

# EasyDD Documentation

Taha Sochi\*

September 14, 2010

---

\*Email: [tahasochi@scienceware.net](mailto:tahasochi@scienceware.net) OR [R5896329@yahoo.co.uk](mailto:R5896329@yahoo.co.uk).

# Contents

<b>Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>5</b>
<b>List of Tables</b>	<b>6</b>
<b>1 Nomenclature</b>	<b>7</b>
<b>2 Introduction</b>	<b>9</b>
<b>3 Main Window</b>	<b>12</b>
3.1 Menus . . . . .	12
3.1.1 File Menu . . . . .	13
3.1.2 Data Menu . . . . .	13
3.1.3 CCD Menu . . . . .	27
3.1.4 Settings Menu . . . . .	30
3.1.5 Fitting Menu . . . . .	31
3.1.6 Sinograms Menu . . . . .	33
3.1.7 MultiBatch Menu . . . . .	37
3.1.8 Help Menu . . . . .	45
3.2 Tools . . . . .	45
3.2.1 Toolbars . . . . .	46
3.2.2 Tool Buttons . . . . .	46
3.2.3 Spin Boxes . . . . .	46
3.2.4 Sliders . . . . .	47
3.2.5 Context Menu . . . . .	47
3.2.6 Status Bar . . . . .	47
3.2.7 Colour Bar . . . . .	47

3.3	Tab Widget . . . . .	48
3.4	List Widgets . . . . .	48
<b>4</b>	<b>2D Plotter</b>	<b>49</b>
4.1	Menus . . . . .	51
4.1.1	File Menu . . . . .	51
4.1.2	Fitting Menu . . . . .	51
4.1.3	Settings Menu . . . . .	52
4.2	Buttons . . . . .	55
4.3	Labels . . . . .	56
<b>5</b>	<b>3D Plotter</b>	<b>56</b>
<b>6</b>	<b>Form Dialog</b>	<b>57</b>
6.1	Menus . . . . .	60
6.1.1	File Menu . . . . .	61
6.1.2	Edit Menu . . . . .	61
6.1.3	Options Menu . . . . .	62
<b>7</b>	<b>Modules of EasyDD</b>	<b>63</b>
7.1	Curve-Fitting . . . . .	63
7.2	Sinogram Treatment . . . . .	65
7.3	EDF Processing . . . . .	65
7.4	Graphic Presentation . . . . .	65
<b>8</b>	<b>Context Menus</b>	<b>66</b>
<b>9</b>	<b>Dialogs</b>	<b>66</b>
9.1	Settings Dialog . . . . .	66
9.2	FTP Dialog . . . . .	68

<b>10 Appendix 1: Tilt Correction Derivation</b>	<b>70</b>
<b>11 Appendix 2: Missing-Rings Correction</b>	<b>72</b>
<b>12 References</b>	<b>74</b>

## List of Figures

1	EasyDD main window. . . . .	12
2	The main window File menu. . . . .	13
3	The main window Data menu. . . . .	14
4	‘Browse for Folder’ dialog. . . . .	15
5	Tab management message. . . . .	17
6	XY dialog. . . . .	18
7	Change File Order dialog. . . . .	19
8	MCA dialog. . . . .	19
9	A sample of an ERD map file for a $16 \times 16$ detector. . . . .	21
10	A sample of a HEXITEC thresholds file for a $20 \times 20$ detector. . . . .	21
11	HEXITEC dialog. . . . .	23
12	SRS scalar dialog. . . . .	27
13	The main window CCD menu. . . . .	27
14	EDF settings dialog. . . . .	30
15	The main window Settings menu. . . . .	31
16	Tab settings dialog. . . . .	31
17	The main window Fitting menu. . . . .	32
18	The main window Sinograms menu. . . . .	34
19	Back projection dialog. . . . .	36
20	Alignment dialog. . . . .	37
21	The main window MultiBatch menu. . . . .	38
22	EDF image display dialog. . . . .	40
23	MultiBatch curve-fitting dialog. . . . .	43
24	The main window Help menu. . . . .	45
25	‘About EasyDD’ dialog. . . . .	45
26	EasyDD main window following data mapping and curve-fitting. . . . .	49

27	The 2D plotter. . . . .	50
28	Plotter File menu . . . . .	51
29	Plotter Fitting menu . . . . .	51
30	The Plotter Settings menu. . . . .	53
31	Plotter statistical weight dialog. . . . .	53
32	Plotter settings dialog. . . . .	54
33	Plotter properties dialog. . . . .	54
34	pseudo-Voigt dialog . . . . .	56
35	A 3D graph of total intensity as a function of voxel position in a tab.	57
36	Form dialog. . . . .	58
37	The form dialog File menu. . . . .	61
38	The form dialog Edit menu. . . . .	62
39	The form dialog Options menu. . . . .	63
40	Settings dialog. . . . .	67
41	FTP dialog. . . . .	68
42	A schematic diagram to demonstrate the derivation of tilt correction.	70
43	The continuum and discrete values for the number of pixels in a ring.	73
44	A sample of 1D diffraction pattern extracted from a 2D EDF pattern.	73

## List of Tables

1	Common shape functions used in EasyDD to describe peak profiles	50
2	Default values for the form restrictions columns . . . . .	60
3	Statistical indicators for pattern fitting and refinement . . . . .	64

## 1 Nomenclature

$\chi^2$	goodness-of-fit index
$A$	area under peak
$C$	number of constraints
$F$	Full Width at Half Maximum
$I^c$	calculated integrated intensity
$I^o$	observed integrated intensity
$m$	mixing factor in pseudo-Voigt function
$O$	number of observations
$P$	number of parameters
$R_B$	Bragg's residual
$R_{\text{exp}}$	expected residual
$R_p$	profile residual
$R_S$	structure factor residual
$R_{wp}$	weighted profile residual
$w$	statistical weight
$X$	position of peak
$xyz$	spatial coordinates
$y^b$	background count rate
$y^c$	calculated count rate
$y^o$	observed count rate
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
ADD	Angle Dispersive Diffraction

ASCII	American Standard Code for Information Interchange
DFT	Discrete Fourier Transform
EDD	Energy Dispersive Diffraction
EDF	European Data Format
ERD	Energy Resolving Detector
ESRF	European Synchrotron Radiation Facility (Grenoble, France)
FFT	Fast Fourier Transform
FTP	File Transport Protocol
FWHM	Full Width at Half Maximum
GUI	Graphic User Interface
HEXITEC	High Energy X-Ray Imaging Technology
LS	Least Squares
MCA	Multi-Channel Analyzer
RAL	Rutherford Appleton Laboratory (Didcot, UK)
SRS	Synchrotron Radiation Source (Daresbury, UK)

## 2 Introduction

The principal objective of EasyDD project was to develop a computer code for batch processing, visualisation and large-scale analysis of huge volumes of spectral data, mainly those obtained from synchrotron radiation sources for powder diffraction applications. Such a utility greatly assists studies on various physical systems and enables far larger and detailed data sets to be rapidly interrogated and analysed. At the start of EasyDD project a number of general objectives were set; the main ones were

- The program must be user friendly to minimise the time and effort required to learn and use. A graphic user interface was therefore adopted in favour of a command line interface although the latter is more common in scientific computing and much easier to develop.
- It should be capable of handling several common data file formats including a generic format so that the program can be used for processing data produced by different detectors and acquisition systems.
- The program must lend itself to future development to meet the ever-increasing demand on data processing and analysis capabilities. Therefore, it should be built on a continuously developing technology to keep in pace with the advancements in computing and GUI technology.
- It must be optimised for the commonly available computational resources; most importantly CPU time and memory. The program therefore tries to set the limits of its data processing capacity to the limits of the available computational resources.

EasyDD can be defined as a high throughput software to manage, process, analyse and visualise scientific data in general and synchrotron X-ray data in particular.

The name ‘EasyDD’ comes from the original name ‘EasyEDD’ which was adopted for historical reasons as the software was developed initially for the users of the energy dispersive diffraction SRS station 16.4. As the program eventually evolved to be more general and can be used for processing ADD data as well as EDD data, the name ‘EasyDD’ was adopted to reflect this extension. In fact, currently there is no restriction on the program being used for general applications not related to synchrotron and powder diffraction, as the program is capable of processing any data having the correct format. One of these formats is a generic  $xy$  style which can be used for all-purpose data. The program, which consists of about 30000 lines of code, is written in C++ language and uses a hybrid approach of procedural and object oriented programming. Its main attribute is the ability to process large quantities of data files with ease and comfort using limited time and computing resources. Most EasyDD algorithms are optimised for speed and hence it can be rightly described as ultra-fast high throughput software.

EasyDD combines Graphic User Interface (GUI) technology, such as dialogs, tooltips, colour coding, and context menus, with standard scientific computing techniques. Its resources include the standard C++ library, Qt toolkit [5] and its extension QwtPlot3D [6] for GUI design, with numerous algorithms, functions and techniques. The ultimate objective of EasyDD is to become a workbench for spectral data analysis. In EasyDD, the main data are stored in a 3D vector where the basic unit is a ‘voxel’ object in which all information relevant to an individual data set are stored. As a user-friendly feature, most components have memory, i.e. the settings are saved on exiting the program. Another user-friendly feature is that the components have explanatory tooltips which give brief description of the function (e.g. button tooltips) or reveal information related to the data represented by that component (e.g. tab tooltips).

Currently, eight input data formats are supported: generic  $xy$ , MCA of the Diamond synchrotron, MCA of the ESRF, ERD, LUCIA, HEXITEC, SRS 16.4, and EDF files. The first is a simple  $xy$  format where the first column in the file contains the  $x$  values (e.g. energy or channel number or angle) while the second column contains the  $y$  values (e.g. intensity or count rate). The second is the MCA format of the Diamond Light Source [1]. The file contains the  $y$  values (number of counts) only, as a function of an implicit channel number, with possible redundant header and footer lines. The third is the MCA format of the European Synchrotron Radiation Facility [2]. The file has only headers to be ignored, and the first data line starts with '@A' with each 16 data entries occupying a single text line ending with backslash '\'. Again, the data in the ESRF MCA files are number of counts versus an implicit channel number. The fourth is the format of the ERD multi-pixel 2D detector of the University of Manchester. The fifth is the format of the LUCIA beamline at the SOLEIL synchrotron [4]. The sixth is the format of the HEXITEC [3] multi-pixel 2D detector. The seventh is the format of the data files obtained from station 16.4 of Daresbury Synchrotron Radiation Source [7] in EDD mode. The eighth is the format of the EDF image files of the ESRF. The code can be easily extended to support other data formats.

One of the main functionalities of EasyDD is to read files from data sets collectively and map their information on GUI components. What is required for this operation is to deposit the files in a directory and invoke the relevant reading function. In the case of the SRS, where the data files have a highly structured format, the files are read and automatically recognised (i.e. SRS, scalars and vectors), and therefore non-SRS files in the source directory are identified and ignored. For the other formats, the file extension is used for type recognition, and hence no foreign files of the same extension should be mixed. On reading the files, the data are stored in memory and mapped on a 2D colour-coded tab. Multiple tabs from dif-

ferent data sources and of different data types can be created at the same time. The tabs can also be removed collectively or individually in any order. On removing a tab, all the data belonging to that tab are erased from memory and hence lost permanently. The remaining tabs will be relabelled to reflect the current state.

In the following sections we outline the main components of EasyDD.

### 3 Main Window

This is a standard GUI dialog with menus, toolbars, a status bar, context menus and so on. The basic functionality of the main window (seen in Figure 1) is to serve as a platform for accessing and managing the other components.

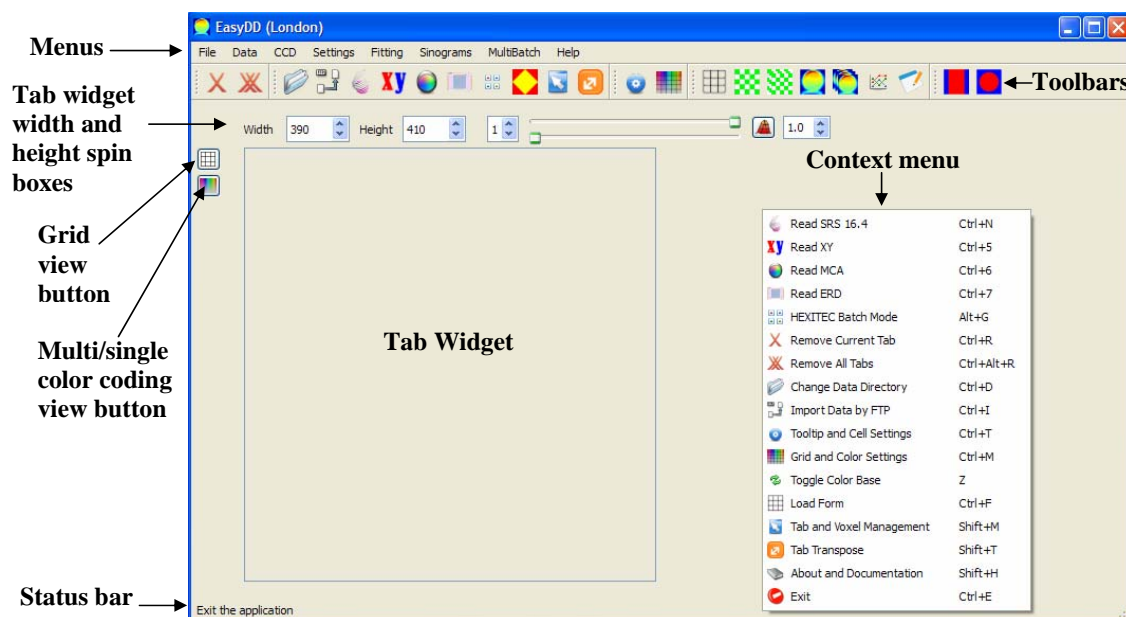


Figure 1: EasyDD main window.

#### 3.1 Menus

The main window has several menus which provide access to the program functions. These menus, some of which have submenus, contain all the main items of the

program. The keyboard shortcut for each item is shown on the right of the item title, and an icon characterising the item is displayed on the left of the title.

### 3.1.1 File Menu

This menu (Figure 2) includes the following items

- Remove Current Tab (✘): to remove the current tab and erase data from memory.
- Remove All Tabs (✘✘): to remove all tabs and clean the memory.
- Save Tab Image (📄): to save the image of the current tab in png format.
- 3D Graph (📊): to plot a 3D graph of the current tab.
- Exit (🚫): for closing the main window and hence exiting the program. All saveable settings will be saved to the disc, and hence activated on the next launch of the program.

