

# COMPUTER PROGRAMS

*J. Appl. Cryst.* (1997). **30**, 171–175

## **DISCUS: a program for diffuse scattering and defect-structure simulation**

TH. PROFFEN<sup>a</sup> AND R. B. NEDER<sup>b</sup> at <sup>a</sup>Research School of Chemistry, Australian National University, Canberra ACT 0200, Australia, and <sup>b</sup>Institut für Kristallographie und Mineralogie, LMU München, Theresienstrasse 41, 80333 München, Germany.  
E-mail: proffen@rsc.anu.edu.au

(Received 29 March 1996; accepted 29 September 1996)

### **Abstract**

The program *DISCUS* is a versatile tool for the analysis of diffuse scattering and for defect structure simulations. The model structure can be created from an asymmetric unit of a unit cell or a complete structure can be read from a file. A Fortran77 style interpreter that includes IF statements and various loops combined with predefined defect types like thermal displacements, waves and microdomains allows one to create all sorts of defect structures. The Fourier-transform segment of the program allows one to calculate neutron as well as X-ray intensities including isotropic temperature factors and anomalous scattering. The calculation of the inverse and difference Fourier transform as well as the Patterson function is also implemented. A model structure can be 'fitted' to observed diffuse scattering data by reverse Monte Carlo (RMC) simulations. The RMC segment allows one to model displacive as well as occupational disorder. The program is completely written in Fortran77 and the source code is available *via* the World Wide Web.

### **1. Introduction**

The analysis of diffuse scattering can be a tedious task. In contrast to the determination of the average crystal structure based on Bragg intensities, no 'general' procedure to obtain information about the defect structure of the studied material exists. Software for the analysis of diffuse scattering is mainly written to treat a diffraction theory for a special problem. Structure simulations have originally been used to create masks for optical transforms (Harburn, Taylor & Welberry, 1983). Computer simulations of defect structures and diffuse scattering were independently introduced by Neder & Wildgruber (1989) and Rahman (1989). Butler & Welberry (1992) give a detailed description of the procedures. The aim of the authors of *DISCUS* is to provide a program that allows one to treat a large variety of disorder problems. Many simulated defect structures start with an ideal crystal structure. *DISCUS* can build a model crystal for the given space-group symbol from the content of an asymmetric unit of the unit cell or read a complete structure from a file. The structure can be modified by predefined 'defect types'. In the present version of *DISCUS* (2.01), these types are thermal displacements according to the isotropic temperature factor, displacement and density waves as well as microdomains.

The program *DISCUS* is driven by a command language that includes a Fortran-like interpreter that allows one to modify a model structure in a very general way using variables, logical expressions and program loops. A sequence of commands can be stored in a macro file for later execution. It is the intention of

the authors to build up a library with macro files for different disorder problems. The Fourier-transform segment of the program allows the user to calculate the intensity distribution along a line or any plane through reciprocal space. *DISCUS* can also calculate scattering amplitude, phase, difference Fourier and Patterson function. The intensities can be obtained for neutron as well as X-ray scattering, optionally including temperature factors and, in the X-ray case, anomalous scattering. The results can be written in various output formats like Postscript and the portable anymap format PNM (Poskanzer, 1989).

A recent approach to the analysis of diffuse scattering is the reverse Monte Carlo (RMC) method (Nield, Keen & McGreevy, 1995). The procedure works as follows: an atom of the model crystal is selected at random and moved by a random amount. The scattering amplitude (in fact only the change due to the selected atom) is calculated and compared with the experimental data. If the move improves the agreement between observed and calculated data, it is accepted; otherwise, the move is only accepted with a certain probability (see §2.4). The RMC section of *DISCUS* allows one also to model occupational disorder. Here, a 'move' is understood as displacement of an atom or switching of the positions of two randomly selected atoms. The program *DISCUS* is able to simulate a wide range of defect structures and to calculate the Fourier transform, *i.e.* the results of a scattering experiment. The qualitative comparison between the diffraction pattern of a model structure and measured data allows one to get a further insight into a particular disorder problem. The RMC section of the program allows even a quantitative analysis of diffuse scattering data.

Besides the various applications in defect structure studies, *DISCUS* is also an excellent tool for teaching diffraction physics. A description of a lecture concept using *DISCUS* can be found in Neder & Proffen (1996).

