Appendix B: SAMXAS MANUAL

Section B.1 INTRODUCTION

This is a brief introduction to SAMXAS, an application to help analyze Quick-EXAFS data from DND-CAT. As a word of caution, there are several requirements to meet before you can use the program. Some of these requirements have been error trapped, many have not. If you run into problems, please inform me (s-webb@nwu.edu) and I will do what I can to solve them in a timely manner. As a note: if you find that you get an error, the most probable cause 90% of the time is that a path is misdirected (you misspelled a name) or simply that the file that you think is there doesn't exist. Please check these options before giving up and complaining to me.

Section B.2 REQUIREMENTS

Section B.2.1 Software

REQUIRED SOFTWARE: (should not be a problem if run on bauges)

(1) Scilab, v. 2.4.1 There should now be a symbolic link to the program file in `/usr/local/bin`.

(2) Autobk. Also linked to.

(3) Tcl/Tk. The GUI interface for Scilab. Again, everything should be linked properly.

If you are NOT running this on bauges, you need to change the first line of EACH of

the *.tcl files in the SAMXAS directory to properly point to where your copy of Tcl and Wish are located.

(4) SAMXAS (of course). The main program files are located in `/usr/users/samxas`. Everyone should have read / execute privileges on the appropriate files. If not let me know and I will try to change it for you. You may want to do as I have done and create an executable in your local bin folder: *i.e.*, I have the following file in `~swebb/bin`:

```
cd /usr/users/samxas/samxas/
./samxas
```

Section B.2.2 Directory Structure

This is a MUST. SAMXAS will not like you if you don't follow this strictly. If you can't handle this imposed structure, tough. The details of the directories are explained below.

```
~your user name
      |
      |- datfiles
      |     |
      |     |- group
      |     |    |
      |     |    |- ave3e
      |     |    |- copymyasse.awk
      |     |    |- dofine
      |     |    |- process3e
      |     |    |
      |     |    |- data directory
      |     |    |      |
      |     |    |     |- fn_00001.dat
      |     |    |     |- fn_00001.dat
      |     |    |     |-    ...
      |     |    |
      |     |    |- data dir 2
      |     |    |
      |     |    |-     ...
      |     |
      |     |- group 2
      |     |
      |     |-  ...
      |
      |- chis
      |    |
      |    |- group
      |    |
```

```
|    |- group 2
|    |
|    |- ...
|
|- mus
|   |
|   |- group
|   |
|   |- group 2
|   |
|   |- ...
|
|- scratch
      |
      |- mu.awk
      |- scitok
```

So that's the basic idea. Here's an explanation if you're not outright familiar with the SAMXAS program.

DATFILES:

This directory contains the raw data from the beamline. I prefer to break the files down into different groups depending on which date the data was collected. This is obviously left to your own preferences on how to group / divide / separate your data. Within each group, the process files are required. These files do not change, but must be copied each time into each group directory. The SAMXAS program will create the script to run and use these programs, so no command line operations are required to process the data. The files can found in the `~/samxas/processfiles` directory.

Within each group directory, each XAS sample has its own data directory. This keeps the data grouped conveniently. You should have the raw `fn_000#.dat` files in this directory. SAMXAS will process the files, writing each `fn_0000#.out` and

`fn_0000*.mu` file as well as doing the averaging to acquire a composite `fn.mu` file. For a complete description of the processing and the outfiles, see the section below.

CHIS AND MUS:

These directories store the SAMXAS normalized data for the EXAFS (chi vs k) and XANES data (mu vs eV). Again, the data is organized within group directories, with the same names as in the datfiles directory. These group directories must exist before normalizing the data with SAMXAS!

SCRATCH:

This directory is a temporary directory for SAMXAS to store files for the chi normalization process. Autobk gets its input from files, so this directory acts as a storage location for those files. The programs scitok and mu.awk are required in this directory for proper execution.

Section B.3 SAMXAS MODULES

The SAMXAS program is designed to be a modular GUI interface to help with the analysis of XAS data. The program is called from within Scilab and executes the various commands between the GUI and Scilab. When you start, you should be presented with three main windows.

Figure B.3.1: SAMXAS Console window.

Figure B.3.2: SAMXAS main control window.

(1) The Scilab window. I prefer to either minimize this window in the background or keep it small and off to the side to track the progress of the commands and errors.

(2) SAMXAS Console. This is a simple communication textbox that reports on some of the progress of the analysis and displays the fitting results. Buttons at the bottom allow the user to save the contents of the window to a file for later use. See Figure B.3.1 for an example.

(3) SAMXAS Main Menu. This is the heart and soul of SAMXAS. In addition to displaying the nifty logo, from this menu bar, you may select the various analysis modules (See Figure B.3.2).

Section B.3.1 Process Data

This module is the part of the analysis routine that converts the raw binary data collected at the beamline to readable and manageable ASCII files. In addition, the true experimental error from the EXAFS measurement is calculated. This is done through a boxcar binning procedure of the data. As the monochromator is slewed, the computer oversamples the detectors at a rate of 12.5 kHz. This creates huge data files, which can easily add up to 5 or 10 MB per scan. To reduce the volume of data, the data is divided in to bins that span a specified eV or k-space interval. The data reported as the value in that bin is the average of all of the measurements taken in the bin. In the XANES pre-edge region, this amounts to approximately 740 samples, whereas in the far-EXAFS region, there are about 6800 samples per bin. For each bin, statistically, we want to calculate the average and the sigma error for the bin.