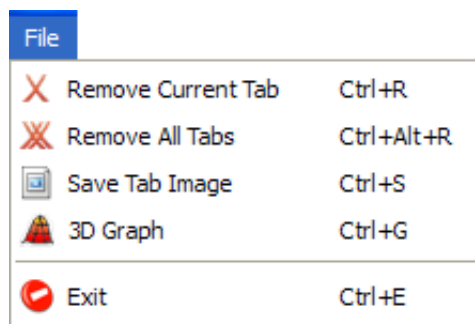


Figure 2: The main window File menu.

### 3.1.2 Data Menu

This menu (Figure 3) includes

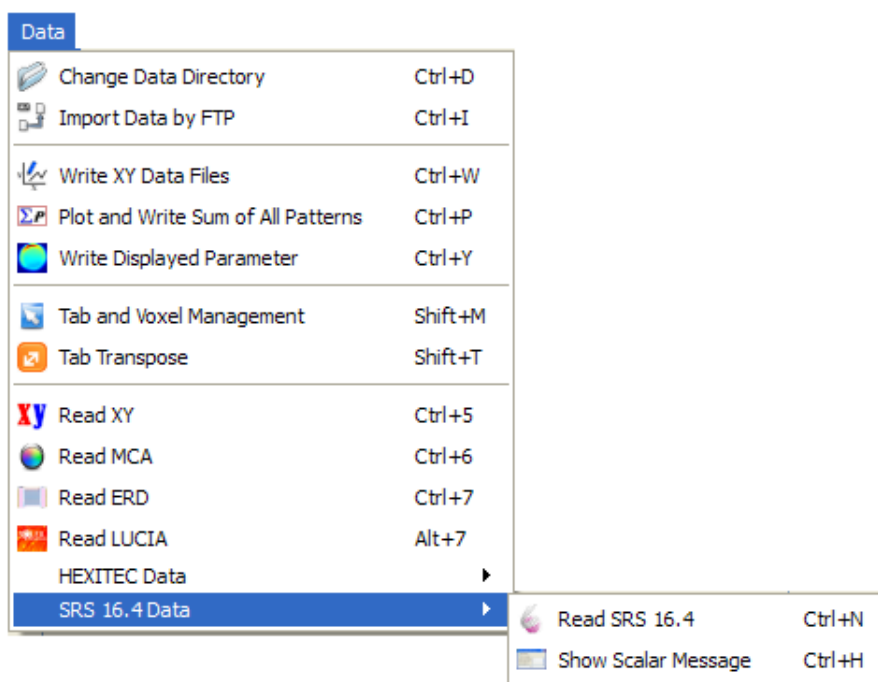


Figure 3: The main window Data menu.

- Change Data Directory (

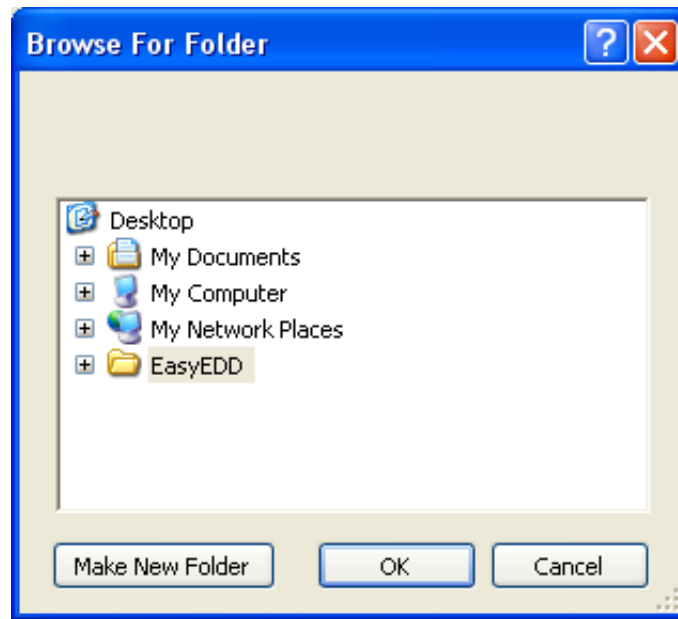


Figure 4: ‘Browse for Folder’ dialog.

(one file of \*.txt type for each voxel) in generic  $xy$  format. The files of each tab are saved to a folder named ‘xyDataTab $N$ ’ where ‘ $N$ ’ stands for the tab number. The naming of files follows the style ‘ $TtRrCc$ ’ where  $t$ ,  $r$  and  $c$  are the indices of tab, row and column respectively. The indices have constant width (e.g. ‘01’ for ‘1’ as a 2-digit number) to keep the order when re-reading these files. This routine facilitates saving, converting between data types, and exporting to other applications for further processing such as visualisation.

- Plot and Write Sum of All Patterns ( $\Sigma P$ ): to plot the sum of all patterns in the current tab and write the numeric data to an  $xy$  text file named ‘spectraSumTab $N$ ’ where  $N$  is the tab number. This function is based on the assumption that all common channels of the patterns of the current tab voxels share the same  $x$  coordinates. However, it is general with regard to the number of data points in each pattern and hence it works even when they are different in size by summing the existing channels and plotting the result for the maximum number of channels. The curve-fitting capacity of the plotter

used in this function is disabled.

- Write Displayed Parameter (

a single keyword with its parameters as the rest of the line will be ignored and hence can be used for storing comments or entries that may be needed in the future. It should be remarked that all these functions are available in the tab context menu (with no need for an input file) by right-clicking on the current tab and choosing the appropriate option. In this case the current cell, row or column is identified by the click position. If a second item (cell, row or column) is required, as in the case of copy and exchange functions, the user will be instructed to click on the second item. Although the use of a context menu is more convenient, the advantage of performing tab and voxel management through an input file is to execute a number of these operations in a single run.

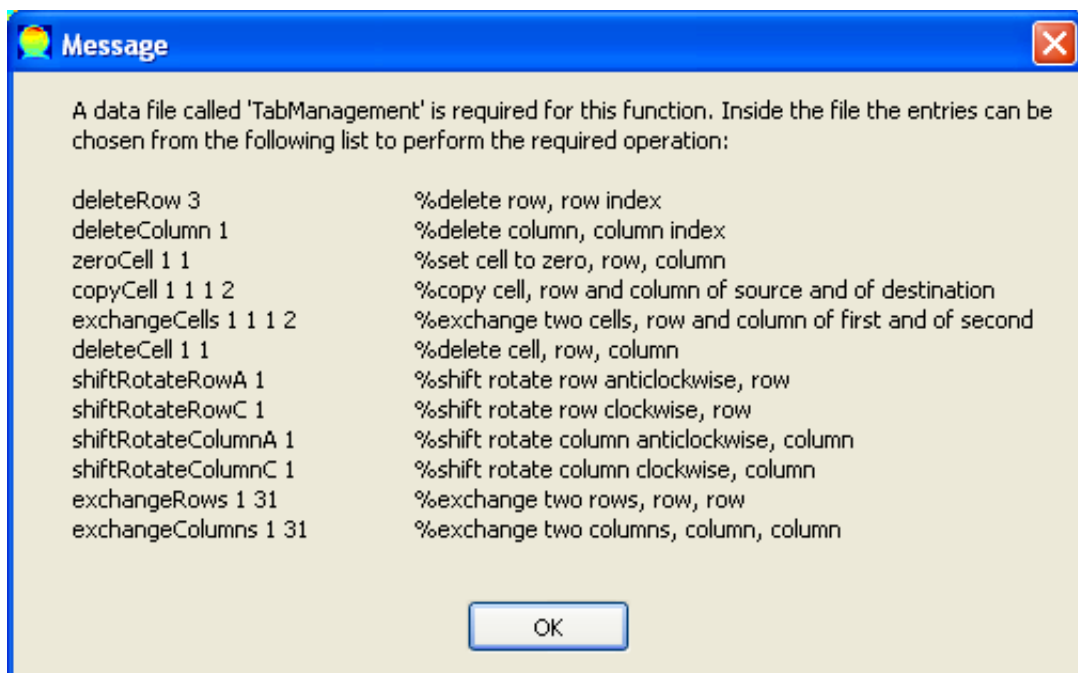


Figure 5: Tab management message.

- Tab Transpose (

value as a second entry in each row. The rest of the row, which may contain other data such as an error index, will be ignored. Each file should contain data for a single voxel represented by a single tab cell. The process starts with the appearance of a dialog (Figure 6) from which the user can choose the number of rows and columns in the tab. This is followed by the ‘Change File Order’ dialog (Figure 7) from which the user can change the order of reading files or delete non-required files. The order of files can be changed by first selecting a file, then moving it up or down in the list using the ‘Up’ and ‘Down’ push buttons. Similarly, a file can be deleted by selecting it first followed by clicking ‘Delete’ button. It should be remarked that the reading operation will proceed only when the source directory contains sufficient number of data files. This is also true for other functions that require reading a certain number of data files from a source directory.

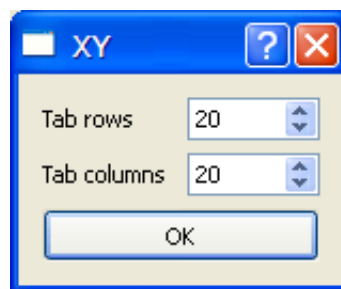


Figure 6: XY dialog.

- Read MCA (🌈): for reading MCA data files. There are two supported MCA formats, as explained on page 11. In both cases, the process starts with the launch of the ‘MCA’ dialog (Figure 8) from which the user determines the number of rows and columns in the tab and the number of header and footer rows to be ignored. This is followed by the ‘Change File Order’ dialog (Figure 7) as in the case of an XY reading. It is noteworthy that for the ESRF MCA files the number of footers should be set to zero to avoid ignoring valid data entries.

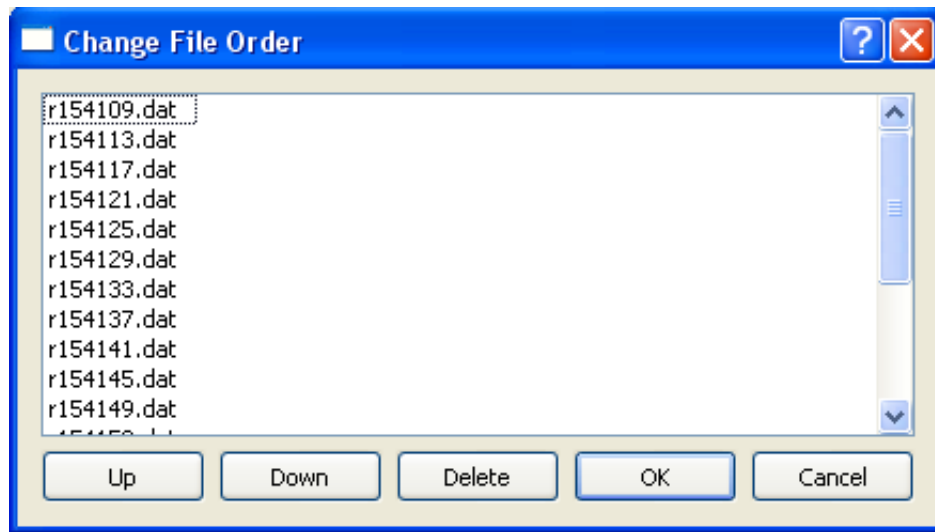


Figure 7: Change File Order dialog.

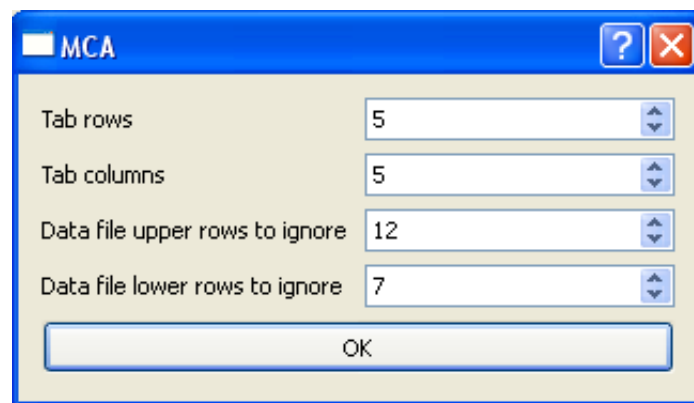



Figure 8: MCA dialog.


- Read ERD (  ): for reading and mapping data files of the ERD detector. Each row<sup>1</sup> in these files corresponds to an event taking place, i.e. a photon being detected by a specific pixel. The data in the rows consist of 5 columns. The first column contains the value in volts of the energy of event. The second column contains the index of the pixel in which the event took place. The pixels are indexed by a 2D array that depicts the physical layout of the detector channels. The third column contains a reference voltage used by the detector electronics. The fourth column identifies the frame in which

<sup>1</sup>The basic description of ERD data is provided by Conny Hansson of University of Manchester.

the event took place. The fifth column contains a Boolean flag for data validation ('1' for valid and '0' for invalid). In the current implementation of this function, the third and fourth columns are ignored. Because the energy is not quantised, a binning process is applied before creating the spectrum. The bin size is determined by the user, and is fed to the program through the map file which is outlined next.

On invoking this function, a dialog appears to enable the user to choose the ERD data file to be read and mapped. This operation requires a map file called 'ERD.map'. The map file contains the following data: number of rows, number of columns and voltage bin size. This is followed, on another text line, by the channels index map which mimics the physical layout of the 2D detector, as seen in Figure 9.

The program ignores all the lines with invalid data, i.e. those whose Boolean flag on column 5 is 0. All the valid lines with the same pixel index (as given on column 2) are then grouped together and the energies (as given on column 1) for each pixel are processed as spectra. The  $x$  values of these spectra are the bin numbers while the  $y$  values are the number of counts, i.e. the number of events in a specific bin. The pixels are then mapped on a tab with the normal colour coding, tool tips, 2D and 3D visualisation, curve-fitting capability and so on, like other data formats. When a channel has less than three events, its cell on the tab is marked with white. The routine is general regarding the detector dimensions provided that a correct map file is supplied.

