

A routine for the computation of imperfect crystal reflectivity.

Lucia Alianelli

Abstract

REF.pro is a program written in IDL [1] for the calculation of reflected and transmitted neutron intensities by imperfect crystals, i.e. crystals with a distortion of the Bragg planes due to mosaicity, d-spacing gradient or bending.

The solutions that we have implemented in the code have a quite large application domain, in particular they are valid for any kind of mosaic distribution, with or without anisotropy. Moreover, any asymmetry angle of the Bragg planes can be considered. The resulting theoretical reflectivity contains, by definition, secondary extinction effects, but not primary extinction. However, the latter can be included by introducing a correction factor for the Q scattering factor. The data file written by REF.pro contains the crystal reflectivity as a function of the neutron K vector modulus and grazing angle. Hence it can be successfully used as input file for the McStas [2] component Monochromator_reflect [3]. The development of this code is the result of a work to better understand and describe imperfect crystals, and mosaic crystals *in primis*, as they are the most commonly used as neutron monochromators and analysers. In addition, as the performance of gradient crystals can be assessed with good accuracy as well, the REF.pro program, used together with Monochromator_reflect, can be a valid tool for simulating the function of novel gradient crystals in neutron scattering instruments.

The REF.pro program can be copied, corrected, extended or improved upon request to the author (email to alianell@ill.fr).

A translation of REF.pro to the C language and the interfacing to McStas is under development. Moreover, the program will be included in the NOP package [4] with a graphical user interface.

1 The theoretical background and the algorithm.

Theory and definitions.

A mosaic crystal can be seen as an agglomeration of perfect small crystallites. The coherence effects which can be observed in the diffraction by perfect crystals are lost in the case of mosaic crystals because of the randomness in the crystallite distribution. Primary extinction is a dynamical diffraction effect and represents the strong attenuation of the beam in a perfect crystal. Typical primary extinction depths t_{ext} are of the order of few μm . The phenomenon of the attenuation of the beam due to diffraction by a mosaic crystal is called secondary extinction. This attenuation is weaker than primary extinction in perfect crystals and involves larger crystal volumes (the typical penetration length due to secondary extinction is of order 1 cm). As a matter of fact, the presence of primary extinction competes with secondary extinction and therefore decreases the reflectivity of the mosaic crystal. The multiple Bragg reflections in a mosaic crystal and the concept of secondary extinction are summarised by the Darwin's equations [5, 6]. An exact and general solution of these equations has been given by Sears [7]. The physical quantities which govern diffraction by a mosaic crystal are the scattering coefficient σ and the attenuation coefficient μ . The Bragg scattering coefficient is written as

$$\sigma = Q \cdot W (\theta - \theta_B) \quad (1)$$

with the scattering factor Q given by:

$$Q = \lambda^3 F_{hkl}^2 / (V_0^2 \sin 2\theta_B) \quad (2)$$

In these equations λ represents the wavelength, F_{hkl} the structure factor, θ_B the Bragg angle and V_0 the unit cell volume. According to Zachariasen [5] the correction for primary extinction consists of a smaller scattering factor Q . We denote by t the perfect crystallite thickness and by t_{ext} the primary extinction depth. The actual Q is decreased by a factor $f(A)$, where $A = t/t_{ext}$. In Bragg geometry (see Fig. 1 for the definition of geometry) the correction to Q is:

$$f(A) = \frac{\tanh A + |\cos 2\theta_B| \tanh |A \cos 2\theta_B|}{A(1 + \cos^2 2\theta_B)} \quad (3)$$

If the thickness of the mosaic blocks t is much smaller than the primary extinction depth t_{ext} , the primary extinction corrections are not necessary. Otherwise, the actual Q is decreased by a factor $f(A)$, which can be used as input for REF.pro. We will give examples for the numerical values of $f(A)$ in section 4.

The mosaic crystal theory assumes that perfect crystallites are oriented almost parallel to the crystal surface (for the Bragg case) following a mosaic distribution $W(\theta - \theta_B)$, θ

being the angle formed by the incident beam and the Bragg planes. $W(\theta - \theta_B)$ is usually written as a Gaussian and the full-width-at-half-maximum η is called the intrinsic mosaic spread or intrinsic mosaicity. The case of anisotropic mosaic distributions is handled, in the REF.pro program, by using a probability distribution that depends also on the crystallite azimuthal angle ϕ :

$$W(\theta, \phi) = c \cdot e^{-4 \log 2 \left[\frac{\sin^2 \phi}{\eta_s^2} + \frac{\cos^2 \phi}{\eta_p^2} \right] (\theta - \theta_B)^2} \quad (4)$$

where η_s and η_p are the FWHMs in the scattering plane and perpendicular to it, respectively and c is a constant giving the correct normalization of W . Different shapes of W can of course be used, for example Lorentzian or pseudo-Voigt (i.e. a combination of Gaussian and Lorentzian). This option is not included, but can easily be implemented.

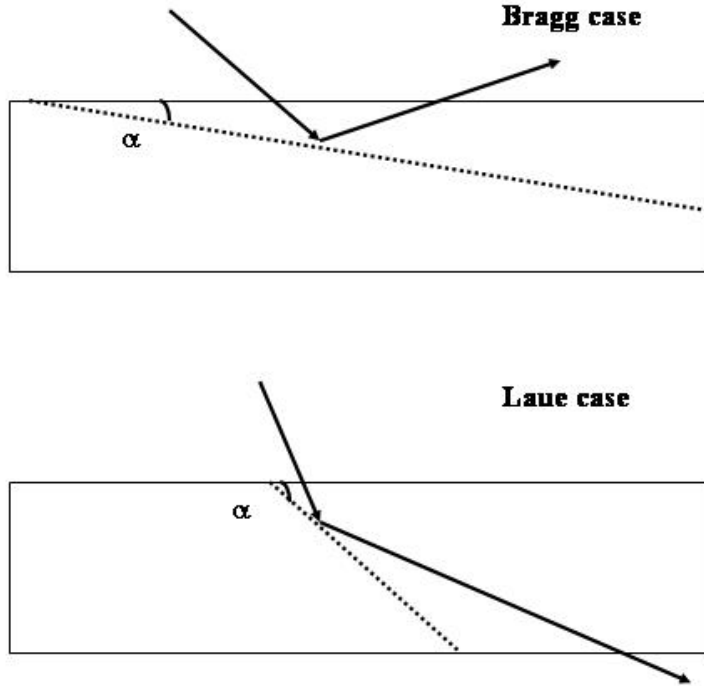


Figure 1: *Illustration of Bragg and Laue cases and definition of the asymmetry angle α . In the Bragg symmetric geometry the asymmetry angle is $\alpha = 0^\circ$. In the Laue symmetric geometry the asymmetry angle is $\alpha = 90^\circ$. The dotted lines represent Bragg planes.*

The attenuation coefficient μ is the sum of nuclear absorption, incoherent scattering, parasitic Bragg scattering, and thermal diffuse scattering (TDS):

$$\mu = \frac{n}{V_0} (\sigma_{abs} + \sigma_{inc} + \sigma_{par} + \sigma_{TDS}) \quad (5)$$

In the last equation n/V_0 is the number of atoms or molecules per unit cell volume.

The nuclear absorption cross-section σ_{abs} does not depend on the temperature and has a simple dependence on energy except for some well known resonant cases.

The parasitic Bragg scattering cross-section σ_{par} is more difficult to calculate. Simultaneous diffraction occurs when a single crystal is orientated in a neutron or x-ray monochromatic beam so that two, or more, sets of planes simultaneously satisfy Bragg's law. This can be observed experimentally when the crystal is orientated to diffract from a particular set of Bragg planes, and is rotated slowly around the the diffraction vector: changes in the intensity can be observed because Bragg's law can be simultaneously satisfied for a different set of planes [8]. In the case of mosaic crystals this effect is observed on a wavelength range larger than that of perfect crystals because of the presence of misoriented blocks. The consequences for the reflectivity of mosaic crystals are discussed in [9]. The calculation of σ_{par} is not included in REF.pro. However, an estimation can be assessed by using the NOP [4] module MAMON.

The TDS cross-section can be calculated as the sum of single phonon and multiple phonon cross-sections:

$$\sigma_{TDS} = \sigma_{single-ph} + \sigma_{multi-ph} \quad (6)$$

There are different approaches and approximations to the problem of TDS [10, 11, 12, 13]: all the authors, except for Binder [13] who gives the coherent cross-section for a polycrystal, use the incoherent approximation for the multi-phonon processes, i.e. they disregard all restrictions coming from the momentum conservation and give a non-zero partial differential cross-section in all scattering directions. This gives the correct result at energies much higher than the Debye energy. At lower energy the main contribution to σ_{TDS} is the single phonon scattering cross-section $\sigma_{single-ph}$ which can be calculated without approximations. At intermediate energies coherence effects, i.e. interference of scattering from different sites have to be considered, but they amount to only a few percent of the total cross-section [11]. We report here the formula given by Freund [14] for the total multi-phonon cross-section:

$$\sigma_{multi-ph} = \sigma_0 \{1 - \exp[-C_2 E (B_0 + B(T))]\} \quad (7)$$

In this equation σ_0 is the sum of the coherent and incoherent cross-sections for the nucleus, the C_2 parameter has to be determined experimentally for each material and $B_0 + B(T)$ is the mean square atomic displacement. The energy dependence of σ_{abs} and of $(\sigma_{abs} + \sigma_{TDS})$ for copper at 15 K and 290 K, calculated using Eq. (7) for the multi-phonon contribution, and with using the C_2 reported by Freund [14], is plotted in Fig. 2. The figure shows that TDS effect has to be included, even at low temperature.

