equivalent to a SHELXL SIMU restraint [Sheldrick, 1997]. The weighted least squares residual is defined as

$$R = w \sum_{i=1}^{3} \sum_{j=1}^{3} (U_{A,ij} - U_{B,ij})^2, \tag{2.27}$$

which, denoting $\Delta U = U_A - U_B$ the matrix of deltas, is the trace of $\Delta U \Delta U^T$. This expression[1] makes it clear that it is invariant under any rotation R, since it

[1] This is known to mathematicians as the square of the Frobenius norm of the matrix ΔU.

transforms ΔU into $R\Delta U R^T$. Since $\mathbf{U}$ is symmetric, *i.e.* $U_{ij} = U_{ji}$, this can be rewritten as

$$R = w\left(\sum_{i=1}^{3}(U_{A,ii} - U_{B,ii})^2 + 2\sum_{i<j}(U_{A,ij} - U_{B,ij})^2\right). \tag{2.28}$$

Therefore the gradient of the residual with respect to the diagonal element $U_{A,ii}$ is then

$$\frac{\partial R}{\partial U_{A,ii}} = 2w(U_{A,ii} - U_{B,ii}). \tag{2.29}$$

Similarly the gradient with respect to the off-diagonal element $U_{A,ij}$ is

$$\frac{\partial R}{\partial U_{A,ij}} = 4w(U_{A,ij} - U_{B,ij}). \tag{2.30}$$

2.1.2.3 Isotropic ADP restraint

Here we minimise the difference between the Cartesian ADPs, $\mathbf{U}_{cart}$, and the isotropic equivalent, $\mathbf{U}_{eq}$. Again, this is an approximate restraint and as such should have a comparatively small weight. A common use for this restraint would be for solvent water, where the two restraints discussed previously would be inappropriate [Sheldrick, 1997]. As in §2.1.2.2, we must remember that we are dealing with symmetric matrices, and we can therefore define the weighted least squares residual as

$$R = w\left(\sum_{i=1}^{3}(U_{ii} - U_{eq,ii})^2 + 2\sum_{i<j}(U_{ij} - U_{eq,ij})^2\right), \tag{2.31}$$

where

$$\mathbf{U}_{eq} = \begin{pmatrix} U_{iso} & 0 & 0 \\ 0 & U_{iso} & 0 \\ 0 & 0 & U_{iso} \end{pmatrix}, \tag{2.32}$$

and

$$U_{iso} = \tfrac{1}{3}\mathrm{tr}(\mathbf{U}_{cart}). \tag{2.33}$$

We expand the summation of the residual as follows

$$R = w\left((U_{11} - U_{iso})^2 + (U_{22} - U_{iso})^2 + (U_{33} - U_{iso})^2 + 2U_{12}^2 + 2U_{13}^2 + 2U_{23}^2\right). \tag{2.34}$$

We can now see by inspection that the derivatives of the residual with respect to the off-diagonal elements are

$$\frac{\partial R}{\partial U_{ij,i<j}} = 4wU_{ij}. \tag{2.35}$$

The derivatives of the residual with respect to the diagonal elements can be generalised as

$$\frac{\partial R}{\partial U_{ii}} = 2w(U_{ii} - U_{iso}). \tag{2.36}$$

2.1.3 Implementation

Some of the differences between typical macro-molecular and full matrix least squares cycles have been described by Bourhis et al. [2009]. Figure 2.1 illustrates the steps involved with building the normal equations. With the inclusion of observations of restraint in the minimisation target function

$$L = L_{\text{data}} + wL_{\text{restraints}}, \tag{2.37}$$

where using a least squares minimiser

$$L_{\text{data}} = \sum_h w_h \left(F_o(h)^2 - k\left|F_c(h)\right|^2\right)^2, \tag{2.38}$$

and

$$L_{\text{restraints}} = \sum_{\text{restraints}} w(T_{\text{obs}} - T_{\text{calc}})^2 \tag{2.39}$$

Due to the extremely large number of parameters in a typical macro-molecular refinement compared to that for the typical small molecule refinement, it is usually prohibitive to construct the normal matrix and solve the observational equations via the Cholesky method. As a result, there is only the need for a single array storing the gradient of the target function (equation 2.37) with respect to each

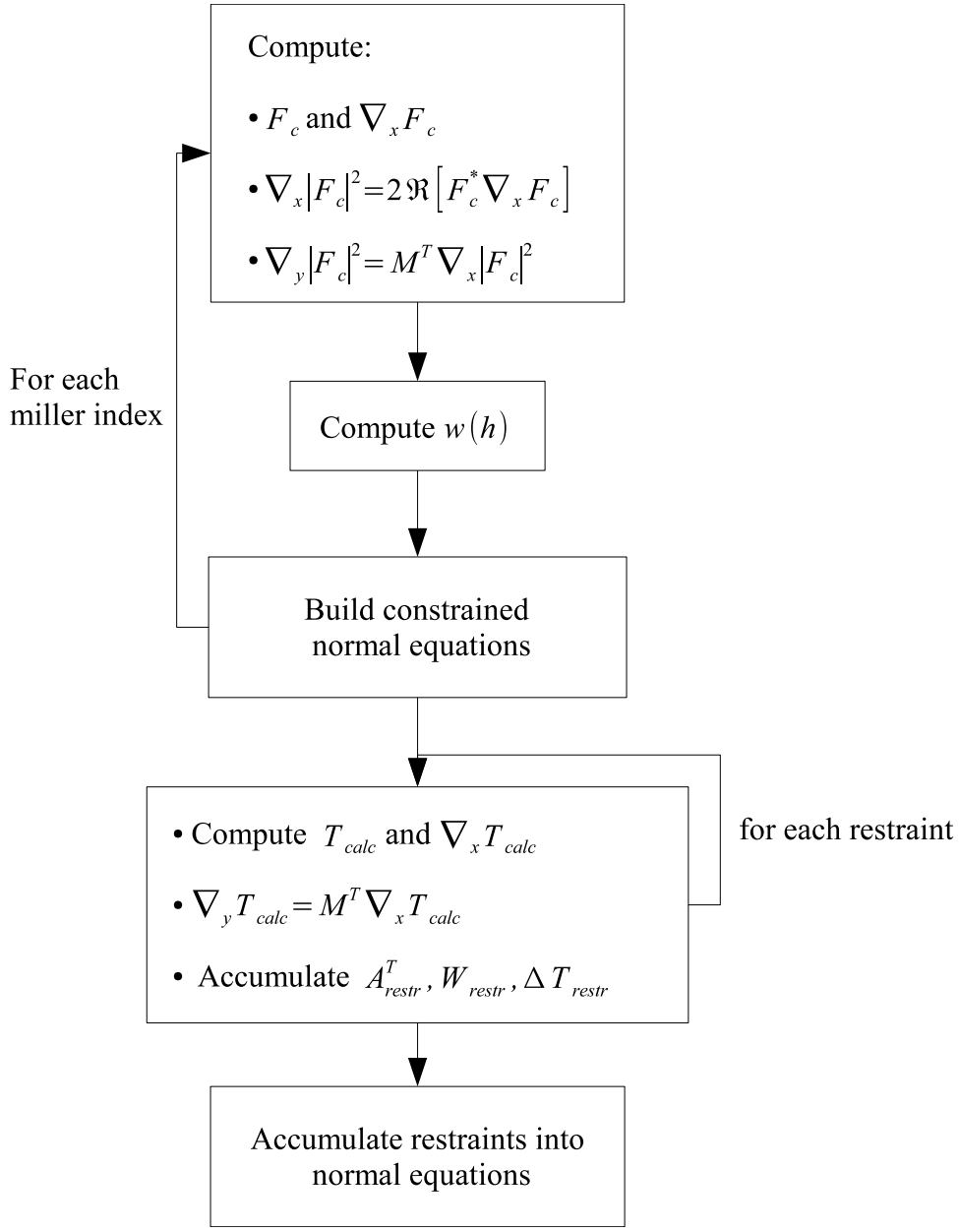


Figure 2.1: Flow diagram illustrating the steps taken when building up the normal equations.

parameter. The gradients ∇L_{data} and $\nabla L_{\text{restraints}}$ can be calculated separately before combining their sum to obtain ∇L which is to be passed to the minimiser. Note that it is possible to, for example, calculate the gradients of the restraints with respect to the sites in Cartesian coordinates (which is generally easier, especially for the geometrical restraints), and only at the very end transform the gradients back to fractional coordinates (it is usually fractional coordinates which are refined) before combining with the gradients from the experimental data. This also means that it is possible to make certain optimisations for the handling of restraints involving symmetry. In contrast, for full matrix least squares refinement the gradient for each restraint must be transformed to fractional coordinates individually (*i.e.* for each row of the design matrix).

One further complication due to the differences between using restraints in a macromolecular compared to a full matrix least squares context is that the minimisers require different gradients. For a restraint

$$L = w\left(T_{\text{obs}} - T_{\text{calc}}\right)^2 \tag{2.40}$$

then a minimiser such as the LBFGS minimiser, as used in the macromolecular refinement program *phenix.refine* [Adams et al., 2010], requires the gradient of L with respect to the parameters

$$\frac{\partial L}{\partial x} = 2w\left(T_{\text{obs}} - T_{\text{calc}}\right)\frac{\partial T_{\text{calc}}}{\partial x}, \tag{2.41}$$

whereas full matrix least squares requires simply $\frac{\partial T_{\text{calc}}}{\partial x}$.

In order to make the restraints function with either minimiser, it was necessary to provide access to both $\frac{\partial L}{\partial x}$ and $\frac{\partial T_{\text{calc}}}{\partial x}$ (of course, the former can be calculated as a by-product of the latter).

The route taken to add restraints into this framework was to build independently those rows of the design matrix associated with the equations of restraint. Since the restraints largely involve relatively few of the crystallographic parameters, it can be efficient to store this part of the design matrix as a sparse matrix. This allows the restraints to be built up without any knowledge of the constraint matrix, and only after the contribution of the data to the normal matrix has been

computed, the contribution of the restraints can be added efficiently with the use of sparse matrix techniques. The restraints framework was designed in such a way that it would be easy to add further restraints (*e.g.* the quotient restraints suggested by Parsons and Flack [2004]). All that is required is the array of derivatives of the restraint with respect to the parameters (one row of the design matrix), the restraint delta, $T_{\mathrm{obs}} - T_{\mathrm{calc}}$, and the weight, w, of the restraint.

As described by Grosse-Kunstleve et al. [2004], the restraints are split into three levels. The restraint class performs all the basic computations needed for gradient-driven refinement. A restraint proxy class holds all the information about the restraint that does not change during the refinement (*e.g.* the sequence ids[1] of the scatterers involved in the restraint, any target values for the restraint, the weight, *etc.*). At the highest level, there is a 'shared' proxy which is an array of proxies of a particular type. These shared proxies can then be passed to the appropriate function to calculate the residuals and gradients, and other information as and when it is required at each refinement cycle. The ADP restraints were designed in the same way as the pre-existing geometry restraints classes.

The SHELXL SIMU, ISOR and DELU instructions for restraints on anisotropic displacement parameters automatically set up the appropriate restraints for adjacent pairs of atoms (and 1,3- pairs in the case of DELU), using the atomic connectivity table or simply the proximity of a pair of atoms [SHELX manual, Sheldrick, 1997]. This can be done for all atoms in the structure, current residue, or given list of atoms. A Python class was implemented to emulate each of these SHELXL instructions and create the appropriate shared proxy arrays for each restraint type. These were tested and compared against structures refined using SHELXL to confirm that both programs setup the same restraints.

It was necessary to add the ability to create the smtbx atomic connectivity table by taking into account the covalent radii of the atoms when deciding whether any two atoms are bonded or not. Previously it was only possible to discriminate bonded from non-bonded by means of a general distance cutoff value. Functionality was also added to take into account disorder when calculating the connectivity table. *Conformer indices* (equivalent to positive values of the PART instruction in SHELXL) are used to denote that bonds should not be generated between

[1] *i.e.* the index into the array of scatterers for a given scatterer.

atoms with different conformer indices (atoms with index equal to zero belong to the major part of the structure and are bonded to atoms of all other indices that are within the bonding distance for the designated scattering types). *Symmetry exclusion indices* are used to suppress generation of bonds to symmetry equivalent atoms, such as when a molecule is disordered over a special position. Further functionality was added to allow fine-tuning of the connectivity table by manual insertion and deletion of individual bonds. The connectivity table is also essential in the initialisation of the geometrical constraints.

2.1.4 Applications

2.1.4.1 Bond similarity restraint

A crystal structure of an Iridium-containing complex contained a disordered mixture of chloroform and hexane solvates refined to an R1-factor of 2.77%. In some positions there was observed same-site disorder of the solvents. Two of these sites were modelled with a hexane and chloroform molecule sharing the same site in a 60 : 40 ratio. The bond lengths of the hexane molecule varied substantially, and a bond similarity restraint was applied. In the resulting restrained crystal structure, less variation in the hexane bond lengths was observed (see Table 2.1). Decreasing the estimated standard deviation associated with the restraint (*i.e.* increasing the weight of the restraint) resulted in the variation in bond lengths being further reduced. The following output of the program lists the deltas associated with each bond as well as the overall residual for the restraint.

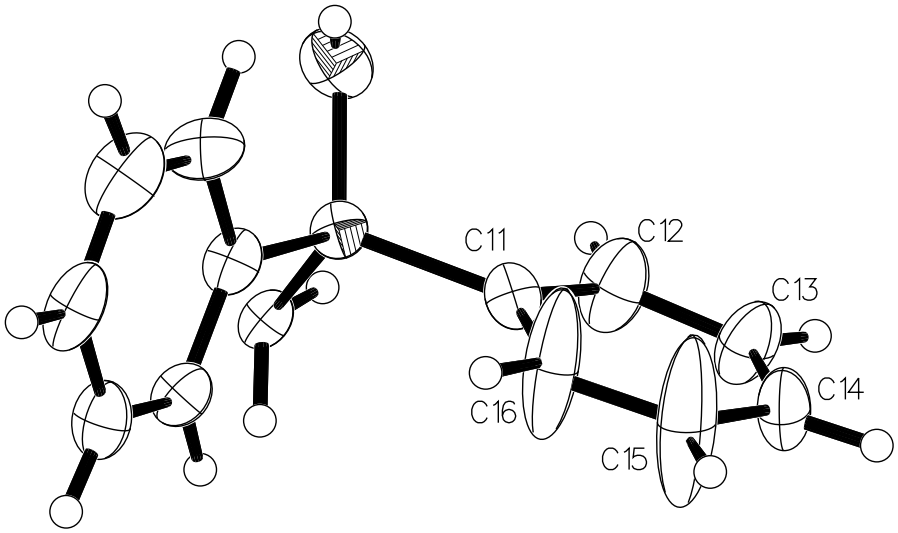
```
                delta     sigma    weight rms_deltas residual
bond C1–C2      0.020 2.00e−02 2.50e+03   3.84e−02 1.47e−03
     C2–C3      0.003 2.00e−02 2.50e+03
     C3–C4     −0.073 2.00e−02 2.50e+03
     C4–C5      0.056 2.00e−02 2.50e+03
     C5–C6     −0.005 2.00e−02 2.50e+03
     C6–C1     −0.002 2.00e−02 2.50e+03
```

Bond		Length (Å)		
		free	$\sigma = 0.02$	$\sigma = 0.01$
C1	C6	1.485(12)	1.487(10)	1.490(8)
C1	C2	1.512(9)	1.509(8)	1.503(7)
C2	C3	1.506(13)	1.492(11)	1.487(8)
C3	C4	1.371(16)	1.416(13)	1.455(9)
C4	C5	1.564(12)	1.544(11)	1.521(8)
C5	C6	1.480(12)	1.484(10)	1.489(8)

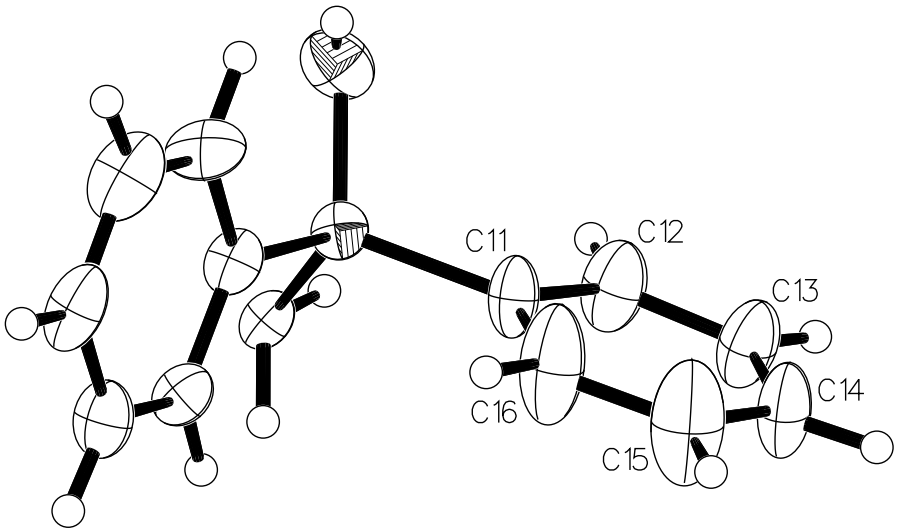
Table 2.1: The C-C bond lengths for a disordered hexane molecule modelled with and without bond similarity restraints.

2.1.4.2 ADP similarity restraints

In a crystal structure containing two phenyl rings, the ADPs of some of the carbon atoms on one of the rings were elongated in a direction perpendicular to the plane of the ring (Figure 2.2a). In this case a rigid bond restraint would have little effect, since a such a restraint only has an effect along the bond vector. ADP similarity restraints were placed upon the six carbon atoms of the phenyl ring with an estimated standard deviation of 0.01, resulting in more conventional looking ADPs (Figure 2.2b).



(a) After unrestrained refinement the ADPs of C15 and C16 are elongated in a direction perpendicular to the plane of the ring.



(b) After refinement with ADP similarity restraints there is less variation in the ADPs of the carbon atoms C11-C16.

Figure 2.2: Demonstration of the effective use of ADP similarity restraints.

2.2 Twinning

A twinned crystal consists of two or more crystals of the same species that are joined together and related by some symmetry operation. The resulting observed diffraction pattern is a superposition of the diffraction pattern of each component after application of the appropriate symmetry operation for each twin component. Problems can sometimes arise in solving structures in the presence of twinning, and it is essential to include the contribution of any twin components in the refinement of the structural parameters in order to get the best possible result.

Each twin component is defined by a rotation matrix (*twin law*) which defines the relative orientation of the twin component to the major component, and the fractional contribution of that component to the total crystal volume.

Twinned crystals can be grouped into four distinct types [Herbst-Irmer and Sheldrick, 1998]:

(a) Twinning by merohedry: The crystal posseses lower symmetry than the crystal system. The twin law belongs to the crystal system, but not to the crystal point group. As a result, the diffraction patterns from the crystal components overlap exactly, and the observed diffraction pattern may appear to have higher symmetry than is actually present. Racemic twinning, where both "hands" of a non-centrosymmetric structure are present is a special case of this subset, from which follows the definition of the Flack parameter [Flack, 1983, §3.1.4].

(b) Twinning by pseudo-merohedry: The metric symmetry is higher than the crystal system of the structure. This kind of twinning is essentially the same as for (a), except that the twin law belongs to a higher symmetry crystal system than the structure. Common examples of this type of twinning include monoclinic structures where $\beta \approxeq 90^\circ$, or $a \approxeq b$.

(c) Twinning by reticular merohedry: Similarly to types *a* and *b*, the diffraction patterns are exactly superimposed, however the symmetry is such that some of the reflections of one component overlap with the systematic absences of the others and *vice versa*. As a result, it may be possible to attempt structure solution using those reflections that contain a contribution from one component only. For examples of the treatment of such twins, see Herbst-Irmer and Sheldrick [2002].

(d) Non-merohderal twinning: The previous types of twinning all require that

the symmetry operator belongs to some crystallographic point group, and can be indexed on a single lattice. In contrast, the components of a non-merohedral twin are related by some arbitrary operator, and each component is indexed on a different lattice with a different orientation matrix. Some reflections may happen to overlap exactly, or be otherwise indistinguishable, while the majority of reflections can be identified as belonging entirely to one twin component. This type of twinning is observable directly in the diffraction pattern and can lead to problems with unit cell determination and indexing, however diffractometer software is becoming increasingly sophisticated in dealing with non-merohedral twinning.

For the first three cases outlined above, where the reciprocal lattices are exactly superimposed, the observed diffracted intensity can be given as the sum over the intensities for all miller indices that contribute to a particular point in the diffraction pattern:

$$F_o^2 = \sum_i^n \alpha_i F_{o_i}^2, \tag{2.42}$$

where α_i is the fractional contribution of twin component i to the crystal. Since the sum over all the fractional contribution must be equal to one, $n-1$ of them can be refined, whereas the last one is expressed as a function of those $n-1$ independent parameters,

$$\alpha_n = 1 - \sum_i^{n-1} \alpha. \tag{2.43}$$

For certain applications it may be necessary to obtain a set of observations that contain only the contribution from the major component. This is essential when calculating an electron density map, and may occasionally be necessary in order to solve a structure successfully. In addition, many early twinned structures were refined against such *detwinned* datasets [Britton, 1972; Grainger, 1969; Murray-Rust, 1973].

In the simplified case of hemihedral twinning, two reflections combine in the following way

$$I_1 = (1-\alpha)J_1 + \alpha J_2 \tag{2.44}$$

$$I_2 = \alpha J_1 + (1-\alpha)J_2, \tag{2.45}$$

where I_1 and I_2 are the observed intensities produced by the superposition of the untwinned intensities, J_1 and J_2 with twin fraction α.

This can be solved algebraically [Britton, 1972; Grainger, 1969; Zachariasen, 1965] to give

$$J_1 = I_1 + \frac{\alpha}{1-2\alpha}(I_1 - I_2) \tag{2.46}$$

$$J_2 = I_2 - \frac{\alpha}{1-2\alpha}(I_1 - I_2). \tag{2.47}$$

These equations become singular as the value of α approaches 0.5, however it is possible to *detwin* the data using the proportionality of related intensities as calculated from the model

$$J_1 = I_1 \frac{F_a^2(1-\alpha)}{F_a^2(1-\alpha) + F_b^2\alpha} + I_2 \frac{F_a^2\alpha}{F_a^2\alpha + F_b^2(1-\alpha)} \tag{2.48}$$

where F_a^2 and F_b^2 are the calculated intensities of reflections related by the twin law. This method has the drawback of being more biased towards the model, and it may be better to use the algebraic method if possible.

Alternatively the data can be reduced to the 'prime' twin component by

$$J_1 = I_1 \frac{F_a^2(1-\alpha)}{F_a^2(1-\alpha) + F_b^2\alpha} \tag{2.49}$$

which is the equation used for Fourier map calculations for twinned structures in JANA [JANA98 manual, Dušek et al., 2001; Petříček and Dušek, 2000] and SHELXL [SHELX manual, Sheldrick, 1997]. This formula is more trivially extended to multiply twinned crystals.

Several methods have been described for estimating the twin fraction based purely on the statistics of the observed intensities [Britton, 1972; Murray-Rust, 1973]. This approach is impossible as the value of α approaches 0.5, since the separation of intensities in that case relies on equation 2.48 and the calculated intensities are not known in the absence of a structural model. In addition, covariance of the twin fraction with any other least squares parameters is ignored.

Most commonly used crystallographic refinement software [CRYSTALS, SHELXL, *etc.* Betteridge et al., 2003; Sheldrick, 2008] use the twin refinement method of

Jameson [1982] and Pratt et al. [1971], where the original, unaltered, observed intensities are used, whilst the F_c^2 are calculated according to equation 2.42. It is this method of twin refinement that has been implemented within smtbx-refine.

The derivatives of the squared structure factors with respect to the model parameters are calculated as

$$\frac{\partial F_c^2}{\partial p_j} = \left(1 - \sum_i^{n-1} \alpha_i\right) \frac{\partial F_{c_n}^2}{\partial p_j} + \sum_i^{n-1} \alpha_i \frac{\partial F_{c_i}^2}{\partial p_j}, \tag{2.50}$$

and the derivatives with respect to the twin fractions, α_i, given by

$$\frac{\partial F_c^2}{\partial \alpha_i} = F_{c_i}^2 - F_{c_n}^2. \tag{2.51}$$

Figure 2.3 outlines the general steps involved in building the normal equations, with the inclusion of twinning.

2.2.1 Testing

As part of the regression test cases that are standard procedure in the cctbx, a simple test case was created from the coordinates of a known small structure (11 atoms, hall symbol P 3 -2c). Synthetic intensities were created based on the existing crystal structure and scaled by a random scale factor, and using unit weights. A twinned dataset was then computed using the pre-existing cctbx hemihedral twinning/detwinning tools [Zwart et al., 2005], using a random twin fraction and the twin law $k, h, -l$. The atomic coordinates and ADPs were shaken with random displacements and a shift of ± 0.1 was applied to the 'true' twin fraction to provide starting values for the refinement.

After refinement with a maximum of 10 cycles, it was confirmed that the twin fractions had successfully refined to the original randomly generated values and that the final least squares objective was equal to zero.

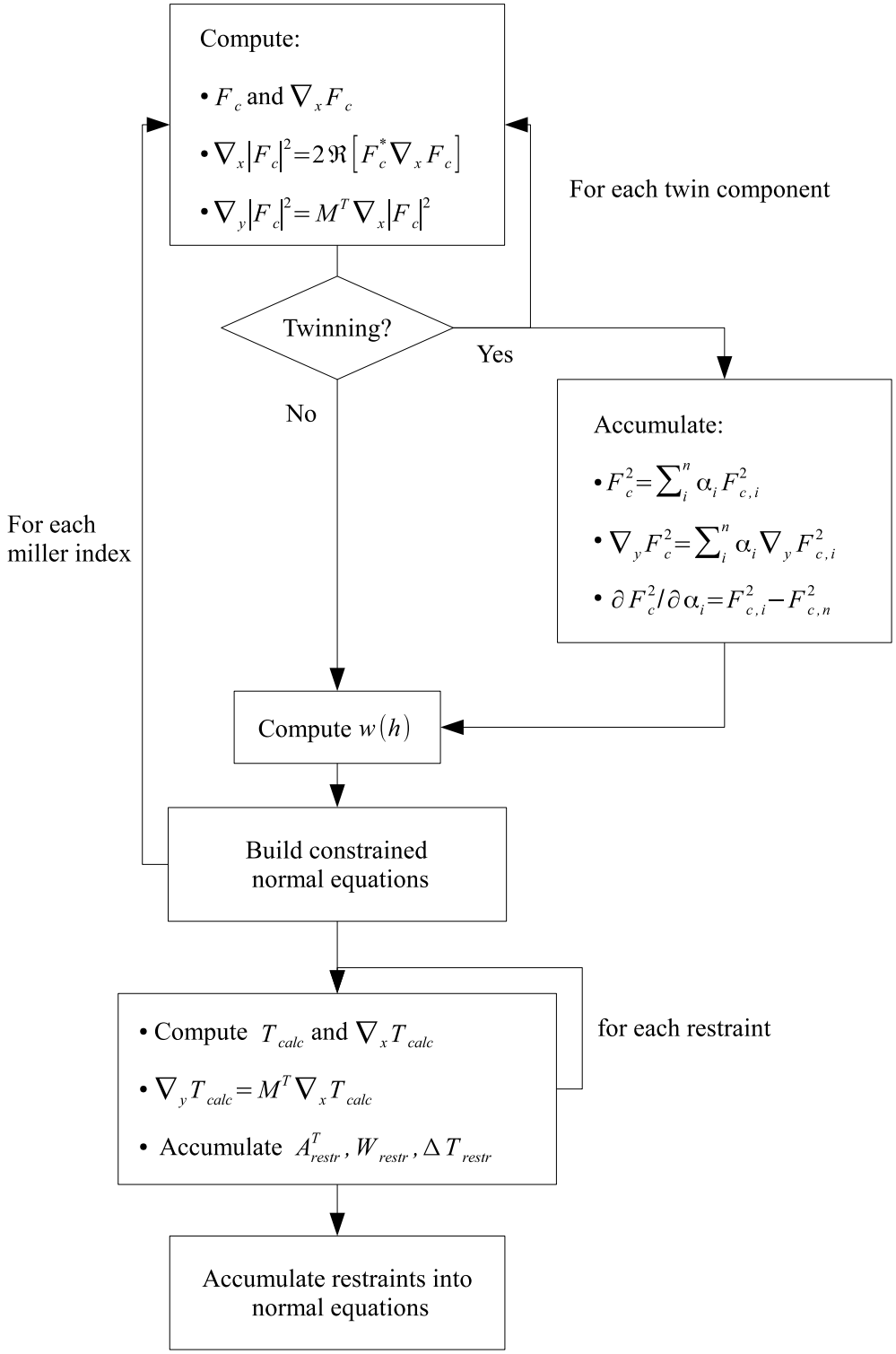


Figure 2.3: Flow diagram illustrating the steps taken when building up the normal equations taking twinning into account.

2.3 Errors on derived parameters

For a function, f, of a set of atomic parameters, p_i, its variance is given by [Sands, 1966]

$$\sigma^2(f) = \sum_{i,j} \left(\frac{\partial f}{\partial p_i}\right) \left(\frac{\partial f}{\partial p_j}\right) cov(p_i, p_j) \tag{2.52}$$

Derived parameters such as bond lengths and angles are a function of both the least squares atomic parameters and the unit cell parameters. As such, the error in a derived parameter is likewise a function of both the atomic and unit cell parameters. If the errors in atomic parameters are considered to be totally uncorrelated with the errors in the cell parameters (*i.e.* their covariance is zero), then the error in a derived parameter can be considered as comprising two independent sources of errors:

$$\sigma^2(f) = \sigma^2_{cell}(f) + \sigma^2_{xyz}(f), \tag{2.53}$$

where $\sigma_{xyz}(f)$ is the part coming from the errors in the least square estimates of the positional parameters, and $\sigma_{cell}(f)$ comes from the errors in the unit cell parameters,

$$\sigma^2_{cell}(f) = \sum_{i,j} \frac{\partial f}{\partial i} \frac{\partial f}{\partial j} cov\,(i, j), \tag{2.54}$$

where $i, j = \{a, b, c, \alpha, \beta, \gamma\}$.

This necessitates the calculation of the derivatives of the function with respect to the unit cell parameters. In order to do so, it is easier to calculate separately the derivative of the function with respect to the elements of the metrical matrix, and also the derivative of the metrical matrix with respect to the cell parameters. The former must be evaluated for every function, whereas the latter is constant for a given unit cell.

$$\frac{\partial f}{\partial i} = \frac{\partial f}{\partial g_{jk}} \frac{\partial g_{jk}}{\partial i}, \; i = a, b, c, \alpha, \beta, \gamma \tag{2.55}$$

Now we consider the application of equation 2.52 to determine the estimated error in the length of the vector $\mathbf{u}$, in fractional coordinates. The length, D, of

the vector $\mathbf{u}$ is given by

$$D = (\mathbf{u}^T \mathbf{G} \mathbf{u})^{\frac{1}{2}}, \tag{2.56}$$

where $\mathbf{G}$ is the metrical matrix.

