DIFFaX v 1.812

by

M. M. J. Treacy, M. W. Deem & J. M. Newsam

3rd July 2005

EMT, $N = \infty$

EMT, $N = 20$
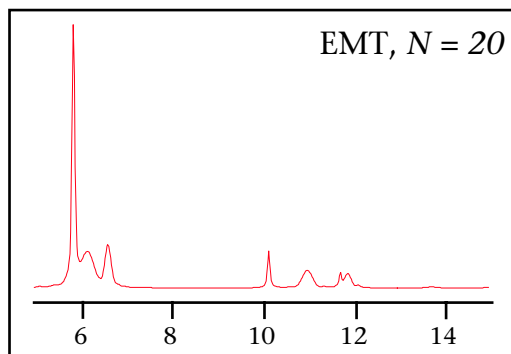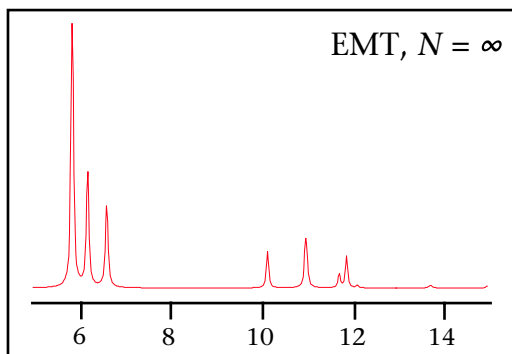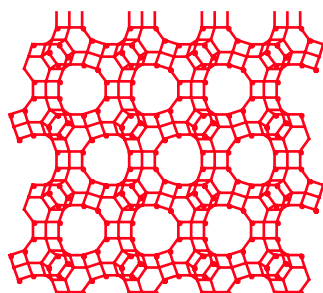
manual written by
Michael M. J. Treacy

DIFFaX v1.812

3rd July, 2005

Contents

List of Figures

1 Introduction.

1.1 What is DIFFaX?

DIFFaX is a fortran computer program that calculates diffraction intensities from crystals that contain coherent planar defects, such as twins and stacking faults. DIFFaX exploits the recurring patterns found in randomized stacking sequences to compute the average interference wavefunction scattered from each layer type occurring in a faulted crystal [1]. It is then a simple step to calculate the incoherent intensity contribution at hkl. Although the physics behind the diffracted intensity algorithm used by DIFFaX is basically the same as that used by other methods, most notably the Hendricks & Teller [2], the difference equation [3], and the Cowley methods [4], the recursion algorithm is significantly simpler to understand. This simplicity prompted the writing of this general purpose program, which can accommodate any type of coherent planar defect in any crystal system. A brief description of the DIFFaX algorithm is presented in section 8 on page 42. The details of the algorithm are described in reference [1].

1.2 About this manual.

The DIFFaX fortran code was written with portability in mind. The source code `DIFFaX.f` is pretty much ansi77-compliant and should compile and run, with few changes, on most modern computers. To run, DIFFaX assumes that a standard tty-type of input/output window is available, such as that found on unix machines. When compiling `DIFFaX.f` on a Macintosh or a Windows-based PC, systems which use a graphical user interface, the compiler must supply a tty-type window that is dedicated to DIFFaX.

The primary goal of this manual is to help explain to the reader how to construct a DIFFaX data file, and how to run the DIFFaX program on this file. Although we have tried to make DIFFaX as simple to run, and as flexible as possible, the preparation of a data file that accurately describes a particular faulted structure can be a considerable challenge. It is the hope that this manual provides enough examples and explanations to simplify the data file building process.

2 Creating structure data files.

DIFFaX structure data files are text files that contain information about the type of radiation to simulate, the structure of the layers, and details of the interlayer connectivities. Although the DIFFaX data file format is straightforward, in the author's experience the creation of error-free files is surprisingly difficult. This difficulty arises because of the unconventional way in which DIFFaX needs the defect structure to be described. Conventionally, crystals are thought of in terms of unit cells, which in turn can be broken down further into asymmetric sub-units of atoms and a set of space group symmetry operators. To use DIFFaX we need to think of crystals in terms

of sheets of atoms, or layers, which can interconnect via stacking operations that occur with a certain probability. Frequently, the plane of the layers will not coincide conveniently with any of the unit cell faces of the parent crystal. This requires a transformation of atom coordinates to a new cell system. For example, to describe 111 twins in face centered cubic materials, the layers (111 planes, and their mirrored counterparts) must be transformed to a hexagonal setting where the new c-axis is parallel to the old cubic [111] direction. The new hexagonal a- and b- axes now lie in the plane of the old cubic 111 layers, and define a two-dimensional hexagonal mesh that is common to all layers in the crystal. The atom coordinates and stacking vectors will need to be defined in terms of the hexagonal setting. Great care needs to be taken when creating the data set. It is wise to compare diffraction spectra for the unfaulted end-members, with those generated from a standard diffraction calculation program to ensure that you are starting from a valid data set. To help the user to avoid wasting time on flawed data files, a conscious design decision was made to make DIFFaX unforgiving when it discovers errors or inconsistencies in the data file. Usually, when a potential problem is found, DIFFaX will quit, forcing the user to repair the data file. This "zero-tolerance" strategy has saved the present author from numerous embarrassing blunders.

2.1   Data line lengths.

The maximum data line length is 200 characters. Anything typed beyond the 200th character may be lost, and may generate error messages. The accepted line length has been increased from the obsolete "punched card" value of 80 characters that applied in DIFFaX versions 1.7xx. 200 characters should be sufficiently wide to handle most reasonable data input lines.

2.2   Case sensitivity.

Data can be written in either upper- or lower-case. DIFFaX is insensitive to the case of the text. However, DIFFaX does assume that it is reading ASCII-defined characters. Some of the text manipulating subroutines will not work if the computer system is not ASCII-based.

2.3   Comments in the data file.

Comments can be embedded in the data files. Anything written between curly braces, *i.e.* {this is a comment}, will be ignored. Comments can be nested, and there can be more than one comment per line. All comments must begin and end on the same line; DIFFaX will assume that the data is faulty if it cannot match pairs of braces on a line. Total line lengths, including comments and braces, must not exceed 200 characters.

2.4 Backus-Naur Description of the datafile.

In Backus-Naur form, the data file format is as follows

```
{user input data file -- in Backus-Naur form}
{comments are in curly braces, blank lines are allowed}
{The pipe symbol | implies that one of a series of arguments must be chosen}
{arguments in parentheses () are optional}

{Header for radiation data} INSTRUMENTAL
{radiation type} X-RAY | NEUTRON | ELECTRON
{wavelength in Å} λ

{Instrumental broadening type} NONE
                              | GAUSSIAN half-width | u v w (TRIM)
                              | LORENTZIAN half-width | u v w (TRIM)
                              | PSEUDO-VOIGT u v w σ (TRIM)

{Header for structure data} STRUCTURAL
{cell dimensions, in Å and °} a b c γ
{diffraction point group symmetry} -1  | 2/M(1) | 2/M(2) | MMM
                                   | -3  | -3M    | 4/M    | 4/MMM
                                   | 6/M | 6/MMM  | AXIAL
                                   | UNKNOWN (% tolerance)
{total number of layer types} n
{in-plane layer widths in Å (optional)} ('width along a' 'width along b')
                                        | ('average diameter')
                                        | (INFINITE)

{New layer definition} LAYER 1
{Layer structure symmetry} NONE | CENTROSYMMETRIC
{Atom data record}
name # x_rel y_rel z_rel 'isotropic Debye-Waller factor B_iso' occupancy
.
.
.

{New layer definition} LAYER number | LAYER = a previous layer number
{If this is not equivalent to a prior layer, then enter ...}
{Layer structure symmetry} NONE | CENTROSYMMETRIC
{Atom data record}
name # x_rel y_rel z_rel 'isotropic Debye-Waller factor B_iso' occupancy
.
.
.

{New layer definition} LAYER number | LAYER = previous layer number
{If this is not equivalent to a prior layer, then enter ...}
{Layer structure symmetry} NONE | CENTROSYMMETRIC
{Atom data record}
name # x_rel y_rel z_rel 'isotropic Debye-Waller factor B_iso' occupancy
.
.
.

name # x_rel y_rel z_rel 'isotropic Debye-Waller factor B_iso' occupancy

{Header for stacking description} STACKING
{How to treat layer sequencing} EXPLICIT | RECURSIVE
{If EXPLICIT}
{either, an explicit sequence} lay# lay# lay# lay# lay# lay# lay# lay#
```

```
lay# lay# lay# lay# lay# lay# lay# lay# lay# lay# lay# lay# lay# lay#
.
.
.
{or, generate sequence} RANDOM 'number of layers in crystal'
{If RECURSIVE}
{number of layers in crystal} 'number of layers' | INFINITE

{Header for stacking parameters} TRANSITIONS
{stacking probability, stacking vector (optional anisotropic stacking
uncertainty factor)}
```

α_{11} R^x_{11} R^y_{11} R^z_{11} $(C^{11}_{11}$ C^{22}_{11} C^{33}_{11} C^{12}_{11} C^{13}_{11} $C^{23}_{11})$

α_{12} R^x_{12} R^y_{12} R^z_{12} $(C^{11}_{12}$ C^{22}_{12} C^{33}_{12} C^{12}_{12} C^{13}_{12} $C^{23}_{12})$

$\vdots$

α_{1n} R^x_{1n} R^y_{1n} R^z_{1n} $(C^{11}_{1n}$ C^{22}_{1n} C^{33}_{1n} C^{12}_{1n} C^{13}_{1n} $C^{23}_{1n})$

α_{21} R^x_{21} R^y_{21} R^z_{21} $(C^{11}_{21}$ C^{22}_{21} C^{33}_{21} C^{12}_{21} C^{13}_{21} $C^{23}_{21})$

α_{22} R^x_{22} R^y_{22} R^z_{22} $(C^{11}_{22}$ C^{22}_{22} C^{33}_{22} C^{12}_{22} C^{13}_{22} $C^{23}_{22})$

$\vdots$

α_{2n} R^x_{2n} R^y_{2n} R^z_{2n} $(C^{11}_{2n}$ C^{22}_{2n} C^{33}_{2n} C^{12}_{2n} C^{13}_{2n} $C^{23}_{2n})$

$\vdots$

α_{n1} R^x_{n1} R^y_{n1} R^z_{n1} $(C^{11}_{n1}$ C^{22}_{n1} C^{33}_{n1} C^{12}_{n1} C^{13}_{n1} $C^{23}_{n1})$

α_{n2} R^x_{n2} R^y_{n2} R^z_{n2} $(C^{11}_{n2}$ C^{22}_{n2} C^{33}_{n2} C^{12}_{n2} C^{13}_{n2} $C^{23}_{n2})$

$\vdots$

α_{nn} R^x_{nn} R^y_{nn} R^z_{nn} $(C^{11}_{nn}$ C^{22}_{nn} C^{33}_{nn} C^{12}_{nn} C^{13}_{nn} $C^{23}_{nn})$

2.5 An example data file

A data file describing random twin faults in diamond, {including comments in braces} is listed below. See appendix A.1 for a more complete description.

```
{data file for cubic diamond, with random intergrowths of hexagonal lonsdaleite}
{probability of cubic stacking = 70%}

INSTRUMENTAL                              {Header for instrumental section}
X-RAY                                     {Simulate X-ray diffraction}
1.5418                                    {X-ray wavelength}
PSEUDO-VOIGT 0.89 -0.32 0.08 0.6 trim     {Instrumental broadening}

STRUCTURAL                                {Header for structural section}
2.52 2.52 2.06 120.0                      {unit cell coordinates}
6/MMM                                     {hexagonal, c axis = cubic [111]}
2                                         {2 layer types:  111 sheet, plus its mirror}
infinite                                  {infinite layer width}

LAYER 1                                   {cubic (111) layer, centrosymmetric}
CENTROSYMMETRIC
 C 1 -0.333333 -0.166667 -0.125 1.0 1.0
{C 1  0.333333  0.166667  0.125 1.0 1.0 -the 2nd centrosymmetric atom}
{                                 -does not need to be declared}

LAYER 2                                   {mirror image of layer 1}
CENTROSYMMETRIC
 C 1  0.333333  0.166667 -0.125 1.0 1.0
{C 1 -0.333333 -0.166667  0.125 1.0 1.0 -the 2nd centrosymmetric atom}
{                                 -does not need to be declared}

STACKING                                  {Header for stacking description}
recursive                                 {Statistical ensemble}
infinite                                  {Infinite number of layers}

TRANSITIONS                               {Header for transitions}
{Transitions from layer 1}
0.7  0.666667  0.333333  1.0              {layer 1 to layer 1, 70% chance}
0.3  0.0       0.0       1.0              {layer 1 to layer 2, 30% chance}

{Transitions from layer 2}
0.3  0.0       0.0       1.0              {layer 2 to layer 1, 30% chance}
0.7 -0.666667 -0.333333  1.0             {layer 2 to layer 2, 70% chance}
```

The same file, expressed more tersely, could read

```
INSTRUMENTAL
X-RAY
1.5418
PSEUDO-VOIGT .89 -.32 .08 .6 trim
STRUCTURAL
2.52 2.52 2.06 120
6/MMM
2
LAYER 1
CENTROSYMMETRIC
C 1 -1/3 -1/6 -1/8 1 1
LAYER 2
CENTROSYMMETRIC
C 1 1/3 1/6 -1/8 1 1
STACKING
recursive
infinite
TRANSITIONS
.7 2/3 1/3 1
.3 0 0 1
.3 0 0 1
.7 -2/3 -1/3 1
```

Note that the data line relating to the layer widths can be omitted (this exception ensures backwards compatibility with earlier DIFFaX versions). All other data lines must appear in the correct sequence.

In DIFFaX v1.804 and later, atom coordinates and stacking vectors can be expressed as fractions. For irrational fractions such as 1/3, this ensures maximum machine precision.

We now describe the data file format in more detail.

2.6   INSTRUMENTAL section.

2.6.1   INSTRUMENTAL header.

Line 1: The first non-comment, non-blank line must contain the keyword INSTRUMENTAL (either upper or lower case). No abbreviations are accepted. The data that follows describes the type of radiation and the instrumental conditions.

2.6.2   The radiation type.

Line 2: The choice is X-RAY, NEUTRON or ELECTRON, in either upper- or lower-case. No abbreviations are accepted.

2.6.3   The radiation wavelength.

Line 3: The numerical value of the radiation wavelength in Ångstroms.

2.6.4 The instrumental broadening.

Line 4: The type of instrumental broadening to simulate in powder pattern calculations. Although instrumental broadening applies only to powder pattern calculations, this line must always be supplied. There are four types supported by DIFFaX namely `NONE`, `GAUSSIAN`, `LORENTZIAN` and `PSEUDO-VOIGT`. We now describe these options in more detail.

2.6.5 Instrumental broadening type, `NONE`.

The raw powder pattern data only is saved to file, with no instrumental broadening applied. The resolution of the spectrum is determined solely by the sampling step size $\Delta\theta$ (see page 34).

2.6.6 `GAUSSIAN` instrumental broadening.

The data is "convolved" with a Gaussian function, whose width may vary across the spectrum. A full-width half-maximum (NOT the standard deviation), in degrees, needs to be specified for this option. The line

```
GAUSSIAN 0.1
```

will specify a 0.1° constant full-width half-maximum Gaussian profile. The data line

```
Gaussian 0.0
```

is equivalent to specifying `NONE`. The full-width half-maximum must not be negative. The data line

```
Gaussian -0.1
```

will generate an error message. Alternatively, a variable full-width half-maximum $\Gamma(\theta)$ can be specified, having the form

$$\Gamma(\theta) = \sqrt{u\tan^2(\theta/2) + v\tan(\theta/2) + w} \tag{1}$$

u, v and w describe how the the full-width half-maximum (FWHM) varies as a function of the scattering angle θ. The intensity profile I_G of the peak centered at θ_1 is described by the equation

$$I_\mathrm{G}(\theta_1; \theta_1 - \theta_2) = \sqrt{\frac{4\ln(2)}{\pi\Gamma(\theta_1)^2}} \, \exp\left(\frac{-4\ln(2)(\theta_1 - \theta_2)^2}{\Gamma(\theta_1)^2}\right) \tag{2}$$

where $\theta_1 - \theta_2$ represents the angular distance from the peak center.

The standard deviation Δ of the peak intensity distribution is

$$\Delta(\theta_1) = \frac{\Gamma(\theta_1)}{\sqrt{8\ln(2)}} \tag{3}$$

The data line

```
Gaussian 0.1 -0.036 0.009
```

specifies a Gaussian broadening profile with $u = 0.1$, $v = -0.036$ and $w = 0.009$. Note that the variable FWHM option is computationally expensive.

2.6.7 `LORENTZIAN` instrumental broadening.

The data is "convolved" with a Lorentzian function, whose width may vary across the spectrum. As for the `GAUSSIAN` option, a full-width half-maximum, in degrees, needs to be specified. The full-width half-maximum must be positive. The entry

```
Lorentzian 0.1
```

will specify a 0.1 degree constant full-width half-maximum, Lorentzian instrumental broadening profile. The line

```
Lorentzian 0.0
```

is equivalent to specifying 'NONE'. In the same way as for the GAUSSIAN option, a variable full-width half-maximum can be specified, with associated u, v and w parameters. The full-width half-maximum is given by equation 1. For a Lorentzian profile, the broadening is described by the equation

$$I_{\mathrm{L}}(\theta_1; \theta_1 - \theta_2) = \frac{2\Gamma(\theta_1)}{\pi\left[\Gamma(\theta_1)^2 + 4(\theta_1 - \theta_2)^2\right]} \tag{4}$$

Thus, the data line

```
Lorentzian 0.1 -0.036 0.009
```

specifies a Lorentzian broadening profile with $u = 0.1$ $v = -0.036$ and $w = 0.009$. Note that the variable FWHM option is computationally expensive.

2.6.8 `PSEUDO-VOIGT` **instrumental broadening.**

The data is broadened according to the pseudo-Voigt function, which is a weighted sum of Gaussian and Lorentzian functions. The broadening can vary with scattering angle. The pseudo-Voigt function requires 4 parameters, u, v, w and σ.

$\quad$ u, v and w control the full-width half-maximum according to equation 1. σ controls the relative amount of Gaussian and Lorentzian components, and must have a value between 0 and 1. $\sigma = 0$ is equivalent to pure Gaussian everywhere, and $\sigma = 1$ is equivalent to pure Lorentzian everywhere. The pseudo-Voigt function is described by the equation

$$I_{\mathrm{PV}}(\theta_1; \theta_1 - \theta_2) = \frac{2\sigma\Gamma(\theta_1)}{\pi\left[\Gamma(\theta_1)^2 + 4(\theta_1 - \theta_2)^2\right]} + (1-\sigma)\sqrt{\frac{4\ln(2)}{\pi\Gamma(\theta_1)^2}}\ \exp\left(\frac{-4\ln(2)(\theta_1 - \theta_2)^2}{\Gamma(\theta_1)^2}\right) \quad (5)$$

The pseudo-Voigt function takes considerably longer to compute than do the Gaussian and Lorentzian options that specify a constant full-width half-maximum. The reason is that the full-width half-maximum changes as a function of scattering angle, and fresh Gaussian and Lorentzian profiles must be computed for each point in the spectrum. When the full-width half-maximum is constant, the broadening profile is computed once, and then rapidly applied to every point. Whenever possible, DIFFaX will attempt to speed up the computation if certain redundancies are detected. For example, the data line

$$\texttt{PSEUDO-VOIGT 0.1 -0.036 0.009 0.6}$$

specifies pseudo-Voigt broadening with $u = 0.1$, $v = -0.036$ $w = 0.009$ and $\sigma = 0.6$. However, the data line

$$\texttt{PSEUDO-VOIGT 0.0 0.0 0.0 0.6}$$

is equivalent to specifying `NONE`, and DIFFaX will not apply any instrumental broadening. The entry

$$\texttt{PSEUDO-VOIGT 0.0 0.0 0.16 1.0}$$

will be interpreted by DIFFaX as if

$$\texttt{LORENTZIAN 0.4}$$

had been specified, since in this instance the FWHM is given by $\Gamma = \sqrt{w}$ (see equation 1), which is independent of scattering angle. Similarly, the data line

$$\texttt{PSEUDO-VOIGT 0.0 0.0 0.16 0.0}$$

is read by DIFFaX as

$$\texttt{GAUSSIAN 0.4.}$$

It should be noted, however, that the `GAUSSIAN` and `LORENTZIAN` options, when supplied with non-zero u, v and w, call the pseudo-Voigt function with $\sigma = 0$ and 1, respectively, and are therefore computationally expensive functions.

For most powder pattern simulations, a `GAUSSIAN` or `LORENTZIAN` broadening profile with a constant full-width half maximum should be adequate, and are significantly faster computationally than the `PSEUDO-VOIGT` broadening profile.

2.6.9   `TRIM` keyword.