Each bin contains *N* oversamples of the same measurement, given that a bin covers a relatively small interval. The average value for the measurement is simply the sum of all the measurements, divided by the number of measurements, *N*. Each of these individual samples has an uncertainty associated with it. This error in each of these measurements can be expressed as the standard deviation of the samples in the entire bin. If we assume that the samples in the bin all have the same uncertainty, using the standard propagation of errors theorem, the error of the average value of the bin is

$$\sigma^2 = \left(\frac{1}{N}\right)^2 N\sigma_i^2$$
$$\sigma = \sigma_i / \sqrt{N}$$

where σ_i is the error of the individual measurements. For most of the spectrum, the measurements are normally distributed within a bin. This allows a standard normal distribution when applying the Monte-Carlo method to propagate the errors through the rest of the calculations. The only region where noticeable problems with this assumption occur is at the absorption edge, where the signal is increasing rapidly.

The module performs most of its analysis by sending scripts through UNIX. The module is not often used, as the data is reduced most often at the beamline, and the raw datfiles are too large to store on the computer for very long. Figure B.3.3 shows an example of the Process Data Module control screen. The first textbox requires the path to the data processing scripts. Since the scripts also alter and write some of these files, a

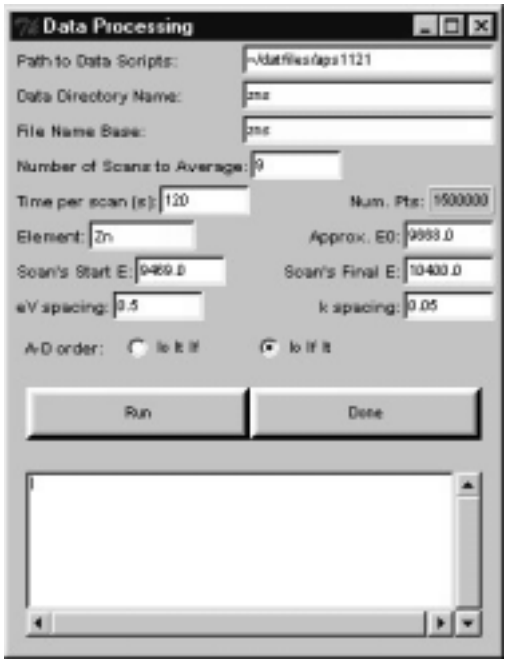

Figure B.3.3: Data processing control window of SAMXAS.

new set of files should always be placed in the group directory. The next textboxes prompt the user for the data directory, the base file name of the data in the directory, and the number of scan repetitions. For the example in Figure B.3.3, the path to the data files is `~/datfiles/aps1121/zns` and which contains datfiles `zns_00001.dat` through `zns_00009.dat`. Next, the time of each scan is entered, and the number of points sampled at 12.5 kHz is displayed in the adjacent box. The element being studied does not need to be entered, but standard operating parameters for selected elements will automatically be entered in the appropriate places if the element is entered. These parameters include the edge energy as well as the starting and ending energies of the scan. The next parameters are the eV and k-space intervals to average. Usually, 0.5 or 1 eV is appropriate, and data is averaged at 0.05 in k-space. The last parameter is the

order of the data channels on the A-D converter on fava. In the early days, the order wasn't standardized, so the process files needed flexibility. This problem should be solved at this point in the evolution of the Q-EXAFS data collection. The standard order is I_0, I_F, and I_T. When the script is run by pressing the "Run" button, the standard output of the UNIX script should be displayed in the scrollbox.

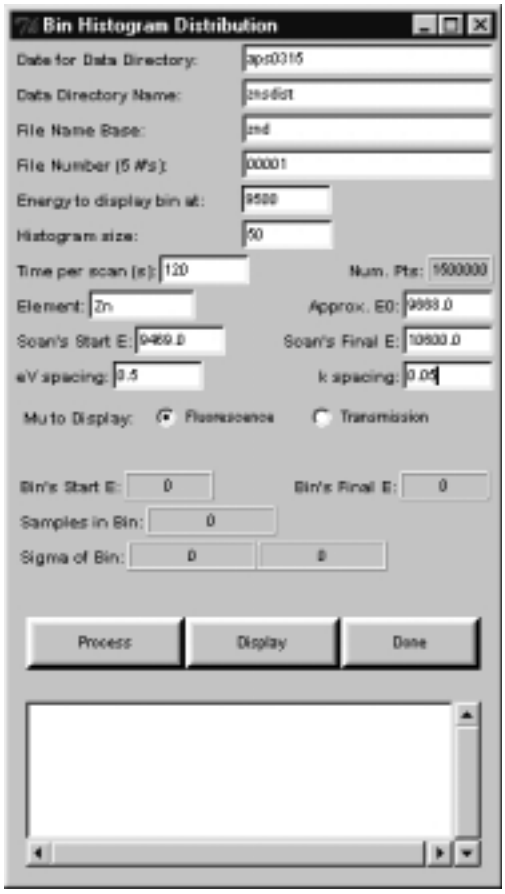

Figure B.3.4: The Bin Histogram Display interface.

Section B.3.2 Data Histogram

The Data Histogram module of SAMXAS allows the user to examine the contents of individual bins of the raw data files. Note that the raw data files must be present in order to run this module. It is not a good idea to do long term storage of the raw data, but rather ftp the files when necessary. This is useful to make sure that the distributions in the bin are close to gaussian, which is an assumption that is made in the propagation of error throughout the program. The program uses a modified version of the data processing scripts to decode the raw data file at a given energy. The resulting files are saved in the datfiles directories and can then be displayed in a histogram format. Figure B.3.4 shows the interface screen for this module.