The derivative of the distance, D, with respect to the elements of the metrical matrix, $\mathbf{G}$, is given by

$$\frac{\partial D}{\partial g_{ii}} = \frac{1}{2} \frac{u_i^2}{D} \tag{2.57}$$

and (given the metrical matrix is symmetric)

$$\frac{\partial D}{\partial g_{ij}} = \frac{u_i u_j}{D}, \text{ for all } i < j. \tag{2.58}$$

Similarly, for the angle between two vectors in fractional coordinates, $\mathbf{u}$ and $\mathbf{v}$, where the angle is defined as

$$\theta = \arccos \frac{\mathbf{u}^T \mathbf{G} \mathbf{v}}{\|\mathbf{u}^T \mathbf{G} \mathbf{u}\| \|\mathbf{v}^T \mathbf{G} \mathbf{v}\|} \tag{2.59}$$

or

$$\theta = \arccos \frac{\mathbf{r_A} \cdot \mathbf{r_B}}{\|\mathbf{r_A}\| \|\mathbf{r_B}\|}, \tag{2.60}$$

where $\mathbf{r_A}$ and $\mathbf{r_B}$ are the Cartesian equivalents of $\mathbf{u}$ and $\mathbf{v}$. The derivative of the angle, θ, with respect to the elements of the metrical matrix, $\mathbf{G}$, is given by

$$\frac{\partial \theta}{\partial g_{ii}} = \frac{1}{2 \sin \theta} \left(\frac{u_i^2 \cos \theta}{\|\mathbf{r_A}\|^2} - \frac{2 u_i v_i}{\|\mathbf{r_A}\| \|\mathbf{r_B}\|} + \frac{v_i^2 \cos \theta}{\|\mathbf{r_B}\|^2} \right) \tag{2.61}$$

and

$$\frac{\partial \theta}{\partial g_{ij}} = \frac{1}{\sin \theta} \left(\frac{u_i u_j \cos \theta}{\|\mathbf{r_A}\|^2} - \frac{u_i v_j + u_j v_i}{\|\mathbf{r_A}\| \|\mathbf{r_B}\|} + \frac{v_i v_j \cos \theta}{\|\mathbf{r_B}\|^2} \right), \text{ for all } i < j. \tag{2.62}$$

The derivative of the metrical matrix with respect to the unit cell parameters,

needed in order to apply equation 2.55, are given below:

$$\begin{aligned}
\frac{\partial g_{11}}{\partial cell} &= (2a, 0, 0, 0, 0, 0) \\
\frac{\partial g_{22}}{\partial cell} &= (0, 2b, 0, 0, 0, 0) \\
\frac{\partial g_{33}}{\partial cell} &= (0, 0, 2c, 0, 0, 0) \\
\frac{\partial g_{12}}{\partial cell} &= (b\cos\gamma, a\cos\gamma, 0, 0, 0, -ab\sin\gamma) \\
\frac{\partial g_{13}}{\partial cell} &= (c\cos\beta, 0, a\cos\beta, 0, -ac\sin\beta, 0) \\
\frac{\partial g_{23}}{\partial cell} &= (0, c\cos\alpha, b\cos\alpha, -ac\sin\beta, 0, 0)
\end{aligned} \tag{2.63}$$

2.3.1 Symmetry

The variance-covariance matrix that is obtained from the inversion of the least squares normal matrix contains the variance and covariance of all the refined parameters. Frequently, it is necessary to compute functions that involve parameters that are related by some symmetry operator of the space group to the original parameters. Sands [1966] suggests that the symmetry should be applied to the variance-covariance matrix to obtain a new variance-covariance matrix for the symmetry generated atoms. Alternatively, and it is this method that is used here, the original variance-covariance matrix can be used if the derivatives in 2.52 are mapped back to the original parameters.

Let the function f depend on the Cartesian site y_c that is generated by the symmetry operator $\mathbf{R}_c$ from the original Cartesian site x_c, *i.e.*

$$\begin{aligned}
y_c &= \mathbf{R}_c x_c \\
&= \mathbf{O}\mathbf{R}_f\mathbf{F}x_c,
\end{aligned} \tag{2.64}$$

where $\mathbf{F}$ and $\mathbf{O}$ are the fractionalisation and orthogonalisation matrices respectively, with $\mathbf{R}_c$ and $\mathbf{R}_f$ the symmetry operator in Cartesian and fractional coordinates respectively.

Then the gradient with respect to the original site can be obtained by

$$\begin{aligned}\nabla_{x_c} f(y_c) &= \mathbf{R}_c^T \nabla_{y_c} f(y_c) \\ &= \mathbf{O}^{-T} \mathbf{R}_f^{-1} \mathbf{O}^T \nabla_{y_c} f(y_c).\end{aligned} \tag{2.65}$$

The variance-covariance matrix that is used in this case should be the one that is transformed to Cartesian coordinates. The variance-covariance matrix for Cartesian coordinates can be obtained from that for fractional coordinates by the transformation

$$\mathbf{V}_c = \mathbf{O}\mathbf{V}_f\mathbf{O}^T, \tag{2.66}$$

where $\mathbf{O}$ is the orthogonalisation matrix, such that

$$\mathbf{x}_c = \mathbf{O}\mathbf{x}_f \tag{2.67}$$

The transformation matrix needed to transform the entire variance-covariance matrix in one operation would be block diagonal, with the 3×3 orthogonalisation matrix, O, repeated at the appropriate positions along the diagonal. This transformation can be computed efficiently using sparse matrix techniques.

2.3.2 Discussion

There have been recent attempts in the literature to absorb the errors in the unit cell parameters into the covariance matrix [Haestier, 2009; Schwarzenbach, 2010]. Methods have been developed which are capable of absorbing into the covariance matrix the errors in the unit cell lengths a, b, c, however complications arise for atoms related by symmetry operations involving translations, so the advantage of this method is unclear. Schwarzenbach [2010] showed that a similar scheme for the standard uncertanties in the unit cell angles α, β, γ is not possible. Furthermore, Schwarzenbach [2010] concludes *the safest course remains to explicitly calculate all derivatives and, since computer time has become cheap, this is also the method to be preferred.* It is the author's opinion that given that the derived parameters of interest are relatively few and the availability of computer algebra tools such as

Mathematica [Wolfram Research, Inc., 2010], it is not particularly onerous to code the required derivatives explicitly for each of the functions of interest.

Dolomanov et al. [2009a] have found that the use of numerical differentiation techniques, as implemented in the Olex2 software, give similar results to using analytical techniques, without the need for calculation of explicit derivatives, with no significant penalty in computing time.

Chapter 3

Reflection Statistics

3.1 Absolute Structure

3.1.1 Anomalous Scattering

Friedel's law [Friedel, 1913] states that for a reflection, hkl, its intensity will be equal to the reflection related by inversion, $\bar{h}\bar{k}\bar{l}$. This is a direct result of the Fourier transform of a real function:

$$F(\mathbf{h}) = \int_{-\infty}^{\infty} f(x) \exp(-i\mathbf{h} \cdot x) dx. \tag{3.1}$$

If $f(x)$ is real, then:

$$F(\mathbf{h}) = F^*(-\mathbf{h}), \tag{3.2}$$

where $\mathbf{h}$ and $-\mathbf{h}$ (or in alternative notation, hkl and $\bar{h}\bar{k}\bar{l}$) are termed *Friedel pairs*. The observed intensity is proportional to the square of the amplitude and, as a result, is centrosymmetric:

$$|F(\mathbf{h})|^2 = |F(-\mathbf{h})|^2 . \tag{3.3}$$

The phases of the two inversion-related reflections are equal in magnitude but opposite in sign:

$$F(\mathbf{h}) = |F(\mathbf{h})| \exp(i\theta_h) \tag{3.4}$$

$$F(-\mathbf{h}) = |F(\mathbf{h})| \exp(-i\theta_h) \tag{3.5}$$

An important consequence of the strict application of Friedel's law is that the diffraction pattern is centrosymmetric regardless of whether the crystal symmetry is centrosymmetric or not. This means that it is impossible to distinguish a non-centrosymmetric crystal structure from its inversion-related image if the atomic scattering factor, f_j, is real. Fortunately, in reality, this is only approximately true and the atomic scattering factor usually contains a real and imaginary *anomalous* (or *resonant*) scattering contribution that is a result of absorption in the scattering of photons by electrons (inelastic scattering):

$$f_j = f_0 + f' + if'' \tag{3.6}$$

This phenomenon causes small deviations from Friedel's law; these differences are commonly referred to as *Bijvoet differences*. Unlike the term coming from elastic scattering, the inelastic term is wavelength, as well as element, dependent. In general, the effect increases with both atomic number and wavelength, although the largest effect is observed close to an absorption edge, which can be obtained with tuneable radiation, such as that found at synchrotrons. It is these small differences in intensities of inversion-related reflections that have led to numerous techniques for distinguishing non-centrosymmetric crystal structures from their inversion-related images.

The first demonstration of the inversion-distinguishing power of anomalous scattering with X-ray diffraction by Coster et al. [1930] was followed by the first recorded absolute-configuration determination of an organic compound by Bijvoet et al. [1951]. Using Zr $K\alpha$ radiation close to the K-absorbtion edge of rubidium, they observed differences in the intensities of reflections related by Friedel's law. From analysis of these differences ("Bijvoet differences") they were able to confirm the absolute configuration of (+)-tartaric acid. Lutz and Schreurs [2008]

recently asked the question "Was Bijvoet right?" when they revisited the absolute-configuration determination of sodium rubidium (+)-tartrate tetrahydrate using modern equipment and up-to-date techniques. Their answer: *an unequivocal 'yes'*.

3.1.2 Hamilton's Ratio Test

Hamilton [1965] advocated the application of his R-factor ratio test for the determination of absolute structure. He suggested using the ratio, R^-/R^+, of the R-factors calculated using the inverted coordinates, $-\mathbf{x}$ and the refined coordinates, $+\mathbf{x}$. Alternatively, the same effect can be obtained by reversing the signs of the if_j'' and keeping the coordinates intact. In the presence of anomalous scattering different values should be obtained for R^+ and R^-, and Hamilton's ratio test could be used to determine whether the difference in the R-factors is significant and the absolute structure can be reliably determined.

3.1.3 Rogers η Parameter

Rogers [1981] highlighted numerous potential difficulties with Hamilton's method, as well as providing examples of misunderstandings and abuses of the method. Problems include overestimation of the probability of correct assignment caused by selective application of dispersion corrections only for the atoms with the strongest anomalous scattering, statistically illusory or even suspect enhanced ratios obtained from comparison of two dispersion-refined models, and difficulties in correctly estimating the correct value for N, the number of degrees of freedom.

As a result, he introduced a parameter, η, to be refined along with the rest of the least squares parameters, a precision for which can be readily computed. The variable η is introduced as a multiplicative factor into the imaginary anomalous dispersion terms to give $i\eta f_j''$. Refinement of η should give values that converge close to $+1$, indicating a correct assignment of absolute structure, or to -1, implying that inversion of the structure is necessary.

3.1.4 Flack x Parameter

Flack [1983] showed the Rogers η parameter to be inadequate under certain condi-

tions in that the value of η determined in a least squares refinement would depend on its starting value. In addition, for structures that are nearly centrosymmetric, the η parameter can give over-precise estimates of the absolute structure. He suggested a new least squares parameter, x, which addressed these faults and converges more rapidly than η. The definition of the x parameter is based on anomalous scattering from twin components related by a centre of inversion (see §2.2 for further details on refinement of twins):

$$|F(\mathbf{h}, x)|^2 = (1 - x)\,|F(\mathbf{h})|^2 + x\,|F(-\mathbf{h})|^2\,. \tag{3.7}$$

With the correct absolute structure, the parameter refines to a value of 0, whereas a value of 1 indicates incorrect assignment of absolute structure. This definition of the parameter allows for the possibility of an inversion twin fraction of anywhere in the range $0 - 100\%$, where the crystal contains $100(1 - x)\%$ of the component whose coordinates are refined in the least squares procedure and $100x\%$ of its image by inversion. The faster convergence of the x parameter is due to x being a *linear* function and η a *quadratic* function of $|F|^2$. With its implementation in (amongst others) the widely used SHELXL refinement program [Sheldrick, 2008], the Flack x parameter has since become the $de facto$ method of absolute structure determination.

Flack and Bernardinelli [2000] published some guidelines on interpreting the Flack x parameter and its associated standard uncertainty u. Under the assumption that the errors are drawn from a Gaussian distribution (for remarks on whether this is in fact always the case, see §3.1.5), for reliable assignment of the absolute structure they require that the value of the Flack x parameter is within three standard deviations of zero. Of equal importance is the size of the standard uncertainty: in the general case, they require that $u < 0.04$; in the event that the formation of inversion twins can be discounted (such as in the crystallisation of an entantiopure compound) then this requirement can be relaxed to $u < 0.1$.

In addition, whilst the Flack x parameter can be calculated outside of a full matrix least squares refinement, this can lead to inaccurate values of x if it deviates significantly from zero and an underestimation of its uncertainty by a factor of up to 3. As a consequence, they recommend that the published Flack x parameter

should always be obtained *via* full matrix least squares refinement where x is varied along with all other parameters.

Parsons and Flack [2004] recently proposed a method of obtaining improved estimates of the Flack x parameter by careful measurement of selected pairs of Friedel opposites in such a way that the systematic errors are the same for both measurements. The ratios

$$D_{\text{obs}} = \frac{I(\mathbf{h}) - I(-\mathbf{h})}{I(\mathbf{h}) + I(-\mathbf{h})} \cong (1 - 2x)\frac{|F(\mathbf{h})|^2 - |F(-\mathbf{h})|^2}{|F(\mathbf{h})|^2 + |F(-\mathbf{h})|^2} \tag{3.8}$$

are then used as additional observations of restraints in a conventional least squares refinement. They found this led to improvements of up to a factor of 3 in the precision of the absolute-structure determination.

Dittrich et al. [2006b] demonstrated that improvements in both the value and standard uncertainty of the Flack x parameter could be obtained with the use of ashperical scattering factors, or 'invarioms', instead of normal spherical scattering factors.

3.1.5 Hooft y Parameter

Hooft et al. [2008] introduced a new probabilistic approach to absolute-structure determination based on intensity differences between Bijvoet pairs. For each Bijvoet pair of reflections, $\mathbf{h}$ and $-\mathbf{h}$, we can define the *Bijvoet differences* $\Delta_o(\mathbf{h}) = |F_o(\mathbf{h})|^2 - |F_o(-\mathbf{h})|^2$ and similarly $\Delta_c(\mathbf{h}) = |F_c(\mathbf{h})|^2 - |F_c(-\mathbf{h})|^2$. If the coordinates of the refined structure are of the correct hand, then the signs of each observed and calculated Bijvoet difference should be matching. Conversely, if the wrong hand was used in the refinement, then the signs would be opposite. This can be generalised to allow for the possibility of twinning by inversion by replacing the change of sign with a continously variable parameter, γ.

$$x_h(\gamma) = \frac{\gamma\Delta_c(\mathbf{h}) - \Delta_o(\mathbf{h})}{\sigma\left(\Delta_o(\mathbf{h})\right)} \tag{3.9}$$

If the variable $x_h(\gamma)$ follows a Gaussian distribution, then

$$p(x_h(\gamma)) = \frac{1}{(2\pi)^{1/2}} \exp(-x_h(\gamma)^2) \tag{3.10}$$

We can calculate the probability of observing the measured data given γ:

$$p\left(\text{observations} \mid \gamma\right) = \prod_h p(x_h(\gamma)) \tag{3.11}$$

For numerical stability, we will calculate $\log(p)$ and hence:

$$\log p(\text{observations} \mid \gamma) \simeq -\frac{1}{2} \sum_h x_h(\gamma)^2 \tag{3.12}$$

From Bayes' theorem for probability densities, the posterior probability density function for γ given the observations is

$$p(\gamma \mid \text{observations}) = \frac{p(\text{observations} \mid \gamma)p(\gamma)}{\int_{-\infty}^{\infty} p(\text{observations} \mid \gamma)p(\gamma)d\gamma}. \tag{3.13}$$

Since the probability density $p(\gamma)$ is unknown, Hooft et al. [2008] propose to use a uniform probability density for γ, however a uniform probability is only defined for a finite interval. We note that, both in theory and in practice, large positive or negative values of γ are unrealistic and therefore propose to restrict γ to a more realistic interval, $-\Gamma \leq \gamma \leq \Gamma$. Equation 3.13 can be given as

$$p(\gamma \mid \text{observations}) = \frac{p(\text{observations} \mid \gamma)}{\int_{-\Gamma}^{+\Gamma} p(\text{observations} \mid \gamma)d\gamma} \tag{3.14}$$

where the mean and variance of γ are given by

$$G = \langle\gamma\rangle = \frac{\int_{-\Gamma}^{+\Gamma} \gamma p(\text{observations} \mid \gamma)d\gamma}{\int_{-\Gamma}^{+\Gamma} p(\text{observations} \mid \gamma)d\gamma}, \tag{3.15}$$

$$\sigma(G)^2 = \operatorname{var}\gamma = \frac{\int_{-\Gamma}^{+\Gamma} (\gamma - G)^2 p(\text{observations} \mid \gamma)d\gamma}{\int_{-\Gamma}^{+\Gamma} p(\text{observations} \mid \gamma)d\gamma} \tag{3.16}$$

Since $p(\text{observations} \mid \gamma)$ is a rapidly falling normal distribution, the denominator is approximately equal to the integral between $-\infty$ and $+\infty$ and we can safely use $\Gamma = \infty$ in the above equations, giving

$$G = \frac{\int_{-\infty}^{\infty} \gamma p(\text{observations} \mid \gamma) d\gamma}{\int_{-\infty}^{\infty} p(\text{observations} \mid \gamma) d\gamma}, \tag{3.17}$$

$$\sigma(G)^2 = \frac{\int_{-\infty}^{\infty} (\gamma - G)^2 p(\text{observations} \mid \gamma) d\gamma}{\int_{-\infty}^{\infty} p(\text{observations} \mid \gamma) d\gamma}. \tag{3.18}$$

The calculated values of $\log p(\text{observations} \mid \gamma)$ are usually very small and therefore we use instead the probability density function

$$p_u(\gamma) = \exp\left(\log p(\text{observations} \mid \gamma) - \log p(\text{observations} \mid \gamma_0)\right), \tag{3.19}$$

where $\log p(\text{observations} \mid \gamma_0)$ is a large value of the probability density function given in equation 3.12. This then results in equations (23) and (24) of Hooft et al. [2008]

$$G = \frac{\int_{-\infty}^{\infty} \gamma p_u(\gamma) d\gamma}{\int_{-\infty}^{\infty} p_u(\gamma) d\gamma} \tag{3.20}$$

and

$$\sigma^2(G) = \frac{\int_{-\infty}^{\infty} (\gamma - G)^2 p_u(\gamma) d\gamma}{\int_{-\infty}^{\infty} p_u(\gamma) d\gamma}. \tag{3.21}$$

As suggested by Hooft et al. [2008], the values G and $\sigma^2(G)$ can be computed by numerical integration within suitable bounds. However, by introducing

$$A = \sum_h \frac{\Delta_c(h)^2}{\sigma^2_{\Delta_o(h)}} \qquad B = \sum_h \frac{\Delta_c(h)\Delta_o(h)}{\sigma^2_{\Delta_o(h)}} \qquad C = \sum_h \frac{\Delta_o(h)^2}{\sigma^2_{\Delta_o(h)}} \tag{3.22}$$

equation 3.12 can be rewritten as

$$\log p(\text{observations} \mid \gamma) \simeq -\frac{1}{2} A \left(\gamma - \frac{B}{A}\right)^2 + \frac{1}{2} C - \frac{B^2}{A} \tag{3.23}$$

The terms not involving γ will appear in all calculated values of $\log p(\text{observations} \mid \gamma)$ and hence will cancel, meaning that equation 3.21 can be now be written

$$G = \frac{\int_{-\infty}^{\infty} \gamma q(\gamma) d\gamma}{\int_{-\infty}^{\infty} q(\gamma) d\gamma}, \tag{3.24}$$

where

$$q(\gamma) = \exp\left(-\frac{1}{2}A\left(\gamma - \frac{B}{A}\right)^2\right). \tag{3.25}$$

It is clear that $q(\gamma)$ follows a normal distribution with $\mu = \frac{B}{A}$ and $\sigma = A^{-\frac{1}{2}}$. Therefore G and $\sigma(G)$ are equal to μ and σ respectively and can be calculated directly without computing the full probability distribution $p_u(\gamma)$.

The value of the parameter G behaves in much the same way as the Rogers η parameter, in that a value close to 1 indicates correct absolute structure assignment whilst a value close to -1 indicates that inversion is necessary. A simple change of variable results in a new y parameter which is comparable with the Flack x parameter:

$$y = (1 - G)/2 \tag{3.26}$$

and

$$\sigma_y = \sigma_G/2 \tag{3.27}$$

Hooft et al. [2008] use the version of Bayes' theorem for probabilities to give

$$p(\gamma_i \mid \text{observations}) = \frac{p(\text{observations} \mid \gamma_i)p(\gamma_i)}{\sum_j p(\text{observations} \mid \gamma_j)p(\gamma_j)} \tag{3.28}$$

for the discrete set of values, $\gamma_1, \gamma_2, ..., \gamma_n$. Similarly to the case of a continuous distribution, they suggest the use of a discrete uniform distribution for $p(\gamma_j)$, i.e. $p(\gamma_j) = 1/n$. In this form, two sets of probabilities are calculated. The first, $p2$(true), is the probability for a two-hypothesis model: the sample can assumed to be enantiopure and hence the absolute structure is either right or wrong. Using a three-hypothesis model (additionally allowing for the possibility of a 50% inversion twin), the three probabilities $p3$(true), $p3$(twin) and $p3$(false) are calculated. It should be noted that Bayes' theorem as employed in equation 3.28 only

strictly applies to probabilities, yet here it is applied to the probability density $p(\text{observations} \mid \gamma_i)$.

3.1.5.1 Treatment of Outliers

The implementation of the procedure that is available within the current version of the software PLATON [Spek, 2003] uses an outlier cutoff that rejects observed Bijvoet differences that are significantly larger than the maximum calculated differences. With the default cutoff factor of $k = 2$, only the reflections where $|\Delta_o(\mathbf{h})| < k \times \max(|\Delta_c(\mathbf{h})|)$ are used in the calculation. When the inversion-distinguishing power is weak, the standard deviations for some of the Bijvoet differences may be of the same order of magnitude as the differences themselves. In such a case, a large percentage of the Bijvoet pairs may be rejected using such an arbitrary outlier cutoff, which may in turn significantly skew the result of the analysis. Figure 3.1 shows how the probability distribution is shifted towards $\gamma = 0$ when a significant number of pairs are rejected using an outlier cutoff. This can be understood by considering the relation $G = \frac{B}{A}$. Under the assumptions that for the rejected Bijvoet pairs $|\Delta_o(\mathbf{h})|$ are larger than $|\Delta_c(\mathbf{h})|$ and that the structure is of the correct hand (*i.e.* the observed and calculated differences have the same sign), then G will tend towards zero since B will approach zero at a faster rate than A.

3.1.5.2 Probability plots

The use of probability plots as a method of assessing errors in crystallography was first suggested by Abrahams and Keve [1971]. A plot of the ordered statistic $x_h(\gamma = 1)$ against the ordered theoretical quantiles of the normal distribution can be used to verify that the errors in the Bijvoet differences do indeed follow a normal distribution. A plot that deviates significantly from linearity indicates that the errors do not follow a normal distribution, whilst a slope for the least squares line of best fit that departs from unity can indicate a misestimation of the assigned standard deviations of the data.

In practice, it is frequently observed that the observations do not closely follow a normal distribution, with values with high deviations being observed with much

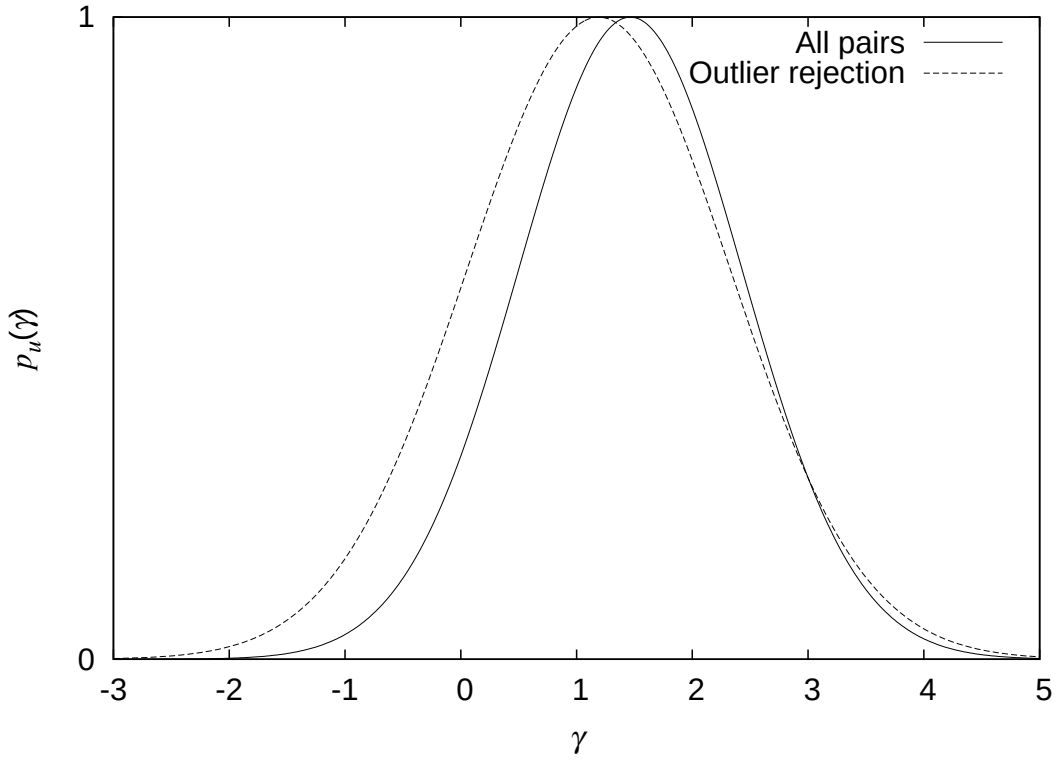


Figure 3.1: The probability density function of $p_u(\gamma)$ with and without rejection of 348 (23%) Bijvoet differences outliers. With rejection of outliers the probability density is shifted towards $\gamma = 0$, giving $G = 1.2(11)$ compared to $G = 1.5(10)$ without such outlier rejection. The probability plot slopes were 0.544 and 0.754 respectively.

higher frequency than would be expected [Hooft et al., 2009]. Figure 3.2a shows the normal probability plot [Abrahams and Keve, 1971] of the Bijvoet differences in such a case. Large tails are observed in the distribution and the least squares line of best fit has a slope significantly above 1.0 and a poor linear correlation coefficient is obtained for the probability plot.

As described by Hooft et al. [2010], a more robust approach may be to use a Student's t-distribution [Hooft et al., 2009; Student, 1908] of the errors. Figure 3.2b shows that the same data more closely fits a Student's t-distribution with $\nu = 2.5$.

3.1.5.3 Student's *t*-distribution

The Student's t-distribution is a continuous probability distribution that, similarly to the normal distribution, is symmetric and bell-shaped, but has larger tails. The distribution has one parameter, ν, which is often referred to as the degrees of freedom, that can be used to control the shape of the distribution. As ν approaches zero the tails of the probability density function become increasingly pronounced. At the limit of $\nu = \infty$ the distribution is indistinguishable from the normal distribution.

The value of ν for the Student's t-distribution is chosen as the one which maximises the linear correlation coefficient of the probability plot [Hooft et al., 2009].

To determine the absolute structure using a Student's t-distribution, equations 3.10 and 3.12 can be replaced by

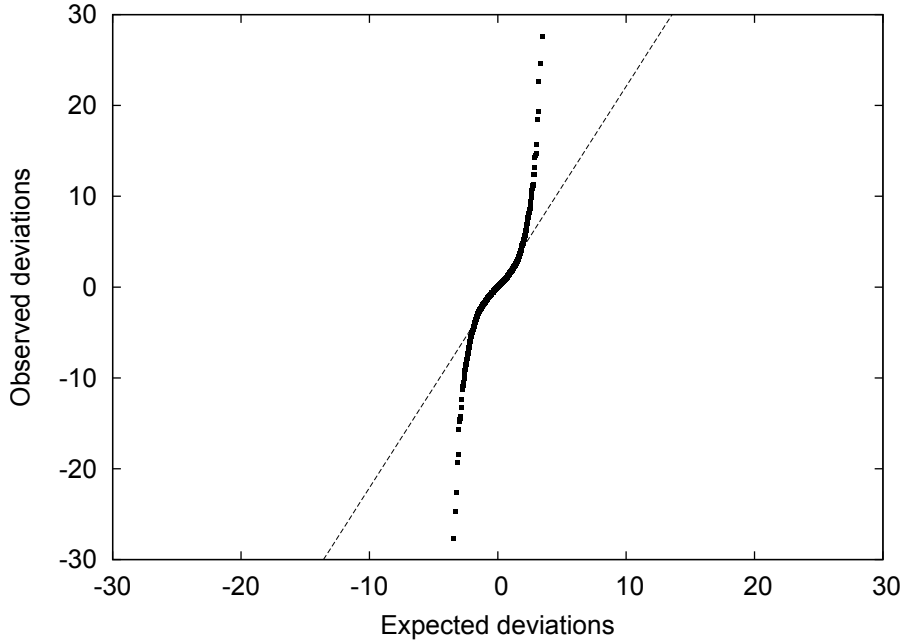
$$p(x_h, \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{(\nu\pi)^{1/2}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x_h^2}{\nu}\right)^{(\nu+1)/2} \tag{3.29}$$

and

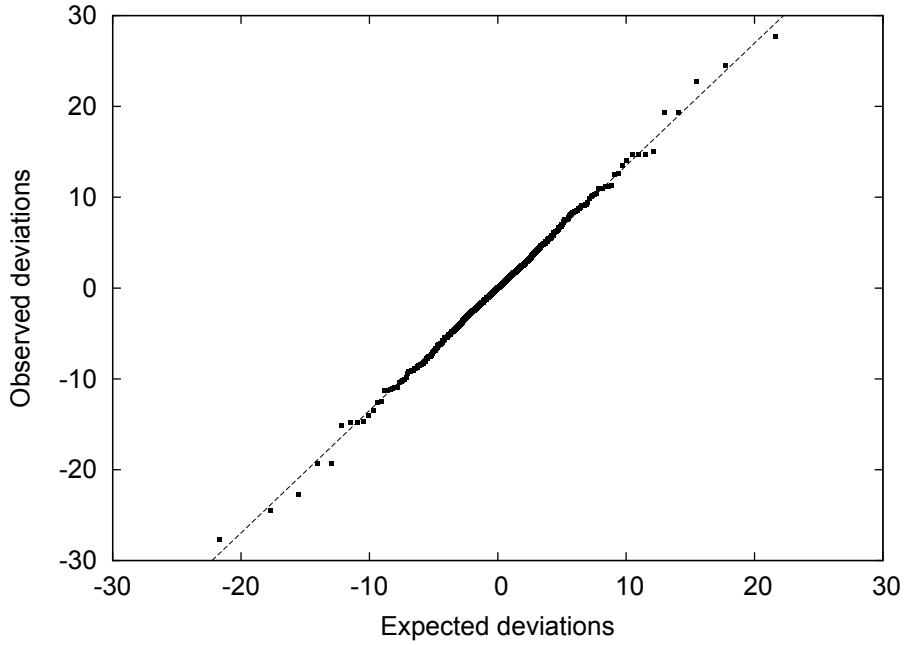
$$\log p(\text{observations} \mid \gamma) \simeq -\frac{(1+\nu)}{2} \sum_h \log\left(x_h^2 + \nu\right) \tag{3.30}$$

respectively.