If one of `GAUSSIAN`, `LORENTZIAN`, or `PSEUDO-VOIGT` are specified, there is a further optional keyword that can be supplied, `TRIM`. `TRIM` tells DIFFaX to ignore intensity information close to the origin when simulating the instrumental broadening for a powder diffraction pattern. The peak at the origin is usually many orders of magnitude more intense than any other peak in the spectrum, and the background that is generated by the tails of this peak, when broadened, can easily swamp peaks close to the origin. When `TRIM` is specified, the unbroadened spectrum that is written to file retains the data at the origin, whereas the broadened data that is written to file suppresses the peak at the origin. It is generally desirable to specify `TRIM` whenever the powder pattern angular range includes the origin. `TRIM` has no effect if the powder pattern angular range requested by the user does not include the origin.

2.6.10   Examples of the `INSTRUMENTAL` data section.

Examples of a typical `INSTRUMENTAL` data section, with optional comments, might look like

```
INSTRUMENTAL
{Radiation type = } X-RAY
{wavelength = } 1.5418
{Instrumental broadening} PSEUDO-VOIGT 0.1 -0.036 0.009 0.6 trim
```

or

```
INSTRUMENTAL
electron      {Radiation type}
0.037         {wavelength}
none          {Instrumental broadening}
```

or simply

```
instrumental
x-ray
```

```
0.7
gaussian 0.1 trim
```

2.7 STRUCTURAL section.

The next part of the data section relates to the cell mesh that defines the symmetry of the crystal layers.

2.7.1 STRUCTURAL header.

Line 5: This section must begin with the keyword STRUCTURAL (either upper or lower case).

2.7.2 Cell dimensions.

Line 6: The cell dimensions, **a**, **b**, **c** and γ. All layers must share the same in-plane mesh dimensions, **a**, **b** and γ. The value of **c**, perpendicular to the **a-b** plane (and therefore parallel to the stacking direction), is needed to provide a reference scale along the stacking direction with which to define the z-component of the stacking vectors. **c** does not have to correspond to any special periodic dimension along the stacking direction for any of the layers. The data line

$$17.437 \ 17.437 \ 14.238 \ 120.0$$

defines a cell with $\mathbf{a} = \mathbf{b} = 17.437$ Å, $\mathbf{c} = 14.238$ Å, and $\gamma = 120°$. Note that the cell axes must be defined so that **c** is parallel to the fault direction, that is, perpendicular to **a** and **b**. This will not always coincide with the **c**-axis of the original crystal system that the layer was clipped from. In the present version of DIFFaX, it is up to the user to transform the cell axial system to that expected by DIFFaX.

2.7.3 Laue symmetry.

Line 7: The expected Laue symmetry of the diffraction data. DIFFaX uses the Laue group symmetry to speed up powder diffraction calculations by reducing the volume of reciprocal space that needs to be included in the intensity integration. It is usually best to specify UNKNOWN, and to let DIFFaX work out the symmetry. There are ten possible diffraction point groups in the presence of streaking along one axis. These ten are a subset of the 12 Laue point groups. The Laue groups are, in turn, that subset of the 32 crystallographic point groups which possess a center of symmetry so that the diffracted intensities obey Friedel's law. The two Laue groups, 3M and M3M, are excluded since the non-orthogonal 3-fold axes will be violated by any streaking that may occur along **c***. The user can specify one of

11

the ten "streaky" point groups, or, if the symmetry is not known, the keyword `UNKNOWN`, whereby DIFFaX will establish the symmetry by randomly sampling reciprocal space. If `UNKNOWN` is entered, the user can specify a % tolerance on intensity deviations for this search. The default tolerance in DIFFaX is 1% (*i.e.* if the symmetry turns out to be `4/MMM`, the maximum deviation in intensity between the hkl, $\overline{k}hl$, $\overline{hk}l$ and $k\overline{h}l$ reflections did not exceed the average intensity of the four reflections by more than one part in a hundred). A twelfth option, `AXIAL`, is provided, which constrains DIFFaX to integrate along the $00l$ axis only. This is useful for turbostratically disordered systems, where lateral coherence between layers is lost. The twelve options are

```
-1
 2/M(1) {diad along streaks}
 2/M(2) {diad perpendicular to streaks}
 MMM
 -3
 -3M
 4/M
 4/MMM
 6/M
 6/MMM
 AXIAL
 UNKNOWN (tolerance)
```

If the user suggests one of the ten point groups, DIFFaX will always check the user's choice, and will override it if the choice is inconsistent with the unit cell dimensions (*i.e.* if **a** is not equal to **b**, then `-3, -3M, 4/M, 4/MMM, 6/M, 6/MMM` are not possible). DIFFaX will provide a measure of the 'goodness of fit', which is the maximum deviation from the symmetry encountered during a random sampling of reciprocal space. In the author's experience, this point group symmetry parameter is useful for debugging new data files, particularly if the cell axes used by DIFFaX are not the same as those of the crystal system from which the layers were clipped.

As a rule, it is easiest and safest to specify `UNKNOWN`, and to let DIFFaX do the work. However, there are rare occasions when DIFFaX finds a higher symmetry than is actually the case. This is caused by the limited random sampling of reciprocal space (the default is 25 tests of symmetry-related hkl values per symmetry element). Erroneous high symmetries can be avoided by using one of three strategies. (1) Specifying a more stringent tolerance, *i.e.* `UNKNOWN 0.01` will specify a tolerance on the comparisons of 0.01% (one part in 10,000). Note that DIFFaX imposes a lower limit on the tolerance of 0.01%. (2) If the user already knows the symmetry, then simply declare it outright. (3) If the cautious user wishes to override the symmetry evaluation altogether, then specify '`-1`', the lowest possible symmetry consistent with Friedel's law. This option may take significantly longer to compute, since all

hkl ($l \geq 0$) are visited.

2.7.4   Number of layer types.

Line 8: Enter here the number of layer types that the faulted structure contains. Layers which are structurally identical, but exhibit different stacking behavior, count as distinct layers.

2.7.5   Layer characteristic widths.

Line 9: This line is optional, and was introduced with DIFFaX v1.801. Enter here the characteristic width, in Ångstroms, of the crystal along the cell **a** and **b** directions. These parameters are used to estimate a Debye-Scherrer type of peak broadening due to a finite layer width. There are several legal formats for this line;

```
'width along a' 'width along b'
'layer diameter'
'INFINITE'
```

In order to maximize compatibility with versions of DIFFaX preceding version 1.80, this line may be omitted altogether. In this instance an `INFINITE` layer diameter is the default.

Note that the peak-broadening contribution due to a finite number of layers stacking along **c** is handled differently by DIFFaX (see the `STACKING` description in section 2.9).

2.7.6   Examples of the `STRUCTURAL` data section.

Typical data entries for the first part of the `STRUCTURAL` section might look like

```
STRUCTURAL
{a, b, c, gamma} 17.437 17.437 14.238 120.0
{Expected diffraction point symmetry} 6/mmm
{Number of layers = } 4
{layer width (optional)} infinite
```

and

```
STRUCTURAL
{a, b, c, gamma} 4.437 3.431 1.2 87.1
{Point symmetry unknown, tolerance = 1%} unknown 1.0
{Number of layers = } 2
{layer widths along a and b} 200 600
```

2.8   Layer structure data.

The data that follows describes the structure of the layers themselves.

2.8.1 Layer header

Line 10: Header for the first layer. It must read `LAYER 1`. This can be in upper or lower case. Abbreviations are not accepted.

2.8.2 Layer symmetry

Line 11: Symmetry of the layer structure. Only two options are accepted, `NONE` and `CENTROSYMMETRIC`. If `CENTROSYMMETRIC`, only the asymmetric half of the atom coordinates need be entered, *taking care to halve the occupancy of those atoms that are shared by the two asymmetric halves.* If centrosymmetric, all layer scattering factors will be treated as having real values in DIFFaX with the complex component ignored. Therefore, specifying centrosymmetric can provide a useful gain in computation speed, particularly for complicated structures containing many atoms per layer. Other symmetries, such as mirror planes etc. are not supported by DIFFaX because they do not provide any opportunities for significant computational speedups.

2.8.3 Layer atom data

Line 12: First record of the layer atomic structure. For each atom, the data must be entered on a single line in the form

`name id# x_rel y_rel z_rel B_iso Occ`

where
`name` = atom name (4 character wide field),
`id#` = numeric specifier, (*i.e.* 1 for the first atom, 2 for 2nd etc...)
`x_rel` = x coordinate relative to a
`y_rel` = y coordinate relative to b
`z_rel` = z coordinate relative to c
`B_iso` = isotropic Debye-Waller factor ($\mathring{A}^2$)
`Occ` = occupancy factor (must be between 0 and 1)

Legal atom names are exactly 4 characters wide, including any spaces. A full list of the names accepted by DIFFaX is

```
'H   ' 'H.  ' 'D   ' 'H 1-' 'He  ' 'Li  ' 'Li1+' 'Be  ' 'Be2+' 'B   '
'C   ' 'C.  ' 'N   ' 'O   ' 'O 1-' 'O 2-' 'F   ' 'F 1-' 'Ne  ' 'Na  '
'Na1+' 'Mg  ' 'Mg2+' 'Al  ' 'Al3+' 'Si  ' 'Si. ' 'Si4+' 'P   ' 'S   '
'Cl  ' 'Cl1-' 'Ar  ' 'K   ' 'K 1+' 'Ca  ' 'Ca2+' 'Sc  ' 'Sc3+' 'Ti  '
'Ti2+' 'Ti3+' 'Ti4+' 'V   ' 'V 2+' 'V 3+' 'V 5+' 'Cr  ' 'Cr2+' 'Cr3+'
```

```
'Mn  '  'Mn2+'  'Mn3+'  'Mn4+'  'Fe  '  'Fe2+'  'Fe3+'  'Co  '  'Co2+'  'Co3+'
'Ni  '  'Ni2+'  'Ni3+'  'Cu  '  'Cu1+'  'Cu2+'  'Zn  '  'Zn2+'  'Ga  '  'Ga3+'
'Ge  '  'Ge4+'  'As  '  'Se  '  'Br  '  'Br1-'  'Kr  '  'Rb  '  'Rb1+'  'Sr  '
'Sr2+'  'Y   '  'Y 3+'  'Zr  '  'Zr4+'  'Nb  '  'Nb3+'  'Nb5+'  'Mo  '  'Mo3+'
'Mo5+'  'Mo6+'  'Tc  '  'Ru  '  'Ru3+'  'Ru4+'  'Rh  '  'Rh3+'  'Rh4+'  'Pd  '
'Pd2+'  'Pd4+'  'Ag  '  'Ag1+'  'Ag2+'  'Cd  '  'Cd2+'  'In  '  'In3+'  'Sn  '
'Sn2+'  'Sn4+'  'Sb  '  'Sb2+'  'Sb5+'  'Te  '  'I   '  'I 1-'  'Xe  '  'Cs  '
'Cs1+'  'Ba  '  'Ba2+'  'La  '  'La3+'  'Ce  '  'Ce3+'  'Ce4+'  'Pr  '  'Pr3+'
'Pr4+'  'Nd  '  'Nd3+'  'Pm  '  'Pm3+'  'Sm  '  'Sm3+'  'Eu  '  'Eu2+'  'Eu3+'
'Gd  '  'Gd3+'  'Tb  '  'Tb3+'  'Dy  '  'Dy3+'  'Ho  '  'Ho3+'  'Er  '  'Er3+'
'Tm  '  'Tm3+'  'Yb  '  'Yb2+'  'Yb3+'  'Lu  '  'Lu3+'  'Hf  '  'Hf4+'  'Ta  '
'Ta5+'  'W   '  'W 6+'  'Re  '  'Os  '  'Os4+'  'Ir  '  'Ir3+'  'Ir4+'  'Pt  '
'Pt2+'  'Pt4+'  'Au  '  'Au1+'  'Au3+'  'Hg  '  'Hg1+'  'Hg2+'  'Tl  '  'Tl1+'
'Tl3+'  'Pb  '  'Pb2+'  'Pb4+'  'Bi  '  'Bi3+'  'Bi5+'  'Po  '  'At  '  'Rn  '
'Fr  '  'Ra  '  'Ra2+'  'Ac  '  'Ac3+'  'Th  '  'Th4+'  'Pa  '  'U   '  'U 3+'
'U 4+'  'U 6+'  'Np  '  'Np3+'  'Np4+'  'Np6+'  'Pu  '  'Pu3+'  'Pu4+'  'Pu6+'
```

In DIFFaXversion 1.80, atom names are case insensitive. Thus 'SI ' and 'Si ' are equivalent.

An example data line for an atom might be

```
C     1 -0.333333 -0.166667 -0.125 1.0 1.0
```

This specifies that the first atom is neutral carbon, 'C␣␣␣', ('C' followed by 3 spaces), at cell position (-0.333333 -0.166667 -0.125) relative to **a**, **b** and **c**, with Debye-Waller B_{iso} factor of $1.0 Å^2$, and occupancy of 1. Although DIFFaX ignores leading spaces in data lines, care must be taken to ensure that the first non-blank character and the three succeeding characters, including blanks, are part of the atom name. The data line

```
C 1 -.333333 -.166667 -.125 1.0 1.0
```

would imply that the atom type is 'C 1 ' (*i.e.* 'C␣1␣'). This would generate an error, since DIFFaX would become confused about the subsequent arguments on the line. The data line

```
C 111 -.333333 -.166667 -.125 1.0 1.0
```

would imply that the first atom was of type 'C␣11', with an id number of 1. The subsequent parameters would be read correctly, but DIFFaX would be unable to find atomic scattering data for 'C␣11' in the file `data.sfc` and would display an error message and stop running.

The atom id number is simply a label to help you distinguish between identical atom names. DIFFaX uses these labels when it generates error messages, to identify which atom's data is suspect. The numbers do not need to be consecutive, and the same number can be used more than once, although this entails the risk of creating confusion if error messages are generated.

It is usual to enter the atomic relative coordinates so that they are within the range 0 to 1. However, DIFFaX will accept numbers beyond these bounds. Internally, DIFFaXtruncates coordinates to the 0–1 range.

Coordinates can also be entered as fractions, thus

```
C 1 -1/3 -1/6 -1/8 1.0 1.0
```

is acceptable.

The isotropic B_{iso} factors are in $Å^2$ units, and are equivalent to $8\pi^2 U_{iso}$, where U_{iso} is the mean square atomic vibration amplitude. The B_{iso} factors must not be negative.

The atom occupancy must be between 0 and 1. DIFFaX carries out a rudimentary check for other atoms that may be on (or near) the same position, including those atoms in potentially neighboring layers, which cumulatively might imply a site occupancy greater than 1.

Lines 13 to 13 + no_atoms(layer 1): Data for the remaining atoms in layer 1, entered as for line 12.

2.8.4 Second Layer.

Line 14 + no_atoms(layer 1): Header for the second layer. If it is structurally distinct from layer 1, it must read LAYER 2. If it is structurally identical to layer 1, then the header can be written LAYER 2 = 1. In this instance, no symmetry or atomic coordinate data is needed and the next line can continue with data for layer 3, or, if there are no more layers, the next section.

Line 15 + no_atoms(layer 1): If structurally distinct from layer 1, then enter symmetry of the layer structure, as for line 11.

Lines 16 + no_atoms(layer 1) to 16 + no_atoms(layer 1) + no_atoms(layer 2): Layer atomic structure data, as for line 12.

Continue adding data for the number of layers specified in line 8.

2.8.5 Examples of the layer structure data section.

Layer structure data describing clustered faulting in diamond might look like

```
LAYER 1
None
C    1   -0.333333   -0.166667   -0.125  1.0  1.0
C    2    0.333333    0.166667    0.125  1.0  1.0

LAYER 2
None
C    1    0.333333    0.166667   -0.125  1.0  1.0
```

```
C       2   -0.333333   -0.166667     0.125  1.0  1.0

LAYER 3
None
C       1   -0.333333   -0.166667    -0.125  1.0  1.0
C       2    0.333333    0.166667     0.125  1.0  1.0

LAYER 4
None
C       1    0.333333    0.166667    -0.125  1.0  1.0
C       2   -0.333333   -0.166667     0.125  1.0  1.0
```

The same data could be entered as

```
LAYER 1
CENTROSYMMETRIC
C       1   -1/3     -1/6     -1/8  1.0  1.0

LAYER 2
CENTROSYMMETRIC
C       1    1/3      1/6     -1/8  1.0  1.0

LAYER 3 = 1

LAYER 4 = 2
```

and would run significantly faster, because DIFFaX will now take advantage of the center of symmetry of the layers, and only two layer structure factors need to be computed. Here, we have exploited the fact that layers 1 and 3, and layers 2 and 4, are structurally identical, and therefore have the same layer scattering factors. With the second data set, DIFFaX knows that it needs to compute only the real components for 2 layer structure factors at each reciprocal lattice point *hkl*. With the first data set, DIFFaX will compute four complex scattering factors. For structures with many atoms per layer, such as zeolites, a significant speedup can be obtained if centers of symmetry and layer equivalences can be spelled out in the data file. However, in the author's experience, errors can easily creep into such simplifications. For example, the hypothetical centrosymmetric layer

```
LAYER 1
None
O    1   -0.2   -0.2   -0.125  1.0  1.0
Si   2    0.0    0.0    0.0     1.0  1.0
O    3    0.2    0.2    0.125  1.0  1.0
```

can be optimized to

```
LAYER 1
centrosymmetric
O    1   -0.2   -0.2   -0.125  1.0  1.0
Si   2    0.0    0.0    0.0     1.0  0.5
```

Note that the occupancy of the `Si` ' atom at the origin has been halved, since it is shared by the two centrosymmetric halves of the layer cell.

2.9 STACKING section.

2.9.1 `STACKING` header.

Line N = 13+2×n + no_atoms(layer 1) + no_atoms(layer 2) + ... + no_atoms(layer n): Header for the stacking definition data. This line must be the keyword `STACKING`. It can be written in upper or lower case. No abbreviation is accepted.

2.9.2 Layer sequencing.

Line N + 1: Describe the layer sequencing in the crystal. There are two options, `EXPLICIT` and `RECURSIVE`. `EXPLICIT` indicates that diffraction intensities are to be calculated for a crystal containing a unique, fixed sequence of layers, which will be described by the user in the lines that follow. `RECURSIVE` indicates that diffraction intensities are to be calculated for a statistical ensemble of crystallites, each with a distinct stacking sequence, but weighted by the probability that such a sequence will occur.

2.9.3 `EXPLICIT` layer sequencing.

Line N + 2: Define the layer sequence. If `EXPLICIT` has been specified in the previous line, this can be entered as an explicit sequence of layers, such as

```
1 1 2 1 2 2 2 3 2 3 2 3 2 1 1 1 3 3 3 3 3 2 1 1 2 3 2 1 2 3 2 2
2 3 3 2 3 2 1 1 1 3 3 3 3 3 2 1
```

The list can occupy more than one line, but the data should be confined to 200 characters per line, including spaces, otherwise some of the layer data will be skipped, or misread. The maximum number of explicit layers that can be defined has been set at 5000. The explicit list must not contain transitions which have zero probability (as defined in the `TRANSITIONS` section, next), as DIFFaX treats this as an error and will stop running. Alternatively, DIF-FaX can be asked to generate a sequence of M layers automatically ($M \leq 5000$), by specifying

```
RANDOM M
```

The probability that a specific layer to layer transition will occur in the sequence will be weighted by the transition probabilities listed in the `TRANSITIONS` section, described next. If you are curious about the sequence that the computer generated, this will be listed in the `DUMP` file, which can be requested at run-time (see sections 3.5 and B.3).

If the layer sequencing is specified as `RECURSIVE`, then this line contains the number of layers that the crystal is to contain. Any number larger than $M > 1022$, will be treated as

infinite [1]. Alternatively, an infinite number can be specified by the keyword `INFINITE`.

2.9.4 Examples of the `STACKING` data section.

Typical `STACKING` data sections might look like

```
{consider every possible permutation of a large number of layers}
STACKING
recursive
infinite
```

or

```
{include every possible permutation of 250 layers}
STACKING
recursive
250
```

or

```
{generate a single random sequence of 250 layers}
STACKING
explicit
random 250
```

or

```
{140 layers explicitly sequenced as follows...}
STACKING
explicit
1 1 2 4 2 2 2 3 2 3 2 3 2 1 1 1 3 3 3 3 3 2 1 1 2 2 3 3 2 3
2 3 2 1 2 3 2 2 2 3 3 2 3 2 4 1 1 3 3 3 3 3 2 1 3 3 2 3 2 1
1 1 2 1 2 4 1 2 1 2 3 3 3 3 3 4 1 2 2 1 2 1 2 3 3 2 2 2 3 3
2 3 3 4 2 4 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 3 3 2 3 2 1 1 1
1 1 1 3 2 1 3 1 3 4 4 4 2 3 2 1 1 1 3 3
```

2.10 TRANSITIONS section.

2.10.1 `TRANSITIONS` header.

Line N + 3: Header for the stacking transition data. This line must be the keyword `TRANSITIONS`, and is case-insensitive.

---

[1]There is, in fact, a good reason for this strange number, 1022. For the `EXPLICIT` option, DIFFaX must compute the $(M+1)$th power of a complex matrix $\mathbf{T}$, *i.e.* $\mathbf{T}^{M+1}$. This can be computationally expensive. In the interests of efficiency, this matrix power can be computed quickly by first representing the value of $M+1$ in binary format, and then computing and storing the values of $\mathbf{T}$, $\mathbf{T}^2$, $\mathbf{T}^4$, $\mathbf{T}^8$ etc... by successive squaring. $\mathbf{T}^{M+1}$ can then be computed by multiplying the appropriate powers of $\mathbf{T}$, the relevant powers being given by the binary flags. In DIFFaX the limit has been set to 10-bit representation, thus the largest value that M+1 can have is $2^{10} - 1 = 1023$.

