



ssNake: A cross-platform open-source NMR data processing and fitting application



S.G.J. van Meerten*, W.M.J. Franssen, A.P.M. Kentgens*

Radboud University, Institute for Molecules and Materials (IMM), Magnetic Resonance Research Center, Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands

ARTICLE INFO

Article history:

Received 23 January 2019

Revised 25 February 2019

Accepted 26 February 2019

Available online 1 March 2019

Keywords:

Processing software

Data analysis

Fitting

Solid-state NMR

Python

ABSTRACT

For solid-state NMR or for unconventional experiments only a very limited number of modern processing and simulation software packages are available. For this reason, we have developed ssNake, an NMR processing program which provides both interactive and script-based processing tools. ssNake is aimed at solid-state NMR experiments, but can also be used for liquid-state experiments. It can read various data formats, including those from all major spectrometer vendors. It has extensive fitting capabilities, which can be used for spectrum deconvolution. ssNake also provides the unique feature of being able to fit multiple spectra (or curves) simultaneously, where some or all of its parameters are shared. This method can be used, for example, to fit quadrupole spectra at various magnetic fields simultaneously. This allows the quadrupole and chemical shift parameters to be accurately determined. ssNake also provides a method of fitting using external simulation programs, such as SIMPSON. This makes fitting very versatile, as it brings together experimental data and simulation software.

© 2019 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A number of computer programs exist for processing NMR data. Some well-known examples are NMRPipe [1], NMRglue [2], MatNMR [3], DMfit [4], Mnova and TopSpin. Most NMR processing software can be divided into two groups: command-line programs, such as NMRPipe, that process large number of files via an input script, and GUI programs like Mnova, which process the data mostly automatically, and show the output in a graphical user interface. The command-line programs are characterized by their efficiency and flexibility, but have a steep learning curve, and often do not support interactive processing. The GUI programs are usually aimed at automatic processing, and become inconvenient when these automatic protocols are not suitable for the problem being addressed. In particular for solid state NMR spectra, where the presence of broad lines and asymmetric lineshapes complicates the use of automated processing methods. In cases where unconventional processing is required, automatic methods might not have been implemented, and custom processing is required. For these purposes, we have developed ssNake, a flexible software package for interactive and script-based processing of NMR data.

The ssNake software package provides a Graphical User Interface (GUI), where interactive processing of NMR data can be performed, and a scripting back-end for more advanced users. Within the GUI, processing macros can be recorded. These can be saved to a file which can easily be converted to a processing script. The GUI is designed to be intuitive and show clear feedback of the actions being performed. This interactive nature of the ssNake GUI is also ideal for educational purposes.

ssNake is written in the Python programming language (Python Software Foundation, <https://www.python.org/>): a cross-platform programming language which is high-level, yet powerful and which has an extensive standard library. The processing algorithms are written using the Numpy and Scipy scientific computing libraries and plotting in ssNake is performed using the Matplotlib library. The GUI is written using the Qt framework and the Python bindings provided by PyQt. ssNake is made available (<https://www.ru.nl/science/magneticresonance/software/ssnake/>) under the GNU General Public License version 3. As both ssNake and the underlying Python software is free, open-source, and cross-platform, it is straightforward to run it on most systems.

In this manuscript we provide a brief introduction to data processing using ssNake and some examples of its versatile fitting routines. For information on the processing routines available in ssNake the user is referred to the reference manual delivered with ssNake. Processing examples with step-by-step explanations are available in the ssNake tutorial. More advanced processing exam-

* Corresponding authors.

E-mail addresses: b.vanmeerten@science.ru.nl (S.G.J. van Meerten), a.kentgens@nmr.ru.nl (A.P.M. Kentgens).

ples are provided on the ssNake tutorials GitHub page (https://github.com/smeerten/ssnake_tutorials).

2. The interface

The main interface of ssNake is shown in Fig. 1. The top of the window shows the Menu and the Toolbar. All the options that are in the menus/submenus can be added to the Toolbar for quick use (the Toolbar shown in Fig. 1 is the default). Loading data in ssNake opens a new tab in the Tab Bar for each dataset. Within ssNake, dataset are called Workspaces. All workspaces are independent, and any action performed from the interface always acts on the active workspace. All elements of the interface below the Tab Bar are part of the workspace, and will change when switching workspaces.

On the right of the Main Plot is the Side Frame. Located in this Side Frame are the dimension selectors. The data shown in Fig. 1 has two dimensions: D1 and D2. The last dimension is the “direct” dimension of the NMR experiment. For each dimension there is a radio button and an input box. The radio button displays the active dimension, while the input box of dimension D1 shows the index of the trace from that dimension which is being displayed. In Fig. 1, D2 is the active dimension and slice number 146 of D1 is viewed.

Located below the Main Plot is the Bottom Frame. It has a button for the “Fourier” action (which can also be found in the menu, or added to the toolbar), some properties of the active dimension (e.g. frequency and spectral width), and some display settings of the plot (type: real/imaginary/absolute, and the units of the x-axis).

The bottommost part of the interface is the Position Frame. The ‘Get Position’ button can be used to obtain a cursor and grab some information from a specific point in the plot, printing the position and values associated with this point, as well as the distance from the previously selected point.

The view of the data can be adjusted using the mouse. When the cursor is over the Main Plot, the left mouse button can be used to create a zoombox to zoom in on certain regions of the plot. The

right mouse button can be used to “drag” the view of the data. The scroll wheel is used to zoom in along the y-axis, while keeping the x-axis fixed. If the right mouse button is held while scrolling, the view along the x-axis is scaled with the y-axis fixed. There is also an alternative scrolling, which will be applied when the Shift-key is held during scrolling. This is used in Array and Stack plots to vary the distance between lines, or in the Contour plot to change the lower bound of the contours. Double clicking the right mouse button restores the zoom to a full view of the data.

3. Processing

NMR data can be loaded from various data formats. Some of the supported data formats are: Varian/Agilent VnmrJ, Bruker Topspin, Chemagnetics Spinsight, JEOL Delta, Magritek, and SIMPSON. ssNake automatically recognizes the format and loads the data. After loading, a new Workspace (i.e. tab) is created which contains the loaded data.

ssNake has a large number of processing routines available. The routines range from basic such as zero filling, apodization, Fourier transformation, etc. to advanced such as Linear Prediction, reconstruction of nonlinear sampled data (NUS), reference deconvolution, and processing of hypercomplex data. For a more complete list of all the processing capabilities of ssNake, the reader is referred to the reference manual (<https://github.com/smeerten/ssnake>).

Processing actions performed on one workspace do not affect other workspaces as all workspaces are independent. It is, however, possible to add or concatenate data from another workspace to the active workspace (e.g. to subtract a background). ssNake supports NMR data with an arbitrary number of dimensions, as well as hypercomplex information along any of these dimensions. Arrayed experiments are treated as data with an additional dimension. For example, a T_1 experiment is treated as 2D data, with every time increment in a subsequent D1 slice.

Any processing action (e.g. phasing, Fourier transform) in ssNake is performed only along the “active” dimension. For example, if the data has two dimension, and D2 is displayed, a Fourier

