SUPERFLIP – a computer program for the solution of crystal structures by charge flipping in arbitrary dimensions

Lukáš Palatinus and Gervais Chapuis

computer programs

SUPERFLIP – a computer program for the solution of crystal structures by charge flipping in arbitrary dimensions

Lukáš Palatinus*‡ and Gervais Chapuis

Laboratoire de Cristallographie, Le Cubotron, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. Correspondence e-mail: palat@fzu.cz

SUPERFLIP is a computer program that can solve crystal structures from diffraction data using the recently developed charge-flipping algorithm. It can solve periodic structures, incommensurately modulated structures and quasi-crystals from X-ray and neutron diffraction data. Structure solution from powder diffraction data is supported by combining the charge-flipping algorithm with a histogram-matching procedure. *SUPERFLIP* is written in Fortran90 and is distributed as a source code and as precompiled binaries. It has been successfully compiled and tested on a variety of operating systems.

1. Introduction

Modern crystallographic computing methods require significant computational effort and are therefore useful only in combination with a computer program that implements them. This is especially true for the algorithms that use iterative procedures and computationally expensive methods like the discrete Fourier transform on large grids.

One such algorithm is the charge-flipping method, which is a structure solution method for the reconstruction of scattering densities from diffraction data. Since its publication (Oszlányi & Sütő, 2004, 2005), it has been successfully applied to a wide range of crystallographic problems, including the solution of simple structures (Wu *et al.*, 2004), structures with high pseudosymmetry (Oszlányi *et al.*, 2006), modulated structures (Palatinus, 2004; Zuñiga *et al.*, 2006; Palatinus *et al.*, 2006) and quasicrystals (Katrych *et al.*, 2007), and structure solution from powder diffraction data (Baerlocher, Gramm *et al.*, 2007; Baerlocher, McCusker & Palatinus, 2007).

The algorithm itself is surpisingly simple. In its original version it has been developed for the reconstruction of electron density from X-ray diffraction data. The electron density is described by discrete sampling on a regular grid in the unit cell. The structure solution proceeds in iterative cycles. The iteration is initialized by assigning random phases to the experimental structure amplitudes. From this trial solution an electron density is calculated by inverse discrete Fourier transform. This electron density is modified so that all grid points with density below a certain positive threshold δ are multiplied by -1 (flipped). New temporary structure factors are calculated by discrete Fourier transform of this modified density. The phases of these temporary structure factors are combined with the experimental amplitudes and this set of structure factors enters the next cycle of iteration. δ is the only parameter of the algorithm.

A modification of the algorithm has been developed (Oszlányi & Sütő, 2005) that improves the probability of convergence by a special handling of the weak reflections. If a reflection is considered weak, the calculated amplitude of its temporary structure factor is retained

and its phase is shifted by $\pi/2$. This induces an additional perturbation in the phase space and leads to an improved performance of the algorithm. For more details on the charge flipping algorithm, see Oszlányi & Sütő (2004 or 2005). Recently, the algorithm was generalized to the reconstruction of scattering densities that are not strictly positive, such as those from neutron scattering (Oszlányi & Sütő, 2007).

The important ingredient of the algorithm is that it does not use the symmetry of the structure during the density reconstruction. The density is treated as if it had symmetry $P1$ and the symmetry in the density is induced only by the symmetry in the diffraction pattern. This makes the structure solution insensitive to possible ambiguities in the determination of the space group. On the other hand, the density reconstructed by charge flipping is randomly shifted and the origin of the space group must be located afterwards.

In order to allow the wide crystallographic community to take advantage of the new algorithm, a computer program named *SUPERFLIP* has been created. The name is derived from 'charge FLIPping in SUPERspace', because the program allows for density reconstruction in arbitrary dimensions, thus making it possible to solve standard periodic structures and aperiodic structures within the same framework. The program provides a self-contained structure-solution tool that can be used for automatic structure solution of simple structures as well as custom-tailored solution of complex problems.

