TOBYFIT version 2.0

Least-Squares Fitting to single crystal data on HET, MARI and MAPS

T.G.Perring

January 2000

Rutherford Appleton Laboratory Chilton DIDCOT Oxfordshire OX11 0QX

TOBYFIT version 2.0

1. Introduction

TOBYFIT_V2[1] allows the simultaneous fitting or simulation of several datasets from each of several runs, where a dataset can be a one-dimensional cut, a two-dimensional slice or an SPE file. Any mixture of these dataset types can be fitted or simulated.

The user selects:

- a cross-section model that is global to all runs, and set initial values for the cross-section parameters

Then for each run:

- sets experimental configuration parameters: instrument parameters (lengths in the spectrometer, moderator and chopper parameters etc.), lattice constants for the sample and temperature
- sets the data files, their corresponding parameter files (.PAR) and (if simulating) the output file names, for any of several datasets, and selects which of these datasets are to fitted or simulated
- background models for each of the datasets for the run

The background models are always local to a given run, but within one run a single background model can be chosen to apply to all the datasets (with parameters constrained to be the same for all the datasets), or background model(s) can be chosen for each dataset, and the parameters set and varied independently for each dataset.

The user then selects which runs (and thereby which datasets within each of those runs) are to be fitted or simulated. These tasks are started by the FIT or SIMULATION commands.

The extensions to TOBYFIT have necessitated major changes to the original TOBYFIT interface. The command shell has two levels. In the executive command shell the cross-section model and parameters are set, and fitting or simulation is implemented. In the run command sub-shell the experiment configuration parameters and dataset information for each run are set.

At present, TOBYFIT allows up to 10 distinct runs, each of which can contain up to 10 datasets. The run information and dataset filenames for the whole can be saved to a file, to be read into a later TOBYFIT session. The idea is to allow the information for an entire experiment to be saved and read into the program. In any one TOBYFIT session, a subset of the datasets for any one run, and a subset of the full list of runs can be selected for fitting.

All operations are controlled from a command line. Some general rules apply about entering commands at the prompt:

- All commands can be abbreviated as long as they are unambiguous within the operating shell. For example, in the executive (that is, top-level) shell the command FILE can be abbreviated as FIL, but not to FI or F because these cannot be distinguished from the FIT command.
- Excess spaces and tabs are ignored.
- Some commands allow a list of integers to be entered, as indicated for example by <dataset_index_list>. The input in such cases can be a comma- or space-separated list of values, a range of values or a mixture of both e.g. “1-3,4-7,8”. Lists of real numbers, as indicated by <par_1> <par_2> <par_3> … can be separated either by commas or spaces.

[1] If you have any queries about TOBYFIT_V2, please contact the author:

T.G.Perring
Rutherford Appleton Laboratory, Didcot, Oxfordshire OX11 0QX, United Kingdom
Tel: +44 (0)1235 445428 Fax: +44 (0)1235 445720
e-mail: t.g.perring@rl.ac.uk

- If a command requires information that is omitted from a command line the user is prompted for that information.

The user can provide his or her own model(s) for the scattering cross-section. The input and output variables for the subroutines that return the cross-section and background are described in Appendix ***. The system dependent installation, compilation and linking of TOBYFIT are described in Appendix ***.

2. Summary of commands

2.1 Executive command shell:

CROSS_SECTION	Set cross-section model and magnetic form factor (if any)
RUN	Enter the RUN shell in which parameters and datasets for the indicated run are set
SELECT	Select the run(s) to be fitted or simulated
DELETE	Delete all information for indicated run(s)
SAVE	Save all the cross-section, run and dataset information to a file
FILE	Set all the cross-section, run and dataset information from a file
SIMULATE	Simulate dataset(s)
FIT	Fit dataset(s)
VARY	To select variable moderator parameters and/or goniometer angles
INTEGRATION	Select integration method and integration control parameters
EXIT or QUIT	Exit TOBYFIT, with a prompt asking for confirmation

2.2 CROSS_SECTION command shell:

P1=nn, P2=nn …	Set values of the indicated parameters
LIST	List the values of the indicated parameters
SAVE	Save the present values of the parameters to a file
FILE	Set the run parameters from a file
EXIT or QUIT	Exit the CROSS_SECTION shell and return to executive command shell

2.3 RUN command shell:

- **Changing experimental configuration parameters**:

P1=nn, P2=nn …	Set values of the indicated parameters
SET	Set instrument or chopper parameters to those for specific instruments or choppers
LIST	List the values of the indicated parameters
SAVE	Save the present values of the parameters to a file
FILE	Set the run parameters from a file
EXIT or QUIT	Exit the RUN shell and return to executive command shell

- **Changing dataset parameters:**

PARAMETER_FILE	Set the detector file name for indicated dataset
INPUT_FILE	Set the file name of input data for the indicated dataset
OUTPUT_FILE	Set the filename for the result of a simulation of the indicated dataset
REMOVE	Remove the indicated detectors from the fit or simulation

RESTORE	Restore all detectors to the fit or simulation
DELETE	Delete all information for the indicated dataset(s)
SELECT	Select datasets for fitting or simulation
DATASETS	Display parameters and information for all the datasets

- **Setting background models for the run and datasets:**

BACKGROUND	Set background model and parameters for the run and datasets

3. Detailed description of commands

3.1 Executive command shell

CROSS_SECTION

The CROSS_SECTION command passes control to sub-shell in which the cross-section model index, the model type (sharp or broad), cross-section parameter values and the magnetic form factor are set. For a list of the commands in the CROSS_SECTION sub-shell, see the section entitled “CROSS_SECTION Command Shell”.

Format: CROSS_SECTION

RUN

For the indicated run, the RUN command passes control to a shell in which the parameters that define the run, the names of the files containing input data, detector parameters and the filenames that will contain the result of simulations or fits can be changed. For a list of the commands in the sub-shell that can be used to change these parameters and the select datasets to be fitted or simulated, see the section entitled “RUN Command Shell”.

Format: RUN <run_index>

SELECT

Selects the run(s) that will be simulated or fitted. The command can also be used to list the selected runs, and the datasets, parameter files and output files for the selected datasets within each run. The command resets the selected runs each time it is typed, i.e. the command is not cumulative. To deselect all runs, enter “0” as the selected run.

Format: SELECT <run_index_list>

SELECT 0 *(to deselect all runs)*

SELECT ? *(lists the selected runs and datasets)*

DELETE

Deletes all entries for the indicated run indices. The command should be used with caution.

Format: DELETE <run_index_list>

SAVE

Saves all the TOBYFIT information: the cross-section parameters, the run parameter and file names, the indices of selected runs and datasets for each run. The saved information includes parameters for the runs and datasets that have not been selected as well, so that the file can be used as a repository of all information for an experiment, and runs and datasets selected for fitting or simulation as required.

If the optional filename is omitted then the parameters are saved to the currently open TOBYFIT file.

Format: SAVE [<filename>]

FILE

Closes the currently open TOBYFIT file, without saving any changes, and then reads a TOBYFIT file from the file entered on the command line.