If we define $a = \mu d / \sin \psi$ and $s = \sigma d / \sin \psi$, with d being the crystal thickness and ψ the angle formed by the incident beam and the surface, the Sears' equations [7] for the

reflected and transmitted beam in symmetric Laue (transmission) and Bragg (reflection) geometries are:

$$R_{Laue\ symm} = \frac{1}{2}e^{-a} (1 - e^{-2s}) \quad (8)$$

$$T_{Laue\ symm} = \frac{1}{2}e^{-a} (1 + e^{-2s}) \quad (9)$$

$$R_{Bragg\ symm} = \frac{s}{\sqrt{a(a+2s)} \coth \sqrt{a(a+2s)} + (a+s)} \quad (10)$$

$$T_{Bragg\ symm} = \frac{\sqrt{a(a+2s)}}{\sqrt{a(a+2s)} \cosh \sqrt{a(a+2s)} + (a+s) \sinh \sqrt{a(a+2s)}} \quad (11)$$

The model we have discussed is valid if the mosaicity is much larger than the Darwin width of the perfect crystal.

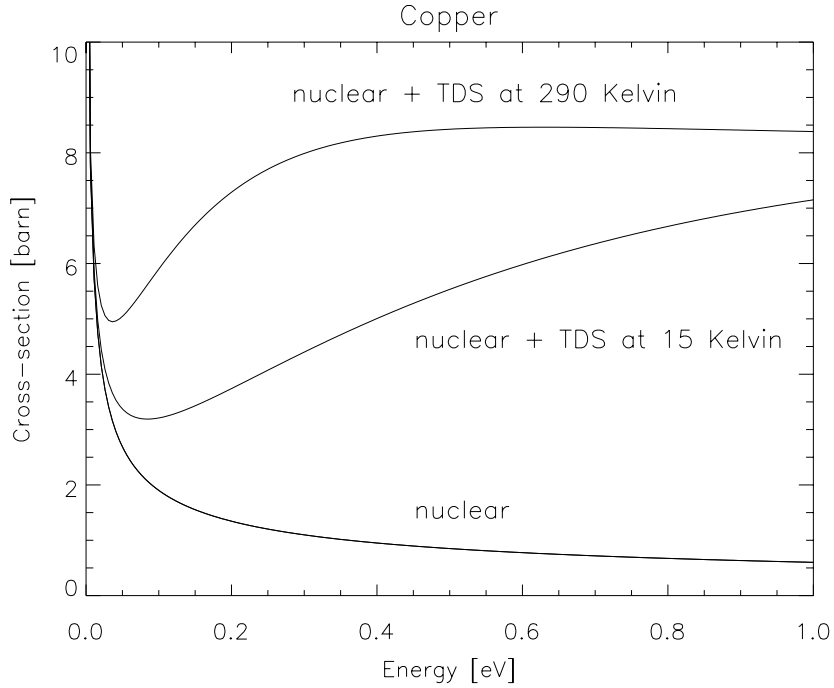


Figure 2: Total attenuation cross-sections $\sigma_{abs} + \sigma_{TDS}$ for copper. The line on the bottom represents the true absorption cross-section σ_{abs} . The temperature dependent σ_{TDS} was calculated using the model developed by Freund [14].

The algorithm.

The algorithm used in REF.pro is based on the layer-coupling model [15, 16], which can be considered to be the discrete form of the Darwin's equations. For a mosaic crystal without bending or d-spacing gradient, the model gives results that are equivalent to the analytical solution of the Darwin's equations [7]. If the mosaic crystal is also bent or has a d-spacing gradient, then the diffraction planes change orientation or d-spacing from one layer to the next. The solution is found by assuming the crystal to be ideally divided into regions, or layers, whose properties remain locally unchanged. The final crystal reflectivity is calculated with the following algorithm:

1) a set of optical matrices for the crystal is written. Each matrix contains the informations on the local absorption and scattering properties of the crystal layer. This means that a crystal with a d-spacing or mosaicity which depend on the position along its depth can be described. These quantities (i.e. mosaicity and d-spacing gradient) can have the desired behaviour, for example linear, quadratic or step-like.

2) the solution is found by recursively multiplying the matrices for the different layers and by applying the boundary conditions.

$P_{0,n}$ and $P_{H,n}$ are the power of the incident and diffracted beam at the n-th layer. In Bragg geometry the equations are the following:

$$\begin{aligned} P_{0,n} &= T_{0,n}P_{0,n-1} + D_{H,n}P_{H,n} \\ P_{H,n-1} &= D_{0,n}P_{0,n-1} + T_{H,n}P_{H,n} \end{aligned} \quad (12)$$

$D_{0/H,n}$ is the probability of the incident/reflected beam being diffracted and not being absorbed by the n-th layer.

$T_{0/H,n}$ is the probability of the beam being transmitted by the n-th layer.

The Darwin equations take the form:

$$\begin{pmatrix} P_{0,n} \\ P_{H,n} \end{pmatrix} = \begin{pmatrix} T_{0,n} - \frac{D_{0,n}D_{H,n}}{T_{H,n}} & \frac{D_{H,n}}{T_{H,n}} \\ -\frac{D_{0,n}}{T_{H,n}} & \frac{1}{T_{H,n}} \end{pmatrix} \begin{pmatrix} P_{0,n-1} \\ P_{H,n-1} \end{pmatrix} \quad (13)$$

The solution at the last layer is:

$$\begin{pmatrix} P_{0,N} \\ P_{H,N} \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} P_{0,0} \\ P_{H,0} \end{pmatrix} \quad (14)$$

where the M_{ij} matrix is obtained by iterating the Eq. (13). In the Bragg geometry we have $P_{H,N} = 0$, then the reflecting power is:

$$\frac{P_{H,0}}{P_{0,0}} = -\frac{M_{21}}{M_{22}} \quad (15)$$

The diffraction and transmission probabilities of the n-th layer can be written as:

$$\begin{aligned}
 D_{0,n} &= \frac{\sigma_n t_n}{\cos \theta_0} e^{-\frac{\mu t_n}{\cos \theta_0}} \\
 D_{H,n} &= \frac{\sigma_n t_n}{|\cos \theta_H|} e^{-\frac{\mu t_n}{|\cos \theta_H|}} \\
 T_{0,n} &= (1 - D_{0,n}) e^{-\frac{\mu t_n}{\cos \theta_0}} \\
 T_{H,n} &= (1 - D_{H,n}) e^{-\frac{\mu t_n}{|\cos \theta_H|}}
 \end{aligned} \tag{16}$$

t_n is the thickness of the n-th layer,

θ_0 is the angle of incidence (i.e. 90° - asymmetry angle $-\theta_B$),

θ_H is the angle of reflection,

$\sigma_n = Q \cdot W(\Delta\theta_n)$ is the equivalent of the secondary extinction coefficient for the n-th layer.

The deviation $\Delta\theta_n$ from the nominal Bragg angle θ_B , at the n-th layer, is:

$$\Delta\theta_n = \Delta\theta_1 \pm \epsilon_n \tag{17}$$

where $\Delta\theta_1$ is the deviation at the first layer. The + sign is for the bent crystal in Bragg geometry and for the gradient crystal in any geometry, the - sign for bent crystal in Laue geometry. The deviation ϵ_n depends on the type of deformation (bending or d-spacing gradient):

$$\epsilon_n = \begin{cases} \frac{1}{R} \sum_{i=1}^n t_i \left[\tan \theta_0 + (\sin^2 \chi - \nu \cos^2 \chi) \tan \theta_B + \frac{1}{2} (1 + \nu) \sin 2\chi \right] & \text{Bent} \\ \epsilon_n = -\tan \theta_B \frac{d_{hkl,n} - d_{hkl}}{d_{hkl}} & \text{Gradient} \end{cases} \tag{18}$$

with χ depending on the asymmetry angle α :

$$\chi = \begin{cases} \pi/2 - \alpha & \text{Laue case} \\ \alpha & \text{Bragg case} \end{cases} \tag{19}$$

In the last equations R is the bending radius, d_{hkl} the d-spacing, ν is the Poisson ratio (defined as the lateral contraction per unit breadth divided by the longitudinal extension per unit length, $\nu = 0.22$ for silicon and 0.27 for germanium) and α is the asymmetry angle as defined in Fig. 1. In the case of assembled wafer crystals, there can be an additional deviation δ_n from the nominal Bragg angle due to different mosaicities of the wafers. This effect can be included as well.

2 The input parameters.

The program can be executed by opening an IDL session and typing:

```
IDL> .run REF
```

The input parameters are:

1) Crystal material, Bragg indices, thickness.

The materials cross-sections database includes at the moment: beryllium, copper, lead, silicon, germanium, graphite, nickel, fluorite. Other materials can be included upon request.

2) Asymmetry angle α , i.e. angle between Bragg planes and crystal surface (clock wise positive, see Fig. 1).

3) Mosaicity in the scattering plane and perpendicular to the scattering plane.

4) Correction factor $f(t/t_{ext})$. Correcting Q corresponds to a correction of the primary extinction effect. If the correction factor is set equal to 1 then there is no correction for primary extinction (see section 4).

5) Average wavelength, wavelength range and number of points N for wavelength.

6) Angular range.