The first data entry boxes on the interface tell the program where to find the data files. This includes the date (or group) path, the sample directory name, the file name, and the index of the data file. Indexes range from 00001 to 99999 depending on how many scans were taken. The index number must be in the five-character format to work properly. The next entry is the energy at which the binning will be examined. The program will search the bins of the data file until it can find a bin that brackets the desired energy. Next, the histogram size is entered. This variable determines how many bars the final histogram plot will produce. The default value is 50. The next boxes have the same purpose as in the Data Processing module (Section B.3.1). It provides the program with the specifics of the datfile. Finally, the last entry determines which mu will be plotted, the fluorescence data or the transmission data.

The program is started by pressing the “Process” button. The UNIX processing script will be run and the results stored in a file in the samples raw data directory. The scrollbox will print the standard output when the file is finished processing. Clicking the “Display” button will activate the histogram. In addition to plotting the histogram, a plot of each channel of data (I_0, I_T, and I_F) is created. I_0 is in black, I_T in green, and I_F in blue. Figure B.3.5 shows examples of the graphical output. Also, the energy values of

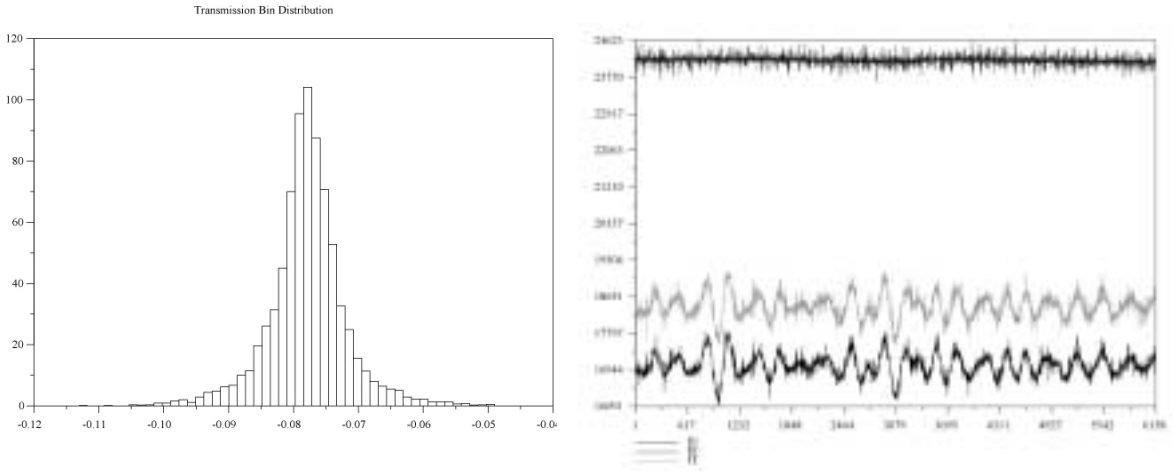

Figure B.3.5: Graphical output of the Bin Histogram module. Left is the histogram plot, right shows all the samples of the raw data in the bin.

the bin, the number of samples in the bin, and the standard deviation of the samples is fed back to the interface window. The percent standard deviation (standard deviation divided by the average value) is also given. Caution should be taken when using this number, as many of the values of mu, particularly for transmission experiments, will be close to zero, giving large percent standard deviations.

Section B.3.3 Normalize Mu

The Normalize Mu module uses the `<fn>.mu` produced from the data processing steps and normalizes the X-ray absorption spectrum to a unit edge step. In the process, it takes the experimental error and carries it through each step of the calculation. This is done through a pseudo-Monte Carlo method. The normalization step fits two curves to the data. The first is a linear fit to the pre-edge XANES portion that establishes the baseline. The second is a quadratic fit to the EXAFS region. When these lines are subtracted out, a normalized, unit edge step spectra results. The approach with the error propagation adds a normally distributed error to each point in the spectrum. The error added to each point is dependent on the experimental error as determined in the data processing step. This procedure of applying an error and normalizing the data is repeated 1000 times. This was determined to sample the errors most effectively a reasonable amount of computing time. At each step, the normalized data array and the coefficients of each of the regressions are stored. The average and standard deviations of the agglomeration of all 1000 iterations are taken and the final result stored in the `mus`

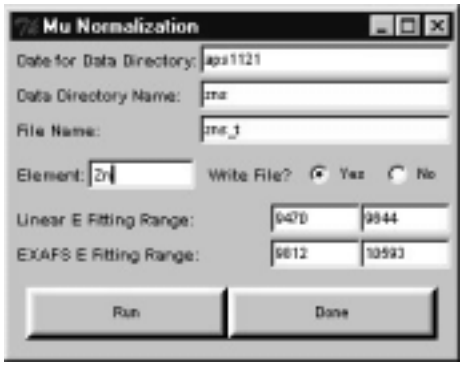

Figure B.3.6: The interface for the mu normalization module of SAMXAS.

directory under the filename `<fn>-n.mu`. The interface for this module is shown in Figure B.3.6.

As is standard for SAMXAS modules, the first few data entry boxes are used to specify which file is to be normalized. Next, a radiobutton prompts the user whether to save the resulting file. The energy regions for the fitting regimes are entered at the bottom of the window. Normally, the linear region consists of the beginning of the file to about 50 eV before the edge. The quadratic extends from about 150 eV after the edge to nearly the end of the file. It is usually wise to not extend the quadratic region to the very end of the file, as the last point does not always register. In the “element” entry, the user enters the element examined. If it is present in the SAMXAS library, the module will suggest typical energy ranges for the normalization step.

The module is started by clicking on the “Run” button. The Monte Carlo simulations take a few moments to complete. A message is displayed in the SAMXAS console window when the calculations are finished. Additionally, the height of the edge step (before normalization) is given. This value can be used to compare the relative amount of absorbing atoms present between samples. Thus, if all the samples have the same mass of sample and have the same thickness, the height of the edge will be proportional to the concentration of the element present. The normalized mu and the associated errors are displayed graphically (see Figure B.3.7).