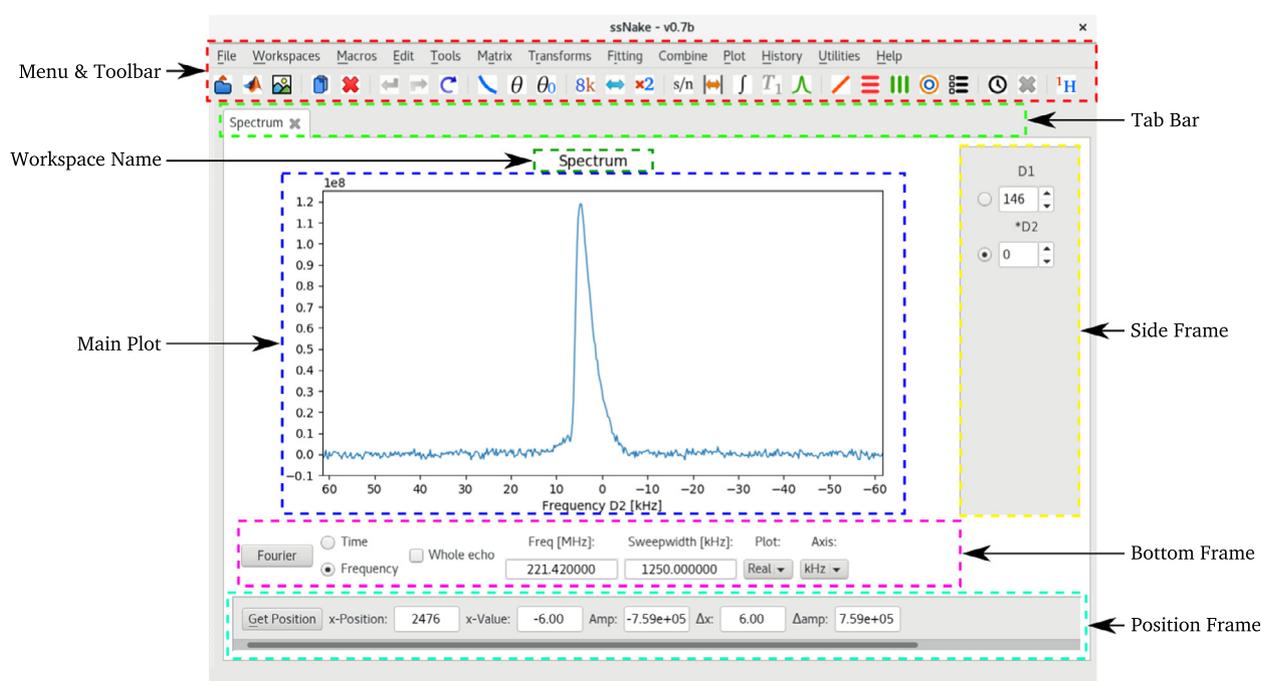


Fig. 1. A screenshot of the main interface of ssNake. The dashed boxes mark specific elements of the interface.

transform will lead to a transformation of all slices along D2. If the user intends to perform an action along D1, the view should be changed to this dimension first.

When performing an operation on the data, the effect of this operation is often “previewed” on the active view. During the previewing only the visible data is processed, which is generally fast and the plot displays live updates. After the user accepts the settings (i.e. “ok” is pressed), the operation is applied to the entire data set.

There are a number of settings which alter the behavior of certain operations. For example, the time/frequency setting which is displayed in the Bottom Frame. This setting shows whether the active dimension is an FID or a spectrum. When performing a Fourier transform a forward or inverse fft is applied based on this setting. Similarly, apodization is always performed on a data set in the time domain: when applied on a spectrum, ssNake first performs an inverse fft, applies the apodization, and then performs a forward fft. This way, the user does not need to consider the order in which many operations are performed. This way, operations like apodization can be previewed directly on the spectrum. The time/frequency setting is managed by ssNake, although it can also be changed manually if required. Performing a Fourier transform twice therefore has no net effect, as ssNake changes the time/frequency setting after the transformation.

3.1. Macro's

For the efficient processing of data, ssNake provides macros. In a macro, a series of processing actions can be saved, and executed on a workspace. A macro can be recorded in any workspace. After the recording is started, all actions performed on this workspace are added to this macro. After the macro is complete the recording can be stopped. The macro can then be applied to other workspaces or the macro can be saved to a file for later use. The saved macro file is in human readable format, so it can be opened and edited by any text editor. Each line in the file represents a single action, with the first word being the name of the function, and in parentheses the parameters relating to this function.

3.2. Script-based processing

Apart from using ssNake from its GUI, it is possible to run it in the Python interpreter or as a script. Using script-based processing can be convenient when a large number of data files need to be processed. Moreover, as regular python statements and flow controls (`for`, `if`) are available, more complex forms of data processing can be performed. For example, a script can be constructed which provides feedback for in progress NMR experiments. The NMR spectrometer records spectra continuously, which are then processed/analyzed using a ssNake script. The script can then use regular Python commands to control the flow system (switch valves, control pumps, etc.). The functions that are available in a script are identical to those used in a macro file, which means a macro can be created using the GUI which can then easily be converted to a script.

The scripting is implemented in an Object Oriented way, which makes it very flexible. NMR data can be loaded using the `autoLoad` function, which takes a filename as input and returns a spectrum object. All processing routines that can be applied from the GUI are class methods of this object and can thus be applied by calling them with the appropriate arguments.

Below an example is given of data which is loaded from a file named `fid` and is subsequently apodized with 10.0 Hz lorentzian broadening and then Fourier transformed. Finally, the data is stored in Matlab format in a file named `output.mat`.

```
data = autoLoad("fid")
data.apodize(lor = 10.0, axis = 0)
data.complexFourier(axis = 0)
saveMatlabFile("output.mat", data)
```

It is also possible to perform simple arithmetic using these data objects. The following example shows how two data files are loaded, the first is multiplied by 12.34, and the second is subtracted from the result.

```
data1 = autoLoad("spectrum1")
data2 = autoLoad("spectrum2")
result = 12.34 * data1 - data2
saveMatlabFile("output.mat", result)
```

This is just a very simple example of the use of this type of command-line processing. A complete demonstration of headless processing using ssNake is beyond the scope of this article.

3.3. Monitoring

From within the ssNake GUI, it is possible to monitor files. If a change to a monitored file is detected, the data will be reloaded, and a user supplied set of macros is performed on this data. This allows for continual updates of experimental data during its recording. This can be particularly useful for multidimensional experiments, to monitor its progress. Naturally, vendor NMR software might also be able to do this, but especially in cases with more involved processing requirements, using ssNake might be of interest.

4. Fitting

Apart from processing, ssNake has a flexible set of fitting routines, mainly aimed at solid state NMR spectral analysis. The available routines are tabulated in Table 1. When entering a fitting routine, the display of the active workspace is changed to a fitting window.

An example of a fitting window is shown in Fig. 2. Each fitting routine has a number of general control buttons shown on the left. Next to that, there are a number of constants (e.g. spin quantum number). After that, the fitting parameters are displayed. Some parameters are shared by all “sites” (e.g. spinning speed). For the others, a dropdown menu controls the number of sites. For each of these sites, the parameters can be set individually. The checkbox next to each parameter can be used to exclude certain parameters from optimization (checked: the parameter is fixed, unchecked: the parameter will be optimized). In a similar way as in the main window, the displayed slice of the data can be controlled from the Side Frame. Each slice of the data has its own set of fitting parameters and can be fit independently from the others.

Table 1

Supported fitting routines. Shown are the minimum required number of dimension for each type of fit, and whether the fit involves powder averaging.

Method	# Dimensions	Powder
Relaxation curve	1	n.a.
Diffusion curve	1	n.a.
Lorentzian/Gaussian spectrum	1	n.a.
CSA spectrum	1	yes
Quadrupole spectrum	1	yes
Quadrupole + CSA spectrum	1	yes
Czjzek distribution spectrum	1	yes
MQMAS spectrum	2	yes
Czjzek distribution MQMAS spectrum	2	yes
Function fit	1	n.a.

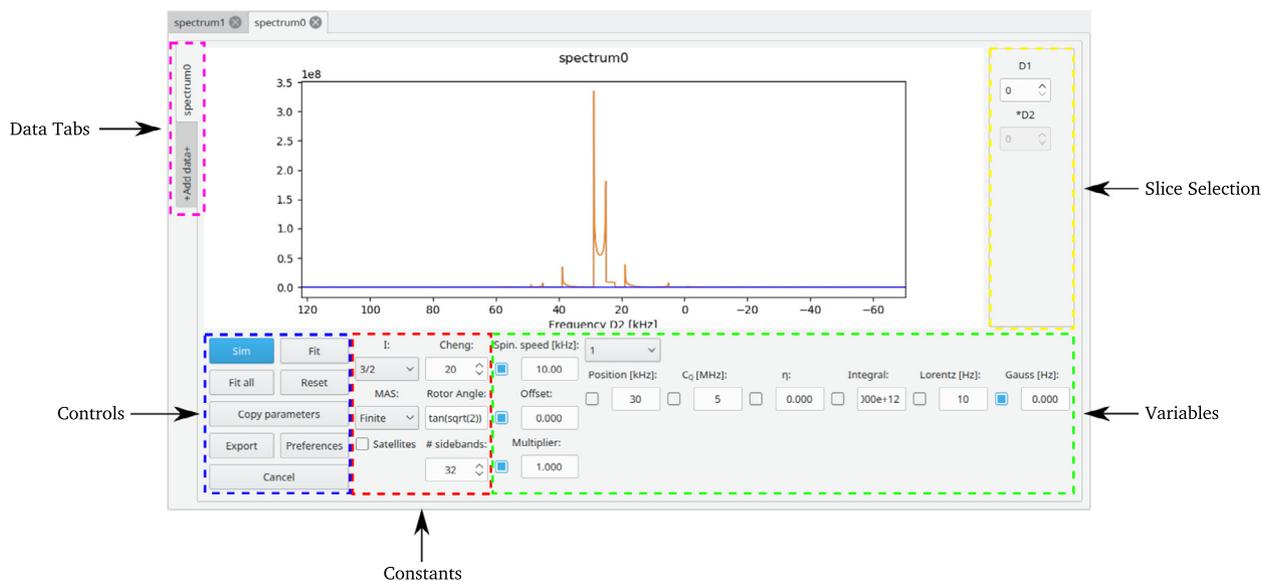


Fig. 2. A screenshot of the quadrupole fitting interface within ssNake. The dashed boxes mark specific elements of the interface.