2.10.2 Layer 1 to layer 1 transition data.

Line N + 4: Stacking data for layer 1 to layer 1 transitions. The data record is usually in the form

$$\texttt{alpha\_ij Rx\_ij Ry\_ij Rz\_ij}$$

where

$\texttt{alpha\_ij}$ = stacking transition probability (between 0 and 1),
$\texttt{Rx\_ij}$ = x component of stacking vector (relative to **a**),
$\texttt{Ry\_ij}$ = y component of stacking vector (relative to **b**),
$\texttt{Rz\_ij}$ = z component of stacking vector (relative to **c**)

a, **b**, and **c** are the cell constants defined on line 6 of the data file (see section 2.7.2 on page 11).

An optional set of anisotropic *Debye-Waller*-type factors, C_{11}, C_{22}, C_{33}, C_{12}, C_{13}, C_{23}, can be specified in parentheses after the first 4 numbers, *i.e.*

$$\texttt{alpha\_ij Rx\_ij Ry\_ij Rz\_ij (C11 C22 C33 C12 C13 C23)}$$

For the want of a better term, and for the author's amusement, these C_{ij} coefficients shall be referred to as the *Fats-Waller* factors in this manual. Note that the ordering of the C_{ij} terms has been modified from that used in versions of DIFFaX prior to version 1.80. Previously, the ordering of the cross-terms had been "cyclic", *i.e.* C11 C22 C33 C12 C23 C31. However, it turns out that by convention, *Debye-Waller* anisotropic temperature factor cross-terms are always listed in "anti-cyclic" order in the literature, that is, in the order B_{11}, B_{22}, B_{33}, B_{12}, B_{13}, B_{23} [1]. To comply with this convention, DIFFaX now reads the *Fats-Waller* factors "anti-cyclically".

As for the atomic *Debye-Waller* factors, the *Fats-Waller* temperature factors must be in units of Å^2. The *Fats-Waller* factors are equivalent to specifying an ellipsoidal error spread for the stacking vectors R_{ij}. These are useful for modeling systems where there is some coherence between nearest neighbor layers, but no long-range coherence, as in the case of liquid crystals or pillared clays. If these factors are not specified, this is equivalent to setting them all to zero.

If the stacking probability $\texttt{alpha\_ij}$ is zero, everything on the line after it is ignored and set to zero.

---

[1]Note that C_{31} and C_{13} are equivalent.

2.10.3 Subsequent transition data.

Lines (N + 4) to (N + n*n): Stacking data for the layer i to layer j transitions. The data is expected in the order

```
layer 1 to layer 1
layer 1 to layer 2
.
.
.
layer 1 to layer n

layer 2 to layer 1
layer 2 to layer 2
.
.
.
layer 2 to layer n
.
.
.
layer n to layer 1
layer n to layer 2
.
.
.
layer n to layer n
```

The sum of the probabilities starting from each layer must all sum to 1.0, *i.e.* $\sum_j \alpha_{ij} = 1.0$ for each i, otherwise DIFFaX will assume an error has been made, and will quit.

2.10.4 Examples of the stacking `TRANSITIONS` data section.

Typical stacking `TRANSITIONS` data section might look like

```
TRANSITIONS                       {Header for transitions}
{Transitions from layer 1}
0.7    0.66667    0.333333  1.0    {layer 1 to layer 1, 70% chance}
0.3    0.0        0.0       1.0    {layer 1 to layer 2, 30% chance}

{Transitions from layer 2}
0.3    0.0        0.0       1.0    {layer 2 to layer 1, 30% chance}
0.7   -0.66667   -0.333333  1.0    {layer 2 to layer 2, 70% chance}
```

Stacking vectors can be entered as fractions, thus

```
TRANSITIONS                          {Header for transitions}
{Transitions from layer 1}
0.7    2/3          1/3        1.0    {layer 1 to layer 1, 70% chance}
0.3    0.0          0.0        1.0    {layer 1 to layer 2, 30% chance}


{Transitions from layer 2}
0.3    0.0          0.0        1.0    {layer 2 to layer 1, 30% chance}
0.7   -2/3         -1/3        1.0    {layer 2 to layer 2, 70% chance}
```

Data with *Fats-Waller* factors might look like

```
TRANSITIONS                 {Header for transitions}
{Transitions from layer 1} {add a spread in R_z}
```

```
0.1   0.5   0.5   1.0    (0.0 0.0 1.0 0.0 0.0 0.0)
0.9   0.0   0.0   1.0    (0.0 0.0 1.0 0.0 0.0 0.0)

{Transitions from layer 2}
0.9   0.0   0.0   1.0    (0.0 0.0 1.0 0.0 0.0 0.0)
0.1   0.5   0.5   1.0    (0.0 0.0 1.0 0.0 0.0 0.0)
```

or, more compactly

```
0.1 0.5 0.5 1.0 (0.0 0.0 1.0 0.0 0.0 0.0)
0.9 0.0 0.0 1.0 (0.0 0.0 1.0 0.0 0.0 0.0)
0.9 0.0 0.0 1.0 (0.0 0.0 1.0 0.0 0.0 0.0)
0.1 0.5 0.5 1.0 (0.0 0.0 1.0 0.0 0.0 0.0)
```

2.11   Dependence of diffraction symmetry on the stacking probabilities.

It is perfectly legitimate to set a layer transition probability to be equal to 1.0, and all the other transitions from that layer to be 0.0. This will define a perfect unfaulted structure, and DIFFaX is capable of handling this limit. However, at this limit, the probability that the other layers will actually exist in the crystal may become zero or undefined. This sudden disappearance of layers from the ensemble may modify the apparent Laue symmetry of the diffraction data relative to the faulted intermediates. This can be confusing if a series of diffraction data for transition probabilities ranging from 0.0 to 1.0 are calculated, because the end-members may appear to have different symmetry to the intermediate members. This is most apparent in the selected area diffraction patterns.

The fraction of layers of type i that actually occur in a crystal, g_i, is given by the set of simultaneous equations

$$g_i = \sum_{j=1}^{N} g_j \alpha_{ji} \qquad \sum_{i=1}^{N} g_i = 1 \tag{6}$$

Therefore, in a two-layer crystal system where the transition probabilities are defined to be

$$\alpha_{11} = 0.9999, \quad \alpha_{12} = 0.0001$$
$$\alpha_{21} = 0.0001, \quad \alpha_{22} = 0.9999$$

the frequency of occurrence of layers 1 and 2 in the crystal are $g_1 = g_2 = 0.5$ (*i.e.* 50% of the crystal is layer 1, 50% is layer 2). The symmetry of the diffraction pattern will reflect the fact that, statistically, both layer types and their associated stacking sequences, are equally represented in the typical crystal. However, if the transition probabilities had been set up as

$$\alpha_{11} = 1.0, \quad \alpha_{12} = 0.0$$
$$\alpha_{21} = 0.0, \quad \alpha_{22} = 1.0,$$

which at first glance would appear to be essentially indistinguishable from the previous example, we find that there is no unique solution to the simultaneous equations 6. DIFFaX resolves the ambiguity in this instance by counting the number of α_{ii} terms, N, that equal 1.0, and assigning $g_i = 1/N$ when $\alpha_{ii} = 1.0$, and $g_j = 0.0$ when $\alpha_{jj} \neq 1.0$. This tactic, however, is a compromise solution. There is no *a priori* justification for this $1/N$ assumption, since, in principle, the α_{ii} can all approach the value 1.0 at different asymptotic rates, *i.e.*

$$\alpha_{11} = 1 - \epsilon, \quad \alpha_{12} = \epsilon$$
$$\alpha_{21} = 2\epsilon, \quad \alpha_{22} = 1 - 2\epsilon,$$

as $\epsilon \to 0$. This would then give $g_1 = 2/3, g_2 = 1/3$, and our compromise $1/N$ solution would be inappropriate for this case.

In order to avoid such discontinuous (and sometimes surprising) changes of diffraction symmetry at these limits, it is prudent not to exceed about 0.9999 for transition probability values when computing patterns for a range of transition probabilities. Thus, instead of the series of values $\alpha_{11} = 0.0, 0.1, 0.2 \ldots 0.9, 1.0$, use $\alpha_{11} = 0.0001, 0.1, 0.2 \ldots 0.9, 0.9999$. The transition probability sum, $\sum_j \alpha_{ij}$, must still come to 1.0. When computing powder patterns, errors caused by an unexpected symmetry change can be circumvented by always specifying the symmetry as `UNKNOWN` in the data file (see section 2.7.3). DIFFaX will then catch any changes in symmetry, and modify the integration range in reciprocal space accordingly. However, this will be of no help when computing selected area diffraction patterns, where the change in symmetry is visually obvious from the distribution of diffraction spots. In this case, non-zero probabilities should be used.

2.12   A more complex example. Fault clustering.

Fault clustering can be simulated by increasing the number of layer types. To describe clustering in diamond, we need at least two versions of layer type 1 and at least two versions of layer type 2, in order to differentiate between stacking histories. Thus, a layer type 1, when preceded by a layer type 1, might connect to subsequent layers in a different manner to a layer type 1 that is preceded by a layer type 2. This memory, or "Reichweite", can be simulated by differentiating between such layers, even though they may be structurally identical. A data file describing clustered twin faults in diamond might look like (see appendix A.2 for more details).

```
{data file for clustered faults in diamond}
{Probability of a fault in diamond cubic stacking = 10%}
{Probability of a fault in lonsdaleite hexagonal stacking = 20%}

{There are four layers in total, two of which are}
{structurally distinct.  They are related by a mirror}
```

```
{in the a-b plane, and are labeled layers A and A'.}
{Layer 1 is an A layer which is preceded by an A layer}
{Layer 2 is an A' layer which is preceded by an A layer}
{Layer 3 is an A layer which is preceded by an A' layer}
{Layer 4 is an A' layer which is preceded by an A' layer}
{1 - 1 is a cubic stacking transition}
{1 - 2 is a stacking fault in a cubic sequence}
{1 - 3 cannot occur}
{1 - 4 cannot occur}
{2 - 1 cannot occur}
{2 - 2 cannot occur}
{2 - 3 is a hexagonal stacking transition}
{2 - 4 is a fault in a hexagonal sequence}
{3 - 1 is a fault in a hexagonal sequence}
{3 - 2 is a hexagonal stacking transition}
{3 - 3 cannot occur}
{3 - 4 cannot occur}
{4 - 1 cannot occur}
{4 - 2 cannot occur}
{4 - 3 is a stacking fault in a cubic sequence}
{4 - 4 is a cubic stacking transition}

INSTRUMENTAL
X-RAY
1.5418
PSEUDO-VOIGT .89 -.32 .08 .6 TRIM

STRUCTURAL
2.52 2.52 2.06 120.0
unknown 1 {We expect -3m.  Let DIFFaX decide to within a tolerance of 1%.}
4


LAYER 1
CENTROSYMMETRIC
C      1      -1/3      -1/6      -1/8      1.0      1.0


LAYER 2
CENTROSYMMETRIC
C      1      1/3      1/6      -1/8      1.0      1.0

LAYER 3 = 1 {layer 3 is structurally identical to layer 1}

LAYER 4 = 2 {layer 4 is structurally identical to layer 2}

STACKING
recursive
infinite


TRANSITIONS
{1}
0.90      2/3      1/3      1.0
0.10      0.0      0.0      1.0
0.00      0.0      0.0      0.0 {1-3 Transition cannot occur}
0.00      0.0      0.0      0.0 {1-4 Transition cannot occur}


{2}
0.00      0.0      0.0      0.0 {2-1 Transition cannot occur}
0.00      0.0      0.0      0.0 {2-2 Transition cannot occur}
0.80      0.0      0.0      1.0
```

```
0.20    - 2/3     - 1/3     1.0

{3}
0.20     2/3      1/3      1.0
0.80     0.0      0.0      1.0
0.00     0.0      0.0      0.0 {3-3 Transition cannot occur}
0.00     0.0      0.0      0.0 {3-4 Transition cannot occur}

{4}
0.00     0.0      0.0      0.0 {4-1 Transition cannot occur}
0.00     0.0      0.0      0.0 {4-2 Transition cannot occur}
0.10     0.0      0.0      1.0
0.90    - 2/3     - 1/3     1.0
```

3   RUNNING DIFFaX.

Over the course of its development, DIFFaX has been tested on several different computer systems and with many different fortran compilers. DIFFaX has been written so that its operation is, as far as is possible, independent of the computer's operating system. There are two ways to run DIFFaX. (i) It can be run interactively, whereby the user is expected to respond to certain prompts via the keyboard, or (ii) it can be run using a control file, which is a text file containing pre-composed answers to DIFFaX's prompts (see section 5 on page 37 for a fuller description of control files). In order to run correctly, DIFFaX expects to find the standard scattering factor data file `data.sfc`, and (if used) the control file `control.dif`, in the same directory as the executable DIFFaX program. If the user wishes to run DIFFaX interactively, the control file `control.dif` must be renamed or removed, otherwise DIFFaX will read it by default.

3.1   Invoking DIFFaX.

Once the data file has been written, DIFFaX is ready to run. The DIFFaX program can usually be invoked by one of these commands

VAX VMS: `run DIFFaX`
unix: `DIFFaX`
Macintosh: "double-click" the application icon "MacDIFFaX_881" (or "MacDIFFaX_PPC").

3.2 DIFFaX start-up screen.

If all is well, the user will be greeted by a start-up message resembling

```
****************************************************
****************************************************
*                                                  *
*  DDDD     II   FFFFFF   FFFFFF            X   X   *
*  D   D    II   F        F          aaaa    X X    *
*  D    D   II   FFFF     FFFF     a    a     X     *
*  D   D    II   F        F        a   aa    X X    *
*  DDDD     II   F        F         aaa a   X   X   *
*                                                  *
****************************************************
***************** DIFFaX v1.801 *******************
****************************************************
************** 14th JULY, 1995 ****************
****************************************************
*                                                  *
*    A computer program for calculating            *
*    Diffraction Intensity From Faulted Crystals   *
*                                                  *
*    Authors: Michael M. J. Treacy                 *
*             Michael W. Deem                      *
*                                                  *
****************************************************

 Enter name of structure file.
```

3.3 Data file name.

The user should respond by typing in the name of the file containing the structure data, and typing a carriage return. If all is well, then, after a few moments, the following lines will appear

```
 Enter name of structure file.
dia.dat
 Looking for structure data file 'dia.dat'
 Opening structure file 'dia.dat'
 Looking for scattering factor data file 'data.sfc'
 Opening scattering factor data file 'data.sfc'
 Enter 1 for DUMP
```

Note, the user's responses are left-indented.

3.4 Data file name errors.

It is common to get errors at this early stage. If the user mistypes the filename (*i.e.* `dia.dot` instead of `dia.dat`), or the file was saved in a directory other than the one that contains DIFFaX, then the user will get an error message that, depending on the operating system, may look like

```
 Enter name of structure file.
dia.dot
```

```
 Looking for structure data file 'dia.dot'
 Problems opening structure data file 'dia.dot'
 IOSTAT =     2
 DIFFaX was terminated abnormally.
```

If the standard scattering factor data file, `data.sfc` is stored in a directory other than the one that contains DIFFaX, then the user will get an error message that, depending on the operating system, may look like

```
 Enter name of structure file.
dia.dat
 Looking for structure data file 'dia.dat'
 Opening structure file 'dia.dat'
 Looking for scattering factor data file 'data.sfc'
 Problems opening scattering factor file 'data.sfc'
 IOSTAT =     2
 DIFFaX was terminated abnormally.
```

Make sure that the full path to the data file is correctly specified (*i.e.* under unix a file path such as `../diamond/dia.dat` could be specified), and that the standard scattering factor data file `data.sfc` is in the same directory as DIFFaX.

3.5 Generating a dump file.

If there were no problems opening the data files, the next step is to decide whether or not to generate a dump of the data file, as DIFFaX interpreted it. If this is the first time that this file is being run, it is a good idea to create a dump file, just to make sure there are no surprises. Responding with a '1' will generate a dump. Anything else will not generate a dump. DIFFaX then reads the data file and `data.sfc`. If there were no errors encountered, the screen will appear as follows.

```
 Enter 1 for DUMP
1
 Reading 'dia.dat'
 'dia.dat' read in.
 Reading scattering factor datafile 'data.sfc'. . .
 Scattering factor data read in.
 Enter 1 for full atomic position dump
```

Whether or not a dump was requested, DIFFaX carries out a set of checks on the data file, to ensure that the values entered are reasonable, or physically possible. At this stage, if the user opted for a dump, a choice is presented whether or not to list all the atomic coordinate data with a full listing of their scattering factor parameters, in the dump file. Responding with a '1' will give a full listing. Anything else will give a shortened listing.

```
 Enter 1 for full atomic position dump
1
 Writing data dump to file 'dia.dmp'. . .
 Enter 1 for a dump of the symmetry evaluations
```

27

Note that the dump file has been named `dia.dmp`. See section 6 for a description of file naming conventions.

3.6 Generating a dump of symmetry evaluations.

If there were no problems generating the data dump file, the user will now be asked whether or not to create a dump of DIFFaX's evaluation of the Laue group symmetry of the diffraction intensities in reciprocal space. This facility is useful mainly for tracking down some of the more mysterious errors in the layer atomic data, when the diffraction symmetry is different to that expected. Responding with a '1' will give a listing of the symmetry evaluations. Anything else will not generate a symmetry dump.

```
 Enter 1 for a dump of the symmetry evaluations
1
 Checking for conflicts in atom positions . . .
 Evaluating point group symmetry of diffraction data. . .
 Writing symmetry data to file 'dia.sym'. . .
 The diffraction data fits the point group symmetry '6/MMM'
   with a tolerance of one part in  12376
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
```

Note that the symmetry evaluation data file is called `dia.sym`. See section 6 for a description of file naming conventions.

3.7 The main menu.

If there were no problems generating the symmetry evaluations file, the user is now provided with five options for the type of diffraction intensity information required.

3.7.1 Intensity at a reciprocal lattice point, hkl.

Menu choice '0' allows the user to calculate the intensity at a reciprocal lattice point hkl, and will also give information about the scattering angle in degrees, the structure factors of each layer, and the ensemble average interference wavefunctions when each of the various layers is at the origin of the crystal. This menu allows the user to repeat the operation for as many points as desired. The user may return to the main menu when done. An example of a typical run is listed below

```
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
0

 CALCULATING INTENSITY AT A POINT. . .
 Enter h, k, l
1 0 0
 2theta =   41.371      degrees
 d =    2.1824
```

```
 1/d =  0.45821

 Layer scattering factors
 f( 1) = ( -3.022460    , 0.0000000    )
 f( 2) = ( -3.022460    , 0.0000000    )

 Average scattered wavefunctions
 psi( 1) = ( -2.375548    , 0.8722551    )
 psi( 2) = ( -2.375548    ,-0.8722551    )

 Intensity at   1  0 0.00000    =   4.083575
 Another point? (y/n)
y

 CALCULATING INTENSITY AT A POINT. . .
 Enter h, k, l
3 0 0
    3  0 0.00000     exceeds 180 degree scattering angle!
 Re-enter. . .
 Enter h, k, l
2 0 0
 2theta =   89.898     degrees
 d =    1.0912
 1/d =  0.91643

 Layer scattering factors
 f( 1) = ( -1.438112    , 0.0000000    )
 f( 2) = ( -1.438112    , 0.0000000    )

 Average scattered wavefunctions
 psi( 1) = ( -1.130309    ,-0.4150323    )
 psi( 2) = ( -1.130309    , 0.4150323    )

 Intensity at   2  0 0.00000    =   0.5914296
 Another point? (y/n)
n

 Enter 1 to return to function menu.
1
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
```

3.7.2 Intensity trace along a reciprocal lattice streak, from hkl_0 to hkl_1.

Menu choice '1' will allow the user to calculate the intensity profile as a function of l for given values of h and k. The results are saved to file. The user will be prompted to enter values for h and k, the beginning, l_0, and ending, l_1, values of l, and the stepsize Δl along l. Naturally, the values should be such that $l_1 > l_0$, and $\Delta l < l_1 - l_0$. The user will also be asked whether or not to use adaptive quadrature to perform the integration. If the interval $l_1 - l_0$ is large, adaptive quadrature will be more accurate, although about 3 times slower than non-adaptive. Depending on the computer, and the number of steps needed to span the interval (equal to $(l_1 - l_0)/\Delta l$), this calculation may take a few minutes. DIFFaX will print out progress reports, announcing what value of l has been reached.

```
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
1
```

```
 CALCULATING INTENSITY ALONG A STREAK. . .
 Enter 1 for adaptive quadrature
1
 Enter h, k, l0, l1, delta l
1 0 0 1 .01
 Writing streak data to file 'dia.str'. . .
 Reached l =    0.04000
 Reached l =    0.09000
 Reached l =    0.14000
 Reached l =    0.19000
 Reached l =    0.24000
 Reached l =    0.29000
 Reached l =    0.34000
 Reached l =    0.39000
 Reached l =    0.44000
 Reached l =    0.49000
 Reached l =    0.54000
 Reached l =    0.59000
 Reached l =    0.64000
 Reached l =    0.69000
 Reached l =    0.74000
 Reached l =    0.79000
 Reached l =    0.84000
 Reached l =    0.89000
 Reached l =    0.94000
 Reached l =    0.99000
 Streak data file, 'dia.str' written to disk.
 Enter 1 to return to function menu.
1
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
```

Note that the streak data file is called `dia.str`. See section 6 for a description of file naming conventions. The file `dia.str` is written out in tab-delimited text format. Each line contains the following information

'l-value' tab 'intensity' carriage return

3.7.3   Integrating intensity along a streak, within an interval hkl_0 to hkl_1.

Menu choice '2' will allow the user to calculate the integrated intensity between two l values for given values of h and k. The user will be prompted to enter values for h and k, and the beginning and ending values of l. The user will also be asked whether or not to use adaptive quadrature to perform the integration. If the interval $l_1 - l_0$ is large, adaptive quadrature will be more accurate, although about 3 times slower than non-adaptive. Typical output will look like the following.

```
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
2

 INTEGRATING INTENSITY IN AN INTERVAL ALONG l. . .
 Enter 1 for adaptive quadrature.
1
```

```
 Enter h, k, l0, l1
0 0 .95 1.05
 Integrated intensity =    12.354973
 Enter 1 to integrate another interval.
1

 INTEGRATING INTENSITY IN AN INTERVAL ALONG l. . .
 Enter 1 for adaptive quadrature.
0
 Enter h, k, l0, l1
0 0 .95 1.05
 Integrated intensity =    12.354973
 Enter 1 to integrate another interval.
1

 INTEGRATING INTENSITY IN AN INTERVAL ALONG l. . .
 Enter 1 for adaptive quadrature.
1
 Enter h, k, l0, l1
0 0 .999 1.001
 Integrated intensity =    11.134940
 Enter 1 to integrate another interval.
0
 Enter 1 to return to function menu.
1
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
```

If the interval $l_1 - l_0$ is small, as in the above example, adaptive and non-adaptive quadrature usually give similar results. However, full adaptive integration takes significantly more time. In the above example, the reciprocal lattice point 001 is a sharp spot, thus intensity varies rapidly in the vicinity of this spot. Although the adaptive algorithm senses the rapid variation and adjusts its integration stepsize accordingly, the result is not necessarily more accurate: the non-adaptive algorithm is robust enough for this example. However, if $l_1 - l_0$ is too large, the answers may diverge. The cautious user should explore the interaction between the stepsize $l_1 - l_0$ and the need for full adaptive integration.