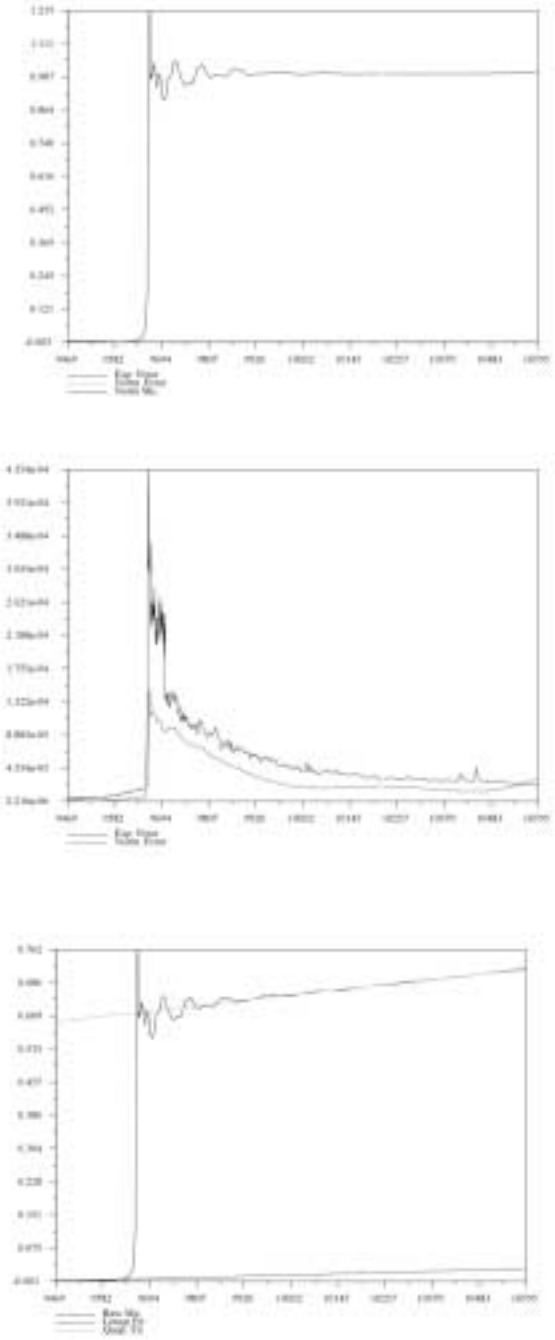

Figure B.3.7: Graphical windows produced by the Normalize Mu routine.

Section B.3.4 Cal Shift Mu

This module is a simple energy calibration process. Since the current setup at DND-CAT does not allow the collection of a simultaneous reference foil, a process is needed that will allow comparison of data from session to session. The monochromator is normally well calibrated by the staff, but offsets a few eV may occur when comparing data scans of the same compound from different collection dates. This is not too important for EXAFS analysis, but it is absolutely critical for XANES analysis. In order to make comparisons, the user must choose a known reference compound that will be run every time at the beamline. The user then defines the inflection point of the edge (maximum of the first derivative spectrum) as a fixed energy. The difference between the measured edge energy and the defined edge is then subtracted from the samples collected in the rest of the sample period. The module also does an interpolation to make sure that all XANES spectra have the same energy spacing over the same eV range. The file saved is given the name `<fn>-ns.mu`. When data fitting or reconstruction is performed in the XANES region, these files are used. It is important to

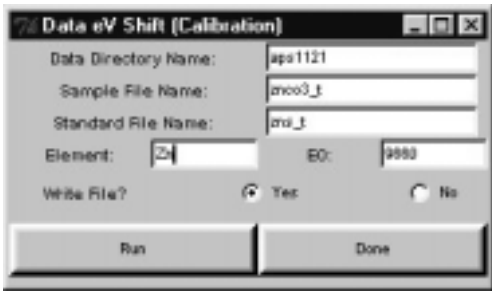

Figure B.3.8: Interface for the energy calibration module.

also realize that in order to make an accurate comparison of XANES data, the same slit conditions should be used. Wider openings of the slits on the beamline can lead to extensive broadening of features, and can radically effect the slope of the edge step.

Figure B.3.8 shows a screenshot of the module interface. The first entry box is the usual group name under which to find the desired files. Since the program uses the normalized data, this step must be performed after the "Normalize Mu" module (Section B.3.3). The second entry box is the name of the data file that will be shifted in energy. The third entry is the name of the calibration reference file. Note that one should also perform the shift on the calibration sample as well. The "element" entry box will provide advice for the "standard" edge values for several of the common elements that are used at Northwestern. The value is placed in the "E0" entry. Other choices can be selected if the element is not in the SAMXAS code or if another value of E_0 is desired. The module is started by clicking the "Run" button. When executed, the inflection point