Format: FILE <filename>

SIMULATE

Calculates the convolution of the model cross-section with the instrumental resolution for the selected datasets within the selected runs.

Format: SIMULATE

FIT

Fits the model cross-section to the selected datasets within the selected runs.

Format: FIT

This command calls the Multi-frills interactive least-squares fitting program. All the commands of Multi-frills are available, and in addition some custom commands to make use of Multi-frills easier with the hierarchical structure of run/datsets/workspace. See Section 4, “Fitting Data”, for more details about fitting data and the custom commands within Multi_frills.

VARY

Allows the parameters in one or more blocks of run parameters (experiment configuration parameters) to be varied in fits, for example the moderator parameters or the goniometer arc angles.

Format:	VARY ALL	Allow all run parameters to be varied
	VARY NONE	Allow no run parameters to be varied
	VARY ?	lists which run parameters are presently allowed to vary
	VARY	Prompts the user to choose in which blocks the parameters are allowed to vary.

The blocks that can be varied at present are listed below. See the list of experiment configuration parameters for the meanings of the run parameters.

Name of block:	Experiment configuration parameters:
MODERATOR	s1, s2, s3, s4, s5
GONIOMETER	gs, gl, dphi

INTEGRATION

Selects the integration method and alters the parameters that control the integration for that method. At present the integration can be performed using a Monte Carlo algorithm or the original TOBYFIT version 1, which can be selected with the names MONTE_CARLO and TOBYFIT_V1. See Appendix A: “Integration methods” for details about the integration methods and the control parameters.

Format:	INTEGRATION ?	Lists the names of the available integration methods, the presently selected method and the values of its control parameters.

INTEGRATION	METHOD	?	Lists the names of the available methods and the present method.
INTEGRATION	METHOD	<name>	Set the integration method.

INTEGRATION	PARAMETERS	?	List the control parameters for the current integration method
INTEGRATION	PARAMETERS	DEFAULT	Sets default values
INTEGRATION	PARAMETERS		Prompts for new values of the control parameters

EXIT or **QUIT**

Exit TOBYFIT, with a prompt asking for confirmation. The two commands are equivalent.

Format: EXIT

QUIT

3.2 CROSS_SECTION command shell

When the command CROSS_SECTION is typed in the executive command shell, control is passed to a sub-shell in which the cross-section model index, the model type (sharp or broad), cross-section parameter values and the magnetic form factor can be changed. The parameters can be listed to the screen with the LIST command, and saved to or read from a file using the SAVE and FILE commands.

Changing parameter values:

The cross-section parameters in the order in which they are LISTed to the screen are as follows:

ICROSS — Cross-section model number. It is used by the internally provided and user supplied subroutines to select the model for fitting or simulating data. **Negative values of ICROSS are reserved for internally provided cross-section models**. A list of the internal models is given in Appendix G. The interpretation of positive values of ICROSS depends on the user-provided subroutines SQW_BROAD and SQW_SHARP.

SHARP — Indicates whether the broad or sharp cross-section models (in SQW_BROAD or SQW_SHARP respectively) are to be used.
SHARP = 0 selects broad cross-section models
SHARP = 1 selects sharp cross-section models

P1 … P20 — Parameter values to be used by the cross-section models. If TOBYFIT is used to fit the data, these are used as the initial values for the least-squares fitting. By default the parameter values will be updated by the fitting algorithm.

ION — Sets the magnetic form factor. The format of the entry is a real number, where the integer part is the atomic number and the decimal part specifies the valence e.g. "25.3" specifies Mn^{3+}. To switch off the magnetic form factor, enter "0.0".

To change parameter values, type the parameter name followed by the new value:

<par_name> = <value>

To change the values of several successive parameters in the order they appear in the list above, enter the series of values on the line following the name of the first parameter:

<par_name> = <value_1> <value_2> ... <value_n>

The two formats can be put on one line. Examples are

icross=50	Changes the cross-section model
icross=50, p1=100, 9, 0, 2, ion=29.1	Changes icross, P1, P2, P3, P4 and ion

LIST

List the values of the indicated parameter(s)

Format:	LIST	prints value of all parameters
	LIST <par_name>	prints the value of the named parameter
	LIST <par_name_1> <par_name_2>	prints value of all parameters between the two named parameters

SAVE

Save the present values of the parameters to a file. If no file name is given, save to the current file; if a file name is given, save to that file and make it the current file. Only the cross-section information is written to the file. To save the entire Tobyfit configuration (cross-section, run and fitting information), the user must be in the executive command shell.

Format: SAVE

SAVE <file_name>

FILE

Set the run parameters from a file, and make that file the current file for subsequent SAVE commands.

Format: FILE <file_name>

EXIT or **QUIT**

Exit the CROSS_SECTION shell and return to the executive command shell. The two commands are equivalent.

Format: EXIT

QUIT

3.3 RUN command shell

When the command RUN <run_index> is typed in the executive command shell, the RUN command shell is entered, in which the run parameters, background and dataset information can be set for the chosen run index.

3.3.1 Changing experimental configuration parameters

Experimental configuration parameters include the incident energy, Ei, parameters that characterise the pulse width from the moderator and the chopper, sample lattice parameters and scattering plane. A full list of configuration parameters is given below.

To change parameter values, type the parameter name followed by the new value:

<par_name> = <value>

To change the values of several successive parameters in the order they appear in the list below (which is the same order they appear in when LISTed to the screen), enter the series of values on the line following the name of the first parameter:

<par_name> = <value_1> <value_2> ... <value_n>

These two formats can be put on one line. Examples are

ei=50	sets incident energy
as=2.50 2.50 4.04	changes the three lattice parameters
ei=50 as=2.50, 2.50, 4.04, temp=14.3	changes Ei, as, bs, cs and temp.

SET

Sets the instrument specific and chopper specific parameters to the values for named spectrometers and choppers.

Format:		
	SET ?	print current spectrometer and chopper names
	SET INSTRUMENT <inst_name>	set the instrument
	SET CHOPPER <freq> <type> [<inst_name>]	Set the chopper frequency (Hz), name and (optionally) the instrument. Recall, for example, the MARI and HET "B" choppers have different parameters but the choppers can be used on either instrument. e.g. SET CHOP 600 S HET

LIST

List the values of the indicated parameter(s)

Format:		
	LIST	prints value of all parameters
	LIST <par_name>	prints the value of the named parameter
	LIST <par_name_1> <par_name_2>	prints value of all parameters between the two named parameters

SAVE

Save the present values of the parameters to a configuration file. If no file name is given, save to the current configuration file; if a file name is given, save to that file and make it the current configuration file.

Format: SAVE

SAVE <file_name>

FILE

Set the run parameters from a configuration file, and make that file the current configuration file.

Format: FILE <file_name>

EXIT or **QUIT**

Exit the RUN shell. The two commands are equivalent.

Format: EXIT

QUIT

Experimental configuration parameters (run parameters):

The run parameters are listed below in the order in which they appear when the LIST command is typed within the run command shell.

Energy and crystal orientation:

ei	Incident energy (meV).
psi	Angle the vector ***u*** defined below makes with the incident beam direction.
elo	Lower limit of energy transfer for simulations of .SPE files.
ehi	Upper limit of energy transfer for simulations of .SPE files.
de	Energy transfer bin-size for simulation of .SPE files.