Figure 3.3 shows the distribution of equation 3.19 for two structures, one where the absolute structure is well defined ($G = 1.6(7)$) and one where it is not ($G = 1.02(2)$). As can be seen clearly from the Figure, the expected value, G, is a



(a) A normal probability plot of the Bijvoet differences showing significant deviations from a normal distribution of the errors. The least squares line of best fit has a slope of 2.21 and the probability plot has a linear correlation coefficient of 0.9416.



(b) A Student's t-distribution probability plot of the same data, with $\nu = 2.5$. The least squares line of best fit has a slope of 1.348 and the probability plot has a correlation coefficient of 0.9994.

Figure 3.2: A comparison of a normal distribution and Student's t fit of the same set of Bijvoet differences.

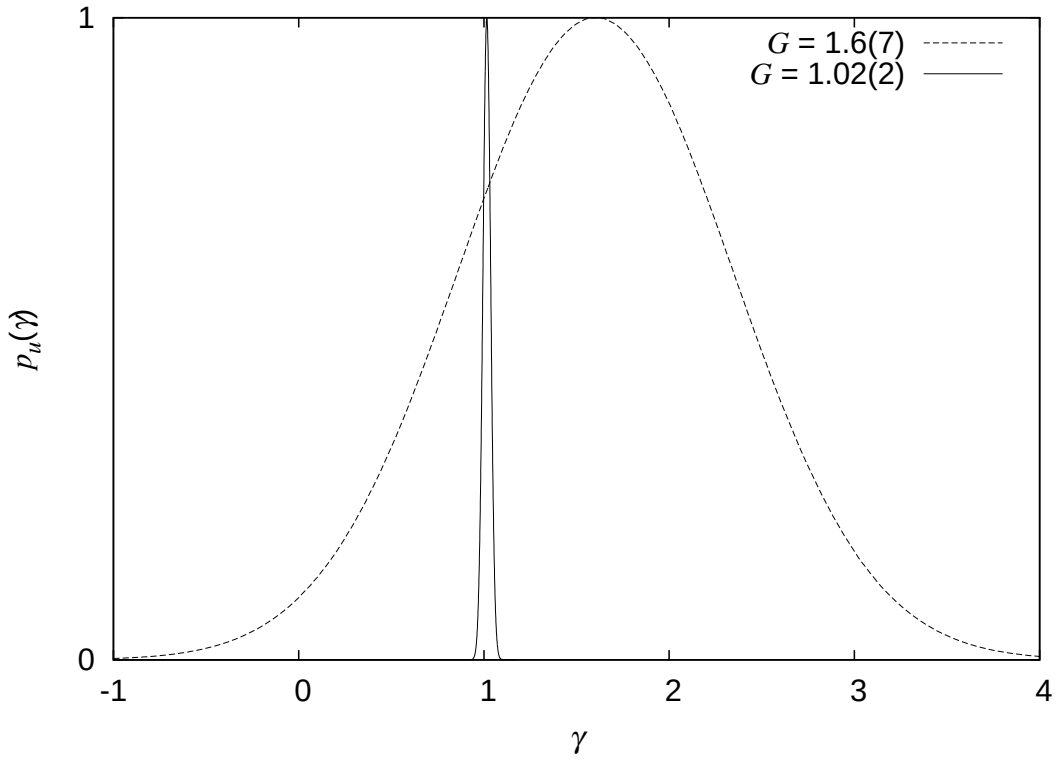


Figure 3.3: The probability density function of $p_u(\gamma)$ for two structures with $G = 1.6(7)$ and $1.02(2)$ respectively.

measure of the location of the distribution and $\sigma(G)$ a measure of the width[1]. The structure with $G = 1.02(2)$ the entire area of the probability density closely surrounds $\gamma = 1$ (indicating the correct absolute structure), leading to values for $p2$ and $p3$ indicating the correct assignment of the absolute structure. In contrast, the structure with $G = 1.6(7)$ has a broad curve for the probability density function, reflected in the values of $p2(\text{false}) = 0.003$, $p3(\text{false}) = 0.003$ and $p3(\text{twin}) = 0.119$.

3.1.5.4 Applications

Comparisons between the Flack x parameter and the analysis of Hooft et al. [2008, 2010] were made for a set of 134 routine in-house data sets for non-centrosymmetric crystal structures. Of the 134 data sets, 99 were in chiral space groups and 3 were measured using copper radiation (the remaining using molybdenum radia-

[1]For a normal distribution the inflection points are situated at $x = \mu \pm 1\sigma$.

tion). Each data set was refined as an inversion twin (*i.e.* refinement of the Flack x parameter) to convergence using smtbx-refine. The twin fraction was refined alongside all other structural parameters so as to take into account any possible correlation between parameters. A Bayesian analysis of the Bijvoet differences was carried out using both the Gaussian and Student's t-distributions. The value of ν for the Student's t-distribution was determined by an automatic procedure that finds the value that maximises the linear correlation coefficient of the Bijvoet differences probability plot, searching over the range $1 \leq \nu \leq 300$. Scaling of the standard deviations of the Bijvoet differences based on the slope of the probability plot was performed for the Student's t-distribution fits, however this was not used when using Gaussian statistics (since in general it isn't a true fit). The complete data recorded are included in Appendix A.

3.1.5.5 Results

Unless stated otherwise, the text below refers to the Student's t-distribution fit.

Using the criteria of Flack and Bernardinelli [2000], it was possible to reliably determine the absolute structure of 63 of the structures analysed using the criterion $u < 0.04$ (strong inversion-distinguishing power), or 92 with the criterion $u < 0.1$ (enantiopure-sufficient inversion-distinguishing power) for both the Flack x and Hooft y parameters, where $u = \sigma(x)$ or $\sigma(y)$. Using the Hooft y parameter alone (*i.e.* only the Hooft y need satisfy the criteria), these numbers could be improved by a further 22 and 2 for $u < 0.04$ and $u < 0.1$ respectively.

Of the 94 structures where the absolute structure was reliably assigned according to the criteria above ($u < 0.1$), there did not seem to be a systematic pattern as to whether the Hooft y parameter was closer to zero, or lower in value than, the Flack x parameter (43 and 45 structures respectively).

Hooft et al. [2008] recommend that a probability plot linear correlation coefficient of at least 0.999 is required in order to establish if the error model used for the data is sufficient. 88 structures meet this requirement for the normal probability plot, whilst a total of 129 structures satisfy the requirement using a Student's t-distribution to model the errors. In all but two cases the Student's t-distribution model gives at least the same or better value of the linear correlation coefficient

for the probability plot when compared to 4 significant figures. This indicates that the automatic optimisation procedure produces values of ν (ranging from 2.4 to 300) that are appropriate models of the error distribution.

The Bijvoet statistics procedure was run both with and without outlier rejection as described in §3.1.5.1. With rejection of outliers it was frequently observed that slope of the probability plot deviated significantly from unity, usually closer to zero. In one case where almost 93% of the reflections were rejected using the criteria of §3.1.5.1 a slope of 0.067 resulted. In contrast when all data were used the probability plot slope was usually much closer to unity, indicating a better fit of the error model. This is the case for 31 out of the 32 structures where more than 10% of the reflections were rejected using an outlier cutoff. The value of ν that was determined when using all data was lower compared to when using a cutoff for 35 out of 43 structures where more than 1% of the reflections were above the cutoff. It is postulated that this analysis demonstrates that the use of the Student's t-distribution is a more robust approach than the use of an abritrary cutoff to reject outliers.

In order to compare the values of the Flack x and Hooft y parameters that were obtained, a total least squares fit of a straight line was attempted using the orthogonal distance regression module of the SciPy scientific tools library for Python [SciPy]. Table 3.1 shows the calculated lines of best fit, whilst Figure 3.4 plots the values obtained for the Flack x and Hooft y parameters using the Student's t distribution. The slope of the straight line is within 2σ(slope) of unity for the normal distribution, and 3σ(slope) for the Student's t distribution. The vast majority of the data points in Figure 3.4 are either along the line of best fit, or the best fit line passes within the error bars. There is an obvious outlier at $(-0.605, 0.299)$ which was identified as structure code 07srv401. On further examination a probable explanation is the poor Bijvoet pair coverage of only 15% for this data set.

3.1.6 Implementation

The Flack x parameter is determined by refinement of an inversion twin along with the rest of the parameters. See §2.2 for further details on the refinement of

Distribution	Slope	Intercept
Normal	0.9874(70)	0.0040(10)
Student's t	0.9855(72)	0.0042(10)

Table 3.1: The total least squares lines of best fit for a plot of the Flack x parameter against the Hooft y parameter and the associated errors.

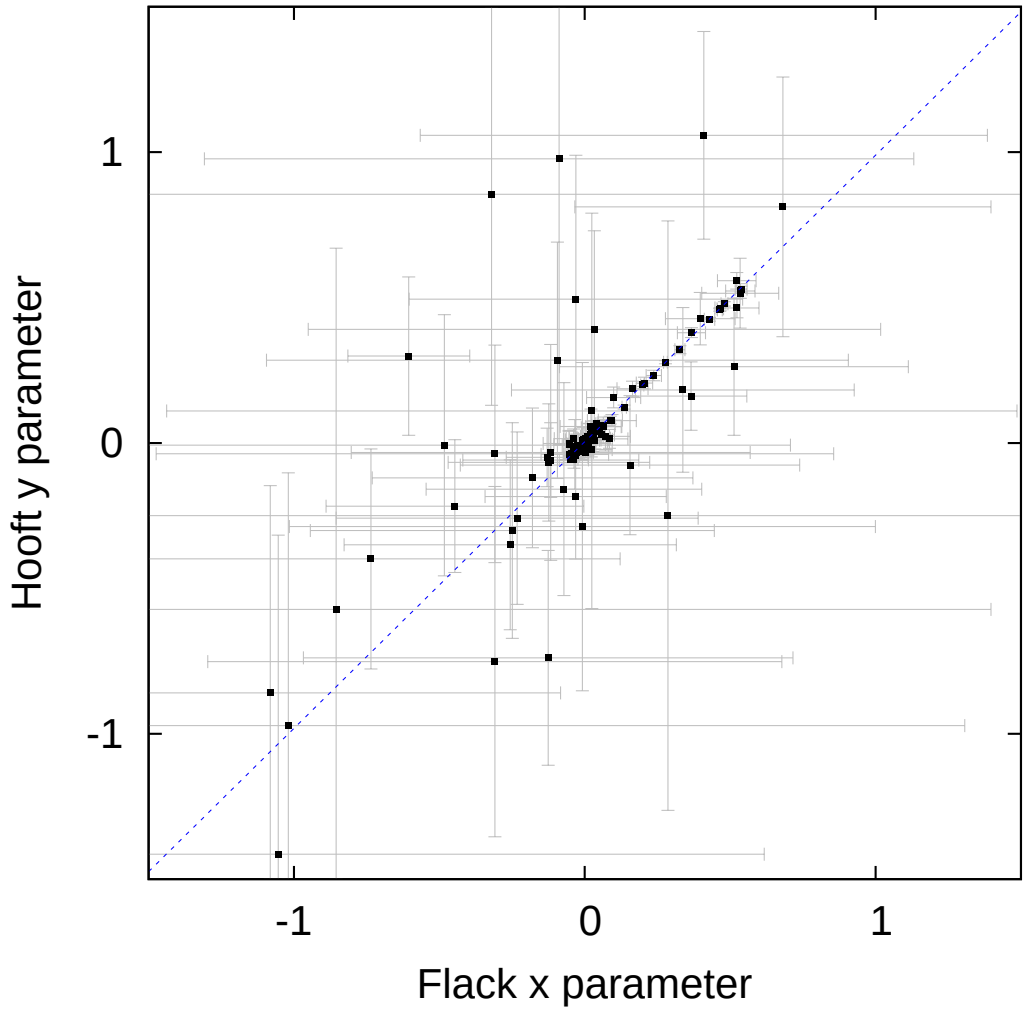


Figure 3.4: A plot of the Flack x parameter against the Hooft y parameter calculated using the Student's t distribution for the error model. The straight dashed line is the total least squares line of best fit of the data, $y = 0.985x + 0.007$. The grey error bars indicate the standard uncertainty in the calculated values of the Flack x and Hooft y parameters respectively.

twins within smtbx-refine.

The Bayesian statistics analysis of the Bijvoet differences as described above is implemented in the smtbx.absolute_structure module. A command line tool was also developed that runs the procedure automatically for a given structure file and/or reflection file, or alternatively for all non-centrosymmetric structures of the given file type that are found in a recursive search in the directory provided.

In the software Olex2 [Dolomanov et al., 2009a] the procedure is run by default after every refinement for non-centrosymmetric structures regardless of the refinement program used and a warning printed if it is suspected that inversion of the structure is necessary, or refinement of an inversion twin may be needed. A more complete output of the analysis can be obtained when viewing the Bijvoet differences probability and scatter plots that are available through the reflection statistics section of the Olex2 GUI. Here the user may choose between using the normal and Student's t-distributions and also choose to use an '.fcf' file (such as can be output by SHELXL) as the source of F_c^2 instead of using internally-calculated structure factors. The plots are displayed using the graph tools that have been developed within Olex2 (see also §3.2; an example of a Bijvoet differences scatter plot is shown in Figure 3.5).

3.2 Reflection Statistics in Olex2

A new tool has been developed within Olex2 for visualisation of the reflection data. Various common plots of the reflection data have been implemented, which frequently can be useful in identifying potential issues with the data and/or model. The framework has been designed in such a way that new graphs can be easily added into the existing framework and exposed to the user through the GUI with minimal effort. The graphs are displayed in Olex2 using custom graph plotting code (see for example Figures 3.5 and 3.6). Optionally a comma-separated values (csv) file can be output for all graphs to enable plotting of the data in external software (as was used for many of the graphs in this chapter).

In addition to the plots that are discussed in more detail below, other plots that have been implemented within Olex2 include plots of scale factor, $R1$-factor and $\frac{F_o}{F_c}$ *vs.* resolution, normal probability plots [Abrahams and Keve, 1971] and

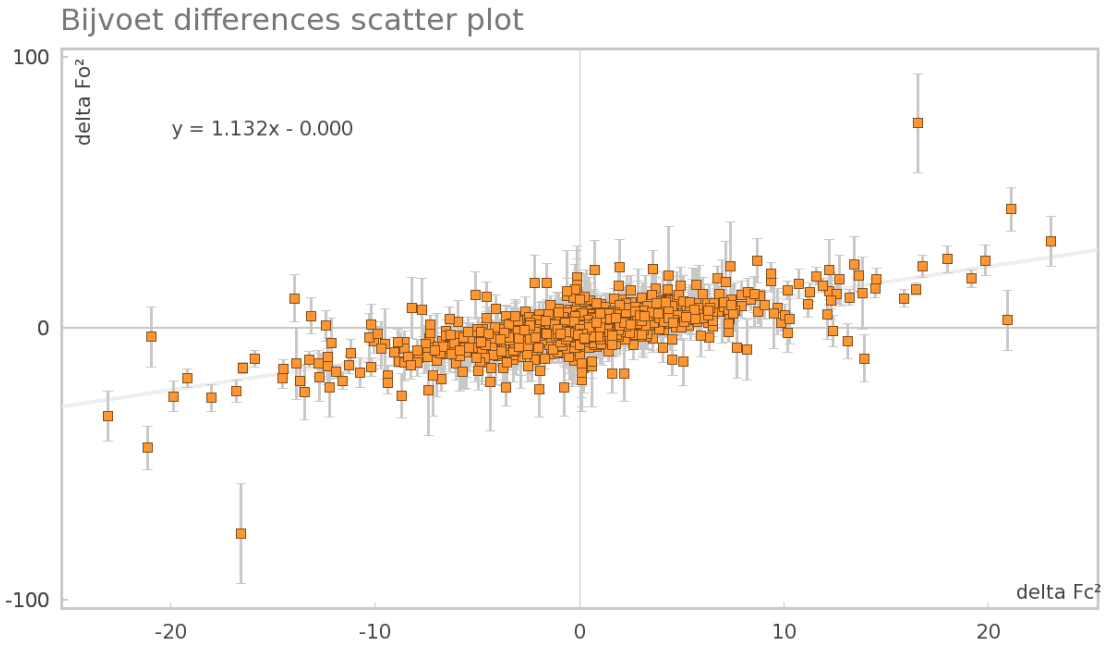


Figure 3.5: An example of a Bijvoet differences scatter plot as displayed in Olex2.

Wilson plots [Giacovazzo et al., 2002].

3.2.1 Cumulative Intensity Distribution

As described by Howells et al. [1950], the cumulative intensity distribution can be useful in distinguishing between centrosymmetric and non-centrosymmetric structures. Stanley [1972] extended the method to aid in the identification of twinned structures.

The data are sorted and grouped into bins by resolution. For each intensity, $z = I/\langle I \rangle$, the fraction of the intensity over the average intensity for the given bin is calculated. Use of z rather than I compensates for the decrease in $\langle I \rangle$ with $\sin\theta$ that is caused by thermal motion and the decrease in the atomic scattering factors. The fractions, $N(z)$, of the reflections whose intensities are less than or equal to z are then plotted against z, as shown in Figure 3.6. The calculated distribution can then be compared against the theoretical distributions for centric and acentric structures:

$$N_{\text{centric}}(z) = \text{erf}(\frac{1}{2}z)^{\frac{1}{2}}, \tag{3.31}$$

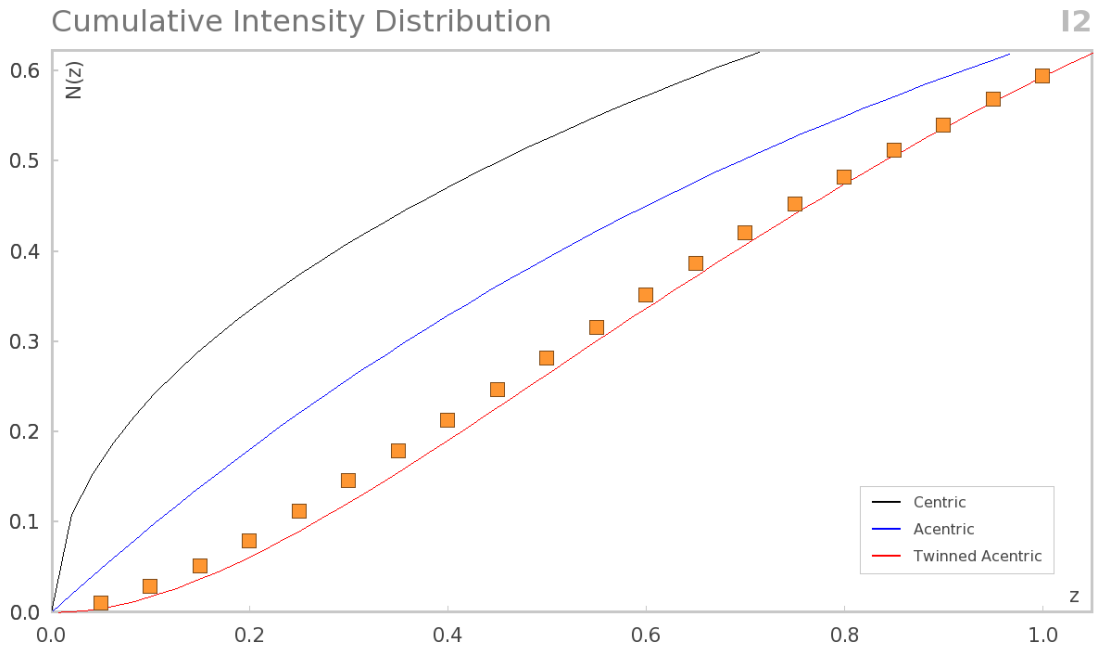


Figure 3.6: An example of the presence of twinning being indicated by the cumulative intensity distribution.

and

$$N_{\text{acentric}}(z) = 1 - \exp(-z) \tag{3.32}$$

where erf is the 'error function'. The theoretical distribution for a twinned acentric structure as determined by Stanley [1972] is given by

$$N(z) = 1 - (1 + 2z)\exp(-2z). \tag{3.33}$$

3.2.2 F_o vs. F_c Plot

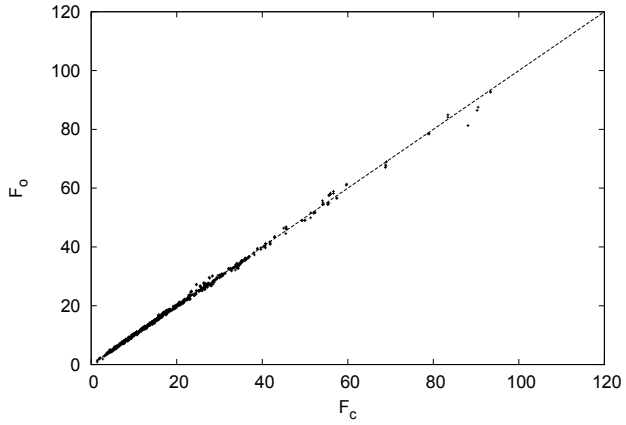
A plot of F_o vs. F_c can indicate problems with the current model and/or data. Figure 3.7a shows a plot of F_o vs. F_c for a twinned structure that also required refinement of an extinction parameter. Figure 3.7b shows the effect on the plot when extinction is neglected; since extinction primarily affects the strong reflections at low angles, the points deviate from the line $y = x$ at larger values of F_o. In Figure 3.7c twinning is not accounted for, resulting in a larger spread of the

data points. A F_o vs. F_c plot can also be used to identify individual outlying reflections that may be omitted from the refinement (*e.g.* a low angle reflection that was partially occluded by the beam stop and hence the measured intensity was much lower than it should be).

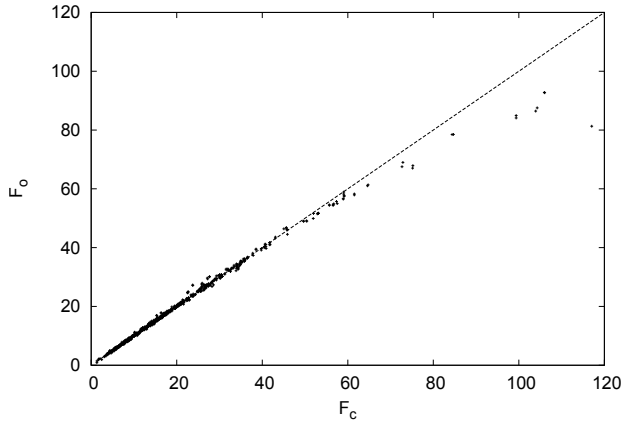
3.2.3 Data Completeness

A plot of the data completeness binned by resolution can give some insight into the quality of a data collection, or give indications of potential problems. For example, whilst not necessarily having a detrimental effect on the least squares refinement, missing low angle reflections can have a significant impact in an electron density map calculation. For procedures such as that discussed in §4 it is important to be aware of such missing reflections.

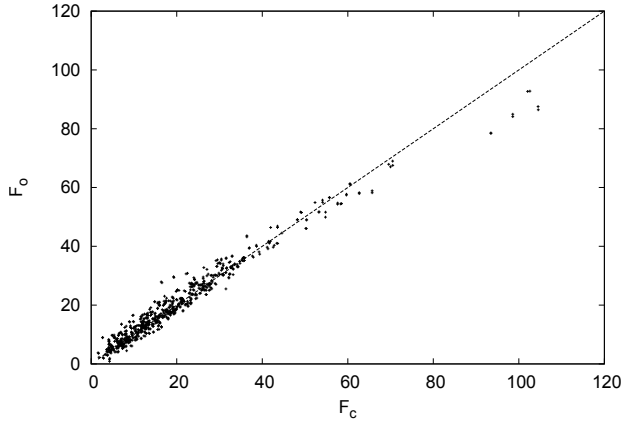
Another example could be an indication that the data were collected assuming higher crystal symmetry than was actually the case. Furthermore, a plot of data completeness against resolution could be instructive in choosing an appropriate value for the CIF data item _diffrn_reflns_theta_full, which is in turn used in computing the value of _diffrn_measured_fraction_theta_full. The definition of _diffrn_reflns_theta_full specifies *the theta angle (in degrees) at which the measured reflection count is close to complete.* According to this definition and by inspection of Figure 3.8, a value of $\sim 27.5^\circ$ would be appropriate for this data set. It is also evident from inspection of the plot that there is at least one low angle reflection missing, which could be important if it was necessary to use the solvent masking procedure described in §4. A list of the missing reflections sorted by resolution is also output by the routine.



(a) Final model



(b) Extinction not refined



(c) Twinning or extinction not refined

Figure 3.7: The effects of twinning and extinction on a plot of F_o vs. F_c. The line $y = x$ is plotted as a dashed line.

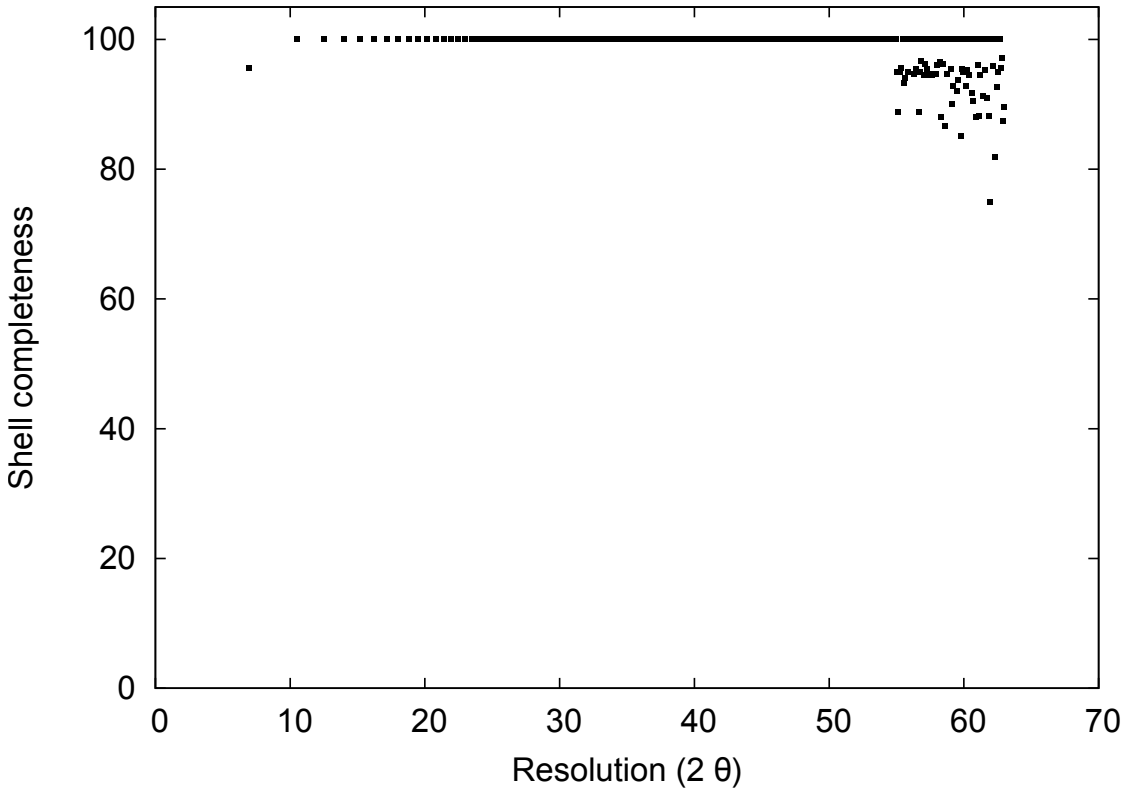


Figure 3.8: A plot of data completeness in resolution shells.

Chapter 4

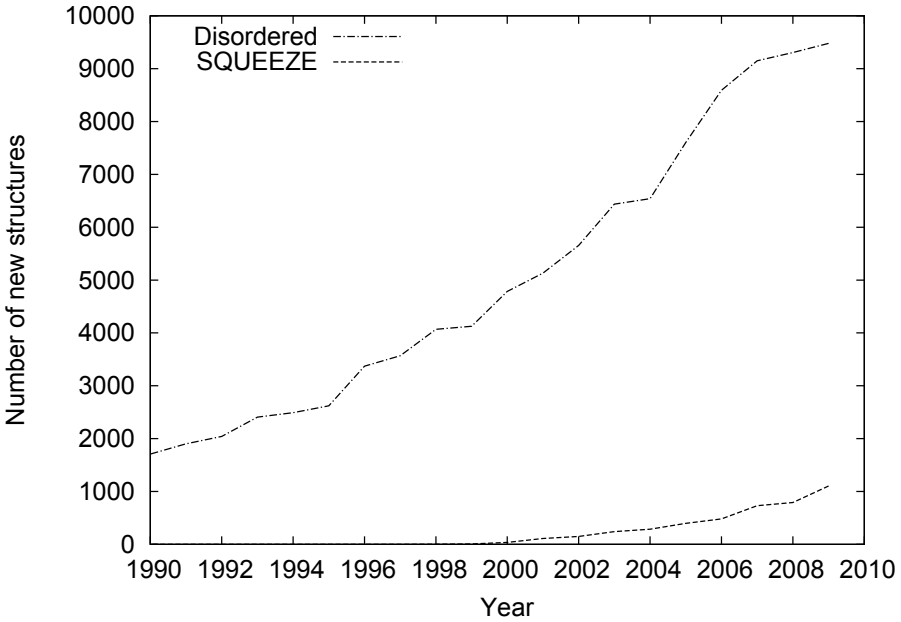
A New Solvent Masking Procedure

4.1 Introduction

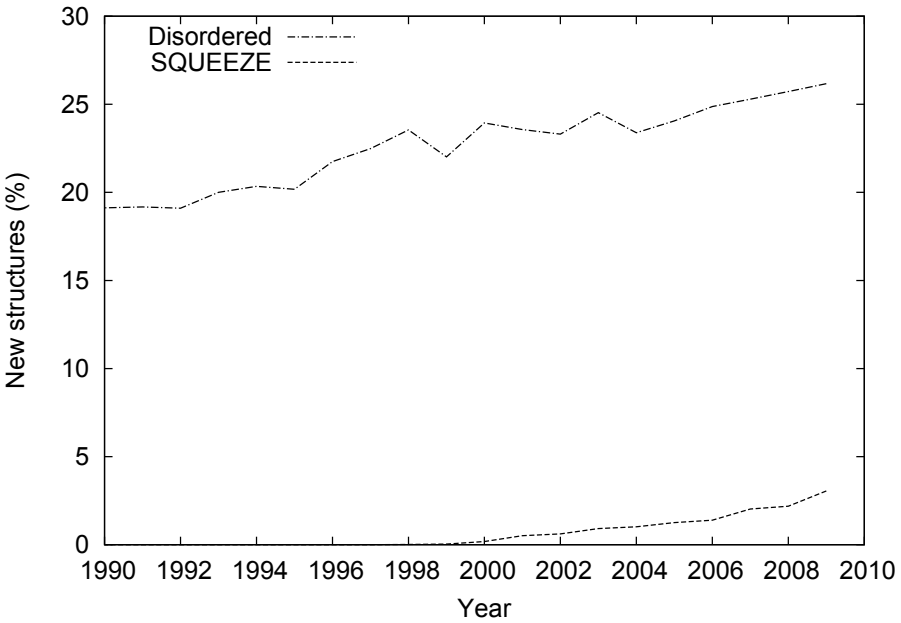
The number of new crystal structures deposited in the Cambridge Structural Database (CSD) [Allen, 2002] that contain disorder of some description appears to be increasing both in real terms and as a proportion of the total number of structures deposited (Figure 4.1). As more than a quarter of crystal structure depositions now contain some form of disorder, it is evident that the correct treatment of disordered crystal structures is more important than ever.