3.7.4   Selected area diffraction pattern.

Menu choice '4' will allow selected area diffraction patterns to be calculated. This option is most useful for calculating kinematical electron diffraction patterns.

```
Enter function number:
0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
4

CALCULATING SELECTED AREA DIFFRACTION PATTERN. . .
Enter 1 for adaptive quadrature.
1
Choose a plane in reciprocal space to view.
      the l-axis is included by default.
1: k = 0.   2: h = 0.   3: h = k.   4: h = -k.
2
Enter maximum value of l.
2
Choose the bit-depth of the image.
```

31

```
    8: - 8-bits   16: - 16-bits
16
File will be saved in unsigned 16-bit format.
Choose intensity scaling
    0: - Logarithmic   1: - Linear
1
Enter a brightness (must be +ve)
  1 - scales intensities to the range 0 - 32767
 10 - scales intensities to the range 0 - 327670,
      but values above 65535 will be saturated
5
h =   0 k =   0            'dia.dat'
l =   -0.00781 intensity =  0.00000
l =    0.08594 intensity =  0.13091E-01
l =    0.17969 intensity =  0.32829E-02
l =    0.27344 intensity =  0.14872E-02
l =    0.36719 intensity =  0.87177E-03
l =    0.46094 intensity =  0.60308E-03
l =    0.55469 intensity =  0.48016E-03
l =    0.64844 intensity =  0.44465E-03
l =    0.74219 intensity =  0.50383E-03
l =    0.83594 intensity =  0.80688E-03
l =    0.92969 intensity =  0.33175E-02
l =    1.02344 intensity =  0.94814E-02
l =    1.11719 intensity =  0.39026E-03
l =    1.21094 intensity =  0.92740E-04
l =    1.30469 intensity =  0.34490E-04
l =    1.39844 intensity =  0.16146E-04
l =    1.49219 intensity =  0.88008E-05
l =    1.58594 intensity =  0.53895E-05
l =    1.67969 intensity =  0.36399E-05
l =    1.77344 intensity =  0.26848E-05
l =    1.86719 intensity =  0.21521E-05
l =    1.96094 intensity =  0.18708E-05
h =   0 k =   1            'dia.dat'
l =   -0.00781 intensity =  0.63827E-01
l =    0.08594 intensity =  0.73861E-01
l =    0.17969 intensity =  0.11971
l =    0.27344 intensity =  0.29790
l =    0.36719 intensity =  0.35620
l =    0.46094 intensity =  0.92463E-01
l =    0.55469 intensity =  0.27839E-01
l =    0.64844 intensity =  0.61255E-03
l =    0.74219 intensity =  0.65866E-02
l =    0.83594 intensity =  0.85806E-02
l =    0.92969 intensity =  0.96500E-02
l =    1.02344 intensity =  0.12521E-01
l =    1.11719 intensity =  0.20760E-01
l =    1.21094 intensity =  0.50273E-01
l =    1.30469 intensity =  0.18049
l =    1.39844 intensity =  0.15803
l =    1.49219 intensity =  0.82232E-01
l =    1.58594 intensity =  0.86818E-01
l =    1.67969 intensity =  0.52458E-01
l =    1.77344 intensity =  0.68915E-02
l =    1.86719 intensity =  0.99358E-03
l =    1.96094 intensity =  0.69756E-04
h =   0 k =   2            'dia.dat'
l =   -0.00781 intensity =  0.92443E-02
l =    0.08594 intensity =  0.10797E-01
l =    0.17969 intensity =  0.17989E-01
l =    0.27344 intensity =  0.46839E-01
l =    0.36719 intensity =  0.59557E-01
l =    0.46094 intensity =  0.16694E-01
```

```
l =      0.55469 intensity =   0.55018E-02
l =      0.64844 intensity =   0.13381E-03
l =      0.74219 intensity =   0.16145E-02
l =      0.83594 intensity =   0.23738E-02
l =      0.92969 intensity =   0.30345E-02
l =      1.02344 intensity =   0.44995E-02
l =      1.11719 intensity =   0.85570E-02
l =      1.21094 intensity =   0.23813E-01
l =      1.30469 intensity =   0.98215E-01
l =      1.39844 intensity =   0.98521E-01
l =      1.49219 intensity =   0.58444E-01
l =      1.58594 intensity =   0.69768E-01
l =      1.67969 intensity =   0.47128E-01
l =      1.77344 intensity =   0.68275E-02
l =      1.86719 intensity =   0.10684E-02
l =      1.96094 intensity =   0.00000
Writing SADP data to file 'dia.sadp7'. . .
 256 x  256 pixels: 16 bits deep.
Enter 1 to return to function menu.

1
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
```
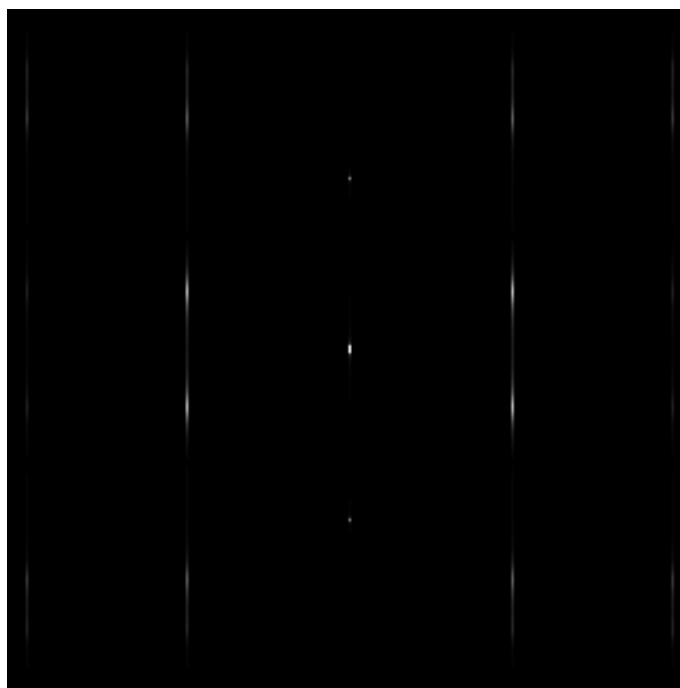

Figure 1: Selected area electron diffraction pattern simulation for diamond/lonsdaleite, showing the $0kl$ spots for a diamond-lonsdaleite stacking fault probability of 0.3. (See data file in section A.1).

The user is first asked to choose a plane in reciprocal space to view. Choosing '1' views the plane containing $h0l$ spots; '2' the plane containing $0kl$ spots; '3' the plane containing hhl spots and '4' the plane containing $h\overline{h}l$ spots. Then the user is asked to enter the maximum value of l to go out to. The maximum values of h and k will be automatically defined by this l value, since the calculated diffraction pattern will be square. The user specifies whether an

33

8 bit or a 16 bit pixel depth is required. The user is then asked whether or not to save the intensity data on a linear or logarithmic scale, by entering '0' or '1' respectively. In linear mode, an intensity scaling factor can be specified. A factor of 1 means that the intensity range (excluding the zero order beam) will be compressed so as to fit within the range 0 to 255 (8 bit binary) or 0 to 32767 (signed 16 bit binary)[1]. A factor of 10 means that the intensity range will be fitted within the range 0 to 2550 (or 0 to 327670), but all values above 255 (32767) will be saturated, and will be set equal to 255 (32767). Since the calculation may take a little while, DIFFaX prints out a progress report, listing the current value of hkl it has reached. Note that the selected area diffraction pattern data file is saved as `dia.sadp`. See section 6 for a description of file naming conventions. The file `dia.sadp` is saved as an unformatted binary file. Each pixel is saved either as an 8 bit number between 0 and 255 or a 16 bit number between 0 and 32767. The image size is 256×256. To display the file the user needs a graphics utility program that can open such binary files. Figure 1 shows the selected area diffraction pattern, `dia.sadp`, as printed from the Macintosh program $Image$[2].

3.7.5 Powder pattern.

Menu choice '3' will allow the user to calculate the powder pattern within a range of 2θ values. The user will be prompted for the beginning and ending 2θ values, and the 2θ stepsize. Then the user will be asked whether or not to use full adaptive quadrature on the broad peaks. When the fault probabilities are high (*i.e.* close to 50%) it is usually safe to ignore the full adaptive quadrature option. However, there is always the risk of errors near any sharper intensity features that may not have been anticipated. Next the user is asked whether or not to use full adaptive quadrature over all l values on the hk rows known to contain sharp peaks. DIFFaX attempts to establish which values of hk are rows of sharp spots. These will always be integrated full-adaptively. If the reply is '1' to this question, DIFFaX will employ full-adaptive integration over all l values, even though most of the intensity is contained near the sharp spots, with relatively nothing in between. Avoiding full adaptive quadrature speeds up the calculation a little. However, there is the risk of integration errors, which may result in some sharp peaks being missed altogether. It is safest always to select full-adaptive quadrature for both these options. Where computation speed is an important criterion and a large series of calculations are to be carried out, it would be prudent to compare the

---

[1]DIFFaX stores the 16-bit data as two sequential bytes. The Fortran code will have to be modified if reverse byte ordering is required.

[2]*Image* is an image-processing program for the Macintosh (requires 8-bit video), originally written by Wayne Rasband, National Institutes of Health, Bethesda, MD. The *Image* program, including Macintosh THINK pascal source code and documentation, is in the public domain and may be freely copied and distributed. *Image* can be obtained via anonymous ftp from `zippy.nimh.nih.gov[128.231.98.32]`. Enter "anonymous" as the user name, and your email address as the password. The `/pub/nih-image` directory contains the latest version of *Image*.

results using the adaptive and non-adaptive options, and explore how they interact with the 2θ stepsize. Making the stepsize too small will also slow up the calculation. For most calculations, it is a matter of trial and error to establish the best combination of parameters. Because this is a time consuming calculation, DIFFaX will periodically print out a progress report, announcing the values of h and k reached. When finished, DIFFaX will inspect the intensities calculated, particularly those near the origin, where the intensities can swamp the values at other angles. These will be suppressed if the TRIM option was included in the instrumental broadening input of the structure data file. If the data file has requested it, an instrumental broadening function will be applied to the trimmed data. The 2θ values, the raw data, and the broadened data are then written to a '.spc' file. If running interactively, DIFFaX will then end. Example screen output is listed below.

```
 Enter function number:
 0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
3

 CALCULATING POWDER DIFFRACTION PATTERN. . .
 Enter angular range:
   2theta min, 2theta max, 2theta increment.
0 170 0.05
 Enter 1 for adaptive quadrature on broad peaks.
1
 Enter 1 for adaptive quadrature over all l values
 on rows with "sharp" spots
1
 Integrating along l at    0   0      'dia.dat'
 Full adaptive integration
 Integrating along l at    1   0      'dia.dat'
 Full adaptive integration
 Integrating along l at    1   1      'dia.dat'
 Full adaptive integration
 Integrating along l at    2   0      'dia.dat'
 Full adaptive integration
 Integrating along l at    2   1      'dia.dat'
 Full adaptive integration
 Adding pseudo-Voigt instrumental broadening . . .
 Writing spectrum data to file 'dia.spc'
 Spectrum written.

 DIFFaX ended normally.
```

The file dia.spc is written out in tab-delimited text format. Each line contains the following information

'2θ degrees' tab 'unbroadened intensity' tab 'broadened intensity' carriage return

The broadened intensity is not written if the data file specified NONE in the line for instrumental broadening type.

4   Speeding up DIFFaX computations.

Despite the fact that the DIFFaX code is replete with tricks that attempt to speed up the computation, DIFFaX runs can take a considerable time to finish, particularly for powder patterns, streaks and selected area diffraction patterns. The most computationally expensive part of the DIFFaX code is the intensity integration. For a powder pattern computation DIFFaX cannot get away with merely sampling the intensity at integral multiples of h, k and l, as can be done for perfectly periodic crystals. DIFFaX needs to integrate along the reciprocal lattice vector l in variable steps of Δl such that $\Delta\theta = \theta(h, k, l + \Delta l) - \theta(h, k, l)$ is constant. Depending on the value of $\Delta\theta$ chosen by the user, Δl can be quite large relative to the width of a typical sharp peak, particularly when l is small or zero. To detect reliably such sharp peaks, DIFFaX employs an adaptive 16-point Gauss-Legendre integration algorithm. In Gauss-Legendre integration, the intensity in the interval Δl is sampled at 16 points. These points are not uniformly spaced, but are carefully chosen so as to sample the intensity optimally. The 16 intensities are added, after being multiplied by equally carefully chosen weights. If the intensity is varying smoothly, the integration can be remarkably accurate. In adaptive integration, the region Δl is further subdivided into two regions, each of length $\Delta l/2$, and the 16-point integration is re-applied to each region, and the results added. If the first 16-point result, and the subsequent 32-point result give identical answers within some error limit, DIFFaX moves on to the next region. If the answers do not agree, usually because the intensity varies rapidly within the region Δl, the regions are repeatedly subdivided and re-integrated until a self-consistent answer is obtained. Adaptive integration, therefore, takes at least three times as long to perform as non-adaptive integration (at least 48 intensity samplings versus 16 intensity samplings).

The user can control whether or not DIFFaX uses adaptive integration. The user is prompted twice

```
 Enter 1 for adaptive quadrature on broad peaks.
```

and

```
Enter 1 for adaptive quadrature over all l values
 on rows with ``sharp'' spots
```

Entering a "0" in response to these prompts disables adaptive integration.

DIFFaX attempts to make intelligent guesses as to when it is encountering a row of sharp spots, so generally it is safe to answer "0" to the latter prompt. DIFFaX will always perform adaptive integration on the sharp peak itself, but will ignore the vast expanses of

low intensity that lie between them. However, there is a danger to switching off adaptive integration on broad peaks. In those instances when fault probabilities are low, peaks that are normally susceptible to broadening, can in fact be quite sharp. It generally is safe to switch off adaptive integration for broad peaks, but the user should be aware that there are risks at the extreme values of stacking probability.

DIFFaX can store spectra with up to 20,000 points. The number of points, N, is given by $N = (\theta_{\max} - \theta_{min})/\Delta\theta$. The time-conscious user should refrain from overzealously specifying unnecessarily small values for $\Delta\theta$. Clearly $\Delta\theta$ should be somewhat smaller than the instrumental peak broadening half-width that is specified in the data file. However, there is usually little to be gained by making $\Delta\theta$ significantly less than about one-tenth the instrumental peak broadening half-width. In fact, a ratio of about one-third or one-fourth will usually be adequate for all but the most demanding graphics. Smaller values of $\Delta\theta$ will increase the number of regions to be integrated and will proportionately slow down the computation.

Another rate-limiting factor is the number of layers, n. The matrix inversion takes $O(n^3)$ to compute. Further, the computation time is linear in the number of atoms in a layer.

If a layer is centrosymmetric, specifying `CENTROSYMMETRIC` in the data file, and listing only the asymmetric half of the atoms (and being careful to halve the occupancies of those atoms that are shared by the two asymmetric halves) will speed up the computation considerably. This is not just because there are fewer atoms, but also because the imaginary components of the wave functions can be ignored. If all layers are centrosymmetric, this trick can more than halve the computation time. DIFFaX does not accept other symmetry descriptions for layer unit cells, such as rotational or mirror symmetries, because there is no significant speed benefit.

When two layers are structurally identical, but merely stack differently, making use of the layer equivalence description (see section 2.8.4 on page 16),

```
LAYER 2 = 1
```

can save a little time.

Finally, be warned that the `PSEUDO-VOIGT` peak-broadening option is computationally expensive (section 2.6.8, page 9). Unless it is vitally necessary, use the fixed-width `GAUSSIAN` (section 2.6.6, page 7) or `LORENTZIAN` (section 2.6.7, page 8) options, as these are substantially faster.

5   DIFFaX control files.

In normal use, DIFFaX will be used to run batches of data files, usually with differing stacking probabilities. After the run, the powder diffraction simulations are compared with

data for the best fit. A typical run might require 11 files for stacking probabilities ranging from 0.001 to 0.999 inclusive (see the discussion in section 2.11 on page 22 for an explanation of why I deliberately avoided specifying the range 0.0 to 1.0). Such a computation can take DIFFaX a considerable time, which makes the interactive mode an unsuitable way to execute such a run.

DIFFaX provides a way to run batch jobs by storing pre-composed answers to the DIFFaX prompts in a control file. The control file default name is "`control.dif`", and should be stored in the same directory as the DIFFaX executable code. A simple way to create a control file is to try a dummy run of DIFFaX, and to record the responses to the prompts. Terminate the DIFFaX run, and create the `control.dif` file with the responses listed one per line. For example, a control file for a typical powder pattern run would look like

```
dia001.dat          {data file name}
0                   {no dump file}
0                   {no symmetry file}
3                   {powder pattern}
0 100 .05           {theta_min, theta_max, delta_theta}
1                   {do adaptive quadrature on broad peaks}
1                   {do adaptive quadrature on all l for rows of sharp spots}
dia100.dat
0
0
3
0 100 .05
1
1
dia200.dat
0
0
3
0 100 .05
1
1
dia300.dat
0
0
3
0 100 .05
1
1
dia400.dat
0
0
3
0 100 .05
1
1
dia500.dat
0
0
3
0 100 .05
1
1
dia600.dat
0
0
```

```
3
0 100 .05
1
1
dia700.dat
0
0
3
0 100 .05
1
1
dia800.dat
0
0
3
0 100 .05
1
1
dia900.dat
0
0
3
0 100 .05
1
1
dia999.dat
0
0
3
0 100 .05
1
1
end                     {no more files, end normally}
```

Note that comments {in curly braces} can be embedded in the control file. At the end of each run on a data file, DIFFaX reads the next line for a new data file name. If it finds a legal file name, it will re-run using the new data. A line containing the keyword "end", tells DIFFaX that the control file is ended. It is not fatal to leave off the keyword end, however, erroneous, and possibly confusing, error messages will be generated.

The control file can also be used to compute the POINT, STREAK, INTEGRATE and SADP options. Care should be taken to compose responses in the correct order, usually by trying a dummy run of DIFFaX and recording the responses.

An example control file, with output, is given in appendix B on page 58.