Primary spectrometer dimensions:
These can be set for particular instruments with the SET INSTRUMENT command, or set individually.

x0	Moderator-chopper distance (m).
xa	Distance from start of collimation near the moderator to chopper (m).
x1	Chopper-sample distance (m).
wa	Width of collimation at its start (mm).
ha	Height of collimation at its start (mm).

Moderator parameters:

s1-s5	Moderator parameters, to be interpreted according to the model for the moderator lineshape defined by IMOD below. The values of the parameters depend on both the incident energy and the moderator type (water, liquid methane, liquid hydrogen etc.).
thetam	Angle of moderator face to the incident beam direction (deg.) *[This parameter is will be altered if a particular instrument is chosen with the SET INSTRUMENT command]*

imod Moderator model:

IMOD = 1 Ikeda-Carpenter model $s(1) = \tau_f$ (μs), $s(2) = \tau_s$ (μs), $s(3) = R$ $(0 \le R \le 1)$

The values of the parameters depend on moderator type, Ei etc. and should be fitted to a monochromatic vanadium if reliable linewidths are to be extracted. See Section 4.1 for how to fit the moderator parameters. At the very least, the incoherent elastic line from the sample should be fitted. Good starting values for the ISIS water and methane moderators are:

Ei > 100 meV: τ_f (μs) = $70/\sqrt{Ei(meV)}$ and fix $\tau_s = 0$, $R = 0$.

Generally, only for the lowest incident energies with high resolution choppers is it necessary to account for the slowing down terms τ_s and R. In this case take $\tau_s = 25$ (water) or 35 (methane). Fit with caution!

Fermi chopper parameters:

If the primary spectrometer dimensions and moderator angle belong to a recognised instrument then the Fermi chopper parameters can be set with the SET CHOPPER command.

pslit	Chopper slit width (mm).
radius	Radius of chopper body (mm).
rho	Radius of curvature of slits (mm).
Hz	Frequency of rotation of chopper (Hz).
tjit	Chopper jitter (FWHH in μs).

Sample parameters:

- Crystallographic parameters:

as, bs, cs	Lattice parameters (Å)
aa, bb, cc	Lattice angles (deg.)
uh, uk, ul	***u*** is first of two vectors that define scattering plane (r.l.u.)
vh, vk, vl	***v*** is second of two vectors that define scattering plane (r.l.u.)

- Refinement of crystal orientation:

The following four parameters can be used to account for a small crystal misorientation. If the crystal is correctly aligned then all four parameters can be set to zero. If not, the parameters can be refined by fitting inelastic data. This is explained in Section 4.2, where the full definition of the parameters is given.

omega	Angle of goniometer x-axis w.r.t. the vector ***u*** (deg.).
gs, gl, dpsi	Goniometer angles that give actual position of the plane defined by ***u*** and ***v*** (deg.).

- Sample shape:

The following parameters are used to account for sample size effects in the resolution. This contribution can be ignored by setting sx=sy=sz=0. In this instance, the sample geometry vectors are not required and can be set to zero. If the sample geometry is to be accounted for, then two vectors must be provided to define the geometry x,y and z axes with respect to the crystal lattice.

xh, xk, xl	Sample geometry x-axis (r.l.u.)
yh, yk, yl	Sample geometry y-axis (r.l.u.)
sx, sy, sz	Characteristic sample dimensions (mm)
isam	Sample geometry (slab, cylinder etc.). The interpretation of sx, sy, sz depends on ISAM.

ISAM = 1 Slab sample sx = width (mm), sy = thickness (mm), sz = height (mm)

- Other sample parameters:

temp Sample temperature (K)
eta Sample mosaic spread (FWHH in deg.)

3.3.2 Changing dataset parameters

PARAMETER_FILE

Changes the name of the file containing detector data for the chosen dataset index number. The command checks that the file exists; if not, then the existing filename is left unchanged.

Format: PARAMETER <dataset_index> <file_name>

PARAMETER ? *(lists the parameter file names for all dataset indices)*

INPUT_FILE

Changes the name of the file containing data to be fitted for the chosen dataset index number. The command checks that the file exists; if not, then the existing filename is left unchanged.

Format: INPUT <dataset_index> <file_name>

INPUT ? *(lists the input file names for all dataset indices)*

OUTPUT_FILE

Changes the name of the file to contain the results of a simulation for the chosen dataset index number.

Format: OUTPUT <dataset_index> <file_name>

OUTPUT ? *(lists the input file names for each dataset indices)*

REMOVE

Remove a range of detectors from the parameter file for the indicated dataset.. If fitting, then the contribution to the signal from the REMOVEd detectors is eliminated - e.g. in an SPE file the detectors are omitted from the fit, or in a cut file the contribution from the indicated detectors is removed from any point in the cut, with due care paid to the normalisation of the remaining pixels that contribute to that point. Note that ‘detector number’ refers to the workspace number in the SPE file from which the cut was taken.

Format: REMOVE <dataset_index> <detector_index_range>

REMOVE ? *(lists the removed detectors for each dataset)*

nb: (1) The command is not cumulative. (2) Presently only applies to cuts, not SPE files. (3) No check is made that the indices of the removed detectors are consistent with the number of detectors in the parameter file.

RESTORE

Restores all the detectors for the indicated dataset indices.

Format: RESTORE <dataset_index_list>

DELETE

Deletes all dataset parameters (file names, removed detector list and background model and parameters) for the indicated dataset(s)

Format: DELETE <dataset_index_list>

SELECT

Select the datasets to be fitted or simulated. The command resets the selected datasets each time it is entered, that is, the command is not cumulative. To deselect all datasets, enter "0" as the selected dataset.

Format: SELECT <dataset_index_list>

SELECT 0 *(to deselect all datasets)*

SELECT ? *(to list all seelected datasets)*

DATASETS

Displays the parameters (file names and removed detectors) for the indicated dataset(s). Does not include the background parameters, for which use the command "BACKGROUND ?" (see below).

Format: DATASETS <dataset_index_list>

DATASETS ? *(to display parameters for all datasets)*

3.3.3 Changing background models and parameters

BACKGROUND

Toggles the scope of the background fitting between LOCAL (each dataset within the run having its own background model) and GLOBAL (the same model applies to all datasets within the run). The command also can be used to set the background model and parameter values if the dataset index is given as the first parameter. To set the background model and parameter values for the case of global scope use *dataset_index* = 0. The background models can be set independently of the current scope of the background fitting, that is, the user can set the background models for each dataset without

Format: BACKGROUND GLOBAL

BACKGROUND LOCAL

BACKGROUND 0 <background_model> <par_1> <par_2> …

BACKGROUND <dataset_index> <background_model> <par_1> <par_2> …

BACKGROUND ? *(to display background parameters for run (if GLOBAL) or all datasets (if LOCAL))*

4. Fitting data

4.1 Introduction

When the FIT command is entered, the program enters the Multi_frills interactive least squares fitting package, which is indicated by the command line prompt '#'. The data files and parameter files for the selected datasets are read and passed to Multi_frills, along with the cross-section and background parameters.

