Electronic Supplementary Material (ESI) for Chemical Communications
This journal is © The Royal Society of Chemistry 2016

Increasing the quantitative bandwidth of NMR measurements

Electronic Supporting Information

*Jane E. Power, Mohammadali Foroozandeh, Ralph W. Adams, Mathias Nilsson, Steven R. Coombes, Andrew R. Phillips and Gareth A. Morris**

Index

1. Generation of chirp swept-frequency pulses

First, the chirp pulses were generated, as in Bruker software in the Bruker format; those used in this work had the parameters a), b) and c). A time-dependent phase correction $\emptyset(x)$ was then subtracted from the first 90° and first 180° chirp elements, as described on p. S4. A Matlab notebook for producing the three pulse shapes, the shapes themselves, and all of the raw data and other software used can be downloaded from DOI: 10.15127/1.276417 and DOI: 10.15127/1.276419.

a) First 90° chirp element

Number of points: 2000
Total sweep width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
Q (for the middle of shape): 5.0

b) First 180° chirp element

Number of points: 2000
Total sweep width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
Q (for the middle of shape): 5.0

c) Second 180° chirp element

Size of Shape: 1000
Total sweep width: 300 kHz
Duration: 1 ms
% smoothed: 5 %
Q (for the middle of shape): 5.0

2. *Calculation of chirp pulse RF amplitudes*

The RF amplitudes $\left(\gamma B_1/2\pi\right)$ required for the three chirp elements are calculated within the pulse sequence as follows:

a) *Step 1*

The RF amplitudes for the first and second 180° chirp elements are calculated using equation S1, where the pulse duration τ_p is 2 ms and 1 ms respectively; using a sweep width, ΔF, of 300 kHz and adiabatic factor, *Q*, of 5, RF amplitudes of 10925.5 Hz and 15451 Hz respectively are obtained.

b) *Step 2*

The RF amplitude required for the first 90° chirp element (3252.8 Hz) is calculated by dividing the RF amplitude of the second 180° chirp element $\left(\gamma B_{1(\max)}/2\pi\right)$, by 4.75 [13,16].

c) *Equation S1*

$$\frac{\gamma B_1}{2\pi} = \sqrt{\frac{\left(\tau_p \times Q \times \Delta F\right)}{2\pi}} \Big/ \tau_p$$

The relative amplitudes in the triple chirp sequence are shown in Fig. S1.

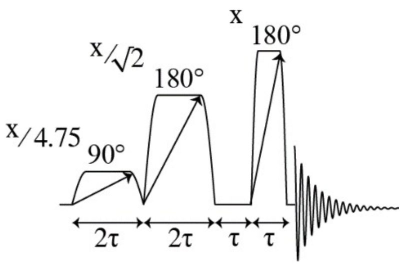


Fig. S1. The triple chirp sequence, CHORUS; RF amplitude (x) $\left(\gamma B_{1(\max)}/2\pi\right)$ for a 300 kHz sweep width is x = 15451 Hz.

3. Double and triple chirp pulse sequences

The double chirp sequence of Bodenhausen *et al* [13] (Fig. S2a), can be used to excite very wide bandwidths. This is demonstrated in the excitation profiles of Fig. S2b, where experimental data (dots) and simulations (green lines), are compared for simple 90° excitation (blue dots) and the double chirp sequence (red dots). All simulations were carried out in Mathematica *v.9* using compiled analytical solutions of the Bloch equations, and used no linearly dependent phase correction (in contrast to the 90° excitation data of Fig. 1b of the main text). The profiles are constructed from experiments in which the frequency of the excitation is varied in equal steps but the receiver is kept on resonance to eliminate any bias caused by the receiver characteristics. Although the bandwidth of excitation is improved compared to the hard pulse, phase errors build up towards the edges of the frequency range, reducing the usable bandwidth of the method (Fig. S2b). A further, hidden, problem is that the signal phase is extremely sensitive to B_1 amplitude, so that B_1 inhomogeneity causes large (> 30%) losses in signal even with modern probes (Fig. S2b).

The double chirp pulse sequence element was therefore extended to a triple chirp (Fig. S3a), to compensate for B_1 sensitivity. The remaining phase variation with frequency was corrected using a 9th order polynomial fit (Equation S2), and subtracting the equivalent time-dependent phase correction $\emptyset(x)$ from the phase of the first two chirp pulses of CHORUS (Fig. S3a). This correction is straightforward, since the chirp pulses have a one-to-one correspondence between frequency and time. A Mathematica notebook for these calculations, and those for Fig. 1b of the main text, is included at the package that can be downloaded from DOI: 10.15127/1.276417.