6 File naming conventions used in DIFFaX.

DIFFaX tries to name output files so that it is reasonably clear from the name what the file contains, and with which data set it is associated. If the structure data file was called name, or name.dat then the output files will be named as follows;

name.dmp data dump file

`name.sym` data symmetry evaluation file

`name.spc` powder diffraction spectrum

`name.str` intensity profile along a streak

`name.sadp` selected area diffraction pattern

Note that DIFFaX reads the filename from the left up to the first period ('.') (or space (' ') on the Macintosh), and uses that as the root filename. Thus the powder diffraction pattern file for the files `dia.dat` or `dia`, would be named equally `dia.spc`. If a file of that name already exists, then DIFFaX will enumerate the filenames sequentially. Thus, if the file `dia.spc` already exists, then it will name the next file `dia.spc1`. If, for example, files `dia.spc`, `dia.spc1` to `dia.spc13`, already exist, then the next file will be `dia.spc14`. However, if the file `dia.spc6` in the previous example had been deleted, then the new file will take the available name, `dia.spc6`, despite the fact that the previous file had been named `dia.spc13`, and therefore will be out of sequence. Similar considerations apply to the `name.dmp`, `name.sym`, `name.str` and `name.sadp` files.

7 Compiling DIFFaX.

DIFFaX source code is contained in the three text files `DIFFaX.f`, `DIFFaX.par` and `DIFFaX.inc`. `DIFFaX.par` and `DIFFaX.inc` are 'include' files and normally the compiler expects them to be in the same directory as `DIFFaX.f` when it is compiled. DIFFaX is written, as far as is practical, in fortran 77, using commonly found extensions, such as `real*8`, rather than `DOUBLE PRECISION`. For legibility the source code is written using lower- and upper-case characters. Useful FORTRAN extensions such as `DO WHILE. . .`, `CYCLE` etc. were scrupulously avoided. Instead, the equivalent (but ugly) `IF . . . GOTO` constructions were used. DIFFaX was developed mainly on Macintosh systems. However, DIFFaX v1.7xx has been compiled and run successfully on various computer systems and FORTRAN compilers. DIFFaX v1.801 has been tested on Macintosh and Silicon Graphics systems. DIFFaX v1.812 has been tested only on the Macintosh using the Absoft f77 compiler. Small modifications need to be made to the source code in order to get DIFFaX to compile properly on the Cray cft77 compiler. This is mainly due to cft77's stricter adherence to the ANSI FORTRAN77 standard, and to the fact that the Cray is a 64-bit machine, which introduces ambiguities as to the meaning of "double precision" (see section 7.3 for further details concerning running on the Cray).

Built-in constants used by DIFFaX, such as fixed array sizes etc. (except for the "detune" parameter discussed in section 8.3) are assigned in the file `DIFFaX.par`. The meaning of each

of the constants is explained in more detail in the `DIFFaX.par` file.

7.1   Binary files.

A problem can arise when writing the binary file `name.sadp`, which is supposed to be an unformatted $256 \times 256 \times 8$ bit file, exactly 65,536 bytes big. Some compilers pre-pend a record header to each 256 byte row of data, increasing the physical length of each row of data. There are two ways around this problem (i) Find out the actual row length (usually 260 bytes) and open the image file as a 260 x 256 byte image (and ignore the first 4 pixels in each row), or (ii) modify the binary file open statement in the source code. In VAX fortran the binary file open statement should have the specifier `recordtype = 'stream'` added, *i.e.* the specific statement in `DIFFaX.f` should read

```
if(sa.ne.op) open(unit = sa, file = outfile, status = 'new',
|       form = 'unformatted', err = 990, iostat = io_err,
|       recordtype = 'stream')
```

The exact fix, however, will depend on the compiler.

7.2   Precision.

DIFFaX uses double precision (`real*8`) floating point variables. On most machines it will run faster using single precision. DIFFaX has been tested most extensively using double precision variables. There is a risk of numeric overflow in the vicinity of sharp peaks if single precision (`real*4`) is used on some complicated structures.

7.3   Double Precision and the Cray.

On the Cray, however, single precision should be specified, using the '`-dp`' cft77 compiler option. Cray double precision is equivalent to `real*16`. This slows up the Cray unnecessarily and uses up a lot of memory. Consequently, if `DIFFaX.f` is compiled in unmodified form, the cft77 compiler will flag assignment statements such as `REAL*8`, `COMPLEX*16` and `INTEGER*4`, and will generate a screed of warning messages. All variable declarations of the form `REAL*4`, `REAL*8`, `INTEGER*1`, `INTEGER*2`, `INTEGER*4`, `LOGICAL*1` etc. should be replaced with the generic `REAL, INTEGER, LOGICAL` etc. The Cray does not recognize the non-standard functions `DCMPLX, DIMAG and DBLE`. These should be replaced everywhere by the single precision `CMPLX, IMAG and REAL` intrinsic functions.

7.4   The "saveall" option.

With some compilers, DIFFaX may crash with bus errors. The author has not traced down the cause of every problem explicitly. For one compiler it had something to do with

ambiguities in the ansi77 standard about the `save` statement. There is, of course, the possibility that there is an elusive bug in the DIFFaX code – perhaps an uninitialized variable is used prematurely. The DIFFaX code has been scoured relentlessly for bugs of this kind. However, there remains the chance that there are still insidious bugs lurking there. If you find bugs, please let the author know about them. He will be delighted to fix them! Meanwhile, declaring all variables to be static invariably 'fixes' the run-time problems. With some compilers, all variables can be forced to be treated as static at compile time by specifying `-saveall` as a compiler option (sometimes this is specified by '-s').

If your compiler does not provide a `-saveall` (or equivalent compiler option), then you can uncomment the `save` statement at the top of the `DIFFaX.inc` file. `DIFFaX.inc` is included in almost every subroutine in `DIFFaX.f`. Further, you should uncomment the `save` statement at the top of those subroutines that do not include `DIFFaX.inc`. These subroutines can be found by searching the file `DIFFaX.f` for the word "`save`".

However, use of the `-saveall` command may significantly slow down program execution, so it may pay to determine whether or not `-saveall` is needed by first compiling without it, and making a trial run of the DIFFaX executable. Run-time messages about bus errors are usually a hint that the `-saveall` statement is needed.

8   How DIFFaX works.

8.1   Diffraction Described Recursively.

DIFFaX takes advantage of the self-similarity of stacking sequences, present in periodic objects constructed by non-deterministic layer stacking, to rapidly compute diffraction intensity. A simple model of the type of recursion found in crystals is provided by a tall brick wall that contains only one type of brick, where the bricks have been cemented in place with flawless regularity. Neglecting the finite height of the wall, the wall has a recursive property, namely that a brick wall is equivalent to a row of bricks plus another (displaced) brick wall (see Figure 2). Crystals can be viewed similarly in terms of sheets of unit cells. This recursive property is reflected in many physical properties of crystals, such as the scattered wavefunction when such crystals are irradiated. Thus, the scattered wave function from a crystal centered on any layer, is equivalent to the scattering contribution from that layer plus the scattered wavefunction from the displaced crystal centered at the next layer. More formally,

$$\Psi(\mathbf{u}) = F(\mathbf{u}) + \exp(-2\pi i \mathbf{u} \cdot \mathbf{R}) \, \Psi(\mathbf{u}) \tag{7}$$

Here, $\Psi(\mathbf{u})$ is the scattered wavefunction at reciprocal lattice vector $\mathbf{u}$, $F(\mathbf{u})$ the scattering contribution from the layer at the origin, and $\mathbf{R}$ is the shift between the two origins. This
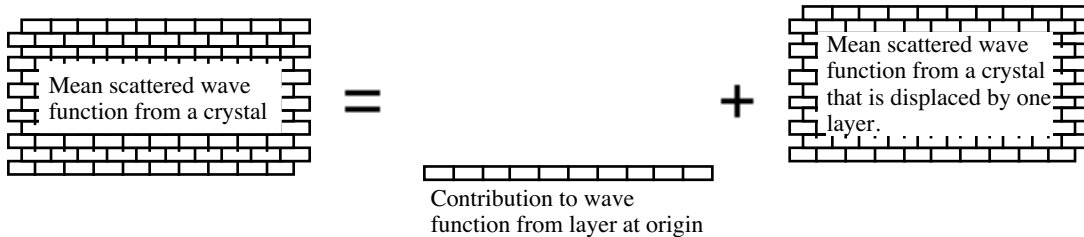
Figure 2: Illustration of the recursive property of periodic objects.

equation embodies Laue's diffraction equation, and leads immediately to a solution for the scattered wavefunction from the whole crystal.

$$\Psi(\mathbf{u}) = \frac{F(\mathbf{u})}{1 - \exp(-2\pi i \mathbf{u} \cdot \mathbf{R})} \tag{8}$$

This equation tells us that there are sharp diffracted peaks at values of $\mathbf{u} = (h, k, l)$ that satisfy $\mathbf{u} \cdot \mathbf{R} = hR_x + kR_y + lR_z = 2n\pi$, where n is an integer, and $\mathbf{R} = (R_x, R_y, R_z)$ is in relative cell coordinates.

This recursive property can be extended to crystals containing defects. Figure 3 shows our brick wall, now containing two types of brick arranged in homogeneous layers. In the crystalline analog, recursion relations between the wavefunctions for crystals centered on the two types of layers can be defined. The stacking of sheets is now determined probablistically, by the introduction of the transition probabilities α_{ij}.

$$\Psi_i(\mathbf{u}) = F_i(\mathbf{u}) + \sum_{j=1,2} \alpha_{ij} \exp(-2\pi i \mathbf{u} \cdot \mathbf{R}_{ij}) \, \Psi_j(\mathbf{u}) \tag{9}$$

The simultaneous equations are readily solved to yield the wavefunctions $\Psi(\mathbf{u})$. Since we are computing the diffraction from a statistical ensemble of crystallites, the intensity is given by the incoherent sum

$$\frac{I(\mathbf{u})}{N} = \sum_{i=1,2} g_i \left(F^*(\mathbf{u})\Psi(\mathbf{u}) + F(\mathbf{u})\Psi^*(\mathbf{u}) - |F_i(\mathbf{u})|^2 \right) \tag{10}$$

where the layer existence probability factors g_i are given by equation 6. Interestingly, the recursive property holds even when the crystal contains only a finite number of layers (see the reprint [1] for more details).

8.2 Finite layer widths.

Debye-Scherrer type peak broadening due to finite layer widths is emulated phenomenologically in DIFFaX v1.801. The proper full calculation would require integration over noninteger values of h and k, which would be forbiddingly time-consuming. Here we present an outline of the methodology used in DIFFaX.
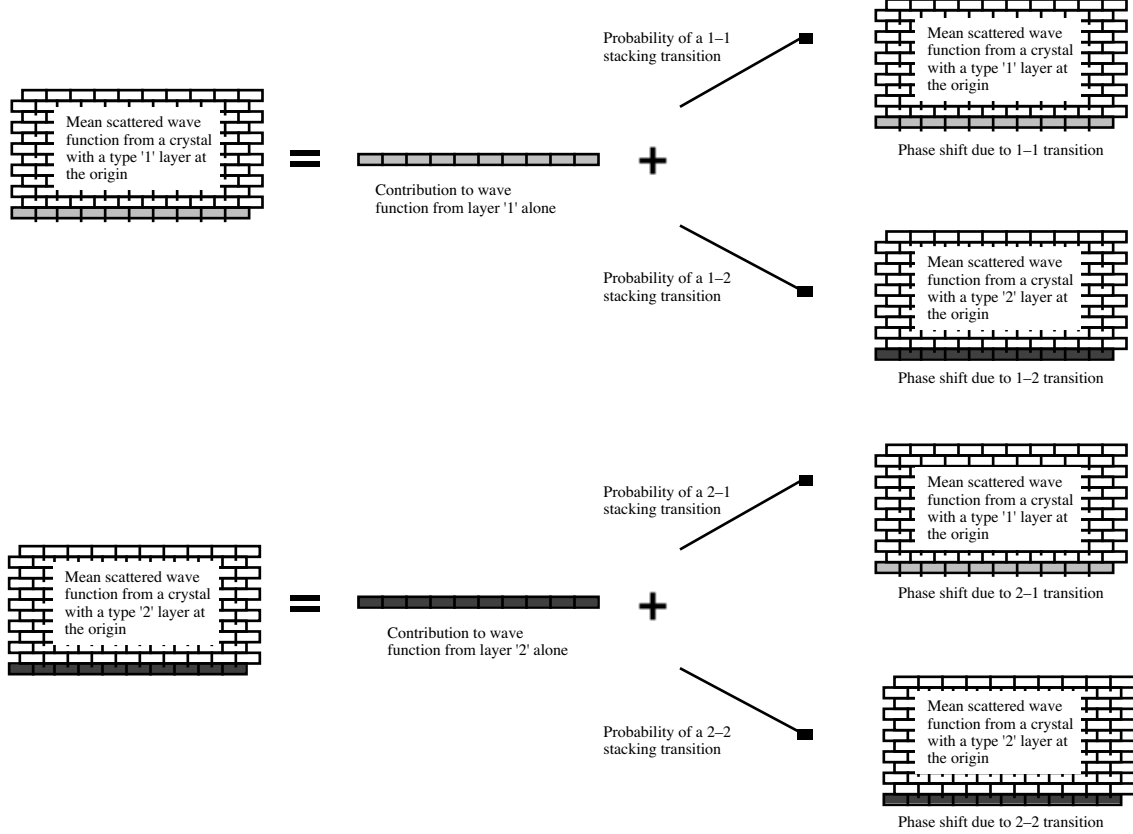
Figure 3: Illustration of recursion in randomly faulted periodic objects.

A fair representation of shape-broadening can be introduced by assuming that, from a statistical ensemble of crystallites, the average half-width of the intensity profile in the vicinity of reciprocal lattice point $h0l$ for an average crystal width W_a along $\mathbf{a}$ is given by

$$\Gamma = \frac{1}{2W_a} \tag{11}$$

The intensity profile is assumed to approximate a Lorentzian in shape, with half-width Γ. The Lorentzian is intended to approximate the more formally accurate 'sinc' function form factor,

$$\left(\frac{\sin(\pi(h' - h)W_a)}{\pi(h' - h)W_a} \right)^2 \tag{12}$$

after having averaged out the oscillations due to the presence of a spread in the W_a values in the powder sample. (Here h' is a continuous variable whose value is in the vicinity of the integer value h).

All crystal shapes are assumed to be rectangular (or cylindrical, depending on whether two widths, or a diameter, were specified), so the more complicated form factors due to special crystal morphology are ignored. When the widths W_a and W_b are not equal, it is

assumed that the contours of equal intensity about a peak are elliptical. This intensity profile in the h–k plane is intercepted by the Ewald sphere at an angle that is determined by the l-value of the reflection (see Figure 4). Thus, the half-width of the peak at scattering vector hkl due to shape broadening can be shown to be (for perpendicular $\mathbf{a}$ and $\mathbf{b}$ axes)

$$\Gamma = \frac{1}{2Q_{hkl}} \left[\left(\frac{Q_{h00}}{W_a} \right)^2 + \left(\frac{Q_{0k0}}{W_b} \right)^2 \right] \tag{13}$$

where Q_{hkl} is the magnitude of the reciprocal lattice scattering vector at hkl, and is given by

$$Q_{hkl}^2 = a_0 h^2 + b_0 k^2 + c_0 l^2 + d_0 hk \tag{14}$$

with the constants

$$\begin{aligned}
a_0 &= \frac{1}{a^2 \sin^2(\gamma)} \\
b_0 &= \frac{1}{b^2 \sin^2(\gamma)} \\
c_0 &= \frac{1}{c^2} \\
d_0 &= \frac{-2\cos(\gamma)}{ab \sin^2(\gamma)}
\end{aligned} \tag{15}$$

Depending on the relative values of W_a and W_b, and the value of l, different peaks will be broadened differently. For non-perpendicular axes, equation 13 is modified slightly to reflect the fact that the W_a and W_b measurements are not mutually perpendicular. Consequently, the shape-broadened half-width of the peak at scattering angle θ in the powder pattern is given by

$$\Gamma_\theta = \frac{\lambda}{2Q_{hkl}} \left[a_0 \left(\frac{h + k\sin(\gamma)}{W_a \cos(\gamma)} \right)^2 + b_0 \left(\frac{k}{W_b} \right)^2 \right] \tag{16}$$

provided $h \neq k = 0$.

The broadening of $00l$ peaks is handled a little differently. When both h and k are zero, the Ewald sphere is tangential to the plane of the shape broadening profile of the $00l$ peak. This produces an asymmetrical peak with a sharp onset and a long tail, whose half-width depends in a complicated way on the relative values of W_a and W_b. If W_a and W_b are close in value, the tail drops off the slowest.

Despite the simplifications and approximations used, the addition of layer-width broadening to powder pattern peaks still introduces a significant computational overhead to the powder pattern calculations. In particular, shape-broadening adds an extra degree of freedom to the simulations that can easily overwhelm the user with data. As a rule, layer widths should be specified as `INFINITE` when exploring the effects of stacking fault probabilities.
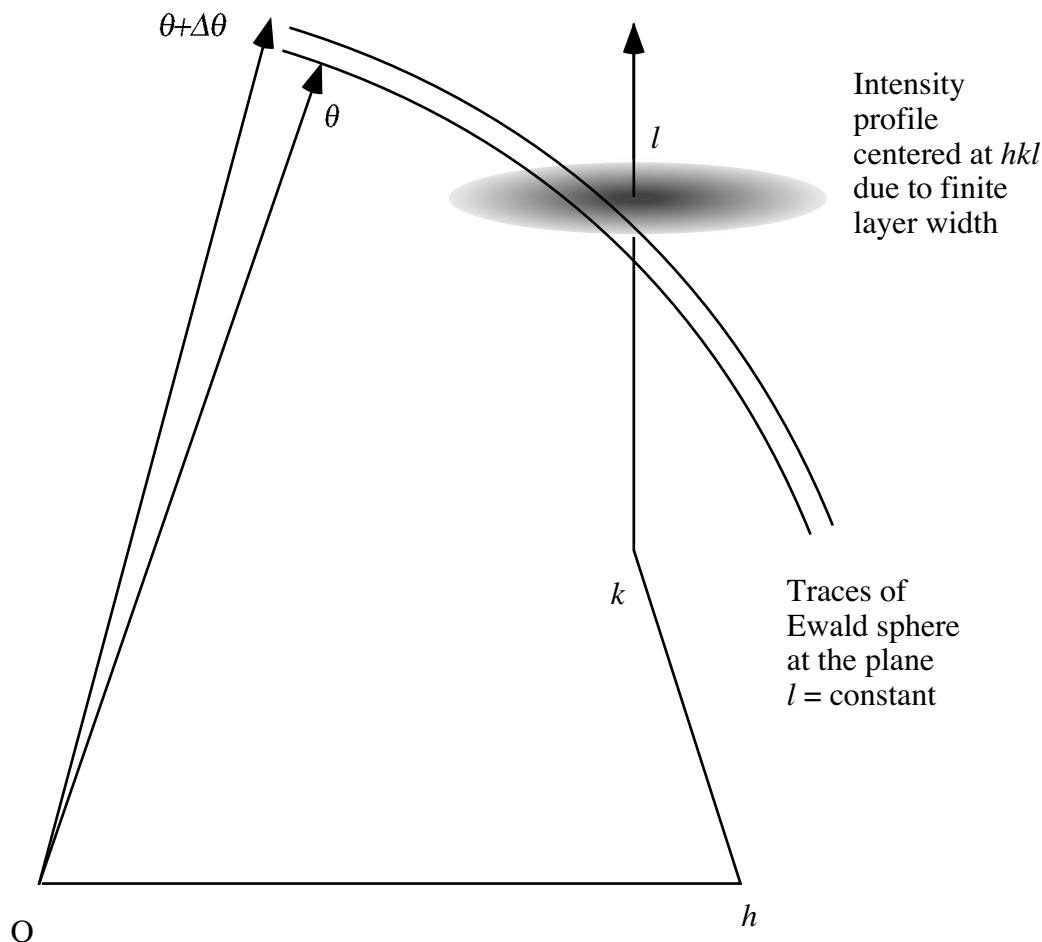
Figure 4: Geometry of the Ewald sphere intercepting a shape-broadened peak at *hkl*.

The addition of the shape broadening profiles should be reserved for final refinements of simulations in systems that are known to have small crystallites, and where the simulations for large crystals look promising.