7) Type of deformation:

7.a) Bending: bending radius and Poisson ratio.

7.b) d-spacing gradient: total variation of the d-spacing over the entire crystal thickness.

REF.pro can be executed without an IDL licence as an XOP [17] macro. For Windows platforms, the user has to edit the program and insert the input parameters in it.

The transcription of the IDL session for computing the diffraction profile of Cu $\langle 220 \rangle$, with a gradient d-spacing (see the examples in section 3) is reported below.

IDL> .run REF

This program calculates the reflectivity of a mosaic gradient crystal or a mosaic bent crystal using the layer-coupling model.

PLEASE REFER TO L. Alianelli, A routine for the computation of imperfect crystal reflectivity, ILL technical report ILL03AL05T, 2003.

REFERENCES:

(1) H.C. Hu

J. Appl. Cryst. (1992) 25, 731-736

(2) H.C. Hu

J. Appl. Cryst. (1993) 26, 251-257

INDEX MATERIAL

0 Beryllium

1 Copper

3 Lead

4 Silicon

5 Germanium

6 Graphite

11 Nickel

12 CaF2

ENTER THE MATERIAL INDEX: 1

ENTER h Miller index: 2

ENTER k Miller index: 2

ENTER l Miller index: 0

ENTER THICKNESS (cm): 0.8

ENTER THE ASYMMETRY ANGLE (DEG): 0

ENTER MOSAICITY in the scattering plane (FWHM, DEGREES): .05

ENTER MOSAICITY in the plane perpendicular to scattering (FWHM, DEGREES): .05

ENTER correction for Q: 1

ENTER average wavelength (A): 1.8

ENTER wavelength range (A): 0

ENTER number of points for wavelength: 1

ENTER angular range (deg): 3

ENTER 0 FOR BENT, 1 FOR GRADIENT: 1

ENTER THE TOTAL VARIATION OF dhkl (A): .02

3 The program output. Examples

The results are written in an ascii file REF.dat with N blocks. Each block has four columns:

1st column: $K = 2\pi/\lambda$ in \AA^{-1} . K is constant within each block.

2nd column: grazing angle in degrees (automatically includes the asymmetry angle α as illustrated in Fig. 1).

3rd column: reflectivity in the scattering plane.

4th column: reflectivity perpendicular to the scattering plane.

An additional TRA.dat file is written by REF.pro: the columns 3 and 4 represent the transmitted (i.e. attenuated) intensity. Hence TRA.dat might be used for assessing the intensity of the beam that is transmitted by the crystal, and accounts for absorption, TDS and Bragg scattering. Typical examples of mosaic copper reflected and transmitted intensities, calculated with REF.pro, are reported in Fig. 3 (REF.dat data in the top, TRA.dat data in the bottom). One consequence of the Darwin's equations is clearly seen in the figures: anomalous absorption, which is not to be confused with the Borrmann anomalous absorption observed in perfect crystals, is present in Bragg geometry ($R + T \neq e^{-a}$, see also Eqs. (10) and (11)), but not in Laue geometry ($R + T = e^{-a}$, see also Eqs. (8) and (9)). The typical asymmetric shapes of the deformed (i.e. gradient or bent) crystals diffraction profiles are depicted in Fig. 4. The reflected intensity subtracted by primary extinction is larger for the gradient crystal case (d-spacing gradient of 1.6 % and total width of the diffraction profile $\sim 1^\circ$) than for the bent crystal case (bending radius of 5 m and total width of the diffraction profile $\sim 0.2^\circ$), because the deformation is stronger. A bent crystal with $R = 0.5$ m would have approximately the same diffraction profile as the gradient crystal in Fig. 4 (left) and the same reduction of reflectivity, due to primary extinction. In other words, with a given crystallite size t , the reduction of the peak reflectivity is the same if the reflectivity profile is the same, regardless of the kind of deformation (bending or gradient). The interesting question, in order to determine t , is how the production and deformation process affects the perfect crystallites size. It is clear that by controlling this parameter (i.e. producing imperfect crystals with t as small as possible) higher peak reflectivities can be obtained.

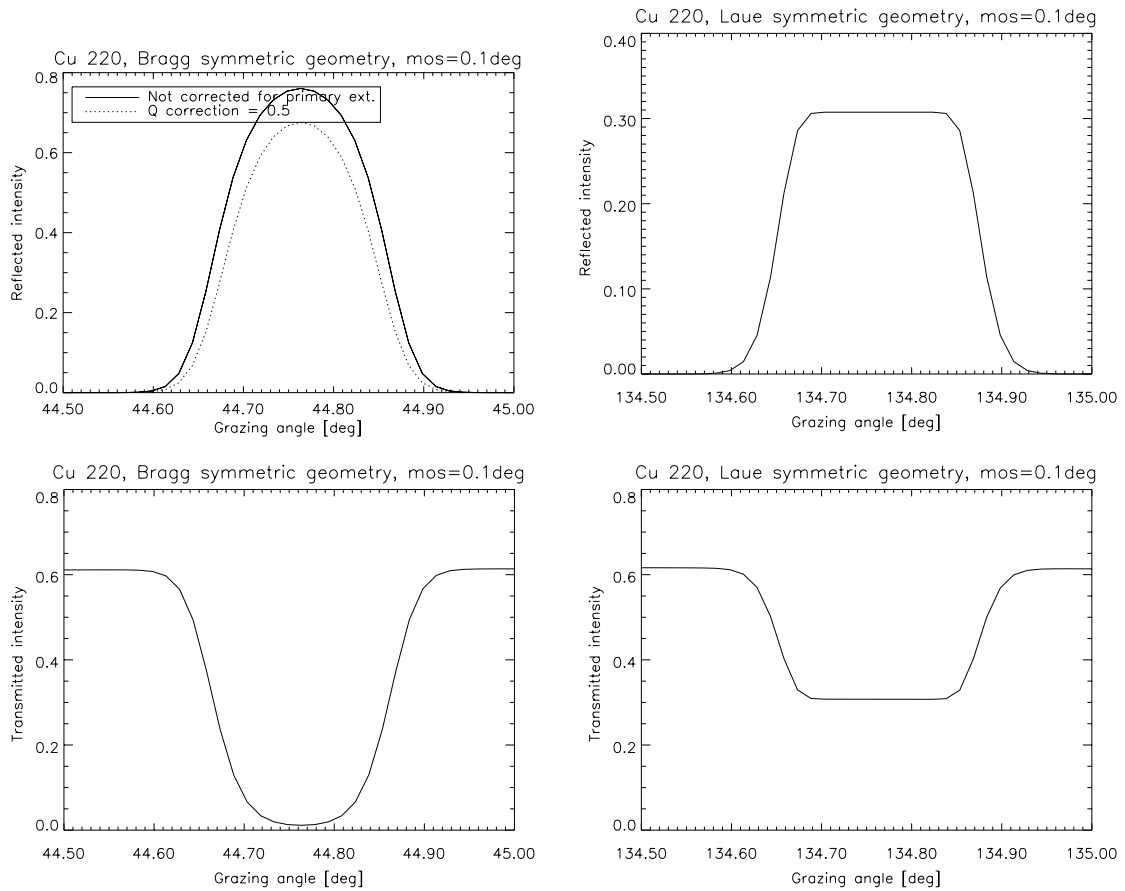


Figure 3: Imperfect crystal diffraction calculated with REF.pro: Cu $\langle 220 \rangle$, $\lambda = 1.8 \text{ \AA}$, $\eta = 0.1^\circ$, $d = 0.8 \text{ cm}$, Bragg (left) and Laue (right) symmetric geometry. TOP: reflectivity. BOTTOM: transmitted intensity.

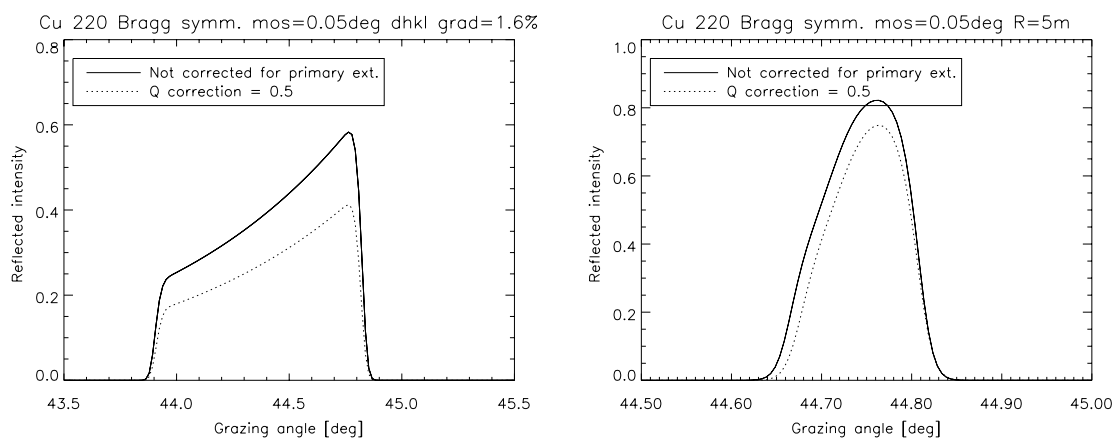


Figure 4: Imperfect crystal reflectivity calculated with REF.pro: Cu $\langle 220 \rangle$, $\lambda = 1.8 \text{ \AA}$, $\eta = 0.05^\circ$, $d = 0.8 \text{ cm}$, Bragg symmetric geometry. LEFT: gradient crystal with $\Delta d_{hkl}/d_{hkl} = 1.6 \%$. RIGHT: bent crystal with $R = 5 \text{ m}$.