### **2. Features**

The *DISCUS* program is accompanied by a detailed manual. The program has an on-line help function that allows access to command descriptions while working with the program. Extensive error checking is performed by the program to handle erroneous input data. The following section gives a brief introduction to the different functions of *DISCUS*.

#### *2.1. Creating a structure*

There are three different ways to create a structure using the program *DISCUS*. The crystal can be expanded from the content of an asymmetric unit using the space-group symbol, a complete structure can be read from a file or a structure can be created by the insertion of individual atoms into a general

triclinic space. The first way creates an initially strictly periodic crystal, the other two and especially the third way can also create an aperiodic structure. For each atom, the program stores its type as chemical-element symbol, its fractional coordinates ( $x, y, z$ ) and its isotropic temperature factor  $B$ . Once a crystal is generated, several tools are provided to modify the crystal. The first group of tools affecting only a single atom allows one to change its chemistry, change its position, delete or copy the atom and insert new atoms. These tools combined with logical expressions and loops of the command language (see §2.5) allow one to introduce all kinds of defects to the crystal. Some common defect types, however, are realized as separate commands in *DISCUS*. This second group of tools includes thermal displacements, waves and microdomains.

The wave section of the program allows one to use transversal and longitudinal displacement waves as well as density waves. Three different wave forms are available: sinusoidal, square and saw-toothed waves. The user can select those atoms that will be affected by the wave tool. The microdomain tool gives the user the possibility to introduce microdomains (small disordered regions in the lattice of the host crystal) into the model crystal. *DISCUS* offers the possibility to define different microdomain types and distribute these types throughout the crystal. *DISCUS* allows one to define the following properties to characterize the individual microdomains and their distribution within the crystal: shape, size, content and orientation. The microdomain distribution throughout the crystal can be on a perfect or paracrystalline superlattice or be a random distribution. Furthermore, the program allows one to introduce correlations between the different microdomain types or orientations. The microdomain section is the most powerful disorder tool implemented in *DISCUS* so far and the options are laid out in a very general fashion in order to facilitate generation of many different types of defects. With this technique, small and extended defects, antiphase domains, finite wave trains or an incoherent intergrowth of two phases can be generated. A created defect structure can be saved to a file and read again by *DISCUS* for later use.

### 2.2. Viewing and analysing a structure

After creating a defect structure or getting a modified structure from an RMC run (see §2.4), the user might want to analyse or view the structure. *DISCUS* can export atom coordinates as  $xy$  files that can be viewed by a number of plotting programs. The user can select the plane and the atom type he or she wants to be extracted. Furthermore, the program can also extract only atoms that are located within or outside of microdomains.

The analysing part of *DISCUS* allows one to get statistical information about the stored structure. The relative abundances of the elements can be calculated. Vacancies are internally stored as atoms of the type 'VOID' and can be used as all other atom types. The program allows one to calculate neighbouring frequencies of atom types and directions defined by the user. Furthermore, bond-length distributions can be extracted as  $xy$  files for graphical presentation. Finally, *DISCUS* can calculate the average structure including the mean displacements from this average position.

### 2.3. Fourier transform

The program *DISCUS* calculates the Fourier transform according to the standard formula for kinematic scattering

[(1a)]:

$$F(h) = \sum_{i=1}^N f_i(h) \exp(2\pi i h r_i) \quad (1a)$$

$$F(h) = \sum_{i=1}^N f_i(h) \exp(2\pi i h r_i) \exp(-B|h|^2/4) \quad (1b)$$

The sum is over all  $N$  atoms in the crystal, where  $f_i$  is the atomic form factor in case of X-rays or the scattering length for neutron scattering and  $r_i$  the fractional coordinate of the atom. The sum is calculated at all points  $h$  in reciprocal space. The form factors for X-ray and neutron scattering for all atoms are stored within the program. The influence of thermal vibrations can be realized in two different ways: A built-in command can be used to randomly displace the atoms according to their isotropic temperature factors and the Fourier transform is calculated by (1a). This corresponds to a snapshot in time, where the radiation sees the structure at rest. Alternatively, the user can calculate the intensity according to (1b) using the Debye–Waller factor, which corresponds to a situation where the radiation sees the time average of the structure. However, the use of (1b) is strictly correct only for (elastic) neutron scattering. The method of explicitly calculating the Fourier transform has several advantages compared to a fast Fourier transform (FFT). The main disadvantage of the much faster FFT is the need to represent the structure by setting a single pixel in a three-dimensional array sized  $2^n \times 2^n \times 2^n$ . This technique would require a  $1024 \times 1024 \times 1024$  FFT array to represent a structure of  $10 \times 10 \times 10$  unit cells, which allows for atomic displacements of an accuracy of about 1% of the unit-cell dimensions. This is, even for modern powerful computers, an enormous task. Furthermore, this description works well only for neutron scattering, whereas X-ray diffraction is approximated only very crudely. On the other hand, the translation of a true representation of the scattering density in the FFT array completely cancels the speed advantage of the FFT. Furthermore, the explicit Fourier transform allows one to calculate the diffraction pattern for sections of any size or shape in reciprocal space. For further discussion about this topic, see Neder & Wildgruber (1989), Rahman (1989) or Butler & Welberry (1992).