The frame contains a ‘Sim’ button to simulate the spectrum of the displayed slice, a ‘Fit’ button to fit this slice, and a ‘Fit All’ button which will go through all slices sequentially, and fit them using their respective parameters.

The parameters of a fit can also be exported to a new workspace. In this case, the user can select from a menu which of the parameters should be exported. ssNake will then create a new workspace where the values of the parameters are used as the y-values. This method is particularly useful for performing different fits sequentially. For example, a (complicated) line shape can be fit for all slices of a spectrum. The amplitudes of these fits can then be exported to a new workspace. Then another fit, such as a curve fit (e.g. relaxation), can be performed on these results.

As ssNake is mainly aimed at solid state NMR, most of the fitting methods concern powder patterns caused by either CSA or quadrupolar broadening. In the case of quadrupolar broadening the satellite transitions can be included/excluded, and also Czjzek distributions can be taken into account. Moreover, simulations of 2D MQMAS spectra are supported (with or without Czjzek distributions). When simulating 1D CSA or quadrupolar spectra, the user can choose between three settings: a static spectrum, infinite MAS, and finite MAS. Of these, the static and infinite MAS methods are fast, while the finite MAS is more computationally intensive. It is also possible to vary the angle of the rotor with respect to the magnetic field, to take into account the effect of an incorrect magic angle setting or fit VAS spectra. When performing a finite MAS simulation, Carousel averaging is employed to simulate the spinning sidebands. [5–7] For this reason, the user must also supply the number of sidebands that need to be calculated (this should be higher than the observed number of sidebands for an accurate simulation).

The basis for the CSA and quadrupolar simulations in ssNake are the spherical tensor operators, which can be found in the book by Duer [8]. Using spherical tensors, rotations (e.g. MAS) can be implemented in a convenient way.

4.1. Simultaneous fitting multiple spectra

There are a number of computer programs which allow simulation and/or fitting of solid-state NMR spectra. Some examples are SIMPSON [9], SPINEVOLUTION [10] and Dmfit [4]. A particularly useful feature of ssNake is that it is able to fit multiple spectra

simultaneously while connecting some or all of their parameters. Connecting a parameter means that its value will depend linearly on the value of the parameter it is connected to. A scaling factor and an offset can be used to modify the relation between the parameters. The connected parameter will follow the equation: $ax + b$, where a is the scaling factor, b is the offset, and x is the value of the parameter connected to. The scaling and offset can be used to include additional information in the fit. This way, the data of multiple spectra can be fit using the least number of free parameters. For example, when the relative intensities of different resonances are known, the integrals can be connected using the scaling factor so only a single absolute intensity parameter is left. On the other hand if, for example, the relative shifts of resonances are known, it is possible to connect them using the offset set to the difference in shift. To set a parameter to be equal to another, the scaling and offset should be $a = 1$ and $b = 0$.

The RMS deviation being minimized depends on the relative amplitudes of the spectra/curves that are being fit. So it is possible to vary the “weight” that each spectrum/curve has on the results by changing the amplitudes.

Two examples of simultaneous fitting of multiple spectra are given below. In the first example, we demonstrate the fitting of two CSA spectra recorded at different spinning speeds. In another example, we show how quadrupole spectra acquired at various magnetic fields can be deconvoluted simultaneously to obtain the quadrupole parameters.

4.1.1. CSA spinning sidebands

A simple example of simultaneous fitting is on CSA spinning sideband spectra at different spinning speeds. From a fast spinning sample the isotropic chemical shift can be determined easily, but the accuracy of the determined anisotropic shift is reduced as the number of sidebands decrease with increasing spinning speed. With ssNake it is possible to fit spectra of multiple spinning speeds simultaneously so that all spectra can be explained with a single set of parameters.

The ^{207}Pb spectrum of PbCl_2 was recorded on a 14.1 T NMR magnet. The spectrum was recorded with the sample spinning at a frequency of 10 kHz and 15 kHz. The spectra were fit with the anisotropy parameters fixed to equal values ($a = 1$ and $b = 0$). In Fig. 3 the spectra and their fits are shown. When the isotropic chemical shifts were also fixed a satisfactory fit could not be

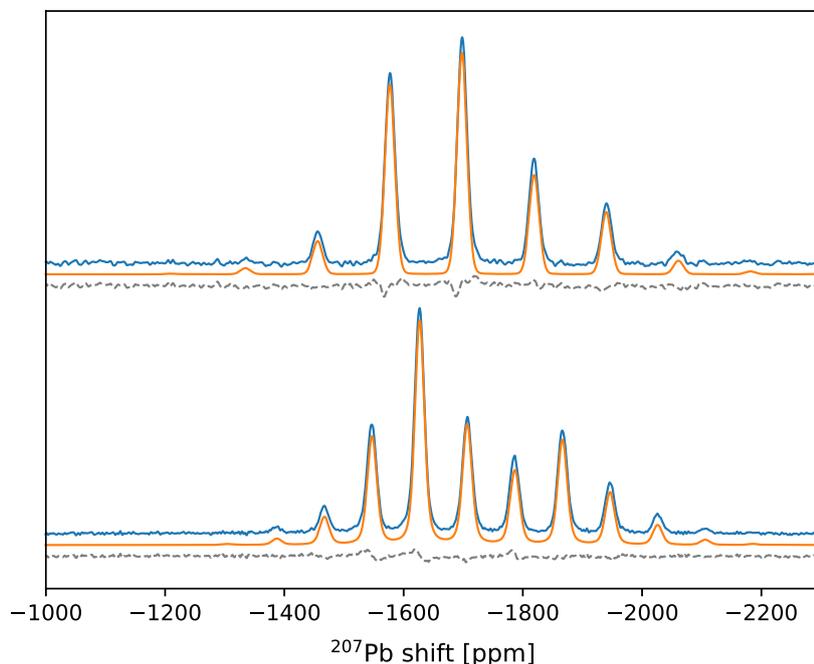


Fig. 3. ^{207}Pb spectra of PbCl_2 spinning at 15 kHz (top) and 10 kHz (bottom) at 14.1 T. The blue and orange lines show the experimental and simulated spectra, respectively. The fitting results are: $\delta_{\text{aniso}} = -316$ ppm, $\eta = 0.44$ and $\delta_{\text{iso}} = -1707$ or -1698 ppm (10 and 15 kHz). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

obtained. It turned out that the chemical shift of PbCl_2 is very sensitive to temperature and the increase in spinning speed raised the temperature of the sample slightly. This temperature difference resulted in a chemical shift difference of 9 ppm. The chemical shift anisotropy is hardly affected by the temperature change. The isotropic chemical shifts have to be fit independently due to slight difference in temperature.