4 Correcting for primary extinction.

The diffraction properties of imperfect crystals suffer from non-ideal behaviour. There can be several sources of imperfections: non-homogeneity of the mosaic structure and primary extinction are two examples. The first can be simulated using the Monte Carlo method [18, 19]. The latter is successfully described by the theory [5]. Examples of numerical values for correcting the Q factor, as a function of the crystallite size and for some Bragg angles, are reported in the Tables A, B, C for the case of mosaic copper, germanium and highly oriented pyrolytic graphite respectively.

Table A. *Mosaic copper: primary extinction corrections for some reflections and Bragg angles in Bragg symmetric geometry.*

hkl	t_{ext} [μm]	Crystallite size t [μm]	λ [\AA]	θ_{Bragg} [deg]	$f(t/t_{ext})$
111	3.7	5	1.05	14.6	0.67
			1.53	21.5	0.69
		10	1.05	14.6	0.39
			1.53	21.5	0.41
		15	1.05	14.6	0.26
			1.53	21.5	0.28
200	4.2	5	0.8	12.8	0.715
			1.4	22.8	0.74
		10	0.8	12.8	0.43
			1.4	22.8	0.46
		15	0.8	12.8	0.29
			1.4	22.8	0.32
220	6.0	5	0.4	9	0.825
			0.8	18.2	0.84
		10	0.4	9	0.57
			0.8	18.2	0.595
		15	0.4	9	0.40
			0.8	18.2	0.43

Table B. *Mosaic germanium: primary extinction corrections for some reflections and Bragg angles in Bragg symmetric geometry.*

hkl	t_{ext} [μm]	Crystallite size t [μm]	λ [\AA]	θ_{Bragg} [deg]	$f(t/t_{ext})$
111	6.0	5	1.9	16.9	0.84
		10	1.9	16.9	0.59
		15	1.9	16.9	0.42
311	11.5	5	1.28	22.0	0.95
			2.994	61.4	0.95
		10	1.28	22.0	0.83
			2.994	61.4	0.835
		15	1.28	22.0	0.70
			2.994	61.4	0.71
335	23	5	1.514	61.3	0.99
		10	1.514	61.3	0.95
		15	1.514	61.3	0.90

Table C. *Examples of primary extinction corrections for highly oriented pyrolytic graphite in Bragg symmetric geometry.*

hkl	t_{ext} [μm]	Crystallite size t [μm]	λ [\AA]	θ_{Bragg} [deg]	$f(t/t_{ext})$
002	2.0	1	4.4	41.1	0.93
		5	4.4	41.1	0.405
004	3.9	1	2.2	41.1	0.98
		5	2.2	41.1	0.67

The effect of primary extinction on the reflectivity of deformed crystals.

Fig. 5 clearly shows the importance of primary extinction on the reflectivity of a gradient crystal. The plotted intensities are the simulated values, divided by the source intensity, hence there is no scaling factor. They have been obtained using three completely different numerical methods: McStas with Monochromator_reflect and the REF.dat datafile (dotted and solid lines), a full Monte Carlo calculation (+ symbols), and the the RESTRAX code [20] (dashed line, data provided by J. Saroun). As the general agreement is very good, we can conclude that the methods used are detailed and realistic. We assume that the McStas data obtained by including primary extinction (solid line) represent the correct model (because the RESTRAX and full Monte Carlo data shown in the same figure do simulate the crystallite size, with different methods). The comparison shows that there is a discrepancy concerning the peak value of the diffraction profiles. The full Monte Carlo and the RESTRAX data are very similar. The McStas peak reflectivity (solid line) is smaller. A possible explanation of the disagreement could be the following: mosaicity

is very small, the crystal layers are almost perfect, and the one that is the closest to the crystal surface diffracts according to dynamical diffraction, i.e. with a peak reflectivity close to 1 (Monte Carlo and RESTRAX results). In the REF.dat data, instead (which are used in the simulations with Monochromator_reflect), the diffraction probabilities are, by definition, convoluted with mosaicity and are not able to show the effect of a possible presence of layers that are almost perfect. In other words, the disagreement could reflect a failure of the mosaic model, due to the very small mosaicity.

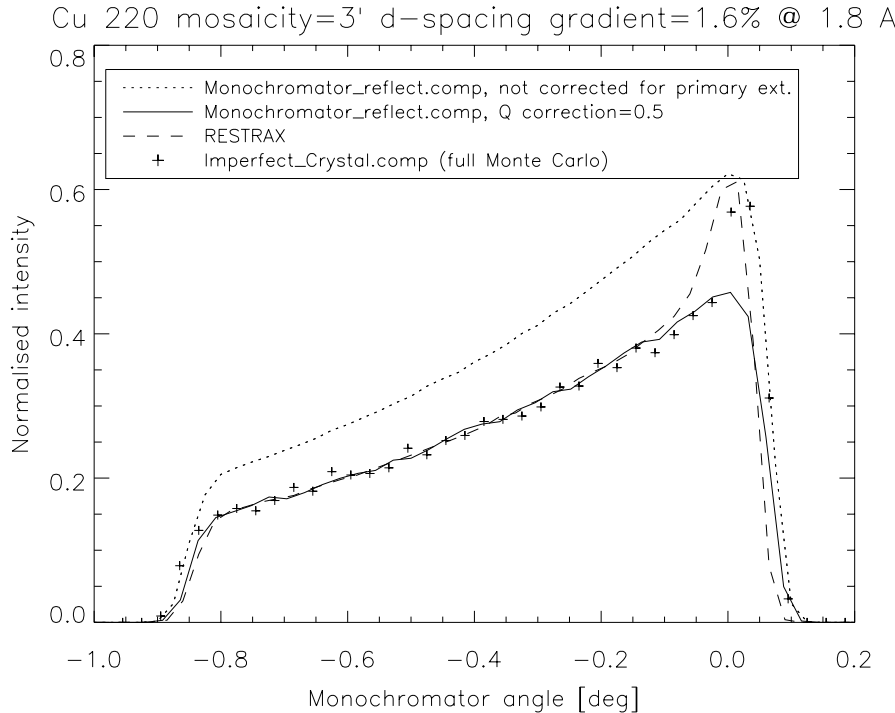


Figure 5: *Diffraction profiles of gradient Cu <220>, $\lambda = 1.8 \text{ \AA}$, $\eta = 3' = 0.05^\circ$, $\Delta d_{hkl}/d_{hkl} = 1.6 \%$, $d = 1 \text{ cm}$, Bragg symmetric geometry. The data have been obtained by using McStas with two new crystal modules Monochromator_reflect [3] (based on a reflectivity datafile created with REF.pro) and Imperfect_Crystal.comp (full Monte Carlo calculation, see [19]). The RESTRAX [20] data have been provided by J. Saroun.*

5 Conclusions

The REF.pro program is a valid tool for estimating the reflectivity of imperfect crystals commonly used as neutron monochromators and analysers. The main use of REF.pro at the moment is for creating datafiles being used by the McStas component Monochromator_reflect. These datafiles represent the crystal reflectivity calculated with the most detailed theories and cross-sections.

6 Acknowledgements

Acknowledgements are due to all the people who have helped with discussions, bibliographic help and ideas. Thanks to: M. Sánchez del Río for the continuity in scientific and computational support; to I. Anderson, R. Felici and K. Andersen for support and stimulating discussions; to B. Hamelin and P. Courtois for the frequent discussions about real imperfect crystals; to S. Ivanov for teaching me basic ideas about the importance of phonon scattering in the role of crystal monochromators; to J. Saroun for carrying out interesting comparisons with the RESTRAX code.

References

- [1] Interactive Data Language <http://www.rsinc.com>
- [2] <http://neutron.risoe.dk/mcstas/>
- [3] N. Wilson “McStas optical component Monochromator_reflect”, *ILL, stage report*, 2003.
- [4] L. Alianelli, M. Sánchez del Río and R. Felici, “NOP: a new software tool for neutron optics”, *accepted for publication by Physica B, Proceedings of the ECNS2003 conference*.
- [5] W.H. Zachariasen, “Theory of X-ray Diffraction in Crystals”, *Dover, New York*, 1945.
- [6] G.E. Bacon and R.D. Lowde, “Secondary Extinction and Neutron Crystallography”, *Acta Cryst.* **1**, pp. 303-314, 1948.
- [7] V.F. Sears, “Bragg Reflection in Mosaic Crystals. I. General solution of the Darwin Equations”, *Acta Cryst.* **A53**, pp. 35-45, 1997.
- [8] H. Cole, F.W. Chambers and H.M. Dunn, “Simultaneous Diffraction: Indexing Umweganregung Peaks in Simple Cases”, *Acta Cryst.* **15**, pp. 138-144, 1962.
- [9] D.A. O’Connor and J. Sownowski, “Parasitic multiple Bragg scattering” *Acta Cryst.* **14**, pp 292-297, 1961.

