The K-space and Background Removal Program

Matthew Marcus

October 18, 2006

I. Introduction

This program is used to get from raw 2-column EXAFS data (signal vs. energy) to k-space data ($k^n\chi(k)$) vs. k). It does the following steps:

1. Pre-edge removal. You specify a linear background and the program subtracts it.
2. Conversion of abscissa to k-space and pruning of data. You specify the assumed edge energy and the start and end of the data range, and it does the abscissa conversion.
3. Cubic spline background removal and multiplication by k^n. You specify the weighting exponent and the maximum distance between knots. It places knots equidistantly (in k) and does the fit. It then computes the EXAFS as:

   $$\chi(k) = (y(k)-f(k))/f(k)$$

   where $y(k)$ is the signal and $f(k)$ is the fit.
4. Display of results including FT and file saving. It displays but doesn't save the FT magnitude as that is left to the FT program.

The spline fit is to cubic splines, and is weighted by the k-exponent unless you tell it not to be via front-panel switch.

II. Pre-edge removal

When you start up the program, you get a file dialog with the default input extensions set to `.r, .t`. The former is the default output extension for 2-column EXAFS files produced by the EXAFS Editor, and the latter is the default for pre-edge

background subtracted files (see below). If the input file is a `.r` or if you press the Remove pre-edge button under the E-space plots tab, the program then pops up the screen shown in Figure 1, otherwise it stays at the main screen explained below. The data are shown in a big graph along with two cursors. You can define the pre-edge, shown as a red line, in several ways, controlled by how you set the Mode control:

Fit between cursors

Fits a linear function to the data between the cursors (y-values of cursors ignored).

Line through cursors

Draws a line through the cursor positions.

Transmission fit

Like Fit between cursors, but transforms the abscissa to $E^{-2.7}$. This does an approximate linearization to transmission data.

Edge fit between cursors

Fits to the following form:

$$y = a + bE + c/(E_0 - E)$$

where E_0 is adjustable by means of a slider which appears when this mode is selected. The purpose of this mode is to approximate the initial rise of the edge. Only the linear part is subtracted off, as represented by the red line. The *b* term is only allowed if the Linear Term? switch is up (default).

Elastic fit

Fits to $y = a + bE + c/(E - E_f)^2$, which roughly approximates the way the elastic peak comes into the tail of the detector response. E_f should be set to the middle of the fluorescence ROI, which is usually close enough to the tabulated fluorescence energy so that you can use that value instead. Again, the data are fit between the cursors, and the linear term appears only when the Linear Term? switch is up.

Edge&elastic fit

Fits to $y = a + b/(E_0 - E) + c/(E - E_f)^2$ with the constant term optional. As with Edge fit, the $b/(E_0 - E)$ term is not subtracted from the data.

After you are satisfied with the pre-edge, hit the Save button to save the pre-edge file with a .t extension. This will be useful for XANES analysis and also as a way of not having to redo the background subtraction if you want to change the *k*-power or E_0. Hit the Accept button to proceed. You can re-do the pre-edge by hitting the appropriate button on the Background or E-space plots tab (*v.i.*). This is sometimes handy for 'polishing up' the pre-edge, e.g. by doing a Transmission fit followed by an Edge fit.

III. Main screen

You should now see the screen shown in Figure 2. The top graph shows the raw data in white with the spline knots in green, the spline background in red, and three cursors, in yellow, green and red. The yellow cursor is the E0. The program tries to guess where that should be, but you should always check and modify it according to whatever convention you're using. You can use the graph tools to expand the edge region so you can see what you're doing. The green cursor marks the start of the data range. The form of the background often depends sensitively on this position, so you should choose it so that the background isn't driven too hard by structure at the edge. Like much of EXAFS analysis, this is somewhat of a matter of taste but should be done consistently. The red cursor marks the end of the data range. If the data are noisy or glitchy at the end, using this cursor to restrict the data range will help keep the spline fit from being driven by noise or artifacts.

The lower graph shows the resulting EXAFS signal. The green squares show the positions of the spline knots. The ordinate label automatically changes to reflect the exponent being used.