8.3 The "detune" parameter.

Most users will not need to modify the internal parameters used within the DIFFaX program, and thus these have all been set at compile time and cannot be modified dynamically by the user. These need to be changed within the file `DIFFaX.par`, and `DIFFaX.f` needs to be recompiled. Under special circumstances, however, the "detune" parameter may need to be adjusted by the user. What is this "detune" parameter? Why is it hidden from the user in this way? To understand the role of the detune parameter, we must first examine what the computer is trying to do when it integrates intensities under sharp peaks. Sharp diffraction peaks from perfect crystals arise from a resonance condition between the incoming wave

and the diffracting lattice. Ideally, such a resonance produces a peak in intensity that can be described as a delta function, with infinite height and essentially zero width, although the integrated intensity under the peak is finite. True delta functions cannot be properly represented in a computer. In particular, they cannot be detected or integrated properly. To help DIFFaX detect and integrate these delta functions, DIFFaX resorts to a trick whereby the diffraction resonance is effectively detuned slightly. The Q-factor of the resonance is no longer infinite (but still large), and importantly, the peak develops a detectable width. The result is that the peak can be integrated by the computer. The detuning is brought about by reducing all the stacking probabilities by a small amount, so that the sum of all transition probabilities from any given layer is slightly less than unity. This is a mathematical trick, and the principle behind this trick is explained in more detail in the reprint [1]. The fraction by which all the stacking probabilities is reduced we refer to as the detune parameter. One way to understand this is to examine equation 7 for the scattered wavefunction from a perfect crystal containing only one layer type

$$\Psi(\mathbf{u}) = \frac{F(\mathbf{u})}{1 - \exp(-2\pi i \mathbf{u} \cdot \mathbf{R})} \tag{17}$$

There is a resonance condition (a sharp peak) when $\exp(-2\pi i \mathbf{u} \cdot \mathbf{R}) = 1$, when the denominator is zero. To make equation 17 more palatable to a computer, we modify the denominator so that

$$\Psi(\mathbf{u}) = \frac{F(\mathbf{u})}{1 - \Omega \exp(-2\pi \mathbf{u} \cdot \mathbf{R})} \tag{18}$$

where Ω is a number slightly less than unity, *i.e.* $\Omega = 1 -$ detune. The denominator now will never be zero and the real part will always be positive. Ordinarily, Ω should be unity, the probability that a layer is followed by another identical layer. A consequence of detuning the sharp peaks in this way, is that they are broadened slightly. This broadening becomes apparent if one plots spectra on a log-linear scale, and can become a nuisance if background diffuse scattering is being simulated and fitted to a power law, or if satellite peaks close to a sharp peak are being simulated. The default value of detune is 0.001. To sharpen the sharp peaks, the value of detune should be decreased. However, if detune is made too small, there is a risk that DIFFaX will miss the sharp peaks, or will fail to integrate them properly. Setting detune to zero may cause DIFFaX to fail, because of a "divide by zero" condition. Sharp peaks will not be integrated properly. Spectra may even contain negative intensities. Consequently, detune is kept away from casual misuse by not including it as a run time variable.

The impact of "detune" on peak width is illustrated in Fig. 5, where the continuous line is for detune $= 10^{-3}$, and the dotted line is for detune $= 10^{-4}$. The peak width is reduced by a factor of ten when detune is reduced by a factor of ten. Similarly, the peak heights
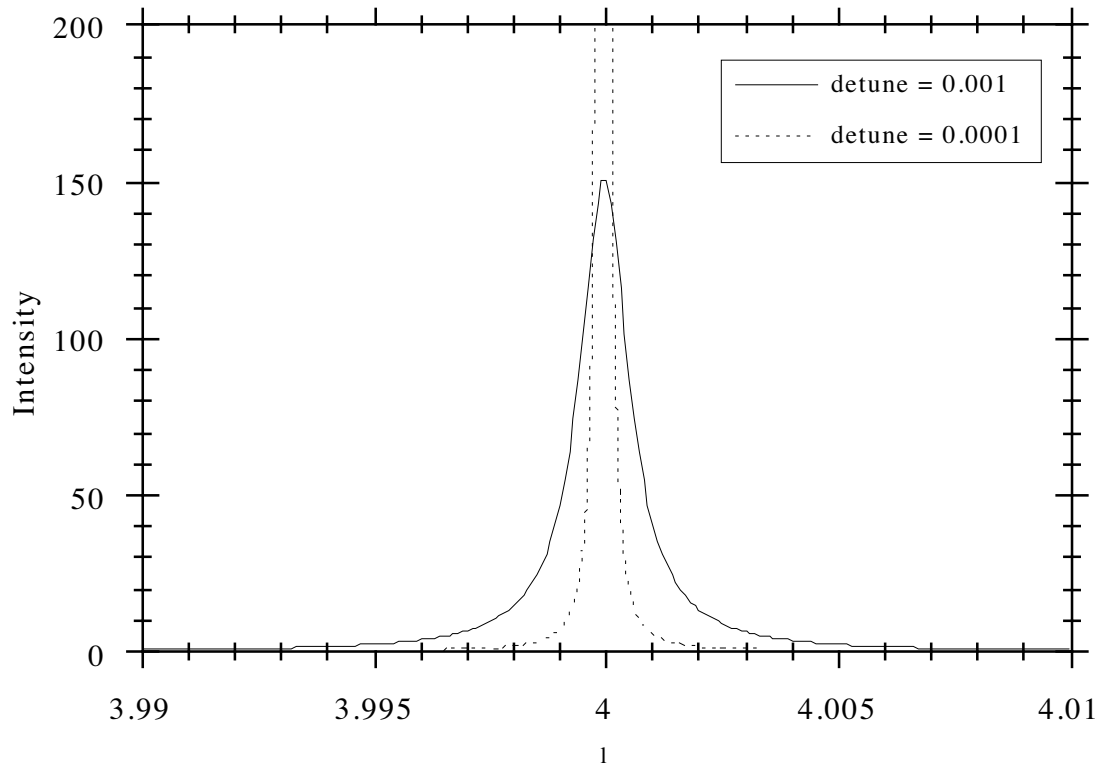
Figure 5: Intensity trace along l showing the effect of changing the parameter "detune" on the width of the 004 peak in unfaulted GaAs (see appendix A.3, page 55).

increase by a factor of ten, such that the integrated area is approximately constant. To change detune, locate the subroutine DETUN in the file DIFFaX.f.

```
      subroutine DETUN()
      include 'DIFFaX.par'
      include 'DIFFaX.inc'
*
      integer*4 i, j
      real*8 delta
*
      delta = eps3
*
* A value of delta = 0.001 is found to be optimum.
* If preferred, user can specify 'delta' interactively
* using the following three lines.
C      write(op,400) 'Enter detune parameter'
C      read(cntrl,*) delta
C      if(CFile) write(op,401) delta
*
      do 10 i = 1, n_layers
        do 20 j = 1, n_layers
          detune(j,i) = ONE - abs(delta)
   20   continue
   10 continue
*
      return
C  400 format(1x, a)
C  401 format(1x, g12.5)
```

```
end
```

Either, change the value `delta` (note `eps3` is defined to be 10^{-3}, and `eps4` is defined to be 10^{-4} etc.) to the new value required, or uncomment the five lines of code beginning with "C" that allow the user to set "detune" interactively. DIFFaX then needs to be recompiled.

9   References

References

[1] M. M. J. Treacy, J. M. Newsam & M. W. Deem 1991 "A General Recursion Method for Calculating Diffracted Intensities From Crystals Containing Planar Faults" *Proc. Roy. Soc. Lond.* **A433**, 499–520.

[2] S. Hendricks and E. Teller 1942 "X-ray interference in partially ordered layer lattices" *J. Chem. Phys.* **10**, 147–167.

[3] B. E. Warren 1959 "X-ray diffraction", pp. 275–312. Reading, Massachusetts: Addison-Wesley.

[4] J. M. Cowley 1981 "Diffraction Physics", pp. 388–400. New York and London: North-Holland.

10   Change history for DIFFaX.

1. DIFFaX v1.812 fixes a bug where the sign of the *Fats-Waller* factors C_{12}, C_{23}, C_{31} was inadvertently ignored.

2. DIFFaX v1.811 fixed some f77 compiler compatibility issues.

3. DIFFaX v1.810 fixed a minor printing bug.

4. DIFFaX v1.809 fixed bug in the computation of the shape broadening due to finite layer widths.

5. DIFFaX v1.808 fixed an f77 compiler compatibility issue.

6. DIFFaX v1.807 modified the default stacking constants when a stacking probability is zero.

7. DIFFaX v1.806 and higher can now generate 16-bit greyscale SAD patterns, as well as 8-bit patterns.

8. In DIFFaX v1.805 atom coordinates and stacking vectors can be entered as fractions. That is, "1/3" is now accepted in place of 0.3333.

9. DIFFaX v1.804 now simulates peak broadening in powder patterns due to finite crystal dimensions perpendicular to the stacking direction. This feature requires an additional line in the structure data file specifying the average layer widths parallel to **a** and **b** (see section 2.7.5 on page 13). Note that broadening due to a finite number of layers is handled differently by DIFFaX (see section 2.9, page 18).

A   Appendices

A.1   Example data file for random faults in diamond.

Diamond belongs to the cubic space group Fd$\overline{3}$m. It supports twinning on the cubic {111} planes, thus the [111] direction becomes the new DIFFaX **c** axis. The structure is best represented in the hexagonal setting, where $(111)_{\text{cubic}}$ is equivalent to $(001)_{\text{hexagonal}}$, and the new cell axes are along $[1\overline{1}0]_{\text{cubic}}$ and $[10\overline{1}]_{\text{cubic}}$. The (111) planes are centrosymmetric about the origin, and two distinct forms occur in any given crystal, related by a mirror plane (or, equivalently, a 60° rotation). When like layers stack, the sequence has (local) cubic symmetry. When mirror-related planes stack alternately, to form the lonsdaleite structure, the sequence has (local) hexagonal symmetry. See figure 6 for a graphic description.

```
Data file dia.dat

{data file for diamond, with random hexagonal intergrowths}
{probability of cubic stacking = 70%}

INSTRUMENTAL                          {Header for instrumental section}
X-RAY                                 {Simulate X-ray diffraction}
1.5418                                {X-ray wavelength}
{gaussian 0.1 trim}                   {Instrumental broadening (much faster)}
PSEUDO-VOIGT 0.1 -0.036 0.009 0.6 TRIM {Instrumental broadening (much slower)}

STRUCTURAL                            {Header for structural section}
2.52 2.52 2.06 120.0                  {unit cell coordinates, a, b, c, gamma}
6/MMM                                 {hexagonal, c = cubic [111]}
2                                     {111 sheet, plus its mirror}
infinite                              {Layers are very wide in the a-b plane}

LAYER 1
CENTROSYMMETRIC
 C    1   -.333333  -.166667  -.125  1.0  1.0
{C    2    .333333   .166667   .125  1.0  1.0, related to 1 by -1}

LAYER 2
CENTROSYMMETRIC
 C    1    .333333   .166667  -.125  1.0  1.0
{C    2   -.333333  -.166667   .125  1.0  1.0, related to 1 by -1}

STACKING                              {Header for stacking description}
recursive                            {Statistical ensemble}
infinite                             {Infinite number of layers}

TRANSITIONS                          {Header for stacking transition data}

{Transitions from layer 1}
0.7   0.666667   0.333333  1.0    {layer 1 to layer 1}
0.3   0.0        0.0       1.0    {layer 1 to layer 2}

{Transitions from layer 2}
0.3   0.0        0.0       1.0    {layer 2 to layer 1}
0.7  -0.666667  -0.333333  1.0    {layer 2 to layer 2}
```

[1̄10] view of diamond layers
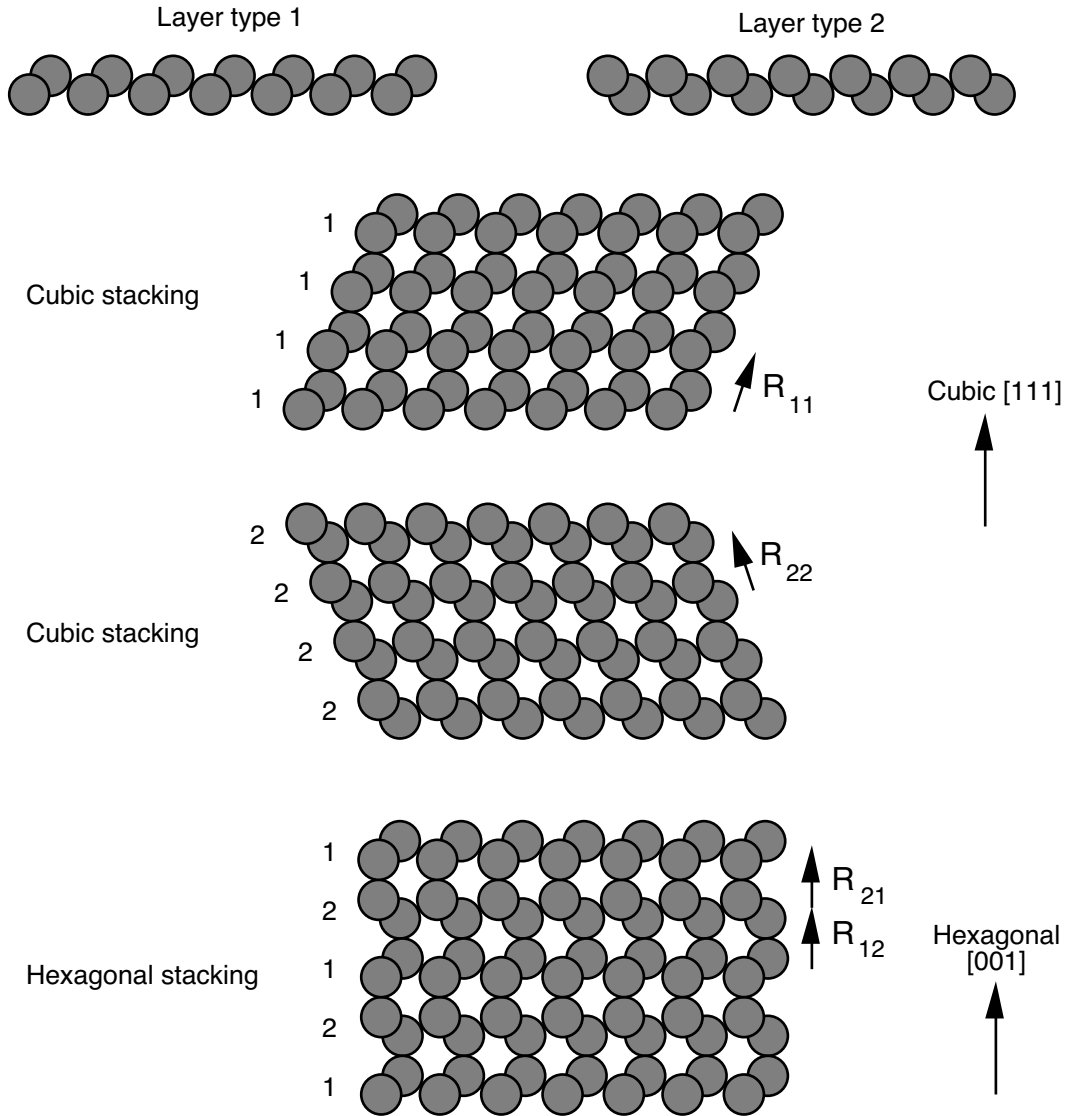
Figure 6: [11̄0] view of diamond/lonsdaleite layers.

A.2   Example data file describing clustered faults in diamond.

Appendix A.1 described diamond/lonsdaleite intergrowths, where the twin faults were randomly dispersed throughout the crystal, without correlation. It is also possible to describe the faulting as if there were a 'memory', or Reichweite, so that the stacking history plays a role in determining the fault probability. For example, a crystal may be stacking in a cubic sequence with a low probability that it will twin. Once twinned, however, the probability of a fault in the new hexagonal sequence may also be low, and the crystal may continue for some while before faulting back to a cubic sequence. Figure 7 illustrates how to describe

52

such faulting by defining four layer types.

4 layers are required to describe fault clustering in diamond
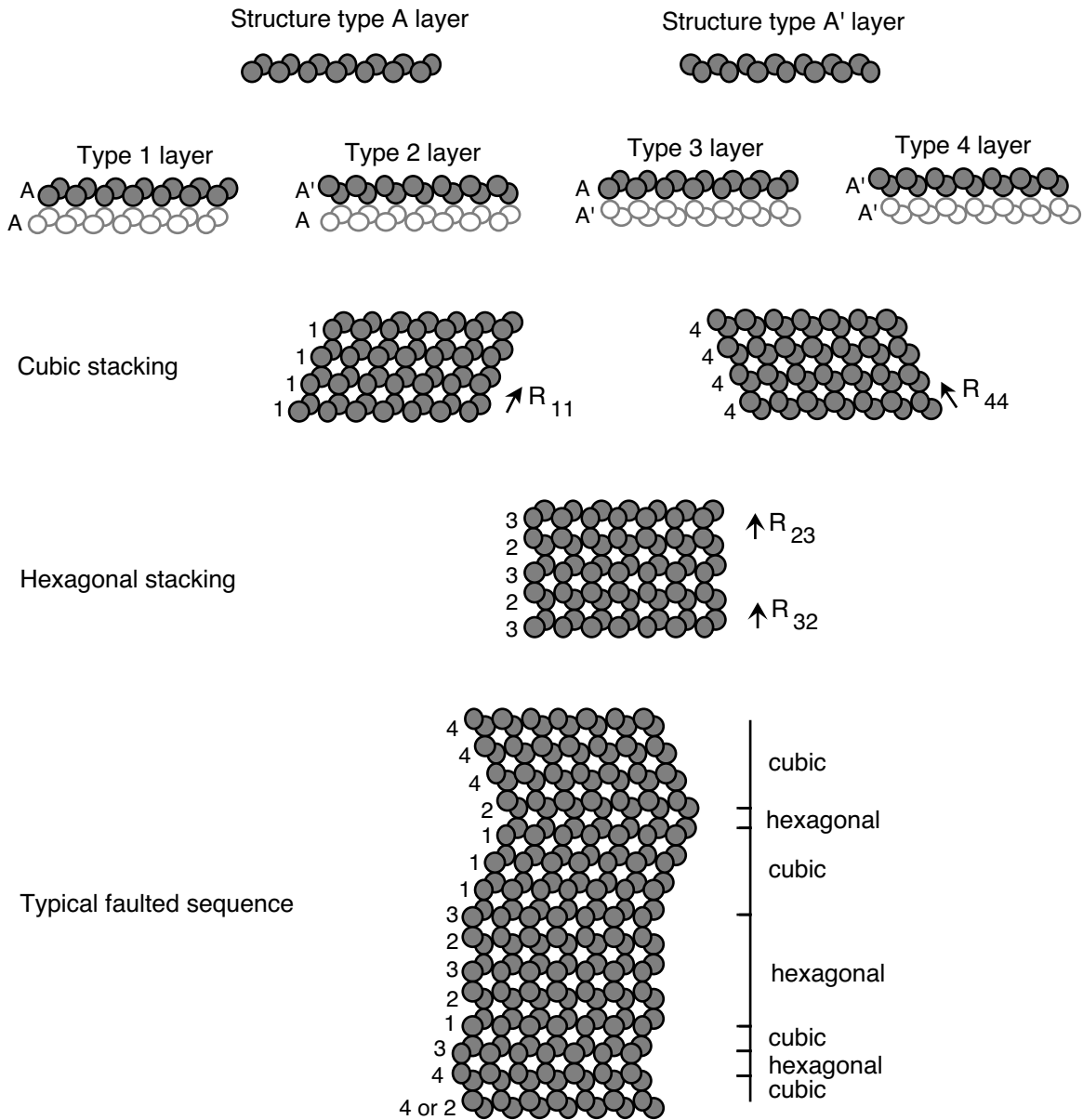
Figure 7: [1$\bar{1}$0] view showing fault clustering in diamond/lonsdaleite.

```
Data file Clustdia.dat

{data file describing clustered faulting in diamond}

{There are four layers in total, two of which are}
{structurally distinct. They are related by a mirror}
{in the a-b plane, and are labeled layers A and A'.}
```

```
{Layer 1 is an A layer which is preceded by an A layer}
{Layer 2 is an A' layer which is preceded by an A layer}
{Layer 3 is an A layer which is preceded by an A' layer}
{Layer 4 is an A' layer which is preceded by an A' layer}
{1 - 1 is a cubic stacking transition}
{1 - 2 is a stacking fault in a cubic sequence}
{1 - 3 cannot occur}
{1 - 4 cannot occur}
{2 - 1 cannot occur}
{2 - 2 cannot occur}
{2 - 3 is a hexagonal stacking transition}
{2 - 4 is a fault in a hexagonal sequence}
{3 - 1 is a fault in a hexagonal sequence}
{3 - 2 is a hexagonal stacking transition}
{3 - 3 cannot occur}
{3 - 4 cannot occur}
{4 - 1 cannot occur}
{4 - 2 cannot occur}
{4 - 3 is a stacking fault in a cubic sequence}
{4 - 4 is a cubic stacking transition}

{In this file. . .}
{the probability of a fault in cubic stacking = 1-2 = 4-3 = 10%}
{                                    thus, 1-1 = 4-4 = 90%}
{the probability of a fault in hexagonal stacking = 2-4 = 3-1 = 20%}
{                                    thus, 2-3 = 3-2 = 80%}

INSTRUMENTAL
X-RAY
1.5418
PSEUDO-VOIGT .89 -.32 .08 .6 trim

STRUCTURAL
2.52 2.52 2.06 120.0
-3M         {Symmetry is not necessarily 6/MMM, as there}
            {may not be equal numbers of 1 layers and 4 layers}
4
infinite

LAYER 1
CENTROSYMMETRIC
C     1   -.333333    -.166667    -.125  1.0  1.0

LAYER 2
CENTROSYMMETRIC
C     1    .333333     .166667    -.125  1.0  1.0

LAYER 3 = 1

LAYER 4 = 2

STACKING
recursive
infinite

TRANSITIONS
{1}
0.90  0.666667  0.333333  1.0
0.10  0.0       0.0       1.0
0.00  0.0       0.0       0.0       {1-3 Transition cannot occur}
0.00  0.0       0.0       0.0       {1-4 Transition cannot occur}

{2}
0.00  0.0       0.0       0.0       {2-1 Transition cannot occur}
```

```
0.00  0.0        0.0        0.0        {2-2 Transition cannot occur}
0.80  0.0        0.0        1.0
0.20 -0.666667 -0.333333   1.0

{3}
0.20  0.666667   0.333333   1.0
0.80  0.0        0.0        1.0
0.00  0.0        0.0        0.0        {3-3 Transition cannot occur}
0.00  0.0        0.0        0.0        {3-4 Transition cannot occur}

{4}
0.00  0.0        0.0        0.0        {4-1 Transition cannot occur}
0.00  0.0        0.0        0.0        {4-2 Transition cannot occur}
0.10  0.0        0.0        1.0
0.90 -0.666667 -0.333333   1.0
```

A.3   Data file describing GaAs/GaAlAs 001 superlattice, `GaAs_GaAlAs.dat`.