4.1.2. Quadrupole coupling at multiple fields

Another example of simultaneous fitting is of quadrupole spectra acquired at different magnetic fields. In Fig. 4a the ^{87}Rb spectra of RbNO_3 acquired at 4 different magnetic fields (7.0 T, 9.4 T, 14.1 T and 20.0 T) are shown. In all experiments, a spinning speed of

25 kHz was used. Since the second order quadrupole frequencies scales inversely with magnetic field, the spectra look quite different. It is difficult to obtain the quadrupole parameters from the independent spectra as they show relatively little lineshape features. With ssNake, all 4 of these spectra could be fit simultaneously where the quadrupole and chemical shift parameters were connected to set them to equal values (with a scaling of $a = 1$ and offset of $b = 0$). The satellite transitions were excluded from the fit and since the spinning speed is much larger than the line-width the calculation for infinite spinning was used. The results are shown in Fig. 4a. The determined parameters describe all four spectra well. In the 20.0 T spectrum, shown in Fig. 5, a small feature can be seen at -25 ppm, which is caused by the satellite

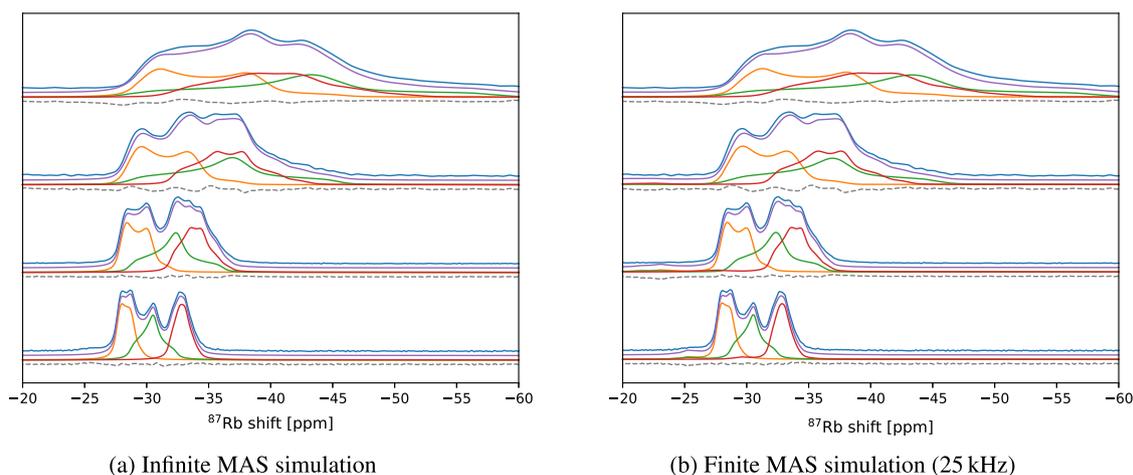


Fig. 4. ^{87}Rb single pulse spectra of RbNO_3 recorded at 7.0, 9.4, 14.1 and 20.0 T from top to bottom, using 25 kHz MAS. The experimental spectra are shown in blue. The orange, red, and green line are the simulated spectra of the three different sites. The fit results were: $\delta_{\text{iso}} = -27.41$, -28.71 and -31.82 ppm, $C_Q = 1.687$, 1.984 and 1.711 MHz and $\eta = 0.17$, 1.0 and 0.58 for the infinite MAS case. For the finite MAS case: $\delta_{\text{iso}} = -27.41$, -28.73 and -31.82 ppm, $C_Q = 1.688$, 1.987 and 1.712 MHz and $\eta = 0.20$, 1.0 and 0.59 . The integral values of the three sites were kept equal in each individual spectrum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

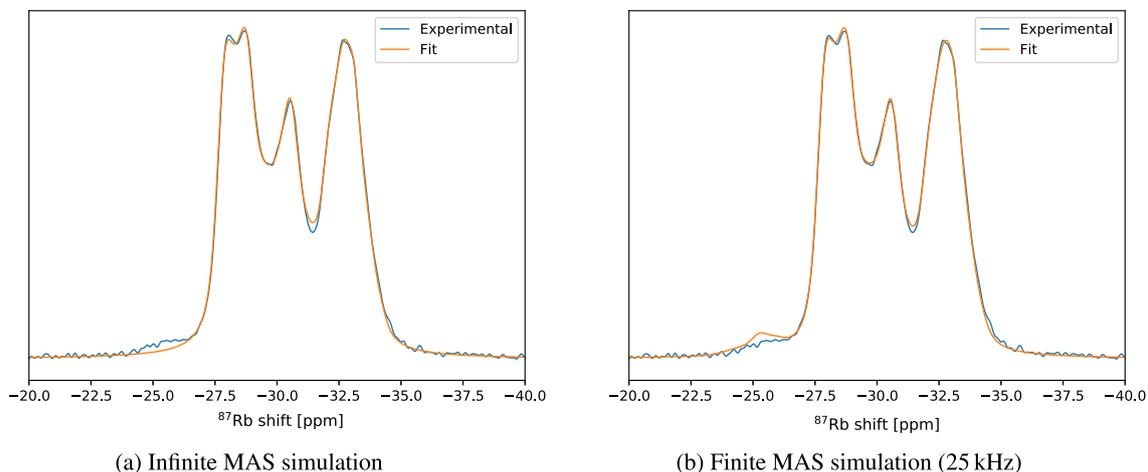


Fig. 5. ^{87}Rb single pulse spectra of RbNO_3 recorded at 20.0 T, using 25 kHz MAS. The spectra are the same as the bottom trace of Fig. 4.

transitions. This can be verified by performing the fit again with the satellite transitions included. Now a finite spinning simulation has to be performed as the spinning speed is less than the line-width of the satellite transitions. As can be seen in Figs. 4b and 5b, the feature is now indeed present in the calculated spectrum. The intensity of the calculated satellite signal disagrees with the experimental spectrum, which is due to excitation efficiency. The satellites are too broad to be excited efficiently by the RF pulses, whereas the calculated spectrum assumes ideal excitation. To improve the fit of the satellites it would be necessary to include the excitation efficiencies in the calculations. This is possible by using the external program fitting routine. It would increase the complexity and computation time while, in this particular case, the improvement in the parameters of the fit is insignificant.

4.2. Fitting quadrupole spectra using a (extended) Czjzek model

The CSA and quadrupole lineshapes generated using powder averaging properly describe the spectra of well-ordered systems. But for disordered systems, such as glassy materials, this method fails. A commonly used model for quadrupole spectra of disordered systems is the Czjzek model [11]. It assumes that the chemical surrounding of the nuclei is symmetric such that the electric field gradient is zero in the absence of disorder. The effect of disorder is introduced by assuming that the EFG tensor components are normally distributed. In terms of C_Q and η the probability density function is given by

$$f(C_Q, \eta) = \frac{C_Q^{d-1} \eta}{\sqrt{2\pi}\sigma^d} \left(1 - \frac{\eta^2}{9}\right) \exp\left\{-\frac{C_Q^2}{2\sigma^2} \left(1 + \frac{\eta^2}{3}\right)\right\}, \quad (1)$$

where σ is the standard deviation of the multivariate Gaussian distribution and d corresponds to the number of independent tensor components [12]. This model has later been extended to the case where the chemical surrounding is asymmetric in the absence disorder [13]. The extended Czjzek probability density function is given by

$$f(C_Q, \eta) = \frac{C_Q^{d-1} \eta}{\sigma^d} \left(1 - \frac{\eta^2}{9}\right) \iiint d\alpha \sin(\beta) d\beta d\gamma \quad (2)$$

$$\times \exp\left\{-\frac{1}{2\sigma^2} \left[C_{Q,0}^2 \left(1 + \frac{\eta_0^2}{3}\right) + C_Q^2 \left(1 + \frac{\eta^2}{3}\right)\right]\right\}, \quad (3)$$

$$-\frac{2}{\sqrt{3}} C_Q C_{Q,0} \left(\sqrt{3} a_{11} + \eta_0 a_{15} + \eta \left(a_{51} + \frac{\eta_0 a_{55}}{\sqrt{3}}\right)\right)\right\}, \quad (4)$$

where $C_{Q,0}$ and η_0 are the quadrupole parameters of the local order contribution. The angular dependence of the efg tensor components a_{ij} are given in the paper by Le Caër and Brand [13,14]. This extended Czjzek model has also been included in ssNake.

To fit (extended) Czjzek spectra, ssNake uses a library of spectra based on a rectangular grid of C_Q and η values. This method is similar to that used in the EASY-GOING deconvolution software [15,16]. There are two ways to obtain such a library of spectra. One option is to let ssNake create the library. The library will then consist of spectra simulated using the same algorithm as used for ordered quadrupole spectra. This implies an ideal excitation of the complete lineshape and an equal excitation efficiency for each spectrum in the library. Another option is to load the library from data files, so the spectra can be created using any NMR simulation software. A convenient choice is SIMPSON [9] with which the complete pulse sequence can be simulated. This way, the experimental parameters such as pulse lengths and amplitudes are taken into account. These experimental parameters could be particularly important for quantification, as the excitation efficiency may be different per resonance.