- Read LUCIA (


---

<sup>2</sup>The basic description of LUCIA data is provided by Olivier Lazzari of Birkbeck College.

```

16 16 0.002      % rows, columns, bin size, followed by the index map
128 136 144 152 160 168 176 184 192 200 208 216 224 232 240 248
129 137 145 153 161 169 177 185 193 201 209 217 225 233 241 249
130 138 146 154 162 170 178 186 194 202 210 218 226 234 242 250
131 139 147 155 163 171 179 187 195 203 211 219 227 235 243 251
132 140 148 156 164 172 180 188 196 204 212 220 228 236 244 252
133 141 149 157 165 173 181 189 197 205 213 221 229 237 245 253
134 142 150 158 166 174 182 190 198 206 214 222 230 238 246 254
135 143 151 159 167 175 183 191 199 207 215 223 231 239 247 255
120 112 104 96 88 80 72 64 56 48 40 32 24 16 8 0
121 113 105 97 89 81 73 65 57 49 41 33 25 17 9 1
122 114 106 98 90 82 74 66 58 50 42 34 26 18 10 2
123 115 107 99 91 83 75 67 59 51 43 35 27 19 11 3
124 116 108 100 92 84 76 68 60 52 44 36 28 20 12 4
125 117 109 101 93 85 77 69 61 53 45 37 29 21 13 5
126 118 110 102 94 86 78 70 62 54 46 38 30 22 14 6
127 119 111 103 95 87 79 71 63 55 47 39 31 23 15 7

```

Figure 9: A sample of an ERD map file for a  $16 \times 16$  detector.

```

27,25,26,27,26,27,29,111,26,26,27,111,25,27,28,27,25,25,26,26
39,36,49,54,39,52,47,52,43,43,51,38,38,42,56,36,41,38,43,38
36,38,49,55,39,50,39,51,44,40,48,40,38,42,52,43,39,38,37,35
39,40,53,61,41,50,41,48,41,43,47,29,43,43,26,68,53,36,43,37
42,40,55,52,40,52,40,55,26,63,57,59,46,42,51,41,53,37,41,38
36,39,51,49,41,75,38,48,40,40,42,37,45,47,42,45,52,39,45,41
43,37,50,49,37,49,47,56,56,42,46,40,43,111,53,45,51,41,38,37
46,36,56,56,49,59,59,38,51,40,42,50,39,57,43,47,41,48,46,41,41
48,37,52,50,40,54,57,45,52,44,39,43,51,42,48,46,58,39,42,37
37,41,50,52,47,48,41,54,44,40,45,39,55,43,53,49,49,42,56,40
40,35,46,49,41,51,43,49,39,42,42,51,52,43,52,50,45,47,44,38
36,39,54,55,42,53,35,47,47,43,25,46,54,43,49,53,53,44,48,40
36,48,53,48,41,60,39,47,45,41,51,49,51,41,48,44,51,45,48,39
38,35,49,49,39,45,39,51,53,42,46,57,46,43,54,50,50,48,41,37
38,43,50,48,39,47,38,45,49,42,51,50,55,39,45,50,52,43,46,37
37,42,54,49,41,51,47,49,49,42,41,48,53,41,54,48,47,55,44,40
37,42,54,53,39,47,50,49,41,43,42,47,43,44,50,49,45,40,50,42
41,36,53,44,43,50,40,49,50,42,40,48,53,38,56,50,45,47,46,40
45,40,46,40,50,52,49,53,58,41,47,50,56,40,51,51,49,45,43,40
37,46,42,46,44,49,54,52,44,44,51,53,47,47,23,47,49,53,50,56

```

Figure 10: A sample of a HEXITEC thresholds file for a  $20 \times 20$  detector.

implicit channel number. The user have the option to display these 960 patterns in a single row or as 32 rows times 30 columns on the EasyDD tab widget.

- **HEXITEC Data:** for reading and mapping data files obtained from the HEXITEC detector of Rutherford Appleton Laboratory. The data<sup>3</sup> from the HEXITEC 2D detector is written as a continuous list of 16 bit values to a binary file. Each file contains a number of frames of data. A single frame contains

<sup>3</sup>The basic description of HEXITEC data is provided by Matt Wilson of RAL.

$N = \mathcal{R} \times \mathcal{C}$  values where  $\mathcal{R}$  and  $\mathcal{C}$  are respectively the number of rows and columns of pixels in the detector. Each pixel in the detector has a different pedestal (offset) level. Therefore, a list of  $N$  pedestal values should be provided in an independent file called ‘pedestals.txt’. Also the noise level on each pixel is different. Therefore, another file called ‘thresholds.txt’ should be provided to define the noise threshold (i.e. upper noise limit) of each pixel. These thresholds may require scaling by the use of a user-defined constant multiplier. To discriminate between the different data file formats, the program recognises the HEXITEC raw data files by the \*.dat extension. The pedestal and threshold values should be positioned in their files in a 2D matrix to mimic the physical layout of the corresponding pixels in the detector. The neighbouring values in each row should be separated by a comma ‘,’. An image of a sample threshold file is shown in Figure 10. The other data are obtained from a dialog, seen in Figure 11. These data include the number of rows and columns, the thresholds multiplier, the number of bins, the minimum and maximum, a Boolean flag to run the end-of-frame algorithm, a Boolean flag to run charge-sharing algorithm followed by another flag for keeping/removing charge sharing events, and a Boolean flag to subtract from pedestals or ignore them. Apart from the \*.dat binary files, all the other input data files should be located in the same directory as the program.

Currently, the program can process the HEXITEC detector data in three modes:

1. HEXITEC Generic Mode (

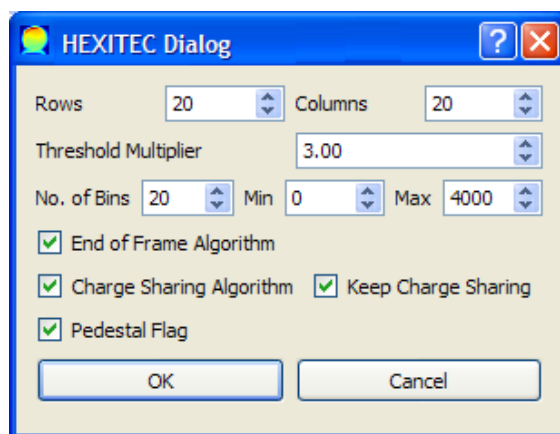




Figure 11: HEXITEC dialog.

initial checks on the raw data. For this mode, only the \*.dat binary file and the number of rows and columns, which are obtained from the HEXITEC dialog, are required.


2. HEXITEC Single Mode (  ): the program reads the raw data from a single file and corrects the individual entries by subtracting these entries from the corresponding pedestals. These corrected data are then compared to the corresponding threshold noise and the entries that fall below the threshold are discarded completely. The data which are below the minimum limit or above the maximum limit are also removed. The remaining data of each pixel are then binned into a histogram according to their values. This histogram, which can be plotted on the 2D plotter like other data formats, represents the number of events as a function of bin number. In addition to the raw data file, the pedestals and thresholds data files are required for this mode of operation. The other data (rows, columns, threshold noise multiplier, number of bins, minimum and maximum limits) are obtained from the HEXITEC dialog. The bins have equal width obtained by dividing the range (i.e. maximum minus minimum) by the number of bins.

3. HEXITEC Batch Mode (

**Note about the end-of-frame algorithm:** when a pixel has an event in frame  $N$  the user may want to ignore that pixel value in frame  $N + 1$ . The end-of-frame algorithm identifies these pixels and applies this correction.


**Note about charge sharing algorithm:** the voltage (signal) generated by a single X-ray photon in the HEXITEC detector can be shared between up to four neighbouring pixels. The reason is that a charge generated by an X-ray photon can be detected by more than one pixel. The purpose of the charge sharing algorithm is to identify and correct such charge sharing events. Charge sharing has two options: either allocate the event to a single pixel (i.e. the one with the largest voltage)

or remove charge sharing by setting the voltage of all pixels involved in charge sharing events to zero and hence removing these events. Charge sharing corrections are conducted after removing the pixel offsets and the end-of-frame corrections, but before the data reduction.

4. HEXITEC Batch Mode with Energy Calibration (

$$E(x, y) = m(x, y)X(x, y) + c(x, y) \quad (1)$$

where  $E(x, y)$  is the calibrated value (in keV) for the pixel in row  $x$  and column  $y$  from the raw data  $X(x, y)$  with linear calibration coefficients  $m(x, y)$  and  $c(x, y)$ . The calibration coefficients are provided in two text files ('m.txt' and 'c.txt') each containing an array of  $m \times n$  values (where  $m$  and  $n$  are the number of rows and columns respectively) similar to the format of the thresholds and pedestals files. Energy calibration is performed after low energy threshold cutting and before charge sharing correction/discrimination. A message appears at the beginning to inform the user about the nature of this function and the sequence of operation. In the case of end-of-frame with charge-sharing, an output file called 'Stat.txt' which contains the number of valid and discarded events is created. It should be remarked that this function requires an input file called 'binNumber2.txt'. This file (which was temporarily introduced for the purpose of testing) contains the number of bins which should equal or exceed the number of bins in the HEXITEC dialog.

5. HEXITEC Batch Mode with Energy Calibration (New Format) (

- SRS 16.4 Data: to read and process data files obtained from station 16.4 of Daresbury Synchrotron Radiation Source. This item contains two functions:

1. Read SRS 16.4 (

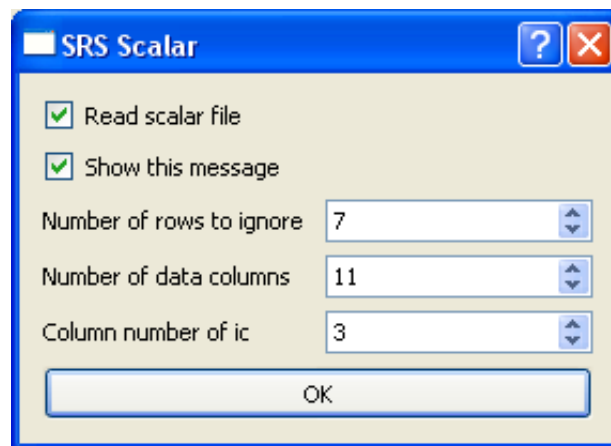



Figure 12: SRS scalar dialog.

2. Show Scalar Message (  ): this launches the scalar dialog (Figure 12).

The purpose of this dialog is to control the scaling of the count rate by the storage ring current ‘ic’. This dialog contains two check boxes: the first is for choosing between scaling to ‘ic’ by reading the scalar files or not, while the second is for determining if this dialog box should appear when invoking ‘Read SRS 16.4’. The dialog also contains three integer spin boxes. The first is for the number of header rows in the scalar data files to be ignored, the second is for the number of numerical data columns in the bottom of the scalar files, and the third is for the ordinal column number of the ‘ic’ data. The ic scaling correction is based on normalising all ic’s in the tab to the ic of the first voxel (1,1) in the tab.

### 3.1.3 CCD Menu

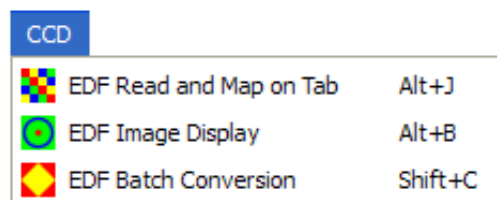




Figure 13: The main window CCD menu.

This menu (Figure 13) is for reading and processing CCD image files of EDF format. These are normal binary files with text headers. The binary data represent the CCD pixel intensity as an implicit function of the pixel position in the CCD array. The header, which occupies the first 24 lines, contains, among other things, information about the binary data type (e.g. long integer), the number of rows and columns in the CCD array, and the endian type. This information is used to read and process the data adequately. Hence, all processes are completely automated as long as the EDF file header contains the required information and has the right format. The CCD menu includes the following items

- EDF Read and Map on Tab : to read and map a number of EDF files. A dialog first appears to enable the user to define the tab dimensions. A default directory named ‘DataEDF’ will be inspected by the program to look for EDF files if a different source directory was not selected by the user. This function can also be used for converting EDF files by mapping them first, followed by invoking ‘Write XY Data Files’ from Data menu. However, this function uses a basic EDF conversion technique (e.g. tilt and other corrections are not considered) and hence it is recommended to use multi-batch numeric conversion from MultiBatch menu.
- EDF Image Display : the purpose of this function is to display an image of a single EDF file (Figure 22) and to identify the centre and radius to be used for extracting the pattern by other processes (i.e. read and map on tab, batch conversion, and multi-batch conversion). The image plotting technique is simple, that is the square of the intensity is plotted at the pixel position according to a single colour coding scheme. The purpose of using the square rather than the intensity itself is to intensify the image features. The centre of the image is indicated by an ‘×’ mark. By default, the position of the centre is taken at the centre of the image rectangle and the radius is zero.

By click-and-drag action, the centre will be moved to the position of click while the radius is taken as the distance between the click and the release positions. A yellow ring is plotted dynamically during the drag action. The 1D pattern will be extracted only from the part of the 2D pattern represented by the pixels inside the yellow ring. When the radius is zero the pixels of the entire image are used to generate the 1D pattern. On clicking on another position within the image, the ring centre is moved to the click position and the ring image is updated. All following batch processes that involve 1D pattern extraction will be based on the centre and radius that are identified by this function. The radius and centre data can be adjusted at any point during the session by invoking this function again and making the necessary amendments. These data will not be saved on closing the program. There are other parameters that can be adjusted within the image display dialog (see ‘EDF Numeric Conversion’ in § 3.1.7). Also, this dialog is launched automatically on invoking the functions that require these EDF parameters.

- EDF Batch Conversion (

quality and speed, because at certain points the quality improvement may not justify the extra cost in processing time and file size. It should be remarked that jpg images require a plugin which if it is not available on the system the jpg production fails. On invoking this function a dialog (Figure 14) appears to enable the user to choose the image type and scale factor and to determine if ASCII text files should be written. These options are independent and hence they can be chosen in any combination including all and none. In the last case the function will terminate with no further action. If numeric conversion is selected the EDF image display dialog (Figure 22) will appear. In this case the program will remind the user about the image radius if the default value (i.e. 0) is used. It is noteworthy that the ASCII numeric conversion uses a basic technique and hence it is recommended to use ‘EDF Numeric Conversion’ from the MultiBatch menu, which can perform numeric conversion even on a single data set.