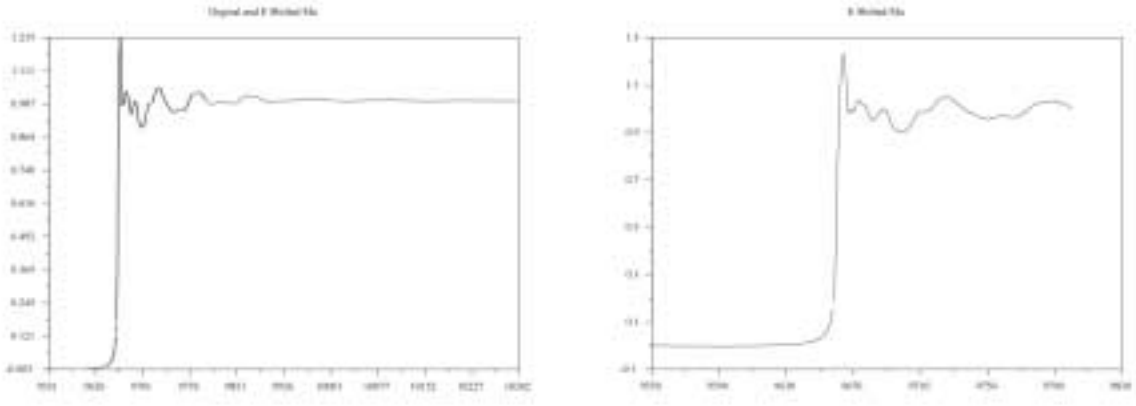

Figure B.3.9: Graphical output of the eV Shift module showing the XANES portion of the spectrum.

and the amount of shift are displayed in the console window. The module also properly takes into account the energy spacing of the raw data, which is also given in the console window. Graphical output consists of a plot of the reference and its shift to the new energy and a plot of the XANES region of the data file. An example is given in Figure B.3.9.

Section B.3.5 Normalize Chi

This module extracts the EXAFS region of the spectrum. This is done by removing the absorption coefficient of the isolated atom from the spectra, *i.e.* removing "the edge step". However, this is not as trivial as it sounds. The absorption of the lone atom is not know *a priori*, but it is usually assumed that a reasonable approximation to this is a smooth curve that follows the spectra without the wiggles and oscillations. Often a set of polynomial splines is used to fit the EXAFS portion of the spectrum. To avoid adverse human bias in this step, the normalization is performed in a manner that is as reproducible as possible, using the AUTOBK program developed by Matthew Newville. AUTOBK uses Fourier signal analysis to fit the spectrum in a manner that minimizes the low frequency oscillations that are thought to be components of the atomic absorption. The largest benefit of this procedure is that it is very reproducible and consistent in normalizing chi across different samples.

AUTOBK is a Fortran code, compiled in UNIX, which reads an input file that contains the relevant parameters. SAMXAS uses the `~/scratch` directory to store the files that

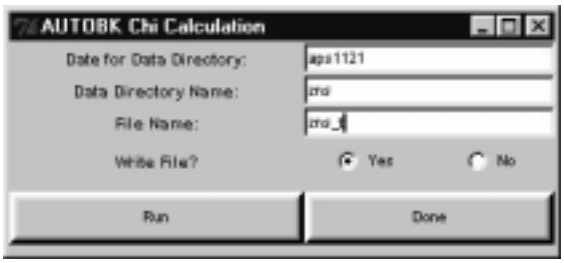

Figure B.3.10: Screenshot of the Chi Normalization module interface.

AUTOBK uses to read and output data. In a similar manner to the Mu Normalization Module, this module again samples the error in the measurements in a Monte-Carlo simulation. A normally distributed random error is applied to the raw spectrum at each step of the simulation. This new data file is saved in the `~/scratch` directory, as well as "`autobk.inp`" which contains the edge parameters of the element. The resulting output file is read back into SAMXAS and saved in a data array. After a thousand iterations, the results are averaged and the standard deviation determined as the error of the chi calculation.

Again, the module contains the usual data entry boxes, shown in Figure B.3.10. The file to normalize is determined by its group name, directory, and file name in these entries. The module is begun by pressing the "Run" button. The calculations take a fair amount of time, due to the constant need to write and read from disk. Once completed, several graphics windows are opened. The displays (shown in Figure B.3.11) include plots of the chi and its error (no k-weighting), the absolute error, and the k^3-weighted chi and k^3-weighted error. The final result is saved in the `chis` directory under the name `<fn>-k.chi`.

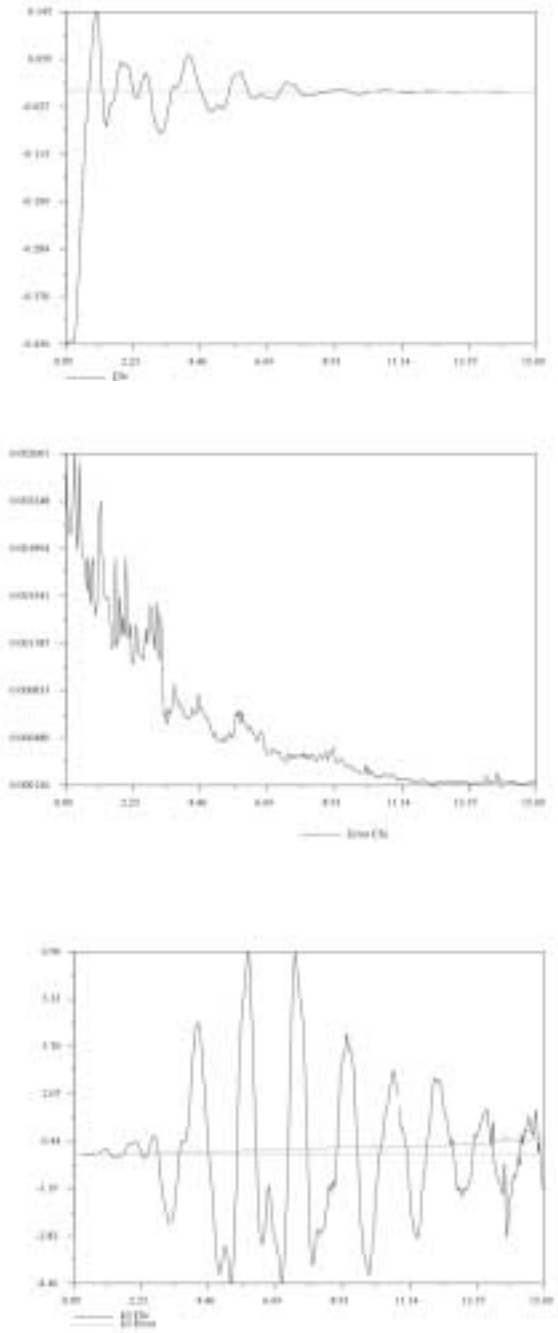

Figure B.3.11: Graphical windows produced by the chi extraction routine showing the EXAFS spectrum and the associated error.