4.2.1. Fitting a ^{35}Cl spectrum with the extended Czjzek model

To demonstrate the extended Czjzek fitting, a ^{35}Cl spectrum of ball-milled MgCl_2 with a diisobutyl phthalate adduct was used. [17] The spectrum was acquired on a 20 T magnet at a spinning speed of 15.625 kHz. This spectrum was fit with both the simple and extended Czjzek routine of ssNake. For both fits, a library was created using ssNake. The library was created on a grid of 2000 points. 100 points along C_Q (0.0–6.0 MHz) and 20 points along η (0.0–1.0). A finite MAS simulation was used for the generation of the library. The result of the fits are shown in Fig. 6.

It is clear that the simple Czjzek is not sufficient in describing the lineshape properly. In particular, the spinning sidebands show a large mismatch between the fit and the experimental spectrum. The resulting standard deviation parameter of the quadrupole distribution is $\sigma = 2.1$ MHz. The extended Czjzek model gives a better fit shown in Fig. 6b. The quadrupole distribution parameters of the extended Czjzek fit are $\sigma = 0.62$ MHz, $C_{Q,0} = 4.2$ MHz and $\eta_0 = 0.0$. This shows that the disorder is overestimated when the local order parameter $C_{Q,0}$ is not included. In the Czjzek fitting window of ssNake one can conveniently enable/disable the use of the extended Czjzek model. The fit shown here was performed using a library of spectra generated by ssNake. The simulations by ssNake do not take into account the excitation efficiencies of the

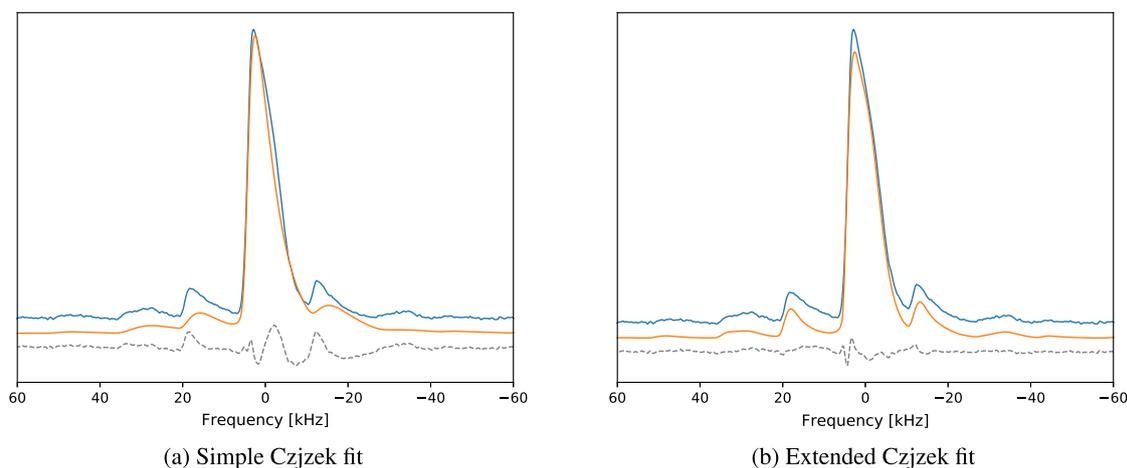


Fig. 6. ^{35}Cl spectrum of MgCl_2 with diisobutyl phthalate adduct fit with the simple and the extended Czjzek model.

RF pulses. To include these excitation efficiencies in the fit, a library simulated using an external program (such as SIMPSON) is required, as was discussed in Section 4.2.

4.3. MQMAS

MQMAS is a well established method to obtain isotropic spectra of quadrupole nuclei which are broadened by the second order quadrupole coupling. It is a 2D pulse sequence where in the indirect dimension the multiple quantum (MQ) evolution is acquired. Under fast spinning, the anisotropy of the second order quadrupole coupling during the MQ evolution is identical (besides a scaling factor) to that during single quantum (SQ) evolution. The scaling of the quadrupole anisotropy is, however, not equal to the scaling of the chemical shift. This means isotropic spectra can be obtained from the MQMAS by a shearing transform. This will remove the second order quadrupole anisotropy without removing the chemical shift information.

MQMAS is a convenient method to separate quadrupole line-shapes which overlap in the 1D spectrum. The parameters of MQMAS fitting in ssNake are fairly similar to those of the quadrupole fitting. Added to the interface are the settings for the Multiple Quantum coherence, the shearing factor and the scaling factor of the indirect dimension. In the list of parameters there are also independent values for the linebroadening (Lorentz, Gaussian) in the direct and indirect dimension.

A typical example is RbNO_3 with its three different ^{87}Rb lines. [18] While in the 1D MAS spectrum these lines overlap, they are well separated in the MQMAS spectrum. For this example we have recorded the z -filtered MQMAS spectra of RbNO_3 at 7.0 T and 14.1 T. Just as with the 1D spectra, the MQMAS spectra at different magnetic fields can be fit simultaneously using connected quadrupole and chemical shift parameters. The results of this fit are shown in Fig. 7. The data was sheared with a factor 7/9 to obtain isotropic information along the indirect dimension. The spectral width of the 3Q dimension was scaled with a factor 9/34 so that the chemical shift is equal in both dimensions.

The MQMAS experiment is particularly useful when applied to disordered systems. This way, the distributions in the quadrupole and chemical shift parameters can be studied. ssNake has a fitting routine for MQMAS spectra using a (extended) Czjzek model as well. It uses a library method similar to that of the 1D Czjzek fitting routine. In this case the library can also be simulated by ssNake, which then results in spectra where an ideal RF excitation of the entire lineshape is assumed. The library can also be loaded from external files as was discussed in Section 4.2. Due to space

limitations the library does not consist of complete MQMAS spectra, but of only 1D arrays containing the integrals over the indirect dimension. This is equivalent to the spectrum obtained in the first increment of the 2D experiment. The algorithm uses the fact that in the infinite spinning case, for a specific C_Q and η , all intensity is located on a straight line in the spectrum. So the construction of the MQMAS spectrum from the library is still computationally efficient, while the generation time and memory requirements for the library are greatly reduced. It does, however, limit the simulation to infinite spinning. The Czjzek MQMAS fitting also has an additional chemical shift distribution parameter σ_{CS} . This is equivalent to a Gaussian apodization along the chemical shift axis. The distribution in chemical shift parameters and the Czjzek distribution are thus assumed to be independent.

Just as with the 1D Czjzek fitting routine the library can also be loaded from data files. The library should consist of 1D spectra with the spectral width comparable to the direct dimension of the MQMAS spectrum. These can be generated by simulating the first increment of the MQMAS experiment for each pair of C_Q and η . The user should keep in mind that spinning sidebands in the 1D spectrum will not be placed correctly in the final MQMAS spectrum due to the way the 2D is created from the library spectra.

4.4. Function fit

In ssNake it is also possible to fit curves/spectra with “arbitrary” functions. The user can input any equation in Python/Numpy format, which is used for fitting. The variable x contains a Numpy array with the x -axis datapoints. And the @ character is used to denote fitting parameters in the equation. The variables between two @ characters are used as the parameter names (e.g. ‘@shift@’). All unique parameter names are displayed in the interface of ssNake and can be used for simulation and fitting.