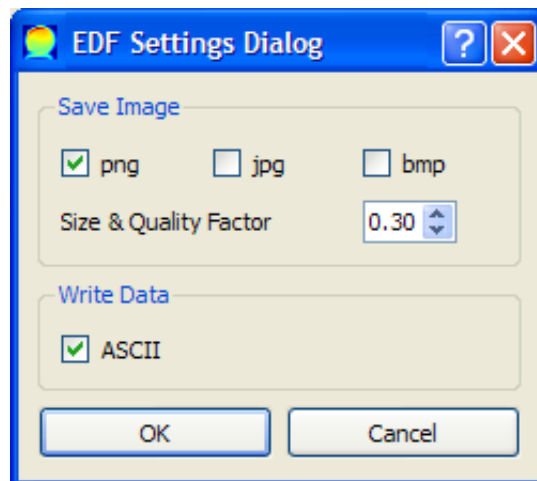



Figure 14: EDF settings dialog.

### 3.1.4 Settings Menu

This menu (Figure 15) includes

- Tooltip and Cell Settings (⚙️): to launch the Settings dialog which will be

explained in § 9.1.

- Grid and Colour Settings (  ): to launch the Tab Settings dialog (Figure 16) which controls the settings of tabs. This dialog contains three check boxes: ‘Grid View’ to turn the grid on/off for all tabs, ‘Single Colour’ to switch between single- and multi-colour coding schemes for all tabs, and ‘Colour Code Base’ to select a colour coding scheme that is based on a certain minimum base value chosen by the user. This base value can be adjusted from the double spin box next to this check box. When ‘Colour Code Base’ box is unchecked, the colour coding scheme will be based on the minimum of all values in the tab. It should be remarked that ‘Grid View’ and ‘Single/Multi Colour Coding’ tool buttons (see § 3.2.2) in the main window can be used to adjust the settings of the individual tabs independently.

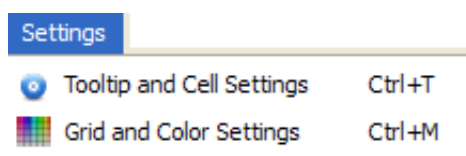


Figure 15: The main window Settings menu.

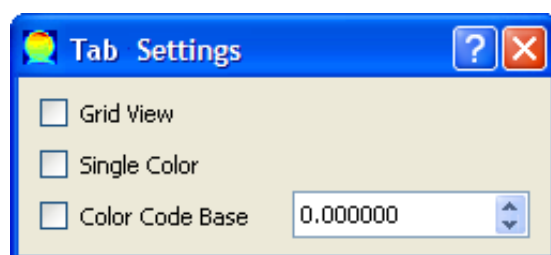


Figure 16: Tab settings dialog.

### 3.1.5 Fitting Menu

This menu (Figure 17) facilitates curve-fitting and writing parameters. In addition to the possibility of fitting all cells in a single tab or multiple tabs, it is possible






Figure 17: The main window Fitting menu.


to fit selected cells of a tab or tabs. With regard to batch curve-fitting, there are three ways of selecting/deselecting individual cells for batch curve-fitting:

1. 'Ctrl' + left mouse click to select and deselect a cell.
2. 'Shift' + two left mouse clicks to select a rectangular range of cells defined by the two clicked cells as diagonal corners.
3. 'Alt' + mouse hovering to select/deselect the cells that are hovered on.

The selected cells for fitting are marked with green colour. The keyboard 'Esc' button can be used to deselect the green cells.

Fitting menu contains the following items

- Load Form (): to load a spreadsheet form (see § 6) that defines and controls the curve-fitting process.
- Batch Fit Green in Current Tab (): to curve-fit all selected (green) cells in the current tab.
- Batch Fit Green in All Tabs (): to curve-fit all selected (green) cells in all tabs.

- Batch Fit All in Current Tab (

### 3.1.6 Sinograms Menu

This menu (Figure 18) includes

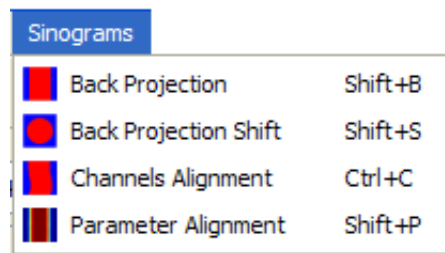



Figure 18: The main window Sinograms menu.

- Back Projection<sup>4</sup> (): to perform back projection algorithm on the current tab which represents a sinogram of rotation versus translation data. The algorithm works only when the rotational dimension is less than or equal to the translational dimension. On calling this function a dialog (Figure 19) will appear. In this dialog there is a ‘Fourier’ section which contains a check box to choose performing a Fourier transformation or not, and two radio buttons to choose fast (FFT) or discrete (DFT) transformation<sup>5</sup>. This is followed by a ‘Data Setting’ section which includes two radio buttons to choose the sinogram type, i.e. rotation (rows) versus translation (columns) or vice versa. This is followed by an ‘Angles’ section which contains two double spin boxes to define the start and the end angles in degrees. The rotation angles are assumed to be equally spaced between these two limits with the angle step size being equal to the range (i.e. difference between the limits) divided by the number of rotations. This is followed by a ‘Read File’ check box, with a neighbouring line edit to enter the name of the angles file. This option is used when the angles are not equally spaced. In both cases the number of angles is assumed to be equal to the rotational dimension. This section is followed by a ‘Channels’ section with three radio buttons to choose one of three options: ‘Back projection for Individual Channels’, ‘Back

<sup>4</sup>Simon Jacques contributed to the implementation of this algorithm.

<sup>5</sup>The DFT algorithm is a contribution of Simon Jacques. FFT and DFT produce different results.

Projection for ‘Total Intensity’ and ‘Back Projection for Partial Intensity’.

The first option performs a back projection on all common channels of all spectra in the tab individually. On performing this operation a vertical line at the position of the mouse pointer will be added to the 2D plotter (see § 4). On hovering the pointer over the plotter the tab will be updated to display the back-projected intensity of the current channel, i.e. the channel at the pointer position. On moving the pointer off the plotter, the tab will continue displaying the intensity of the last visited channel on the plotter. The second option will perform a back projection once on the total intensity (i.e. the sum of intensities of all channels) of the tab spectra. Similarly, the third option will perform back projection once on the partial intensity (i.e. the sum of intensities of some channels) in a certain channels range. The start and the end of the channel range are identified by the two integer spin boxes below the third radio button. Finally, there are two push buttons: ‘OK’ to proceed and ‘Cancel’ to abort the operation.

- Back Projection Shift (

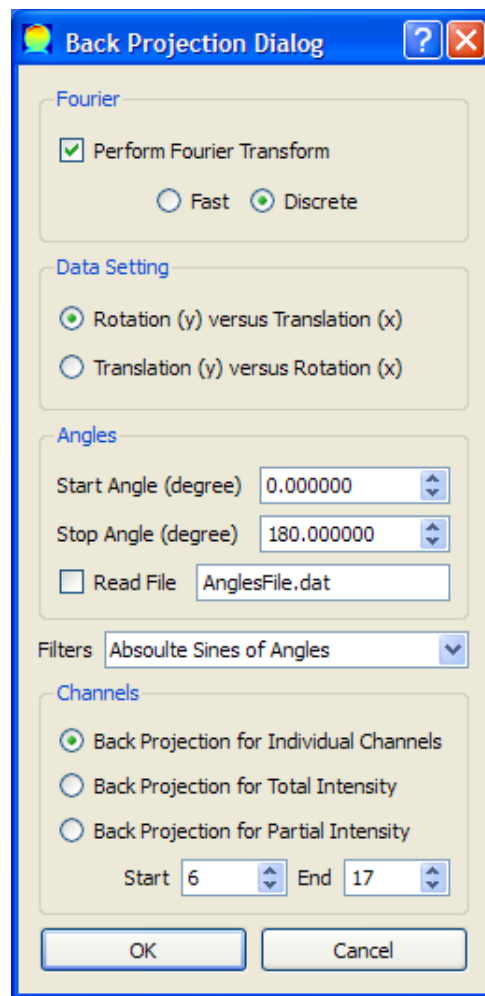


Figure 19: Back projection dialog.

alignment parameter between the corresponding cells in the current row and the reference row. The final alignment of the row is that of the least squares. The algorithm requires applying curve-fitting on all voxels in the tab beforehand. On invoking this function a dialog (Figure 20) appears. This dialog contains a number of spin boxes to identify the algorithm parameters. These are: inflation factor (number of divisions of each cell), displacement number (number of divisions to be displaced left and right), parameter index (index of fitting parameter in the fitting form to be used for alignment), and upper cut-off limit (maximum value for the intensity to be set to when it exceeds that limit). The inflation factor should be between 1 and 50 inclusive. The

displacement number should be between zero and a number of divisions occupying half the tab width to avoid computational waste. The purpose of the upper cut-off limit is to obtain an alignment image with sharp features. It should be remarked that this algorithm does not conserve the intensity of spectra channels as the values of the inserted divisions will be added to the intensity of the related cells.

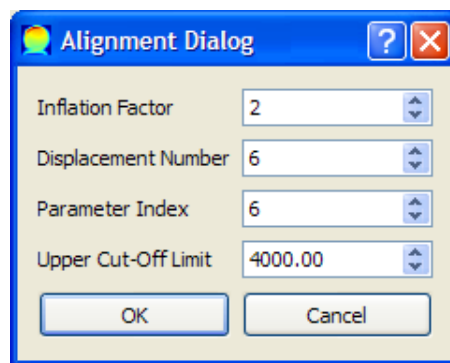



Figure 20: Alignment dialog.

- Parameter Alignment (  ): this function is similar to the previous one but the alignment is performed on the total intensity, and the aligned intensity is displayed on the tab. This can help in testing the algorithm with a certain set of alignment parameters before applying channels alignment.

### 3.1.7 MultiBatch Menu

The items in this menu (Figure 21) share a number of common features. First, they work on a number of data sets (including a single set) each one of which is stored in a particular directory. Second, they look to a default directory called ‘MultiBatch’ as the source of data where the data sets are stored unless the user selected another directory by using ‘Change Data Directory’ from Data menu. Third, they work on an *xy* file format except ‘Curve Fitting’ which works on a number of formats including *xy*, and EDF functions which by nature work on EDF files. The files used by the ‘Time Sum’ function have a single user-defined

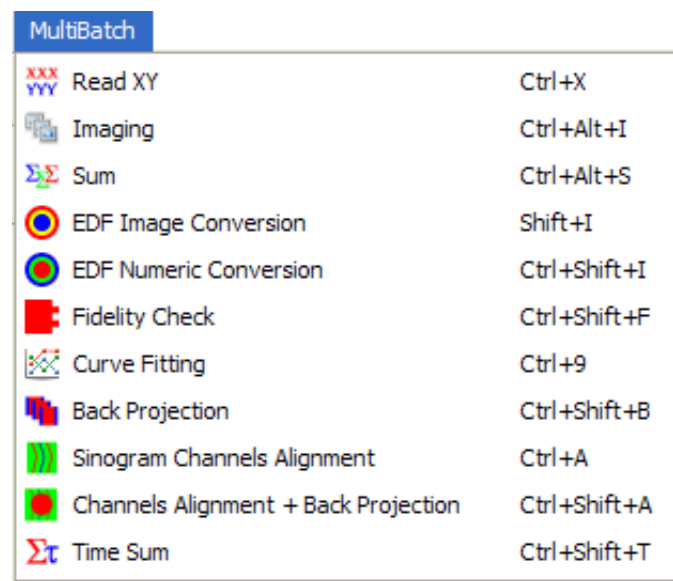


Figure 21: The main window MultiBatch menu.

name, and hence are not identified by type. Another common feature is that MultiBatch functions store the processed data either in the source directory or in its subdirectories in an organised fashion using meaningful names to facilitate managing and post-processing of results. This menu includes the following items

- Read XY (): to read a number of data sets and map them on separate tabs. A maximum of 10 data sets are read in each run if the source directory contains more than 10.
- Imaging (): to produce png images representing the total intensity of spectra, similar to what the tabs display. The dimensions (rows and columns) of the images are required when invoking this function. No image will be produced for a data set if the number of data files in its directory is less than the product of the two dimensions. The image files are saved to the source directory, each with a name identical to the name of the data directory of that set with a '.png' extension.
- Sum (): to produce sum spectra by adding the intensity of the correspond-

ing channels. There is no restriction on the number of files in each data set. The number of channels in the sum spectrum is the maximum of that of the individual spectra in the data set. This function is based on the assumption that all common channels of the data set share the same  $x$  coordinates. The sum files are saved to the source directory, each with a name identical to the name of the data directory of that set with ‘\_Sum’ extension.

- EDF Image Conversion (

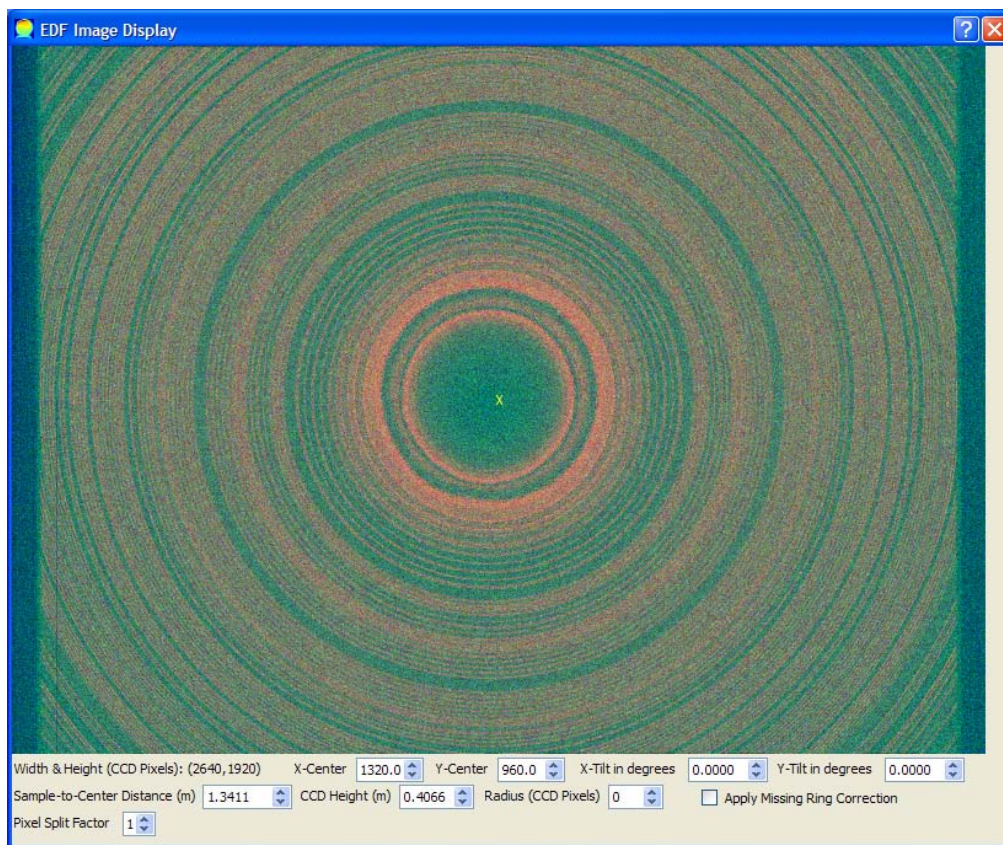


Figure 22: EDF image display dialog.

fidelity check on the text files.