The speed at which diffuse diffraction patterns can be calculated is of great importance and efforts to speed up this part of the program are still in progress. As a first step, the time-consuming calculation of the trigonometric functions needed for the Fourier transformation have been replaced by a look-up table. This table is calculated at program start-up. The size of the look-up table, *i.e.* the accuracy of the look-up values, can be adjusted before compiling the program. There are two problems to consider when simulating diffraction patterns: First, simulated crystals are still small; second, all atoms in the crystal are treated as if they scatter coherently. The first problem causes finite-size effects that can be in general a significant part of the calculated diffraction pattern. They can be avoided by the application of periodic boundary conditions:  $\Delta h = 1/\Delta x$ , where  $\Delta h$  is the grid size of the calculated reciprocal layer and  $\Delta x$  is the corresponding dimension of the model crystal. In other words, the whole crystal is defined as a single unit cell of an imaginary supercrystal and the calculated points in reciprocal space are the Bragg reflections of this supercrystal. It is the user's responsibility to make sure the periodic boundary conditions are matched. The second problem arises from the

fact that in the program all atoms of the model crystal are treated as if they scatter coherently, since the amplitudes are added. For large simulated crystals, this may lead to unexpected high-frequency oscillations that severely modulate the diffuse scattering. If that kind of oscillation is observed, the best strategy is the use of several smaller crystals followed by the summation of the individual intensities. A detailed discussion of these problems is given in the *DISCUS* manual.

The program can output the scattering intensity, the real and imaginary part, the amplitude and the phase angle of the Fourier transform. The current version (2.01) of *DISCUS* includes the possibility of inverse and difference Fourier transformation as well as the calculation of the Patterson function. A variety of input formats can be read for these inverse transforms, including output lists from *SHELXL93* (Sheldrick, 1993).

#### 2.4. Reverse Monte Carlo simulations

In general, Monte Carlo methods can be described as statistical simulation methods involving sequences of random numbers to perform the simulation. An excellent application of these kinds of statistical methods is the study of diffuse scattering and subsequently the solution of the underlying defect structure. The RMC method (Nield *et al.*, 1995) works

similarly to the 'direct' Monte Carlo method (Welberry & Butler, 1994) but rather than minimization of the total energy of the crystal, the difference between observed and calculated intensity is minimized. The RMC process starts with the selection of a random site within the model crystal that is altered by a random move. Those moves can be either shifts or a switch with another randomly selected atom. The scattering intensity is recalculated for the generated move and the goodness-of-fit parameter  $\chi^2$  is computed:

$$\chi^2 = \sum_{i=1}^N [I_{\text{obs}}(q_i) - I_{\text{calc}}(q_i)]^2 / \sigma^2. \quad (2)$$