2. Program description

The principal input to *SUPERFLIP* is one input file, which contains the instructions for the program. The program reads in the input file, processes the instructions found therein and produces the output. User interaction with the program during the runtime is possible, but is limited only to changing several parameters of the algorithm. The principal output of the program is the reconstructed scattering density. The progress of the program is recorded in a log file.

The input file is in a free format and it contains keywords followed by one or more values. The only information that must be always contained in any input file is the lattice parameters, the expected

‡ Permanent address: Institute of Physics, Academy of Science of the Czech Republic, Na Slovance 2, 182 21 Prague, Czech Republic.

symmetry of the structure, the dimensions of the grid and the list of reflections. The program has many other options which allow a detailed control of the execution; however, default values work in most cases.

The program run can be divided into three major steps:

(*a*) Reading the data and checking for consistency. The program reads the instructions from the input file. It checks the consistency of the symmetry operations, namely if each matrix represents a valid rotation matrix and if the operations form a complete space group. In the next step, the list of reflections is expanded according to the given symmetry to produce a full list of reflections in the whole sphere. If Friedel pairs are present in the input file, their intensities are averaged, because the charge-flipping algorithm does not take the anomalous scattering into account and therefore the Friedel pairs are not independent even in a non-centrosymmetric structure. At the end of the data processing, the coverage of reciprocal space as a function of $\sin\theta/\lambda$ can be calculated to provide the user additional means of checking the completeness and consistency of the data and symmetry.

(*b*) Charge flipping. This is the central part of the program. The expanded list of reflections is assigned random starting phases and the charge-flipping iteration is started. The cycle is repeated until convergence is reached or until the maximum allowed number of iteration cycles is exceeded. The crucial step in this stage is the proper selection of δ, which will be discussed in §2.1.

(*c*) Searching for the proper position of the symmetry operations in the density. It has already been mentioned that charge flipping reconstructs the electron density without taking the space-group symmetry into account. However, the symmetry is contained in the reflection intensities and possibly in the systematic absences. Thus, the resulting electron density will exhibit, albeit approximately, the correct space-group symmetry, but with an origin arbitrarily placed in the unit cell. *SUPERFLIP* therefore searches the resulting density for the location of the symmetry operations and, optionally, averages the electron density according to the symmetry. The details of this process are described in §2.2.

In the following sections some issues of particular importance for the performance of the program are discussed in detail.

2.1. Searching the proper value of δ

The flipping threshold δ is the only free parameter of the original algorithm and its selection is critical for the success of the algorithm. No method is currently known for predicting the optimal value of δ *a priori*, as it depends in a complicated manner on the resolution of the data, the thermal parameters and the type of atoms present in the cell. Fortunately, extensive testing of charge flipping on a variety of structures has allowed us to design an empirical method for determining δ.

The method is based on the observation of the behaviour of the total charge in the current density and the amount of the flipped charge. The total charge in the map c_{tot} is obtained as a sum $\sum \rho_i$ over all grid points, or simply as the Fourier coefficient $F(\mathbf{0})$ of the current density. The total flipped charge c_{flip} is defined as $c_{\mathrm{flip}} = \sum_{\rho_i < \delta} |\rho_i|$, where the summation runs over all density pixels i with $\rho_i < \delta$. It has been observed that for proper values of δ the ratio $c_{\mathrm{tot}}/c_{\mathrm{flip}}$ tends to lie close to 0.9 and this value is independent of the input data, in particular independent of the grid size, data resolution, symmetry and unit-cell volume.

This method was implemented in *SUPERFLIP* in the following way. A trial value of δ is selected so that with this value 80% of the pixels of the initial random density will be flipped. Ten iteration cycles are performed with this trial value in order to pass the few transi-

tional iteration cycles at the beginning of the iteration and reach a stable plateau. After ten cycles the ratio $c_{\mathrm{tot}}/c_{\mathrm{flip}}$ is checked. If it falls in the interval $\langle 0.8, 1.0 \rangle$, the current value of δ is accepted and the iteration continues. Otherwise δ is decreased, if $c_{\mathrm{tot}}/c_{\mathrm{flip}} < 0.8$, or increased, if $c_{\mathrm{tot}}/c_{\mathrm{flip}} > 1.0$, and the procedure is repeated with the modified δ. Needless to say, the user also has the possibility of defining δ manually by giving its value in the input file.