The EDF image display dialog contains a number of widgets:

1. Label to display the width and height of the EDF image in pixels.
2. Two spin boxes to define the  $x$  and  $y$  coordinates, which correspond to the horizontal and vertical dimensions respectively, of the centre of the image rings in pixels. By default, these coordinates are set at the centre of the EDF rectangle at the start of the program. The centre is indicated on the image by a yellow 'x' mark which moves to the new position dynamically when the coordinates are adjusted.
3. Two spin boxes to define the  $x$  and  $y$  tilts in degrees (see § 10) within the range  $-89^\circ$  to  $89^\circ$ .

4. Spin box for the distance in metres between the sample and the centre of the CCD detector.
5. Spin box for the height (i.e. vertical dimension as displayed on the dialog) in metres of the CCD detector.
6. Spin box for the radius (in pixels) of the circle that identifies the part of the EDF image to be used for numeric conversion, i.e. only the rings inside the circle will be used for extracting the pattern. The default value for the radius at the start of the program is zero meaning that the whole image will be used. A yellow ring representing the circle is displayed on the dialog when the radius is greater than zero. This ring is dynamically resized on adjusting the value of the radius from the spin box.
7. Check box for applying the missing-rings correction based on the continuum approximation algorithm (see § 11) which scales the incomplete rings to the continuum value at that radius.
8. Spin box to define a split factor for the pixels within the range 1-5. This factor was introduced because the pixel can be too coarse as a unit of length and storage due to its finite measurable size resulting in uneven distribution of intensity in the data points of the 1D pattern. The pixels of the CCD grid can therefore be split by using this factor to improve the resolution and obtain smoother patterns.

The settings of some of these widgets are saved on exiting the program. It should be remarked that the EDF image display dialog has tooltips showing the  $x$  and  $y$  coordinates (in pixels) of the mouse pointer position. Moreover, the centre of the image can be moved to a new position by mouse click, and hence the yellow ring is moved and the image centre spin boxes are updated.

The radius can also be adjusted by a mouse click-and-drag action, and hence the radius spin box will be updated.

- Fidelity Check (

On starting multi-batch curve-fitting, a dialog (Figure 23) appears, in which there are three radio buttons to choose the data type, i.e. ‘SRS 16.4’, ‘XY’ and ‘MCA’. The second choice requires the number of rows and columns, while the third requires this plus the number of headers and footers to be ignored. These data are input through the integer spin boxes next to the corresponding radio button. At the bottom of the dialog there are three push buttons: ‘Select Data Directory’ to identify the source directory, ‘OK’ to proceed, and ‘Cancel’ to abort the process. Next to ‘Cancel’ button there is a check box and two integer spin boxes. The check box is used to select formatting the data in the written files by fixing the width of columns and the number of decimal places to be displayed. These values are input through the adjacent spin boxes.

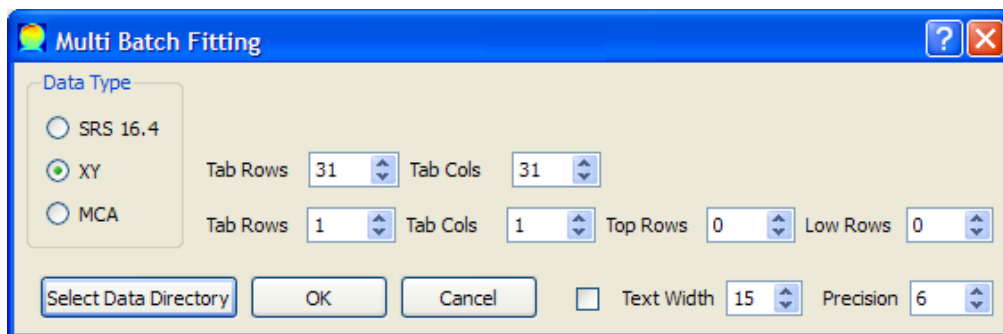



Figure 23: MultiBatch curve-fitting dialog.

- Back Projection (  ): to perform multi-batch back projection in one of its three modes. The dimensions of the data sets and the parameters and options of back projection are determined through dialogs that appear sequentially on invoking this function. Any data set that contains insufficient number of files will be ignored. For back projecting individual channels, the resulting data files are saved to a directory having the same name as the parent directory with ‘\_BP’ extension and located inside this directory. The files are named ‘BP\_RrCc.txt’ where  $r$  and  $c$  are the row and column indices. For

back projecting total or partial intensity, the resulting file of each data set is saved to the main source directory with a name identical to the name of the directory of that data set with a ‘\_BP’ extension.

- Sinogram Channels Alignment (

the common name of the files (e.g. ‘Par3’) whose data will be summed. The user will also be offered the option to normalise each sum to the maximum value within the particular sum file.

### 3.1.8 Help Menu

This menu (Figure 24) includes only one item:

- About (📖): a dialog (Figure 25) is launched when invoking this function. This dialog briefly defines EasyDD, offers contact information, and provides a link to access EasyDD documentation which is stored in the ‘Documentation’ directory. The latter should be a subdirectory of the program directory for normal operation.

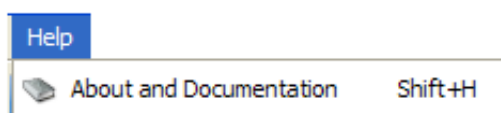


Figure 24: The main window Help menu.

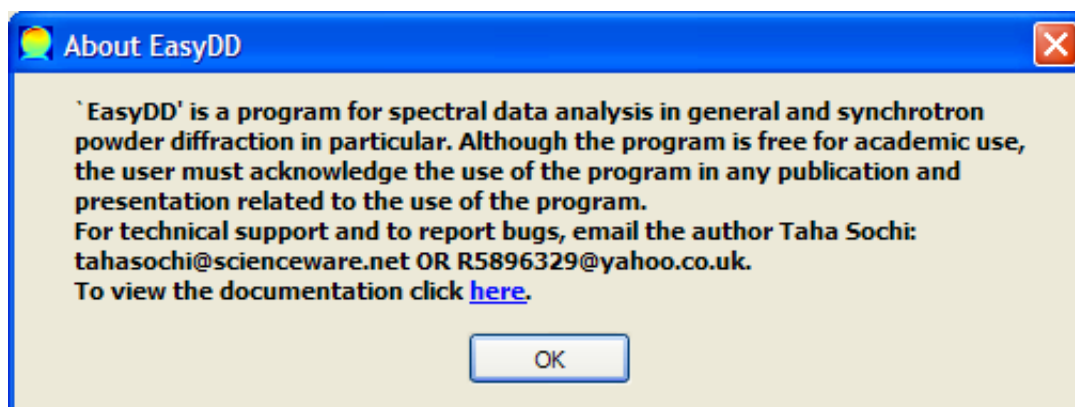


Figure 25: ‘About EasyDD’ dialog.

## 3.2 Tools


These include toolbars, tool buttons, spin boxes, sliders, context menus, and a status bar.

### 3.2.1 Toolbars

Most menus have associated toolbars, as seen in Figure 1. These toolbars can be turned on and off by right-clicking on the toolbar area and choosing from a menu. Each toolbar contains the main items of the corresponding menu. By default, all toolbars are displayed when starting the program. The selected setting will not be saved on exiting the program.

### 3.2.2 Tool Buttons

The main window contains three tool buttons (Figure 1):

- Grid View Button (

### 3.2.3 Spin Boxes

The main window contains four spin boxes (Figure 1):

- Width spin box: to change the width of the tab widget window within the range 0-2000 pixels.
- Height spin box: to change the height of the tab widget window within the range 0-2000 pixels.
- Colour code spin box: to select the colour coding range scheme (see § 3.2.4).
- 3D Graph spin box: to scale the intensity on the  $z$ -axis of the 3D graph (see § 5).

### 3.2.4 Sliders

The main window contains two sliders (Figure 1). The purpose of these sliders is to adjust the minimum and the maximum of the colour code range. Currently there are two available schemes which can be selected from the neighbouring integer spin box

1. '1' (default) to adjust the values of the cells that fall outside the new range so that they have the value of the adjacent limit.
2. '2' to eliminate these cells from the colour coding scheme and hence mark them with white.

### 3.2.5 Context Menu

The context menu (Figure 1) can be accessed by a right-click on the main window outside the tab widget area. This menu contains a number of items from all menus.

### 3.2.6 Status Bar

This is a standard bar (Figure 1) to display explanatory comments about the item that is currently hovered upon by the mouse.

### 3.2.7 Colour Bar

The colour bar (Figure 26) appears on reading and mapping data on tabs. The bar displays the minimum and maximum colour values of the current tab, whether the tab displays the intensity or the residuals or refined parameters after batch curve-fitting. These values are instantly updated on changing the tab selection and on moving the colour sliders to adjust the colour range. They are also updated when changing the displayed parameter by using the list widgets (see 3.4) following batch curve-fitting.

### 3.3 Tab Widget

This is a widget (Figure 1) that can accommodate a number of 2D colour-coded scalable tabs (Figure 26) for data mapping with graphic and text tooltips to show all essential file and voxel properties. The tabs also have a context menu with 12 items for managing rows, columns and cells in the tab. The tab widget space is a scroll area where a horizontal/vertical scroll bar emerges if the width/height of the current tab exceeds the width/height of the tab widget. On changing tab selection, the size of the tab automatically adjusts to fit the current tab.

When a tab cell is double-clicked, the 2D plotter (see § 4) is launched, if it is not already launched, and the pattern of the clicked cell is plotted. This cell is marked with white, but the ‘Esc’ key can be used to recover its original colour. The plotter is dynamically updated when the mouse pointer hovers over the tab so that the plotter displays the pattern of the hovered cell. By pulling the pointer outside the tab, the plotter displays the pattern of the double-clicked cell. The plotter is automatically closed when the tab widget is cleared by removing all tabs.

The tab graphic tooltip is an image of the pattern of the hovered cell while the text tooltips include its file and voxel properties, as well as the residuals and refined parameters if curve-fitting is performed. The scheme for the tab tooltips is that as long as the plotter is absent the mode of tooltips (i.e. text or graph) is determined by user choice (see § 9.1). However, when the plotter is launched the tooltips automatically become text regardless of the user choice because a graphic tooltip is redundant.

### 3.4 List Widgets

List widgets (Figure 26) are created on performing batch curve-fitting on a tab, i.e. one list widget for each refined tab. These widgets contain items which include intensity, statistical indicators and fitting parameters. On selecting each one of

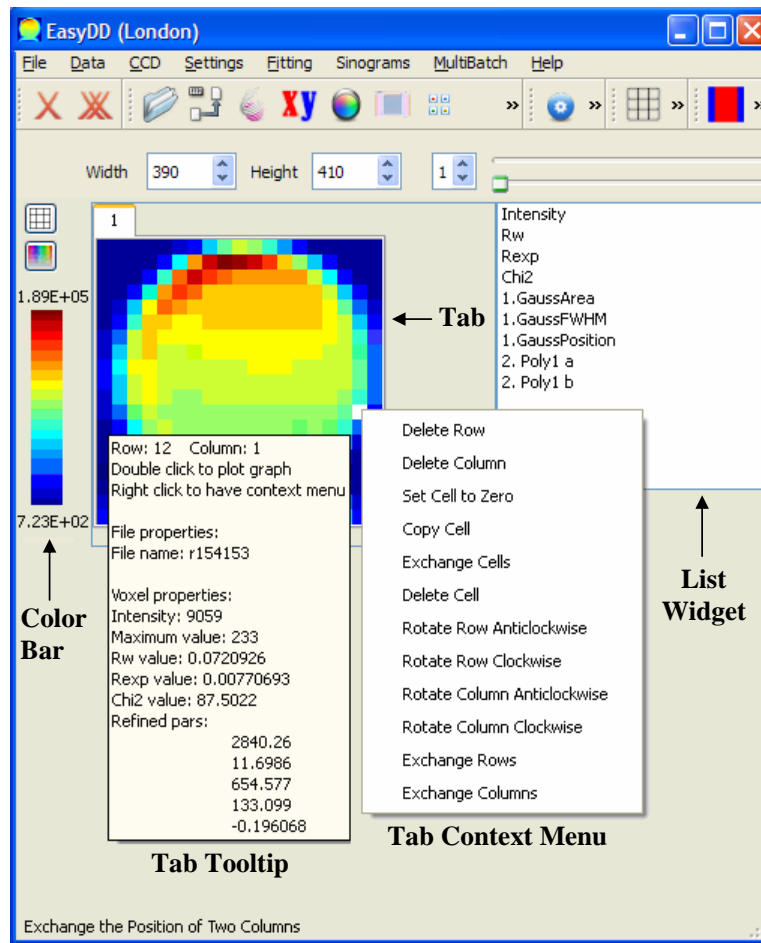


Figure 26: EasyDD main window following data mapping and curve-fitting.

these items, by mouse or by up or down arrow keys, the current tab colour coding and colour bar change to display the variation of the selected item.

