Alan’s steps to reconstruct data and observations.

1. Initialize paths. Type ‘toolbox\_path’ at the Matlab prompt. Choose ‘Yes’
2. Load processing GUI. Type ‘ProcessGUI’ at the Matlab prompt.
3. Load data. Go to File -> Load Data. Load the .d01 file
4. Load scenario. Go to File -> Load Scenario. In the $TOOLBOX\_ROOT/epri/Scenario folder choose ‘PulseRecon.scn’
5. Then load parameter file. In the same folder as above choose ‘Pulse T1inv RAPID processing.par’
6. Select the profile correction file. In the drop-down selection list that reads ‘Data load parameters’ choose ‘FFT and projection correction’. Then in the property list, for ‘Profile correction file’ (open file selection dialog by highlighting this row, and clicking on the ellipses (…) button below the property list), choose the ‘cavity\_profile.mat’ file that came with the data
7. Select the output folder. On the bottom-right side of the screen, select the ellipses (…) button, and choose the desired output folder for the reconstructed images.
8. Begin processing. Click on the ‘Process’ button. For the ‘Value (fbp.Q)’ popup, choose the default by clicking the ‘Ok’ button. For the ‘Value (fft.profile\_file)’ popup also choose the default by clicking the ‘Ok’ button.
9. Data should reconstruct.
10. Recon can be intercepted in the $TOOLBOX\_ROOT/epri/epri\_reconstruct.m file. For our purposes right before Line 46 seems appropriate. I inserted a ‘keyboard’ statement here to take a look at the data.
11. The sinogram data can be converted back to k-space with ‘A = fftshift( fft( in\_y, [], 1 ), 1 );’ . Looking at this data (‘figure; imagesc( abs( squeeze( A(:,1,:) ) ) )’) it appears the center of k-space oscillates (signal is not constant across all projections at the center of k-space). It would be interesting to see projections along the 3 physical gradient axes to see if this could be corrected prospectively or adjusted retrospectively in reconstruction
12. It appears gradient amplitudes are listed in mat\_info.GradX, mat\_info.GradY, mat\_info.GradZ. This should make it easy to define the k-space trajectory for frequency encoding lines with endpoints defined by these three variables. ‘figure; plot3( mat\_info.GradX, mat\_info.GradY, mat\_info.GradZ )’ gives an idea of the gradient ordering.