2.2. Symmetry of the reconstructed density

A reliable and fast location of the position of the symmetry elements in the density is crucial for the performance of any charge-flipping program that has the ambition of being widely used. The result of the charge-flipping iteration is a density that is randomly shifted in the unit cell. Moreover, the density obeys the correct space-group symmetry only approximately. While such density already contains the structural information, it would be very inconvenient to interpret it. This is especially true for densities with dimensions higher than three. Thus, a method must be found to determine automatically the offset of the origin of the density from the conventional origin of the space group, so that the density can be appropriately shifted and then averaged over the symmetry-equivalent grid points.

Perhaps somewhat surprisingly, it is not trivial to determine the origin shift, especially if the method is required to work in arbitrary dimensions. In the following paragraphs we describe the method that is implemented in *SUPERFLIP*.

Let $\{\mathbf{R}|\tau\}$ be a symmetry operation described with respect to the conventional origin of the space group. A density ρ^{sg}, which contains this symmetry operation and whose origin is placed at the conventional origin of the space group, fulfills $\rho^{\mathrm{sg}}(\mathbf{R}\mathbf{r} + \tau) = \rho^{\mathrm{sg}}(\mathbf{r})$. A shift of the density from the conventional origin by a vector $\mathbf{s}$ will produce a density ρ^{sh} with

$$\rho^{\mathrm{sh}}(\mathbf{r}) = \rho^{\mathrm{sg}}(\mathbf{r} - \mathbf{s}). \tag{1}$$

The symmetry operation $\{\mathbf{R}|\tau\}$ is transformed by the shift $\mathbf{s}$ to $\{\mathbf{R}|\tau + \mathbf{d}\}$ with $\mathbf{d}$ given by

$$\mathbf{d} = (\mathbf{I} - \mathbf{R})\mathbf{s}, \tag{2}$$

where $\mathbf{I}$ is the unit matrix. Thus, if the origin shift $\mathbf{s}$ is unknown, we can use $\mathbf{d}$ to determine $\mathbf{s}$. Because $(\mathbf{I} - \mathbf{R})$ is in general not invertible, a single symmetry operation might not be sufficient to determine all components of $\mathbf{s}$.

The above expressions are valid if the density ρ^{sh} is perfectly symmetrical with respect to the symmetry operation $\{\mathbf{R}|\tau + \mathbf{d}\}$. However, the output of the charge-flipping algorithm is a density ρ^{cf} that fulfills the symmetry only approximately. Because it is shifted from the conventional origin of the space group by an unknown shift vector, it can be considered an approximation to the density ρ^{sh} defined above. The vector $\mathbf{d}$ can be found in ρ^{cf} by finding the maximum of the correlation integral between $\rho^{\mathrm{cf}}(\mathbf{r})$ and $\rho^{\mathrm{cf}}(\mathbf{R}\mathbf{r} + \tau + \mathbf{d})$ as a function of $\mathbf{d}$. The correlation integral is given by

$$C(\mathbf{d}) = \int \rho^{\mathrm{cf}}(\mathbf{r})\rho^{\mathrm{cf}}(\mathbf{R}\mathbf{r} + \tau + \mathbf{d}) \, \mathrm{d}\mathbf{r}. \tag{3}$$