A crystal structure determination is a time and space-averaged picture of the electron density, *i.e.* an average of every unit cell in the crystal, and averaged over the time of the diffraction experiment. Two kinds of disorder are possible: positional, when an atom/fragment/molecule can occupy two or more similar orientations or positions, and substitutional, when two or more atoms or molecules can occupy the same site in different unit cells. Positional disorder can be subdivided into dynamic and static, where the former describes real motion in the solid state, and the latter is simply different orientations in different unit cells.

At this point, it is apposite to recall that the very nature of the Fourier transform relationship between the electron density and the structure factors relates



(a) Number of new structures.



(b) Percentage of new structures.

Figure 4.1: New structures deposited with CSD per year that are disordered, or have used the SQUEEZE routine.

every point in the unit cell to every structure factor:

$$\rho(x) = V^{-1} \sum_{h} |F_h| \exp(i\phi_h) \exp(-2\pi i h \cdot x) \tag{4.1}$$

$$F(h) = \int_{cell} \rho(x) \exp(2\pi i h \cdot x)\, dx. \tag{4.2}$$

The crystallographic model is simply an interpretation of the measured electron density, usually in terms of an atomic model. Therefore, any deficiencies in the model can adversely affect the parameters of the ordered part of the model, as their values change to compensate for deficiencies elsewhere in the least squares minimisation. This can affect both the geometry of the molecule and also the calculated standard uncertainties of refined and derived parameters. It is therefore evident that it is important to account for any disorder in the best possible manner during the process of crystal structure determination.

Orientational disorder is most commonly modelled with two or more overlapping fragments, often requiring the extensive use of restraints and/or constraints to keep the model chemically reasonable. When appropriate, a somewhat more elegant alternative may be to model atoms as continuously disordered along some special figure, such as a line, a ring or the surface of a sphere, as featured by the program CRYSTALS [Schröder et al., 2004].

However, there are often cases where extensive disorder or unknown solvent composition is such that neither approach is appropriate. Van der Sluis and Spek [1990] suggested a method whereby the contribution to the calculated structure factors of the disordered solvent area is calculated *via* a Fourier transform of that area. This solvent contribution can then either be added to that calculated from the ordered part of the structure, or alternatively subtracted from the observed data before further cycles of refinement. This method has been made widely popular by the SQUEEZE routine available through the software PLATON [Spek, 2003]. As can be seen in Figure 4.1 it appears that this method is becoming increasingly accepted as an appropriate method to deal with cases of severe disorder, with, at the time of writing[1], a total of over 4500 structures in the CSD where the

[1]CSD V5.32 database, November 2010 update

SQUEEZE routine has been applied, with over 1100 in 2009 alone.

4.2 Theory

The total calculated structure factor, F_h^{calc}, can be considered as being composed of two parts, that from the ordered atomic model of the structure, F_h^{model}, and the diffuse solvent[1] contribution, F_h^{diff}. These are related according to

$$F_h^{calc} = F_h^{model} + F_h^{diff}. \tag{4.3}$$

The diffuse solvent contribution can be calculated from the discrete Fourier transform of the electron density difference map over the solvent area

$$F_h^{diff} = V_g \sum_{x_j \in S} \Delta\rho(x_j) \exp(2\pi i h \cdot x_j), \tag{4.4}$$

where V_g is the volume per grid point and S is the set of grid points x_j that define the solvent area.

This difference map is in turn calculated according to

$$\begin{aligned} \Delta\rho(x) = V^{-1} \sum_h \big[& s \left| F_h^{obs} \right| \exp(i\phi_h^{calc}) \\ & - \left| F_h^{model} \right| \exp(i\phi_h^{model}) \big] \exp(-2\pi i h \cdot x), \end{aligned} \tag{4.5}$$

where s is the overall scale factor, $\left|F_h^{obs}\right|$ is the observed structure factor, ϕ_h^{model} is the phase of F_h^{model}, ϕ_h^{calc} the phase of F_h^{calc} and V is the volume of the unit cell.

We can optimize the diffuse scattering contribution, F_h^{diff}, by iteratively applying equations 4.5 and 4.4. Given the initial structure model, ϕ_h^{calc} and the true scale factor, s, are unknown. In the first cycle the true values are substituted with $\phi_h^{calc} = \phi_h^{model}$ and the scale factor obtained from the starting model is used. In subsequent cycles, the phases and scale factor calculated using the provisional solvent contribution from the previous cycle are used to provide an improved estimate of the true values.

[1]The term "solvent" is used rather loosely throughout this chapter to include anions and cations, which could also be treated in the same way.

By examining equation 4.2 it is possible to see that when substituting h for the 000 reflection, the resulting structure factor, F_0 is equal to the integral of the electron density over the whole unit cell, *i.e.* the total electron count of the unit cell. The average density level of the difference map is zero and a summation of the electron density over the solvent area gives the value

$$F_0 = \sum_{x_j \in S} V_g \Delta\rho(x_j), \tag{4.6}$$

where the count over the region outside the solvent area is equal to $-F_0$.

The average density of the difference map can be raised to such a level that a summation over the points outside the solvent area gives zero, enabling an approximation of the number of electrons in the solvent region to be given by

$$F_0^{diff} = F_0[V/(V - V_s)], \tag{4.7}$$

where V_s is the volume of the solvent region and V is the unit cell volume. A contribution of F_0^{diff}/V is added to $\Delta\rho(x_j)$ before the next iteration of the procedure.

4.2.1 Refinement

Once the diffuse solvent contribution to the calculated structure factors has been determined, it is then necessary to include this contribution in the refinement of the ordered part of the structure. The most straightforward way of doing this would be to refine against the total calculated structure factor as defined by equation 4.3. This would be the preferred method if the desired refinement program is capable of accepting fixed contributions to the structure factors, which is the case with our own refinement program, smtbx-refine [Bourhis et al., 2011].

In order to use the method with the refinement program SHELXL [Sheldrick, 2008], an alternative approach is used to modify the observed structure factors:

$$F_h^{obs'} = s \left| F_h^{obs} \right| \exp(i\phi_h^{model}) - F_h^{diff} \tag{4.8}$$

4.2.2 Incomplete Data

It is well known that the low angle data contain much of the large-scale electron density variation throughout the unit cell, whilst the high angle data encode the fine details of the electron density. This can be clearly demonstrated in the use of high-pass and low-pass filters on the Fourier transforms of two-dimensional images as shown in Figure 4.2 [See also Figure 8 of Aubert and Lecomte, 2007]. Figure 4.2b is obtained from the Fourier transform in Figure 4.2e where only the data close to the origin (centre of the Fourier transform), and it is clear that, whilst the details in the buildings are lost, the large-scale intensity variation in the original image remains. When the data close to the origin (low angle data) are excluded (Figure 4.2f), the image is now mostly even in intensity across the image, however the image retains the details in the buildings (Figure 4.2c).

If some of the low angle data are missing (for example, obstructed by the beam stop), then this can have a detrimental impact on the iterative procedure outlined above. A key step involves adding a contribution, F_0^{diff}, that is calculated by a summation of the electron density over the solvent area. It is evident that if missing low angle data causes the overall levels of the electron density to be incorrect, these errors will propagate through the procedure and eventually lead to an incorrect electron count. In our experience just one missing low angle reflection can be enough to cause significant errors in the estimation of the electron count in the solvent region.

With this in mind, we propose a modification to the above procedure that can be used to compensate for such missing data. At the beginning of each cycle, before the application of equation 4.5, the missing observed amplitudes are substituted by the $|F_{calc}|$ obtained as a result of the previous cycle. As such, the missing amplitudes are allowed to float freely throughout the procedure, from the initial starting point of those amplitudes calculated from the ordered part of the model. It is expected that this modification will lead to a more accurate estimate of the diffuse contribution and the electron count within the solvent region than would be obtained by essentially including the contribution of the unobserved amplitudes as zero.

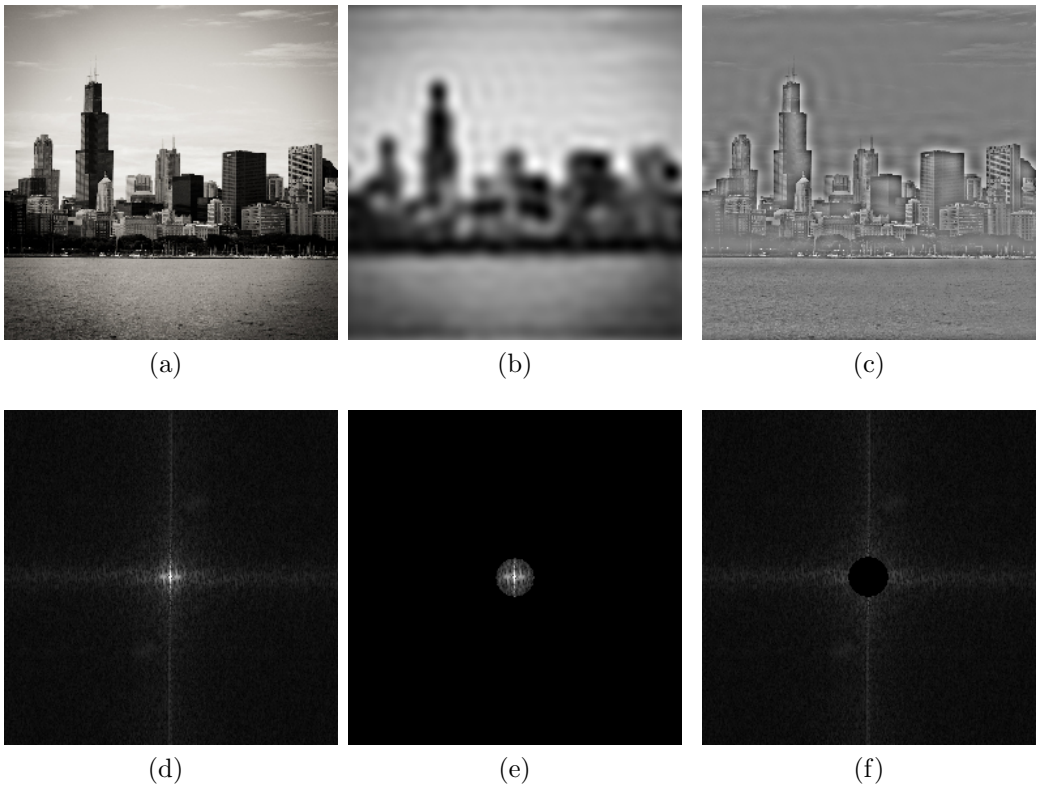


Figure 4.2: An image of the Chicago skyline (a) and its Fourier transform (d); (b) is the image reconstructed after application of a low-pass filter to the Fourier transform (e); (c) is the image reconstructed after application of a high-pass filter to the Fourier transform (f). Fourier transforms calculated using the FTL-SE software [JCrystalSoft, 2010].

4.2.3 Twinned Data

In order to obtain a correct electron density map in the case of twinned data, it is necessary to first deconvolute, or 'detwin', the data. Detwinning of intensity data and methods of doing so, are discussed in detail in §2.2. Regardless of the method used to detwin the data, it is necessary to know the twin fraction(s) sufficiently accurately in order to obtain a reasonable estimate of the untwinned intensities. In our experience the twin fraction obtained through least squares refinement of an incomplete model does not usually lead to a sufficiently accurate difference electron density map for the solvent masking procedure to work well.

4.2.4 Standard Uncertainties

In the case of using a solvent mask as an alternative to modelling the solvent with a disordered model and under the assumption that the model of the disorder is a good fit to the data (at least comparable to that attained with the use of a solvent mask), then it is to be expected that the standard uncertainties on the refined parameters will be artificially reduced. The standard uncertainty for parameter p_j is given by

$$\sigma(p_j) = \left((\mathbf{A}^{-1})_{jj} \frac{\sum_{i=1}^{N} w_i \Delta_i^2}{N - P} \right)^{1/2} \tag{4.9}$$

where $\mathbf{A}^{-1}$ is the inverse least squares normal matrix, $w_i\Delta_i^2$ are the weighted residuals, for N observations and P parameters.

Under the assumption that the value of $(\mathbf{A}^{-1})_{jj}$ is unchanged and that the fit of the model to the data is identical in both cases, it is evident that the value of the denominator will increase as the number of parameters decreases, thus causing the standard uncertainty of the refined parameters to be underestimated.

Consider a hypothetical data set for which there are 4000 unique reflections after merging with 316 parameters refined for the ordered part of the model. A dichloromethane molecule is disordered over 3 orientations and each part is modelled anisotropically for each atom (this is unlikely considering a 3 part disorder) and the positions of the hydrogen atoms are refined using riding constraints [Watkin, 2008]. This gives a total of 27 parameters for each part (3 site param-

eters and 6 anisotropic displacement parameters per non-hydrogen atom). An occupancy parameter is refined for each disorder component (with a restraint that the sum of the occupancies is equal to unity), giving a total of 84 parameters to model the disorder. From equation 4.9 a reduction in the standard uncertainties of 1.15% would be expected after using the solvent masking procedure, taking into account only the difference in the number of parameters. More realistically, only the major disorder component is refined anisotropically, giving a total of 54 parameters required to model the disorder. Now a reduction of 0.74% in the standard uncertainties would be expected. Since the number of data is usually much greater than the number of parameters[1], the reduction in the standard uncertainties due to a change in the number of parameters is expected to be small.

It is important to remember, however, that the solvent masking procedure is not intended as a *replacement* for correct atomic models of solvent disorder, but rather as a *complementary* technique for the cases when an atomic model is insufficient, or would lead to chemically nonsensical results. In such a case it is expected that a significantly better fit of the model to the data can be obtained using a solvent mask and hence the standard uncertainties would decrease by a much greater amount than would be caused purely by the reduction in number of refined parameters.

4.3 Method

The first step is to identify the areas of the unit cell that are accessible to solvent molecules. A grid is set up where the grid step is chosen relative to the high resolution limit, usually by a factor of 1/4. Initially all the grid points are set to be 1. All points within a distance of r_i from atom i, where r_i is the sum of the van der Waals radius and the solvent probe radius, r_{probe}, are set to 0. All grid points with the value 0 are then tested to see if they are within a distance r_{shrink} of a grid point marked 1 and are themselves set to 1 if this is the case. The grid points marked 1 are thus the solvent accessible region. This two-step process

[1]The current recommendation of the IUCr journals is for a data/parameter ratio > 10 for centrosymmetric and > 8 for non-centrosymmetric structures for a 'quality structure determination' (http://journals.iucr.org/services/cif/checking/platon.html).

has the effect of smoothing the surface area of the solvent region relative to just including all points that are outside the van der Waals radii of the atomic model. The search for solvent accessible voids uses the procedure originally developed for the cctbx bulk-solvent and scaling module [Afonine et al., 2005].

The above procedure acts in the space group P 1, however it can be optimised by taking into account the space group symmetry. All atoms are moved to the standardised asymmetric unit and this is expanded with symmetry equivalents within a buffer region equivalent to the sum of the maximum van der Waals radius and the solvent probe radius. Solvent accessible volumes are then carried out as described previously and the space group symmetry is applied to the resulting map to yield all the solvent regions in the unit cell. This approach gives substantial speed increases for higher symmetry space groups and large unit cells.

Independent voids are then identified using a simple flood fill[1] algorithm and, for each void, a centre of mass and moment of inertia is calculated. Each void is labelled with a sequential integer. The solvent accessible volume constitutes all grid points with value greater than zero.

The solvent contribution to the structure factors is calculated following the iterative procedure outlined in section 4.2.

4.4 Implementation

4.4.1 Computational Crystallography Toolbox

The procedure outlined above is implemented as part of the Small Molecule Toolbox (smtbx) which is part of the Computational Crystallography Toolbox (cctbx) [Grosse-Kunstleve et al., 2002].

The high-level code controlling the flow of the program is written in Python, whilst the computationally intensive calculations (Fast Fourier Transform, void search, structure factor calculations, etc.) are written using C++, which is exposed to Python using the Boost.Python Library (http://www.boost.org/).

This combination allows for rapid prototyping of new ideas, whilst maintaining the performance benefits of a compiled language.

[1] http://en.wikipedia.org/wiki/Flood_fill

4.4.2 Olex2

The procedure is integrated within the Olex2 software package [Dolomanov et al., 2009a], which can be used for visualisation of the calculated solvent accessible voids and F_h^{diff} and F_h^{calc} electron density maps. Once calculated, it is straightforward to include the solvent mask in the refinement of the ordered part of the structure, either with our own refinement program, smtbx-refine [Bourhis et al., 2011] or alternatively with SHELXL [Sheldrick, 2008].

The details of the calculations and subsequent refinement are seamlessly propagated into the CIF output by Olex2 ready for publication.

4.5 Test Structures

There are a number of tests that can be carried out to test the validity of the procedure. In the first instance, a crystal structure can be taken where the solvent content is both known and ordered. The procedure can be carried out using both the original observed data and using structure factors calculated from the model, and with and without prior least squares refinement. The electron count estimated by the procedure should be close to that expected for the solvent that is omitted from the model and the subsequent least squares procedure should give a similar outcome to that obtained with an atomic model of the solvent.

The completion of missing data can be tested similarly using a test case with ordered solvent, where one or more low angle reflections are missing (or manually omitted) and the results of the procedure can be compared with and without using the set completion technique. The amplitudes of the omitted reflections can be followed throughout the iterations, in order to observe whether their values converge close to the true values.

The outcomes of the procedure obtained for several test structures and applications are tabulated in Table 4.1. Analysis of the differences in geometry after the procedure is presented in Table 4.2.

Compound I

1-methyl-3-phenyl-7,8-dimethoxy-3H-pyrazolo(3,4-c)isoquinoline acetonitrile solvate [Bogza et al., 2005, CSD code YAKRUY], space group $P\bar{1}$, $a = 7.086(1)$, $b = 10.791(3)$, $c = 12.850(2)$ Å, $\alpha = 104.16(2)°$, $\beta = 105.87(2)°$, $\gamma = 95.86(1)°$, containing one acetonitrile molecule per asymmetric unit.

A synthetic data set was created from the full atomic model to $d_{\min} = 0.7$ Å with 100% completeness. The acetonitrile solvate molecule was then discarded from the model with the solvent masking procedure used in place. An electron count of 43.4 was found for a single void per unit cell, with volume 174 Å^3. The structure refined to $R1 = 0.88\%$ using unit weights. The 001 reflection was then discarded and the procedure repeated, with the resulting electron count of 40.5 and $R1 = 1.81\%$. The procedure was run once more, this time using the set completion technique, giving an electron count of 43.4 and $R1 = 0.89\%$.

To test the technique more extensively, approximately 5% of the reflections were discarded at random, resulting in a completeness of 95.3% (Figure 4.3). Without the use of the set completion technique, an electron count of 34.5 was obtained and $R1 = 5.20\%$. Using the set completion technique gave an electron count of 43.4 and $R1 = 0.89\%$. The amplitudes of three of the omitted low angle reflections were followed at each iteration. From the results shown in Figure 4.4 it can be seen that in each case the amplitude of the reflection converges close to the 'true' value.

Compound II

1(2,5)-Thiophena-3,7-dioxa-2,8-dioxo-5(5,5')(9,10-bis(4-methyl-1,3-dithiol-2-ylidene)-9,10-dihydroanthracena)cyclo-octaphane dichloromethane solvate [Godbert et al., 2001, CSD code QIPZEU], space group $P2_1/c$, $a = 11.407(5)$, $b = 17.160(8)$, $c = 15.607(7)$ Å, $\beta = 99.83(2)°$. The asymmetric unit contains one dichloromethane molecule which is disordered over two orientations, modelled with 50% occupancy for each position. The refinement converges to $R1 = 7.15\%$ for 2901 reflections where $I \geq 2u(I)$. Bond similarity restraints on the C-Cl distances were required in order to keep the geometry of the dichloromethane molecule chemically reasonable. Alternatively a solvent mask was used instead

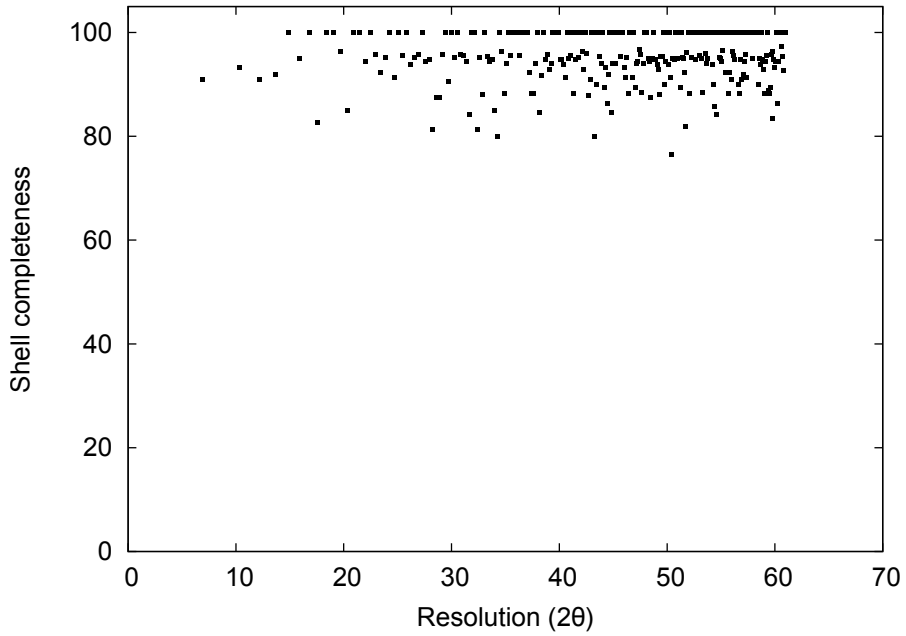


Figure 4.3: The completeness in resolution shells for compound I after approximately 5% of the reflections were discarded at random.

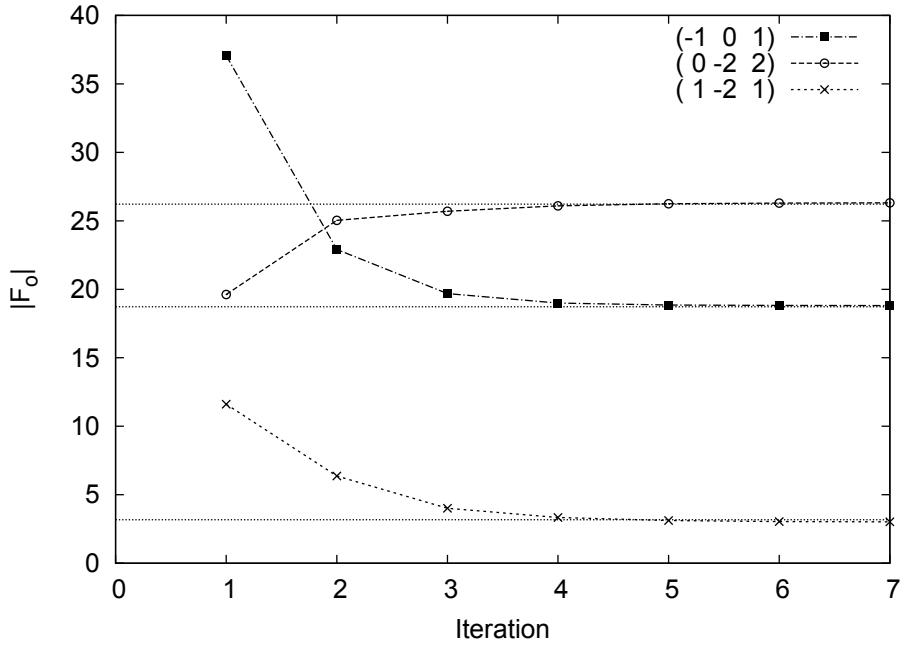


Figure 4.4: The amplitudes of three 'floating' missing structure factors at each iteration of the solvent masking procedure. The lightly dashed horizontal lines indicate the 'true' amplitudes.

of modelling the dichloromethane atomically. Two symmetry-related voids were found in the unit cell, each with a volume of 235 $Å^3$ and an estimated electron count of 79.7 per void. Refinement using the solvent mask gave a slightly reduced $R1 = 6.99\%$. Although the completeness was high (99.98%) to $2\theta = 50°$, it was noted that the (100) reflection was not measured. The solvent masking procedure was repeated using the set completion technique, giving an estimated electron count of 84.8 and a marginally lower $R1 = 6.95\%$ after refinement to convergence. Although there was not a significant change in the $R1$-factor, the estimated electron count obtained using the set completion technique was much closer to the expected value for two dichloromethane molecules ($2 \times 42e^- = 84e^-$).

Compound III

9-(4-chlorophenyl)-5,6-dimethoxy-10,11,19-triazatetracyclo[9.8.0.$0^{3,8}$.$0^{13,18}$] nonadeca-1(19),3(8),4,6,13(18),14,16-heptaen-12-one acetone solvate [Batsanov, 2000], space group $C2/c$, $a = 20.085(5)$, $b = 7.717(1)$, $c = 28.907(3)$ Å, $\beta = 97.77(1)°$. The asymmetric unit contains half an acetone solvate molecule sited on a special position. Refinement of the full atomic model converges to $R1 = 5.03\%$. The acetone molecule was then discarded in the model and the solvent masking procedure was used in its place. Four voids were found per unit cell, each with a volume of 124 $Å^3$ and an estimated electron count of 25.9 and $R1 = 5.11\%$. It was noted that the (002) reflection was missing and therefore the procedure was repeated using the set completion technique, giving an electron count of 32.9 and $R1 = 5.00\%$.

Compound IV

Tetrachloro-(1,2-bis(diphenylphosphoryl)ethane-O,O')-tin(iv) acetone solvate [Batsanov et al., 2009, CSD code QOZQAY], space group $P\bar{1}$, $a = 9.796(2)$, $b = 11.497(1)$, $c = 16.171(3)$ Å, $\alpha = 99.91(1)°$, $\beta = 102.85(1)°$, $\gamma = 110.81(1)°$. The structure contains one acetone molecule per asymmetric unit, which is disordered over two parts in the ratio 80:20. With the use of bond similarity restraints, the refinement converges to $R1 = 2.64\%$ for 7533 reflections where $I \geq 2u(I)$. When a solvent mask was used instead of the atomic model of

the disordered acetone, one void was found in the unit cell with a volume of $309\,\text{Å}^3$ with an estimated electron count of 63.6 electrons per unit cell. This is comparable to the expected count for two acetone molecules ($2 \times 32e^- = 64e^-$). Refinement using the solvent mask converged to $R1 = 2.47\%$.

4.6 Applications

Compound V

1,13(1,4)-Dibenzena-7,19(2,6)-bis(9,10-bis(4,5-bis(methylthio)-1,3-dithiol-2-ylidene)-9,10-dihydroanthracena)-3,6,8,11,15,18,20,23-octaoxa-2,12,14,24-tetraoxotetracosaphane dichloromethane hexane solvate [Christensen et al., 2001, CSD code QOSDIL], space group $P\bar{1}$, $a = 14.525(2)$, $b = 15.647(2)$, $c = 18.238(2)\,\text{Å}$, $\alpha = 88.59(2)°$, $\beta = 86.97(2)°$, $\gamma = 79.82(2)°$. The dichloromethane and hexane molecules are severely disordered along channels through the unit cell. In the original publication the disorder was modelled with arbitrary chlorine and carbon atoms, with $R1 = 7.63\%$. Five of the six largest residual electron density peaks ($0.6 - 0.65\,\text{e}^-\,\text{Å}^{-3}$) are found within the solvent region. After application of the solvent masking procedure, $R1 = 6.73\%$ and the highest residual electron density peak ($0.64\,\text{e}^-\,\text{Å}^{-3}$) is close to one of the sulphur atoms, with no significant residual electron density peaks within the solvent region. A single void was found that runs along a channel parallel to the b-axis, with volume $596\,\text{Å}^3$ and an estimated electron count of 143.5 electrons per unit cell.

Compound VI

2-(3'-(t-Butyldimethylsiloxy)-1'-oxo-1',3'-dihydroisoindol-3'-yl)-1,2'-propano-1,2-dicarba-closo-dodecaborane pentane solvate [Batsanov et al., 2001, CSD code QOYXOR], space group $P\bar{1}$, $a = 14.448(1)$, $b = 14.680(1)$, $c = 16.137(1)\,\text{Å}$, $\alpha = 101.58(1)°$, $\beta = 90.07(1)°$, $\gamma = 96.13(1)°$. The pentane solvent is severely disordered along channels through the unit cell. In the published structure the disordered solvent is modelled with arbitrary carbon atoms of varying fixed

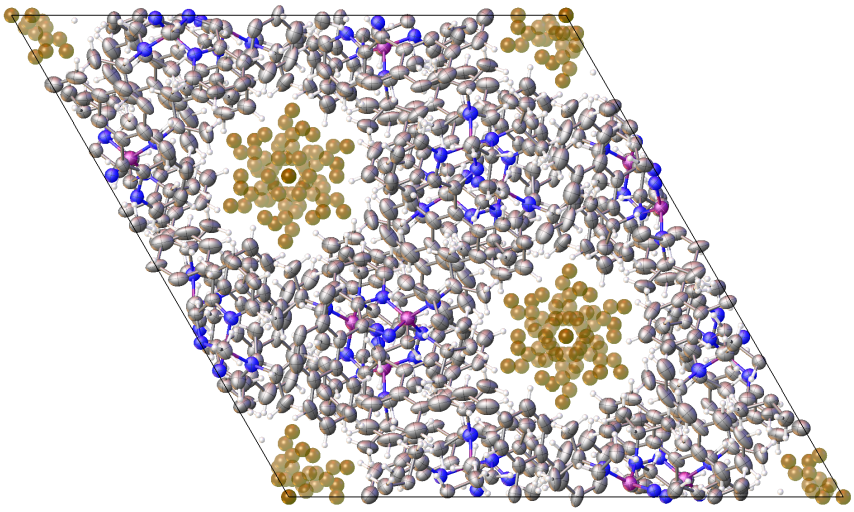
occupancy, with $R1 = 8.35\%$. The ten largest residual electron density peaks ($0.65 - 0.79\,\mathrm{e}^{-}\,\mathrm{\AA}^{-3}$) are found within the solvent region. After application of the solvent masking procedure (electron count = 117.7), $R1 = 6.15\%$ and there are no significant residual electron density peaks within the solvent region. However, it was noted that there were several missing low angle reflections, (001), (100), (010), (−101) and (101). The procedure was repeated using the set completion technique, resulting in a significantly larger electron count of 235 and $R1 = 5.57\%$.