Section B.3.6 View FT

The Fourier transformation (FT) module is one of the more widely used pieces of the SAMXAS package. Although it does not provide much in the way of quantitative analysis, it does provide for qualitative characterization and a check on quality of the data collected. The EXAFS signal that is calculated by AUTOBK is given in momentum space (also known as k-space) and has units of inverse angstroms. Thus, the magnitude of a Fourier transformation of the EXAFS signal gives the distances, in angstroms, of the major backscatterers around the central absorbing atom. The Fourier transform is often called a radial distribution function (RDF) or radial structure function (RSF). However, the transformation gives rise to a phase shift that changes the peak from its true position by approximately 0.2 to 0.5 Å. The magnitude of the RDF peaks will be a function of the distance from the absorber, the type of backscatering atom, and the number of atoms in

Figure B.3.12: Interface for the Fourier transform module.

the shell. Additionally, the back transform of a RDF peak can be used as a Fourier filtering step to isolate the EXAFS signal from a single shell. The SAMXAS module provides an easy interface from displaying these transformations and performing simple analysis.

The FT module interface is shown in Figure B.3.12. The first two data entry boxes provide the usual information of the desired file's location. The next frame enters the type of transform and the ranges of the spectrum to be transformed. If the "Forward & Back" button is selected, the program will also display the back transform and show the R-ranges of the back FT on the plot. Even if a back transform is not asked for, a range must be placed in these boxes. The next frame gives the option of applying one of three windowing functions to the spectrum during a forward transformation. Application of

$$W(k) = 1 - \left(\frac{k - k_{\min} - \frac{1}{2}(k_{\max} - k_{\min})}{\frac{1}{2}(k_{\max} - k_{\min})} \right)^2$$

windowing function forces the data to smoothly approach zero, typically at the first and last 5-10% of the data range. This reduces "ringing" of the major peaks in the FT by forcing the spectrum to be continuous at the edges. The square window provides no change to the data. The Welch windowing function has quadratic "sills" and is described by the following function.

The commonly used Hanning window provides a little more smoothing and is given by the following function.

$$W(k) = \frac{1}{2}\left(1 - \cos 2\pi\left(\frac{k - k_{\min}}{k_{\max} - k_{\min}}\right)\right)$$

The next frame allows any of the transformed data to be saved to a file. Forward transforms in R-space are given the filename `<fn>-r.dat`, and the back transforms in k-space as `<fn>-fk.chi`. Finally, the last frame selects the k-weighting to apply to the data and the padding to add to the data. Since the FT algorithm is actually a fast Fourier transform method, the data must have a length that is an integral power of two. Thus, the common practice is to "pad" the data with zeros. The program will not run if the number entered in this entry is not a power of two. Typically, the selected number is relatively large to insure a fine grid in the R-space result. The R-space grid is determined by

$$\delta R = \pi / N_{fft} \delta k$$

where δk is the k-space grid size and N_{fft} is number of final points after the data is padded. A common choice of N_{fft} is 1024 or 2048.

The module is run by clicking the "Run" button. Again, similar to the normalization modules, the error of the RDF is determined using a Monte-Carlo simulation of the error from the EXAFS. The error from the chi is weighted with the desired weighting scheme and the error sampled within a normal distribution. These simulated errors are added

to the data and a series of 1000 FTs performed. The resulting RDF is the average of the transform series and the error is determined by the standard deviation. Note that if the data is noisy or has a significant degree of error, the RDF peaks determined by this

