APPENDIX C

Multi-File Fitter

C.1 Introduction

comparatively large data sets automate analysis process

The first version of this fitting package was created with the help of A. Leishman during his 2014 summer REU.

C.2 $GRAS_{ANS}P$

The Graphical Reduction and Analysis SANS Program [ILL website/GRASP manual], or GRASP, is a data analysis package developed by Charles Dewhurst of the Institute Laue Langevin for use with anisotropic or diffraction-based SANS measurements. "GRASP presents a modern package and general purpose white of tools to deal with the diverse requirements for reduction and analysis of two-dimensional small angle scattering and diffraction data." MFF was built within the GRASP m-code software, and thus also requires the commercial Matlab package in order to run. However, there is also a freely available "runtime" version of the application that can stand alone from Matlab.

C.3 Principles of Fitting Algorithm

(a) Reference Files: fits a 1-peak and a 2-peak file, used to get initial guess values (GRASP)

(b) Data Smoothing & Binning

(c) First Cycle: Fix intensities & fwhm, determine centers (MATLAB)

(d) Second Cycle: Fix centers & fwhm, determine intensities (MATLAB)

(e) Third Cycle: Fix centers, intensities free, determine fwhm from files with dxc ¿ 2*fwhm (MATLAB)

(f) Fourth Cycle: Fix fwhm, centers & intensities free (MATLAB), run a total of 3 times

C.3.1 Fitting

In all cases, data is fit to a multi-peak Guassian and providing good initial guess values is imperative. Matlab uses a Non-linear Least Squares method,

In it's most general form, all fitting cycles/algorithms use one equation:

$$y = y_0 + \sum_{i=1}^{3} \frac{i_{0i}\sqrt{\log 4}}{w * \sqrt{\pi/2}} \exp\left(\frac{-2(x - x_{ci})^2}{w^2/\log 4}\right) \tag{C.1}$$

where y_0 is the background, i_{0i} the integrated intensity of the ith peak, w is the FWHM, and x_{ci} is the center of the ith peak. The FWHM of the Bragg peaks is a convolution of the instrumental resolution and the intrinsic width of the scattering lattice. In the case of the azimuthal width, the instrumental resolution is not as good. For this reason, the peaks were all constrained to have the same width, and for later fitting cycles this value was fixed. For more details on the resolution and SANS in general, please see chapter **??**. For the metastable L phase, all fitting cycles fixed $i_{02} = 0$ as this MS-to-GS transition never displayed the coexistence of three domain orientations.

C.3.2 Binning and Smoothing

C.3.3 Program Structure

MFF is a complex ... using several functions, scripts, and data structures

Data structures:

1. Fit Data
2. Figure and table handles
3. Fitting options

C.4 Installation and Use

To install MFF,

C.4.1 Main GUI

1. Sector Window
2. Smoothing Options
3. Fit Algorithm Selector
4. Optional Plots
5. Reference Files and modifications
6. fit options
7. View Params

   displays table of current fit data; offers a "refit" option and displays error messages

   runs function mf_fitter_table()
8. Save Options

   experiment extension (i.e. where data should be stored)

   example:

```
/Users/edewaard/Dropbox/Eskildsen Research/2016_09 MgB2 ILL D33/Analysis/
```

   particular data run folder

   preferably specifies important parameters of run, such as AC amplitude and Temperature

   example

```
9_3G_14K
```

box to enter data that varies with Numor, here it is "Applied AC Cycles" but this could be modified to ask for any experimental parameter.

check boxes for how plots should be saved

i.e. .jpg, .eps, .pdf, .fig

9. Go

Runs the selected fitting algorithm with the various options/inputs

C.4.2

C.5 Modifying MFF

The underlying structure of the MFF can be applied to any series of SANS data you want to fit. The things that will change will be the fitting algorithm and the types of plots you would like to make/store.

Steps:

1. Determine a fitting alogrithm

   you can still call M's data smoothing code and matlabFit.m

   matlabFit.m can be modified for any equation, simply write it as a string in "fit type"

   Helpful Hint: debug the code line-by-line once to make sure you know what order the confidence intervals of the variables are stored.

2. Write a script, and insert it into the drop down menu
3. and plotting code to "plot options"
4. add relevant data (i.e. for my system, the number of AC cycles, but in principle this could be magnetic field, temperature, etc.) to the mf_fitter data structure.

C.6 Directory of Matlab Files within Package

files in this section currently:

mf_fitter_mpf3, matlabFit, CM (maybe belongs in another script) mf_fitter_table (need to fill in more details here) mf_fitter_plotMaker getData,

C.6.1 getData

```
function [phi, Int, Int_err] = getData(img_num)
```

This function exports the "raw" neutron count data. It requires a depth number

```
img_num
```

as an input, which selects which data to export. It then uses the given sector box to export the data

```
phi, Int, Int_err
```

using 0.1 for the angle binning. Data will then be binned and potentially smoothed.

C.6.2 mf_fitter_plotMaker.m

```
function [cm_han, ps_han, int_han, int_tot_han, ad_han] = mf_fitter_plotMaker()
```

This function requires no inputs, and outputs handles to the various plots it generates. The handles are output so that they can be closed (or saved) in the main script if desired. Plots made:

1. Colormap
2. $\Delta\varphi$
3. Bragg Peak Intensities
4. Sum of Intensities
5. "Peak Decay" - plots the fits/data offset by numor, convenient for quickly checking fits

insert examples of each plot type?

C.6.3 mf_fitter_mpf3.m

```
function [ ] = mf_fitter_mpf3()
```

This function takes no inputs and gives no outputs. Rather, it executes the main fitting algorithm as outlined above, tailored to the superheated VL where two peaks continuously rotate.

C.6.4 matlabFit.m

```
function [fitobject, ci, fig_h] = matlabFit(opt, phi, int, interr,  ...
problemVarNames, problemVarValues, start, lower, upper)
```

This function calls the matlab fitting tool. For inputs, it requires the data to be fit $(phi, int, interr)$, the variables in the fit equation that will have fixed values $(problemVarNames, problemVarValues)$, and bounds/initial values for the variables to be fit $(start, lower, upper)$. It then performs the Non-linear Least Squares fitting based on the general fit equation, Eq. C.1 and outputs the resulting fit values $(fitobject)$, confidence intervals (ci), and a plot of the data with fit (figure handle stored in *fig_h*).

There is also an option input ($opt = 1$) that gives an additional constraint on the integrated intensities,

$$i_{01}; \quad i_{02} = 0; \quad i_{03} = 1 - i_{01} \tag{C.2}$$

and multiplies the entire sum of exponentials by a "total intensity" pre-factor i_{TOT}. This is used for the superheated VL where the two metastable domain orientations rotate continuously towards the GS. More specifically, when the peaks are very close together, they appear to be one single broad peak. However, fitting with a single Gaussian yields a FWHM that is much larger than a true GS. A two-peak Gaussian can be used, but there is large amount of uncertainty in how much intensity is in each peak. Calculating the total intensity based on these fitted values and propagating the

error in the standard way yields a value with large error bars on these files. The total intensity is actually well known, and the large error bars overestimate the true error. Using the fit constraints above, Eq. C.2, enables a more accurate determination of the error on the total intensity (well known) versus the error on the intensity ratio between the two peaks (not well known). Setting $opt = 0$ is the default fit type as described above.

C.6.5 CM.m

makes the colormap

trick with matlab pcolor command

C.6.6 mf_fitter_table.m

```
function[ ] = mf_fitter_table()
```

formats the fit data into a table for the user to view. in principle, it offers the user the option to pull up a fit and try to make it better/examine it more closely