Compound VII

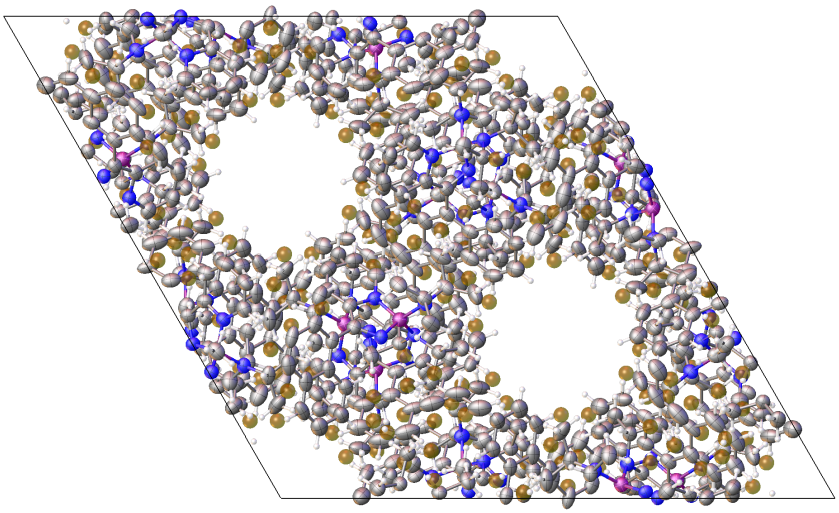
Space group $R\bar{3}$, $a = 27.065(3)$, $b = 27.065(3)$, $c = 24.318(3)\,\mathrm{\AA}$ [Batsanov, 2009]. Hexane solvent molecules are severely disordered along the 3-fold axis. After refinement with no attempt made to model the solvent, an $R1$-factor of 9.24% was achieved. The 15 largest residual electron density peaks (in the range $0.5 - 1.3\,\mathrm{e}^{-}\,\mathrm{\AA}^{-3}$) are all in the channel parallel to the c-axis (Figure 4.5a). The solvent masking procedure found 3 symmetry equivalent voids parallel to the c-axis, giving a total solvent accessible volume of $3687\,\mathrm{\AA}^{3}$ per unit cell (24% of the unit cell volume) and an estimated electron count of 866 electrons per unit cell. Figure 4.6 shows the electron density map for F^{diff} as displayed in Olex2. A much improved $R1$-factor of 4.60% was obtained for 2564 reflections $I \geq 2u(I)$. No significant residual electron density was observed in the voids and the highest residual electron density peaks ($< 0.4\,\mathrm{e}^{-}\,\mathrm{\AA}^{-3}$) were in the vicinity of the atomic model (Figure 4.5b).

4.7 Discussion

It is encouraging that for Compound I the set completion technique gives much improved results for the cases where there are missing reflections. Almost identical results are obtained compared to the complete data set, even for the case where almost 5% of the reflections are missing. Compounds II and III demonstrate that the set completion technique gives estimated electron counts closer to the expected value when used with original data sets where one or more low angle reflections are unobserved. From Table 4.2 it can be seen that only relatively small decreases in

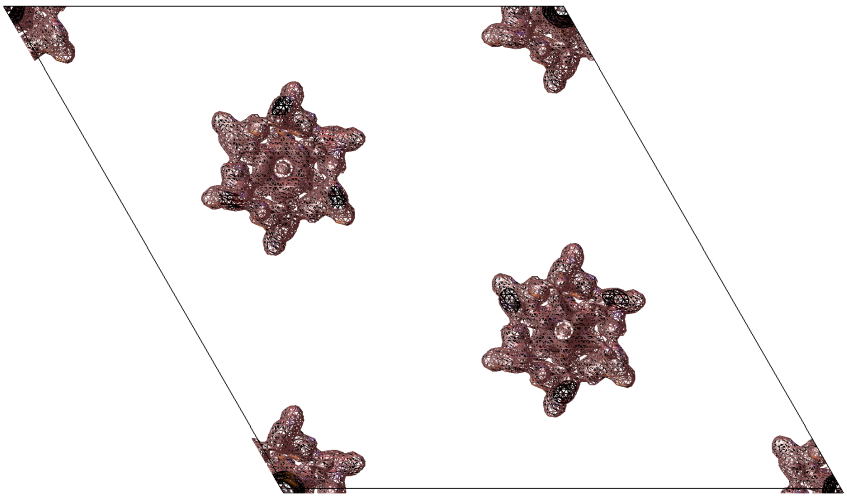


(a) The 15 highest residual electron density peaks are all found within the channel parallel to the c-axis.

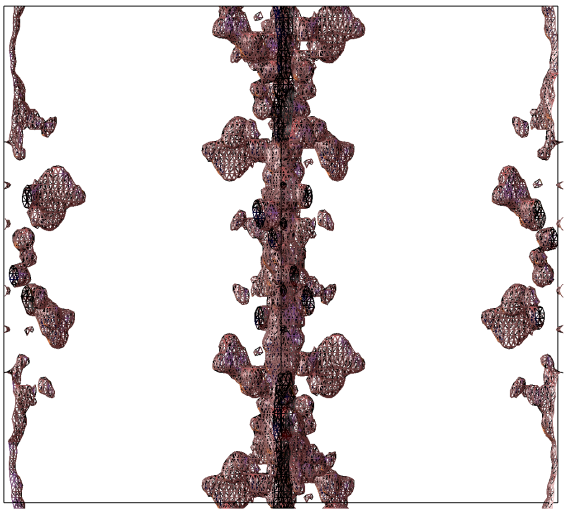


(b) After the use of a solvent mask there are no significant residual electron density peaks within the solvent channels.

Figure 4.5: The view of the unit cell down the c-axis for compound VII. The peaks in the difference electron density are displayed as transparent light-brown spheres.



(a) The view down the c-axis.



(b) The view perpendicular to the **ab** vector.

Figure 4.6: Two alternate views of electron density map for F^{diff} for compound VII.

		Atomic model		Mask		Set completion			
Compound	Solvent	R1(%)	e^-	R1(%)	e^-	R1(%)	e^-	$V_s(\text{Å})^3$	$V_s(\%)$
I	CH_3CN	0.00	44	0.88	43.4	-	-	186	20.6
I†	CH_3CN	0.00	44	1.81	40.5	0.89	43.4	186	20.6
I‡	CH_3CN	0.00	44	5.20	34.5	0.89	43.4	186	20.6
II	CH_2Cl_2	7.15	84	6.99	79.7	6.95	84.8	479	15.6
III	$(CH_3)2CO$	5.04	128	5.11	103.6	5.00	131.6	497	11.2
IV	$(CH_3)2CO$	2.64	64	2.47	63.6	-	-	309	19.4
V	CH_2Cl_2, C_6H_{14}	7.63	-	6.73	143.5	-	-	596	14.6
VI	C_5H_{12}	8.35	-	6.15	117.7	5.57	235	1103	33.1
VII	C_6H_{14}	9.24	-	4.60	865.8	-	-	3844	24.9

Table 4.1: Comparisons of the results obtained using the solvent masking procedure with and without the set completion technique and the original atomic models. V_s is the total solvent accessible volume per unit cell. Also given is the solvent accessible volume as a percentage of the unit cell volume. †The 001 reflection was rejected. ‡Approximately 5% of the reflections were discarded.

	Bond lengths (Å)		Bond Angles (°)	
Compound	RMS deltas	s.u. % decrease	RMS deltas	s.u. % decrease
II	0.00328	5.7	0.24015	5.5
III	0.00099	1.7	0.07437	1.9
IV	0.00068	6.7	0.03596	7.1
V	0.00507	18.1	0.27379	17.0
VI	0.00416	37.5	0.14702	37.0
VII	0.01527	56.8	0.78640	54.4

Table 4.2: The root mean square (RMS) differences in bond lengths and angles, and the percentage decrease in their standard uncertainties after application of the solvent masking procedure.

the standard uncertainties of the geometrical parameters are observed compared to those obtained when using an atomic model. Whilst for these test compounds a full atomic model of the solvent is undoubtedly the correct approach to take, the results show that the solvent masking procedure can give comparable results.

All three applications show significant improvements after the use of the procedure and larger decreases in the standard uncertainties are observed, ranging from 17% for Compound V to $> 50\%$ for Compound VII. After the use of the set completion technique to compensate for several missing low angle reflections, the estimated electron count obtained for Compound VI is significantly larger with an improved least squares fit. This difference in electron count could be significant when attempting to estimate the solvent composition of the crystal. Compounds V and VI were both published before the method of van der Sluis and Spek [1990] was popularised (Figure 4.1) and consequently arbitrary atoms of varying types and occupancies were used to model the severely disordered solvents. Given that the current procedure gives demonstrable improvements in both the least squares fit and the geometry of the ordered part of the structure whilst also giving further information in terms of the estimated electron count which may be useful in identifying the solvent composition (in conjunction with ancillary information), it is the author's opinion that it is more meaningful to use the solvent masking procedure in such cases, provided the use and outcomes of the procedure are clearly reported.

Chapter 5

The Crystallographic Information Framework (CIF)

5.1 iotbx.cif

5.1.1 Introduction

The CIF (Crystallographic Information File) syntax [Hall et al., 1991] has become firmly established [Brown and McMahon, 2002] as the file format for deposition and archiving of small molecule crystal structures, and increasingly their structure factors. Whilst the PDB is still the preferred file format for deposition of macromolecular crystal structures, the CIF format is nonetheless important to macromolecular software through their extensive use of the PDB chemical components[1] and REFMAC monomer libraries [Vagin et al., 2004]. The IUCr maintain CIF dictionaries for describing the results of powder diffraction [Toby, 1998] and electron density studies [Mallinson, 2003], and for describing incommensurately modulated crystal structures [Madariaga, 2002]. The Crystallography Binary Format (CBF) and Image-supporting Crystallographic Information File (ImgCIF) [Hammersley et al., 2003] are extensions to the CIF format to support inclusion of binary data in the CIF, in particular raw experimental data from area detectors. The CIF is probably one of the most well known file formats within the field of chemistry,

[1] http://www.wwpdb.org/ccd.html

since it is predominantly the form in which synthetic chemists receive the results of a crystal structure analysis carried out on their behalf.

Evidently the CIF is intrinsically involved in a wide variety of crystallographic applications from data collection to publication and archiving of the outcomes of crystallographic studies. In addition there is a wealth of crystal structure coordinates and reflection data freely available in the CIF format through the Crystallographic Open Database (COD) [Gražulis et al., 2009] and the large quantity of data available as supplementary material for papers published in IUCr journals, for which many possible uses can be imagined. As such it is vital for a crystallographic library such as the cctbx to provide high quality tools for reading, creation and manipulation of CIFs, and extraction of crystallographic data from them.

Several CIF programming libraries have been developed for various languages, including FORTRAN [Hall and Bernstein, 1996], C [Ellis and Bernstein, 2001; Westbrook et al., 1997], Objective C [Chang and Bourne, 1998], .NET [Lin, 2010], Perl [Bluhm, 2000] and Python [Hester, 2006]. For some time PyCIFRW [Hester, 2006] has been distributed with the cctbx source code bundles, however there was only limited support within the cctbx for PyCIFRW. Since the parsing of CIF files in PyCIFRW occurs using an interpreted language (Python [Python Software Foundation]), parsing of extremely large CIFs (*e.g.* reflection files, dictionary files) can be considerably slower than when using comparable compiled parsers. As a result there existed several partial CIF parsers within the cctbx, each hand-crafted to suit the specific task in hand (separate tools for reading the PDB chemical components and REFMAC monomer libraries; as part of the phenix.cif_as_mtz tool; for reading fcf reflection files as output by SHELXL [Sheldrick, 2008]).

During the development of the tools described in earlier chapters within the context of the smtbx and Olex2, it became apparent that the CIF format would play a central part in presenting the results of the procedures developed, in addition to a need for providing an interface for managing the contents of the CIF within Olex2. Therefore it was decided to implement a new CIF framework within the iotbx (input/output toolbox) module of the cctbx.

Given the availability of a clearly defined formal grammar for the CIF syntax[1], it was decided to use the ANTLR parser generator [Parr, 2007] for generation of a

[1] http://www.iucr.org/resources/cif/spec/version1.1/cifsyntax

lexer and parser from a formally defined grammar. ANTLR was chosen because of its support for multiple programming languages, in particular its support of Python and C/C++. In addition, the associated ANTLRWorks GUI development environment features a number of tools that aid the development of grammars, such as visualisation of syntax diagrams and rule dependency graphs. This enabled the majority of the development to be focused on the design of the internal representation of the CIF model, whilst ensuring that the resulting parser closely follows the formal CIF grammar. The code is structured in such a way that the parser is quite distinct from the model, meaning an alternative representation of the model could be used with the same parser, and conversely a different parser could be used to populate the existing iotbx.cif model. The CIF grammar in ANTLR format is included in Appendix B. This grammar is suitable for generating CIF parsers in any of the programming languages supported by ANTLR (including C, C#, Objective C, Java, JavaScript, Python, Ruby). Figure 5.1 shows a simplified rule dependency graph generated for the CIF grammar using the ANTLRWorks GUI.

5.1.2 Using iotbx.cif

Developers familiar with the built-in dictionary type of the Python programming language [Python Software Foundation] will be immediately at home with the syntax of the iotbx.cif representation of the CIF model.

The top level object is iotbx.cif.model.cif, which is the type equivalent to a full CIF file. This contains zero or more data blocks, which are accessed by data block name using the traditional Python dictionary square brackets notation for accessing a dictionary by key. Using a valid data block name, this returns a CIF data block of the type iotbx.cif.model.block. A CIF data block consists of a sequence of data items and associated values. A data item can be associated with either one value, or a list of values (as part of a CIF loop), and a given data item can only be found once per data block. These values can in turn be accessed using the square bracket notation to retrieve the value(s) associated with a specified data item (tag).

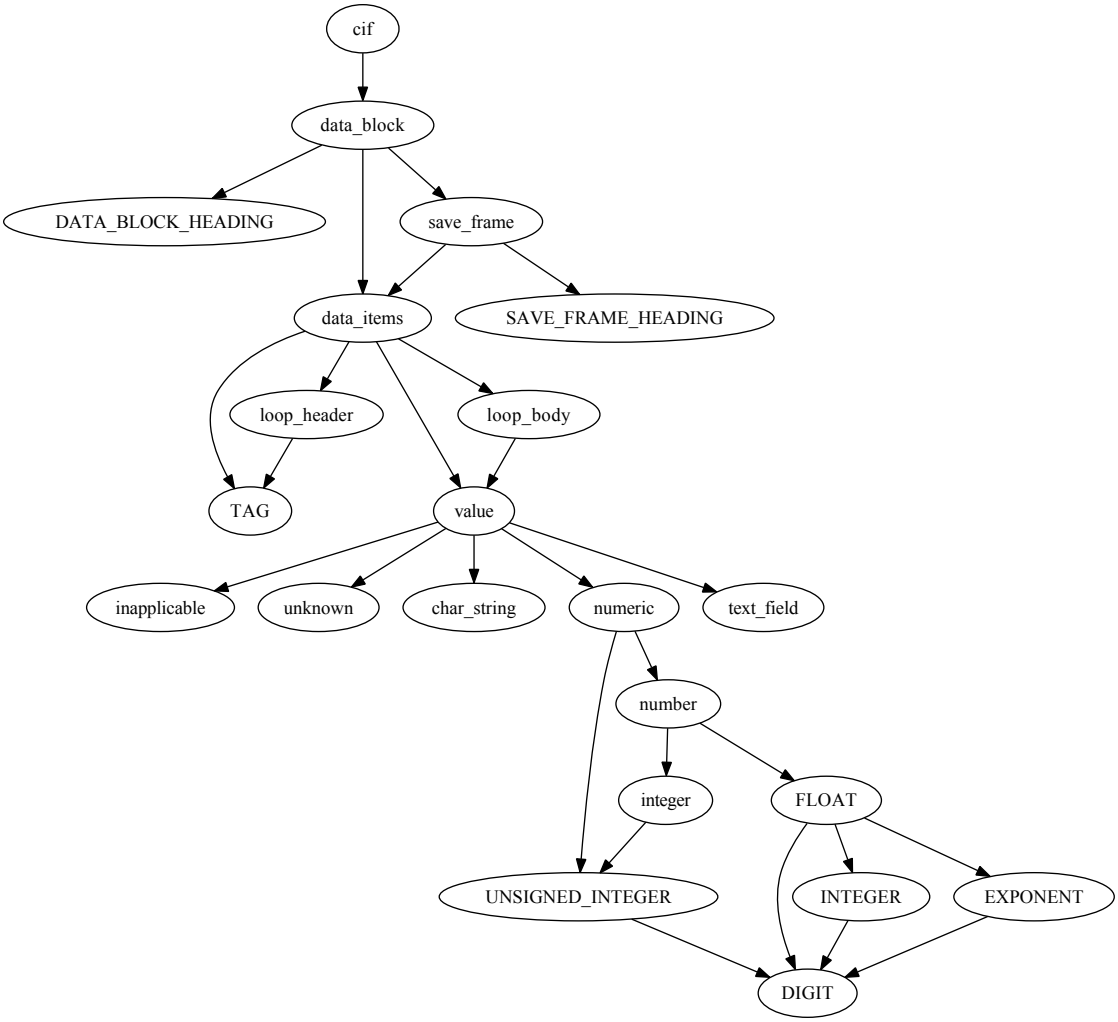


Figure 5.1: A simplified rule dependency graph for the CIF grammar.

```
import iotbx.cif
cif_model = iotbx.cif.reader(file_path='myfile.cif').model()
cif_block = cif_model['my_block_name']
hall_symbol = cif_block['_space_group_name_Hall']
```

Then:

```
>>>print hall_symbol
-P 2yn
```

Looped items are stored by columns, and the full list for a given looped item can be accessed by the data name as shown below.

```
>>>sym_ops = cif_block['_space_group_symop_operation_xyz']
>>>for sym_op in sym_ops:
...   print sym_op
x,y,z
-x+1/2,y+1/2,-z+1/2
-x,-y,-z
x-1/2,-y-1/2,z-1/2
```

The full loop object can be extracted by the name of the loop. The name of the loop is taken to be the longest common substring starting with an underscore character, and followed by (but not including) an underscore (in the case of DDL1 CIFs) or point (in the case of DDL2 CIFs) character separator. This follows the IUCr guidelines for reserved prefixes for local dictionary extensions[1]. Once a loop has been extracted, this can then be used to iterate through by rows, or to add further rows or columns to the loop. The following example demonstrates the creation of a CIF loop describing the given distance restraints.

```
loop = model.loop(header=(
  "_restr_distance_atom_site_label_1",
  "_restr_distance_atom_site_label_2",
  "_restr_distance_site_symmetry_2",
  "_restr_distance_target",
  "_restr_distance_target_weight_param",
  "_restr_distance_diff"))
for proxy in proxies:
  restraint = geometry_restraints.bond(
    unit_cell=unit_cell,
```

[1] http://www.iucr.org/resources/cif/spec/ancillary/reserved-prefixes

```
    sites_cart=sites_cart,
    proxy=proxy)
  loop.add_row((
    site_labels[proxy.i_seqs[0]],
    site_labels[proxy.i_seqs[1]],
    space_group_info.cif_symmetry_code(proxy.rt_mx_ji),
    restraint.distance_ideal,
    math.sqrt(1/restraint.weight),
    restraint.delta))
cif_block.add_loop(loop)
```

5.1.2.1 CIF output

CIF objects (model.cif, model.block, model.loop) can be exported in several ways. The simplest way is using the Python "print" statement as follows:

```
f = open('myfile.cif', 'w')
print >> f, my_cif_model
f.close()
```

The show() method of the CIF objects allows more fine-tuning of the output, including the amount of indentation used for looped data and the width of the data name field. For more advanced formatting, a Python formatting string can be provided to control the output of individual loops (in contrast to the default behaviour where items are single space-separated). The following example demonstrates the use of the tools provided by iotbx.cif to output reflection data in a format similar to that output by SHELXL using the 'LIST 4' instruction:

```
f = open('myfile.fcf'), 'w')
cif = iotbx.cif.model.cif()
mas_as_cif_block = iotbx.cif.miller_arrays_as_cif_block(
  f_calc_sq, array_type='calc')
mas_as_cif_block.add_miller_array(f_obs_sq, array_type='meas')
cif['my_block_name'] = mas_as_cif_block.cif_block
format_string='%-4i'*3 + '%-12.2f'*2 + '%-10.2f'
cif.show(out=f, loop_format_strings={'_refln':format_string})
f.close()
```

Below is an example of part of the _refln_* loop generated by this short script.

```
loop_
  _refln_index_h
  _refln_index_k
  _refln_index_l
  _refln_F_squared_calc
  _refln_F_squared_meas
  _refln_F_squared_sigma
   0   0   1       129.73       59.30       3.63
   0   1   0       157.87      157.99       4.45
   0   1   1       176.00      185.32       3.75
   1   0   0       142.63      141.28       2.62
   0  -1   1      2024.44     2010.97      30.75
```

5.1.3 Validation of CIFs against data dictionaries

Successful parsing without errors of a given CIF indicates only that it is syntactically correct. CIF dictionaries allow for a machine-readable formal description of allowed data items, and for possible restrictions on the attributes of their associated values. A collection of application-specific dictionaries are maintained by the Committee for the Maintenance of the CIF Standard (COMCIFS), and can be used to validate the contents of a given CIF. The CIF data dictionaries abide by the CIF syntax, with two distinct dictionary definition languages (DDL1 and DDL2) currently in use.

In the context of iotbx.cif, once loaded, a CIF can be validated as follows:

```
from iotbx.cif import validatation

cif_dic = validation.smart_load_dictionary(name='cif_core.dic')
cif_model.validate(cif_dic, show_warnings=True)
```

The smart_load_dictionary function allows for a dictionary to be loaded from a variety of sources, including a locally stored version, downloaded from an arbitrary URL, or via lookup in a cif dictionary register[1] allowing use of the most up-to-date version of the dictionary. A list of potential errors and warnings found during the validation is output by the procedure. The error handling is designed such that it

[1] e.g. ftp://ftp.iucr.org/pub/cifdics/cifdic.register

is possible for an application making use of iotbx.cif to override the default error handler with one specific to the needs of the application.

5.1.4 Interconversion with cctbx crystallographic objects

An essential part of any crystallographic library or software is a means to easily export/extract crystallographic information to/from the CIF format. As such, two key crystallographic objects in the cctbx, namely xray.structure and miller.array, have methods enabling easy interconversion of either object with a CIF.

```
from cctbx import miller, xray

xray_structure = xray.structure.from_cif(file_path='my.cif')
xray_structure.show_summary()
miller_arrays = miller.array.from_cif(file_path='my.fcf')
f_calc_sq = miller_arrays['_refln_F_squared_calc']
f_obs_sq = miller_arrays['_refln_F_squared_meas']
f_obs_sq.show_comprehensive_summary()
```

Tools have been developed in order to support output of the requisite structural information for publication of a structure determination. This includes the export of an xray.structure into CIF format, and also the inclusion of geometrical features such as bonds and angles. Optionally the covariance matrices for the refined parameters and for the unit cell parameters can be provided to enable the calculation of standard uncertainties for both refined and derived parameters (see also §2.3).

Support was recently added for the new restraints CIF dictionary [Brown and Guzei, 2011] which is intended to allow for the description in CIF format of the restraints and constraints used in a least-squares refinement.

5.1.5 Performance

Since the program uses a compiled parser rather than a pure Python parser, it is expected that parsing would be of sufficient speed to make handling of large CIF files acceptable, particularly for the case of reading in files containing structure factors. To test the performance of the parser and the procedures extracting

File ext.	No. files	Build errors	Parsing errors	CPU time (s)	Average (ms)
cif	136405	-	3	2409	18
cif	136405	1943	3	3549	26
hkl	13738	-	21	1845	134
hkl	13738	83	21	2121	154

Table 5.1: Performance of iotbx.cif when tested on the Crystallography Open Database (COD) using a server with 4 times AMD 12-core Opteron™Processor 6174, 2.2GHz, running Fedora 14.

File name	File size	Read time (ms)		Write time (ms)		Validation time (ms)	
	(kB)	(a)	(b)	(a)	(b)	(a)	(b)
fg3210.cif	25	26	34	20	25	41	47
fg3210CPsup2.hkl	84	43	51	117	150	48	58
4hhb.cif	705	733	891	1830	2309	2023	1955
cif_core_2.4.1.dic	469	125	191	78	102	254	298
mmcif_std_2.0.09.dic	1717	729	1057	356	461	1549	2018

Table 5.2: Execution times on (a) Intel®Core™2 Duo E6750 PC, 2.66 GHz, 3GB RAM running Windows Vista, and (b) Server with 4 times AMD 12-core Opteron™Processor 6174, 2.2GHz, running Fedora 14.

crystallographic information, a short script was run over all the CIF files in the Crystallography Open Database (COD) [Gražulis et al., 2009].

A total of 136405 CIF files were parsed in 2409 seconds of CPU time, at an average of 18 ms per file. 3 files were found to contain syntax errors; the remaining all parsed successfully. The procedure was repeated constructing instances of xray.structure; a total of 134462 instances were successfully constructed in 3549 seconds of CPU time, at an average of 26 ms per file. Table 5.1 also gives the results obtained when the procedure was run over 13738 CIF format reflection files found in the COD. The average parsing time for a reflection file was 134 ms, increasing to 154 ms when construction of a miller.array was attempted after parsing. Table 5.2 gives the performance of iotbx.cif tools on typical small molecule and protein data files and two selected dictionary files.[1]

The results show good performance for both the parser and the procedures extracting crystallographic information from the CIF model. With the increasing availability of multi-core processors, it is clear that, in conjunction with the large

[1] 4hbb.cif was downloaded from the PDB website [Fermi et al., 1984]; fg3210.cif and fg3210CPsup2.hkl were obtained from the IUCr website [Yufit and Howard, 2011].

number of tools provided by the cctbx, the iotbx.cif is suitable for performing large-scale analyses of crystal structures, since the overhead of reading structures from CIFs is minimal.

The performance of CIF output for files containing loops with a large number of values can be improved significantly by using the advanced loop formatting option described in §5.1.2.1, since each value will no longer be checked individually to determine if quoting of the value is necessary.

5.1.6 Common CIF syntax errors and error recovery

As a result of comprehensive testing of the iotbx.cif parser a number of commonly encountered syntax errors were identified, some of which are listed below. Among the sources of CIF files used are the Crystallography Open Database (COD), a selection of CIFs obtained from the websites of IUCr journals, the PDB chemical components library, the REFMAC monomer library and an in-house database of crystal structures.

1. Missing starting or closing quotes.
2. Missing starting and closing quotes for a string containing whitespace.
3. Some text prepended to CIF but not using CIF comment format (*e.g.* CheckCIF output).
4. Mismatching semicolons.
5. More than one data value per tag.
6. Missing data value for a tag.
7. Incomplete CIF - *e.g.* missing data block heading
8. Intended data block heading contains whitespace or illegal character(s). This can happen if a program uses the file name as the data block heading when creating a CIF but does not remove/substitute whitespace or illegal characters.

9. Non-ASCII characters - data values have been “copy and pasted” from other sources, for example, this could be an author’s name or place name.

10. Unquoted string with ’[’ as the first character.

11. Wrong number of values for loop.

Item 10 was the syntax error most commonly observed in the publicly available databases (*i.e.* excluding the in-house database). The CIF grammar explicitly forbids the characters ‘[’ and ‘]’ from being the first character of an unquoted string[1]:

> Matching square bracket characters, ’[’ and ’]’, are reserved for possible future introduction as delimiters of multi-line data values. At this revision of the CIF specification a data value may not begin with an unquoted left square bracket character ’[’. (While not strictly necessary, the right square bracket character ’]’ is restricted in the same way in recognition of its reserved use as a closing delimiter.)

It appears that the syntax checking routines used by the IUCr and the COD, and also CheckCIF/PLATON do not currently consider this a syntax error, which is in conflict with the formal definition of the CIF grammar.

The most commonly encountered syntax error for CIF format reflection files is item 11, although this error can affect any CIF containing a loop. The number of values in a loop must be an exact multiple of the number of tags in the loop header and it is an error if this is not the case. This is probably the hardest error to diagnose since it is not associated with a specific line number, only the particular loop, which may be many thousands of lines long in the case of reflection data, and hence the entire loop is rendered invalid. Frequently this error can be attributed to manual editing of the file resulting in one or more value being accidentally deleted. More worryingly, it is occasionally the result of a program outputting the data in fixed-field format when one of the values takes up the full width of its fixed field, losing a whitespace separator in the process. One such example can be found on line 3708 of the file sk1436Isup2.hkl which can be obtained from the supplementary materials accompanying Picard et al. [2001]:

[1] http://www.iucr.org/resources/cif/spec/version1.1/cifsyntax#general

line 3708: 0 0 42 40917268.00 43615084.002105532.75 o

This highlights one of the dangers of attempting to combine fixed-width formatting with a whitespace separated file format such as the CIF.

Some of the syntax errors outlined above are to varying extents recoverable parsing or lexing errors. Missing quotes potentially can be detected and missing tokens inserted when an end-of-line (EOL) character is encountered, since a quoted string can not extend past an EOL character. For errors such as multiple values for a tag that is not part of a loop, or a tag with no value given, the parser may recommence parsing at the next valid token it finds, discarding those invalid tokens. For invalid characters (items 9 and 10) the invalid characters can either be accepted, or the offending tag-value pair can be discarded (the current implementation does the latter). The most dangerous error is that of a missing closing semicolon for a semicolon text field, since the rest of the file up until the end-of-file (EOF) character is consumed as part of the semicolon text field. Upon reaching the EOF character an error is emitted by the lexer, but recovery from this error is not possible.

On the one hand, it may be desirable for a program to be as accommodating of errors as possible on input whilst ensuring that the output is as correct as possible. On the other hand, there are clear advantages in having software that raises clear errors when syntax errors are encountered, as this would discourage the proliferation of incorrectly formed CIFs. Indeed, one would not expect a computer code compiler to be accepting of errors, syntax or otherwise, in source code files.

The syntax errors observed generally fall into two categories: either as the result of manual editing of the CIF introducing some error; or some bug or oversight on behalf of the crystallographic software used to create or manipulate the CIF introduces a syntax error. The latter category of error can be fixed easily if software authors are aware of any potential pitfalls in CIF output. It should be the aim of every crystallographic program that creates CIFs to ensure that they are syntactically correct. Errors that are introduced by manual editing can be eliminated if there exists software that provides the means to manage the information output in the CIF with minimal effort. The following section describes the steps that have been made to address both of these issues within Olex2. If further editing of the

CIF is required, it would be preferable to use a dedicated tool such as enCIFer [Allen et al., 2004] or publCIF [Westrip, 2010] instead of manual editing of the file since these provide comprehensive syntax and dictionary validation in addition to many other tools to aid preparation of the final CIF.

5.1.7 Discussion

With the addition of the iotbx.cif module, the cctbx module now comprehensively supports most major small molecule and macromolecular crystallographic file formats (SHELX ins/res and hkl, CIF, PDB[1], CCP4 maps[2], X-PLOR format[3], MTZ format[4] and others).

The iotbx.cif module is now used heavily in the Olex2 software, as described in the following section. Additionally, the tools provided by the iotbx.cif module are currently being used extensively, in conjunction with the COD as a source of structural models and associated reflection data, in the evaluation of different approaches to minimization, including full-matrix, conjugate gradient, lbfgs and new algorithms under development [Grosse-Kunstleve, 2011].

5.2 CIF as a publication and archiving format

Publication of a small molecule crystal structure usually requires the submission of the results of the structural determination in a CIF format file, with increasingly the structure factors also required in CIF format. In addition, publication within an IUCr journal usually requires the use of the IUCr full validation suite (*checkCIF/PLATON*[5]), which performs a large number of consistency checks to highlight any potential errors in the structure determination or reporting of the results [Spek, 2009].