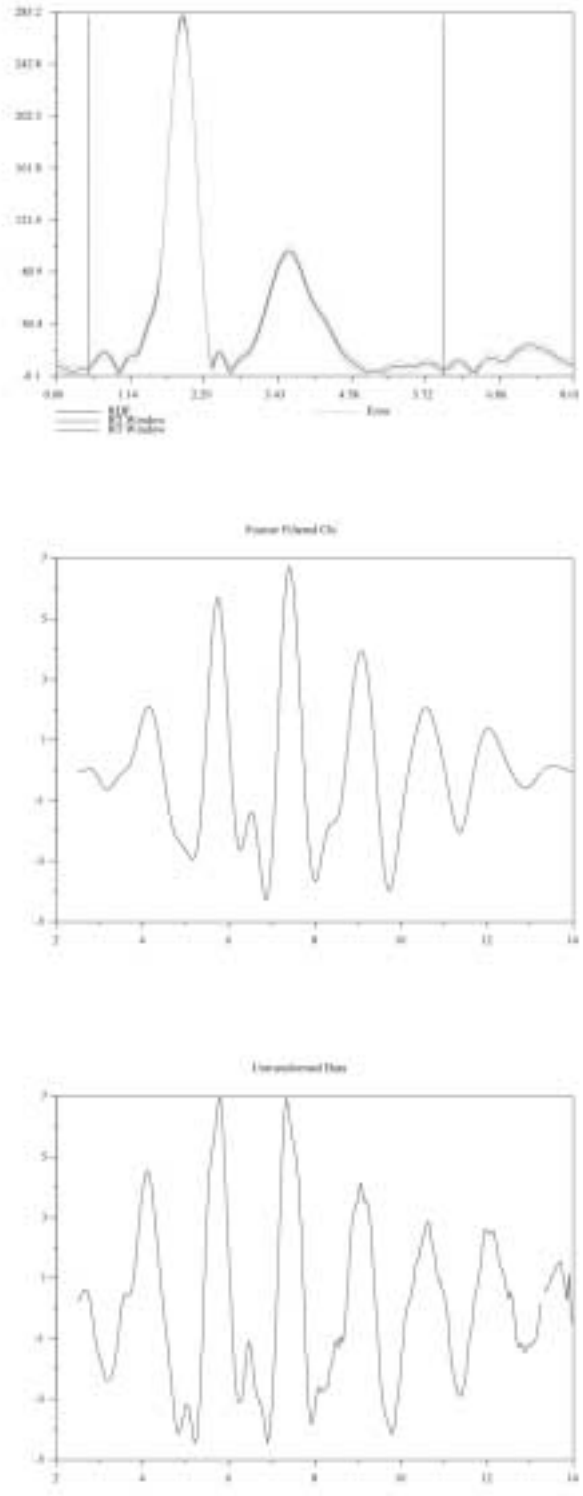

Figure B.3.13: Graphical output from the Fourier transformation module of SAMXAS.

process will be broadened and possibly swamped out by the excessive noise. Several graphics windows are created to display the results. These are shown in Figure B.3.13. Note that the error of the RDF is plotted as an envelope around the transformed data. The module also prints the height of the largest peak in the RDF and its location in R-space on the SAMXAS console.

Section B.3.7 Data Fitting

This is section of the SAMXAS package is probably the most complicated to use. It also the module which has been under the most development during the writing of the program. In general, when fitting experimental data, many iterations of the fitting process are required in order to fit data effectively. Many elements can contribute to a data fit "gone bad". The most common problem is data that is noisy, *i.e.* there are large errors associated with the data. This is most often because a sample is dilute or has a strong background fluorescence component. The errors should not be large in the standard, known compounds. If this occurs, the problem is likely to be in the sample preparation step. The second element of "bad" fitting is having an incomplete standard database. If there are components in the sample that are not included in the basis set of compounds, then the fitting procedure will not return the correct composition of the sample. The user must be careful to inspect the results with scrutiny to make sure that they are significant. Later in this section, the fine details of how to perform a good fit will be described.

The problem when it comes to fitting spectral data is to find a composition vector for a basis set of standard compounds that describes the features of the experimental data with in the limits of the error. An added limitation is that the results must have physical meaning, *i.e.,* it makes no sense to have a component present with a negative concentration. To make the matter complicated, the system is also overdetermined. There are at least several hundred data points in a data spectrum, yet there are 3 to 5 variables (the component proportions) to solve. In essence, the situation boils down to a classic problem of spectral deconvolution. There are many ways to approach this general problem and many algorithms have been used at one time or another in SAMXAS. A traditional least-squares method or singular value decomposition is simple and fast, but does not necessarily guarantee that the components will not be negative. A modified non-negative least square fitting may also be employed, but most solutions will end up with the sum of components that do not equal one. The current version uses a constrained minimization technique (quadratic linear programming) to tackle the problem. This is done since it allows for these types of constraints to be built into to the solving process.

The problem statement can be put in mathematical terms as:

$$k^3\chi_{\text{sample}}(k) = \sum_i f_i k^3 \chi_{\text{standard-i}}(k)$$

with the two constraints that:

$$\sum_i f_i = 1 \qquad\qquad f_i \geq 0$$

where f_i is the fraction of the metal bound in the coordination shell corresponding to the standard *i*. This just states that the k-weighted EXAFS of the sample must be the sum of the k-weighted EXAFS from each of the components. Additionally, the answer must be limited in the sense that we cannot have negative mass and all the mass must be accounted for (all components sum to one). The quadratic linear programming method takes the following approach. Given a vector of data spectrum, **D**, and a basis matrix of standards, **M**, we want to minimize:

$$\|\mathbf{Mf} - \mathbf{D}\|^2$$

where **f** is the composition vector of the sample. The minimization is subject to:

(a) sum to one constraint

$$\mathbf{1}^{\mathrm{T}}\mathbf{f} = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}\mathbf{f} = 1$$

(b) positivity constraint

$$f_i \geq 0, \text{for all } i$$

Thus, the linear quadratic programming problem minimizes:

$$\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{Hf} - \mathbf{c}^{\mathrm{T}}\mathbf{f}$$

where $\mathbf{H} = \mathbf{M}^{\mathrm{T}}\mathbf{M}$, and $\mathbf{c} = \mathbf{M}^{\mathrm{T}}\mathbf{D}$ subject to the same constraints given above. Again, the error in the fitting procedure is determined by using a quasi-Monte Carlo method. In the fitting case, the error is sampled normally and applied to both the sample and standards. Thus, the noise level of the data files can be crucial to achieving good fitting results. The final fitted value is determined by averaging the results of the 1000 replicates and the error of the fit by the standard deviation of the result.

As mentioned earlier, fitting experimental data is not simply a single attempt process. In order to verify that the results are significant, many iterations of the fitting process are normally performed. The basic requirement of a good fit is that the combination of standards should be able to reproduce all of the major structures in the EXAFS pattern and most of the secondary structures. An example of a major structure is the large oscillations, whereas secondary structures refers to features such as shoulders and dimples in the major oscillations. It is common that an initial fitting using SAMXAS will include all of the major entries in the users database. However, it is rare that all of these compounds are present in the sample. The results of the first fit will usually describe the experimental data sufficiently, but will include several components that are present at amounts smaller than the uncertainty of their composition. These should be removed one at a time, starting with the smallest first. As the solution iterates, one should always verify that the removal of the last component has not significantly degraded the way the fit describes the experimental data. Additionally, components that are present as less than 5 to 10 percent are candidates for removal, as it is unlikely that the contribution from their spectra will be significant relative to the errors.

