QCPMG processing in ssNake

5th February 2018

1 Introduction

The following will explain how QCPMG NMR data can be processed in ssNake. The tutorial delivered with the ssNake program is considered as prior knowledge. If you have not yet studied this, please do so before continuing with this example.

QCPMG-sequence stands for Quadrupolar Carr-Purcell-Meiboom-Gill sequence. The sequence is used to measure a consecutive series of echoes, which are all recorded. Via careful processing the signal-to-noise ratio of this data can be higher that if only a single echo would have been recorded. The experiment is particularly helpful for broad quadrupolar patterns, where the decay of the signal is much faster than the intrinsic decay of the magnetization (T_2). In this case, many echoes can be recorded in one go.

QCPMG experiments can be processed in number of ways to produce either a spikelet pattern, or a more regular NMR spectrum. Both have their advantages in specific circumstances.

2 Data

The used data is a ^{35}Cl spectrum of magnesium chloride ($MgCl_2$) recorded at 20 T using 15.6 kHz MAS.

3 Processing

3.1 Sum echo spectrum

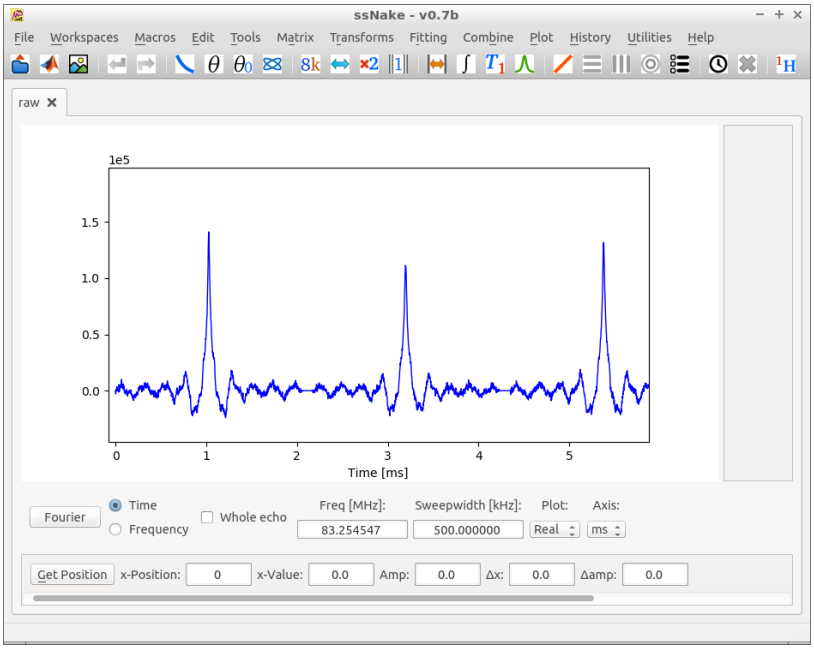
The most general way of analysing the series of echoes that are recorded in a QCPMG experiment is by adding all the individual echoes, and process it as you would a single echo experiment. This gives a spectrum that is easy to interpret and has a higher signal-to-noise when compared to a regular echo. However, processing can be tricky, and requires a careful method.

The data supplied in this tutorial has 137 measured echoes, consisting of 1088 data points each. In order to add these, we must split the data in 137 parts, to form a pseudo 2D data set with shape 137×1088. Setting the number of echoes is done when measuring the spectrum, but can also be counted afterwards. Getting the number of points can be more tricky. Dividing

the number of data points by 137 should give this, but in our case the QCPMG data has zeroes appended to the echoes. What we do know is that only with the correct number of points (1088) the splitting of the data gives an aligned result.