The graph to the lower left shows the FT magnitude in white and that of the spline itself in red. Both are done using a Kaiser–Bessel window You can monitor the cleanliness of the signal by looking at that and can adjust the data bounds accordingly. By default, autoscaling is turned off on this graph, so you may have to click on the y-axis autoscale tab to get the graph to show up properly. You can use the height of the first significant peak as an indicator of whether you have too many knots.

The Background tab (shown in Figure 2) is used to modify how the background subtraction is carried out. By default, you are doing a cubic spline fit with a settable number of knots. You can change the number of knots by adjusting the Max delta-k control. The number of knots is shown in the obvious indicator. The k-power is set by default to 3, which is the most popular value. Change this if you want some other value. The button called Look at spline filter effect is used to get an idea of how much of the EXAFS for a given shell is 'eaten' by the spline background. This button pops up a screen as shown in Figure 3. The idea is as follows: Make up an artificial EXAFS signal which looks like:

$$y_s(k)=1+k^{-n}\sin(2kR)$$

and feed it to the spline routine. Now, the spline will 'attempt' to follow the wiggles in this signal. The white curve in the Wiggle graph shows the result of this fit, with a k^n multiplier. Similarly, we make up another artificial signal

$$y_c(k)=1+k^{-n}\cos(2kR)$$

whose spline fit is shown in the red curve. The average of these two, in quadrature, is shown as the white curve. Now, to see how much either fit resembles a sine wave at the given frequency, we look at what is effectively a Fourier component of the fit:

$$F=[2(<\cos(2kR)f(k)>^2+<\sin(2kR)f(k)>^2)]^{1/2}$$

where <...> means an average over the data range. If the spline matched the input wiggle exactly, that is 'ate up' all the EXAFS, this quantity would be 1. If the spline didn't resemble the wiggle at all, F would be nearly 0. The F values for the sine-wave ($y_s(k)$) and cosine-wave ($y_c(k)$) test wiggles are shown as Sin amp and Cos amp, with the quadrature average of these two quantities being Avg. amp.

Thus, the idea is that one slides the R(A) slider around and looks at the average quantities and wiggle graphs to see how short an apparent (no phase shift) distance one can detect without serious spline effects. In the example shown here, we see that the amplitudes are around 20%. Amplitudes below about 10-15% seem to guarantee that the Fourier magnitude is unaffected and that the spline doesn't follow the EXAFS too closely. The present example is one where the apparent distance has been set short in order to illustrate a spline-fit effect. Notice also that the wiggle graphs show sinusoidal-looking curves, rather resembling the original wiggles. If one moves the R(A) slider to the right to select a longer apparent distance, these curves lose amplitude and become less smooth, with pronounced breaks at the knots. The amplitudes never go much below 10% because the data range isn't infinite.

There is a slider on the Background tab called Spline squeeze. By default, it's set at 1.0, which makes the spline work as described above. If you move Spline Squeeze to smaller numbers (left), the knots at low-k get farther apart and those at high k closer together. Moving this control to the right squeezes the spline knots closer together at the start of the spectrum. This can be handy for adjustment of the

background. Look at the FT plot and the Look at spline filter effect window to make sure you're doing the right thing.

The Background tab has two tabs within it. We have been working with the Knots tab. The Tension tab, shown in Figure 4, offers an alternative way to do the fitting. Here, the spline is a continuous function which gets 'stiffer' as Tension increases from 0 (follows the input exactly) to 1 (a straight line). The values required cover a wide range, which is why the control has a logarithmic labeling scheme. The digital display to its right can be used for fine adjustment as each 'click' of its spinner yields a 1% change in the value. The Initial Weight control adjusts the ratio of the stiffness between the left and right sides. These two controls interact in a way best probed by experimentation.

After inspecting the spline effect, you can return to the main screen. The tab control has two more tabs, shown in Figure 5. The File I/O tab has the obvious functions, but it also offers an entry into the FT. The k-space file is the output you usually want - $k^n\chi(k)$) vs. k. It has a `.b` extension by default, which matches the input conventions for the FT, Linear Fit and PCA programs.