- [10] G. Placzek, “The scattering of Neutron by Systems of Heavy Nuclei”, *Phys. Rev.* **86**, pp. 377-388, 1952.
- [11] W. Marshall and S.W. Lovesey, “Theory of thermal neutron scattering”, *Clarendon Press, Oxford*, 1971.
- [12] V.F. Turchin , “Slow neutrons”, *Israel Program for Scientific Translations*, 1965.
- [13] K. Binder, “Total Coherent Cross Sections for the Scattering of Neutrons from Crystals”, *Phys. Stat. Sol.* **41**, pp. 767-779, 1970.
- [14] A.K. Freund, “Cross-sections of materials used as neutron monochromators and filters”, *Nucl. Instr. and Meth.* **213**, pp. 495-501, 1983.
- [15] Hua-Chen Hu, “A layer-coupling model for neutron reflectivity and transmissivity calculation in deformed mosaic crystals”, *J. Appl. Cryst.* **25**, pp. 731-736, 1992.
- [16] Hua-Chen Hu, “Neutron diffraction in flat and bent mosaic crystals for asymmetric geometry”, *J. Appl. Cryst.* **26**, pp. 251-257, 1993.
- [17] <http://www.esrf.fr/computing/scientific/xop2.1/>
- [18] L. Alianelli, “Characterization and modelling of imperfect crystals for thermal neutron diffraction”, *PhD Thesis, Université Joseph Fourier, Grenoble, France*, 2002, published by *dissertation.de*.
- [19] L. Alianelli, M. Sánchez del Río, R. Felici, K. Andersen and E. Farhi, “A novel Monte Carlo algorithm for simulating crystals with McStas”, *accepted for publication by Physica B, Proceedings of the ECNS2003 Conference*.
- [20] J. Saroun and J. Kulda, “RESTRAX - a program for TAS resolution calculation and scan profile simulation”, *Physica B: Condensed Matter* **234-236**, pp. 1102-1104, 1997.

A Appendix - REF.pro source file

```

print,'
PRINT,'This program calculates the reflectivity of a mosaic gradient crystal'
print,'or a mosaic bent crystal using the layer-coupling model.'
print,'
print,'PLEASE REFER TO L. Alianelli, A routine for the computation of '
print,'imperfect crystal reflectivity, ILL technical report ILL03AL05T, 2003.'
print,'
PRINT,'REFERENCES:'
print,'
PRINT,'(1) H.C. Hu'
print,'J. Appl. Cryst. (1992) 25, 731-736'
print,'
PRINT,'(2) H.C. Hu'
print,'J. Appl. Cryst. (1993) 26, 251-257'
print,'

print,'
print,'          *****'
print,'          INDEX          MATERIAL
print,'          0          Beryllium
print,'          1          Copper
;print,'          2          Niobium
print,'          3          Lead
print,'          4          Silicon
print,'          5          Germanium
print,'          6          Graphite
;print,'          7          Bismuth
;print,'          8          Quartz
;print,'          9          Sapphire
;print,'          10         MgF2
print,'          11         Nickel
print,'          12         CaF2
print,'          *****'
print,'
read,i,prompt='ENTER THE MATERIAL INDEX: '
print,'
read,h,prompt='ENTER h Miller index: '

```

```

print,'
read,k,prompt='ENTER k Miller index: '
print,'
read,l,prompt='ENTER l Miller index: '
print,'
read,d,prompt='ENTER THICKNESS (cm): '
print,'
N_lay=500.*d
;read,E,prompt='ENTER energy (eV): '
;print,'
delmu=0.0
;read,delmu,prompt='enter delta mu (cm-1)''
read,alpha,prompt='ENTER THE ASYMMETRY ANGLE (DEG): '
print,'
;read,T,prompt='ENTER TEMPERATURE (K) : '
;print,'
T = 298.
read,tau_z,prompt='ENTER MOSAICITY in the scattering plane (FWHM, DEGREES): '
print,'
read,tau_y,prompt='ENTER MOSAICITY in the plane perpendicular to scattering (FWHM, DE
print,'
read,corr,prompt='ENTER correction for Q: '
print,'
read,lambda0,prompt='ENTER average wavelength (A): '
print,'
read,delta_lambda,prompt='ENTER wavelength range (A): '
print,'
lambda_p = 1L
read,lambda_p,prompt='ENTER number of points for wavelength: '
print,'
read,range,prompt='ENTER angular range (deg): '
print,'

k_bol = 0.86174D-4 ;Boltzmann constant eV K-1
E0 = 0.0818D ;Conversion factor: energy (eV) at 1 Angstrom
PI = 3.14159265358979323846
tp = 2.D*PI
DEG2RAD = (PI/180.)
alpha = alpha*DEG2RAD ;Angle between Bragg planes and crystal surface
GEOM = 0.

```

```
DEF = 0.
R_curv = 0.
delta_d_hkl = 0.
Poisson = 0.
chi = 0.
d = d*1.D+8 ;Angstrom
di = DblArr(N_lay)+d/N_lay ;layer thickness (A)
dth = DblArr(N_lay)
alpha1_p = 201
delta = DblArr(alpha1_p,N_lay)
W_z = DblArr(alpha1_p,N_lay)
W_y = DblArr(alpha1_p,N_lay)
sigma_bragg_z = DblArr(alpha1_p,N_lay)
sigma_bragg_y = DblArr(alpha1_p,N_lay)
D_0_z = DblArr(alpha1_p,N_lay)
D_0_y = DblArr(alpha1_p,N_lay)
D_H_z = DblArr(alpha1_p,N_lay)
D_H_y = DblArr(alpha1_p,N_lay)
Da_0_z = DblArr(alpha1_p,N_lay)
Da_0_y = DblArr(alpha1_p,N_lay)
Da_H_z = DblArr(alpha1_p,N_lay)
Da_H_y = DblArr(alpha1_p,N_lay)
Ta_0_z = DblArr(alpha1_p,N_lay)
Ta_0_y = DblArr(alpha1_p,N_lay)
Ta_H_z = DblArr(alpha1_p,N_lay)
Ta_H_y = DblArr(alpha1_p,N_lay)
MATR_z = DblArr(alpha1_p,N_lay,2,2)
MATR_y = DblArr(alpha1_p,N_lay,2,2)
M_z = DblArr(2,2)
M_y = DblArr(2,2)
Ma1_z = DblArr(2,2)
Ma1_y = DblArr(2,2)
REF_z = DblArr(alpha1_p)
REF_y = DblArr(alpha1_p)
TRA_z = DblArr(alpha1_p)
TRA_y = DblArr(alpha1_p)
corrTRA_z = DblArr(alpha1_p)
corrTRA_y = DblArr(alpha1_p)
ddd_z = Dblarr(N_lay)
ddd_y = Dblarr(N_lay)
```

```

ddd_z=FindGen(N_lay)*di(0)+di(0)
ddd_y=FindGen(N_lay)*di(0)+di(0)
sigma_s_z = Dblarr(alpha1_p)
sigma_s_y = Dblarr(alpha1_p)

read,DEF,prompt='ENTER 0 FOR BENT, 1 FOR GRADIENT: '
IF (DEF EQ 0.) THEN read,R_curv,prompt='ENTER THE RADIUS OF CURVATURE (m): '
IF (DEF EQ 0.) THEN read,poisson,prompt='ENTER THE POISSON RATIO: '
IF (DEF EQ 1.) THEN read,delta_d_hkl,prompt='ENTER THE TOTAL VARIATION OF d_hkl (A):

tau_z = tau_z*DEG2RAD ;Mosaicity in rad, FWHM
tau_y = tau_y*DEG2RAD ;Mosaicity in rad, FWHM
h=h*1.D
k=k*1.D
l=l*1.D

;angular range
range = range*DEG2RAD ;angular range in rad
alpha1_max = range/2.
alpha1 = FindGen(alpha1_p)*2.*alpha1_max/(alpha1_p-1.)-alpha1_max
peak = where(abs(alpha1) EQ min(abs(alpha1)))

coh_sig = DblArr(13)
inc_sig = DblArr(13)
sig_abs0 = DblArr(13)
A_mass = DblArr(13)
T_deb = DblArr(13)
n_over_V = DblArr(13)
V0 = DblArr(13)
C2 = DblArr(13)
b_c = DblArr(13)
F_hkl0 = DblArr(13)
REF_hkl0 = DblArr(13)
IMF_hkl0 = DblArr(13)
a_cell = DblArr(13)
b_cell = DblArr(13)
c_cell = DblArr(13)
alpha_cell = DblArr(13)
beta_cell = DblArr(13)

```

```

gamma_cell = DblArr(13)
d_spacing = DblArr(13)

;*****Materials parameters*****
; 1 Beryllium
; Structure: Hexagonal

coh_sig(0) = 7.63D
inc_sig(0) = 0.0018D
sig_abs0(0) = 0.0056D
A_mass(0) = 9.01D
T_deb(0) = 1100.D
C2(0) = 7.62D
a_cell(0) = 2.2866D ;cell parameter (A)
b_cell(0) = 2.2866D
c_cell(0) = 3.5833D
alpha_cell(0) = 90.*DEG2RAD
beta_cell(0) = 90.*DEG2RAD
gamma_cell(0) = 120.*DEG2RAD
d_spacing(0) = 1./sqrt(4./3.*(h*h+h*k+k*k)/a_cell(0)+1*1/c_cell(0))
V0(0)=a_cell(0)^2*c_cell(0)*sin(60.*DEG2RAD)
;n_over_V(0) = 0.1234D ;Number of atoms per unit cell volume (A-3)
n_over_V(0) = 2./V0(0)
b_c(0) = 1.D-4*SQRT(coh_sig(0)/(4.D*PI)) ;bound coherent scattering in A
x1=h*.333333+k*.666667+l*0.25
x2=h*.666667+k*.333333+l*0.75
REF_hk10(0) = b_c(0)*(cos(tp*x1)+cos(tp*x2))
IMF_hk10(0) = b_c(0)*(sin(tp*x1)+sin(tp*x2))
F_hk10(0) = SQRT(REF_hk10(0)^2.+IMF_hk10(0)^2.)

; 2 Cu
; Structure: Cubic

coh_sig(1) = 7.485D ;COPPER cross sections
inc_sig(1) = 0.55D ;barn
sig_abs0(1) = 2.094D ;True absorption at 1 Angstrom
A_mass(1) = 63.54D ;Mass in 12C units
T_deb(1) = 300.D ;Debye temperature
C2(1) = 12.D ;C2 parameter (A-2 eV-1), see ref 1)