A published CIF will contain information from numerous sources at various stages of the crystal structure determination process: data collection, data pro-

[1] http://www.wwpdb.org/documentation/format32/v3.2.html
[2] http://www.ccp4.ac.uk/html/maplib.html
[3] http://psb11.snv.jussieu.fr/doc-logiciels/msi/xplor981/formats.html
[4] http://www.ccp4.ac.uk/html/mtzformat.html
[5] http://journals.iucr.org/services/cif/checking/checkfull.html

cessing, structure solution, structure refinement and molecular graphics software, to name but a few. Tools for aggregating information related to the structure at hand were developed within the Olex2 software and are available as part of the Report module. Relevant information is automatically extracted from experimental files present in the working directory (*e.g.* numerous Bruker-specific output files, Agilent *.cif_od, Rigaku CrystalClear.cif). Further information can be entered through the GUI where there are sections containing fields relevant to the diffraction, crystal and publication sections of the CIF. Details entered during the report stage about authors, journals and diffractometers are stored locally, to avoid having to input the same data repeatedly for different structures (as most people tend to collaborate with the same people regularly, and only have access to a limited range of diffractometers).

A complete list of data items extracted and managed by Olex2 can be viewed in CIF format using an internal text editor, where items can be edited or removed, or new CIF items added, with Olex2 storing the changes, before merging with the CIF file generated by the refinement program. Both the information harvested from the experimental files and that entered manually by the user are stored separately to the CIF from the refinement, meaning that if it later becomes necessary to re-refine or even re-solve a particular structure, all the information can still be merged with the final CIF.

After refinement using smtbx-refine within Olex2, a CIF is created using the tools described in §5.1 containing structural information as detailed in §5.1.4 and, also details of the intensities recorded and details regarding the refinement of the structural parameters. A CIF format reflection file can also be output after refinement if required.

If the solvent masking procedure (see §4) was used in the refinement, details of the procedure are also included into the final CIF.

In summary, we aim to ensure a CIF file output by Olex2 is as complete and correct as possible and ready for publication and/or archiving with minimal effort. The iotbx.cif plays a central role in this, both in the tools it provides for exporting crystallographic information from the cctbx, and in the role it plays in the management of CIF items within Olex2.

Chapter 6

Concluding Remarks

Described within this thesis are numerous contributions that have been made in various areas of crystallographic computing as part of the EPSRC-funded project, *Age Concern: Crystallographic Software for the Future*. Tools have been implemented within the smtbx (small molecule toolbox), and made available to crystallographic users through the Olex2 software [Dolomanov et al., 2009a].

As part of the new full matrix least squares refinement program being developed within smtbx/Olex2 under the project, a framework was implemented enabling the inclusion of observations of restraints in the refinement. Pre-existing cctbx restraints were adapted to conform with the new framework, and new restraints on geometry and anisotropic displacement parameters were added. The geometrical restraints were extended to allow for symmetry equivalent atoms. Support was added in the new iotbx.cif module for inclusion of the restraints in CIF format according to the recently created CIF restraints dictionary.

Calculations of errors on derived parameters such as bond lengths and angles is an essential part of the preparation of a small molecule crystal structure for publication. In conjunction with the iotbx.cif module, tables of bond lengths and angles and their associated errors are now included in the CIF output after refinement with smtbx-refine in Olex2.

Refinement of (pseudo-)merohedrally twinned structures was implemented, which also enables the refinement of the Flack x parameter as part of the determination of absolute structure.

In §3.1 a brief outline was given of the evolution of methods for the determi-

nation of absolute structure through the use of anomalous scattering. The probabilistic approach to absolute structure determination developed by Hooft et al. [2008] was implemented, both using the Gaussian distribution and Student's t-distribution to model the experimental errors. It was shown that it is preferable to use the Student's t-distribution as the error model, rather than an arbitrary outlier cutoff which can bias the results of the procedure.

134 non-centrosymmetric structures were analysed in order to compare the results obtained using the new probabilistic procedures with those obtained from the refinement of the Flack x parameter [Flack, 1983]. It was shown that the Hooft y parameter usually gives comparable values to the Flack x parameter, but frequently has a lower standard uncertainty, which may increase the confidence with which a conclusion on the absolute structure can be made.

The determination of absolute structure by both methods is now automatically carried out after refinement of non-centrosymmetric structures in Olex2. Also discussed were the new graphs for analysis of reflection statistics that have been implemented and made available through Olex2.

The procedure of van der Sluis and Spek [1990], intended for improved refinements of crystal structures affected by severely disordered solvent, was implemented in the smtbx. A fast void search routine is used which can lead to significant speed improvements for large, high symmetry structures. A modification to the procedure was proposed and implemented, which had demonstrable improvements on the results obtained when one or more low angle reflections were missing. Several test cases were used to verify that the procedure gave results comparable with those obtained with a standard atomic solvent model, and three applications of the procedure showed the significant improvements that were obtained in the refinement of the ordered part of the structure for cases where severe disorder meant that an atomic solvent model was not possible.

Finally, a new library was added to the cctbx to provide an interface between the cctbx and the Crystallographic Information Framework (CIF) file format. A fast parser was created from the formal definition of the CIF grammar using the ANTLR parser generator. Interconversion between cctbx crystallographic objects and the CIF format was added, and also validation of CIFs against CIF data dictionaries. A discussion of the commonly encountered syntax errors gave several

examples and pointed out potential reasons why errors may appear in published CIF files. The iotbx.cif is now relied upon heavily by the Olex2 software, and is actively being used to aid the development of new minimisation algorithms.

As a whole, the "Age Concern" project has provided a solid foundation for future developments in small molecule crystallographic computing. In the smtbx we have developed a modern and extensible framework for the solution and refinement of small molecule crystal structures that provides much of the functionality of commonly-used refinement programs [Betteridge et al., 2003; Sheldrick, 2008]. We have implemented some of the latest ideas and algorithms in the literature, including the charge flipping algorithm for structure solution [Oszlányi and Süto, 2008], and probabilistic approach to absolute structure determination of Hooft et al. [2008, 2010]. We have also added further tools in the cctbx which we hope will prove useful to the wider crystallographic software developer community. The code is open source and hosted on SourceForge[1], which we hope will encourage contributions from other developers in the future.

In Olex2 we have provided a reference application that allows new developments to be made available rapidly to a wide audience. In combination with the structure solution and refinement tools provided by the smtbx we have a program that can take a crystal structure determination from structure solution and refinement through to publication of the results.

Whilst a considerable amount has been achieved by the project, there is a much greater area of crystallographic computing that has thus far been unexplored by the project. Potential areas for future work range from small projects, which could include adding support for the quotient restraints suggested by Parsons and Flack [2004] for improved absolute structure determination or support for refinement against multiple datasets, to much larger projects, such as the addition of aspherical form factors both for use with a library of multipole parameters [Coppens and Volkov, 2004; Dittrich et al., 2006a] or charge density refinement [Hansen and Coppens, 1978].

[1] http://cctbx.sourceforge.net/

Appendix A

Absolute Structure Results

Table A.1: Bijvoet pair analysis for 134 non-centrosymmetric structures using Gaussian statistics. An asterisk indicates the use of copper radiation.

Structure	Hooft y	$\sigma(y)$	$P2(false)$	$P3(true)$	$P3(false)$	$P3(twin)$	Slope	Corr. Coeff.
06avc06	-0.0080	0.0134	0	1	0	0	0.769	0.9997
06avc07	0.0148	0.0116	0	1	0	0	0.876	0.9996
06avc13	0.0181	0.0236	0	1	0	0	0.957	0.9998
06avc16	-0.7199	0.5943	0.031	0.778	0.025	0.197	1.011	0.9998
06avc17	-0.0454	0.0149	0	1	0	0	1.217	0.9956
06avc18	-0.2123	0.5691	0.1	0.625	0.069	0.306	0.987	0.9994
06avc19	0.0225	0.0159	0	1	0	8.34E-197	0.980	0.9998
06avc32	-0.0041	0.0223	0	1	0	2.34E-111	0.966	0.9999
06avc35	-0.0033	0.0181	0	1	0	5.49E-168	1.028	0.9996
06srv042	0.0155	0.0185	0	1	0	0	1.022	0.9993
06srv055	-0.1517	0.1940	0	0.995	0	0.005	1.082	0.9839
06srv134	-0.0235	0.0551	0	1	0	0	0.842	0.9839
06srv148	0.0083	0.0393	0	1	0	0	1.081	0.9982
06srv151	0.0241	0.0496	0	1	0	0	1.013	0.9998
06srv155	0.4265	0.0565	0	0	0	1	1.652	0.9987
06srv178	-0.0456	0.0126	0	1	0	0	0.932	0.9992
06srv179	-0.0680	0.0366	0	1	0	0	0.771	0.9992
06srv184	-0.0425	0.0121	0	1	0	0	0.812	0.9989
06srv185	0.0342	0.0484	4.92E-087	1	4.92E-087	1.02E-020	0.832	0.9971
06srv191	0.2742	0.2490	0.026	0.446	0.012	0.542	0.944	0.9998
06srv220	-0.0343	0.0081	0	1	0	0	0.988	0.9998
06srv222	-0.0408	0.1138	0	1	0	0	1.123	0.9976
06srv227	0.0019	0.0033	0	1	0	0	1.018	0.9997
06srv230	0.1814	0.0210	0	1	0	0	0.926	0.9933
06srv241	0.6094	0.9169	0.532	0.296	0.337	0.367	0.799	0.9992
06srv243	0.2303	0.0177	0	1	0	0	1.003	0.9993
06srv263	-0.2325	0.4827	0.041	0.715	0.031	0.254	0.754	0.9996
06srv264	0.1897	0.3228	0.048	0.556	0.028	0.416	0.873	0.9994

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
06srv274	-0.0222	0.0259	0	1	0	0	0.985	0.9986
06srv290	-0.0309	0.0198	0	1	0	0	1.013	0.9999
06srv320*	0.0329	0.0113	0	1	0	0	1.351	0.9938
06srv321*	0.0898	0.0111	0	1	0	3.32E-283	1.069	0.9944
06srv325	0.4190	0.3314	0.324	0.275	0.132	0.593	1.010	0.9995
06srv329	0.1614	0.0343	0	1	0	0	1.027	0.9996
06srv359	-0.0089	0.0276	0	1	0	0	1.005	0.9999
06srv364	0.5294	0.0100	n/a	0	0	1	1.012	0.9995
06srv370	-0.0981	0.1809	0	0.995	0	0.005	1.292	0.9978
06srv377	0.4582	0.0053	n/a	0	0	1	0.977	0.9996
06srv380	-0.0204	0.0094	0	1	0	0	0.982	0.9960
06srv387	-0.0051	0.0071	0	1	0	0	0.783	0.9995
07srv015	-0.0314	0.0091	0	1	0	0	0.812	0.9996
07srv039	-1.0638	1.4550	0.323	0.452	0.216	0.332	0.721	0.9985
07srv040	0.1433	0.0031	0	1	0	0	2.210	0.9416
07srv042	-0.0272	0.0095	0	1	0	0	0.879	0.9982
07srv043	0.0052	0.0203	0	1	0	0	1.564	0.9854
07srv062	0.1956	0.1115	0	0.899	0	0.101	1.097	0.9965
07srv067	-0.0321	0.3849	0.027	0.707	0.019	0.273	0.964	0.9998
07srv073	-0.0028	0.0042	0	1	0	0	0.839	0.9998
07srv077	-0.2518	0.3329	0.001	0.905	0.001	0.094	0.895	0.9996
07srv079	-0.0281	0.0064	0	1	0	0	1.030	0.9998
07srv090	0.0239	0.0485	0	1	0	0	1.105	0.9990
07srv094	0.3053	0.3599	0.182	0.406	0.09	0.503	1.065	0.9482
07srv100	-0.0438	0.0113	0	1	0	0	0.914	0.9992
07srv129	-0.0115	0.0079	0	1	0	0	0.892	0.9998
07srv142	0.8226	0.4777	0.804	0.116	0.477	0.407	0.935	0.9999
07srv157	0.0326	0.0093	0	1	0	0	0.813	0.9979
07srv161	0.4321	0.0240	0	0	0	1	1.313	0.9930
07srv167	-0.0027	0.0103	0	1	0	0	0.750	0.9997
07srv170	0.0164	0.0536	0	1	0	0	0.926	0.9999

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
07srv172	-0.6411	0.8142	0.152	0.592	0.106	0.302	0.870	0.9993
07srv205	-0.0322	0.0048	0	1	0	0	0.963	0.9998
07srv244	0.0780	0.0230	0	1	0	0	0.896	0.9997
07srv258	-0.0137	0.0272	0	1	0	0	1.273	0.9985
07srv275	0.4256	0.0067	n/a	0	0	1	0.963	0.9993
07srv288	-0.2090	0.3245	0.001	0.897	0.001	0.101	0.879	0.9969
07srv295	0.1843	0.0276	0	1	0	0	0.939	0.9992
07srv299	-1.2650	0.7821	0.053	0.743	0.041	0.215	1.089	0.9990
07srv310	0.2779	0.0062	0	2.47E-157	0	1	0.880	0.9995
07srv312	-4.2008	3.3647	0.398	0.403	0.266	0.331	1.077	0.9464
07srv331	0.0511	0.0234	0	1	0	0	0.926	0.9998
07srv345	0.4616	0.0040	n/a	0	0	1	1.036	0.9995
07srv346	0.0082	0.0137	0	1	0	5.75E-281	0.832	0.9984
07srv363	0.0478	0.0046	0	1	0	0	0.829	0.9999
07srv376	0.5600	0.0303	1	0	0	1	0.934	0.9996
07srv379	-0.0601	0.4089	0.034	0.699	0.025	0.276	0.911	0.9996
07srv387	-0.0774	0.2351	0	0.951	0	0.049	0.858	0.9991
07srv401	-0.0524	0.2162	0	0.962	0	0.038	0.862	0.9851
07srv422	0.0281	0.0134	0	1	0	0	2.294	0.9749
07srv430	0.0174	0.0065	0	1	0	0	0.801	0.9999
07srv457	-0.2467	0.2863	0	0.954	0	0.046	0.778	0.9995
07srv460	0.0228	0.0096	0	1	0	0	0.678	0.9994
07srv466	-0.0145	0.0172	0	1	0	0	0.883	0.9999
07srv484	0.0663	0.0052	0	1	0	0	0.691	0.9998
07srv489	0.0325	0.0044	0	1	0	0	0.634	0.9999
07srv490	0.0407	0.0076	0	1	0	0	0.923	0.9999
07srv513	-0.0118	0.0087	0	1	0	0	0.850	0.9996
08srv023	-0.0558	0.0139	0	1	0	0	0.904	0.9998
08srv048	-0.0129	0.0071	0	1	0	0	0.863	0.9999
08srv051	-0.0454	0.0149	0	1	0	0	1.217	0.9956
08srv079	0.0163	0.0252	0	1	0	0	1.263	0.9983

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
08srv087	0.0122	0.0427	5.78E-117	1	5.78E-117	4.64E-029	0.967	0.9999
08srv088	-0.0313	0.0109	0	1	0	0	0.855	0.9997
08srv096	0.1979	0.0066	0	1	0	4.54E-261	0.749	0.9977
08srv097	0.0042	0.0036	0	1	0	0	0.774	0.9990
08srv104	0.0318	0.0667	0	1	0	0	0.894	0.9992
08srv128	1.5327	1.0618	0.714	0.19	0.475	0.335	0.836	0.9803
08srv132	-0.1642	1.1372	0.374	0.408	0.244	0.348	0.899	0.9990
08srv145	-6.6132	2.3783	0.221	0.547	0.155	0.298	1.056	0.9787
08srv168	0.0149	0.0118	0	1	0	0	0.792	0.9881
08srv218	0.0329	0.0339	0	1	0	0	0.808	0.9977
08srv221	-0.0365	0.0073	0	1	0	0	0.986	0.9997
08srv222	0.5235	0.0278	1	0	0	1	0.899	0.9996
08srv261	0.0140	0.0144	0	1	0	0	1.512	0.9932
08srv269	0.2589	0.4260	0.209	0.437	0.116	0.448	0.946	0.9994
08srv274	0.4784	0.0084	n/a	0	0	1	0.917	0.9999
08srv287	0.4412	0.1340	0.036	0.005	0	0.995	0.959	0.9678
08srv290	0.0225	0.0080	0	1	0	0	0.888	0.9998
08srv303	-8.1923	5.5875	0.431	0.38	0.288	0.332	0.845	0.9993
08srv336	0.0378	0.0114	0	1	0	0	0.904	0.9999
08srv421	-0.0082	0.0079	0	1	0	0	0.960	0.9995
08srv439*	-0.1926	0.2110	0	0.993	0	0.007	1.141	0.9949
08srv470	-0.0362	0.0088	0	1	0	0	1.748	0.9971
09srv026	-0.0136	0.0118	0	1	0	0	0.895	0.9992
09srv028	0.0549	0.0137	0	1	0	0	0.838	0.9999
09srv052	0.0561	0.0272	0	1	0	0	0.905	0.9999
09srv071	-0.3849	0.4345	0.009	0.837	0.008	0.156	0.874	0.9993
09srv073	0.1435	0.4850	0.18	0.496	0.109	0.395	0.761	0.9948
09srv129	0.3191	0.0152	0	0	0	1	0.903	0.9997
09srv137	0.0216	0.0046	0	1	0	0	0.765	0.9998
09srv140	1.0617	0.3792	0.98	0.015	0.736	0.249	0.942	0.9999
09srv160	0.3783	0.0174	0	0	0	1	0.977	0.9999

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
09srv162	-0.0381	0.0054	0	1	0	0	0.836	0.9998
09srv210	0.0132	0.0136	0	1	0	6.92E-279	0.845	0.9998
09srv225	-0.0256	0.0130	0	1	0	0	0.912	1.0000
09srv230	-0.0499	0.1091	0	1	0	0	0.922	0.9998
09srv231	-0.1549	1.3080	0.405	0.389	0.265	0.346	0.952	0.9997
09srv238	0.4152	0.7740	0.465	0.332	0.288	0.381	0.608	0.9972
09srv254	-0.0147	0.0353	0	1	0	0	0.905	0.9966
09srv277	0.0191	0.0344	7.69E-177	1	7.69E-177	5.29E-043	0.920	0.9999
09srv288	0.1047	0.7166	0.317	0.429	0.199	0.372	0.949	0.9999
09srv313	0.0082	0.0080	0	1	0	0	0.901	0.9997
09srv356	0.0330	0.0077	0	1	0	0	0.804	0.9994
09srv380	-0.6683	0.4029	0.001	0.944	0.001	0.056	0.915	0.9994
09srv386	0.0185	0.1023	0	1	0	0	1.101	0.9520

Table A.2: Bijvoet pair analysis for 134 non-centrosymmetric structures using Student's t statistics. An asterisk indicates the use of copper radiation.

Structure	Hooft y	$\sigma(y)$	$P2(false)$	$P3(true)$	$P3(false)$	$P3(twin)$	ν	Slope	Corr. Coeff.
06avc06	-0.0078	0.0104	0	1	0	0	23.4	0.740	1.0000
06avc07	0.0144	0.0103	0	1	0	0	24.4	0.844	0.9999
06avc13	0.0194	0.0227	0	1	0	0	29.0	0.924	0.9999
06avc16	-0.7518	0.6022	0.031	0.779	0.025	0.196	38.4	0.983	0.9999
06avc17	-0.0333	0.0181	0	1	0	0	7.5	1.055	0.9996
06avc18	-0.2878	0.5641	0.078	0.66	0.056	0.284	19.7	0.930	0.9999
06avc19	0.0222	0.0156	0	1	0	1.57E-200	38.4	0.950	1.0000
06avc32	-0.0046	0.0216	0	1	0	7.23E-119	150.5	0.959	0.9999
06avc35	-0.0055	0.0188	0	1	0	1.20E-151	22.3	0.990	0.9999
06srv042	0.0144	0.0189	0	1	0	0	19.7	0.959	1.0000
06srv055	-0.1846	0.2145	0	0.991	0	0.009	4.5	0.830	0.9999
06srv134	-0.0054	0.0481	0	1	0	0	4.3	0.644	0.9998
06srv148	0.0104	0.0432	0	1	0	0	10.4	0.976	0.9999
06srv151	0.0225	0.0504	0	1	0	0	29.0	0.980	1.0000
06srv155	0.4275	0.0903	0	0	0	1	12.7	1.530	0.9999
06srv178	-0.0436	0.0119	0	1	0	0	19.7	0.872	0.9999
06srv179	-0.0577	0.0289	0	1	0	0	15.0	0.720	0.9999
06srv184	-0.0406	0.0104	0	1	0	0	13.1	0.753	0.9999
06srv185	-0.0018	0.0419	2.57E-112	1	2.57E-112	2.46E-031	7.8	0.729	0.9998
06srv191	0.2619	0.2355	0.013	0.47	0.006	0.524	75.8	0.932	0.9999
06srv220	-0.0343	0.0080	0	1	0	0	46.6	0.969	0.9999
06srv222	-0.0587	0.1282	0	1	0	0	10.4	0.999	0.9999
06srv227	0.0021	0.0034	0	1	0	0	26.5	0.982	0.9998
06srv230	0.2063	0.0202	0	1	0	0	6.0	0.800	0.9990
06srv241	0.8553	0.7257	0.663	0.211	0.414	0.375	15.0	0.747	0.9999
06srv243	0.2322	0.0180	0	1	0	0	19.7	0.945	1.0000
06srv263	-0.3010	0.3707	0.003	0.879	0.003	0.119	26.2	0.727	0.9998
06srv264	0.1823	0.2832	0.019	0.597	0.011	0.391	20.6	0.842	0.9997

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
06srv274	-0.0225	0.0255	0	1	0	0	300.0	0.980	0.9985
06srv290	-0.0308	0.0202	0	1	0	0	75.8	1.000	1.0000
06srv320*	0.0290	0.0160	0	1	0	4.52E-170	5.7	1.121	0.9999
06srv321*	0.0785	0.0132	0	1	0	2.55E-189	5.9	0.929	0.9995
06srv325	0.3905	0.3387	0.278	0.31	0.119	0.571	19.7	0.954	0.9999
06srv329	0.1565	0.0357	0	1	0	0	21.4	0.989	0.9999
06srv359	-0.0092	0.0279	0	1	0	0	52.5	0.986	0.9999
06srv364	0.5293	0.0102	n/a	0	0	1	38.4	0.980	0.9997
06srv370	-0.1201	0.2400	0	0.961	0	0.039	10.4	1.159	0.9999
06srv377	0.4577	0.0052	n/a	0	0	1	22.4	0.940	0.9999
06srv380	-0.0266	0.0095	0	1	0	0	7.2	0.848	0.9999
06srv387	-0.0055	0.0055	0	1	0	0	19.9	0.753	0.9999
07srv015	-0.0322	0.0077	0	1	0	0	21.2	0.781	1.0000
07srv039	-1.4140	1.0969	0.17	0.586	0.12	0.294	10.4	0.652	0.9998
07srv040	0.1231	0.0069	0	1	0	0	2.5	1.349	0.9994
07srv042	-0.0242	0.0089	0	1	0	0	10.9	0.814	0.9999
07srv043	0.0091	0.0295	0	1	0	0	5.7	1.220	0.9991
07srv062	0.1612	0.1174	0	0.961	0	0.039	7.0	0.956	0.9996
07srv067	-0.0327	0.3712	0.02	0.725	0.015	0.26	77.5	0.955	0.9999
07srv073	-0.0028	0.0035	0	1	0	0	38.4	0.816	0.9999
07srv077	-0.2584	0.2965	0	0.947	0	0.053	22.3	0.863	0.9999
07srv079	-0.0279	0.0066	0	1	0	0	40.7	1.010	0.9999
07srv090	0.0125	0.0548	0	1	0	0	13.0	1.025	1.0000
07srv094	-0.1588	0.3663	0.007	0.816	0.006	0.178	2.6	0.647	0.9991
07srv100	-0.0429	0.0105	0	1	0	0	16.4	0.858	0.9999
07srv129	-0.0108	0.0072	0	1	0	0	41.9	0.875	0.9998
07srv142	0.8119	0.4466	0.827	0.101	0.484	0.415	113.1	0.927	1.0000
07srv157	0.0270	0.0074	0	1	0	0	10.4	0.730	0.9999
07srv161	0.4643	0.0341	0	0	0	1	5.9	1.133	0.9996
07srv167	-0.0028	0.0077	0	1	0	0	300.0	0.747	0.9997
07srv170	0.0164	0.0497	0	1	0	0	300.0	0.922	0.9999

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
07srv172	-0.8594	0.7122	0.064	0.713	0.049	0.239	19.7	0.816	0.9999
07srv205	-0.0321	0.0046	0	1	0	0	39.4	0.944	0.9999
07srv244	0.0770	0.0205	0	1	0	0	38.4	0.870	0.9998
07srv258	-0.0025	0.0351	0	1	0	0	10.5	1.183	0.9999
07srv275	0.4250	0.0064	n/a	0	0	1	19.7	0.909	0.9998
07srv288	-0.3504	0.2922	0	0.971	0	0.029	10.4	0.776	0.9997
07srv295	0.1862	0.0259	0	1	0	0	19.7	0.880	0.9999
07srv299	-0.9722	0.8692	0.124	0.63	0.09	0.28	12.9	1.011	0.9999
07srv310	0.2775	0.0055	0	1.90E-151	0	1	19.7	0.834	0.9999
07srv312	-2.2872	3.6897	0.449	0.367	0.299	0.334	2.5	0.663	0.9976
07srv331	0.0526	0.0217	0	1	0	0	41.7	0.908	0.9999
07srv345	0.4613	0.0042	n/a	0	0	1	20.9	0.997	0.9999
07srv346	0.0066	0.0115	0	1	0	0	11.8	0.771	0.9997
07srv363	0.0477	0.0038	0	1	0	0	75.8	0.816	0.9999
07srv376	0.5577	0.0284	1	0	0	1	22.3	0.899	0.9999
07srv379	-0.0373	0.3741	0.021	0.725	0.016	0.26	19.7	0.866	0.9999
07srv387	-0.0670	0.2013	0	0.98	0	0.02	19.7	0.806	0.9999
07srv401	0.2987	0.2721	0.052	0.389	0.021	0.59	3.4	0.638	0.9938
07srv422	0.0597	0.0296	0	1	0	0	3.6	1.740	0.9991
07srv430	0.0174	0.0052	0	1	0	0	75.8	0.791	0.9999
07srv457	-0.2172	0.2283	0	0.989	0	0.011	21.2	0.750	0.9998
07srv460	0.0207	0.0064	0	1	0	0	19.7	0.644	0.9998
07srv466	-0.0144	0.0152	0	1	0	0	300.0	0.878	0.9999
07srv484	0.0658	0.0036	0	1	0	0	45.6	0.678	0.9999
07srv489	0.0325	0.0028	0	1	0	0	300.0	0.631	0.9999
07srv490	0.0400	0.0070	0	1	0	0	54.0	0.906	1.0000
07srv513	-0.0129	0.0074	0	1	0	0	23.3	0.818	1.0000
08srv023	-0.0557	0.0126	0	1	0	0	43.3	0.887	0.9999
08srv048	-0.0129	0.0061	0	1	0	0	75.8	0.850	0.9999
08srv051	-0.0333	0.0181	0	1	0	0	7.5	1.055	0.9996
08srv079	0.0139	0.0325	0	1	0	0	10.8	1.175	0.9996

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
08srv087	0.0120	0.0413	2.21E-124	1	2.21E-124	5.49E-031	300.0	0.963	0.9999
08srv088	-0.0303	0.0094	0	1	0	0	23.5	0.823	0.9999
08srv096	0.1995	0.0051	0	1	0	3.56E-280	11.9	0.693	0.9990
08srv097	0.0029	0.0027	0	1	0	0	15.0	0.722	1.0000
08srv104	0.0568	0.0595	0	1	0	0	14.5	0.834	0.9999
08srv128	0.9768	0.9845	0.621	0.244	0.4	0.356	3.6	0.639	0.9992
08srv132	-0.2500	1.0136	0.325	0.441	0.213	0.346	14.2	0.841	0.9997
08srv145	-6.5915	3.2249	0.333	0.451	0.226	0.323	3.3	0.749	0.9990
08srv168	0.0146	0.0113	0	1	0	9.96E-282	4.4	0.615	0.9998
08srv218	0.0281	0.0275	0	1	0	0	10.4	0.724	0.9998
08srv221	-0.0383	0.0072	0	1	0	0	27.6	0.951	0.9999
08srv222	0.5227	0.0251	1	0	0	1	22.7	0.865	0.9999
08srv261	0.0100	0.0220	0	1	0	0	5.9	1.307	0.9995
08srv269	0.2843	0.4066	0.213	0.42	0.114	0.466	19.7	0.892	0.9999
08srv274	0.4784	0.0077	n/a	0	0	1	300.0	0.912	0.9999
08srv287	0.5150	0.1199	0.686	0	0	1	3.3	0.659	0.9986
08srv290	0.0228	0.0071	0	1	0	0	38.4	0.864	0.9999
08srv303	-7.8521	4.7957	0.41	0.395	0.274	0.331	21.4	0.816	0.9996
08srv336	0.0377	0.0104	0	1	0	0	75.8	0.891	0.9999
08srv421	-0.0065	0.0076	0	1	0	0	21.4	0.924	0.9999
08srv439*	-0.0762	0.2392	0	0.946	0	0.054	5.8	0.995	0.9996
08srv470	-0.0332	0.0163	0	1	0	0	10.4	1.567	0.9993
09srv026	-0.0126	0.0105	0	1	0	0	19.7	0.838	0.9999
09srv028	0.0548	0.0115	0	1	0	0	300.0	0.834	0.9999
09srv052	0.0563	0.0246	0	1	0	0	300.0	0.901	0.9998
09srv071	-0.3988	0.3786	0.002	0.904	0.002	0.094	19.7	0.822	0.9999
09srv073	-0.0077	0.4489	0.075	0.621	0.05	0.328	5.9	0.663	0.9994
09srv129	0.3207	0.0137	0	0	0	1	26.2	0.869	1.0000
09srv137	0.0213	0.0035	0	1	0	0	28.0	0.739	0.9999
09srv140	1.0575	0.3571	0.988	0.01	0.762	0.228	300.0	0.937	0.9999
09srv160	0.3792	0.0170	0	0	0	1	75.8	0.964	0.9999

Continues on next page...

Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	$P2(false)$ *(cont'd)*	$P3(true)$ *(cont'd)*	$P3(false)$ *(cont'd)*	$P3(twin)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
09srv162	-0.0381	0.0045	0	1	0	0	38.4	0.809	1.0000
09srv210	0.0131	0.0116	0	1	0	0	39.6	0.830	0.9998
09srv225	-0.0260	0.0119	0	1	0	0	150.5	0.906	1.0000
09srv230	-0.0497	0.1007	0	1	0	0	300.0	0.917	0.9998
09srv231	-0.5721	1.2418	0.333	0.442	0.22	0.338	22.2	0.917	1.0000
09srv238	0.4948	0.4944	0.495	0.275	0.27	0.455	10.4	0.539	0.9999
09srv254	0.0148	0.0328	0	1	0	0	7.4	0.787	0.9998
09srv277	0.0190	0.0317	3.31E-207	1	3.31E-207	1.18E-050	225.3	0.915	0.9999
09srv288	0.1100	0.6798	0.301	0.437	0.188	0.375	89.3	0.940	0.9999
09srv313	0.0080	0.0080	0	1	0	0	300.0	0.897	0.9997
09srv356	0.0345	0.0061	0	1	0	0	19.7	0.756	0.9999
09srv380	-0.7389	0.3692	0	0.974	0	0.026	20.2	0.882	0.9997
09srv386	-0.0392	0.1199	0	1	0	0	2.4	0.687	0.9955

Table A.3: Flack parameter and miscellaneous information for 134 non-centrosymmetric structures. An asterisk indicates the use of copper radiation.