Following the prompt '#', the user types commands to

- list, remove or modify the data points
- set up, fix or constrain the initial parameters
- plot the data points and/or the calculated function for any of the data sets
- adjust least squares fitting constants
- perform the least squares fit

On completion of a fit, the resulting parameter values are listed along with their estimated errors and a value for χ^2. The program returns to command mode so that the user can plot the fit and choose new initial conditions (e.g. by clearing fixed parameters) before trying again. The results of fits are recorded in an ASCII log file and lists of parameters may be stored in files for subsequent printing. The plots may also be stored on disk in Postscript files for output to a laser printer, or the calculated function written to ASCII files in tab delimited form suitable for input to proprietary programs such as Excel or SigmaPlot. All the commands of Multi-frills are available, and in addition some custom commands to make use of Multi-frills easier with the hierarchical structure of run/datasets/workspace. These are described in the following section.

Once the user is satisfied with the fit, type EXIT to leave the Multi_frills interface to return to TOBYFIT. The cross-section and background parameter values in TOBYFIT are automatically updated to the values obtained during the fitting procedure.

4.2 Organisation of data and customised Multi_frills commands.

The data is held in Tobyfit in a hierarchical structure of runs, datasets, and workspaces within a dataset. The run and dataset hierarchy has been described earlier in the introduction, Section 1. If a dataset is an .SPE file, then it will have as many workspaces as there are entries in the corresponding .PAR file. If a dataset is a 2D slice with bin centres at ($x_0+m.\Delta x$, $y_0+n.\Delta y$) for m=1→M, n=1→N, then there will N workspaces, each corresponding to a given n, and containing data for x with m=1→M. If the dataset is a 1D cut, then there is only one workspace.

Multi_frills can only produce simple x-y plots, that is, only one workspace can be selected for plotting at any one time. To help navigate within Multi_frills the following two commands are available.

@USE

Multi-FRILLS user command to change spectrum number according to the run index, dataset index and workspace index within that dataset.

Format: @USE [<run> <dataset>] <workspace>

e.g.	@use <workspace>	*change workspace number within the current run and datset*
	@use <run> <dataset>	*change run and dataset (workspace set to smallest valid value)*

@use <run> <dataset> <workspace> *changes all three*

@LIST

Multi-FRILLS user command to list all the selected runs and datasets to the screen.

Format: @LIST

4.3 Fitting the moderator parameters S1-S5

Before fitting inelastic data, the contribution of the moderator pulse width to the resolution function should be determined for each of the different incident energies of the runs. Ideally, datasets should be taken from a vanadium standard with incident energy, chopper package and frequency equal to those of the runs being analysed. As an alternative during the course of an experiment, it may be possible to fit the incoherent elastic line from the sample. Use the built-in sharp cross-section model number –2, and set all the parameters to zero except the first :

```
> cross_section
cross> icross=-2, sharp=1
cross> p1=1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ion=0
cross> exit
>
```

This cross-section corresponds to a planar dispersion relation, with zero gradient and passing through the point $(\boldsymbol{q},\omega) = (0,0,0,0)$. Depending on how the data has been normalised, the value of P1 may need to be several orders of magnitude larger or smaller than unity.

Set the instrument, chopper and sample lattice parameters (any reasonable dummy lattice parameters are acceptable), and read in the vanadium data file. Because the scattering from vanadium is isotropic, you can use a single spectrum .SPE file. On HET, MARI or MAPS a good choice is the first spectrum of the .SPE file created by HOMER with the default RINGS map. This spectrum corresponds to the detectors with the smallest scattering angles. Once the .SPE file and its corresponding .PAR file have been set in TOBYFIT, remove all but the first spectrum with the REMOVE command e.g. if the data files have been set as run 1, dataset 5, and there are 12 spectra in the .SPE file:

```
run_1> remove  5  2-12.
```

Before fitting the data, set the moderator parameters as variable:

```
> vary
   ALL, NONE or <CR> for options : <CR>

moderator parameters  (F-type T,F) : T
goniometer parameters (F-type T,F) : F
```

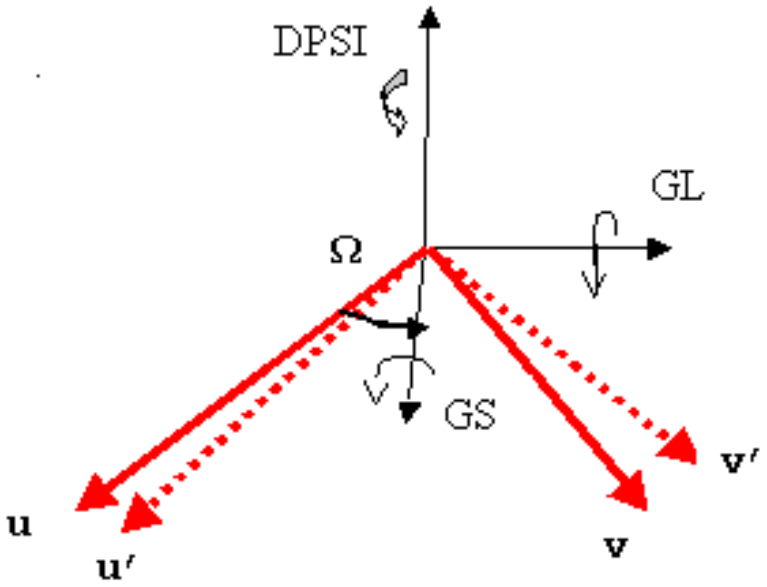
The moderator parameters can now be fitted. Good starting values for the parameters with depend on the moderator material and temperature, and the lineshape model that is being selected.

4.2 Refining the crystal orientation angles DPSI, GL and GS

Not uncommonly the inelastic data reveal that the crystal was not oriented precisely as supposed. This can happen if the crystal was aligned elsewhere before being placed in the spectrometer on sample environement with only the rotation about a single axis. The nominal scattering plane may not then coincide with the symmetry plane of the spectrometer. One, two or all three orientation angles can be refined in Tobyfit_v2 if inelastic scattering data is available which is sensitive to those degrees of freedom. For example, on HET the scattering in one of the two horizontal banks is sensitive to rotations about a vertical axis, i.e. to the correction DPSI to the angle PSI. The data in one of the two vertical banks is sensitive to rotation about a horizontal axis perpendicular to the incident neutron beam, so that two datasets separated by 90º in PSI are together sensitive to both corrections GL and GS to the settings of the arcs on a goniometer. (Note: do not use assumed symmetry of the nominal crystal orientation to sum data at the same scattering angle from both horizontal or both vertical detector banks, as then the sensitivity to the correction angles is lost. The misorientation will then manifest itself solely in a broadening of peaks in the data, rather than shifts of peak positions.)