## 4 2D Plotter

This is a plotter (Figure 27) for obtaining a 2D graph of a spectrum (e.g. intensity versus energy) for any voxel in the current tab. It is also used to create basis functions and spreadsheet forms for curve-fitting. The plotter capabilities include:

- Creating, drawing, modifying and clearing fitting basis functions by simple click or by click-and-drag actions. The available basis functions are poly-

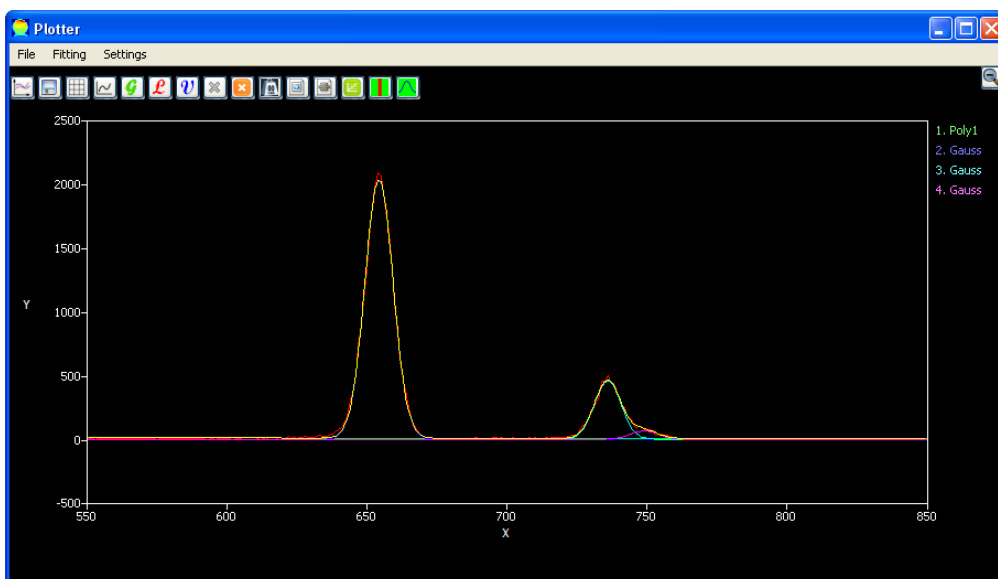


Figure 27: The 2D plotter.

nomials  $\leq 6$ , Gauss, Lorentz and pseudo-Voigt. The last three are given in Table 1.

- Performing curve-fitting on the plotted pattern by a Levenberg-Marquardt nonlinear least squares algorithm.
- Saving image of the plotted pattern in 8 different formats, i.e. png, bmp, jpg, jpeg, ppm, xbm, xpm and pdf.

Table 1: Some of the common shape functions used in EasyDD to describe individual peak profiles.  $A$  is area under peak,  $F$  is FWHM,  $X$  is position of peak and  $m$  is a dimensionless mixing factor ( $0 \leq m \leq 1$ ).




Function	Equation
Gaussian	$G = \frac{2A}{F} \sqrt{\frac{\ln(2)}{\pi}} e^{-\frac{4\ln(2)(x-X)^2}{F^2}}$
Lorentzian	$L = \frac{2A/(F\pi)}{1+4(x-X)^2/F^2}$
pseudo-Voigt	$V = mL + (1 - m)G$

## 4.1 Menus

The plotter has three menus:

### 4.1.1 File Menu

This menu (Figure 28) contains three items:

1. Save image (  ): to save the plotter image in one of 8 formats.
2. Remove basis curves (  ): to remove the basis fitting curves and keep the refined curve.
3. Remove all curves (  ): to remove all curves including the refined curve.

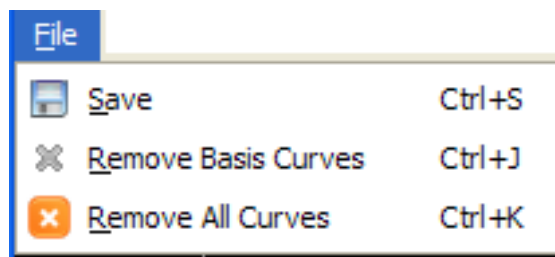


Figure 28: The Plotter File menu.

### 4.1.2 Fitting Menu

This menu (Figure 29) contains several items and subitems:

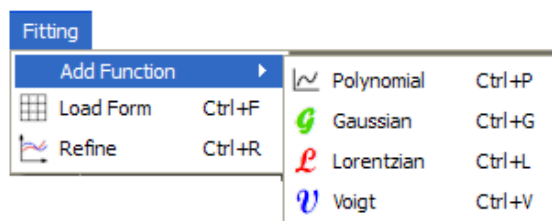




Figure 29: The Plotter Fitting menu.

1. Add Function: this item includes:

- (a) Polynomial (- 2. Load Form (

### 4.1.3 Settings Menu

This menu (Figure 30) contains six items:

1. Statistical weight (

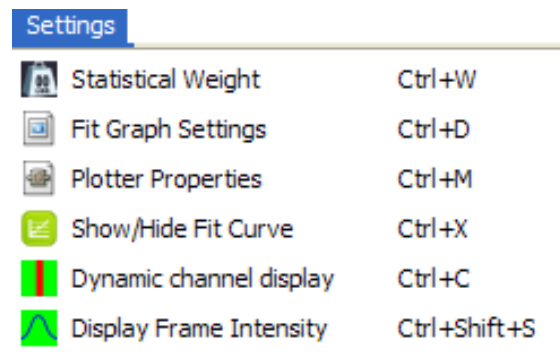


Figure 30: The Plotter Settings menu.

radio buttons: a constant which is in the range  $1-10^9$  and can be adjusted from a double spin box, a Poisson distribution with variance being equal to the observed count rate, and a reciprocal Poisson where the statistical weight equals the observed count rate.

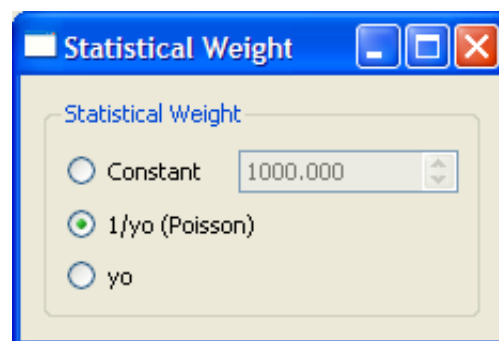


Figure 31: Plotter statistical weight dialog.

2. Fit Graph Settings (

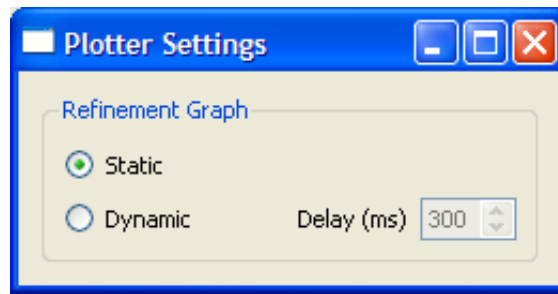


Figure 32: Plotter settings dialog.

which contains three main components:

- (a) Check box to show/hide plotter tooltips, which are the  $x$  and  $y$  coordinates of the mouse pointer position.
- (b) Another check box to convert the channel number for SRS 16.4 data in the plotter tooltip to energy using a linear scaling relation. The ‘a’ and ‘b’ coefficients, which are in the range  $-10^8$  to  $10^8$ , stand for slope and intercept respectively and can be adjusted from the neighbouring double spin boxes.
- (c) ‘Change Background Colour’ push button to change the plotter background colour. Three colours are available: black, white, and dark gray. The colour cyclically changes on pressing this button.

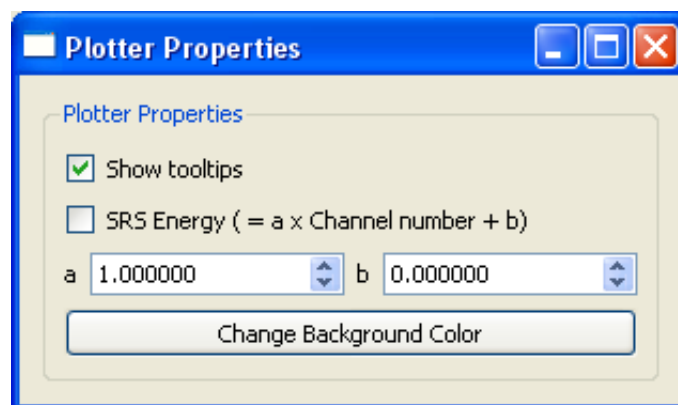




Figure 33: Plotter properties dialog.

4. Show/Hide Fit Curve (

The default of the last two functions when launching the program is to be off. Invoking these functions henceforth toggles their functionality on and off. Currently, these two functions work independently so they can be both on or off or in a mixed state.

## 4.2 Buttons

The plotter contains seventeen tool buttons; fifteen are for accessing the items of the menus and are included for user convenience. The other two, which are located on the right of the plotter, are

1. Zoom out (

It is noteworthy that the  $xy$  frame of the plotter is fixed on zooming in so that the displayed pattern on hovering the mouse pointer will be within the zoomed

$xy$  range. The dynamic adjustment to accommodate the whole  $xy$  range of the pattern will be resumed on zooming out to the initial state before zooming in.

### 4.3 Labels

Associated with each fitting basis curve is a sensitive label (coloured as its curve) on the far right margin of the plotter. On clicking the label, by left or right mouse buttons, a dialog corresponding to the basis curve is launched. From this dialog the curve parameters can be adjusted using the relevant input boxes and hence the plotter is instantly updated. The dialog also contains an image of the fitting curve equation and two push buttons: one to clear the curve and the other to close the dialog. An example of these dialogs is presented in Figure 34.

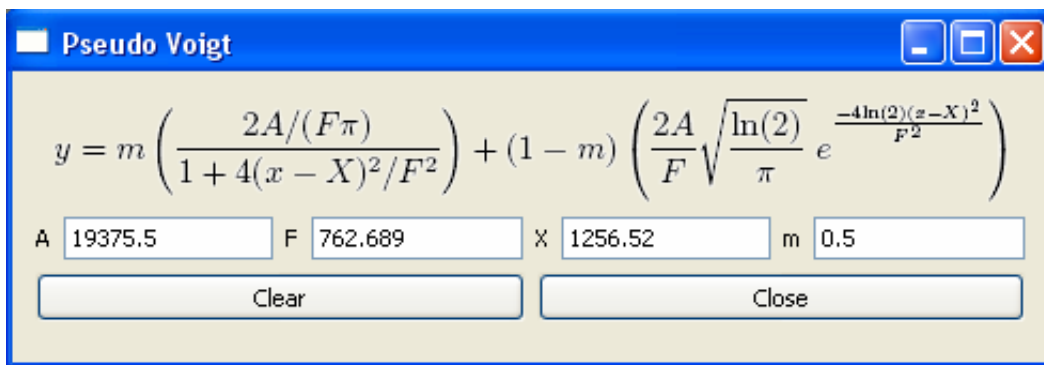


Figure 34: A pseudo-Voigt dialog.

## 5 3D Plotter

This plotter (Figure 35) is used to create a 3D graph of the current tab where the  $x$  and  $y$  axes stand for the number of rows and columns respectively, while the  $z$ -axis represents the voxel intensity. The plot can be resized, zoomed in and out, and rotated in all orientations. To keep proportionality of the plot dimensions, the intensity is automatically scaled by a scale factor which is displayed beside the tab number. No more than one plotter can be launched for any tab. The plotter is

closed when its tab is removed. Moreover, the label of the 3D plot is updated when the corresponding tab number changes by removing a tab. The spin box next to the 3D graph button on the main window is for scaling the  $z$ -axis. The range of this scale factor is between 0.1 and 100, with a step size of 0.1. On performing batch curve-fitting in one of its various modes the residuals and the refined parameters can also be visualised by the 3D plotter although the scaling may not be adequate.

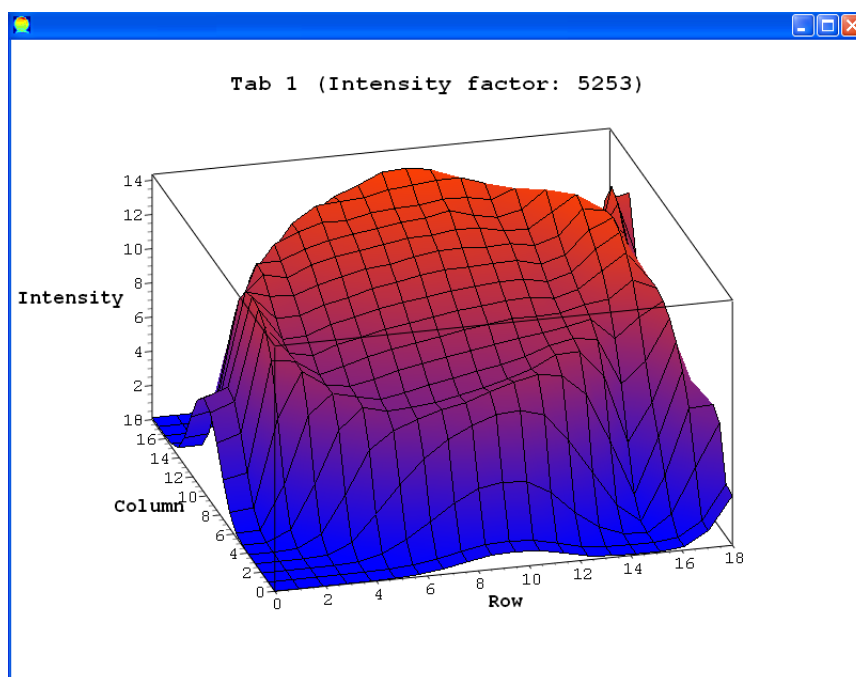
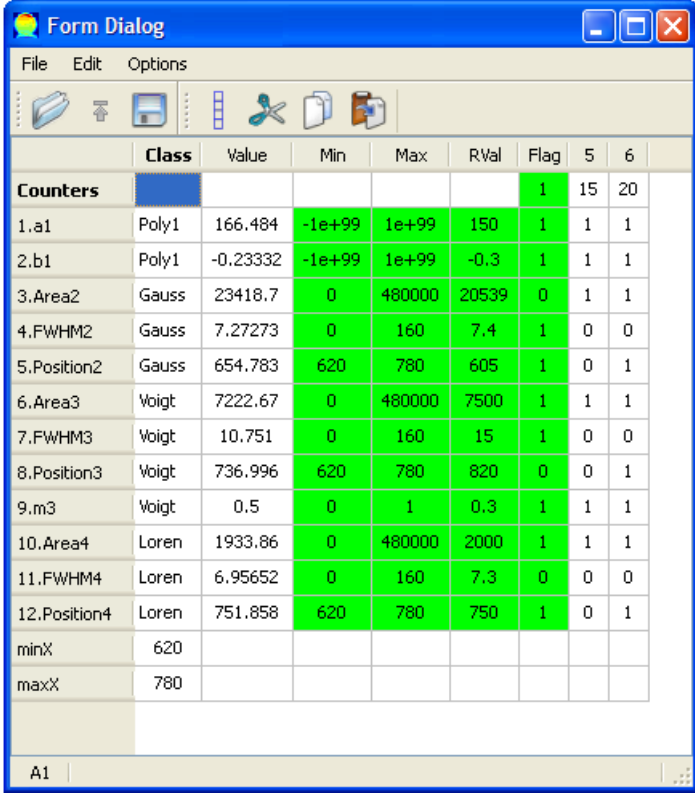


Figure 35: A 3D graph of total intensity as a function of voxel position in a tab.