There is also a Write k^0 interpolated file button. This creates a file with k^0 weighting, interpolated to a uniform grid with interval 0.05A-1, starting at 0. Such a file is what IFEFFIT-based programs like Artemis or Sixpack want. The default file extension is `.chi`, as used by Artemis.

The Enter FT program button takes you into the FT program as if you'd double–clicked its executable. A file dialog will pop up. If you have written a `.b` file, you can push the FT k–space file button, which will take you into the FT program with no file dialog. The FT will be done on the file you've just written.

The E-space plots tab has these functions:

1. Remove pre-edge. This brings up the initial screen seen in Figure 1. Use this if you decide you don't like the way you did the pre-edge when you started. This is typically used with transmission data, for which you do the Transmission pre–edge, then Edge fit.
2. Differentiate? If you flip this switch up, it differentiates the data shown in the upper graph on the main screen. This mode is good for locating an edge by the traditional prescription.
3. Manceau E0. This is an attempt (not altogether successful) at implementing Manceau's favorite edge criterion, that is taking the edge to be at the halfway-up point. It evaluates the edge jump by extrapolating the spline fit to the edge position. It then finds an edge position which is halfway to that point. This procedure is iterative; pushing the button performs one iteration. The problem comes when there is a significant white line in that the spline background will be driven by that white line. A possible cure for this problem is to move the start-of-data cursor (green) to a point at which the spline background is smooth.
4. Normalize? In the default (Edge jump) position, the `.t` file is divided by the edge jump computed as above, otherwise it's unnormalized.

To summarize, the procedure is:

1. Do pre-edge subtraction and return to the main screen.
2. Select the *k*-power on the tab control.

3. Select the number of knots (Delta-k), E0 and the data range with reference to the FT and the spectrum on the bottom left and right, respectively, or alternatively go to the Tension tab and adjust the parameters thereon.
4. If you want to make sure your spline isn't too floppy, use the Look at spline filter effect button to see how the proposed spline fit treats artificial data.
5. Write out the files(s).

This program does not contain any tools for handling curved pre-edge backgrounds or post-edge backgrounds other than cubic splines. However, the backgrounds it does provide seem to be sufficient for most uses, especially if applied consistently between samples and references.