4.4.1. REDOR

An example of a function fit, is the analysis of a solid state REDOR curve. When properly processed, this experiment leads to a intensity difference due to the dipolar coupling as a function of the dephasing time. In the case of a two spin system, these curves are easily simulated using the equation [19]

$$\Delta S/S = 1 - \frac{\sqrt{2}\pi}{4} J_{1/4}(\sqrt{2}D\tau) J_{-1/4}(\sqrt{2}D\tau), \quad (5)$$

where $J_{1/4}$ and $J_{-1/4}$ are Bessel functions of the first kind, τ is the dipolar evolution time, and D the dipolar coupling constant. In the

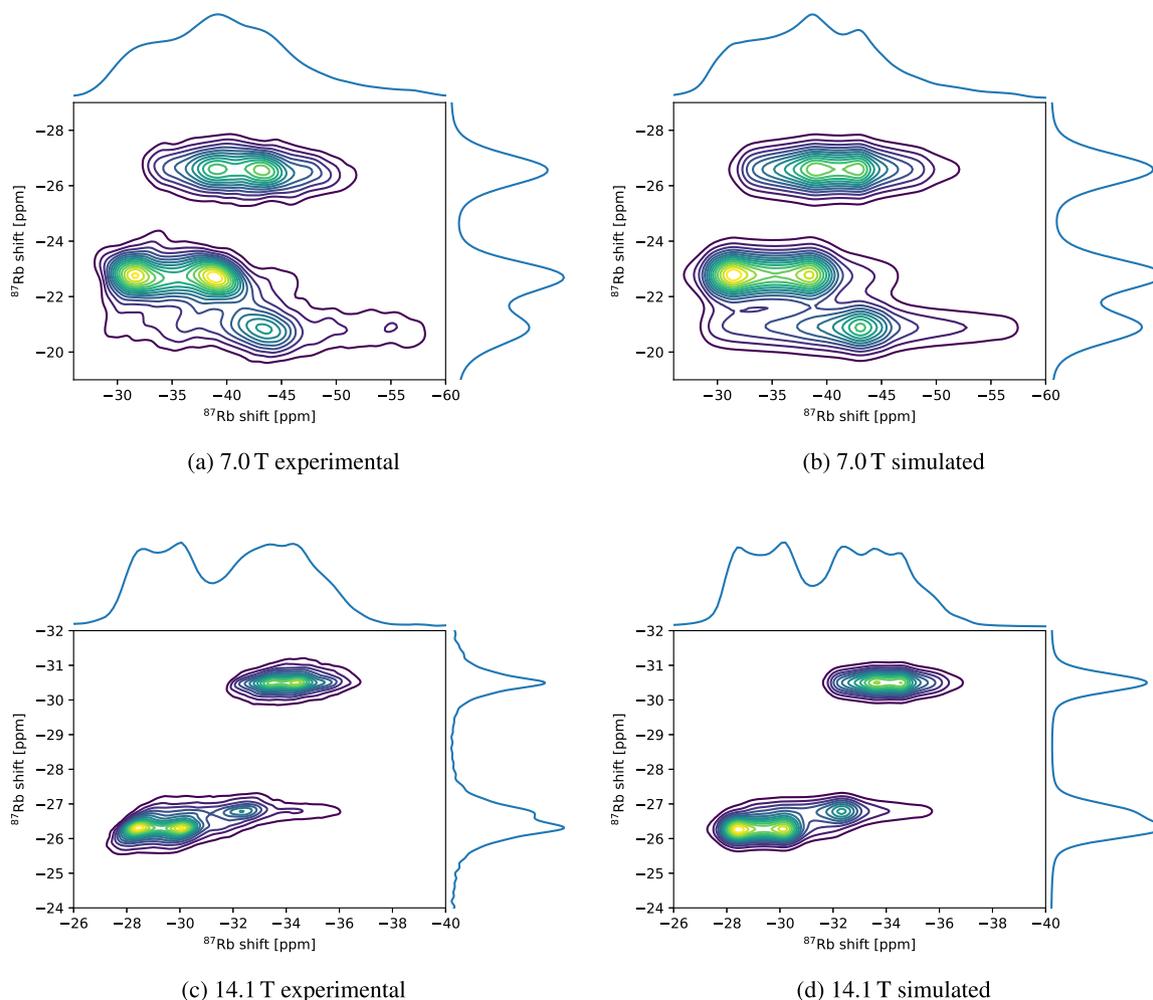


Fig. 7. ^{87}Rb zf-3QMAS spectra of RbNO_3 at two different magnetic fields. Showing both the experimental spectrum (a and c) as well as the simulated result from a fit (b and d). Both fits were performed simultaneously, with coupled quadrupolar and chemical shift parameters. The fit results were: $\delta_{\text{iso}} = -27.40, -28.75$ and -31.80 ppm, $C_Q = 1.72, 1.96$ and 1.76 MHz and $\eta = 0.20, 1.0$ and 0.57 .

Function Fit method of ssNake this formula can be implemented using the following function:

```
1 - sqrt(2) * pi/4.0 * jv(0.25, sqrt(2) * @D@ * x) * jv(-
0.25, sqrt(2) * @D@ * x)
```

with `jv` a Bessel function of the first kind, provided by the `scipy.special` library (which is integrated in `ssNake`). Fig. 8 shows a ^{13}C - ^{15}N REDOR curve of ^{13}C , ^{15}N labelled glycine, together with a fit using this function. This fit results in a dipolar coupling 808.5 Hz, which corresponds to a C–N distance of 1.56 Å (literature shows 1.47 Å [20] determined by XRD and 1.51 Å [21] determined by NMR).

4.4.2. Reaction kinetics

The Function Fit is also an ideal method for analyzing reaction kinetics. For example, NMR spectra can be recorded of a reaction mixture over time. The peaks corresponding to each reactant/product can be integrated or deconvoluted to determine the concentration. This results in a concentration curve for each molecule as a function of reaction time. The Function Fit can then be used to fit these curves simultaneously with a specific reaction equation for each molecule. Even though the equations are different, they share rate constants, which should be equal in all fits. This can be achieved by connecting the parameters. The Function Fit also provides Ordinary Differential Equation (ODE) solvers from the

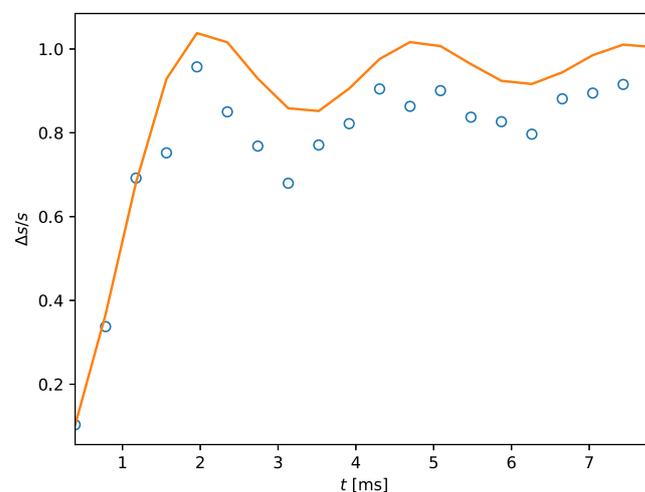


Fig. 8. REDOR curve of glycine C_α , ^{13}C - ^{15}N labelled, and ^{13}C - ^{15}N REDOR, proton decoupling, 5 kHz MAS, 400 MHz. The experimental curve is shown in blue and the simulation in orange. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

`scipy.integrate` library. This makes it possible to fit the concentrations of reactants/products during reactions, as they can be described by a series of coupled differential equations.

In this example we studied the reaction of methylamine (MA) with dimethylformamide (DMF) forming *N*-methylformamide (NMF) and dimethylamine (DMA):



The rate equations for this reaction form a set of four coupled differential equations given by:

$$\frac{d[\text{MA}]}{dt} = \frac{d[\text{DMF}]}{dt} = -k[\text{MA}][\text{DMF}] \quad (7)$$

$$\frac{d[\text{NMF}]}{dt} = \frac{d[\text{DMA}]}{dt} = k[\text{MA}][\text{DMF}] \quad (8)$$

where k is the reaction constant.

This reaction was followed using NMR for over 60 h. The integrals of the NMR resonances were used to determine the concentrations of MA and NMF during the reaction. In Fig. 9 these concentrations are shown. The MA curve was fit using the Function Fit with the following Python expression:

```
odeint(lambda C,t,k: [-C[0]*C[1]*k, -C[0]*C[1]*k, C[0]*C[1]*k, [0]*C[1]*k], [@MA@, @DMF@, @NMF@], x, (@k@,))[0]
```

This uses the ODE solver `odeint` provided by Scipy. For more information, the reader is referred to the documentation of Scipy

(<https://www.scipy.org/>). A `lambda` operator is used to set up a function which generates the reaction rates for MA, DMF and NMF. The `lambda` operator is a very useful concept in Python. A more detailed description of the `lambda` operator can be found in the Python documentation (<https://www.python.org/>). The NMF curve was fit simultaneously with the same expression but where the last indexing operation was replaced by `[:,2]`. The concentration of DMA was not taken into account as it does not influence the reaction rates. The parameters `@MA@`, `@DMF@` and `@NMF@` are the starting concentrations of MA, DMF and NMF and `@k@` is the reaction constant. The starting concentration of DMF was fixed to 9.2 M. The MA and NMF starting concentrations were optimized in the fit, just as the reaction constant k . This resulted in $[\text{MA}]_0 = 1.36$ M, $[\text{NMF}]_0 = 0.037$ M and $k = 2.3 \times 10^{-5} \text{ M}^{-1} \text{ s}^{-1}$. The curves of the fit are shown as the orange lines in Fig. 9.