The lattice of GaAs has the cubic diamond framework, with the Ga atoms on the (0,0,0) positions and the As on the (1/4,1/4,1/4) positions. GaAlAs also uses the diamond framework, but in this case the Ga and Al share the (0,0,0) sublattice, and the As occupies the (1/4,1/4,1/4) sublattice. Thus, in a supperlattice structure, the As sublattice is continuous. The superlattice can be described by three layer types; type 1 being pure Ga, type 2 pure As, and type 3 being admixed Ga and As. Pure GaAs can be simulated by alternating Ga and As layers. (Pure GaAs could also be described by one layer that is an amalgam of layer types 1 and 2.) In this example we simulate a finite superlattice by defining an explicit sequence of layers, and we list the layer sequence in the data file. Note that GaAlAs has a slightly larger unit cell dimension than GaAs. In the superlattice, however, the in-plane a and b cell dimensions are the same, because of the tetragonal distortion induced by coherency strains. However, the layer separation along c is not the same in the two materials. This difference is accommodated in the data file by defining a common c cell dimension of 5.65 Å, and making the c-component of the stacking vector for the Al-rich layers to be slightly larger. The exact amount is computed from the formula for the strain along z

$$\epsilon_z = \frac{1+\sigma}{1-\sigma}\epsilon_0 \tag{19}$$

where the strain $\epsilon_0 = (a_{\mathrm{GaAlAs}} - a_{\mathrm{GaAs}})/(a_{\mathrm{GaAlAs}} + a_{\mathrm{GaAs}})$, with a being the cubic unit cell dimensions for each composition, and σ being the isotropic Poisson ratio for the material. All of the strain is absorbed in the GaAlAs layer because in this instance, we assume that the superlattice is grown on bulk GaAs. Typically, $\sigma = 1/3$, and $\epsilon_0 = 0.0018$, giving $\epsilon_z = 0.0036$. Since for GaAs, $R_z = 0.25$, then for GaAlAs, $R_z = 0.25(1 + 0.0036) = 0.2509$.

Data file `GaAs_GaAlAs.dat`

`{data file for an explicit sequence of GaAs/GaAlAs layers}`
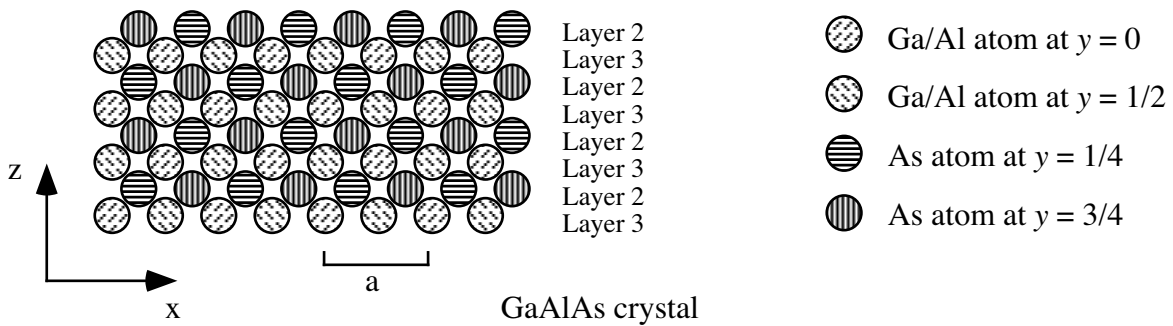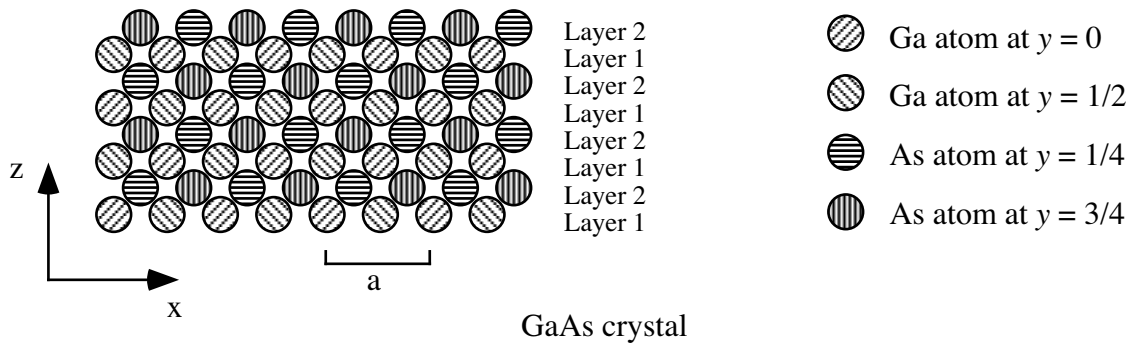
Figure 8: [010] views of GaAs and GaAlAs crystals.

```
{with 7 GaAs layers followed by 10 GaAlAs layers}
{In the GaAlAs, Al:Ga ratio is 3:7, with Ga and Al randomly distributed.}
{There are a total of 680 layers}
{The sequence 1 2 1 2 ... represents GaAs, and the sequence 2 3 2 3 ...}
{represents GaAlAs}

INSTRUMENTAL                       {Header for instrumental section}
X-RAY                              {Simulate X-ray diffraction}
1.5418                            {X-ray wavelength}
none                              {No Instrumental braodening}

STRUCTURAL                        {Header for structural section}
5.65 5.65 5.65 90.0              {unit cell coordinates}
Axial                            {We are only interested in the 00l trace}
3                                {3 layers}
infinite

LAYER 1                                  {(001) plane}
NONE
Ga3+    1  0.0  0.0  0.0  1.0  1.0
Ga3+    2  0.5  0.5  0.0  1.0  1.0        {face centered atom}

LAYER 2                                  {(001) plane}
NONE
As      1  0.0  0.0  0.0  1.0  1.0
As      2  0.5  0.5  0.0  1.0  1.0        {face centered atom}

LAYER 3                                  {(001) plane, Ga and Al randomized}
```

56

Figure 9: [010] view of GaAs/GaAlAs superlattice.

```
NONE
Ga3+     1   0.0   0.0   0.0   1.0   0.7
Al3+     1   0.0   0.0   0.0   1.0   0.3
Ga3+     2   0.5   0.5   0.0   1.0   0.7      {face centered Ga3+ atom}
Al3+     2   0.5   0.5   0.0   1.0   0.3      {face centered Al3+atom}

STACKING                                 {Explicitly defined superlattice}
explicit
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
1 2 1 2 1 2 1 2 1 2 1 2 1 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
```

```
TRANSITIONS                        {Header for stacking transition data}

{Transitions from layer 1}
0.0   0.25   0.25   0.25     {layer 1 to layer 1}
1.0   0.25   0.25   0.25     {layer 1 to layer 2, diamond glide}
0.0   0.25   0.25   0.25     {layer 1 to layer 3}

{Transitions from layer 2}
0.5   0.25   -0.25   0.25     {layer 2 to layer 1, diamond glide}
0.0   0.25   -0.25   0.25     {layer 2 to layer 2}
0.5   0.25   -0.25   0.2509  {layer 2 to layer 3, GaAlAs is strained along c-axis}

{Transitions from layer 3}
0.0   0.25   0.25   0.25     {layer 3 to layer 1}
1.0   0.25   0.25   0.2509  {layer 3 to layer 2, GaAlAs is strained along c-axis}
0.0   0.25   0.25   0.25     {layer 3 to layer 3}
```

B DIFFaX output using a control file.

B.1 The Control File

A control file which will run consecutively the 5 data files found in appendices A.1 to A.3 is given below. The file must be called `control.dif`.

```
{Control file script to run the data given in appendices A.1 through A.3}
{Text in curly braces are comments. Matching braces must be on the same line.}
{This file contains the pre-composed responses to DIFFaX's prompts.}
{When DIFFaX finds this file, control.dif, in the same directory, DIFFaX}
{reads this file for its data instead of prompting the user.}
{This allows the user to run long ''batch'' jobs unattended.}

dia.dat             {File describing random faults in diamond}
1                       {We want a data dump}
1                       {We want a full atomic position dump}
1                       {We want a dump of symmetry evaluations}
3                       {We want a powder pattern}
0 170 .05               {0 to 170 degrees, in steps of 0.05 degrees}
1                       {Full adaptive quadrature on broad peaks}
1                       {Full adaptive quadrature over all l on sharp rows}
Clustdia.dat        {File describing clustered faults in diamond}
0                       {We do not want a data dump}
0                       {We do not want a dump of symmetry evaluations}
3                       {We want a powder pattern}
0 170 .05               {0 to 170 degrees, in steps of 0.05 degrees}
1                       {Full adaptive quadrature on broad peaks}
1                       {Full adaptive quadrature over all l on sharp rows}
GaAs_GaAlAs.dat    {File describing an explicitly defined superlattice}
0                       {No data dump. No symmetry dump - file specified 'Axial'}
1                       {We want a line trace, or streak}
1                       {Full adaptive quadrature on broad peaks}
0 0 3.5 4.5 .001        {h, k, l0, l1, delta_l}
0                       {Do not return to the function menu}
end             {Stop}
```

B.2   Output from the Control File

The output from DIFFaX, using the above control file, is listed below. The values read from the control file are indented.

```
**************************************************
**************************************************
*                                                *
*  DDDD     II   FFFFFF   FFFFFF         X   X  *
*  D   D    II   F        F        aaaa    X X   *
*  D    D   II   FFFF     FFFF    a    a    X    *
*  D   D    II   F        F       a   aa   X X   *
*  DDDD     II   F        F        aaa a   X   X *
*                                                *
**************************************************
***************** DIFFaX v1.801 ******************
**************************************************
**************** 14th JULY, 1995 *****************
**************************************************
*                                                *
*   A computer program for calculating          *
*   Diffraction Intensity From Faulted Crystals  *
*                                                *
*   Authors: Michael M. J. Treacy               *
*            Michael W. Deem                     *
*                                                *
**************************************************

Enter name of structure file.
  dia.dat
Looking for structure data file 'dia.dat'
Opening structure file 'dia.dat'
Enter 1 for DUMP
  1
Reading 'dia.dat'
'dia.dat' read in.
Scattering factor data read in.
Enter 1 for full atomic position dump
  1
Writing data dump to file 'dia.dmp'. . .
Using a ''detune'' value of  0.10000E-02
Enter 1 for a dump of the symmetry evaluations
  1
Checking for conflicts in atom positions . . .
Evaluating point group symmetry of diffraction data. . .
Writing symmetry data to file 'dia.sym'. . .
The diffraction data fits the point group symmetry '6/MMM'
  with a tolerance of one part in  12376
Enter function number:
0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
  3

CALCULATING POWDER DIFFRACTION PATTERN. . .
Enter angular range:
  2theta min, 2theta max, 2theta increment.
 0.0000         170.0          0.5000E-01
Enter 1 for adaptive quadrature on broad peaks.
  1
Enter 1 for adaptive quadrature over all l values
on rows with ''sharp'' spots
  1
Integrating along l at    0   0      'dia.dat'
Full adaptive integration
```

```
Integrating along l at    1   0      'dia.dat'
Full adaptive integration
Integrating along l at    1   1      'dia.dat'
Full adaptive integration
Integrating along l at    2   0      'dia.dat'
Full adaptive integration
Integrating along l at    2   1      'dia.dat'
Full adaptive integration
Adding pseudo-Voigt instrumental broadening
Writing spectrum data to file 'dia.spc'. . .
Spectrum written.

Enter name of structure file.
  Clustdia.dat
Looking for structure data file 'Clustdia.dat'
Opening structure file 'Clustdia.dat'
Enter 1 for DUMP
  0
Reading 'Clustdia.dat'
'Clustdia.dat' read in.
Scattering factor data read in.
Using a ''detune'' value of  0.10000E-02
Enter 1 for a dump of the symmetry evaluations
  0
Checking for conflicts in atom positions . . .
Evaluating point group symmetry of diffraction data. . .
Diffraction point symmetry is 6/MMM
  to within a tolerance of one part in    100
Enter function number:
0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
  3

CALCULATING POWDER DIFFRACTION PATTERN. . .
Enter angular range:
  2theta min, 2theta max, 2theta increment.
 0.0000        170.0        0.5000E-01
Enter 1 for adaptive quadrature on broad peaks.
  1
Enter 1 for adaptive quadrature over all l values
on rows with ''sharp'' spots
  1
Integrating along l at    0   0      'Clustdia.dat'
Full adaptive integration
Integrating along l at    1   0      'Clustdia.dat'
Full adaptive integration
Integrating along l at    1   1      'Clustdia.dat'
Full adaptive integration
Integrating along l at    2   0      'Clustdia.dat'
Full adaptive integration
Integrating along l at    2   1      'Clustdia.dat'
Full adaptive integration
Adding pseudo-Voigt instrumental broadening
Writing spectrum data to file 'Clustdia.spc'. . .
Spectrum written.

Enter name of structure file.
  GaAs_GaAlAs.dat
Looking for structure data file 'GaAs_GaAlAs.dat'
Opening structure file 'GaAs_GaAlAs.dat'
Enter 1 for DUMP
  0
Reading 'GaAs_GaAlAs.dat'
'GaAs_GaAlAs.dat' read in.
Scattering factor data read in.
```

```
Using a ''detune'' value of  0.10000E-02
Checking for conflicts in atom positions . . .
Checking for conflicts in layer stackings . . .
Axial integration only selected.
Enter function number:
0 POINT, 1 STREAK, 2 INTEGRATE, 3 POWDER PATTERN, 4 SADP
  1

CALCULATING INTENSITY ALONG A STREAK. . .
Enter 1 for adaptive quadrature
  1
Enter h, k, l0, l1, delta l
  0   0   3.50000   4.50000   0.00100
Writing streak data to file 'GaAs_GaAlAs.str'. . .
Reached l =     3.54900
Reached l =     3.59900
Reached l =     3.64900
Reached l =     3.69900
Reached l =     3.74900
Reached l =     3.79900
Reached l =     3.84900
Reached l =     3.89900
Reached l =     3.94900
Reached l =     3.99900
Reached l =     4.04900
Reached l =     4.09900
Reached l =     4.14900
Reached l =     4.19900
Reached l =     4.24900
Reached l =     4.29900
Reached l =     4.34900
Reached l =     4.39900
Reached l =     4.44900
Reached l =     4.49900
Streak data file, 'GaAs_GaAlAs.str' written to disk.
Enter 1 to return to function menu.
   0
Enter name of structure file.
end
DIFFaX ended normally.
```

B.3   DIFFaX output for diamond data file `dia.dat` in appendix A.1

Contents of the data dump file, `dia.dmp` for random faults in diamond.

The control file `control.dif` suggested in appendix B.1 requested a dump of the data
file. The data file was called `dia.dat`. The data dump is called `dia.dmp`. This file name is
assigned automatically by DIFFaX. The file `dia.dmp` should look like;

Contents of file `dia.dmp`

```
                  Number of layers =    2
              Number of unique layers =    2
       Number of different atom types =    1
 cell_a,      cell_b,      cell_c,      cell_gamma
  2.5200       2.5200       2.0600       120.00
 Layers are to be treated as having infinite lateral width.
 radiation type = X-RAY
 Wavelength lambda =   1.5418
 Diffraction point group symmetry specified = 6/MMM
 Instrumental broadening to be simulated = PSEUDO-VOIGT
```

```
Full width half-maximum =  0.10000
  Intensity near origin will be ignored

LAYER    1 is equivalent to fundamental LAYER    1
   Existence probability =  0.50000
LAYER    2 is equivalent to fundamental LAYER    2
   Existence probability =  0.50000

fundamental LAYER    1
symmetry = CENTROSYMMETRIC
LAYER composition is:
                      C   :    2.00


ATOM number  1 in layer 1 is 'C    '
   x_rel       y_rel        z_rel          DW          Occ
 -0.33333     -0.16667     -0.12500       1.0000       1.0000
X-ray scattering factor data Ai, Bi
  2.3100       20.844       1.0200       10.207       1.5886
  0.56870      0.86500      51.651       0.21560

fundamental LAYER    2
symmetry = CENTROSYMMETRIC
LAYER composition is:
                      C   :    2.00


ATOM number  1 in layer 2 is 'C    '
   x_rel       y_rel        z_rel          DW          Occ
  0.33333      0.16667     -0.12500       1.0000       1.0000
X-ray scattering factor data Ai, Bi
  2.3100       20.844       1.0200       10.207       1.5886
  0.56870      0.86500      51.651       0.21560


Average crystal composition is:
                      C   : 1.0000



STACKING
Stacking is to be treated RECURSIVELY by DIFFaX.
Number of layers along the fault direction is INFINITE


TRANSITIONS
Layer stacking probabilities and stacking vectors.
   i  j |   alpha_ij      Rx_ij        Ry_ij         Rz_ij
   -----|-------------------------------------------------
        |
   1  1 |  0.70000      0.66667      0.33333       1.0000
   1  2 |  0.30000      0.00000      0.00000       1.0000
        |
   2  1 |  0.30000      0.00000      0.00000       1.0000
   2  2 |  0.70000     -0.66667     -0.33333       1.0000

Anisotropic Layer stacking 'uncertainty' factors.
   i  j |   C11      C22      C33      C12      C13      C23
   -----|-------------------------------------------------------
        |
   1  1 |  0.000    0.000    0.000    0.000    0.000    0.000
   1  2 |  0.000    0.000    0.000    0.000    0.000    0.000
        |
   2  1 |  0.000    0.000    0.000    0.000    0.000    0.000
   2  2 |  0.000    0.000    0.000    0.000    0.000    0.000
```

B.4   Contents of the symmetry evaluation data file `dia.sym`

Contents of the symmetry evaluation data file

The control file `control.dif` also requested a dump of the symmetry evaluations. The symmetry evaluation file is called `dia.sym`. This file name is assigned automatically by DIFFaX. In abbreviated form, the file `dia.sym` should look like;

Contents of file `dia.sym` (partial listing)

```
SYMMETRY EVALUATIONS FOR DATA IN FILE 'dia.dat'
Number of trials per symmetry element =   25
Threshold intensity =            0.612862E-04

Testing for 2-fold axis

   h     k      l                  Intensity
   0     1    1.2675               9.535522
   0    -1    1.2675               9.535522
                            ---------------------
        Average Intensity =        9.535522
     Average variation = +/-       0.000000  (+/- 0.93E-14%)
 Intensity tolerance = +/-         0.095355
 Intensities are consistent with a 2-fold

   h     k      l                  Intensity
   1     0    0.2110               11.197628
  -1     0    0.2110               11.197628
                            ---------------------
        Average Intensity =        11.197628
     Average variation = +/-       0.000000  (+/- 0.24E-13%)
 Intensity tolerance = +/-         0.111976
 Intensities are consistent with a 2-fold

.
.
.
.

   h     k      l                  Intensity
   1     1    1.2210               0.002129
  -1    -1    1.2210               0.002129
                            ---------------------
        Average Intensity =        0.002129
     Average variation = +/-       0.000000  (+/- 0.00     %)
 Intensity tolerance = +/-         0.000021
 Intensities are consistent with a 2-fold

INTENSITY DISTRIBUTION HAS A 2-FOLD AXIS

Testing for a 3-fold axis

   h     k      l                  Intensity
   1     0    0.3126               31.816023
  -1     1    0.3126               31.816217
   0    -1    0.3126               31.816023
                            ---------------------
        Average Intensity =        31.816088
     Average variation = +/-       0.000086  (+/- 0.27E-03%)
 Intensity tolerance = +/-         0.318161
 Intensities are consistent with a 3-fold

   1     0    1.7739               0.763649
```

```
  -1     1      1.7739                          0.763618
   0    -1      1.7739                          0.763649
                              ----------------------
          Average Intensity =                   0.763639
        Average variation = +/-                 0.000014  (+/- 0.18E-02%)
    Intensity tolerance = +/-                   0.007636
    Intensities are consistent with a 3-fold


.
.
.


   0     1      1.1768                          3.078191
  -1     0      1.1768                          3.078191
   1    -1      1.1768                          3.078178
                              ----------------------
          Average Intensity =                   3.078186
        Average variation = +/-                 0.000006  (+/- 0.18E-03%)
    Intensity tolerance = +/-                   0.030782
    Intensities are consistent with a 3-fold

    INTENSITY DISTRIBUTION HAS A 3-FOLD AXIS


    Testing for mirror about the h-l plane

     h     k      l                          Intensity
     1     1      0.6100                         0.003514
     2    -1      0.6100                         0.003514
                              ----------------------
          Average Intensity =                   0.003514
        Average variation = +/-                 0.000000  (+/- 0.81E-06%)
    Intensity tolerance = +/-                   0.000035
    Intensities are consistent with an h-l mirror plane

     1     1      0.6782                         0.003939
     2    -1      0.6782                         0.003939
                              ----------------------
          Average Intensity =                   0.003939
        Average variation = +/-                 0.000000  (+/- 0.13E-05%)
    Intensity tolerance = +/-                   0.000039
    Intensities are consistent with an h-l mirror plane


.
.
.


     1     1      0.2655                         0.007861
     2    -1      0.2655                         0.007861
                              ----------------------
          Average Intensity =                   0.007861
        Average variation = +/-                 0.000000  (+/- 0.85E-07%)
    Intensity tolerance = +/-                   0.000079
    Intensities are consistent with an h-l mirror plane

    THERE IS A MIRROR ABOUT THE H-L PLANE


    The diffraction data fits the point group symmetry '6/MMM'
       with a tolerance of one part in  12376
```