- Open the data in the `Raw` directory

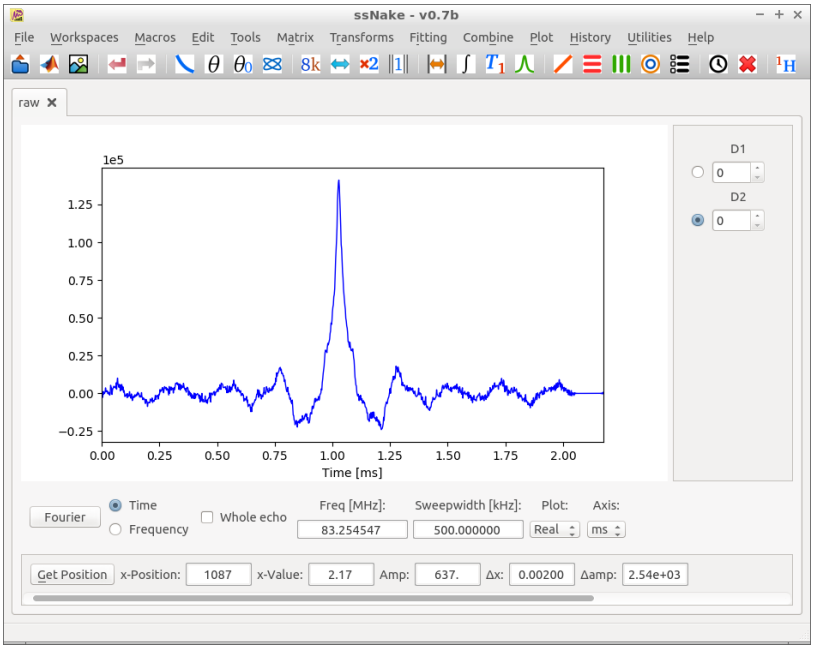
Each echo has some zeroes appended to it, which is a consequence of how the Varian equipment is measuring the QCPMG data. This can be seen in the following Figure (from the start of the FID):



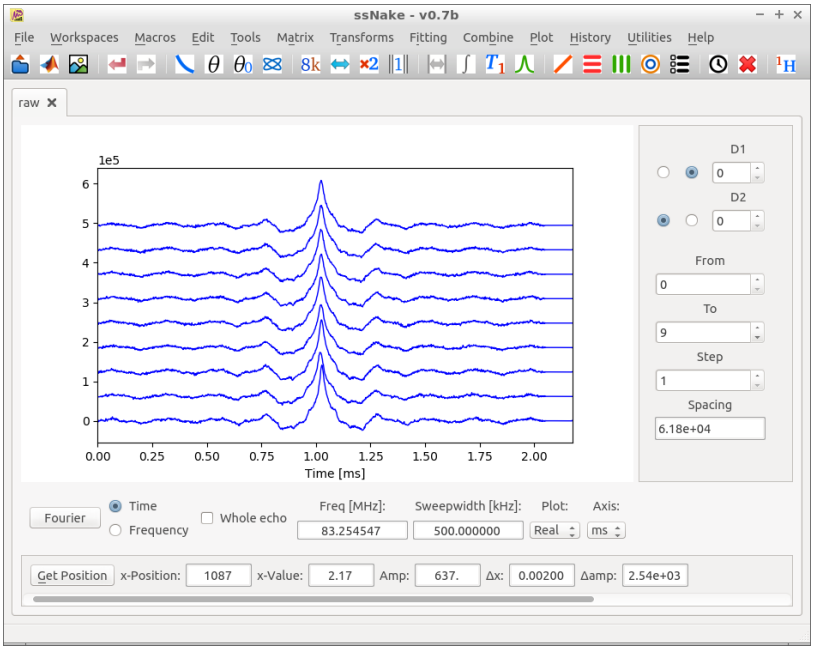
which shows the first 2 and a half echoes. We now must split the data in the 137 echoes.