Structure	Formula	Space group	No. Bijvoet pairs	Bijvoet pair coverage (%)	Flack x	$\sigma(x)$
06avc06	C42 H50 N2 O4 Ti	P 21 21 21	4006	99	0.0092	0.0202
06avc07	C29 H48 Cl2 N2 O4 Ti	P 21 21 21	3128	98	0.0031	0.0256
06avc13	C32 H50 N2 O4 Ti	P 21 21 21	2944	100	0.0211	0.0286
06avc16	C21 H27 N O2	P 21 21 21	2045	99	-0.3093	0.9871
06avc17	C69 H66 N2 O4 Ti	P 1 21 1	4574	61	0.0017	0.0273
06avc18	C16 H27 N O2	P 21 21 21	1817	98	-0.0081	1.0078
06avc19	C63 H60 Cl2 N2 O4 Ti	P 21 21 21	6326	97	0.0304	0.0207
06avc32	C48 H50 N2 O4 Ti	P 21 21 21	3972	100	-0.0034	0.0273
06avc35	C27 H39 N O4 Ti	P 21 21 21	2841	100	-0.007	0.0284
06srv042	C14 H10 B10 Cl10	P 21 21 21	3331	99	0.0082	0.0349
06srv055	C33 H47 F3 O	P 1 21 1	7627	99	-0.0315	0.3118
06srv134	C25 H31 N3 O S	P 1 21 1	5872	100	-0.0536	0.0531
06srv148	C21 H24 N2 S	P 1 21 1	1860	81	-0.0053	0.0612
06srv151	C30 H34 O5 Si	P n a 21	3386	100	0.0708	0.0858
06srv155	C28 H29 Al2 Cl8 P3	P b a 2	7141	91	0.397	0.1196
06srv178	C42 H48 Cl4 N4 Pd2 S2	P 1 21 1	5547	100	-0.0427	0.0164
06srv179	C25 H31 Cl2 N3 O Pd S	P 41	2983	100	-0.0373	0.0501
06srv184	C25 H31 Cl2 N3 O Pd S	P 1 21 1	3810	99	-0.0382	0.0146
06srv185	C14 H8 S	C 1 c 1	1522	98	-0.0536	0.0895
06srv191	C14 H18.967 B10 F	P 21 21 21	4047	100	0.513	0.5999
06srv220	C26 H25 Cl4 N3 O Pd S	P 21 21 21	3405	100	-0.0372	0.0144
06srv222	C32 H29 Al F19 Li O2	C 1 c 1	2891	70	-0.117	0.3028
06srv227	C14 H19 Br	P 21 21 21	1564	100	0.0075	0.0065
06srv230	C35 H31 Cl2 Fe N P2	P n a 21	3649	96	0.205	0.028
06srv241	C24 H30 O	P c a 21	2204	99	-0.3209	2.9414
06srv243	C84.5 H73 Cl3 F6 N O0.5 P5 Ru2	P 1	8451	96	0.2372	0.0264
06srv263	C14 H16 F N5	P 21 21 21	1546	100	-0.2492	0.6945
06srv264	C13 H14 F N5	P 21 21 21	1487	100	0.3372	0.5896

Continues on next page...

Structure *(cont'd)*	Formula *(cont'd)*	Space group *(cont'd)*	No. Bijvoet pairs *(cont'd)*	Bijvoet pair coverage (%) *(cont'd)*	Flack x *(cont'd)*	$\sigma(x)$ *(cont'd)*
06srv274	C19 H18 O2 S2	P 41 21 2	757	100	0.0219	0.0723
06srv290	C25 H31 Cl2 N3 O Pd S	P 21 21 21	2532	84	-0.0168	0.0256
06srv320*	C29 H31 Cl2 N3 O8.25 S	P 21 21 21	1672	92	0.0195	0.031
06srv321*	C30 H29 Cl2 N3 O8.5 S	P 21 21 21	1800	92	0.0877	0.0415
06srv325	C10 H11 N O5	P 21 21 21	1091	100	0.0331	0.984
06srv329	C19 H28 Si	P 21 21 21	2053	100	0.0991	0.0932
06srv359	C22 H29 P S	P 21 21 21	2306	100	-0.0242	0.0537
06srv364	C5.98 H6.92 I O1.96	P 21 21 21	1726	100	0.5376	0.0201
06srv370	C21 H33 B O2	P 1 c 1	5066	95	-0.1797	0.5515
06srv377	C36.75 H45 Cl5 N2 O4 P2 Pt	C 1 2 1	4777	100	0.4643	0.0084
06srv380	C23 H34 B Fe N O3	P 1 21 1	3173	99	-0.02	0.0149
06srv387	C25.5 H31 Cl2.91 N3 O Pd S	P 1 21 1	10611	90	-0.0043	0.0114
07srv015	C47.8 H67.6 Cl11.1 N6 O2 Pd2 S2	P 21 21 21	8231	100	-0.0293	0.0226
07srv039	C11 H8 O	P 21 21 21	745	100	-1.0539	1.6713
07srv040	C7 H12 Br0.684 N O3	P 1	2621	91	0.1354	0.0076
07srv042	C54 H77.4 Cl7 N6 O4.2 Pd2 S2	P 21 21 21	8420	100	-0.0027	0.0184
07srv043	C46 H66 Cl8 N6 O2 Pd2 S2	P 1 21 1	4844	81	0.0349	0.0447
07srv062	C24 H50 O5 Si4	C 1 2 1	979	23	0.3656	0.1913
07srv067	C10 H20 B N O4	P 21 21 21	1466	100	-0.1171	0.6859
07srv073	C6 H15 Cl5 P2 Pt	P 21 21 21	1925	100	-0.0079	0.0042
07srv077	C9 H18 B N O4	P 21 21 21	1342	100	-0.2326	0.6225
07srv079	C26 H35 Cl2 N3 O2 Pd S	P 1 21 1	3607	92	-0.0148	0.0104
07srv090	C19 H29 N3 O7 Si	P 1 21 1	3036	99	0.0106	0.084
07srv094	C8 H11 F2 N3 O	P n a 21	2407	100	-0.0715	0.4739
07srv100	C44 H62 Cl4 N6 O3 Pd2 S2	P 1 21 1	6824	100	-0.033	0.0151
07srv129	C45 H64 Cl6 N6 O2 Pd2 S2	P 41 21 2	2461	96	-0.0059	0.0436
07srv142	C23 H28 B N O4	P 1 21 1	2463	86	0.6816	0.7156
07srv157	C40 H32 N2 P2 Pt	C 1 c 1	3693	77	0.0236	0.0073
07srv161	C68 H79 Ag2 N14 O14.5	P m c 21	5249	100	0.5232	0.0757
07srv167	C18 H15 Au Cl P	P 21 21 21	1854	100	-0.0015	0.0084
07srv170	C25 H31 N3 O S	P 1 21 1	2526	99	0.0123	0.0764

Continues on next page...

Structure *(cont'd)*	Formula *(cont'd)*	Space group *(cont'd)*	No. Bijvoet pairs *(cont'd)*	Bijvoet pair coverage (%) *(cont'd)*	Flack x *(cont'd)*	$\sigma(x)$ *(cont'd)*
07srv172	C26 H36 O7	P 1 21 1	2774	89	-1.0817	0.9984
07srv205	C23 H27 Cl4 N3 O Pd S	P 21 21 21	4926	95	-0.033	0.0102
07srv244	C11 H23 B Cl N O2	C 1 2 1	1528	100	0.0931	0.0841
07srv258	C14 H25 N O5 Si	P 1 21 1	2192	93	0.014	0.0639
07srv275	C16 H32 N2 Ni P2	I -4 c 2	1289	98	0.429	0.0181
07srv288	C25 H40 N2 O4	P 1 21 1	3317	100	-0.2564	0.5711
07srv295	C13 H31 Al Cl5 P3	C 1 c 1	3706	97	0.164	0.0533
07srv299	C22 H30 N4 O2	C 1 2 1	2230	100	-1.0193	2.3259
07srv310	C13 H31 Cl5 P2 Pt	P 21 21 21	2530	97	0.2774	0.0064
07srv312	C20 H26 N4 O2	P 1 21 1	1847	93	-3.4475	3.3778
07srv331	C30 H32 Cl3 O P S2	P 1 21 1	5681	71	0.0245	0.0367
07srv345	C69 H56 Cl4 Fe2 P4 Pd S2	P n a 21	12223	89	0.4657	0.0112
07srv346	C29 H37 Cl2 N3 O Pd2 S	P 1 21 1	4343	100	-0.0067	0.0261
07srv363	C12 H29 Cl3 P2 Pt	P c a 21	4414	99	0.0468	0.0055
07srv376	C12 H29 Al Cl5 P3	C 1 c 1	2786	100	0.5225	0.0664
07srv379	C16 H10 O	P 1 21 1	1243	95	-0.3091	1.1646
07srv387	C18 H17 F N4 O	P 1 c 1	9327	99	-0.1235	0.3468
07srv401	C14 H14 N2 O S	P c a 21	362	15	-0.6052	0.2095
07srv422	C26 H24 P2 S	P 1	4589	77	0.0637	0.0404
07srv430	C22 H24 B I S	P n a 21	2579	88	0.0186	0.0113
07srv457	C18 H10 F3 N O2	P 21 21 21	1840	100	-0.4468	0.4433
07srv460	C12 H16 Br N3 O7	C 1 2 1	1879	99	0.0155	0.0114
07srv466	C27 H46 Al N4 P	P 21 21 21	3639	100	-0.0204	0.0449
07srv484	C16 H22 Cl2 F6 O2 P2 Pt	P 1 n 1	2589	96	0.0415	0.0037
07srv489	C11 H26 Cl2 P2 Pt	P 21 21 21	1784	96	0.0301	0.0037
07srv490	C54.5 H42.73 Cl O2 P2 Pd	P c a 21	7709	78	0.0397	0.0117
07srv513	C71.7 H84.9 Cl11.8 N9 O3 Pd3 S3	I 1 2 1	12117	100	-0.0322	0.0194
08srv023	C14 H14 Ag2 N4 O7 S	P 21 21 21	2200	100	-0.0485	0.025
08srv048	C40 H37 Cl14 N3 O Pd S	P 1 21 1	6976	100	-0.0196	0.0121
08srv051	C51 H42 O2 P2 Ru	P 1 21 1	4574	99	-0.0285	0.0263
08srv079	C21 H16 O2 S	P n a 21	1796	86	0.0848	0.0634

Continues on next page...

Structure *(cont'd)*	Formula *(cont'd)*	Space group *(cont'd)*	No. Bijvoet pairs *(cont'd)*	Bijvoet pair coverage (%) *(cont'd)*	Flack x *(cont'd)*	$\sigma(x)$ *(cont'd)*
08srv087	C75.5 H75 Cl7 N6 O3 Pd2 S2	P 42 21 2	2794	100	0.0275	0.0546
08srv088	C23 H27 Cl4 N3 O Pd S	P 21 21 21	3469	100	-0.027	0.0159
08srv096	C122 H144 Ir2 N6 O2	P 1 21 1	11081	95	0.1974	0.0053
08srv097	C122 H136.85 F4 Ir2 N6 O2	P 1 21 1	14260	95	-0.0026	0.0038
08srv104	C16 H29 N O4 Si	I 1 2 1	2163	84	0.021	0.106
08srv128	C12 H16 N4 O10	P 1 21 1	1955	96	-0.0885	1.2197
08srv132	C14 H12 N2	P 1 21 1	1090	78	0.2862	2.3055
08srv145	C9 H8 N2 O	C 1 c 1	901	86	-5.4028	2.6435
08srv168	C23 H17 Br N2	P c a 21	1874	100	0.015	0.0135
08srv218	C23 H27 N3 O S	P 1	5053	98	0.056	0.046
08srv221	C13 H13 Ag F6 N3 O P S	P 21 21 21	2087	100	-0.0534	0.0182
08srv222	C56 H59 Ag B F4 N12 O5 S4	P c a 21	5461	99	0.5347	0.0506
08srv261	C28 H36 B2 O2 S	P n a 21	3604	95	0.0034	0.0476
08srv269	C12 H14 N2 O2	P 21 21 21	1376	96	-0.0941	1.0005
08srv274	C10 H26 Cl6 P2 Sn	P 21 21 21	2550	96	0.4804	0.0117
08srv287	C13 H13 N3 O S	P c a 21	2722	97	0.5345	0.1321
08srv290	C29 H53 Cl5 P2 Pt	P 21 21 21	4116	100	0.0095	0.0086
08srv303	C24 H16 N2	C 1 m 1	723	100	-4.0389	6.0562
08srv336	C6 H15 Br5 P2 Pt	P 21 21 21	1692	100	0.0345	0.0131
08srv421	C53 H47 Fe N P2	P 21 21 21	5147	99	-0.0033	0.0109
08srv439*	C22 H19 N	P n a 21	1089	91	0.1555	0.5838
08srv470	C39 H36 B Cl2 N6 O3 P Ru	P c a 21	4897	99	-0.0371	0.0242
09srv026	C35 H46 N P2 Rh S2	P 21 21 21	6543	94	-0.0375	0.0202
09srv028	C45 H55 Cl P3 Rh S4	P 1	2012	32	0.0508	0.016
09srv052	C17 H19 O P	P 21 21 21	1865	100	0.064	0.0608
09srv071	C26 H29 B O2	P 1 21 1	2970	100	-0.7351	0.8569
09srv073	C11 H8 F4 N2 O	P n a 21	1165	100	-0.4831	1.1906
09srv129	C28 H64 O4 P6 Rh2	P 1 21 1	5643	98	0.3271	0.0174
09srv137	C12 H29 Cl3 N2 O2 W	P 21 21 21	2375	96	0.0128	0.0048
09srv140	C35 H35 N3 O	P 21 21 21	3596	96	0.4092	0.9754
09srv160	C24 H20 Al Cl4 N2 P	P 41 21 2	2613	94	0.3668	0.0483

Continues on next page...

Structure *(cont'd)*	Formula *(cont'd)*	Space group *(cont'd)*	No. Bijvoet pairs *(cont'd)*	Bijvoet pair coverage (%) *(cont'd)*	Flack x *(cont'd)*	$\sigma(x)$ *(cont'd)*
09srv162	C43 H37 F6 P2 Pd Sb	P 32	8105	95	-0.0445	0.0089
09srv210	C4 H8 Cl Li O2	P 21 21 21	1102	95	0.0071	0.029
09srv225	C66 H56 Cl6 O2 P2 Pd2	P 21 21 21	7065	100	-0.0278	0.0151
09srv230	C36 H30 N3 P	P n a 21	2196	96	-0.1295	0.1404
09srv231	C38 H41 N3 O	P n a 21	2722	100	-0.8549	2.252
09srv238	C21 H16 F10 N2	P 1 21 1	4197	100	-0.0305	0.5734
09srv254	C11 H7 F3 N2 O2 S	P 1 21 1	1644	99	-0.0388	0.0665
09srv277	C16 H15 Cl2 F3 N2 O5	P 21 21 21	2140	99	0.0179	0.045
09srv288	C15 H15 N3 O3	P n a 21	2394	100	0.0241	1.4624
09srv313	C11 H25 O2 P2 Rh	P -4 21 c	2128	98	0.0016	0.017
09srv356	C9 H9 Br O2	P 21 21 21	1074	100	0.029	0.0086
09srv380	C16 H23 B O2	P 21 21 21	1827	100	-0.126	0.8418
09srv386	C15 H16 F3 N O3 S	P n a 21	1250	99	-0.0341	0.1161

Table A.4: Bijvoet pair analysis for 134 non-centrosymmetric structures using Gaussian statistics. An asterisk indicates the use of copper radiation.

	Outlier cutoff					No cutoff				
Structure	Hooft y	$\sigma(y)$	ν	Slope	Corr. Coeff.	Hooft y	$\sigma(y)$	ν	Slope	Corr. Coeff.
07srv312	1.1709	1.0409	300.0	0.067	0.9983	-2.2872	3.6897	2.5	1.077	0.9983
08srv303	0.3818	1.5915	27.1	0.107	0.9984	-7.8521	4.7957	21.4	0.845	0.9985
07srv299	0.2625	0.3541	300.0	0.249	0.9991	-0.9722	0.8692	12.9	1.089	0.9990
08srv145	-1.2687	1.1829	75.8	0.181	0.9989	-6.5915	3.2249	3.3	1.056	0.9989
09srv231	-0.4777	0.5939	300.0	0.287	0.9986	-0.5721	1.2418	22.2	0.952	0.9986
06avc16	0.0846	0.3532	21.0	0.389	0.9988	-0.7518	0.6022	38.4	1.011	0.9990
09srv238	0.8580	0.2562	300.0	0.218	0.9984	0.4948	0.4944	10.4	0.608	0.9983
08srv132	0.7880	0.6972	75.8	0.454	0.9994	-0.2500	1.0136	14.2	0.899	0.9994
08srv128	-0.4674	0.5788	300.0	0.343	0.9983	0.9768	0.9845	3.6	0.836	0.9982
09srv140	0.5658	0.2677	42.4	0.563	0.9995	1.0575	0.3571	300.0	0.942	0.9995
07srv039	-0.8176	0.8199	300.0	0.398	0.9991	-1.4140	1.0969	10.4	0.721	0.9991
06avc18	-0.4804	0.4099	300.0	0.584	0.9995	-0.2878	0.5641	19.7	0.987	0.9994
07srv142	0.4218	0.3529	300.0	0.588	0.9996	0.8119	0.4466	113.1	0.935	0.9995
08srv269	0.5656	0.3359	39.1	0.632	0.9996	0.2843	0.4066	19.7	0.946	0.9997
06srv155	0.4720	0.0624	300.0	1.018	0.9993	0.4275	0.0903	12.7	1.652	0.9992
09srv288	0.6612	0.5960	300.0	0.716	0.9998	0.1100	0.6798	89.3	0.949	0.9998
06srv325	0.3743	0.2944	10.7	0.696	0.9988	0.3905	0.3387	19.7	1.010	0.9998
07srv094	-0.1623	0.2726	300.0	0.578	0.9996	-0.1588	0.3663	2.6	1.065	0.9996
07srv172	-0.5137	0.6878	10.4	0.657	0.9976	-0.8594	0.7122	19.7	0.870	0.9998
07srv077	0.2891	0.2434	300.0	0.641	0.9997	-0.2584	0.2965	22.3	0.895	0.9997
07srv379	0.0276	0.3508	10.4	0.672	0.9982	-0.0373	0.3741	19.7	0.911	0.9999
06srv263	-0.0970	0.3100	300.0	0.541	0.9998	-0.3010	0.3707	26.2	0.754	0.9998
09srv071	-0.1819	0.3206	75.8	0.668	0.9999	-0.3988	0.3786	19.7	0.874	1.0000
09srv380	-0.4681	0.3146	101.0	0.685	0.9999	-0.7389	0.3692	20.2	0.915	0.9999
07srv288	-0.1854	0.2476	22.1	0.655	0.9996	-0.3504	0.2922	10.4	0.879	0.9999
07srv457	0.0973	0.1927	300.0	0.567	0.9990	-0.2172	0.2283	21.2	0.778	0.9989
08srv287	0.5627	0.0889	21.0	0.569	0.9995	0.5150	0.1199	3.3	0.959	0.9998

Continues on next page...

	Outlier cutoff					No cutoff				
Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
07srv387	0.2618	0.1825	38.4	0.692	0.9998	-0.0670	0.2013	19.7	0.858	0.9999
07srv067	0.2509	0.3325	300.0	0.804	0.9999	-0.0327	0.3712	77.5	0.964	0.9999
06srv241	1.0571	0.6809	19.7	0.661	0.9993	0.8553	0.7257	15.0	0.799	0.9999
06srv055	-0.0485	0.1814	13.9	0.748	0.9990	-0.1846	0.2145	4.5	1.082	1.0000
06srv264	0.1551	0.2671	12.9	0.739	0.9990	0.1823	0.2832	20.6	0.873	0.9998
06srv370	0.1362	0.2266	19.7	1.107	0.9993	-0.1201	0.2400	10.4	1.292	0.9999
08srv439*	0.0843	0.2311	5.7	0.881	0.9942	-0.0762	0.2392	5.8	1.141	0.9998
09srv073	-0.4896	0.4105	7.7	0.601	0.9973	-0.0077	0.4489	5.9	0.761	0.9999
07srv062	0.1972	0.1138	10.4	0.918	0.9980	0.1612	0.1174	7.0	1.097	0.9996
07srv401	0.0694	0.2132	300.0	0.739	0.9995	0.2987	0.2721	3.4	0.862	0.9994
09srv386	-0.0437	0.1125	4.5	0.760	0.9902	-0.0392	0.1199	2.4	1.101	0.9998
08srv218	0.0485	0.0263	22.4	0.734	0.9996	0.0281	0.0275	10.4	0.808	1.0000
06srv191	0.2182	0.2286	300.0	0.898	0.9999	0.2619	0.2355	75.8	0.944	0.9999
06srv222	-0.0302	0.1254	11.0	1.009	0.9985	-0.0587	0.1282	10.4	1.123	0.9998
08srv104	0.0768	0.0594	13.7	0.826	0.9990	0.0568	0.0595	14.5	0.894	0.9998
07srv422	0.0688	0.0295	4.5	1.692	0.9850	0.0597	0.0296	3.6	2.294	0.9994
07srv258	0.0042	0.0348	10.8	1.170	0.9985	-0.0025	0.0351	10.5	1.273	0.9999
07srv090	0.0172	0.0543	15.0	1.021	0.9992	0.0125	0.0548	13.0	1.105	1.0000
06srv151	0.0338	0.0500	38.4	0.973	0.9998	0.0225	0.0504	29.0	1.013	0.9999
08srv261	0.0117	0.0219	6.2	1.290	0.9940	0.0100	0.0220	5.9	1.512	0.9997
07srv170	0.0405	0.0499	300.0	0.918	0.9999	0.0164	0.0497	300.0	0.926	0.9999
08srv079	0.0159	0.0324	10.6	1.171	0.9982	0.0139	0.0325	10.8	1.263	0.9996
07srv161	0.4645	0.0339	6.3	1.123	0.9939	0.4643	0.0341	5.9	1.313	0.9998
09srv230	-0.0164	0.1013	300.0	0.913	0.9998	-0.0497	0.1007	300.0	0.922	0.9998
06srv134	-0.0104	0.0480	4.5	0.645	0.9853	-0.0054	0.0481	4.3	0.842	0.9999
08srv051	-0.0308	0.0181	7.7	1.054	0.9958	-0.0333	0.0181	7.5	1.217	0.9995
06avc17	-0.0049	0.0200	24.5	1.021	0.9997	-0.0333	0.0181	7.5	1.217	1.0000
06srv178	-0.0424	0.0119	15.0	0.869	0.9991	-0.0436	0.0119	19.7	0.932	0.9999
08srv222	0.5206	0.0251	24.1	0.864	0.9996	0.5227	0.0251	22.7	0.899	0.9999
06srv230	0.2061	0.0201	7.0	0.791	0.9956	0.2063	0.0202	6.0	0.926	0.9997

Continues on next page...

	Outlier cutoff					No cutoff				
Structure *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*	Hooft y *(cont'd)*	$\sigma(y)$ *(cont'd)*	ν *(cont'd)*	Slope *(cont'd)*	Corr. Coeff. *(cont'd)*
06srv148	0.0115	0.0431	10.4	0.999	0.9984	0.0104	0.0432	10.4	1.081	0.9998
09srv028	0.0547	0.0115	300.0	0.832	0.9999	0.0548	0.0115	300.0	0.838	0.9999
06srv329	0.1603	0.0356	22.9	0.986	0.9996	0.1565	0.0357	21.4	1.027	0.9999
07srv376	0.5579	0.0284	21.6	0.898	0.9996	0.5577	0.0284	22.3	0.934	0.9999
06srv274	-0.0206	0.0255	300.0	0.978	0.9986	-0.0225	0.0255	300.0	0.985	0.9985
06srv359	-0.0073	0.0278	49.2	0.984	0.9999	-0.0092	0.0279	52.5	1.005	0.9999
09srv254	0.0151	0.0328	7.4	0.787	0.9966	0.0148	0.0328	7.4	0.905	0.9998
09srv052	0.0586	0.0246	300.0	0.900	0.9999	0.0563	0.0246	300.0	0.905	0.9998
06srv243	0.2337	0.0180	19.7	0.949	0.9995	0.2322	0.0180	19.7	1.003	1.0000
07srv043	0.0095	0.0295	5.7	1.217	0.9853	0.0091	0.0295	5.7	1.564	0.9991
08srv470	-0.0312	0.0163	10.4	1.564	0.9970	-0.0332	0.0163	10.4	1.748	0.9994
07srv295	0.1855	0.0259	19.7	0.879	0.9992	0.1862	0.0259	19.7	0.939	0.9999
06srv290	-0.0275	0.0202	75.8	0.998	0.9999	-0.0308	0.0202	75.8	1.013	0.9999
09srv129	0.3205	0.0137	26.5	0.868	0.9997	0.3207	0.0137	26.2	0.903	1.0000
07srv244	0.0770	0.0205	38.4	0.869	0.9997	0.0770	0.0205	38.4	0.896	0.9998
07srv466	-0.0141	0.0152	300.0	0.878	1.0000	-0.0144	0.0152	300.0	0.883	0.9999
07srv430	0.0174	0.0052	103.8	0.792	0.9999	0.0174	0.0052	75.8	0.801	0.9999
09srv160	0.3791	0.0170	69.9	0.963	0.9999	0.3792	0.0170	75.8	0.977	0.9999
06avc13	0.0198	0.0227	29.0	0.924	0.9998	0.0194	0.0227	29.0	0.957	0.9999
06srv179	-0.0575	0.0289	14.6	0.719	0.9991	-0.0577	0.0289	15.0	0.771	0.9999
06srv042	0.0143	0.0189	19.7	0.959	0.9993	0.0144	0.0189	19.7	1.022	1.0000
07srv331	0.0540	0.0217	42.0	0.908	0.9998	0.0526	0.0217	41.7	0.926	0.9999
07srv345	0.4613	0.0042	21.5	0.996	0.9995	0.4613	0.0042	20.9	1.036	0.9999
09srv162	-0.0381	0.0045	38.4	0.809	0.9998	-0.0381	0.0045	38.4	0.836	1.0000
07srv015	-0.0318	0.0077	22.0	0.781	0.9996	-0.0322	0.0077	21.2	0.812	1.0000

Appendix B

CIF Grammar

A grammar for the CIF syntax in the format required for the ANTLR [Parr, 2007] parser generator. Lexer rules (*i.e.* those that generate tokens during lexing) are denoted with upper case rule names. Parser rules are denoted with lower case names.

```
/** CIF Version 1.1 Working specification grammar

Translated from the grammar defined at

http://www.iucr.org/resources/cif/spec/version1.1/cifsyntax#bnf

*/

grammar cif;

options {
  language=C;
}
```

```
/*-------------------------------------------------------------------------
 * PARSER RULES
 *-------------------------------------------------------------------------*/

// The start rule
parse
  : cif (EOF | '\u001a' /*Ctrl-Z*/) ;

/*-------------------------------------------------------------------------
 * BASIC STRUCTURE OF A CIF
 *-------------------------------------------------------------------------*/

cif
  : (COMMENTS)? (WHITESPACE)*
    ( data_block ( WHITESPACE* data_block )* (WHITESPACE)* )?
  ;

loop_body
  : value ( WHITESPACE+ value)* ;

save_frame
  : SAVE_FRAME_HEADING ( WHITESPACE+ data_items )+ WHITESPACE+ SAVE
  ;

data_items
  : TAG WHITESPACE* value | loop_header WHITESPACE* loop_body
  ;
```

```
data_block
  : DATA_BLOCK_HEADING ( WHITESPACE+ ( data_items | save_frame ) )*
  ;

loop_header
  : LOOP_ ( WHITESPACE+ TAG )+ WHITESPACE
  ;

/*----------------------------------------------------------------------
 * TAGS AND VALUES
 *----------------------------------------------------------------------*/

inapplicable
  : '.' ;

unknown
  : '?' ;

value
  : inapplicable | unknown | '-' | char_string  | numeric| text_field
  ;

integer
  : ( '+' | '-' )? UNSIGNED_INTEGER ;

number
  : integer | FLOAT ;
```

```
numeric
  : number | ( number '(' (UNSIGNED_INTEGER)+ ')' ) ;

char_string
  : CHAR_STRING ;

text_field
  : SEMI_COLON_TEXT_FIELD ;

/*------------------------------------------------------------------
 * LEXER RULES
 *------------------------------------------------------------------*/

/*------------------------------------------------------------------
 * CHARACTER SETS
 *------------------------------------------------------------------*/

fragment EOL
  : ( '\n' | '\r' | '\r\n' ) ;

fragment DOUBLE_QUOTE
  : '"' ;

fragment SINGLE_QUOTE
  : '\'' ;

fragment ORDINARY_CHAR
  : '!' | '%' | '&' | '(' | ')' | '*' | '+' | ',' | '-' | '.' |
    '/' | ( '0'.. '9' ) | ':' | '<' | '=' | '>' | '?' | '@' |
```

```
    ('A'..'Z') | ('a'..'z') | '\\' | '^' | '`' | '{' | '|' |
    '}' | '~'
  ;

fragment NON_BLANK_CHAR_
  : ORDINARY_CHAR | DOUBLE_QUOTE | SINGLE_QUOTE |
    '#' | '$' | '_' | '[' | ']' | ';'
  ;

fragment TEXT_LEAD_CHAR
  : ORDINARY_CHAR | DOUBLE_QUOTE | SINGLE_QUOTE |
    '#' | '$' | '_' | '[' | ']' | ' ' | '\t'
  ;

fragment ANY_PRINT_CHAR
  : ORDINARY_CHAR | '#' | '$' | '_' | '[' | ']' | ' ' | '\t' | ';'
  ;

TAG : '_' (NON_BLANK_CHAR_)+ ;

/*-------------------------------------------------------------------
 * RESERVED WORDS - define these after semicolon text field
 *-------------------------------------------------------------------*/

fragment DATA_
  : ( 'D' | 'd' ) ( 'A' | 'a' ) ( 'T' | 't' ) ( 'A' | 'a' ) '_' ;

fragment SAVE_
  : ( 'S' | 's' ) ( 'A' | 'a' ) ( 'V' | 'v' ) ( 'E' | 'e' ) '_' ;
```

```
LOOP_
   : ( 'L' | 'l' ) ( 'O' | 'o' ) ( 'O' | 'o' ) ( 'P' | 'p' ) '_' ;

GLOBAL_
   : ( 'G' | 'g' ) ( 'L' | 'l' ) ( 'O' | 'o' ) ( 'B' | 'b' )
     ( 'A' | 'a' ) ( 'L' | 'l' ) '_'
   ;

STOP_
   : ( 'S' | 's' ) ( 'T' | 't' ) ( 'O' | 'o' ) ( 'P' | 'p' ) '_' ;

/*----------------------------------------------------------------------
 * SPECIAL KEY WORDS
 *----------------------------------------------------------------------*/

DATA_BLOCK_HEADING
   : DATA_ (NON_BLANK_CHAR)+ ;

SAVE_FRAME_HEADING
   : SAVE_ (NON_BLANK_CHAR)+ ;

SAVE
   : SAVE_ ;

/*----------------------------------------------------------------------
 * NUMERICS
 *----------------------------------------------------------------------*/
```

```
fragment DIGIT
  : '0'..'9' ;

fragment EXPONENT
  : ( ( 'e' | 'E') | ( 'e' | 'E')( '+' | '-' ) ) (DIGIT)+ ;

fragment INTEGER
  : ( '+' | '-' )? (DIGIT)+ ;

FLOAT
  : INTEGER EXPONENT | ( ( '+' | '-' )? ( (DIGIT)* '.' (DIGIT)+)
    | (DIGIT)+ '.' ) (EXPONENT)?
  ;

UNSIGNED_INTEGER
  : (DIGIT)+ ;

/*------------------------------------------------------------------
 * CHARACTER STRINGS AND FIELDS
 *------------------------------------------------------------------*/

fragment UNQUOTED_STRING
  : (({ GETCHARPOSITIONINLINE() > 0 }?=> ';')
     | ORDINARY_CHAR ) (NON_BLANK_CHAR_)*
  ;

// a single quoted string such as 'a dog's life' is legal
fragment SINGLE_QUOTED_STRING
  : SINGLE_QUOTE
```

```
    ( ( (SINGLE_QUOTE NON_BLANK_CHAR_)=>SINGLE_QUOTE )
      | ANY_PRINT_CHAR | DOUBLE_QUOTE )*
    SINGLE_QUOTE
  ;

fragment DOUBLE_QUOTED_STRING
  : DOUBLE_QUOTE
    ( ( (DOUBLE_QUOTE NON_BLANK_CHAR_)=>DOUBLE_QUOTE )
      | ANY_PRINT_CHAR | SINGLE_QUOTE )*
        DOUBLE_QUOTE
  ;

CHAR_STRING
  : UNQUOTED_STRING | SINGLE_QUOTED_STRING | DOUBLE_QUOTED_STRING;

SEMI_COLON_TEXT_FIELD
  : ( { GETCHARPOSITIONINLINE() == 0 }?=> ';')
    ( ( ANY_PRINT_CHAR | SINGLE_QUOTE | DOUBLE_QUOTE )* EOL
    ( (TEXT_LEAD_CHAR
      ( ANY_PRINT_CHAR | SINGLE_QUOTE | DOUBLE_QUOTE )* )? EOL)* )
    ';'
  ;

/*------------------------------------------------------------------
 * WHITE SPACE AND COMMENTS
 *------------------------------------------------------------------*/

COMMENTS
  : ( ( '#' (ANY_PRINT_CHAR | SINGLE_QUOTE | DOUBLE_QUOTE )*
```

```
      ( EOL | \{ LA(1) == EOF \}? ) )+ )
    \{ $channel = HIDDEN; \}
  ;

// Redefine this as non-fragment so can be seen by the parser
NON_BLANK_CHAR
  : NON_BLANK_CHAR_ ;

WHITESPACE
  :   ( '\t' | ' ' | EOL | '\u000C' )+ ;
```

Appendix C

Additional Information

Courses Attended

- 03/2008 BCA PCG Rietveld Refinement School, Durham University, UK.
- 08/2008 Kyoto Crystallographic Computing School, Japan.
- 03/2009 12th BCA/CCG Intensive Teaching School in X-Ray Structure Analysis, Durham University, UK.

