New Staffing:
I've hired an undergraduate at Sarah Lawrence who can do some video editing and Java programming.

Timeline:
Fitting criteria module material done and delivered to Steve: October 24

FT applet development complete: November 18
FT material (including text) complete and delivered to Steve: November 28

Foruier transforms outline:

Contextualization:

Begin by reminding users of EXAFS cartoon physics and k-space. If users are not already familiar with the idea of k-space, they should be sent back to the appropriate module. (If that module does not yet exist, they can just be told they should be familiar with it first.)

I have a Python applet that shows the outgoing and scattered waves, and shows the resulting absorption. It can also be set in a mode where it "scans," changing energy and collecting data. It also allows for atoms to be moved to different distances.

Perhaps my student can adapt that applet to Java, and have it display the result in k-space.

Through discussion of the applet, get user viscerally familiar with the relation between interatomic distance and k-space oscillations.

Need for Fourier transforms:

Show a second scattering atom at a different distance, and how that changes k-space.
Simple animation shows contribution from each scatterer in k-space.
Introduce R-space, and show spikes corresponding to absorber-scatterer distances.

Complex transform:

Magnitude gives k-space frequency, phase gives k-space phase. (Show some figures to make this clear, like a phase shifted k-space without changing frequency.) Can also be expressed as a real part and an imaginary part.
In this kind of FT, real part alone (or imaginary part alone) has all of the information of both magnitude and phase.

Physical Complications:

This part can be handled primarily by bullet points, with perhaps a few simple figures:

--Atoms are not hard spheres, causing shifts in where peak appears in FT

--Atoms vibrate, causing absorber-scatterer positions to vary and broadening peaks

--Static disorder also causes absorber-scatterer positions to vary. This can be random (like in a glass) or systematic (e.g. unresolved split shell)

Mathematical Artifacts:

So far, we've been discussing FT as if we had continuous data on an infinite domain. Real data is neither.
Since data is discrete, there is a maximum in R-space. In practice, as long as k-space is sampled at reasonably short intervals (0.05 inverse angstroms is good), this is not a concern for EXAFS analysis.

But the finite domain introduces more serious artifacts.
(Controversy alert! I am fully convinced there are two ways to discuss these artifacts. One is to consider the finite domain to result from the convolution with a boxcar function. The other is to treat the function as if it repeats outside of the finite domain. Both are mathematically equivalent. I prefer the latter description, as I think it leads to a more intuitive understanding of windowing. I also don't think it's necessary for people learning this stuff to immediately be familiar with both descriptions. So I'm planning to only give the description of the FT of a function on a limited domain as being equivalent to the FT of the same function repeated an infinite number of times. There are some in the EXAFS community who don't like that description, though.)
Ideally, we have an applet here. It shows a sine wave, and lets the user move the right end of a boxcar window around. It then shows the "infinitely repeating" version of the function, along with the FT.

One disadvantage of teaching this in this manner is that it may lead people to believe that they should favor zero-crossings as endpoints for the window. It's pretty easy to make the argument, however, that a zero-crossing of chi(k) is not likely to be a zero-crossing for the individual paths that make it up, and thus choosing a zero-crossin is simply going to modify different paths in different ways. A few figures will make this clear.

FFT:

Finally, note that FFT algorithms such as that used by iFeffit zero-fill to make the number of points a power of 2. Show what that looks like, starting with data, and then showing it zero-filled, as a repeating function, and finally the FT.

Windows:

Stress that in fitting the same procedure is followed with fits and data, or samples and standards. (Give just enough about fitting to let people who got to this module because they want to do LCA or PCA on FT's the basic idea.) Therefore artifacts aren't as severe a problem as they first appear.
But, if they broaden peaks too much, they may hurt the ability to resolve peaks. And if the sidebands are too pronounced, they may cause paths from outside the range of interest to make a significant contribution. (Show both with figures)

Therefore windows are generally applied to bring the data more smoothly to zero, or perhaps a small value, at the endpoints of the windows. Repeat the example of zero-filled data with some common windows along with the infinitely repeating version and the FT, to show how windows help.

Emphasize that the particular window chosen should not affect the analysis in a substantial way.

k-weighting:

Show data that dies off quickly.
Suggest that analyses on that data will tend to emphasize the low-k part of the signal.
Note that different elements tend to have contributions that peak at different values of k-space (use the typical figure).
Note that if different paths have different amounts of disorder, that will also affect their relative importance in different regions of k-space.
Therefore we usually want to have all of our chosen k-space region contribute significantly to the FT.
The solution is k-weighting. Show k-weighting with FT (perhaps an applet allows choice of kw 0, 1, 2, or 3). Indicate again that it is done to both fit and model, if fitting is being done. If PCA or LCA, of course must be done the same way to all samples and standards.