The sum in (2) is over all measured data points  $q_i$ .  $I_{\text{obs}}$  is the experimental and  $I_{\text{calc}}$  the calculated intensity. The parameter  $\sigma$  is taken as independent of  $q$  and is treated as a parameter of the modelling. The change of the goodness of fit from the configuration 'old' to 'new' is given by  $\Delta\chi^2 = \Delta\chi_{\text{new}}^2 - \Delta\chi_{\text{old}}^2$ . Every move that improves the fit ( $\Delta\chi^2 < 0$ ) is accepted. Those moves that worsen the fit ( $\Delta\chi^2 > 0$ ) are accepted with the probability of  $P = \exp(-\Delta\chi^2/2)$ . The parameter  $\sigma$  acts like the temperature in the direct Monte Carlo method and controls how many 'bad' moves will be accepted. The present version of *DISCUS* allows three different types of RMC moves: Gaussian-distributed random shifts of an atom,

```

1  r[0]=0.85
2  r[2]=0.5
3  i[1]=20
4  i[2]=20
5  i[3]=1
6  #
7  read
8  cell si.c11,i[1],i[2],i[3]
9  #
10 do i[6]=1,i[3]
11 do i[5]=1,i[2]
12 #
13 i[4]=1
14 i[7]=(i[6]-1)*i[2]*i[1] + (i[5]-1)*i[1] + i[4]
15 if(ran(0).lt.r[2]) then
16 remove i[7]
17 endif
18 #
19 do i[4]=2,i[1]
20 #
21 i[7]=(i[6]-1)*i[2]*i[1] + (i[5]-1)*i[1] + i[4]
22 r[1]=ran(0)
23 #
24 if( m[i[7]-1] .eq. 0 ) then
25 #
26 if( r[1].lt.r[0]) then
27 remove i[7]
28 endif
29 else
30 #
31 if( r[1].lt. (1-r[0]) ) then
32 remove i[7]
33 endif
34 endif
35 enddo
36 enddo
37 enddo
38 #
39 @plot sro.struc,si
40 #
41 @fourier -0.5,-0.5,0.0, 3.5,-0.5,0.0, -0.5,3.5,0.0,
42 #
43 @output sro.inten,inten,stan

```

(a)

```

1  four
2  xray
3  wvle cual
4  #
5  ll $1,$2,$3
6  lr $4,$5,$6
7  ul $7,$8,$9
8  na $10
9  no $11
10 #
11 show
12 run
13 exit

```

(b)

Fig. 1. (a) Macro file for the short-range-order example (see text). (b) Macro file for calculation of the Fourier transform called by the example macro.

switch of the atom type of two sites and finally the switch of the displacement from the average site of two atoms. The last type does not change the overall displacement (temperature factor) or the average structure, in contrast to the introduction of 'new' shifts. The program can either use fixed scaling and background parameters or adjust one or both parameters after each accepted move. The RMC segment allows one to use any combination of X-ray and neutron data or data measured at different wavelengths simultaneously. The RMC method might lead to a good agreement between calculated and observed scattering data but the resulting structure might be unlikely or even impossible from a chemical point of view. Several user-defined parameters within *DISCUS*, like a minimal distance between the different atom types, allow one to avoid some of those 'impossible' configurations. In the authors' opinion, RMC modelling is a powerful tool for the analysis of diffuse scattering especially when one is starting from an already disordered structure; however, the resulting structure of those calculations has to be analysed carefully.

### 2.5. Command language and Fortran-style interpreter

The program *DISCUS* is controlled by a command language that includes the use of variables and a Fortran-style interpreter. The interpreter provides variables linked to structural information (e.g. size of the crystal, coordinates or atom type of a certain site etc.) and free variables, loops, logical constructions, basic arithmetic and built-in functions. The program supports Fortran77-like IF-ELSEIF-ENDIF control constructions that use the same kind of logical expressions as in Fortran77 programs. Loops can be programmed using the DO command. The first kind of loop (DO ... ENDDO) is equivalent to the Fortran77 DO loop and is executed in a predefined number of cycles. In addition, two other DO loops are implemented in *DISCUS*: a loop that is executed while a logical expression is true (DO WHILE ... ENDDO) and one that is executed until a logical expression is true (DO ... ENDDO UNTIL). Loops and if constructions can be nested up to a level defined by a parameter that can be adjusted when the program is compiled.

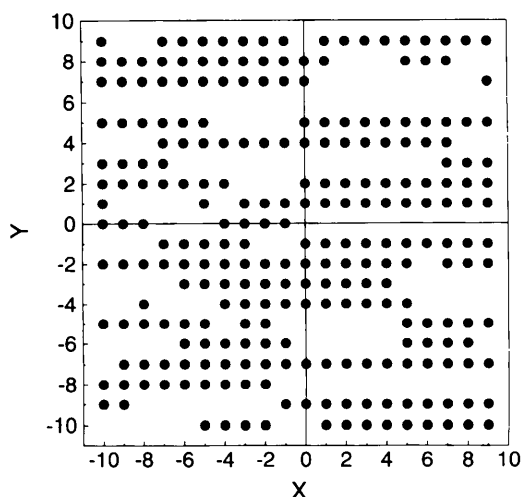


Fig. 2. Plot of the  $xy0$  plane of the created defect structure. The filled circles represent the remaining Si atoms after the 'sro' macro has been executed.