```

```

a_cell(1) = 3.61496D ;cell parameter (A)
b_cell(1) = 3.61496D
c_cell(1) = 3.61496D
alpha_cell(1) = 90.*DEG2RAD
beta_cell(1) = 90.*DEG2RAD
gamma_cell(1) = 90.*DEG2RAD
d_spacing(1) = a_cell(1)/sqrt(h*h+k*k+l*l)
;n_over_V(1) = 0.0847D ;Number of atoms per unit cell volume (A-3)
V0(1) = a_cell(1)^3.
n_over_V(1) = 4./V0(1)
b_c(1) = 1.D-4*SQRT(coh_sig(1)/(4.D*PI)) ;bound cohe scatter length in A
x1=.0
x2=k*.5+l*.5
x3=h*.5+l*.5
x4=h*.5+k*.5
REF_hkl0(1) = b_c(1)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4))
IMF_hkl0(1) = b_c(1)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4))
F_hkl0(1) = SQRT(REF_hkl0(1)^2.+IMF_hkl0(1)^2.)

; 3 Niobium
;
;coh_sig(2) = 6.253D
;inc_sig(2) = 0.0024D
;sig_abs0(2) = 0.6390D
;A_mass(2) = 92.91D
;T_deb(2) = 280.D
;n_over_V(2) = 0.0556D
;C2(2) = 23.50D
;d_cell(2) = 3.3004D ;cell parameter (A)
;b_c = 1.D-4*SQRT(coh_sig/(4.D*PI)) ;bound coherent scattering length in A

; 4 Lead
; Structure: Cubic

coh_sig(3) = 11.115D
inc_sig(3) = 0.003D
sig_abs0(3) = 0.0928D
A_mass(3) = 207.21D
T_deb(3) = 280.D
C2(3) = 150.D

```

```

a_cell(3) = 4.9508D ;cell parameter (A)
b_cell(3) = 4.9508D
c_cell(3) = 4.9508D
alpha_cell(3) = 90.*DEG2RAD
beta_cell(3) = 90.*DEG2RAD
gamma_cell(3) = 90.*DEG2RAD
V0(3) = a_cell(3)^3.
n_over_V(3) = 4./V0(3)
;n_over_V(3) = 0.0323D
b_c(3) = 1.D-4*SQRT(coh_sig(3)/(4.D*PI)) ;bound coher scatt length in A
x1=.0
x2=k*.5+l*.5
x3=h*.5+l*.5
x4=h*.5+k*.5D
REF_hk10(3) = b_c(3)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4))
IMF_hk10(3) = b_c(3)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4))
F_hk10(3) = SQRT(REF_hk10(3)^2.+IMF_hk10(3)^2.)

; 5 Silicon
; Structure: Cubic

coh_sig(4) = 2.163D
inc_sig(4) = 0.004D
sig_abs0(4) = 0.0889D ;Freund's paper, printing mistake?
sig_abs0(4) = 0.0951D ;Neutron news
A_mass(4) = 28.09D
T_deb(4) = 420.D
C2(4) = 6.36D
a_cell(4) = 5.43070D ;cell parameter (A)
b_cell(4) = 5.43070D
c_cell(4) = 5.43070D
alpha_cell(4) = 90.*DEG2RAD
beta_cell(4) = 90.*DEG2RAD
gamma_cell(4) = 90.*DEG2RAD
d_spacing(4) = a_cell(4)/sqrt(h*h+k*k+l*l)
V0(4) = a_cell(4)^3.
n_over_V(4) = 8./V0(4)
;n_over_V(4) = 0.0499D
b_c(4) = 1.D-4*SQRT(coh_sig(4)/(4.D*PI)) ;bound cohe scatte length in A
x1=.0

```

```

x2=k*.5+l*.5
x3=h*.5+l*.5
x4=h*.5+k*.5
x5=h*.25+k*.25+l*.25
x6=h*.25+k*.75+l*.75
x7=h*.75+k*.25+l*.75
x8=h*.75+k*.75+l*.25
REF_hk10(4) = b_c(4)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4)+cos(tp*x5)+cos(tp*x6)+cos(tp*x7)+cos(tp*x8))
IMF_hk10(4) = b_c(4)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4)+sin(tp*x5)+sin(tp*x6)+sin(tp*x7)+sin(tp*x8))
F_hk10(4) = SQRT(REF_hk10(4)^2.+IMF_hk10(4)^2.)

; 6 Germanium
; Structure: Cubic

coh_sig(5) = 8.42D
inc_sig(5) = 0.18D
sig_abs0(5) = 1.2160D
A_mass(5) = 72.60D
T_deb(5) = 290.D
C2(5) = 9.00D
a_cell(5) = 5.65735D ;cell parameter (A)
b_cell(5) = 5.65735D
c_cell(5) = 5.65735D
alpha_cell(5) = 90.*DEG2RAD
beta_cell(5) = 90.*DEG2RAD
gamma_cell(5) = 90.*DEG2RAD
d_spacing(5) = a_cell(5)/sqrt(h*h+k*k+l*l)
V0(5) = a_cell(5)^3.
n_over_V(5) = 8./V0(5)
;n_over_V(5) = 0.0442D
b_c(5) = 1.D-4*SQRT(coh_sig(5)/(4.D*PI)) ;bound coher scat length in A
x1=.0
x2=k*.5+l*.5
x3=h*.5+l*.5
x4=h*.5+k*.5
x5=h*.25+k*.25+l*.25
x6=h*.25+k*.75+l*.75
x7=h*.75+k*.25+l*.75
x8=h*.75+k*.75+l*.25
REF_hk10(5) = b_c(5)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4)+cos(tp*x5)+cos(tp*x6)+cos(tp*x7)+cos(tp*x8))