Conferences Attended

- 04/2008 British Crystallographic Association Spring Meeting, University of York, UK.
- 08/2008 XXI Congress and General Assembly of the International Union of Crystallographers, Osaka, Japan.
- 04/2009 British Crystallographic Association Spring Meeting, Loughborough University, UK.
- 08/2009 25th European Crystallographic Meeting, Istanbul, Turkey.

- 11/2009 CCG Autumn Meeting, University of Newcastle-Upon-Tyne, UK.
- 04/2010 British Crystallographic Association Spring Meeting, University of Warwick, UK.
- 07/2010 American Crystallographic Association Annual Meeting, Chicago, IL, USA.
- 11/2010 CCG Autumn Meeting, The Royal Society of Edinburgh, UK.

Posters and Oral Presentations Outside Durham University

- 04/2008 **Oral** Olex2: The New Molecular Tool. Young Crystallographers session of the BCA Spring Meeting, York.
- 04/2008 **Poster** History and metadata in Olex2. BCA Spring Meeting, York.
- 08/2008 **Poster** Workflow and metadata in Olex2. IUCr Congress, Osaka, Japan.
- 04/2009 **Poster** Harnessing the Power of the cctbx with Olex2. BCA Spring Meeting, Loughborough.
- 08/2009 **Poster** Harnessing the Power of the cctbx with Olex2. 25th ECM, Istanbul, Turkey.
- 07/2010 **Oral** A New Solvent Masking Procedure. ACA Annual Meeting, Chicago, IL, USA.
- 08/2010 **Oral** Small Molecule Software for the 21st Century. Lawrence Berkeley National Laboratory, CA, USA.

Publications

- O. V. Dolomanov, L. J. Bourhis, R. J. Gildea, J. A. K. Howard, and H. Puschmann. *OLEX2*: a complete structure solution, refinement and analysis program. *Journal of Applied Crystallography*, 42(2):339-341, Apr 2009.
- L. J. Bourhis, R. J. Gildea, O. V. Dolomanov, J. A. K. Howard, and H. Puschmann. Small molecule toolbox. *Newsletter of the IUCr Commission on Crystallographic Computing*, 10:19-32, 2009.
- O. V. Dolomanov, L. J. Bourhis, R. J. Gildea, J. A. K. Howard, and H. Puschmann. Olex2. *Newsletter of the IUCr Commission on Crystallographic Computing*, 10:46-49, 2009.

Appendix D

Supplementary Electronic Materials

D.1 cctbx source code

Source code bundles are provided. The latest source code bundles and installation instructions can be obtained from http://cci.lbl.gov/build/all.html, where instructions for accessing the SVN repository can also be found.

Self-extracting cctbx sources for Unix

Under Unix, if Python 2.3 through 2.7 is pre-installed on the target platform, the smaller cctbx_bundle.selfx can be used. However, in general it will be best to use the cctbx_python_271_bundle.selfx file because the installation script will automatically install a recent Python before proceeding with the installation of the cctbx modules.

The Unix bundles include a file cctbx_install_script.csh. This script is known to work with:

- Linux: any gcc $\geq$ 3.2 - Mac OS 10.4 or higher with Apple's compiler

Other Unix platforms will most likely require adjustments of the build scripts.

Run the following command in any new, empty directory:

```
perl cctbx_bundle.selfx
```

This installs all cctbx modules from scratch. Python 2.3 or higher must be pre-installed on the target machine. The first python on PATH is used. To install with a different, specific python, add the full path to the command line, e.g.:

```
perl cctbx_bundle.selfx /usr/local/bin/python
```

Manually building from sources under Windows 2000 or higher

The cctbx installation requires Visual C++ 8.0 (Visual Studio .NET 2005) or higher.

To install Python under Windows it is best to use a binary installer from the Python download page, http://www.python.org/download/. The default choices presented by the installation wizard are usually fine.

Recent self-contained cctbx sources are available in the self-extracting file cctbx_bundle.exe. To unpack this file in a new, empty directory enter

```
cctbx_bundle.exe
```

This creates a subdirectory cctbx_sources. The installation procedure should be executed in another directory, e.g.

```
mkdir cctbx_build
cd cctbx_build
C:\python27\python.exe ..\cctbx_sources\libtbx\configure.py smtbx iotbx
```

The last command initializes the cctbx_build directory and creates a file setpaths.bat (among others). This file must be used to initialize a new shell or process with the cctbx settings

```
setpaths.bat
```

To compile enter

```
libtbx.scons
```

On a machine with multiple CPUs enter

```
libtbx.scons -j N
```

where N is the number of CPUs available.

Note that libtbx.scons is just a thin wrapper around SCons. The SCons documentation applies without modification.

To run some regression tests after the compilation is finished enter::

```
setpaths_all.bat
libtbx.python %SCITBX_DIST%\run_tests.py
libtbx.python %CCTBX_DIST%\run_tests.py --Quick
```

The output should show many OK. A Python Traceback is an indicator for problems.

D.2 Olex2 binaries

Current development builds of Olex2 are provided for Windows, Mac and Linux platforms.

Windows installation

Run installer.exe alongside the olex2.zip or olex2-x64.zip file to perform offline installation of the development version of Olex2. The default installation folder is in in the **C:\Program Files** directory, but you can change that to another location if you prefer or don't have access to that area. The different versions will install into sub-folders called Olex2-1.1, Olex2-1.1-beta etc. Different versions of Olex2 can be installed next to each other and will operate entirely independently.

Mac OS X installation

Unzip the file mac-intel-py26.zip into a new folder, and edit the start script to point it to the right location of the olex2.app. Then use this start script to launch Olex2.

Linux installation

The binaries provided should be compatible with most Linux distributions (but may not be optimised for your machine architecture). We provide Suse 10.1 binaries, which you can find in suse101x32-py26.zip and suse101x64-py26.zip. Unzip the correct file for your machine and modify the start script inside olex2 folder, to point to the right location of the executable.

Linux RPMs for several Fedora and Centos versions are kindly provided by Dr. John Warren. Further information on installing these can be obtained at http://www.olex2.org/content/folder-linux.

References

S. C. Abrahams and E. T. Keve. Normal probability plot analysis of error in measured and derived quantities and standard deviations. *Acta Crystallographica Section A*, 27(2):157–165, Mar 1971. doi: 10.1107/S0567739471000305. 40, 42, 48

P. D. Adams, P. V. Afonine, G. Bunkóczi, V. B. Chen, I. W. Davis, N. Echols, J. J. Headd, L.-W. Hung, G. J. Kapral, R. W. Grosse-Kunstleve, A. J. McCoy, N. W. Moriarty, R. Oeffner, R. J. Read, D. C. Richardson, J. S. Richardson, T. C. Terwilliger, and P. H. Zwart. *PHENIX*: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallographica Section D*, 66(2):213–221, Feb 2010. doi: 10.1107/S0907444909052925. 2, 4, 9, 17

P. V. Afonine, R. W. Grosse-Kunstleve, and P. D. Adams. A robust bulk-solvent correction and anisotropic scaling procedure. *Acta Crystallographica Section D*, 61(7):850–855, Jul 2005. doi: 10.1107/S0907444905007894. 63

F. H. Allen. The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallographica Section B*, 58(3 Part 1):380–388, Jun 2002. doi: 10.1107/S0108768102003890. 54

F. H. Allen, O. Johnson, G. P. Shields, B. R. Smith, and M. Towler. CIF applications. XV. *enCIFer*: a program for viewing, editing and visualizing CIFs. *Journal of Applied Crystallography*, 37(2):335–338, Apr 2004. doi: 10.1107/S0021889804003528. 86

E. Aubert and C. Lecomte. Illustrated Fourier transforms for crystallography.

Journal of Applied Crystallography, 40(6):1153–1165, Dec 2007. doi: 10.1107/S0021889807043622. 59

A. S. Batsanov. Private communication, 2000. 67

A. S. Batsanov. Private communication, 2009. 69

A. S. Batsanov, A. E. Goeta, J. A. K. Howard, A. K. Hughes, and J. M. Malget. The synthesis of closo- and nido-(aminoalkyl)dicarbaboranes: a re-examination of contradictory literature reports, crystal structure of [7-H3N(CH2)3-7,8-c2b9h11]nh2nh2. *J. Chem. Soc., Dalton Trans.*, 12:1820–1826, 2001. doi: 10.1039/B102009I. 68

A. S. Batsanov, R. M. K. Deng, K. B. Dillon, A. E. Goeta, J. A. K. Howard, J. Meldrum, P. K. Monks, H. Puschmann, and H. J. Shepherd. Crystal and molecular structures of some six-coordinate tin(iv) halogeno complexes with phosphorus-containing ligands. *Heteroatom Chemistry*, 20(3):136–143, 2009. ISSN 1098-1071. doi: 10.1002/hc.20525. 67

P. W. Betteridge, J. R. Carruthers, R. I. Cooper, K. Prout, and D. J. Watkin. *CRYSTALS* version 12: software for guided crystal structure analysis. *Journal of Applied Crystallography*, 36(6):1487, Dec 2003. doi: 10.1107/S0021889803021800. 24, 90

J. M. Bijvoet, A. F. Peerdeman, and A. J. van Bommel. Determination of the Absolute Configuration of Optically Active Compounds by Means of X-Rays. *Nature*, 168(4268):271–272, Aug 1951. doi: 10.1038/168271a0. 33

W. Bluhm. Star (cif) parser. http://pdb.sdsc.edu/STAR/index.html, 2000. 75

S. Bogza, K. Kobrakov, A. Malienko, I. Perepichka, S. Sujkov, M. Bryce, S. Lyubchik, A. Batsanov, and N. Bogdan. A versatile synthesis of pyrazolo [3, 4-c] isoquinoline derivatives by reaction of 4-aryl-5-aminopyrazoles with aryl/heteroaryl aldehydes: the effect of the heterocycle on the reaction pathways. *Organic & Biomolecular Chemistry*, 3(5):932–940, 2005. ISSN 1477-0520. 65

L. J. Bourhis, R. J. Gildea, O. V. Dolomanov, J. A. K. Howard, and H. Puschmann. Small molecule toolbox. *Newsletter of the IUCr Commission on Crystallographic Computing*, 10:19–32, 2009. 15

L. J. Bourhis, R. J. Gildea, O. V. Dolomanov, J. A. K. Howard, and H. Puschmann. 2011. 58, 64

D. Britton. Estimation of twinning parameter for twins with exactly superimposed reciprocal lattices. *Acta Crystallographica Section A*, 28(3):296–297, May 1972. doi: 10.1107/S0567739472000786. 23, 24

I. D. Brown and I. A. Guzei. Restraints CIF Dictionary. 2011. URL ftp://ftp.iucr.org/pub/cif_core_restraints.dic. 81

I. D. Brown and B. McMahon. CIF: the computer language of crystallography. *Acta Crystallographica Section B*, 58(3 Part 1):317–324, Jun 2002. doi: 10.1107/S0108768102003464. 74

M. C. Burla, R. Caliandro, M. Camalli, B. Carrozzini, G. L. Cascarano, L. De Caro, C. Giacovazzo, G. Polidori, D. Siliqi, and R. Spagna. *IL MILIONE*: a suite of computer programs for crystal structure solution of proteins. *Journal of Applied Crystallography*, 40(3):609–613, Jun 2007. doi: 10.1107/S0021889807010941. 3

W. Chang and P. E. Bourne. CIF Applications. IX. A new approach for representing and manipulating STAR files. *Journal of Applied Crystallography*, 31 (3):505–509, Jun 1998. doi: 10.1107/S0021889897017019. 75

C. A. Christensen, A. S. Batsanov, M. R. Bryce, and J. A. K. Howard. Molecular saddles. 7.1 new 9,10-bis(1,3-dithiol-2-ylidene)-9,10-dihydroanthracene cyclophanes: synthesis, redox properties, and x-ray crystal structures of neutral species and a dication salt. *J. Org. Chem.*, 66(10):3313–3320, 2001. doi: 10.1021/jo001524k. 68

P. Coppens and A. Volkov. The interplay between experiment and theory in charge-density analysis. *Acta Crystallographica Section A*, 60(5):357–364, Sep 2004. doi: 10.1107/S0108767304014953. 90

D. Coster, K. S. Knol, and J. A. Prins. Difference in the intensities of x-ray reflection from the two sides of the 111 plane of zinc blende. *Z. Phys.*, 63: 345–369, 1930. 33

B. Dittrich, M. Strumpel, M. Schafer, M. Spackman, and T. Koritsanszky. Invarioms for improved absolute structure determination of light-atom crystal structures. *Acta Crystallographica Section A: Foundations of Crystallography*, 62(3): 217–223, 2006a. ISSN 0108-7673. 90

B. Dittrich, M. Strumpel, M. Schäfer, M. A. Spackman, and T. Koritsánszky. Invarioms for improved absolute structure determination of light-atom crystal structures. *Acta Crystallographica Section A*, 62(3):217–223, May 2006b. doi: 10.1107/S0108767306010336. 36

O. V. Dolomanov, L. J. Bourhis, R. J. Gildea, J. A. K. Howard, and H. Puschmann. *OLEX2*: a complete structure solution, refinement and analysis program. *Journal of Applied Crystallography*, 42(2):339–341, Apr 2009a. doi: 10.1107/S0021889808042726. 3, 31, 48, 64, 88

O. V. Dolomanov, L. J. Bourhis, R. J. Gildea, J. A. K. Howard, and H. Puschmann. Age concern - the background. *Newsletter of the IUCr Commission on Crystallographic Computing*, 10:46–49, 2009b. 1

M. Dušek, V. Petříček, M. Wunschel, R. E. Dinnebier, and S. van Smaalen. Refinement of modulated structures against X-ray powder diffraction data with *JANA*2000. *Journal of Applied Crystallography*, 34(3):398–404, Jun 2001. doi: 10.1107/S0021889801003302. 24

P. Ellis and H. Bernstein. CBFlib: An API for CBF/imgCIF Crystallographic Binary Files with ASCII Support, 2001. 75

G. Fermi, M. Perutz, B. Shaanan, and R. Fourme. The crystal structure of human deoxyhaemoglobin at 1.74 Å resolution. *Journal of Molecular Biology*, 175(2): 159–174, 1984. ISSN 0022-2836. 82

H. D. Flack. On enantiomorph-polarity estimation. *Acta Crystallographica Section A*, 39(6):876–881, Nov 1983. doi: 10.1107/S0108767383001762. iii, 22, 34, 89

H. D. Flack and G. Bernardinelli. Reporting and evaluating absolute-structure and absolute-configuration determinations. *Journal of Applied Crystallography*, 33(4):1143–1148, Aug 2000. doi: 10.1107/S0021889800007184. 35, 45

G. Friedel. Sur les symtries cristallines que peut rvler la diffraction des rayons X. *C.R. Acad. Sci. Paris*, 157:1533–1536, 1913. 32

C. Giacovazzo, H. L. Monaco, G. Artioli, D. Viterbo, G. Ferraris, G. Gilli, G. Zanotti, and M. Catti. *Fundamentals of Crystallography*. Oxford University Press, 2002. 8, 9, 49

N. Godbert, A. S. Batsanov, M. R. Bryce, and J. A. K. Howard. Molecular saddles. 4.1 redox-active cyclophanes by bridging the 9,10-bis(1,3-dithiol-2-ylidene)-9,10-dihydroanthracene system: synthesis, electrochemistry, and x-ray crystal structures of neutral species and a dication salt. *The Journal of Organic Chemistry*, 66(3):713–719, 2001. doi: 10.1021/jo001014q. PMID: 11430087. 65

C. T. Grainger. Pseudo-merohedral twinning. The treatment of overlapped data. *Acta Crystallographica Section A*, 25(3):427–434, May 1969. doi: 10.1107/S0567739469000866. 23, 24

S. Gražulis, D. Chateigner, R. T. Downs, A. F. T. Yokochi, M. Quirós, L. Lutterotti, E. Manakova, J. Butkus, P. Moeck, and A. Le Bail. Crystallography Open Database – an open-access collection of crystal structures. *Journal of Applied Crystallography*, 42(4):726–729, Aug 2009. doi: 10.1107/S0021889809016690. 75, 82

R. W. Grosse-Kunstleve. Evaluation of different approaches to minimization: full-matrix and conjugate gradient as implemented in shelx-76 and shelxl-97, lbfgs and new algorithms under development. Private communication, 2011. 86

R. W. Grosse-Kunstleve and P. D. Adams. On the handling of atomic anisotropic displacement parameters. *Journal of Applied Crystallography*, 35(4):477–480, Aug 2002. doi: 10.1107/S0021889802008580. 7

R. W. Grosse-Kunstleve and P. D. Adams. State of the toolbox: an overview of the computational crystallography toolbox (cctbx). *Comp. Comm. Newsletter*, 1, 2003. 4

R. W. Grosse-Kunstleve, N. K. Sauter, N. W. Moriarty, and P. D. Adams. The *Computational Crystallography Toolbox*: crystallographic algorithms in a reusable software framework. *Journal of Applied Crystallography*, 35(1):126–136, Feb 2002. doi: 10.1107/S0021889801017824. 4, 63

R. W. Grosse-Kunstleve, P. V. Afonine, and P. D. Adams. cctbx news: Geometry restraints and other new features. *Newsletter of the IUCr Commission on Crystallographic Computing*, 4:19–36, 2004. 18

J. Haestier. Handling cell-parameter errors in crystallographic data. *Journal of Applied Crystallography*, 42(5):798–809, Oct 2009. doi: 10.1107/S0021889809024376. 30

S. R. Hall and H. J. Bernstein. CIF Applications. V. *CIFtbx*2: extended tool box for manipulating CIFs. *Journal of Applied Crystallography*, 29(5):598–603, Oct 1996. doi: 10.1107/S0021889896006371. 75

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. doi: 10.1107/S010876739101067X. 74

W. C. Hamilton. Significance tests on the crystallographic R factor. *Acta Crystallographica*, 18(3):502–510, Mar 1965. doi: 10.1107/S0365110X65001081. 34

A. P. Hammersley, H. J. Bernstein, and J. D. Westbrook. Image CIF Dictionary (imgCIF) and Crystallographic Binary File Dictionary (CBF). 2003. URL ftp//ftp.iucr.org/pub/cif_img.dic. 74

N. K. Hansen and P. Coppens. Testing aspherical atom refinements on small-molecule data sets. *Acta Crystallographica Section A*, 34(6):909–921, Nov 1978. doi: 10.1107/S0567739478001886. URL http://dx.doi.org/10.1107/S0567739478001886. 90

R. Herbst-Irmer and G. M. Sheldrick. Refinement of Twinned Structures with *SHELXL*97. *Acta Crystallographica Section B*, 54(4):443–449, Aug 1998. doi: 10.1107/S0108768197018454. 22

R. Herbst-Irmer and G. M. Sheldrick. Refinement of obverse/reverse twins. *Acta Crystallographica Section B*, 58(3 Part 2):477–481, Jun 2002. doi: 10.1107/S0108768102001039. 22

J. R. Hester. A validating CIF parser: *PyCIFRW*. *Journal of Applied Crystallography*, 39(4):621–625, Aug 2006. doi: 10.1107/S0021889806015627. 75

F. L. Hirshfeld. Can X-ray data distinguish bonding effects from vibrational smearing? *Acta Crystallographica Section A*, 32(2):239–244, Mar 1976. doi: 10.1107/S0567739476000533. 10, 12

R. W. W. Hooft, L. H. Straver, and A. L. Spek. Determination of absolute structure using Bayesian statistics on Bijvoet differences. *Journal of Applied Crystallography*, 41(1):96–103, Feb 2008. doi: 10.1107/S0021889807059870. iii, 36, 37, 38, 39, 44, 45, 89, 90

R. W. W. Hooft, L. H. Straver, and A. L. Spek. Probability plots based on Student's *t*-distribution. *Acta Crystallographica Section A*, 65(4):319–321, Jul 2009. doi: 10.1107/S0108767309009908. 42

R. W. W. Hooft, L. H. Straver, and A. L. Spek. Using the *t*-distribution to improve the absolute structure assignment with likelihood calculations. *Journal of Applied Crystallography*, 43(4):665–668, Aug 2010. doi: 10.1107/S0021889810018601. 42, 44, 90

J. A. K. Howard and D. J. Watkin. Age concern - the background. *Newsletter of the IUCr Commission on Crystallographic Computing*, 10:7–18, 2009. 1

E. R. Howells, D. C. Phillips, and D. Rogers. The probability distribution of X-ray intensities. II. Experimental investigation and the X-ray detection of centres of symmetry. *Acta Crystallographica*, 3(3):210–214, May 1950. doi: 10.1107/S0365110X50000513. 49

G. B. Jameson. On structure refinement using data from a twinned crystal. *Acta Crystallographica Section A*, 38(6):817–820, Nov 1982. doi: 10.1107/S0567739482001673. 25

JCrystalSoft. FTL-SE. http://jcrystal.com/products/ftlse/index.htm, 2010. viii, 60

Khronos Group. OpenGL the industry standard for high performace graphics. http://www.opengl.org. 3

Y. Lin. ASTAR: a .NET class library for STAR/CIF manipulation. *Journal of Applied Crystallography*, 43(4):916–919, Aug 2010. doi: 10.1107/S0021889810018145. 75

M. Lutz and A. M. M. Schreurs. Was Bijvoet right? Sodium rubidium (+)-tartrate tetrahydrate revisited. *Acta Crystallographica Section C*, 64(8):m296–m299, Aug 2008. doi: 10.1107/S0108270108022415. 33

G. Madariaga. CIF Dictionary for Modulated Structures. 2002. URL ftp://ftp.iucr.org/pub/cif_ms.dic. 74

P. Mallinson. Electron density CIF dictionary. 2003. URL ftp://ftp.iucr.org/pub/cif_rho.dic. 74

P. Murray-Rust. The crystal structure of $[Co(NH_3)_6]_4Cu_5Cl_17$: a twinned cubic crystal. *Acta Crystallographica Section B*, 29(11):2559–2566, Nov 1973. doi: 10.1107/S0567740873007004. 23, 24

G. Oszlányi and A. Süto. The charge flipping algorithm. *Acta Crystallographica Section A*, 64(1):123–134, Jan 2008. doi: 10.1107/S0108767307046028. 5, 90

L. Palatinus and G. Chapuis. *SUPERFLIP* – a computer program for the solution of crystal structures by charge flipping in arbitrary dimensions. *Journal of Applied Crystallography*, 40(4):786–790, Aug 2007. doi: 10.1107/S0021889807029238. 3

T. Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Programmers. Pragmatic Bookshelf, first edition, May 2007. ISBN 0978739256. 75, 110

S. Parsons and H. Flack. Precise absolute-structure determination in light-atom crystals. *Acta Crystallographica Section A*, 60(a1):s61, Aug 2004. doi: 10.1107/S0108767304098800. 18, 36, 90

V. Petříček and M. Dušek. *JANA98, The crystallographic Computing System*. Institute of Physics, Academy of Sciences of the Czech Republic, Praha, Czech Republic, 2000. 24

S. Picard, P. Gougeon, and M. Potel. $Rb_4Mo_21Se_24$ containing Mo_12 and Mo_15 clusters. *Acta Crystallographica Section C*, 57(4):335–336, Apr 2001. doi: 10.1107/S0108270100019466. 84

L. Potterton, S. McNicholas, E. Krissinel, J. Gruber, K. Cowtan, P. Emsley, G. N. Murshudov, S. Cohen, A. Perrakis, and M. Noble. Developments in the *CCP*4 molecular-graphics project. *Acta Crystallographica Section D*, 60(12 Part 1): 2288–2294, Dec 2004. doi: 10.1107/S0907444904023716. 2

C. S. Pratt, B. A. Coyle, and J. A. Ibers. Redetermination of the structure of nitrosylpenta-amminecobalt(iii) dichloride. *J. Chem. Soc. A*, pages 2146–2151, 1971. doi: 10.1039/J19710002146. 25

Python Software Foundation. Python programming language. http://www.python.org/. 4, 75, 76

D. Rogers. On the application of Hamilton's ratio test to the assignment of absolute configuration and an alternative test. *Acta Crystallographica Section A*, 37(5): 734–741, Sep 1981. doi: 10.1107/S0567739481001629. 34

D. E. Sands. Transformations of variance-covariance tensors. *Acta Crystallographica*, 21(6):868–872, Dec 1966. doi: 10.1107/S0365110X66004092. 27, 29

L. Schröder, D. J. Watkin, A. Cousson, R. I. Cooper, and W. Paulus. *CRYSTALS* enhancements: refinement of atoms continuously disordered along a line, on a

ring or on the surface of a sphere. *Journal of Applied Crystallography*, 37(4): 545–550, Aug 2004. doi: 10.1107/S0021889804009847. 56

D. Schwarzenbach. On uncertainty estimates of crystallographic quantities including cell-parameter uncertainties. *Journal of Applied Crystallography*, 43(6): 1452–1455, Dec 2010. doi: 10.1107/S0021889810039592. 30

SciPy. Scientific tools for python. http://www.scipy.org/. 46

G. Sheldrick. *The SHELX-97 Manual*. Dept. of Structural Chemistry, Univ. of Göttingen, Göttingen, Germany, 1997. 9, 12, 13, 14, 18, 24

G. M. Sheldrick. A short history of *SHELX*. *Acta Crystallographica Section A*, 64 (1):112–122, Jan 2008. doi: 10.1107/S0108767307043930. 3, 24, 35, 58, 64, 75, 90

A. L. Spek. Single-crystal structure validation with the program *PLATON*. *Journal of Applied Crystallography*, 36(1):7–13, Feb 2003. doi: 10.1107/S0021889802022112. 3, 40, 56

A. L. Spek. Structure validation in chemical crystallography. *Acta Crystallographica Section D*, 65(2):148–155, Feb 2009. doi: 10.1107/S090744490804362X. 86

E. Stanley. The identification of twins from intensity statistics. *Journal of Applied Crystallography*, 5(3):191–194, Jun 1972. doi: 10.1107/S0021889872009185. 49, 50

Student. The probable error of a means. *Biometrika*, 6(1):1–25, 1908. doi: 10.1093/biomet/6.1.1. 42

B. H. Toby. Powder CIF dictionary. 1998. URL ftp://ftp.iucr.org/pub/cif_pd.dic. 74

A. A. Vagin, R. A. Steiner, A. A. Lebedev, L. Potterton, S. McNicholas, F. Long, and G. N. Murshudov. *REFMAC*5 dictionary: organization of prior chemical knowledge and guidelines for its use. *Acta Crystallographica Section D*, 60(12 Part 1):2184–2195, Dec 2004. doi: 10.1107/S0907444904023510. 74

P. van der Sluis and A. L. Spek. BYPASS: an effective method for the refinement of crystal structures containing disordered solvent regions. *Acta Crystallographica Section A*, 46(3):194–201, Mar 1990. doi: 10.1107/S0108767389011189. iii, 5, 56, 73, 89

J. Waser. Least-squares refinement with subsidiary conditions. *Acta Crystallographica*, 16(11):1091–1094, Nov 1963. doi: 10.1107/S0365110X63002929. 7

D. Watkin. Structure refinement: some background theory and practical strategies. *Journal of Applied Crystallography*, 41(3):491–522, Jun 2008. doi: 10.1107/S0021889808007279. 8, 61

J. D. Westbrook, S.-H. Hsieh, and P. M. D. Fitzgerald. CIF Applications. VI. *CIFLIB*: an application program interface to CIF dictionaries and data files. *Journal of Applied Crystallography*, 30(1):79–83, Feb 1997. doi: 10.1107/S0021889896008643. 75

S. P. Westrip. *publCIF*: software for editing, validating and formatting crystallographic information files. *Journal of Applied Crystallography*, 43(4):920–925, Aug 2010. doi: 10.1107/S0021889810022120. 86

Wolfram Research, Inc. Mathematica Edition: Version 8.0, 2010. 31

wxWidgets. wxWidgets cross-platform GUI library. http://www.wxwidgets.org. 3

D. S. Yufit and J. A. K. Howard. Cyclopentanone and cyclobutanone. *Acta Crystallographica Section C*, 67(3):o104–o106, Mar 2011. doi: 10.1107/S0108270111004069. 82

W. H. Zachariasen. Dispersion in quartz. *Acta Crystallographica*, 18(4):714–716, Apr 1965. doi: 10.1107/S0365110X65001640. 24

P. H. Zwart, R. W. Grosse-Kunstleve, and P. D. Adams. Xtriage and Fest: automatic assessment of X-ray data and substructure structure factor estimation. *CCP4 Newsletter*, 43:contribution 7, 2005. 25