A BREAK command allows one to conditionally terminate the execution of one or several nested loops and if constructions. The arithmetic expressions understood by *DISCUS* include the operators +, -, \*, / and \*\* (power) as well as brackets.

All trigonometric functions (SIN, COS, TAN) as well as their inverse functions (ASIN, ACOS, ATAN) are present. Arithmetic functions include square root (SQRT), absolute value (ABS) and exponential (EXP). Furthermore, conversion functions between real and integer variables are included (INT, NINT, FRAC, MOD) and random generators for uniformly and Gaussian-distributed values. Additionally, crystallographic functions are implemented to calculate bond angles, lengths and angles of vectors in direct and reciprocal space as well as scalar and vector products in real and reciprocal space. Another powerful capability of *DISCUS* is the use of macros. A macro is simply a set of commands stored in a file. Macros can be nested, i.e. macro files can be called within macros. It is also possible to pass variables to the macro, which allows one to create a macro library for individual standard procedures. All input and output file names can be numbered automatically. This capability allows one to perform a cycle of different simulations and to use a counter to distinguish the output results.

### 2.6. Input/output formats

The structure files are text files containing a title, space-group symbol and lattice parameters followed by name (CHARACTER\*4), coordinates and isotropic temperature factor for each atom. This format is valid for input and output of structural information. The results of Fourier-transform calculations can be stored in different formats: Postscript, data files to be displayed with *KUPL* (Proffen, 1995), PNM files (Poskanzer, 1989) to be used with various programs and finally  $xyz$  files to be read by *GNUPLOT* (Williams & Kelley, 1989). It should be mentioned that the *PBMPLUS* package (Poskanzer, 1989) enables the user to convert the *DISCUS* output to most graphic formats. The structure can be displayed by extraction of a layer of the model crystal in an  $xy$  file that can be read by common plotting programs like *KUPL* or *GNUPLOT*. For more information about the programs used to display and convert

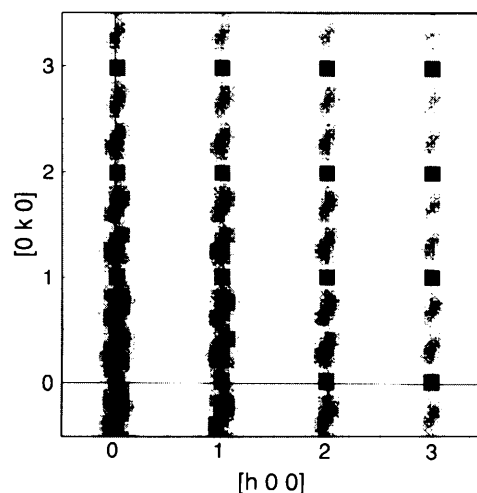


Fig. 3. Contour plot of the  $hk0$  plane of the Fourier transform (X-ray Cu  $K\alpha$  radiation) of the created defect structure (see text for details).

*DISCUS* output, see the *DISCUS* World Wide Web (WWW) pages (see below).