Definitions of the correction angles DPSI, GL and GS

The scattering plane is defined by vectors in reciprocal space (*uh*,*uk*,*ul*) and (*vh*,*vk*,*vl*). The actual directions of these vectors, **u'** and **v'**, are related to the nominal directions in the symmetry plane of the spectrometer, **u** and **v**, by rotation about two goniometer arcs and a perpendicular axis as shown in the figure below:



Note that ***u*** and ***v*** are not necessarily perpendicular. The misorientation is described by rotations performed in the following order:

1) rotation about the smaller goniometer arc by angle GS (in degrees). This arc is assumed to be mounted on the larger goniometer arc.
2) Rotation of the larger gonimeter arc by angle GL (in degrees),
3) Rotation about a vertical axis by DPSI (in degrees).

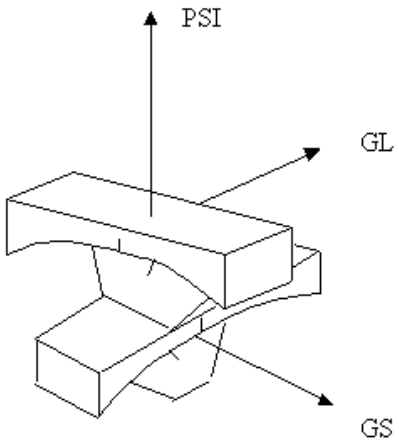


Figure: Definition of the goniometer arcs. Positive rotations of both GS and GL are shown.

The vector about which the smaller goniometer arc rotates is contained in the plane at angle OMEGA to the nominal vector ***u***. This parameter cannot be refined. Normally, OMEGA should be set to zero, so that the corrections GS and GL are about virtual goniometer arcs. This situation prevails, for example, when the crystal is mounted on sample environment which:

(1) has rotation only about an axis perpendicular to the scattering plane (such as the rotating closed-cycle refrigerators at ISIS), or
(2) the sample is mounted on a physical goniometer (such as the Franke and Heydrich goniometer at ISIS) but the corrections GS and GL are not of themselves of interest.

The parameter OMEGA is provided for the instance when it is desired to connect the rotations GS and GL to movements on a physical gonimeter. Usually, the vector ***u*** will be oriented along one of the two arcs of a goniometer, in which case OMEGA = 0º or ±90º.

To refine DPSI, use peaks in detector banks within the scattering plane. If OMEGA=0, then to refine GL choose peaks in detector banks above and below the scattering plane when ***u*** is along the incident beam, and to refine GS use peaks in the same detector banks but when ***u*** is perpendicular to the incident beam.

Once the parameters DPSI, GL and GS have been determined, they can be fixed for all runs that use the same mounting of the crystal.

Appendix A: Integration methods

The integration method is selected using the command:

INTEGRATION METHOD <name>

where the currently available methods are MONTE_CARLO and TOBYFIT_V1. Appropriate default values for the integration control parameters for the currently selected method can be set using the command:

INTEGRATION PARAMETERS DEFAULT

In the case of the Monte Carlo integration method, for example, these include the accuracy request and the number of points at which the integrand is sampled.

A.1 Monte Carlo integration method

The user can select any or all of the following contributions to the resolution of the spectrometer with the INTEGRATION PARAMETERS command:

- moderator lineshape
- moderator area
- chopper pulse shape
- chopper jitter
- sample volume
- detector area
- detector depth
- energy bin width
- mosaic spread

The command additionally prompts for the following parameters that control the integration for each pixel:

MC_TYPE	0 = random number generator used to sample from the various contributions to the instrumental resolution 1 = quasi-random number sequence used to sample from the contributions. The algorithm uses a Sobol sequence which provides a self-avoiding set of random vectors in the n-dimensional space corresponding to the n independent integration variables of the selected contributions to the resolution function. The advantage of this method is that error in the integration converges as 1/N where N is the number of samplings from the n-dimensional space. Sampling using a conventional random number generator produces converges as $1/N^{1/2}$.
MC_LOOP_MIN	Minimum number of Monte Carlo points per pixel
MC_LOOP_MAX	Maximum number of Monte Carlo points per pixel
MC_RELACC	Relative error criterion for convergence.

The algorithm checks every MC_LOOP_MIN cycles (up to a maximum of MC_LOOP_MAX) if the relative accuracy estimate is less than MC_RELACC. If so, then the integration loop ceases for that pixel. If the integral is precisely zero after MC_LOOP_MIN cycles, then the integration loop also ceases.

A.2 TOBYFIT_V1 integration method

A.2.1 Method

The TOBYFIT_V1 method uses a simplified model of the resolution function that accounts only for the moderator and chopper pulse-widths (including chopper jitter), the energy bin-width, and the area of the detector. The mosaic spread of the sample can optionally be accounted for as well. The method is essentially identical to that used in the original TOBYFIT program. The method proceeds as follows:

- The detector is split up into a grid of equally spaced points on a Cartesian grid according to parameter MC below.
- For each point on this grid a one-dimensional convolution is performed between the dispersion relation and the locus of ($\delta\mathbf{Q}$,$\delta\hbar\omega$) for the principal axis of the resolution function where only the moderator, chopper and energy bin widths are accounted for. The correct moderator lineshape (as selected with the run parameter IMOD) is convoluted with a Gaussian that approximates the chopper and energy bin-widths to give the distribution along this trajectory.
- The integrals from each point on the detector grid are averaged.

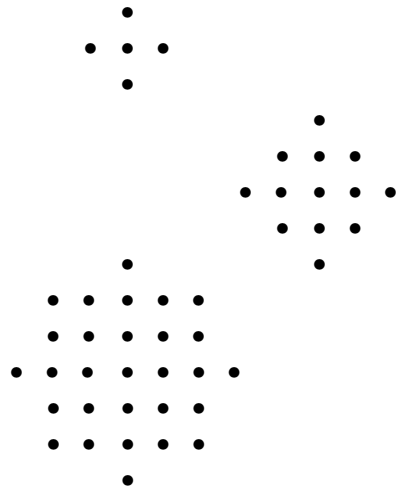
The mosaic spread can be included by setting the parameter N_MOS to a positive non-zero integer. The mosaic spread is assumed to be Gaussian and isotropic. The weighted average at the points within a radius N_MOS on a Cartesian grid gives the final result, where the weights are given by the Gaussian mosaic function and sum to unity. The examples below should make the algorithm clear:

N_MOS = 0 no mosaic spread included

N_MOS = 1 radius = ½ fwhh 5 points

N_MOS = 2 radius = ¾ fwhh 13 points

N_MOS = 3 radius = fwhh 29 points



N_MOS ≥ 4 radius = 2% contour

Beware that the time to perform any calculation is proportional to the number points and therefore increases as the square of N_MOS.

WARNING: Including the mosaic spread can lead to unusually multi-peaked results if the mosaic spread is comparable or larger than the typical angular divergences arising from the detector width or height. This is because the calculated scattering is the weighted sum of a number of patterns, each corresponding to zero mosaic spread but displaced by the angular separation between the points in the patterns above. With large mosaic spread it is advisable to use the Monte Carlo integration method.

A.2.2 Integration control parameters

MC — No. points on detector grid.

The true number of points is generally a little different from MC. If the detector has width W and height H, then the spacing between points is given by $\Delta = W.H / MC$ and the actual number of points on the detector is $MC_{true} = nint(W/\Delta).nint(H/\Delta)$.