As can be immediately seen from this equation, the correlation function $C(\mathbf{d})$ is given by the convolution $\rho^{\mathrm{cf}}(\mathbf{r}) \star \rho^{\mathrm{cf}}[-(\mathbf{R}\mathbf{r} + \tau)]$. Using the convolution theorem, the correlation function can be expressed as

$$C(\mathbf{d}) = \mathcal{F}^{-1}[F(\mathbf{H}) \cdot F'(\mathbf{H})^*], \tag{4}$$

where $F(\mathbf{H})$ and $F'(\mathbf{H})$ represent the structure factors of $\rho^{\mathrm{cf}}(\mathbf{r})$ and $\rho^{\mathrm{cf}}(\mathbf{R}\mathbf{r} + \tau)$, respectively, $*$ denotes complex conjugation and $\mathcal{F}^{-1}$ the

inverse Fourier transform. From the properties of the Fourier transform it follows that $F'(\mathbf{H})$ is related to $F(\mathbf{H})$ by

$$F'(\mathbf{H}) = F(\mathbf{HR})\exp(2\pi i\mathbf{H}\tau). \tag{5}$$

Using equations (4) and (5), the correlation function $C(\mathbf{d})$ of the discretized electron density can be evaluated efficiently and quickly by the fast Fourier transform of the product of the structure factors. The position of the absolute maximum of $C(\mathbf{d})$ defines the best estimate for the vector $\mathbf{d}$, modulo any lattice translation vector.

To determine the position $\mathbf{s}$ of the conventional origin of the space group, it is sufficient to locate a common origin for a set of generators of the space group. For each generator $g_i = \{\mathbf{R}_i|\tau_i\}$, $i = 1, \ldots, N_{\text{gen}}$, a corresponding vector $\mathbf{d}_i$ can be found using the method described above. We then obtain a set of equations for the origin shift $\mathbf{s}$ in the form

$$\mathbf{d}_i + \mathbf{t}_i = (\mathbf{I} - \mathbf{R}_i)\mathbf{s}, i = 1, \ldots, N_{\text{gen}}, \tag{6}$$

where $\mathbf{t}$ represents any lattice translation vector. If $\mathbf{t}_i$ could be omitted, equation (6) would represent a set of $D.N_{\text{gen}}$ linear equations (D being the dimension of the electron density, normally three) for the components of the origin shift $\mathbf{s}$. These equations could be solved by standard methods of linear algebra. However, the infinitely many translation vectors $\mathbf{t}_i$ cannot be omitted, because each of them corresponds to an equivalent, but in general different, position of the generator g_i. A combination of $\mathbf{t}_i$ that leads to a common solution $\mathbf{s}$ cannot be found *a priori*. Luckily, the number of possible $\mathbf{t}_i$ can be restricted, because at least one solution for $\mathbf{s}$ lies in the first unit cell, *i.e.* we have for the components of $\mathbf{s}$, $s_j \in \langle 0; 1 \rangle, j = 1, \ldots, D$. With this restriction, the infinite number of possible translation vectors $\mathbf{t}_i$ for each g_i can be reduced to a finite, and usually small, number N_i^{trans} vectors, such that equation (6) with these translation vectors substituted for $\mathbf{t}_i$ has a solution in the first unit cell. A common solution $\mathbf{s}$ can result from any combination of choices of the translation vectors, leading to $\prod_{i=1}^{N_{\text{gen}}} N_i^{\text{trans}}$ sets of linear equations. A solution that satisfies any of these sets of equations is a possible space-group origin $\mathbf{s}$.

The set of equations (6) can be overdetermined, and in such a case $\mathbf{s}$ must be determined by a least-squares procedure. If the set of equations (6) is underdetermined, any of the infinitely many solutions represents an acceptable origin shift.

In practice, the number of generators and the number of acceptable translation vectors are usually small, and therefore also the total number of sets of equations $\prod_{i=1}^{N_{\text{gen}}} N_i^{\text{trans}}$ is small. Moreover, it is not necessary to solve all sets, because the calculation can be stopped as soon as the first solution is found. As a result, the procedure described in this section is rather fast and takes only a few seconds or even a fraction of a second on an ordinary computer.