This is just a simple example of how the Function Fit can be employed together with an ODE solver to obtain rate constants from reaction curves. This method can be used to investigate far more complex reactions involving more reaction steps or intermediates. All that is required is adjusting the Python expression to the correct rate equations, and adding more concentration curves to the simultaneous fit. This makes the Function Fit a very flexible and powerful method to interactively fit complicated curves.

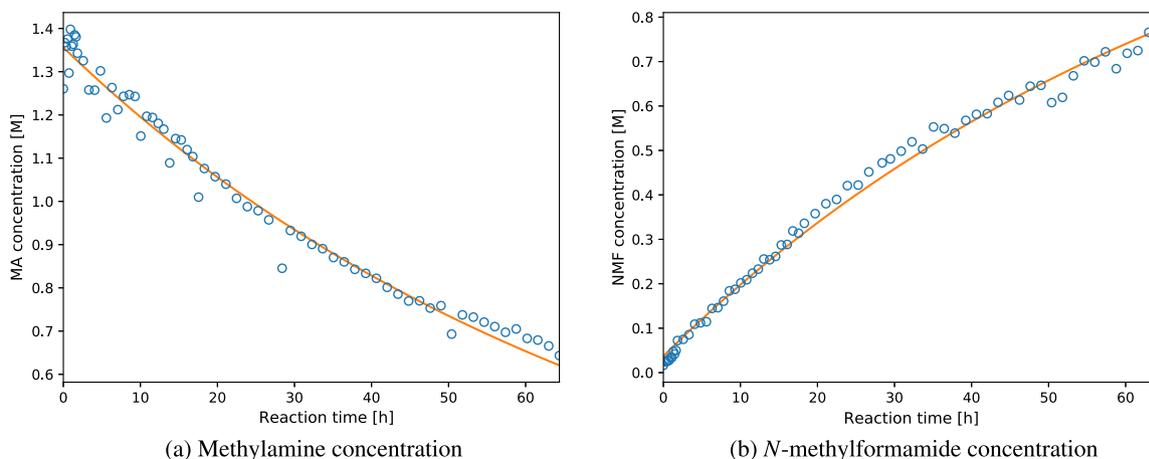


Fig. 9. Concentrations of MA and NMF as a function of reaction time as determined from the intensity of the NMR resonances. The orange lines show the fit obtained using the Function Fit of `ssNake`. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

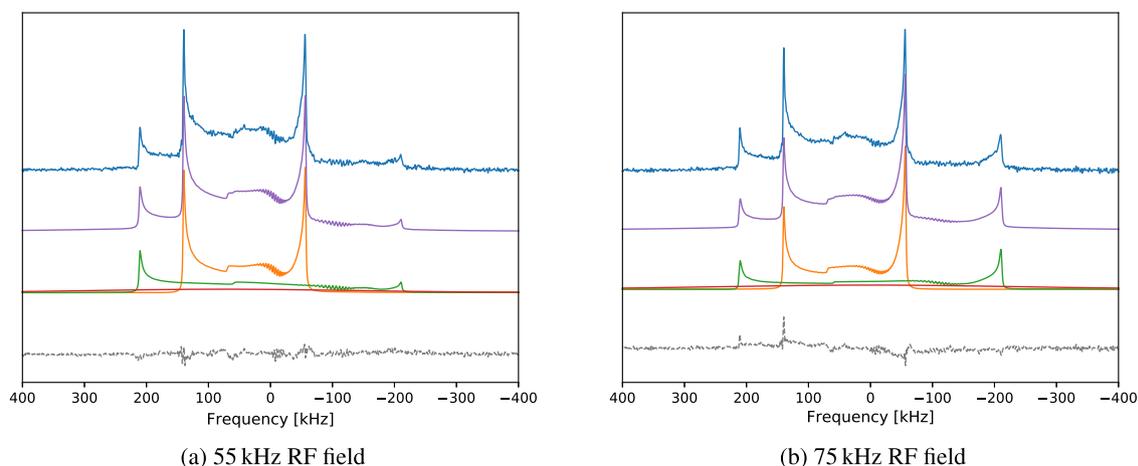
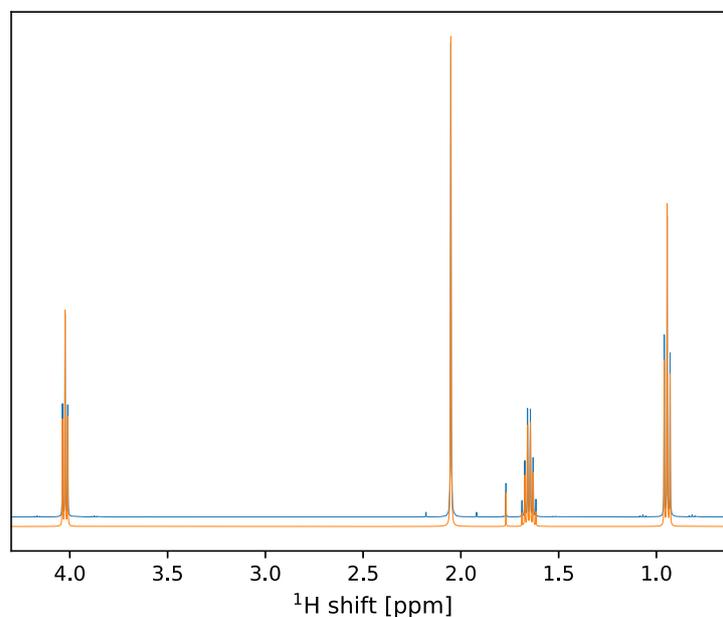


Fig. 10. ^{139}La spectra of LaSO_4 acquired using 55 kHz (a) and 75 kHz (b) RF field strength for the excitation pulses. The fit results were: $C_Q = 35.6$ and 52.1 MHz, and site 2 was at a 124 ppm lower shift than site 1.

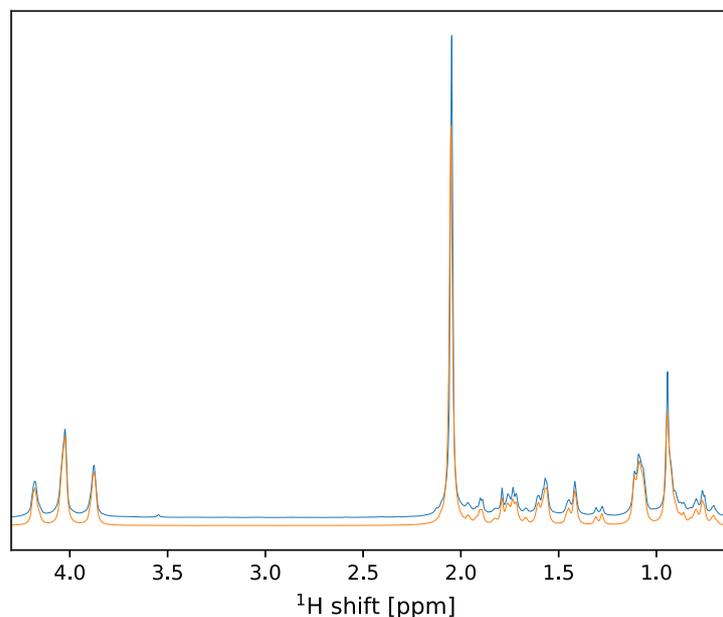
4.5. External fit

ssNake also offers the option to fit spectra using external simulation software. This works for any program which can be run from the command line with a file/script as input and which generates a file as output, which can be loaded by ssNake. The external fit works in a similar way to the function fit. The user selects a script to be used with the external fit. ssNake then scans this file for any occurrences of the @ character, which is used to mark a parameter just as in the function fit method.

When an external spectrum is simulated, ssNake first creates a temporary directory. In this directory it puts a copy of the input file/script with the @ parameters replaced by their corresponding values. It then runs the command of the external program with the path to the file as the only input argument. ssNake then waits for the external program to complete and then checks the directory for any new file. It will then attempt to load the data from that file and Fourier transform the data if necessary. An interpolation is performed to match the data points with those of the experimental data. When the simulation (or fit) is completed, the temporary directory is removed. The rest of the fitting routine is similar to the others.