The fitting module allows comparisons of the fitting results to be made with both the XANES and EXAFS portions of the spectrum. This allows a useful check on the fit solutions. For example, if an EXAFS fit is performed that appears to satisfy the user, that same solution should also be able to describe the major XANES features as well. Thus, any glaring problems with the fit may be noticed in the XANES reconstruction of the fit.

The XANES region is primarily used in this way as a verification procedure, as the XANES region has fewer features and oscillations. Additionally, at high energies (*e.g.* Cd) the XANES features of differing compounds are nearly identical. This is not to say that the XANES spectrum does not contain useful information. The XANES is very sensitive to the coordination geometry and oxidation state of the central absorber. However, the effects are complicated and the synergies are not clear, thus care must be taken when performing a direct XANES fit.

Another process that can be used to justify the number of components used in the fit is principal component analysis (PCA). PCA is a mathematical manipulation of the data that determines the number of independent components needed to describe the system. For instance, one can place all of the spectra for a set of core samples from sediments into a file and run the analysis. Since the SAXAS fitting is accurate to approximately five percent, we could use the PCA to determine how many components are needed to explain 95 percent of the variance. When going back to the SAMXAS fitting, we would be unjustified using more than that many standards to fit the samples. SAMXAS at this time does not incorporate a PCA routine, but there are many available on the market, such as Minitab.

The interface for the SAMXAS fitting is shown in Figure B.3.14. The top frame has the normal data entries for the path and name of the sample file to be fit. The second frame has entries for up to eight standard file names to be entered. Next to each path is a checkbox in the “Use Me” column. Only those standards which have the box checked

Figure B.3.14: Data fitting module interface.

Figure B.3.15: Parameter input box required for performing fits on EXAFS data. This box is created whenever the "EXAFS Fit" radiobutton is active in the "Data Fitting Manager" window.

will be used in the fit. This allows rapid adjustment of the number and types of standards used in the fitting process. The next frame specifies whether the EXAFS portion or the XANES portion of the spectra will be used for the fit. If the “EXAFS” radio button is clicked, then the EXAFS fit parameters box is also created (Figure B.3.15). The EXAFS fit parameter box contains many variables similar to the Fourier transform module (Section B.3.6). The ranges in k-space to extract the data from and R-space for backtransform are selected in the upper frame. If the “FT Filtering” radiobutton is selected, the backtransformed data will be fit, opposed to the raw, unfiltered data. Lastly, the windowing scheme, padding lengths, and k-weighting are chosen. These values must be entered, even if your fitting method does not employ a FT. The next frame in the fitting module window determines how the results will be displayed. If “Compare Fit to EXAFS” is checked, then an EXAFS reconstruction based on the fitting results will be compared to the EXAFS of the sample. Similarly, if “Compare Fit to XANES” is checked the reconstruction based on the XANES data will be made. The user can click on one, both, or neither of the boxes, depending on the need.

The buttons at the bottom determine the actions of the module. If “Default” is clicked, common components are entered for a zinc analysis. The “Run” button begins the fitting procedure and sends text output to the console as well as creating several graphics windows (Figure B.3.16). The text output depends on which comparisons are being made, *i.e.* with the XANES or EXAFS reconstructions. The first outputs are an acknowledgement that the all of the correct files were loaded and that the fit has been completed. After this, the names of the standards are displayed across the console in

table headings. The first number below the name of the component is the percentage of that species found in the sample. Again, results below the range from 5 to 10 percent should not be trusted, and that component should be eliminated in the next fit. Below these results are the estimates for the error of the composition, given by X standard deviations of the Monte Carlo method.

The quality of the fit is reported next for each of the reconstructions performed. For example, if the fit result is only compared to the EXAFS reconstruction, only the EXAFS quality of fit is given. If the fit is compared to both the EXAFS and the XANES data, then the quality of the fit is reported for both data regions. The quality of fit is reported in two different parameters, the chi-square value and the R-value. The chi-square is given in a traditional sense of the term, defined as:

$$\chi^2 = \sum \left(\frac{x_{fit} - x_{data}}{\sigma_{data}} \right)^2$$

In the EXAFS reconstruction, the chi-square is also reported in as relative and normalized values. Both of these terms attempt to normalize the chi-square in terms of the number of independent data points in the fitting and the number of standards used in the fit. As always, caution should be used in using these values. A small decrease in these parameters can nearly always be achieved by increasing the number of components in the fitting process. If the new fit does not explain any extra features in

the EXAFS spectrum, is unlikely to be truly important in the fit. The R-value is another attempt at defining the "goodness of fit" and is defined as:

$$R = \sum \frac{\left(x_{fit} - x_{data}\right)^2}{\left(x_{data}\right)^2}$$

The R-value is a measure of how close the fit is to the actual data based on a sum-of-squares determination of the fractional misfit. Typically, R-values greater than a few percent should be considered unacceptable.

If one of the data reconstructions performed is an EXAFS fit, the data is also fit using singular value decomposition (SVD). The results of the fit as well as an estimate of the error are printed in the SAMXAS console under the same columns that were displayed in the first fit. In this case, the error is not determined by a Monte Carlo process, but rather through the covariance matrix used in the decomposition. This is another check on the fitting process. Although it is not a non-negative fitting process, it can give useful insights into which components in the fit are truly important. For example, components that are reported as negative results in this fitting scheme are good candidates for removal in the next fitting iteration.

Section B.3.8 Quit

This option is pretty much self explanatory. The program is ended and all graphics windows closed.