2.3. Handling of higher-dimensional cases

It has been shown by Palatinus (2004) that the charge-flipping algorithm can be generalized to incommensurately modulated cases in a straightforward manner. Similarly, it has been shown by Katrych *et al.* (2007) that charge flipping can be used for the solution of structures of quasicrystals. Both these types of structures are usually described in a higher-dimensional space (superspace), and the real structure in physical space is obtained as a three-dimensional section of this higher-dimensional structure. The number of dimensions is four or more for incommensurately modulated structures, five for decagonal quasicrystals and six for icosahedral quasicrystals. The only modification to the original charge-flipping algorithm is that for higher-dimensional structures the reflections are indexed by D integers ($D > 3$) instead of three, and the scattering density is described

on a D-dimensional discrete grid. The space-group symmetry must correspondingly be defined by the $D \times D$ rotation matrices and D-dimensional translation vectors, as is usual in the field of modulated structures and quasicrystals. Note that the method for the symmetry search described in §2.2 is completely independent of the dimensionality of the structure, and can be used directly for the higher-dimensional cases.

In order to be as general as possible and adopt any type of structure, even those yet undiscovered, *SUPERFLIP* is written so that the number of dimensions can be completely arbitrary. This is true both for the total dimensionality of the structure and for the dimensionality of the physical space. Thus, *SUPERFLIP* can be also used to solve one- and two-dimensional problems, possibly incommensurately modulated, as well as any other problem of any dimensionality.

3. Additional features of the program

The program offers a broad range of features that are extensively described in the user manual. Here, we give only a summary of the most frequently used options that have not been described above, together with a short description.

(*a*) Apart from the charge-flipping algorithm, *SUPERFLIP* can also use the low-density elimination method (Shiono & Woolfson, 1992). This method is closely related to charge flipping, but differs in the density-modification step.

(*b*) The reflection input can have variable format. Any of the following items can be used: intensity, amplitude, real and imaginary components of the structure factor, intensity/amplitude + phase. The order of the entries can be also defined by the user. This makes it possible to import virtually any reflection list without any modifications.

(*c*) The user has the option of providing a reference density. *SUPERFLIP* will then shift the reconstructed density so that it matches the reference. This is indispensable if several calculations are to be compared directly, because the space group usually has more possible origins in the unit cell, but these origins are not equivalent points of the structure. Thus, the structures derived from different charge-flipping runs need not necessarily have the same origin, and if they need to be compared directly they must be shifted to a common origin using a reference density.

(*d*) The reconstructed density can be tested for the presence of user-defined symmetry operations. Because the density is reconstructed in $P1$, it is possible to analyse it for the presence of different symmetry operations. Using this option it is, for example, possible to distinguish a mirror plane from a glide plane or to test for the presence of an inversion centre.

(*e*) The value of δ that is optimal for convergence is not optimal for obtaining a density as noise-free as possible. Therefore, it is possible to follow the 'structure-solving' iteration by several cycles of 'density-polishing' iteration, which suppresses the noise in the density considerably and leads to better definition of the weak features and more symmetrical densities.

(*f*) Difficult structures sometimes need several attempts before the iteration converges. In extremely difficult cases, the quality of the reconstruction can also vary from one calculation to another. Therefore, it is possible to repeat the whole calculation automatically many times and save only the best results.

(*g*) It turns out that the symmetry analysis of a density can be interesting independently of the charge-flipping process. Therefore, *SUPERFLIP* can perform a symmetry analysis of a user-supplied

density without performing the charge-flipping iteration. The symmetry analysis can be performed on any density distribution supplied in one of the formats supported by *SUPERFLIP*.

(*h*) Charge flipping can be used not only as a structure-solution method but also as a structure-completion method (Oszlányi & Sütő, 2004). In practice this means that the starting phases of the structure factors need not be random, but may correspond to a partial structure model. Charge flipping can then be used for structure completion as a very efficient alternative to the usual Fourier recycling. It is possible to provide the non-random starting model either as a model density or by specifying the phases of the input structure factors.