## 6 Form Dialog

The main use of spreadsheet forms is in large-scale curve-fitting operations. The idea is to prepare a form within the plotter and save it to the disc. It is then imported to the main window for batch fitting a number of cells or tabs in the tab widget, or for multi-batch fitting of a number of data sets. Importing a previously prepared form to the plotter can also be used for curve-fitting a single pattern since the basis functions and frame settings, which include zoom view and  $xy$  ranges,

are reserved in the form. When the form dialog (Figure 36) is launched from the plotter, the dialog interacts with the plotter, i.e. it automatically lists the existing basis functions in the plotter. Moreover, the plotter is updated on modifying the parameters of the basis functions in the dialog.



The screenshot shows a window titled 'Form Dialog' with a menu bar (File, Edit, Options) and a toolbar. Below the toolbar is a table with the following data:

	Class	Value	Min	Max	RVal	Flag	5	6
<b>Counters</b>						1	15	20
1.a1	Poly1	166.484	-1e+99	1e+99	150	1	1	1
2.b1	Poly1	-0.23332	-1e+99	1e+99	-0.3	1	1	1
3.Area2	Gauss	23418.7	0	480000	20539	0	1	1
4.FWHM2	Gauss	7.27273	0	160	7.4	1	0	0
5.Position2	Gauss	654.783	620	780	605	1	0	1
6.Area3	Voigt	7222.67	0	480000	7500	1	1	1
7.FWHM3	Voigt	10.751	0	160	15	1	0	0
8.Position3	Voigt	736.996	620	780	820	0	0	1
9.m3	Voigt	0.5	0	1	0.3	1	1	1
10.Area4	Loren	1933.86	0	480000	2000	1	1	1
11.FWHM4	Loren	6.95652	0	160	7.3	0	0	0
12.Position4	Loren	751.858	620	780	750	1	0	1
minX	620							
maxX	780							

At the bottom of the dialog, there is a status bar with the text 'A1'.

Figure 36: Form dialog.

The form dialog consists of a number of menus, toolbars and a status bar similar to those in the main window. The dialog also has a number of columns:

1. A gray column for labelling and classifying the parameters of basis functions. Each label starts with an ordinal index followed by the name or abbreviation of the parameter, followed by the index of the function to which the parameter belongs.
2. A column, labelled 'Class', in the white area for classifying the basis functions, i.e. 'Gauss' for Gaussian, 'Loren' for Lorentzian, 'Voigt' for pseudo-Voigt, and

‘Polyn’ for polynomial of order  $n$ . In the end of this column the  $x$  range (i.e. minimum and maximum) of the form is displayed. The single, batch and multi-batch curve-fitting operations are performed only on the pattern data within this range.

3. Another column, labelled ‘Value’, for the initial values of the parameters. These initial values are automatically changed to the refined values following a single curve-fitting operation in the plotter.
4. Four green columns for restrictions. The idea of restrictions is to set the refined parameters to certain values when they exceed predefined lower or upper limits. The first of these columns ‘Min’ is for the lower limit. The second column ‘Max’ is for the upper limit. The third column ‘RVal’ is the restriction value which the parameter should be set to when it exceeds one of the two limits. The fourth column ‘Flag’ is for the Boolean flags. At the top of this column there is a global flag for the restrictions to be applied (when it is ‘1’) or not (when it is ‘0’). The other entries in the column are the individual flags for each parameter. The restriction on a certain parameter is applied only when both the global flag and the flag for that particular parameter are true (i.e. ‘1’). The default values for these four columns are shown in Table 2. The individual flag of a particular parameter can be set/reset (i.e. toggle between ‘0’ and ‘1’) by double-clicking the corresponding cell. Double-clicking the cell of the global flag will set/reset the whole column of individual flags. The value of the global flag can be modified by typing the value directly inside its cell. This can also be used for modifying the individual flags.
5. Column(s) for counter(s) and Boolean flags for the least squares (LS) fitting routine. These columns can be added by the user to fix the number of LS cycles and determine which parameters should be refined. Hence, these columns

are absent when the form is created. On the top of each column there is a counter of positive integer type. This counter determines the number of iteration cycles in the LS routine for that set of Boolean values. Double-clicking the counter cell will set/reset the whole column of Boolean flags, i.e. they toggle between 0 and 1. The Boolean values can also be changed individually either by typing directly in the cell, or by double-clicking the cell. A '0' flag means the corresponding parameter is held constant during the LS routine, while a '1' flag means the parameter will be refined and adjusted in the LS routine. The default values for the counter and Boolean flags are 1 and 0, respectively. The number of cycles in the LS routine is fixed when these columns exist, and this number equals the sum of counters. When these columns are absent, the number of cycles will vary depending on the pattern, basis functions and initial values of fitting parameters. In this case, LS cycles continue until convergence is achieved within a certain marginal error. Moreover, all parameters will be refined in the LS routine.

Table 2: The default values of the form restrictions parameters for Gauss, Lorentz, pseudo-Voigt and polynomial basis functions. The default value for the global restriction flag is '0'. The 'Plotter' in the table refers to the frame of the plotter, as viewed by zooming in and out, when the form was created.






	Min	Max	RVal	Flag
Area	0	Plotter area	Initial value	0
FWHM	0	Plotter width	Initial value	0
Position	Plotter lower limit	Plotter upper limit	Initial value	0
$m$	0	1	Initial value	0
Poly Coefficients	$-10^{99}$	$10^{99}$	Initial value	0

## 6.1 Menus

The form dialog contains three menus: File, Edit and Options.

### 6.1.1 File Menu

This menu (Figure 37) contains five main items:

1. Load Form (  ): to load a form file to the dialog. The dialog recognises form files of \*.sp type only.
2. Unload Form (  ): to unload a form from the dialog and erase it from memory.
3. Save (  ): to save a modified form to its \*.sp source file.
4. Save As (  ): to save a form to a new \*.sp file.
5. Exit (  ): to exit the form dialog and return to the parent widget, i.e. main window or 2D plotter.

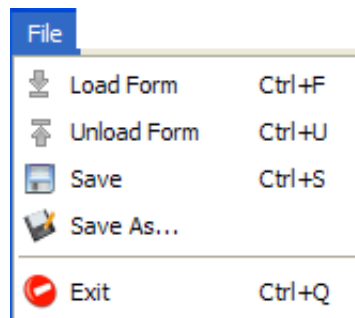


Figure 37: The form dialog File menu.

This menu may also contain a number of variable items. These are the names of the most recently loaded \*.sp files (up to five) made available for quick load. These items appear, when they exist, between ‘Save As’ and ‘Exit’.

### 6.1.2 Edit Menu

This menu (Figure 38) contains six items:

1. Add Column (  ): to add a column for LS counter and Boolean flags.

2. Cut (✂️): to cut the content of selected cell(s).
3. Copy (📄): to copy the content of selected cell(s).
4. Paste (📄): to paste the cut or copied contents.
5. Delete (ⓧ): to delete the content of selected cell(s).
6. Select: this item contains:
  - (a) Row: to select the row of the current cell.
  - (b) Column: to select the column of the current cell.
  - (c) All: to select all cells in the form.

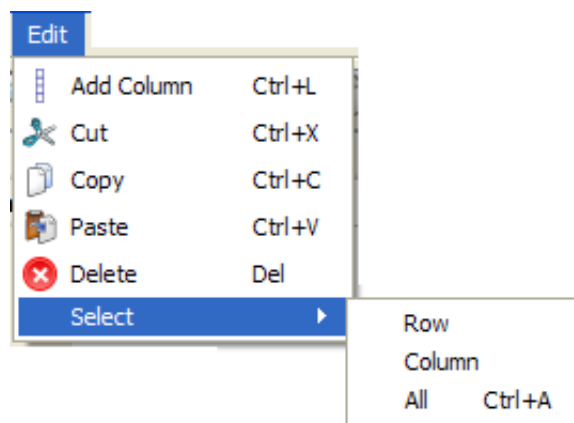


Figure 38: The form dialog Edit menu.

### 6.1.3 Options Menu

This menu (Figure 39) contains one item:

1. Show Grid: to turn the grid on and off.

The File and Edit menus have associated toolbars (Figure 36) which can be turned on and off by right-clicking the toolbar area and selecting the option. Each

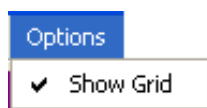


Figure 39: The form dialog Options menu.

toolbar contains the main items of its menu. By default, all toolbars are displayed when starting the program, and their settings are not saved on exit. The form dialog has also a context menu which contains the items of frequent use. The context menu can be accessed by right-click on the form main area.

## 7 Modules of EasyDD

EasyDD contains four main modules; these are

### 7.1 Curve-Fitting

One of the main modules of EasyDD is curve-fitting by least squares minimisation using the Levenberg-Marquardt algorithm. Curve-fitting can be performed on a single pattern, or as a single batch process over multiple patterns, or as a multi-batch process over multiple forms and data sets. Curve-fitting can be done on single or multiple peaks using a number of profile basis functions with and without polynomial background modelling. The number of least squares cycles can be fixed or vary according to the convergence criteria. The parameters to be refined can also be selected with possible application of restrictions. The range of data to fit can be selected graphically or by using a prepared form. Some relevant statistical indicators for the fitting process, as outlined in Table 3, are computed in the curve-fitting routine.

Single curve-fitting is performed from the plotter on the pattern of the cell selected by a double-click. On running this mode of fitting a file called 'aVector' is created. This file contains a summary of the initial and final values for the

Table 3: Statistical indicators for pattern fitting and refinement.  $y^o$  and  $y^c$  are the observed and calculated count rate respectively,  $w$  is the statistical weight,  $O$ ,  $P$  and  $C$  are the number of observations, adjusted parameters in the calculated model, and applied constraints respectively,  $I^o$  and  $I^c$  are the observed and calculated integrated intensity respectively, and  $i$  and  $k$  are counting indices.

Statistical indicator	Definition
Profile residual	$R_p = \frac{\sum_i  y_i^o - y_i^c }{\sum_i y_i^o}$
Weighted profile residual	$R_{wp} = \left[ \frac{\sum_i w_i (y_i^o - y_i^c)^2}{\sum_i w_i y_i^{o2}} \right]^{1/2}$
Expected residual	$R_{exp} = \left[ \frac{O - P + C}{\sum_i w_i y_i^{o2}} \right]^{1/2}$
Bragg residual	$R_B = \frac{\sum_k  I_k^o - I_k^c }{\sum_k  I_k^o }$
Structure factor residual	$R_S = \frac{\sum_k  \sqrt{I_k^o} - \sqrt{I_k^c} }{\sum_k  \sqrt{I_k^o} }$
Goodness-of-fit index	$\chi^2 = \left[ \frac{R_{wp}}{R_{exp}} \right]^2 = \frac{\sum_i w_i (y_i^o - y_i^c)^2}{(O - P + C)}$

fitting parameters and a number of statistical indicators. As indicated earlier, single curve-fitting applies to the  $x$  range in the current view of the plotter.

Batch curve-fitting is carried out from the main window. Spectral data should be mapped on tab(s) and a form should be loaded through the main window beforehand. Batch fitting runs only on the parts of the patterns within the  $x$  range of the loaded form. For successful operation, the form  $x$  range should be within the  $x$  range of the spectral data. Curve-fitting in this mode can be performed on a single pattern, a number of randomly selected patterns from one tab or a number of tabs. It can also be performed on all patterns in a tab or a number of tabs with no need for cell selection. Two data files are created to summarise the outcome of batch curve-fitting. The first is called ‘batch’ and contains a summary of the input and output data in general as well as data for each fitted cell. These data are labelled with ‘ $t$ - $r$ - $c$ ’ where  $t$ ,  $r$  and  $c$  are indices for the tab, row and column of the cell. The second file, called ‘OlivierMatlab’, is a summary intended for use in post

processing. This file contains the refined parameters of the refined cells identified by 't r c' labels.

Multi-batch curve-fitting is carried out from the main window. It is an automated, large-scale fitting operation on multiple data sets. The use of this mode of operation, to replace repeated manual application of batch fitting, produces more reliable and organised results and saves a huge amount of time and effort.

## 7.2 Sinogram Treatment

The main item in this module is back projection which is a tomographic technique for reconstructing a 2D image in real space from a number of 1D projections and their angles. It relies on the application of the inverse Radon transform and the Fourier Slice Theorem. The raw data for back projection are sinograms consisting of a set of rotation versus translation measurements. Back projection can run in single and multi-batch modes, and can be applied on total or partial intensity as well as individual channels, with possible application of Fourier transform and filtering.

## 7.3 EDF Processing

The purpose of this module is to extract the information from EDF binary files obtained from CCD detectors. Two forms of extraction are available: conversion to normal 2D images in one of three formats (png, jpg and bmp), and squeezing to 1D patterns in *xy* text format.

## 7.4 Graphic Presentation

Examples of this module are mapping, visualisation and imaging to produce graphs in 1D, 2D and 3D spaces. These include creating tomographic and surface images in single, batch and multi-batch modes, as well as *xy* plots for spectral patterns.

Some of these graphic techniques use a direct display on the computer monitor, while others save the results to the computer storage in the form of image files.

## 8 Context Menus

EasyDD contains several context menus which are accessed via right-click:

- Main window context menu: contains selected items from the main window menus.
- Main window toolbars context menu: used to turn the main window toolbars on and off.
- Tab context menu: contains 12 items for manipulating cells, rows and columns of the active tab.
- Form dialog context menu: contains selected items from the form menus.
- Form dialog toolbars context menu: used to turn the form toolbars on and off.