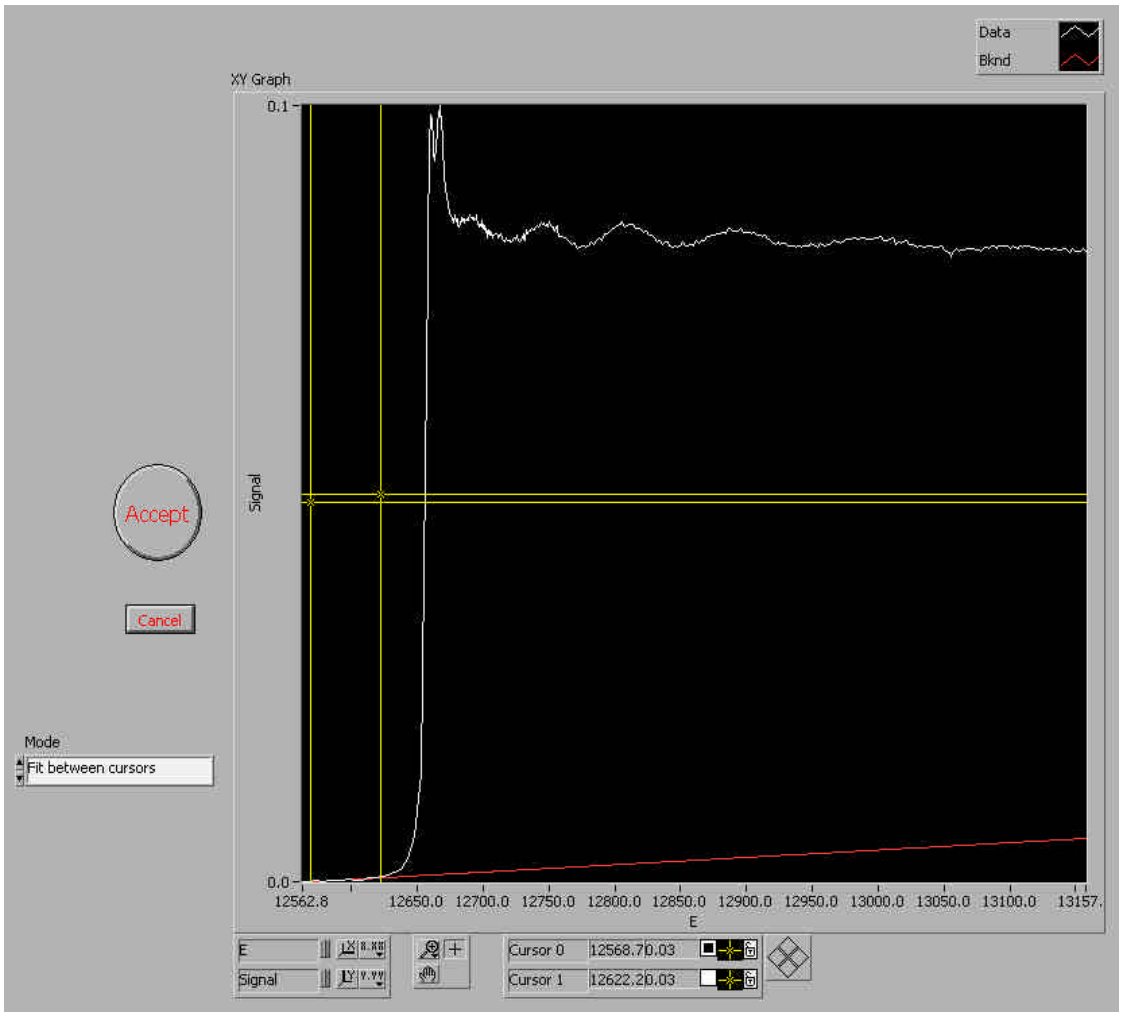

Figure 1. The pre-edge removal screen. The data with abscissae between those of the yellow cursors is fit to a line, shown in red, which is subtracted from the data..