(a) 11.7 T spectrum



(b) 1.0 T spectrum

Fig. 11. ^1H spectra of propyl acetate recorded at 11.7 T (a) and 1.0 T (b). The spectra are fit simultaneously using the Jellyfish as an external program. The experimental data is shown in blue and the simulated data is shown in orange. The fit results for $\text{CH}_3\text{—C(=O)—O—CH}_2\text{—CH}_2\text{—CH}_3$ (a,b,c,d) were: $\delta_{\text{iso}} = 2.051, 4.024, 1.652$ and 0.945 ppm. $J_{\text{bc}} = 6.77$ Hz, $J_{\text{cd}} = 7.45$ Hz and $J_{\text{bd}} = -0.14$ Hz. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.5.1. Quadrupole excitation efficiency

An example of an external program for fitting is SIMPSON, which is a general simulation program for solid-state NMR spectroscopy. SIMPSON accepts an input file as input argument. In this input file, a spin system and pulse sequence can be scripted. This way, any solid-state experiment can be simulated and thus used for fitting. As shown before for RbNO_3 , the excitation efficiency influences the spectra of quadrupole nuclei. An example of a spectrum in which the excitation efficiency should be taken into account is the ^{139}La spectrum of LaSO_4 which is shown in Fig. 10. The spectrum contains two quadrupolar lineshapes. The spectrum was recorded using pulses with a RF field strength of 55 kHz and 75 kHz. It can be seen that the lineshapes are significantly different in those two experiments. In particular the rightmost side of the line with the largest quadrupole coupling is distorted in the low RF-field experiment. This not only complicates the determination of the quadrupole parameters, but for resonances with large differences in quadrupole coupling the quantification also becomes difficult. To include the RF excitation during the pulse sequence, the calculation is performed using SIMPSON. The input file for SIMPSON is given in the [supplementary material](#).

The satellite transitions are not explicitly taken into account in the simulation due to the long computation time. Instead, the satellite transitions have been approximated by a single Gaussian represented by the red line in Fig. 10. The calculated spectra using SIMPSON show a good match with the experimental data for both the low and high RF-field experiments. It is also interesting to note that the oscillations in the two quadrupolar lineshapes are not an artifact of the simulation, but are present in the experimental spectra as well.

4.5.2. Strong coupling pattern

While ssNake is a solid-state NMR oriented program, its routines can also be used to address liquid-state NMR problems. An example of this is the simultaneous fitting of high and low-field proton spectra of the same compound/mixture, where strong scalar couplings are active. In this example, we employed an in-house developed J-coupling simulation engine called Jellyfish. Jellyfish has a GUI, in which spectra can be simulated in an interactive way. The engine is, however, also accessible from a script, and can thus be connected to ssNake's external fitting routine.

Fig. 11 shows a simultaneous fit of ^1H spectra of propyl acetate recorded at high (11.7 T) and low (1.0 T) magnetic field. Using ssNake's ability to couple parameters in a fit, the chemical shifts and J-couplings were linked, allowing a simultaneous fit of these spectra. A small water signal was also present, and has been included in the fit. Due to a temperature difference between the two spectra, this water signal appears at a slightly different chemical shift in both spectra, and this parameter has therefore not been linked during the fit. Fitting these spectra simultaneously, allows for both a proper identification of chemical shifts (which are most accurately determined in the high-field spectrum), and a good description of the J-couplings (which have dramatic effects at low magnetic field). Using ssNake's fitting routines, the complicated linking of parameters in this case can be implemented in a straightforward way, even though the simulation engine has no fitting capabilities itself.

5. Conclusions

ssNake is a versatile NMR processing program, which offers both a graphical interface and a command-line interface. The GUI

is designed to be intuitive and user-friendly. On the other hand, ssNake also offers scripting, which is ideal for efficient processing of a large number of files. Or it can be used to create custom feedback control systems, which respond to the NMR data in real-time. ssNake also provides a number of different fitting routines. A unique feature of ssNake is that it allows fitting of multiple spectra simultaneously, where some, or all, of its parameters can be connected. This is relevant, for example, for determining quadrupole parameters from spectra recorded at multiple magnetic fields. ssNake also provides an "External Fit", which can use other NMR simulation software to generate spectra for fitting. This makes fitting using ssNake quite powerful and versatile, in particular for solid-state experiments where the details of the pulse sequence could greatly influence the interpretation of the data.

Acknowledgements

This research received funding from The Netherlands Organization for Scientific Research (NWO) in the framework of Technology Area COAST (053.21.115). The authors would like to thank Gerrit Janssen, Hans Janssen and Ruud Aspers for technical support. Moreover, we thank Vincent Breukels for his additions to ssNake and our beta-testers for their input: Koen Tijssen, Ole Brauckmann, Merijn Blaakmeer, Fleur van Zelst, Julija Romanuka and many others.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <https://doi.org/10.1016/j.jmr.2019.02.006>.

References

- [1] F. Delaglio, S. Grzesiek, G. Vuister, G. Zhu, J. Pfeifer, A. Bax, *J. Biomol. NMR* 6 (1995) 277.
- [2] J.J. Helmus, C.P. Jaroniec, *J. Biomol. NMR* 55 (2013) 355.
- [3] J.D. van Beek, *J. Magn. Reson.* 187 (2007) 19.
- [4] D. Massiot, F. Fayon, M. Capron, I. King, S. Le Calvé, B. Alonso, J.-O. Durand, B. Bujoli, Z. Gan, G. Hoatson, *Magn. Reson. Chem.* 40 (2002) 70.
- [5] M. Edén, *Concepts Mag. Reson. Part A* 17A (2003) 117.
- [6] M. Edén, *Concepts Mag. Reson. Part A* 18A (2003) 1.
- [7] M. Edén, *Concepts Mag. Reson. Part A* 18A (2003) 24.
- [8] M.J. Duer, *Introduction to Solid-state NMR Spectroscopy*, Wiley-Blackwell, 2005.
- [9] M. Bak, J.T. Rasmussen, N.C. Nielsen, *J. Magn. Reson.* 147 (2000) 296.
- [10] M. Veshtort, R.G. Griffin, *J. Magn. Reson.* 178 (2006) 248.
- [11] G. Czjzek, J. Fink, F. Götz, H. Schmidt, J.M.D. Coey, J.-P. Rebouillat, A. Liénard, *Phys. Rev. B* 23 (1981) 2513.
- [12] P.J. Knijn, P.J.M. van Bentum, E.R.H. van Eck, C. Fang, D.L.A.G. Grimminck, R.A. de Groot, R.W.A. Havenith, M. Marsman, W.L. Meerts, G.A. de Wijs, A.P.M. Kentgens, *PCCP* 12 (2010) 11517.
- [13] G.L. Caër, R.A. Brand, *J. Phys.: Condens. Matter* 10 (1998) 10715.
- [14] G.L. Caër, B. Bureau, D. Massiot, *J. Phys.: Condens. Matter* 22 (2010) 065402.
- [15] D.L. Grimminck, B.J. Polman, A.P. Kentgens, W. Leo Meerts, *J. Magn. Reson.* 211 (2011) 114.
- [16] D.L.A.G. Grimminck, B. van Meerten, M.H.W. Verkuijlen, E.R.H. van Eck, W.L. Meerts, A.P.M. Kentgens, *J. Mag. Reson. (San Diego, Calif.: 1997)* 228 (2013) 116.
- [17] E.S.M. Blaakmeer, G. Antinucci, E.R.H. van Eck, A.P.M. Kentgens, *J. Phys. Chem. C* 122 (2018), [acs.jpcc.8b05123](https://doi.org/10.1021/acs.jpcc.8b05123).
- [18] J. Skibsted, H.J. Jakobsen, *J. Phys. Chem. A* 103 (1999) 7958.
- [19] T. Gullion, Rotational-echo, double-resonance NMR, in: *Modern Magnetic Resonance*, Springer, 2008, pp. 713–718.
- [20] R.E. Marsh, *Acta Crystallogr. A* 11 (1958) 654.
- [21] R.A. Haberkorn, R.E. Stark, H. Van Willigen, R.G. Griffin, *J. Am. Chem. Soc.* 103 (1981) 2534.