B.5 Contents of the spectrum intensity file `dia.spc`

The spectrum data is stored as a text file in `dia.spc`. The spectrum data records are printed in tab-delimited format, *i.e.*

'2θ in degrees' tab 'raw spectrum intensity' tab 'instrumentally broadened intensity'

The spectrum data file, in abbreviated form, should look like;

Contents of file `dia.spc`

```
0.00000E+00    .000000         .000000
0.50000E-01  0.144523E+08      .000000
0.10000E+00  0.174739E+07      .000000
0.15000E+00    446514.         .000000
0.20000E+00    162174.         .000000
0.25000E+00    72398.4         .000000
0.30000E+00    37032.3         .000000
0.35000E+00    20864.0         .000000
0.40000E+00    12634.9         .000000
0.45000E+00    8092.45         .000000
0.50000E+00    5420.24         .000000
0.55000E+00    3765.62         .000000
0.60000E+00    2696.97         .000000
0.65000E+00    1981.96         .000000
0.70000E+00    1488.98         .000000
0.75000E+00    1140.20         .000000
0.80000E+00    887.827         .000000
0.85000E+00    701.580         .000000
0.90000E+00    561.714         .000000
0.95000E+00    455.031         .000000
0.10000E+01    372.518         .000000
.
.
.
.
.
0.42600E+02    4.94228         5.00760
0.42650E+02    5.06303         5.12953
0.42700E+02    5.19179         5.25957
0.42750E+02    5.32847         5.39769
0.42800E+02    5.47297         5.54386
0.42850E+02    5.62516         5.69802
0.42900E+02    5.78480         5.86009
0.42950E+02    5.95161         6.02996
0.43000E+02    6.12520         6.20746
0.43050E+02    6.30506         6.39241
0.43100E+02    6.49056         6.58464
0.43150E+02    6.68092         6.78397
0.43200E+02    6.87521         6.99031
0.43250E+02    7.07237         7.20378
0.43300E+02    7.27120         7.42486
0.43350E+02    7.47046         7.65471
0.43400E+02    7.66894         7.89569
0.43450E+02    7.86566         8.15231
0.43500E+02    8.06030         8.43294
0.43550E+02    8.25393         8.75318
0.43600E+02    8.45062         9.14288
0.43650E+02    8.66125         9.66243
0.43700E+02    8.91362         10.4459
0.43750E+02    9.28563         11.8648
0.43800E+02    10.0503         15.6333
0.43850E+02    12.7649         28.6585
0.43900E+02    74.8518         54.1308
```

```
0.43950E+02   115.226       61.2062
0.44000E+02   13.2889       34.3516
0.44050E+02   9.99240       16.9498
0.44100E+02   9.09912       12.0554
0.44150E+02   8.65015       10.3552
0.44200E+02   8.33094       9.42702
0.44250E+02   8.05508       8.80419
0.44300E+02   7.79252       8.32827
0.44350E+02   7.53186       7.92957
0.44400E+02   7.26908       7.57430
0.44450E+02   7.00346       7.24517
0.44500E+02   6.73576       6.93314
.
.
.
.
.
.
0.16900E+03   0.785915E-01  0.511187E-01
0.16905E+03   0.786012E-01  0.503597E-01
0.16910E+03   0.786109E-01  0.495880E-01
0.16915E+03   0.786207E-01  0.488042E-01
0.16920E+03   0.786305E-01  0.480089E-01
0.16925E+03   0.786405E-01  0.472027E-01
0.16930E+03   0.786505E-01  0.463863E-01
0.16935E+03   0.786605E-01  0.455605E-01
0.16940E+03   0.786706E-01  0.447260E-01
0.16945E+03   0.786808E-01  0.438837E-01
0.16950E+03   0.786911E-01  0.430346E-01
0.16955E+03   0.787014E-01  0.421796E-01
0.16960E+03   0.787117E-01  0.413196E-01
0.16965E+03   0.787222E-01  0.404558E-01
0.16970E+03   0.787327E-01  0.395891E-01
0.16975E+03   0.787432E-01  0.387207E-01
0.16980E+03   0.787538E-01  0.378517E-01
0.16985E+03   0.787644E-01  0.369832E-01
0.16990E+03   0.787751E-01  0.361164E-01
0.16995E+03   0.452758E-01  0.352524E-01
0.17000E+03   0.250168E-15  0.343923E-01
```

B.6   Plotted output from the control file run

The powder pattern data can be plotted by any plotting program that accepts tab-delimited text data files. The following plots were constructed using a Macintosh plotting program called Igor$^{\text{TM}}$.
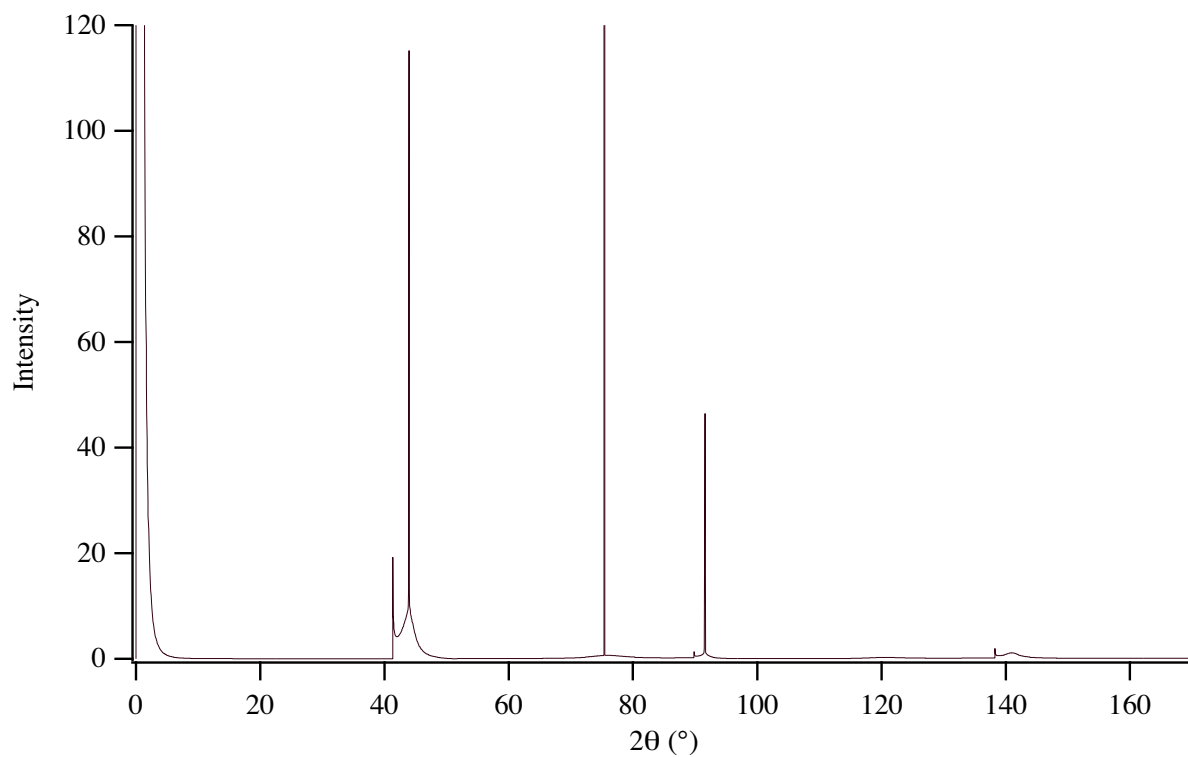
Figure 10: Plot of unbroadened data in file `dia.spc`.
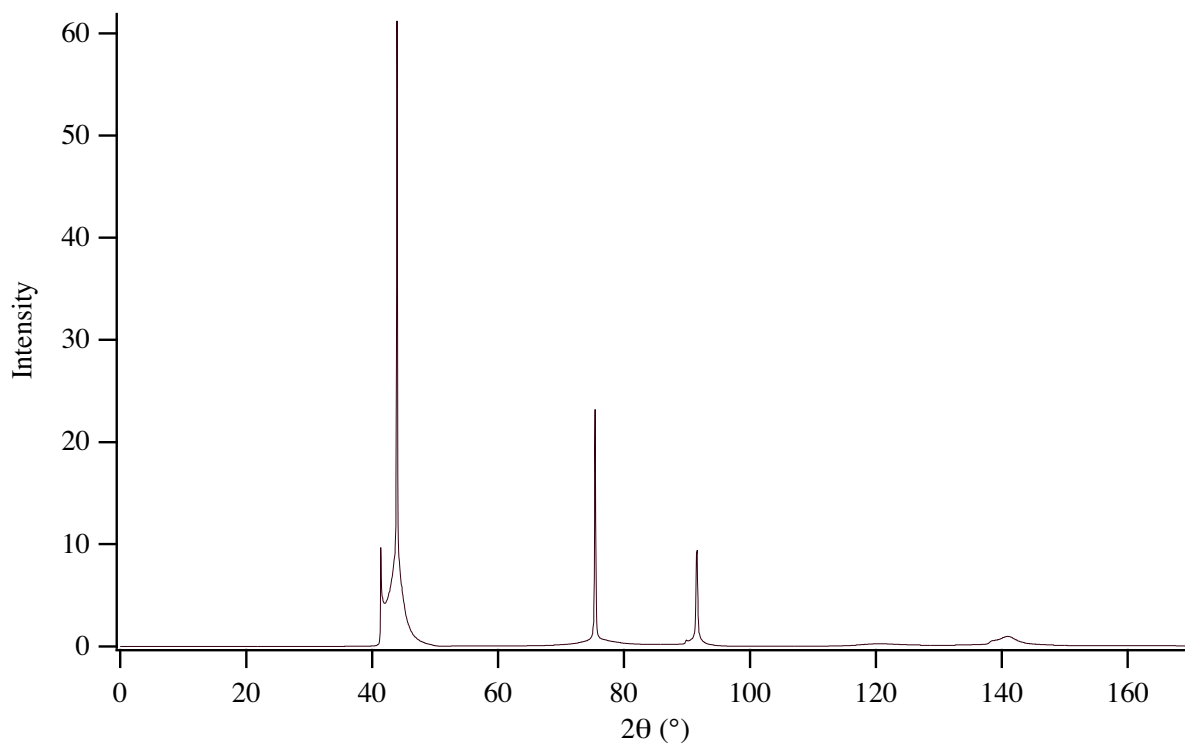
Figure 11: Plot of pseudo-Voigt broadened data in file `dia.spc`.

67
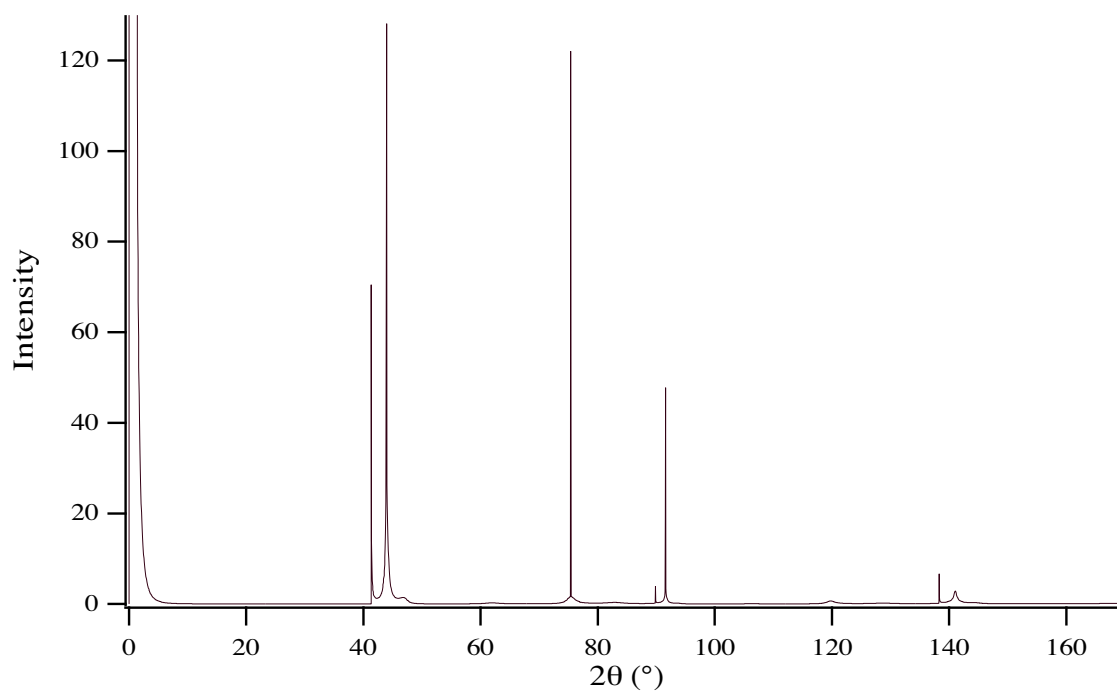
Figure 12: Plot of unbroadened data in file `Clustdia.spc`.
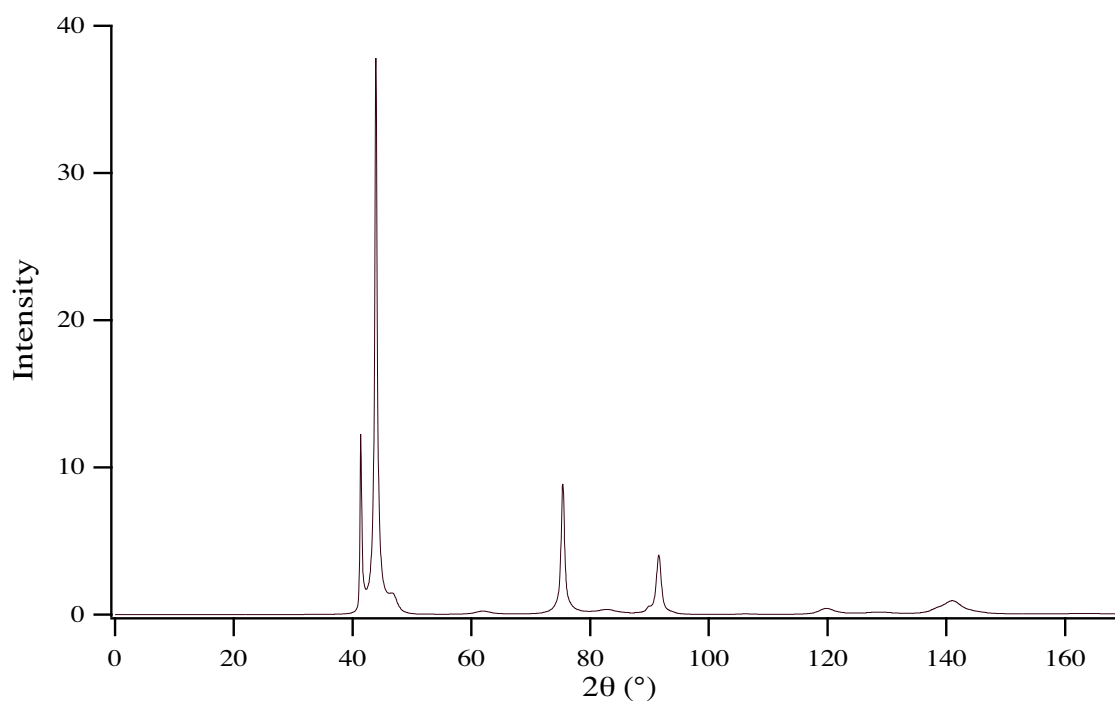
Figure 13: Plot of pseudo-Voigt broadened data in file `Clustdia.spc`.
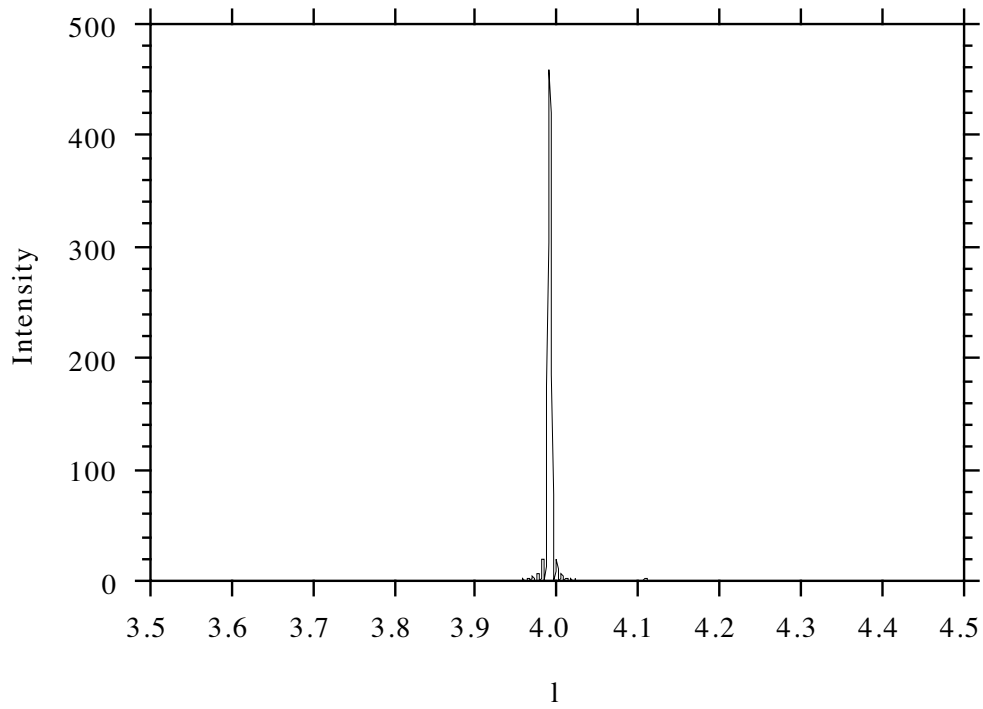
Figure 14: Plot of the "streak" data in GaAs/GaAlAs.str.
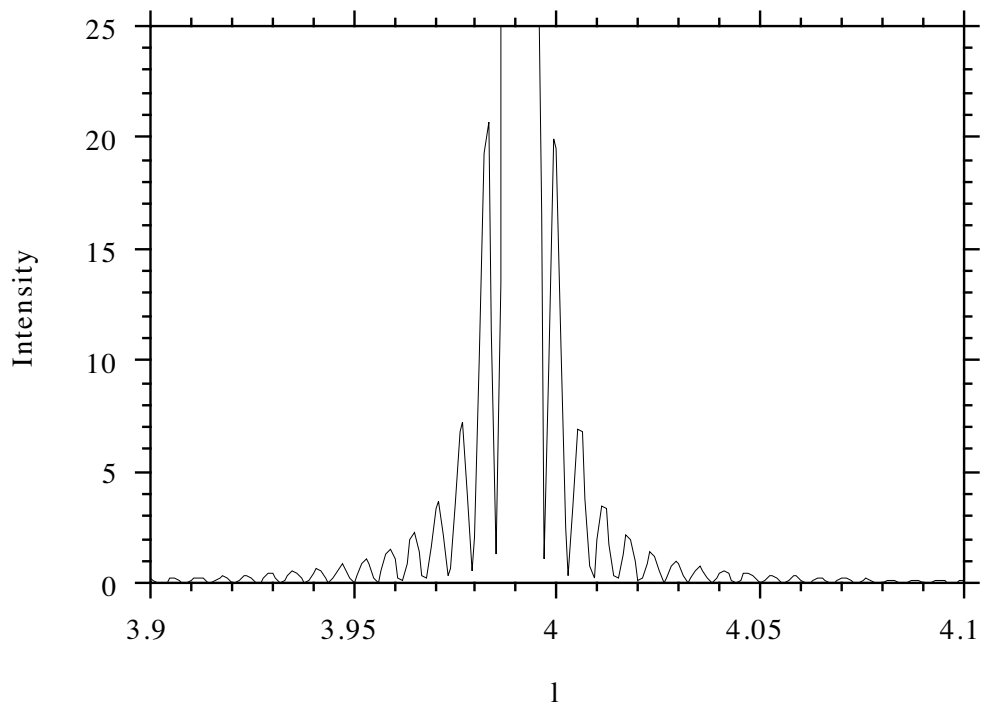
Figure 15: Plot of the "streak" data in GaAs/GaAlAs.str, expanded about $l = 4$.

Index