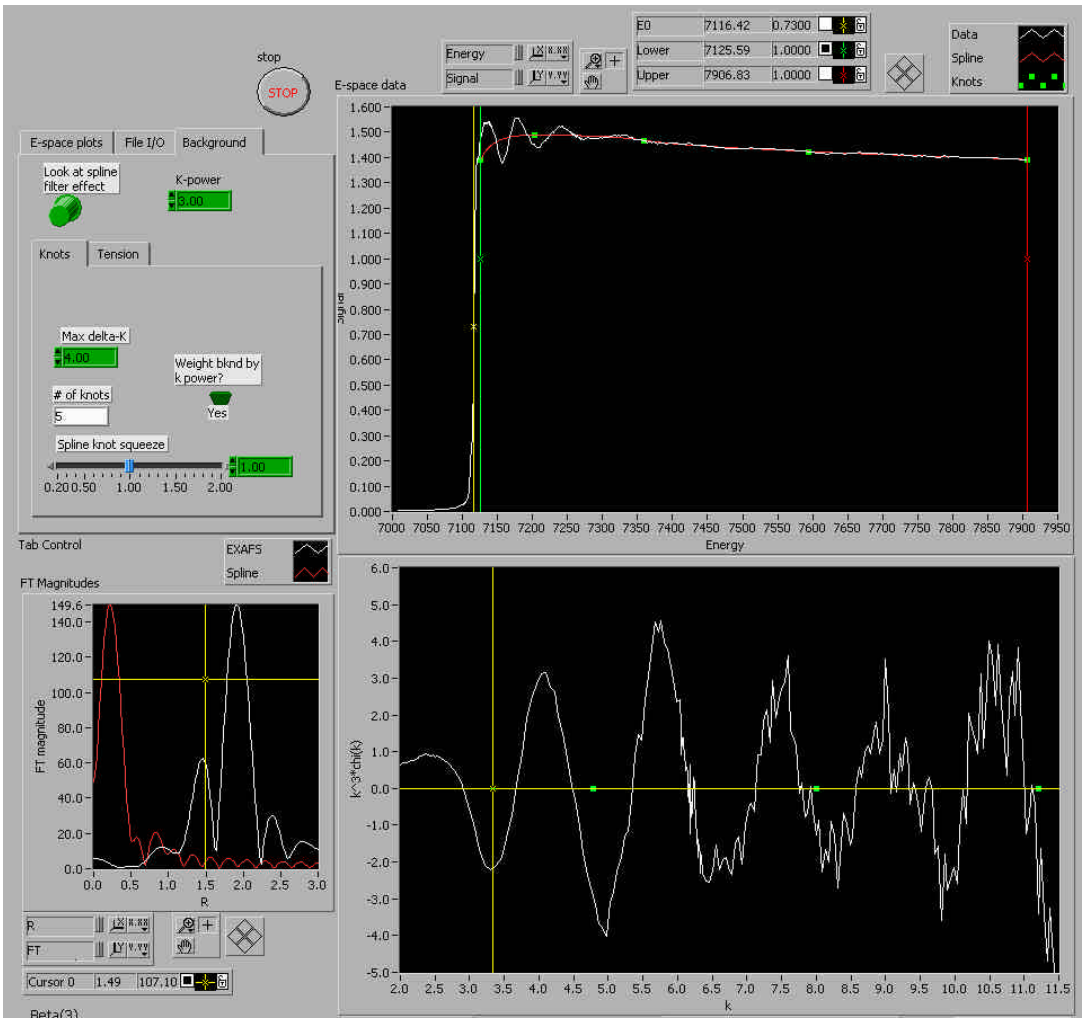

Figure 2. The main screen showing the Background tab. The spline background has five knots and the weighting power is 3.