## 9 Dialogs

EasyDD have some dialogs of special importance and hence require particular attention:

### 9.1 Settings Dialog

The purpose of this dialog (Figure 40) is to control the tooltips and grid size of the tabs. The dialog has two main sections:

1. The first is for setting the tab tooltips. This section contains two radio buttons:

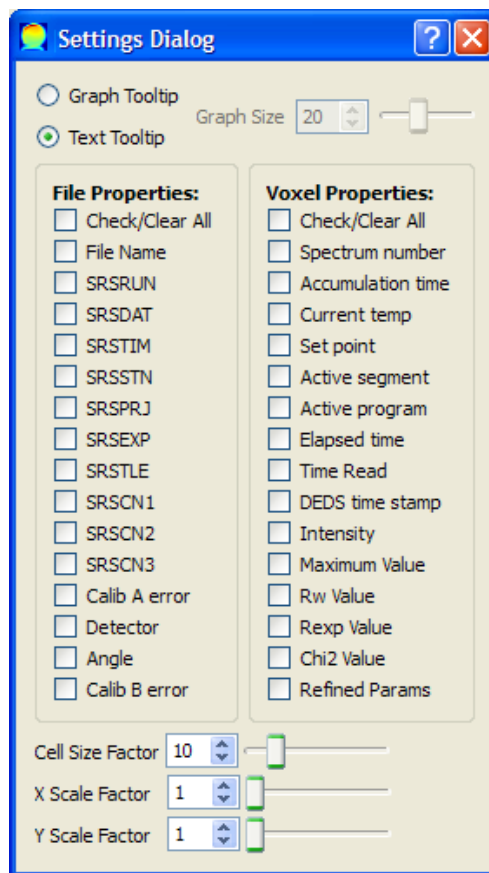


Figure 40: Settings dialog.

- (a) Graph Tooltip: to display a graphic tooltip, that is the graph of the spectrum of the cell to which the mouse is pointing. If this option is selected, the size of the graph can be adjusted from the ‘Graph Size’ integer spin box and the associated slider.
- (b) Text Tooltip: to display a text tooltip. If this option is chosen, two groups of tooltips are accessed; one for the file properties of the cell, and the other for its voxel properties. The latter include the initial properties like intensity, and the acquired properties such as statistical indicators and fitting parameters. The properties in both groups can be selected/deselected individually or as a group. Most of these properties belong to SRS 16.4 data files.

2. The second section is for adjusting the size and aspect ratio of the tab cells. The size is modified from the ‘Cell Size Factor’ spin box and slider, while the aspect ratio is modified from the ‘X Scale Factor’ and ‘Y Scale Factor’ spin boxes and sliders. When the width of the current tab is too narrow, the tab widget handler at the top-left corner can become too small and hence the other tabs cannot be accessed. In this case, the tab width should be increased by increasing the cell size factor or the x-scale factor to gain access.

## 9.2 FTP Dialog

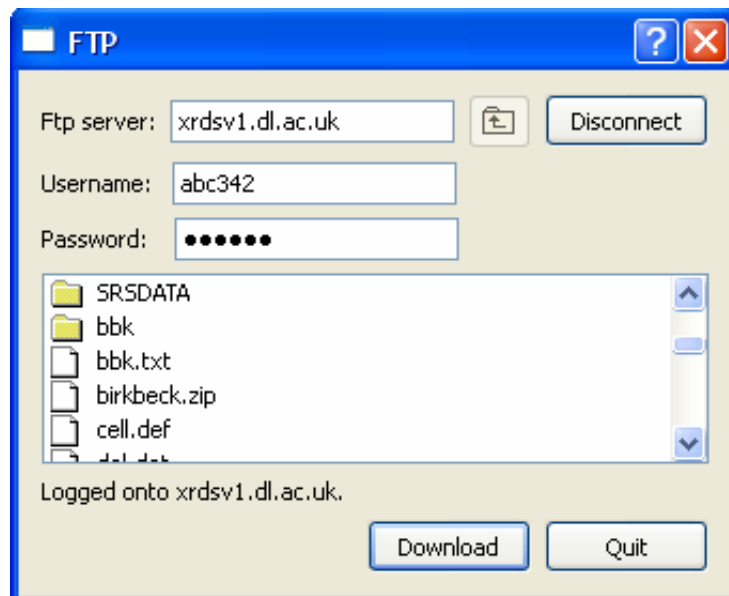



Figure 41: FTP dialog.

This dialog (Figure 41) is for importing files by File Transport Protocol (FTP). The dialog has three line editors: ‘Ftp server’ to input the address of the remote server, ‘Username’ and ‘Password’ to input these attributes. The ‘Connect/Disconnect’ push button is for connecting/disconnecting to the server after entering valid data. On connecting to the server, the folders and files on the server root directory appear on the list widget beneath the line editors. A folder can be opened and its contents inspected by double-clicking on its icon in the list widget.

On pressing the button () next to the first line editor, the list widget view returns to the parent directory. A file can be downloaded from the remote server by selecting it first followed by pressing the 'Download' button. The 'Quit' push button is used to close the FTP dialog. The current functionality of the FTP utility is primitive and hence requires major development.

## 10 Appendix 1: Tilt Correction Derivation

In this derivation we use a standard Cartesian coordinate system in a plane perpendicular to the original beam direction which passes through the centre of the image and hence the origin of the coordinate system. For simplicity we use a unit circle centred at the origin and lying in this plane. In the following derivation we have:

$\phi$  is the angle of an arbitrary point on the unit circle ( $0 \leq \phi < 2\pi$ )

$\theta_x$  is the tilt in the  $x$  direction

$\theta_y$  is the tilt in the  $y$  direction

$\theta$  is the actual tilt at an arbitrary point on the tilted CCD plane. It is obvious that  $\theta$  is a function of  $\phi$  and can have negative as well as positive values.

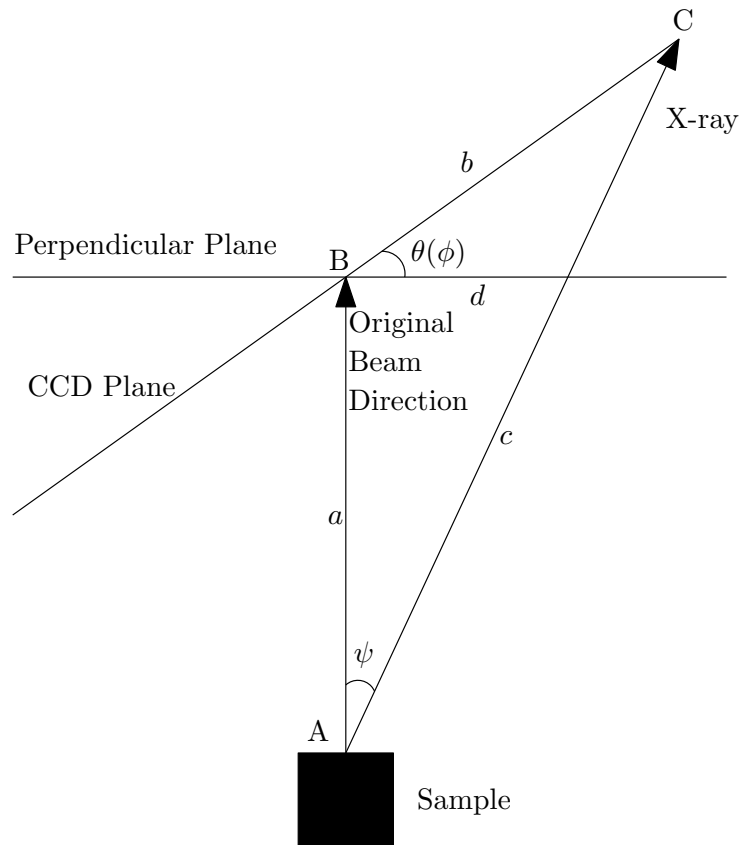


Figure 42: A schematic diagram to demonstrate the derivation of tilt correction.

From a simple geometric argument, the actual tilt as a function of  $\phi$  is given by

$$\theta(\phi) = \arctan[\cos \phi \times \tan \theta_x + \sin \phi \times \tan \theta_y] \quad (2)$$

As can be seen, this formula retrieves the correct (and obvious) tilt in the  $x$ ,  $y$ ,  $-x$  and  $-y$  directions.

Now referring to Figure 42, which is in the plane identified by the original beam and the ray directions, and on applying the cosine rule on the triangle ABC we have

$$c = \sqrt{a^2 + b^2 - 2ab \cos \left( \theta + \frac{\pi}{2} \right)} \quad (3)$$

Using the sine rule we obtain the scattering angle

$$\psi = \arcsin \left[ b \times \frac{\sin(\theta + \pi/2)}{c} \right] \quad (4)$$

And finally the required quantity,  $d$ , can be obtained from

$$d = a \times \tan \psi \quad (5)$$

It is obvious that there are other methods for finding  $d$ . Although this derivation is demonstrated for  $\theta > 0$ , as seen in Figure 42, it is general and valid for  $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$ .

## 11 Appendix 2: Missing-Rings Correction

Since the charge-coupled device detector has a rectangular shape, the rings whose radius exceeds a certain limit, by having a radius that exceeds the distance between the ring centre and one or more of the rectangle sides, will not be complete and therefore the pattern will have a reduced integrated intensity at high scattering angles. To compensate for the loss of intensity from these incomplete diffraction rings, a simple and time saving technique is used. This technique scales the incomplete rings to the continuum value at that radius by multiplying the total intensity of the incomplete rings by the ratio of the continuum circumference at that radius to the actual number of pixels in that ring. In multi-batch operations, this number is computed only once while serially reading the values of the pixels of the representative EDF file. These scale ratios are stored in a 1D vector and used during the application of the numeric conversion routine.

Figure 43 displays an example of the actual (discrete) number of pixels of the image rings as a function of radius in pixels alongside the continuum value of  $2\pi r$  represented by the straight line. As seen, the two curves match very well for the complete rings. This indicates that the continuum is a very good approximation, except for the rings which are too close to the image corners. The meaning of the two cusps is obvious as the radius of the rings increases and exceeds the rectangle sides in one direction (i.e. vertical or horizontal) and then the other. Figure 44 presents a sample diffraction pattern obtained from an EDF image with and without the application of the missing-rings correction.

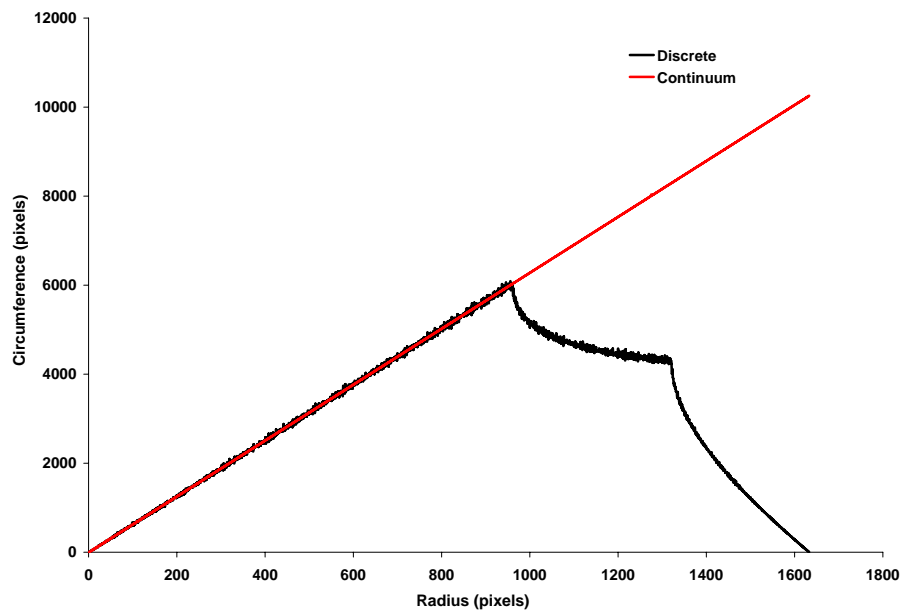


Figure 43: The continuum approximation alongside the discrete value for the number of pixels in a ring.

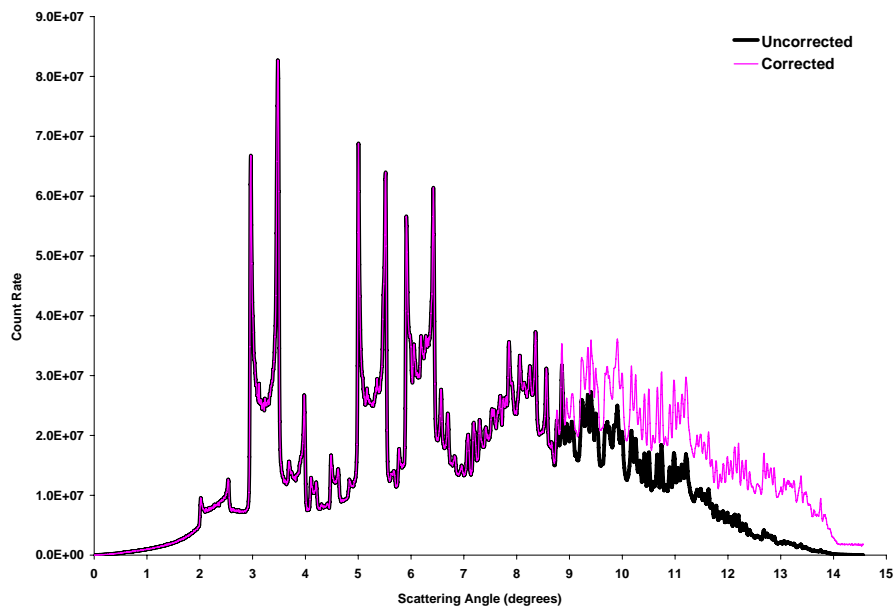


Figure 44: A sample of 1D diffraction pattern extracted from a 2D EDF pattern. The pink curve is obtained with the missing-rings correction while the black is obtained without this correction.

## 12 References

- [1] Diamond Light Source website: [www.diamond.ac.uk](http://www.diamond.ac.uk). 11
- [2] European Synchrotron Radiation Facility (ESRF) website: [www.esrf.eu](http://www.esrf.eu). 11
- [3] High Energy X-Ray Imaging Technology website:  
[www.hexitec.co.uk/index.htm](http://www.hexitec.co.uk/index.htm). 11
- [4] LUCIA website:  
<http://www.synchrotron-soleil.fr/Recherche/LignesLumiere/LUCIA>.  
11, 20
- [5] Qt website: <http://qt.nokia.com>. 10
- [6] QwtPlot3D website: <http://qwtplot3d.sourceforge.net>. 10
- [7] Synchrotron Radiation Source (SRS) website: [www.srs.ac.uk/srs](http://www.srs.ac.uk/srs). 11