- Set the size to $137 \cdot 1088 = 149056$ data points (Matrix $\longrightarrow$ Sizing)
- Split the data: Matrix $\longrightarrow$ Split (137 sections)

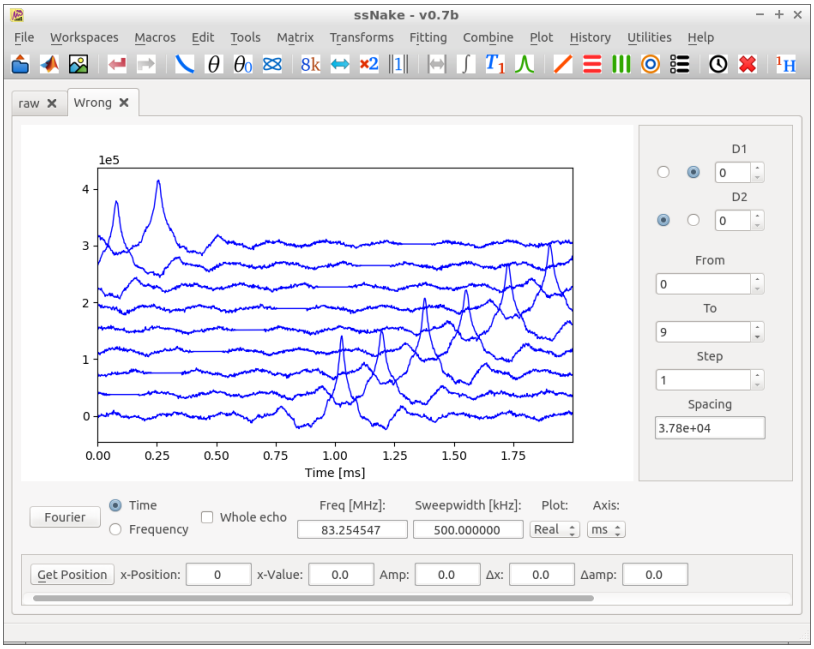
Now we have all the separate echoes. Viewing D2 gives:



which shows the first echo. Scrolling through D1, we can see that for each echo, the position is the same: the splitting has been performed correctly. A stack plot of the first 10 echoes for example gives:



which looks good. If we had used 1000 points per echo we would see:



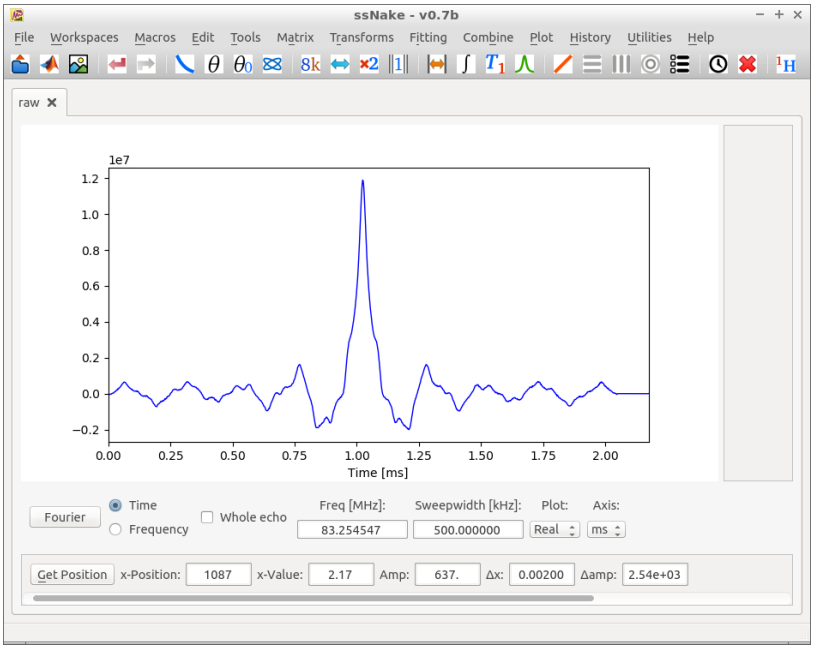
which is clearly very, very, wrong. A good check is to overlay the first and last echo. The 'zero' part at the end should be at the same position.

3.1.1 Option 1: Directly summing the echoes

Using the properly split data we can sum the echoes:

- Go to D1 (using the radio button in the side frame)
- Sum along this dimension using: Matrix ⟶ Region ⟶ Sum (with no input, just push Ok).