A good choice for the non-PSD tubes on HET and MARI is MC=12, and for the PSDs is MC=1-4 depending on the variation of the scattering cross-section across the pixel.

N_MOS — Mosaic spread grid density.

EPSREL — Relative accuracy request for one-dimensional convolution.

EPSABS — Absolute accuracy request for one-dimensional convolution.
The less stringent of EPSREL and EPSABS is used to test convergence, that is, the algorithm returns if its estimate of the error on the result satisfies:

$$|ERROR| \leq Max (EPSABS, EPSREL \times |RESULT|)$$

FRAC — The integration limits along the principal axis are where the resolution function is a fraction FRAC of the peak value.

ROUTINE — Chooses the integration method;
= 0 Romberg integration (see Numerical Recipes in FORTRAN 2nd edition p.134)
= 1 Locally adaptive integration routine D01AJF from NAG subroutine library (where available)

Romberg Integration (ROUTINE = 0):

JLO — Minimum no. of intervals into which the integration range is divided = 2^{JLO-1}.

JHI — Maximum no. of intervals into which the integration range is divided = 2^{JHI-1}.

Subroutine D01AJF (ROUTINE = 1):

MAXSUB — Maximum number of sub-divisions of integration range.

Appendix B: User supplied subroutines

The user must supply three subroutines, SQW_BROAD, SQW_SHARP and BACKGROUND. These return respectively:

SQW_BROAD:	$S(\mathbf{Q},\omega)$
SQW_SHARP:	$A(\mathbf{Q})$ and $\omega_D(\mathbf{Q})$ in $S(\mathbf{Q},\omega) = A(\mathbf{Q})\delta(\omega-\omega_D(\mathbf{Q}))$
BACKGROUND:	Background intensity

Example subroutines for broad and sharp cross-section models are available with the installation of TOBYFIT. See Appendix D: “System specific installation” for more details.

B.1 SUBROUTINE SQW_BROAD (icross, p, qx, qy, qz, eps, weight)

Calculates the scattering law as a function of momentum transfer and energy transfer. Use, for example, to fit or simulate the signal from broad scattering functions or damped dispersion relations.

Input variables:

icross	cross-section model number
p	array of cross-section parameter values (at present p(1)...p(20))
qx, qy, qz	momentum transfer in laboratory coordinate frame ($Å^{-1}$)
eps	energy transfer (meV)

Output variables:

weight	$S(\mathbf{Q},\omega)$ at (qx, qy, qz, eps)

In addition, a number of parameters are available to the user through common blocks in the `INCLUDE` statements: `INCLUDE 'SAMPLE_PARAMETERS.INC'` and `INCLUDE 'FUNCTION_CALLS.INC'`. These are:

arlu(3)	reciprocal lattice parameters
angrlu(3)	reciprocal lattice angles
temp	temperature (Kelvin)
gau_ht	GAU_HT × exp (GAU_PRE × x^2) = normalised Gaussian; FWHH = 10% of
gau_pre	energy resolution (useful for artificially broadening a dispersion relation by a negligible amount)
uinv	Matrix to convert Q-vector from components in laboratory coordinate frame (x-axis along $\mathbf{k_i}$ and the z-axis vertically upwards) to crystal cartesian coordinates (x-axis is along $\mathbf{a}^*$, the z-axis along $\mathbf{a}^* \times \mathbf{b}^*$)
ubinv	Matrix to convert Q-vector from components in laboratory coordinate frame to reciprocal lattice units.
n_function_calls	A counter that can be incremented to record the number of calls made to SQW_BROAD

Normally the first calculation that is performed in SQW_BROAD or SQW_SHARP is to convert the components of momentum transfer into reciprocal lattice units:

$$qrlu(i) = \Sigma_j\ ubinv(i,j)\ q(j) \quad \text{where } q(1) \equiv qx,\ q(2) \equiv qy,\ q(3) \equiv qz,$$

Function subroutines available to the user include:

BOSE(eps,T)	Returns eps/(1-exp(-eps/k_BT)) in meV
FORM_TABLE(qsqr)	Returns the magnetic form factor for the ion set with the FORM command in the executive command shell.

B.2 SUBROUTINE SQW_SHARP (icross, p, qx, qy, qz, wdisp, weight)

Calculates the energy of a dispersion relation and its spectral weight as a function of momentum transfer.

Input variables:

icross	cross-section model number
p	array of cross-section parameter values (at present p(1)...p(20))
qx, qy, qz	momentum transfer in laboratory coordinate frame (Å^{-1})

Output variables:

wdisp	energy of dispersion relation at qx, qy, qz
weight	S(**Q**,ω) at (qx, qy, qz, eps)

Just as for SQW_BROAD, the parameters ARLU ... N_FUNCTION_CALLS are available to the user through common blocks in the `include` statements, and in particular the matrix UBINV to calculate the momentum transfer in reciprocal lattice units from the components qx, qy, qz. The functions BOSE and FORM_TABLE are also available.

B.3 SUBROUTINE BACKGROUND (iback, bp, x, y, answer)

Calculates the background cross-section. The program does not convolute the background with the instrumental resolution function.

Input variables:

iback	background model number
bp	array of background parameters (at present bp(1) ...bp(10))
x, y	x and y coordinates that define position on spectrum: .SPE file: x = energy transfer (y = 0; ignored) 1-D cut: x = position along cut (y = 0; ignored) 2-D slice: (x, y) = location on 2-D slice plane If pixels have been filtered out through use of the REMOVE command, then the value(s) of x and y are recalculated to be the average of those for the remaining pixels.

Output variables:

answer	S(**Q**,ω) for the background

In addition to *x* and *y*, a number of other variables are available:

ei	Incident energy (meV)
eps	Energy trnsfer (meV)
x2	Sample-detector distance (m)
phi	Scattering angle (radians)
beta	Azimuthal scattering angle (radians). The rotation is defined as positive for anti-clockwise rotation about the direction of the incident beam. On HET, the 2.5m west bank has BETA=0, the north bank has BETA=π/2.

Appendix C: Compatibility with the original TOBYFIT program

Two versions of the original TOBYFIT program exist. One version uses the integration method described earlier as TOBYFIT_V1, and is restricted to fitting several cuts from one run only, or a single .SPE file. The other uses a simplified version of the Monte Carlo integration method described above, and can fit only one cut or .SPE file. The two versions have different formats for the .DAT file that contains run parameters, cross-section and background parameters.

C.1 Old parameter files.

TOBYFIT_V2 will recognise the .DAT files of both versions of TOBYFIT. The cross-section parameters will be made global to all runs, and the run parameters and background parameters will be loaded into Run_1 and Dataset_1. The moderator pulse-shape parameters have different meanings in TOBYFIT_V2; the parameters read from the original TOBYFIT .DAT files are used to calculate the equivalent values for TOBYFIT_V2. Default values are given to parameters required by TOBYFIT_V2 but not contained in the TOBYFIT .DAT files. You must provide the names of the detector parameter and input/output data files, as these are not contained in the TOBYFIT .DAT files.

C.2 BACKGROUND subroutine.