```



```
IMF_hk10(5) = b_c(5)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4)+sin(tp*x5)+sin(tp*x6))
F_hk10(5) = SQRT(REF_hk10(5)^2.+IMF_hk10(5)^2.)
```

```
; 7 Graphite
; Structure: Hexagonal
```

```
coh_sig(6) = 5.551D
inc_sig(6) = 0.001D
sig_abs0(6) = 0.0018D
A_mass(6) = 12.D
T_deb(6) = 1050.D
C2(6) = 20.D
a_cell(6) = 2.456D ;cell parameter (A)
b_cell(6) = 2.456D
c_cell(6) = 6.696D
alpha_cell(6) = 90.*DEG2RAD
beta_cell(6) = 90.*DEG2RAD
gamma_cell(6) = 120.*DEG2RAD
d_spacing(6) = 1./sqrt(4./3.*(h*h+h*k+k*k)/(a_cell(6))^2.+1*1/(c_cell(6))^2.)
V0(6)=a_cell(6)^2*c_cell(6)*sin(60.*DEG2RAD)
n_over_V(6) = 4./V0(6)
;n_over_V(6) = 0.1136D
b_c(6) = 1.D-4*SQRT(coh_sig(6)/(4.D*PI)) ;bound co scatt length in A
x1=.0
x2=1*.5
x3=h*.333333+k*.666667
x4=h*.666667+k*.333333+1*0.5
REF_hk10(6) = b_c(6)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4))
IMF_hk10(6) = b_c(6)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4))
F_hk10(6) = SQRT(REF_hk10(6)^2.+IMF_hk10(6)^2.)
```

```
; 8 Bismuth (uncomplete)
```

```
;
;coh_sig(7) = 9.148D
;inc_sig(7) = 0.0084D
;sig_abs0(7) = 0.0200D
;A_mass(7) = 209.D
;T_deb(7) = 300.D
;n_over_V(7) = 0.0323D
;C2(7) = 110.D
```

```
;d_cell(7) = 1.D ;cell parameter (A)
;b_c = 1.D-4*SQRT(coh_sig/(4.D*PI)) ;bound coherent scattering length in A
;

; 9 Quartz (uncomplete)
;
;coh_sig(8) =
;inc_sig(8) =
;sig_abs0(8) = 0.0889D
;A_mass(8) = 20.03D
;T_deb(8) = 500.D
;n_over_V(8) = 0.0266D
;C2(8) = 8.5D
;d_cell(8) = 1.D ;cell parameter (A)
;b_c = 1.D-4*SQRT(coh_sig/(4.D*PI)) ;bound coherent scattering length in A
;
;
; 10 Sapphire (uncomplete)
;
;coh_sig(9) = .0D
;inc_sig(9) = .0D
;sig_abs0(9) = 0.2556D
;A_mass(9) = 20.39D
;T_deb(9) = 1040.D
;n_over_V(9) = 0.0236D
;C2(9) = 4.60D
;d_cell(9) = 1.D ;cell parameter (A)
;b_c = 1.D-4*SQRT(coh_sig/(4.D*PI)) ;bound coherent scattering length in A
;
; 11 MgF2 (uncomplete)
; Structure: Tetragonal
;
;coh_sig(10) = .
;inc_sig(10) =
;sig_abs0(10) = 0.0461D
;A_mass(10) = 20.77D
;T_deb(10) = 440.D
;n_over_V(10) = 0.0298D
;C2(10) = 6.D
;d_cell(10) = 1.D ;cell parameter (A)
```

```
;b_c = 1.D-4*SQRT(coh_sig/(4.D*PI)) ;bound coherent scattering length in A
;

; 12 Nickel
; Structure: Cubic

coh_sig(11) = 13.3D
inc_sig(11)= 5.2D
sig_abs0(11)= 2.4972D
A_mass(11) = 58.69D
T_deb(11) = 400.D
C2(11) = 11.176D
a_cell(11) = 3.524D ;cell parameter (A)
b_cell(11) = 3.524D
c_cell(11) = 3.524D
alpha_cell(11) = 90.*DEG2RAD
beta_cell(11) = 90.*DEG2RAD
gamma_cell(11) = 90.*DEG2RAD
d_spacing(11) = a_cell(11)/sqrt(h*h+k*k+l*l)
V0(11) = a_cell(11)^3.
n_over_V(11) = 4./V0(11)
;n_over_V(11) = 0.0914D
b_c(11) = 1.D-4*SQRT(coh_sig(11)/(4.D*PI))
x1=.0
x2=k*.5+l*.5
x3=h*.5+l*.5
x4=h*.5+k*.5
REF_hkl0(11) = b_c(11)*(cos(tp*x1)+cos(tp*x2)+cos(tp*x3)+cos(tp*x4))
IMF_hkl0(11) = b_c(11)*(sin(tp*x1)+sin(tp*x2)+sin(tp*x3)+sin(tp*x4))
F_hkl0(11) = SQRT(REF_hkl0(11)^2.+IMF_hkl0(11)^2.)

; 13 Fluorite CaF2
; Structure: Cubic

Fcoh_sig = 4.017
Finc_sig = 0.0008
Cacoh_sig = 2.78
Cainc_sig = 0.05
coh_sig(12) = 10.8140
```

```

inc_sig(12) = 0.0516
sig_abs0(12) = 0.249833
A_mass(12) = 10000000000000.0      ;SEARCH DATA
T_deb(12) = 400.D
C2(12) = 0.0      ;SEARCH DATA
a_cell(12) = 5.46295D*(1.+1.9406D-5*(T-298.)+1.8744D-8*(T-298.)^2.) ;cell parameter (
b_cell(12) = 5.46295D*(1.+1.9406D-5*(T-298.)+1.8744D-8*(T-298.)^2.) ;cell parameter (
c_cell(12) = 5.46295D*(1.+1.9406D-5*(T-298.)+1.8744D-8*(T-298.)^2.) ;cell parameter (
alpha_cell(12) = 90.*DEG2RAD
beta_cell(12) = 90.*DEG2RAD
gamma_cell(12) = 90.*DEG2RAD
d_spacing(12) = a_cell(12)/sqrt(h*h+k*k+l*l)
V0(12) = a_cell(12)^3.
n_over_V(12) = 12./V0(12)
Cab_c = 1.D-4*SQRT(Cacoh_sig/(4.D*PI)) ;bound coher scat length in A
Fb_c = 1.D-4*SQRT(Fcoh_sig/(4.D*PI)) ;bound coher scat length in A
b_c(12) = 1.D-4*SQRT(coh_sig(12)/(4.D*PI)) ;bound coher scat length in A
x1=.0
x2=h*.25+k*.25+l*.25
x3=h*.25+k*.75+l*.25
REF_hkl0(12) = Cab_c*cos(tp*x1)+Fb_c*(cos(tp*x2)+cos(tp*x3))
IMF_hkl0(12) = Cab_c*sin(tp*x1)+Fb_c*(sin(tp*x2)+sin(tp*x3))
F_hkl0(12) = 4.D*SQRT(REF_hkl0(12)^2.+IMF_hkl0(12)^2.)

coh_sig = coh_sig(i)
inc_sig = inc_sig(i)
sig_abs0 = sig_abs0(i)
A_mass = A_mass(i)
T_deb = T_deb(i)
n_over_V = n_over_V(i)
V0 = V0(i)
C2 = C2(i)
b_c = b_c(i)
F_hkl0 = F_hkl0(i)
d_hkl = d_spacing(i)

;***** Freund, NIM (1983):*****
B_zero = 2873./(T_deb*A_mass)
sig_bat = coh_sig+inc_sig

```

```

X = T_deb/T
y = DblArr(5001)
fact = dblArr(23)
fact(0) = 1.
fact(1) = 1.
fact(2) = 2.
fact(4) = 4.*3.*fact(2)
fact(6) = 6.*5.*fact(4)
fact(8) = 8.*7.*fact(6)
fact(10) = 10.*9.*fact(8)
fact(12) = 12.*11.*fact(10)
fact(14) = 14.*13.*fact(12)
fact(16) = 16.*15.*fact(14)
fact(18) = 18.*17.*fact(16)
fact(20) = 20.*19.*fact(18)
fact(22) = 22.*21.*fact(20)
be = DblArr(23)
rr = DblArr(23)
y = FIndGen(5001)*x/5000.
y(0) = 0.001*y(1)
fy = y/(exp(y)-1.)
phix = (INT_TABULATED(y,fy,/double))/x
IF (x GE 7.0) THEN phix = 1.642/x
B_T = 4.*B_zero*phix/x
DW = exp(-(B_zero+B_T)/(2.D*d_hkl)^2.) ;Debye Waller factor
be(0) = 1.
rr(0) = be(0)*x^(-1.)/(fact(0)*(5./2.))
be(1) = -0.5
rr(1) = be(1)*x^(1.-1.)/(fact(1.)*(1.+5./2.))
FOR ii = 1,11 DO BEGIN
  sum = 0.0D
  FOR mm = 1,100 DO BEGIN
    sum = sum + 1.0/mm^(2.0*ii)
  ENDFOR
  be(2*ii) = (-1.)^(ii-1.)*2.*fact(2*ii)/(2.*PI)^(2.*ii)*sum ;Bernoulli numbers
  rr(2*ii) = be(2*ii)*x^(2.*ii-1.)/(fact(2*ii)*(2.*ii+5./2.))
  if (2*ii eq 22) then rr(2*ii) = rr(2*ii)/2.
ENDFOR
r_sum = 0.D
r_sum = total(rr)

```

```

F_hkl = F_hkl0*DW ;structure factor

;*****
; open datafiles for writing
openw,lunR,'REF.dat',/get_lun
openw,lunT,'TRA.dat',/get_lun
;printf,lunR,alpha1_p, alpha1_p*lambda_p+1, 0, 0
;*****

theta_bragg0 = ASIN(lambda0/(2.D*d_hkl)) ;rad
IF (theta_bragg0 LT alpha) THEN BEGIN ;LAUE
  GEOM = -1.
  index = N_lay-1
ENDIF ELSE BEGIN
  GEOM = 1. ;Bragg
  index = 0
ENDELSE
chi = PI/2.*(1.-GEOM)/2.+GEOM^3.*alpha

lambda_min = lambda0 - delta_lambda/2.
lambda_max = lambda0 + delta_lambda/2.
dlam = delta_lambda/(lambda_p-1.)

if (lambda_p eq 1) then dlam=0.0

;Start lambda loop
FOR kk = 0,lambda_p-1 DO BEGIN
  lambda = lambda_max-kk*dlam
  E = E0/lambda/lambda
  theta_bragg = ASIN(lambda/(2.D*d_hkl)) ;rad
  sig_abs = sig_abs0*lambda
  phi = alpha + theta_bragg+alpha1 ;See Fig.1 of Sears
  phi_p = abs(alpha - theta_bragg-alpha1)
  csi = SIN(phi)/SIN(phi_p)

;***** Freund, NIM (1983):*****
sig_sph =sig_bat/(36.*A_mass)*sqrt(k_bol*T_deb/E)*3.3/x^(7./2.)
IF (x LE 6.) THEN sig_sph=sig_bat /((36.*A_mass)*sqrt(k_bol*T_deb/E)*r_sum
sig_mph=(A_mass/(A_mass+1.))^2.*sig_bat*(1.-exp(-(B_zero+B_T)*C2*E))