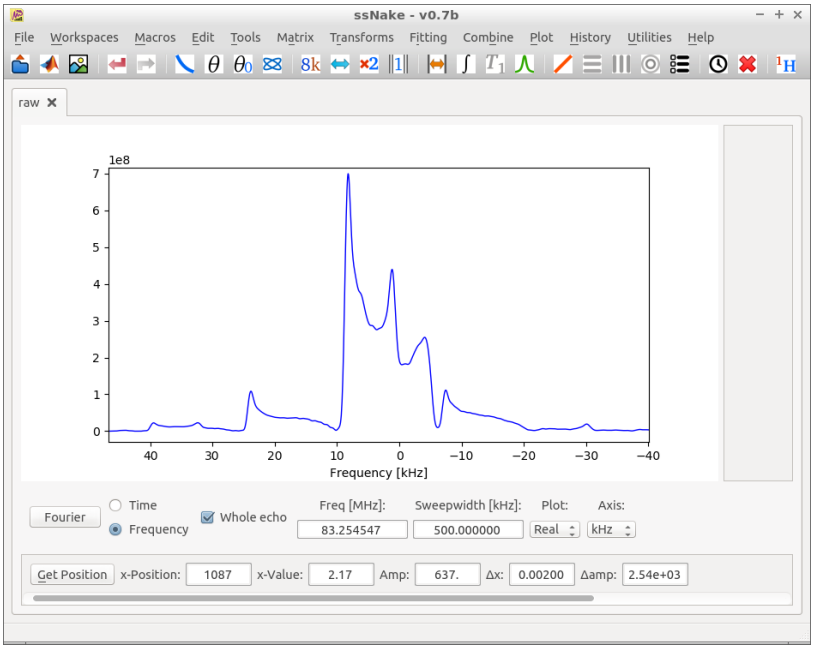
This gives:



We will now process the data using whole echo processing. More on this can be found in the Whole Echo processing tutorial. In short:

- Swap the echo using Tools ⟶ Swap echo at point 512
- Zero fill to 8192 points using Matrix ⟶ Sizing
- Fourier transform with the Fourier button in the bottom frame
- Phase a bit using Tools ⟶ Phasing (0th order: 0.220, 1st order: 68)
- 800 Hz Gaussian apodization (Tools ⟶ Apodize)

This shows (zoomed):

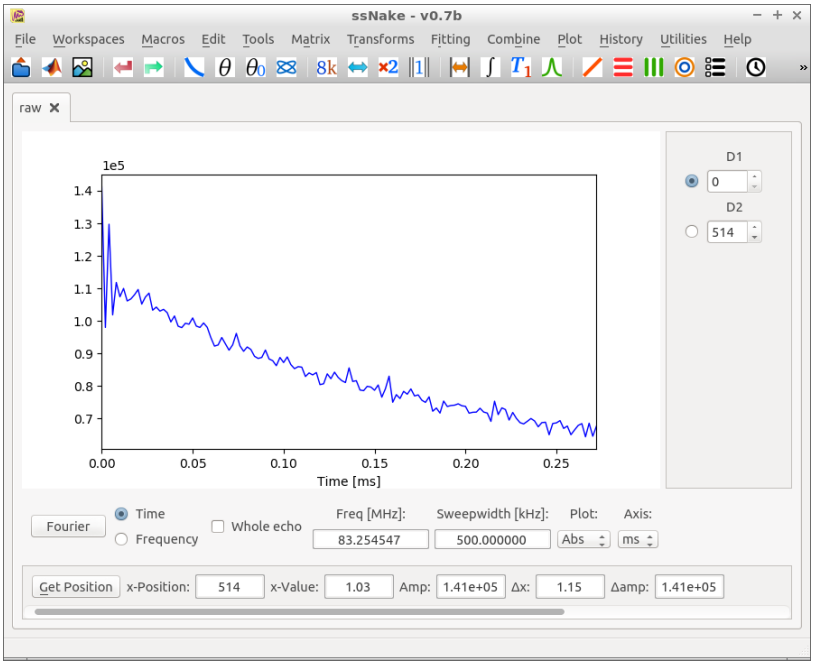


Which is the final spectrum using this processing method. It is saved as `EchoSumOption2.mat` in the data directory.