If you copied but did not alter the default subroutine BACKGROUND.FOR then you can copy and use the new BACKGROUND.FOR (see Appendix D: “System specific installation”), which will provide the same background models with same meaning (1=quadratic in *x*; 2=exponential decay in *x*; 101=quadratic in energy transfer; 102=exponential decay in energy transfer).

If you have written your own BACKGROUND.FOR, then you will need to make two changes to the Fortran source code for it to be compatible with TOBYFIT_V2:

(i) The argument list of the subroutine BACKGROUND contains an extra parameter, y, because of the necessity to allow for two-dimensional slices. Alter the first line from:

```
subroutine background (iback, bp, x, result)
```

to:

```
subroutine background (iback, bp, x, y, result)
```

NB: In the case of .SPE files and 1-D cuts, the second coordinate *y* is set by TOBYFIT_V2 to zero.

(ii) Remove the call to BACKGROUND_INIT and the declaration of the variables EI, EPS, PHI, BETA, X2 and PSI. Instead, insert the line

```
include ‘background_parameters.inc’
```

This defines and allows access to the parameters EI, EPS, PHI, BETA and X2.

Alternatively, copy the new BACKGROUND.FOR into your working area and replace the IF..ENDIF block that defines the background models with a copy of your own background model definitions.

C.3 SQW_BROAD and SQW_SHARP subroutines.

To make your existing SQW_BROAD and SQW_SHARP subroutines work with TOBYFIT_V2 you will have to make two changes to the Fortran source code:

(i) TOBYFIT_V2 uses the value of a variable called N_FUNCTION_CALLS to print diagnostic information during fitting and simulation. For this information to be available, Old SQW_BROAD and SQW_SHARP routines should be updated to include the statement:

```
include 'function_calls.inc'
```

and in the body of the subroutine add the line of code:

```
n_function_calls = n_function_calls + 1
```

(ii) Remove the logical name TOBYFIT_SOURCES: from the line:

```
include 'tobyfit_sources:sample_parameters.inc'
```

to leave:

```
include 'sample_parameters.inc'
```

The include statement will otherwise pick up the include file for the original TOBYFIT, which is inconsistent with TOBYFIT_V2. When you link TOBYFIT_V2 following the instructions in Appendix D: "System specific installation", the correct location for the include files will be searched.

In addition, if you used the input variables (qx, qy, qz) anywhere else than calculating qh, qk, ql and qsqr in the lines:

```
qh = ubinv(1,1)*qx + ubinv(1,2)*qy + ubinv(1,3)*qz
qk = ubinv(2,1)*qx + ubinv(2,2)*qy + ubinv(2,3)*qz
ql = ubinv(3,1)*qx + ubinv(3,2)*qy + ubinv(3,3)*qz
```

and

```
qsqr = qx**2 + qy**2 + qz**2
```

you will have to make another change:

(iii) The components of the momentum transfer ***Q*** as passed to SQW_BROAD and SQW_SHARP are expressed in different orthonormal coordinate frames in the two versions of TOBYFIT. In the original TOBYFIT , the components (qx, qy, qz) are expressed in crystal cartesian coordinates, a right-handed frame where the *x*-axis is along **a*** and the z-axis along **a***$\times$ **b*** . In TOBYFIT_V2, the components (qx, qy, qz) are expressed in laboratory coordinates, with the *x*-axis along the incident wavevector and the *z*-axis vertically upwards. However, unless these components of momentum transfer are explicitly used there is no incompatibility because the elements of the matrix UBINV are correspondingly different to correctly convert the components (qx, qy, qz) to reciprocal lattice units.

Appendix D: System specific installation

D.1 ISIS VMS cluster

If it is not included in your LOGIN.COM, first run the standard HET/MARI/MAPS command file:

```
$ @het$disk1:[hetmgr]login
```

This will set up all the logical definitions needed to compile and link TOBYFIT_V2.

If you are converting from the original TOBYFIT programs, then read Appendix C: “Compatibility with the original TOBYPLOT program”. There are one or two minor modifications that need to be made to your existing SQW_BROAD, SQW_SHARP and BACKGROUND routines. Example subroutines can be found in:

TOBYFIT_V2_SOURCES:SQW_BROAD.FOR

TOBYFIT_V2_SOURCES:SQW_SHARP.FOR

TOBYFIT_V2_SOURCES:BACKGROUND.FOR

Once you have prepared these three subroutines, then run the command file:

```
$ @tobyfit_v2_sources:tobyfit_v2_link_custom
```

This will create the executable version on TOBYFIT_V2 in your working area, and a debug version in your scratch area. The sizes of the files will be about 2000 blocks and 4000 blocks respectively.

If you need to link to other subroutines, copy the command file TOBYFIT_V2_SOURCES:TOBYFIT_V2_LINK_CUSTOM.COM to your working area and edit accordingly.

D.2 Other systems

D.2.1 General instructions

● Download the source code appropriate for your system, using the instructions at the web site:

<contact T.G.Perring for the source code>

There are about 400 files in total, consisting of Fortran source code (with extensions .FOR and .F), include files (.PAR, .INC and .DEF), and a data file (.DAT).

If necessary, modify the user subroutines SQW_BROAD.FOR, SQW_SHARP.FOR and BACKGROUND.FOR.

● Compile all the Fortran files. Compiler options for different systems and compilers are given below. The source code is mostly Fortran77 with a few common extensions, for example variable names with more than 6 characters, line length up to 132 characters and DO...WHILE loops. There are a small number of routines that perform system dependent functions, for example constructing full file names from an environment variable and file name, renaming files, inquiring of the system clock etc. Versions of these routines that

compile under the G77 compiler (Unix/Linux and WNT) and Digital Fortran (Unix, WNT and VMS) are included in the installation.

● The compiled code then needs to be linked with the PGPLOT library. If this free software is not already installed on your system, then ask your system administrator to install it. Alternatively, download the source code from

http://astro.caltech.edu/~tjp/pgplot/

and install according to the instructions available at that web page. Compilation instructions for various compilers and operating systems appear below.

● Before running the program, two environment variables need to be defined:

TMPDIR	Default location for all default file names produced by TOBYFIT.
TOBYFIT_V2_SOURCES	Location for source code of TOBYFIT_V2 and any data files used by the program.

To define the environment variables:

VMS systems:

```
$ define   TMPDIR   sys$scratch
$ define   TOBYFIT_V2_SOURCES   my_disk:[my_directory]
```

Unix systems

The environment variables ***must*** be defined in upper case.

Bash shell:

export TMPDIR=“/users/my_area/temp”

export TOBYFIT_V2_SOURCES=”/users/my_area/tobyfit_v2”

Cshell:

setenv DISPLAY /users/my_area/temp

setenv TOBYFIT_V2_SOURCES /users/my_area/tobyfit_v2

Windows NT:

Follow the links:

Start menu → Settings → Control Panel → System → Environment

and set the environment variable names and their values. Typically, define TMPDIR as c:\temp. Define TOBYFIT_V2_SOURCES as the location of the source code for Tobyfit_v2. You will probably have to restart the computer for the definitions to become active.