(*i*) *SUPERFLIP* supports the band-flipping variant of the charge-flipping algorithm (Oszlányi & Sütő, 2007), which allows for the solution of structures with negative scattering densities, mainly from neutron scattering experiments.

(*j*) The standard version of the algorithm uses a fixed value of δ during the whole iteration. It has been suggested by Wu *et al.* (2004) that a dynamic δ could be used instead. In this version, δ is redetermined in each cycle so that a constant proportion of the density pixels are flipped. *SUPERFLIP* supports both the static and the dynamic handling of δ.

(*k*) *SUPERFLIP* supports the histogram-matching procedure to enhance the performance of the structure solution from powder diffraction data. The procedure uses an expected histogram of the density to modify the density during the iteration and to repartition the intensities of the overlapping reflections. For a detailed description of the procedure and an overview of the solved structures, see Baerlocher, McCusker & Palatinus (2007).

4. Output description

The reconstructed density can currently be saved in three formats. The first is the binary '*m*81' format of the crystallographic computing system *JANA2000* (Petříček *et al.*, 2000) which can be directly read by *JANA2000* and viewed by its plotting module CONTOUR. The second supported format is the *X-PLOR* format. *X-PLOR* is a software package for structural biology (Brünger, 1992). This is an ASCII format that can be easily transformed into any other format if needed. This format can be read, for example, by the software package *CHIMERA* (Pettersen *et al.*, 2004) to produce three-dimensional isosurface plots of the density. The third supported format is the *m*80 format of *JANA2000*, which contains a list of the structure factors corresponding exactly to the density. This format is very concise in terms of file size. It can be used by the FOURIER module of *JANA2000* to produce arbitrary sections through the electron density. The *m*81 format is designed for densities up to six dimensions. The *X-PLOR* format has been generalized to arbitrary dimensions, but only its three-dimensional version can be read by other software. The *m*80 format can adopt structures with a completely arbitrary dimensionality. Finally, it should be noted that additional formats can be added at any time upon request.

5. Analysis of the output density

SUPERFLIP itself does not perform any analysis of the resulting electron density in terms of locating the atomic positions and assigning the elements to the atomic maxima. However, a separate program, *EDMA*, exists that can be used to perform the analysis. *EDMA* was originally developed as part of the *BayMEM* suite (van Smaalen *et al.*, 2003) to analyse electron densities obtained by the maximum entropy method, but it has also been extended for the

analysis of densities obtained by *SUPERFLIP*. *EDMA* analyses the density for maxima and assigns atomic types to these maxima according to the qualitative or quantitative chemical composition supplied by the user. *EDMA* can output the result of the analysis as a CIF, an INS file for the *SHELX* program suite or an *m*40 file for *JANA2000*. *EDMA* can also analyse the electron density of modulated structures.

6. Software and hardware requirements and availability

The program is written in standard Fortran90 and can thus be compiled and run on any system with a working Fortran90 compiler. The program requires a FFTW3 library for the fast Fourier transform (Frigo & Johnson, 2005; http://www.fftw.org). This library must be installed on the system before *SUPERFLIP* is compiled. *SUPERFLIP* also makes use of the *LAPACK* linear algebra library (Anderson *et al.*, 1999; http://netlib.org/lapack/). However, the necessary code is contained in the installation package of *SUPERFLIP* and the *LAPACK* libraries do not have to be installed separately.

Apart from being available as a source code, precompiled binaries of *SUPERFLIP* are available for the operating systems Windows and MacOS X. These binaries are self-standing and do not need any supplementary libraries to run.

The program does not have any special hardware requirements. The program requires typically several MB or tens of MB of memory to handle common three-dimensional structures. However, the required memory can exceed 1 GB if very large structures are considered, especially large modulated structures or quasicrystals.

SUPERFLIP is freely available for download at http://superspace.epfl.ch/superflip. The complete installation package contains the source code, documentation and sample input files. Precompiled binaries for Windows and MacOS X are also available for download. The same webpage also allows the download of the density-analysis program *EDMA* as source code or precompiled binaries. Please note, however, that despite being available on the *SUPERFLIP* web page, *EDMA* is not part of *SUPERFLIP*.