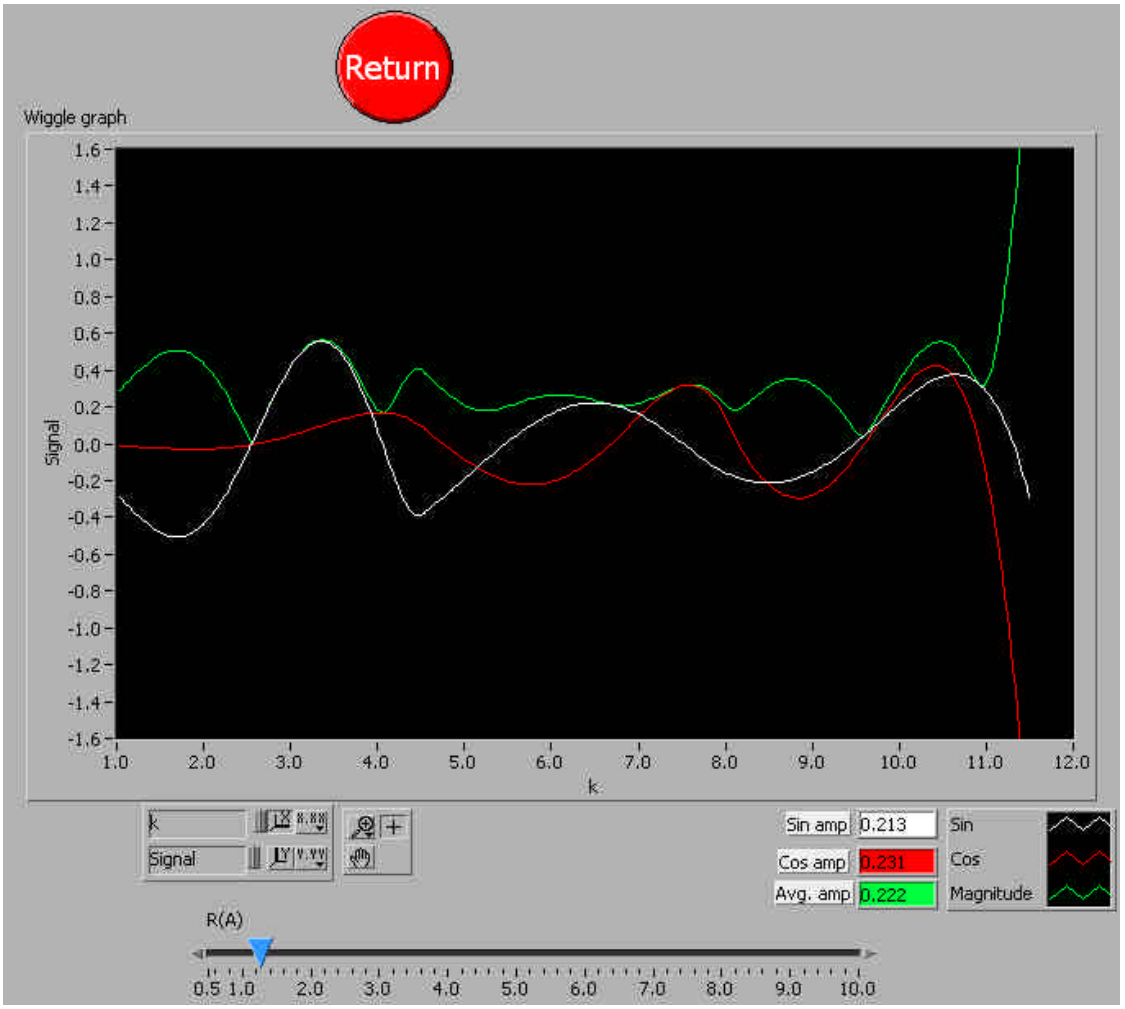

Figure 3. Spline-effect screen. The distance R(A) is set low on purpose in order to demonstrate the effect of a spline fit which compromises the EXAFS.