3.1.2 Option 2: T2 weighting

Summing all the echoes as has been done in Option 1 is fine in some cases, but can lead to signal-to-noise drawbacks. We are adding the echoes to gain a higher signal-to-noise when compared to a single echo experiment (and not a QCPMG). But what if our last echo has no signal anymore? Clearly, adding this echo only introduces noise into our sum of echoes. We want only to add echoes that give us a gain in signal-to-noise. The best way to do this seems to multiply all our 137 echoes by their maximum intensity: weak echoes with only noise have hardly any intensity, and are reduced, while intense echoes get a good boost. As the echo intensity is reducing due to T_2 effects, this scaling method is often called 'T_2 weighting'.

To perform a T_2 weighting, we must first get the T_2. We start with the data on which we have just done the splitting in 137 parts (before the Option 1 section processing). The echo tops are in this case located at data point 514. Fill in this number in the D2 box and press enter. This should give:

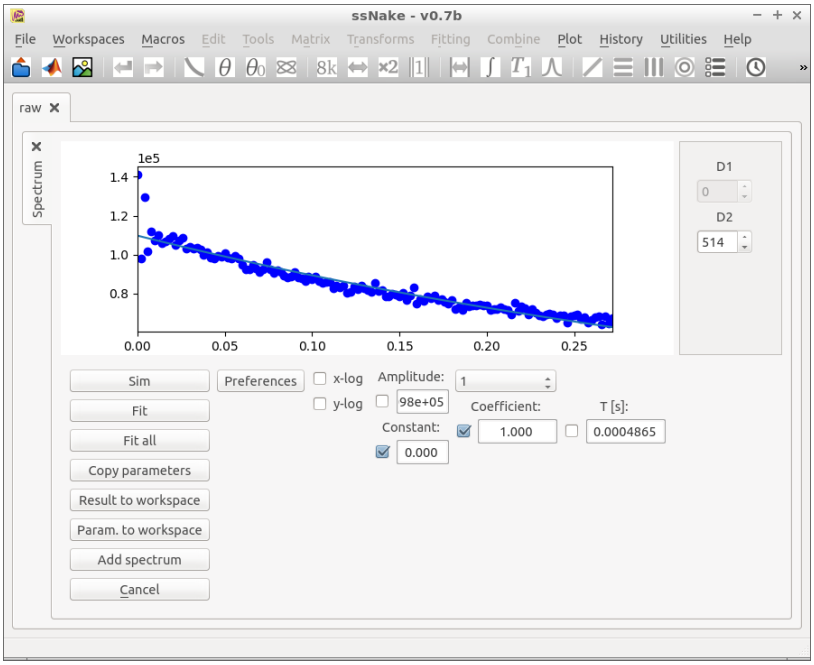


We now want to fit this line with a T_2 decay. DO note that the spectral width in this case was invented by ssNake. This means that our fitted T_2 will be in a random unit, but as long as we apply the weighting in the same unit, we should be fine[1]. Let's fit a $T2$:

- Open the relaxation fitting frame (Fitting $\longrightarrow$ Relaxation Curve)
- Set the 'Constant' at 0, and the 'Coefficient' at 1
- Set the 'T' variable at 0.001 as an initial guess
- Tick the boxes next to the 'Constant' and 'Coefficient' (this makes them fixed, so that they do not change during the fitting)
- Push 'Fit' (and a second time to get a better result)

This gives:

[1]Alternatively, we could define the spectral width as the inverse of the time between the echo tops.



with a T_2 of 0.0004865 second. Close the Window using the 'Cancel' button

We now wish to use this T_2 to scale our echo intensities. We will do this using Lorentzian apodization, which has the same function as a T_2 decay (exponential decay). The value that we will use is: $\mathrm{LB} = 1/(T_2 \cdot \pi) = 654$ Hz.