D.2.2 System and compiler specific compilation options

D.2.2.1 G77 compiler (Unix systems)

Compile with the following options:

```
g77 –ffixed-line-length-none –fno-backslash –malign-double –w –c *.for *.f
```

`-ffixed-line-length-none`: Ensures that the compiler can interpret source code extending beyond 72 characters per line.

`-fno-backslash`: By default the backslash character is interpreted as a control character in some compilers, including G77. This option ensures that backslashes are not given special status.

`–malign-double`: This option is needed only on Intel x86 processors, and ensures that COMMON variables are aligned on natural boundaries. The `-w` option suppresses the warning messages that are produced which indicate that the compiler is performing the alignment.

Link the resulting object files with the PGPLOT library to produced TOBYFIT_V2.EXE (the link to the PGPLOT library will in general depend on your installation of PGPLOT):

```
g77 *.o –lpgplot –lX11 –o tobyfit_v2.exe
```

D.2.2.2 Digital Fortran (Digital Unix systems)

Compile with the following options:

```
f77 –extend_source –align dcommons –assume backslash –w –c *.for *.f
```

`-extend_source`: Ensures that the compiler can interpret source code extending beyond 72 characters per line.

`-assume backslash`: By default the backslash character is interpreted as a control character in some compilers. This option ensures that backslashes are not given special status.

`–align commons`: This option ensures that COMMON variables are aligned on natural boundaries. The `-w` option suppresses the warning messages that are produced which indicate that the compiler is performing the alignment.

Link the resulting object files with the PGPLOT library to produced TOBYFIT_V2.EXE (the link to the PGPLOT library will in general depend on your installation of PGPLOT):

```
f77 *.o –lpgplot –lX11 –o tobyfit_v2.exe
```

D.2.2.3 Digital Fortran (alpha VMS systems)

Compile with the following options:

```
fortran/extend_source/align=common=natural
```

D.2.2.4 Digital Visual Fortran (Windows NT and Unix)

Ensure that you request extended source code and aligned variables, by following:

Project menu → settings → fortran(tab) → fortran data → common element alignment = 8

Project menu → settings → fortran(tab) → fortran language→fixed form line length = 132

Other compilers and operating systems

The source code should compile with equivalent compiler options to allow extended line lengths and no special interpretation of backslash characters (usually the backslash is interpreted as a control character on unix compilers). Alignment of the variables in common blocks is not essential, but will normally result in a significant improvement in program performance.

The possible exceptions are the subroutines that perform system dependent manipulations. The subroutines are PROMPT.FOR, SPAWN.FOR and those whose names begin with SYS_..., for example SYS_FULL_FILENAME.FOR. If you compile and link Tobyfit_v2 on a system other than one listed above,

please let the author have details on compiler options and changes that you had to make to the source code. If you have difficulties, please contact the author for advice.

Appendix E: Calculation of the magnetic form factor

TOBYFIT_V2 uses the same method as the original TOBYFIT programs to calculate the magnetic form factor. The <j0> form factors for the 3d and 4d transition metals, and the lanthanides and actinides are available, using the analytical approximation detailed in the International Tables for Crystallography, Volume C (1992), Section 4.4.5 (pp391-399). A look-up table is created from which the form factor can estimated by linear interpolation in the user supplied subroutines using the function FORM_TABLE (see Appendix B: “User supplied subroutines”, and below).

If you wish to use the subroutines in other programs, you need to know about the four sub programs detailed below.

Subroutine SET_FORM_ION (value, ierr)

Sets the magnetic form factor. The format of the subroutine argument is a real number, where the integer part is the atomic number and the decimal part specifies the valence e.g. "29.2" specifies Cu^{2+}. To switch off the magnetic form factor, enter "0.0". [Calls SET_FORM_FREEMAN (nf,, form(nf)).]

Subroutine SET_FORM_TABLE

Initialises a look-up table to be used by the function FORM_TABLE below. SET_FORM_TABLE requires that the magnetic ion has been set by a previous call to SET_FORM_ION above.

Double precision function FORM (qsqr)

Calculates the magnetic form factor at QSQR, the square of the momentum transfer in $Å^{-1}$. Requires that the magnetic ion has been set by a previous call to SET_FORM_ION above.

Double precision function FORM_TABLE (qsqr)

Uses a look-up table to return the magnetic form factor at QSQR, the square of the momentum transfer in $Å^{-1}$. Requires that the magnetic ion has been set by a previous call to SET_FORM_ION and that the look-up table has been initialised with SET_FORM_TABLE.

Appendix F: Frequently asked questions

● How do I include several dispersion relations ?

You have to use SQW_BROAD, which returns S(**Q**,eps) at a given **Q** and eps. For the integration algorithms in Tobyfit to work, the dispersion relations must be given a narrow width of 10% of the (vanadium) energy resolution. The two parameters GAU_HT and GAU_PRE are provided for this purpose (see Appendix B.1). If there are N dispersion relations, with frequencies WDISP(I) and spectral weights WT(I) at the input value of **Q** , then the return value of S(**Q**,eps) should be set to:

```
 weight = gau_ht*( wt(1)*exp(gau_pre*(eps-wdisp(1))**2) +
+                  wt(2)*exp(gau_pre*(eps-wdisp(2))**2) +
                             :
+                  wt(n)*exp(gau_pre*(eps-wdisp(n))**2)   )
```

This is in fact the same algorithm that is performed by Tobyfit when SQW_SHARP is called, except that the implementation is hidden from the user. However, SQW_SHARP can only be used to return one dispersion relation and its spectral weight.

● What is the cause of the error "Cannot open file (CONNFL)" ?

These occur when Tobyfit is unable to find a named file, or, if the file is present, is unable to open the file. Make sure that you have defined the two environment variables (logical names on VMS):

TMPDIR Location of all temporary files and default location for postscript files, fitting log file

TOBYFIT_V2_SOURCES Location of Tobyfit_v2 source code

Note that, if on a PC running Windows NT, these have been set in the 'Systems' window of the 'Control Panel', then you will have to reboot the computer for the definitions to be operative.

Appendix G: Built-in cross-section models

Negative values of the index ICROSS are reserved for internally provided cross-section models. User supplied cross-section models with a negative value of ICROSS will be ignored, even if there is no internally provided model corresponding to that value.

The built-in cross-section models are described below. Suggestions for additional models can be e-mailed to the author (preferably with accompanying source code).

Sharp cross-section models

● ICROSS = -1

Planar approximation to a dispersion relation. If $\omega = \omega(h,k,l)$, then the dispersion relation is approximated by:

$$\omega = \omega(h_0, k_0, l_0) + (\partial\omega/\partial h)(h-h_0) + (\partial\omega/\partial k)(k-k_0) + (\partial\omega/\partial l)(l-l_0)$$

P1	Spectral weight
P2-P4	h_0, k_0 and l_0 respectively
P5	$\omega(h_0, k_0, l_0)$
P6-P8	$\partial\omega/\partial h$, $\partial\omega/\partial k$ and $\partial\omega/\partial l$ respectively, in units of meV/r.l.u.

● ICROSS = -2

The same as above, except that the numerical values of the gradient terms are in meV/Å^{-1}.

Broad cross-section models

There are presently no internally provided broad cross-section models.