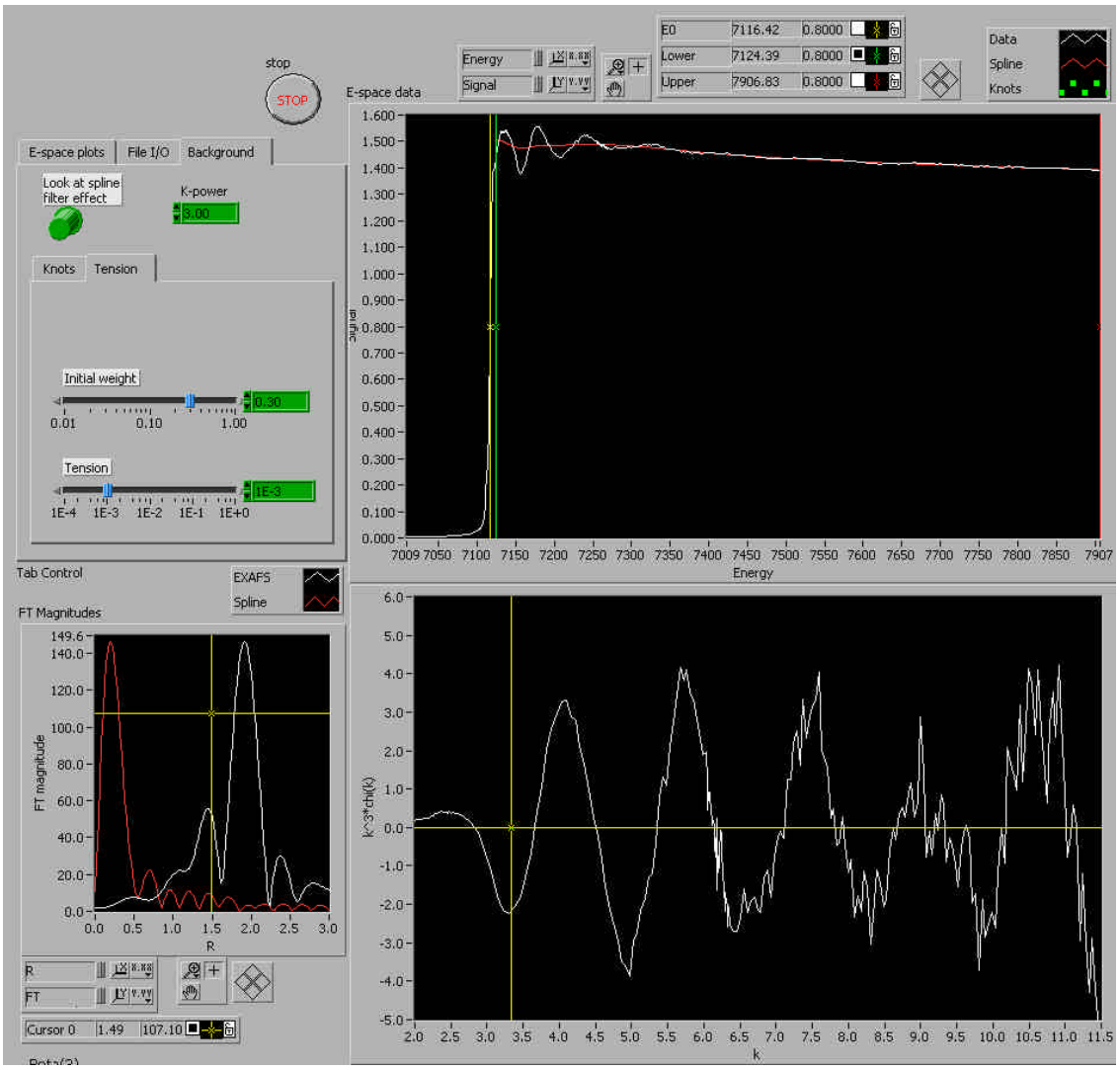

Figure 4. The Tension tab under Background.

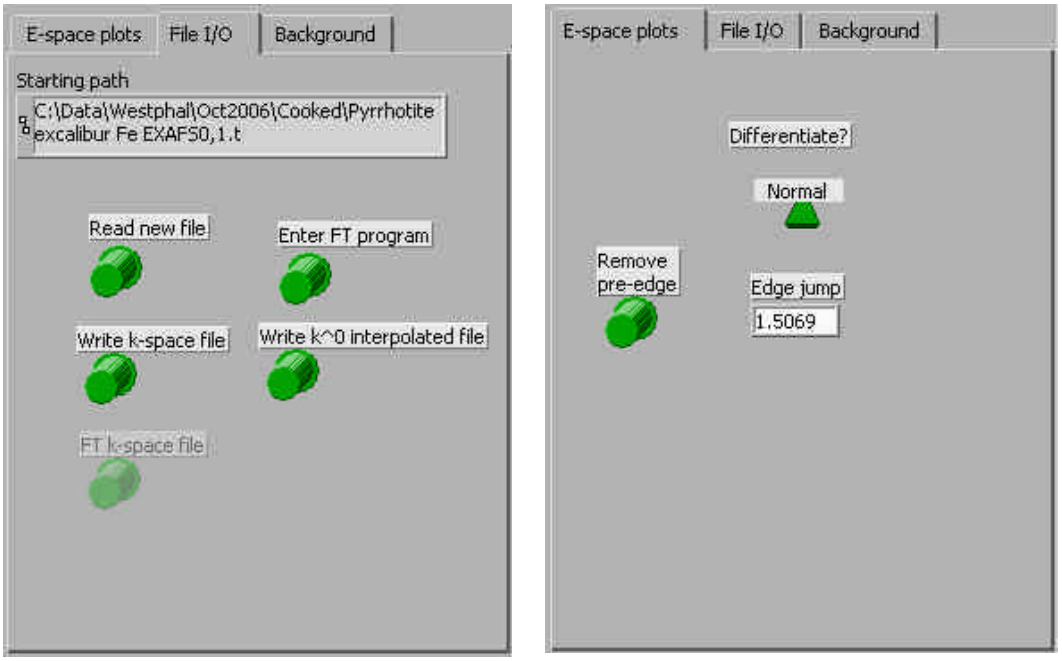

Figure 5. The two tabs not shown in Figure 2.