APPENDIX C

Multi-File Fitter

Multi-file Fitter (MFF) was designed to automate the data analysis process for the typical "Buzz" experiment on the vortex lattice (VL) MgB_2. This software suite interfaces with GRASP to fit entire data sequences with multiple files as a function of some control parameter. For the data presented in this thesis, that control parameter was always the number of applied AC cycles. In principle, MFF could easily be adapted for a broad range of control parameters, such as magnetic field or temperature. The first version of this fitting package was created with the help of A. Leishman during his 2014 summer REU.

C.1 $GRAS_{ANS}P$

The Graphical Reduction and Analysis SANS Program, 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 is freely available from the Institute Laue-Langevin [1] and has been adopted by many scattering facilities. "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. As the runtime version is already compiled, it is not possible to make modifications to this version of the software.

C.2 Installation

Instructions on how to install GRASP can be found online [1]. To add Multi-File Fitter to GRASP, start by adding the MFF folder with the "User Modules" folder of GRASP. Make sure all of the files are in your Matlab path; this can be done by right clicking on the folder in the "Current Folder" window and selecting "Add to Path" → "Selected Folders and Subfolders." Next, we need to a way for GRASP to launch MFF. One simple way of doing this is to edit the modify_main_menu_items.m file, which can be found in the main_interface folder of GRASP. After opening this file, scroll down to the "User Module" section (approximately line 512) and add the following line of code:

```
uimenu(grasp_handles.menu.user_modules.root,'separator','on',....
'label','Multi-File Fitter','callback','mf_GUI_window','enable','on');
```

Save modify_main_menu_items.m after adding this line of code. This should be all the MATLAB coding required to use MFF. Note: If you cannot/ do not want to modify GRASP code, MFF can be accessed and ran by typing the command mf_GUI_window in the Command Window of Matlab while running GRASP.

C.3 Use

Multi-file Fitter can now be launched from the GRASP window. Simply select "User Modules" → "MF Fitter". This will launch the main MFF graphical user-interface (GUI), which is shown in Fig. C.1

outline of various buttons on the main screen.

Figure C.1. Main GUI for Multi-file Fitter.

C.3.1 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 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.4.1 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.4.2 Binning and Smoothing

C.5 Program Structure

MFF is a complex ... using several functions, scripts, and data structures. These underlying features will be discussed here.

C.5.1 Data Structure

Everything from the resulting fit data to the user input options is stored within the main data structure, mf_fitter. The main fields are:

1. algorithm_options:

   This field stores any specifics related to the selected fitting algorithm. For example, the user has the option to turn on / off viewing all of the fits from a particular cycle of the algorithm.

2. fit_data

   This field contains all of the parameters from the fit type, such as Eq. **??** described above. They are stored as a function of the file number or "numor," which in principle corresponds directly with the control parameter.

3. handles

   This field stores the handles (which act similar to pointers) for all windows and input fields, such as the GUI window or the control parameter text entry box. This enables, for example, the software to use the 'get' command to check what values have been input by the user on the GUI.

4. save_options

   This field stores the folder name and full path for saving all desired outputs from the software suite are . It also stores figure extension selections (i.e. should figures be stored as .eps, .fig, .jpg, and/or .pdf.

5. user_inputs

   This field contains information input through the GUI by the user. This includes the subfields:

   (a) control_parameter

   The control parameter is what is changing in each numor and will be the x-axis for final plotting, this could be the number of applied AC cycles or temperature for example.

(b) int_cutoff

This subfield specifies what minimum peak intensity is necessary for being included in code to determine guess values for future cycles.

(c) dphi_cutoff

This subfield specifies the minimum peak separation iss necessary for being included in code to determine guess values for future cycles.

(d) reference_files

Reference_files are peaks that are clearly resolvable, in terms of e.g. the intensity and separation; fit parameters from these are used as initial guess values.

(e) smoothing

Smoothing includes options for the fwhm and step size for binning / gaussian smoothing.

C.6 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.7 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.7.1 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.7.2 grasp$_{p}lot_{f}it_{c}allbacks_{2}.m$

```
function output = grasp_plot_fit_callbacks_2(to_do,option)
```

This function was copied from the original GRASP code. It was modified to store the chi^2 from a fit. line 939 - mf$_{f}itter.temp.chi2; ButIhaveacommentthatitmightnotbenecessary$

DONE

C.7.3 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.7.4 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.7.5 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.

option 3 and 4 were introduced to generalize opt 1 and 2 to not restrain the fits. Other fit types could be included in this function using different option names for generalizability

C.7.6 CM.m

makes the colormap

trick with matlab pcolor command

C.7.7 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

BIBLIOGRAPHY

1. GRASP. URL `https://www.ill.eu/instruments-support/instruments-groups/groups/lss/grasp/download/`.

2. See Supplemental Material.

3. J. F. Annett. *Superconductivity, Superfluids and Condensates*. Oxford University Press, 2004.

4. N. Avraham, B. Khaykovich, Y. Myasoedov, M. Rappaport, H. Shtrikman, D. E. Feldman, T. Tamegal, P. H. Kes, M. Li, M. Konczykowski, K. van der Beek, and E. Zeldov. 'Inverse' melting of a vortex lattice. *Nature*, pages 1–4, May 2001.

5. G. E. Bacon. *Neutron Diffraction*. Oxford University Press, Glasgow, second edition, 1962.

6. L. J. Bannenberg, F. Qian, R. M. Dalgliesh, N. Martin, G. Chaboussant, M. Schmidt, D. L. Schlagel, T. A. Lograsso, H. Wilhelm, and C. Pappas. Reorientations, relaxations, metastabilities and domains of skyrmion lattices. *Phys. Rev. B*, 96:184416, 2017.

7. A. Bauer, A. Chacon, M. Wagner, M. H. R. Georgii, A. Rosch, C. Pfleiderer, and M. Garst. Symmetry breaking, slow relaxation dynamics, and topological defects at the field-induced helix reorientation in MnSi. *Phys. Rev. B*, 95: 024429, Jan. 2017.

8. K. Binder and E. Stoll. Scaling Theory for Metastable States and Their Lifetimes. *Phys. Rev. Lett.*, 31:47–51, 1973.

9. G. Blatter, M. V. Feigel'man, V. B. Geshkenbein, A. I. Larkin, and V. M. Vinokur. Vortices in high-temperature superconductors. *Rev. Mod. Phys.*, 66 (4):1125–1388, Oct. 1994.

10. R. Cubitt, M. R. Eskildsen, C. D. Dewhurst, J. Jun, S. M. Kazakov, and J. Karpinski. Effects of two-band superconductivity on the flux-line lattice in magnesium diboride. *Phys. Rev. Lett.*, 91:047002, 2003.