- Use Tools $\longrightarrow$ Apodize and apply 654 Hz Lorentzian apodization along D1

The data can now be further processed using the methods of Option 1 above. This T_2 weighted spectrum should have a better signal-to-noise that that of Option 1, although this particular data has no really low intensity echoes, so the difference will be small. The final spectrum is saved in the data directory as `EchoSumOption2.mat`.

3.2 Spikelet spectrum

A wholly different method for processing the data from a QCPMG experiment is the spikelet method. It features a direct Fourier transform of the FID, with no splitting and summing of the echoes. The spectrum it leads to has a series of spikes (i.e. spikelets) in the spectrum, with a distance of $1/T$, with T the time between two echoes. The advantage of the spikelet method is that all the signal is concentrated in the spikes, leading to a huge increase in signal-to-noise ratio. The disadvantage is that, while the tops of the spikes follow the intensity distribution of the 'regular' echo spectrum, the area betwene the spikes gives no information. If only a few spikelets are present, the shape of the quadrupolar line becomes obscured.

- Open the data in the `Raw` folder
- Zero fill to 524288 (Matrix $\longrightarrow$ Sizing)

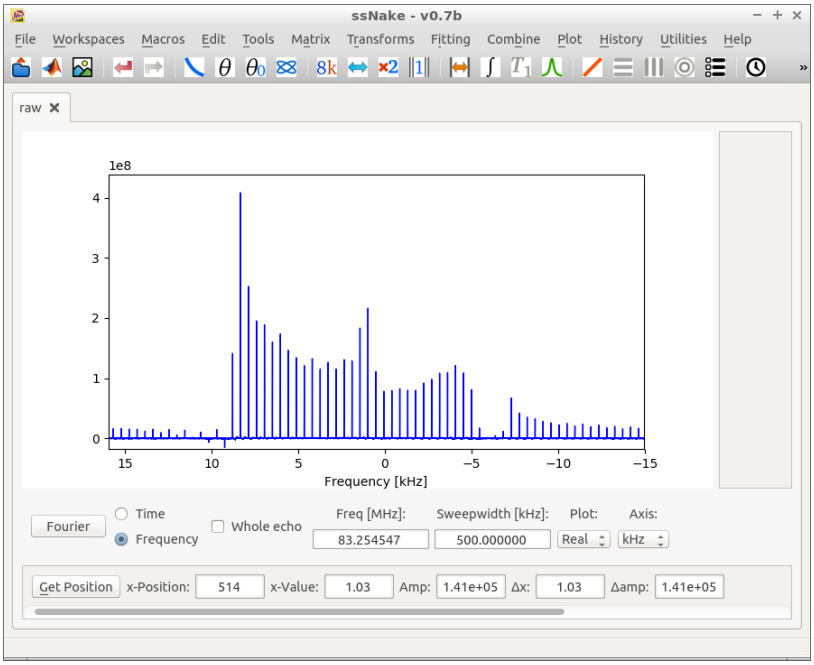
We now need to apply a first order phase correction, to make sure the FID starts at the first echo top. The position of this top is data point 514. To correct this we need $\theta = 360 \cdot n = 185040$ degrees first order phasing:

- Phase with 185040 first order phasing (Tools ⟶ Phasing)

And to get a good spectrum:

- Fourier transform (using the 'Fourier' button)
- Gaussian apodization of 4 Hz
- Phase (0th order: -104, 1st order: -618)

This gives the final spectrum:



which is saved as `spikelets.mat` in the data folder.

When we performed the Gaussian apodization, we also removed the tail of the first echo (which was shifted to the end of the FID by the large first order phase correction). This suppresses the formation of a regular echo line shape under our QCPMG spikelets at the cost of intensity. In this case, we have many intense echoes, and the added baseline of this tail does not help us. If we do wish to retain this, we must make sure that the apodization is performed *before* the large first order phase shift.