```

```

;*****

QQ = corr*(lambda)^3.D*F_hkl^2./V0^2./SIN(2.D*theta_bragg)
QQzero = corr*(lambda)^3.D*F_hkl0^2./V0^2./SIN(2.D*theta_bragg)
sig_tds = sig_sph +sig_mph
sig_tot = inc_sig+sig_tds + sig_abs
mu = n_over_V*sig_tot+delmu ;cm-1

;***** here deformed crystal calculation*****

theta_0 = PI/2.-alpha-theta_bragg
theta_H = PI/2.+GEOM*abs(alpha-theta_bragg)
b_asymm = COS(theta_0)/COS(theta_H)
a_0 = DblArr(N_lay)+mu*1.D-8*di/COS(theta_0)
a_H = DblArr(N_lay)+mu*1.D-8*di/abs(COS(theta_H))
ex_a_0 = EXP(-a_0)
ex_a_H = EXP(-a_H)

IF (DEF EQ 0.) THEN BEGIN
  csi_g = TAN(theta_0)
  csi_d = (SIN(chi)*SIN(chi)-poisson*COS(chi)*COS(chi))*TAN(theta_bragg)+0.5*(1.+pois
  epsilon = (FindGen(N_lay)+1.)*di/(R_curv*1.D+10)*(csi_g+csi_d)
  FOR aa=0,alpha1_p-1 DO BEGIN
    delta(aa,*) = alpha1(aa)+GEOM*epsilon(*)
  ENDFOR
ENDIF ELSE BEGIN
  dth = FindGen(N_lay)*delta_d_hkl/(N_lay-1.)/d_hkl*TAN(theta_bragg)
  FOR aa=0,alpha1_p-1 DO BEGIN
    delta(aa,*) = alpha1(aa)+dth(*)
  ENDFOR
ENDELSE

FOR aa=0,alpha1_p-1 DO BEGIN
  W_z(aa,*) = SQRT(4.*ALOG(2.)/PI)/tau_z*EXP(-delta(aa,*)^2*4.*ALOG(2.)/tau_z^2)
  W_y(aa,*) = SQRT(4.*ALOG(2.)/PI)/tau_y*EXP(-delta(aa,*)^2*4.*ALOG(2.)/tau_y^2)
ENDFOR

; Different normalisation of W for different description of anisotropic mosaicity
; and giving unique peak reflectivity
;if (tau_z ne tau_y) then begin

```

```

; FOR ee=0,N_lay-1 DO BEGIN
;   mwz = MAX(W_z(*,ee))
;   mwy = MAX(W_y(*,ee))
;   W_z(*,ee) = W_z(*,ee)/mwz*(mwz+mwy)/2.
;   W_y(*,ee) = W_y(*,ee)/mwy*(mwz+mwy)/2.
; ENDFOR
;endif else begin
;endelse

FOR aa=0,alpha1_p-1 DO BEGIN
  D_0_z(aa,*) = QQ*W_z(aa,*)*di(*)/COS(theta_0)
  D_0_y(aa,*) = QQ*W_y(aa,*)*di(*)/COS(theta_0)
  D_H_z(aa,*) = QQ*W_z(aa,*)*di(*)/abs(COS(theta_H))
  D_H_y(aa,*) = QQ*W_y(aa,*)*di(*)/abs(COS(theta_H))
  Da_0_z(aa,*) = D_0_z(aa,*)*ex_a_0(*)
  Da_0_y(aa,*) = D_0_y(aa,*)*ex_a_0(*)
  Da_H_z(aa,*) = D_H_z(aa,*)*ex_a_H(*)
  Da_H_y(aa,*) = D_H_y(aa,*)*ex_a_H(*)
  Ta_0_z(aa,*) = (1.-D_0_z(aa,*))*ex_a_0(*)
  Ta_0_y(aa,*) = (1.-D_0_y(aa,*))*ex_a_0(*)
  Ta_H_z(aa,*) = (1.-D_H_z(aa,*))*ex_a_H(*)
  Ta_H_y(aa,*) = (1.-D_H_y(aa,*))*ex_a_H(*)
ENDFOR

;*****

sigma_bragg_z = QQ*W_z
sigma_bragg_y = QQ*W_y

IF (theta_bragg LT alpha) THEN BEGIN ;LAUE
FOR aa=0,alpha1_p-1 DO BEGIN
  MATR_z(aa,*,0,0) = Ta_0_z(aa,*)
  MATR_y(aa,*,0,0) = Ta_0_y(aa,*)
  MATR_z(aa,*,1,0) = Da_H_z(aa,*)
  MATR_y(aa,*,1,0) = Da_H_y(aa,*)
  MATR_z(aa,*,0,1) = Da_0_z(aa,*)
  MATR_y(aa,*,0,1) = Da_0_y(aa,*)
  MATR_z(aa,*,1,1) = Ta_H_z(aa,*)
  MATR_y(aa,*,1,1) = Ta_H_y(aa,*)
  M_z(*,*) = MATR_z(aa,0,*,*)

```



```

M_y(*,*) = MATR_y(aa,0,*,*)
FOR ll=1,N_lay-1 DO BEGIN
  Mal_z(*,*) = MATR_z(aa,ll,*,*)
  Mal_y(*,*) = MATR_y(aa,ll,*,*)
  M_z(*,*)=MAL_z(*,*)##M_z(*,*)
  M_y(*,*)=MAL_y(*,*)##M_y(*,*)
ENDFOR
REF_z(aa) = M_z(0,1)
REF_y(aa) = M_y(0,1)
TRA_z(aa) = M_z(0,0)
TRA_y(aa) = M_y(0,0)
ENDFOR
ENDIF ELSE BEGIN ;Bragg
FOR aa=0,alpha1_p-1 DO BEGIN
  MATR_z(aa,*,0,0) = Ta_0_z(aa,*)-Da_0_z(aa,*)*Da_H_z(aa,*)/Ta_H_z(aa,*)
  MATR_y(aa,*,0,0) = Ta_0_y(aa,*)-Da_0_y(aa,*)*Da_H_y(aa,*)/Ta_H_y(aa,*)
  MATR_z(aa,*,1,0) = Da_H_z(aa,*)/Ta_H_z(aa,*)
  MATR_y(aa,*,1,0) = Da_H_y(aa,*)/Ta_H_y(aa,*)
  MATR_z(aa,*,0,1) = -Da_0_z(aa,*)/Ta_H_z(aa,*)
  MATR_y(aa,*,0,1) = -Da_0_y(aa,*)/Ta_H_y(aa,*)
  MATR_z(aa,*,1,1) = 1./Ta_H_z(aa,*)
  MATR_y(aa,*,1,1) = 1./Ta_H_y(aa,*)
  M_z(*,*) = MATR_z(aa,0,*,*)
  M_y(*,*) = MATR_y(aa,0,*,*)
  FOR ll=1,N_lay-1 DO BEGIN
    Mal_z(*,*) = MATR_z(aa,ll,*,*)
    Mal_y(*,*) = MATR_y(aa,ll,*,*)
    M_z(*,*)=Mal_z(*,*)##M_z(*,*)
    M_y(*,*)=Mal_y(*,*)##M_y(*,*)
  ENDFOR
  REF_z(aa) = -M_z(0,1)/M_z(1,1)
  REF_y(aa) = -M_y(0,1)/M_y(1,1)
  TRA_z(aa) = M_z(0,0)+REF_z(aa)*M_z(1,0)
  TRA_y(aa) = M_y(0,0)+REF_y(aa)*M_y(1,0)
ENDFOR
ENDELSE
peak_ind = WHERE(REF_z EQ max(REF_z),count)
dpeak = peak - peak_ind
;corrected 30 Sept 2003:
corrTRA_z(*) = TRA_z(*)*EXP(mu*1.D-8*d/cos(theta_0))*EXP(-mu*1.D-8*d/cos(theta_0-del

```

```

corrTRA_y(*) = TRA_y(*)*EXP(mu*1.D-8*d/cos(theta_0))*EXP(-mu*1.D-8*d/cos(theta_0-del

print,'
print,'
print,'thickness (cm)          ',d*1.D-8
print,'(hkl)                    ',h,k,l
print,'d_hkl (Angstrom)        ',d_hkl
print,'energy (eV)              ',E
print,'wavelength/A            ',lambda
print,'Bragg angle/deg          ',theta_bragg/DEG2RAD
print,'F_hkl/angstrom           ',F_hkl
print,'T/K                       ',T
print,'Debye Waller factor      ',DW
print,'Q (cm-1)                 ',QQ*1.D+8
print,'Attenuation sig_tot(barn)                ',sig_tot
print,'Attenuation mu(cm-1)      ',mu
print,'secondary ext. coeff. (cm-1)          ',max(QQ*W_z)*1.D+8
print,'sec. ext. length (cm)          ',1./max(QQ*W_z)*1.d-8
print,'sec. ext. depth (cm)          ',1./max(QQ*W_z)*1.d-8*cos(theta_0)
print,'Layer thickness (cm)          ',di(0)*1.d-8
print,'Asymmetry factor b           ',b_asymm
print,'mosaicity in scattering plane (deg)    ',tau_z/DEG2RAD
print,'peak reflectivity in scatt. plane     ',max(ref_z)
print,'mosaicity perp. to scatt. plane (deg)  ',tau_y/DEG2RAD
print,'peak reflectivity perp. to scatt. plane ',max(ref_y)
print,'
FOR hh = dpeak(0),alpha1_p-1 DO BEGIN
  printf,lunR,2.*PI/lambda,(theta_bragg+delta(hh,index)+alpha)/DEG2RAD,ref_z(hh-dpeak
  printf,lunT,2.*PI/lambda,(theta_bragg+delta(hh,index)+alpha)/DEG2RAD,corrtra_z(hh-d
ENDFOR

ENDFOR
;end of lambda loop
free_lun,lunR
print,'
print,'This program writes the results in a file called REF.dat '
print,'There are four columns:'
print,'1rst column:      K = 2*PI/wavelength [Angstrom-1]'
print,'2nd column:      grazing angle [degrees], including possible asymmetry angle'

```

```
print,'3rd column:      reflectivity corresponding to mosaicity in scattering plane'
print,'4th column:      reflectivity corresponding to mosaicity in plane perpendicular
print,'
print,'There is an additional file TRA.dat with'
print,'cols 3 and 4 representing the transmitted'
print,'(i.e. attenuated because of absorption + '
print,'TDS + Bragg scattering) intensity.'
print,'
print,'
```

END