### 3. Example

The features and usage of *DISCUS* are illustrated by a basic example that is part of the program distribution. The example macro creates a two-dimensional crystal that shows occupational short-range order in the [100] direction. The macro file is shown in Fig. 1(a). The leading numbers are for reference and not part of the macro itself. Lines 1–5 initialize the variables used,  $r[0]$  represents the short-range order parameter, here set to a value of 0.85, *i.e.* preferred clustering of occupied or vacant sites. The variable  $r[2]$  represents the probability for an atom at the beginning of a row. The values of  $i[1]$ ,  $i[2]$  and  $i[3]$  define the size of the simulated crystal. In lines 7 and 8 the structure is expanded from the information in the file 'si.cll', which contains a single Si atom on (0, 0, 0). The subsequent DO loops (lines 10 and 11) are executed for every unit cell in the  $y$  and  $z$  directions. The index of the first atom in the row is calculated (lines 13 and 14). The atom is removed with a probability of  $r[2]$  (lines 15–17). Then, a loop over all atoms of that row is started (line 19) and if the previous position is vacant the atom is removed with the probability of  $r[0]$  (lines 24–28); otherwise, it is removed with the probability of  $1 - r[0]$  (lines 31–33). The resulting structure (Fig. 2) is stored for plotting by calling of the macro 'plot' with the given parameters (line 39). Then, the Fourier transform (Fig. 3) is calculated by calling of the macro 'fourier' with parameters giving the corners and the number of points of the reciprocal plane to be calculated (line 41). The contents of the fourier macro file are given in Fig. 1(b). Line 1 starts the Fourier segment of the program. The radiation (X-rays) and the wavelength (Ca  $K\alpha$ ) are selected next (lines 2 and 3). The plane in reciprocal space to be calculated and the number of grid points are specified by the commands in lines 5 to 9. The parameters \$1, \$2 ... \$11 are replaced by the values given in line 41 of Fig. 1(a). After a summary of the current setting is displayed (line 11), the Fourier transform is finally executed (line 12). To get back to the main command level of *DISCUS*, the 'exit' command is required at the end of the macro (line 13). Back in the example macro, the results are saved by execution of the 'output' macro file (line 43). Note that *DISCUS* stores all results internally until they are written to output files. For more information and a listing of the other macros used, again refer to the *DISCUS* WWW pages given below.

This very basic example was used to demonstrate the capabilities and easy use of the program *DISCUS*. Applications of *DISCUS* to 'real' crystallographic problems include the study of surface relaxations on X-ray intensities of microcrystals

(Neder, 1995) and the simulation of superstructures of the decagonal phase  $\text{Al}_{70}\text{Ni}_{15}\text{Co}_{15}$  (Neder, Hradil & Frey, 1995).

### 4. Language, documentation and availability

The program *DISCUS* is written in standard Fortran77 and was successfully tested on the following platforms: PC (Lahey Fortran Compiler, MSDOS), PC under LINUX (using f2c), Sun Sparcstation (SUNOS 4.1.3), HP workstation and VMS (VAX and ALPHA). All system-specific routines are stored in a separate file (sys\_'operatingsystem'.f). In the authors' opinion, *DISCUS* can be easily ported to other platforms given a Fortran77 compiler. The necessary amount of memory and disk space depends on the size of the intended model crystals. All array dimensions are defined *via* PARAMETER commands in separate include files in order to allow easy modification for individual needs. The *DISCUS* distribution contains the complete source code including Makefiles for various platforms, a manual of about 200 pages (Postscript) and various example macro files. Additional information about the program and the most recent *DISCUS* distribution can be found on the following WWW servers: <http://www.kri.physik.uni-muenchen.de/geo/crystal/discus/discus.html> (Germany) and <http://rschp2.anu.edu.au:8080/proffen/discus/discus.html> (Australia). Those who have difficulties accessing these WWW pages can obtain a copy directly from the authors.

This work was supported in part by funds of the DFG (grant no. Pr 527/1-1).

### References

- Butler, B. D. & Welberry T. R. (1992). *J. Appl. Cryst.* **25**, 391–399.
- Harburn, G., Taylor, C. A. & Welberry, T. R. (1983). *Atlas of optical transforms*. London: Bell & Hyman.
- Neder, R. B. (1995). *Z. Kristallogr.* **210**, 415–417.
- Neder, R. B., Hradil, K. & Frey, F. (1995). *Z. Kristallogr. Suppl.* **9**, 382.
- Neder, R. B. & Proffen, Th. (1996). *J. Appl. Cryst.* **29**, 727–735.
- Neder, R. B. & Wildgruber, U. (1989). *Z. Kristallogr.* **186**, 209.
- Nield, V. M., Keen, D. A. & McGreevy, R. L. (1995). *Acta Cryst.* **A51**, 763–771.
- Poskanzer, J. (1989). *PBMPLUS software package*. <ftp://sunsite.amu.edu.au>.
- Proffen, Th. (1995). *KUPL. A command language driven plot program*. <ftp://ftp.physik.tu-muenchen.de>.
- Rahman, S. H. (1989). *Z. Kristallogr.* **186**, 116.
- Sheldrick, G. M. (1993). *SHELXL93. Crystal structure refinement program*. University of Göttingen, Germany.
- Welberry, T. R. & Butler, B. D. (1994). *J. Appl. Cryst.* **27**, 205–231.
- Williams, T. & Kelley, C. (1986). *GNUPLLOT. Interactive plot program*. <ftp://sunsite.amu.edu.au>.