7. Conclusions

SUPERFLIP is a computer program for the solution of crystal structures using the charge-flipping method. The program has been applied to a broad range of crystal structures, including complex modulated structures and quasicrystals. It has been also used to solve several structures from powder diffraction data, some of them extremely complex (Baerlocher, Gramm *et al.*, 2007; Baerlocher, McCusker & Palatinus, 2007). The aim of the program is to provide an alternative to the established structure-solution software. The program is designed so that it can solve simple structures fully automatically, but at the same time provide sufficient versatility and transparency to allow for the solution of complicated cases. One of the strengths of the program is the way of analysing the symmetry of the reconstructed density.

SUPERFLIP, in combination with the density-analysis program *EDMA*, provides a self-contained structure-solution system. In order to facilitate user interaction with these programs, an interface for *SUPERFLIP* and *EDMA* is available in the crystallographic computing system *JANA2000* (Petříček *et al.*, 2000) and in the *CRYSTALS* package (Betteridge *et al.*, 2003). These interfaces provide a convenient way of creating the input files for *SUPERFLIP* and *EDMA*.

computer programs

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. & Sorensen, D. (1999). *LAPACK Users' Guide*, 3rd edition. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics.

Baerlocher, C., Gramm, F., Massüger, L., McCusker, L., Zhanbing, H., Hovmöller, S. & Xiaodong, Z. (2007). *Science*, **315**, 1113–1116.

Baerlocher, C., McCusker, L. & Palatinus, L. (2007). *Z. Kristallogr.* **222**, 47–53.

Betteridge, P. W., Carruthers, J. R., Cooper, R. I., Prout, K. & Watkin, D. J. (2003). *J. Appl. Cryst.* **36**, 1487.

Brünger, A. (1992). *X-PLOR – A System for X-ray Crystallography and NMR.* New Haven, USA: Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University.

Frigo, M. & Johnson, S. G. (2005). *Proc. IEEE*, **93**, 216–231.

Katrych, S., Weber, T., Kobas, M., Massüger, L., Palatinus, L., Chapuis, G. & Steurer, W. (2007). *J. Alloys Compd.* **428**, 164–172.

Oszlányi, G. & Sütő, A. (2004). *Acta Cryst.* A**60**, 134–141.

Oszlányi, G. & Sütő, A. (2005). *Acta Cryst.* A**61**, 147–152.

Oszlányi, G. & Sütő, A. (2007). *Acta Cryst.* A**63**, 156–163.

Oszlányi, G., Sütő, A., Czugler, M. & Párkányi, L. (2006). *J. Am. Chem. Soc.* **128**, 8392–8393.

Palatinus, L. (2004). *Acta Cryst.* A**60**, 604–610.

Palatinus, L., Dušek, M., Glaum, R. & El Bali, B. (2006). *Acta Cryst.* B**62**, 556–566.

Petříček, V., Dušek, M. & Palatinus, L. (2000). *The Crystallographic Computing System JANA2000*. Prague, Czech Republic: Institute of Physics.

Pettersen, E., Goddard, T., Huang, C., Couch, G., Greenblatt, D., Meng, E. & Ferrin, T. (2004). *J. Comput. Chem.* **25**, 1605–1612.

Shiono, M. & Woolfson, M. M. (1992). *Acta Cryst.* A**48**, 451–456.

Smaalen, S. van, Palatinus, L. & Schneider, M. (2003). *Acta Cryst.* A**59**, 459–469.

Wu, J. S., Spence, J. C. H., O'Keeffe, M. & Groy, T. L. (2004). *Acta Cryst.* A**60**, 326–330.

Zuñiga, F. J., Palatinus, L., Cabildo, P., Claramunt, R. M. & Elguero, J. (2006). *Z. Kristallogr.* **221**, 281–287.