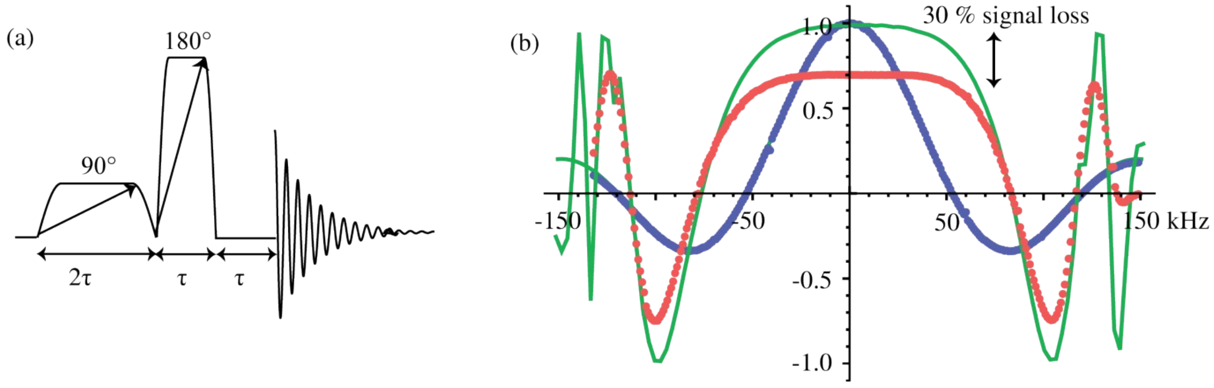


Fig. S2. (a) Double chirp sequence of Bodenhausen *et al.*;[13] (b) excitation profiles of experimental data (dots) for a doped water sample and simulations (solid lines) for a hard pulse (blue) and for the chirp pulse sequence (red) of Fig. S2a, over a frequency range of 300 kHz.

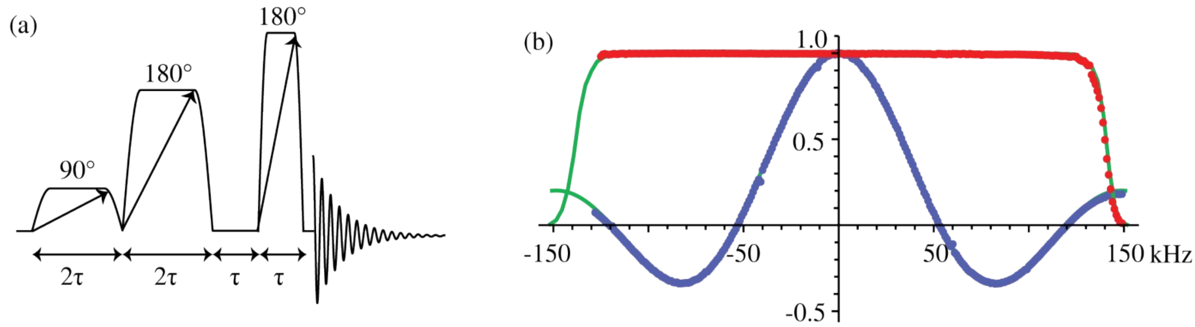


Fig. S3 (a) The triple chirp sequence, CHORUS, adapted from the double chirp (Fig. S2a); (b) excitation profiles of experimental data (dots) for a doped water sample and simulations (solid lines) for a hard pulse (blue) and for the chirp pulse sequence (red) of Fig. S3a, over a frequency range of 300 kHz.

a) Equation S2

$$\emptyset(x) = 262.552 + 11.4628\,x - 19.0628\,x^2 + 27.249\,x^3 + 155.681\,x^4$$
$$- 177.911\,x^5 - 429.253\,x^6 + 356.946\,x^7 + 365.884\,x^8 - 245.483\,x^9$$

where $x = (2t - pw) / 2pw$, pw is the chirp pulse width and t is the time from the beginning of the pulse.

4. Excitation profiles: CHORUS vs ABTRUSE

The ABSTRUSE sequence (Fig. S4(b)), developed by Shaka and co-workers[19] is based on hyperbolic secant (HS) pulses, while CHORUS is based on linearly swept chirp pulses (Fig. S4a). Figures S4(c) and S4(d) compare the excitation profiles achievable with the two sequences, Fig. S4(c) showing the in-phase component of magnetization immediately after the pulse sequence and Fig. 4(d) the signal obtained after a linear phase correction is applied to the ABSTRUSE signal (no correction is needed for CHORUS). The same maximum RF amplitude $\left(\gamma B_{1(\max)}/2\pi\right)$ of 15 kHz and duration 6τ of 6 ms were used in each case, with the range of frequency sweep set as in section 1 above for CHORUS, and according to equation [7] and figure caption 6 of reference 19 of the main text for ABSTRUSE, respectively.

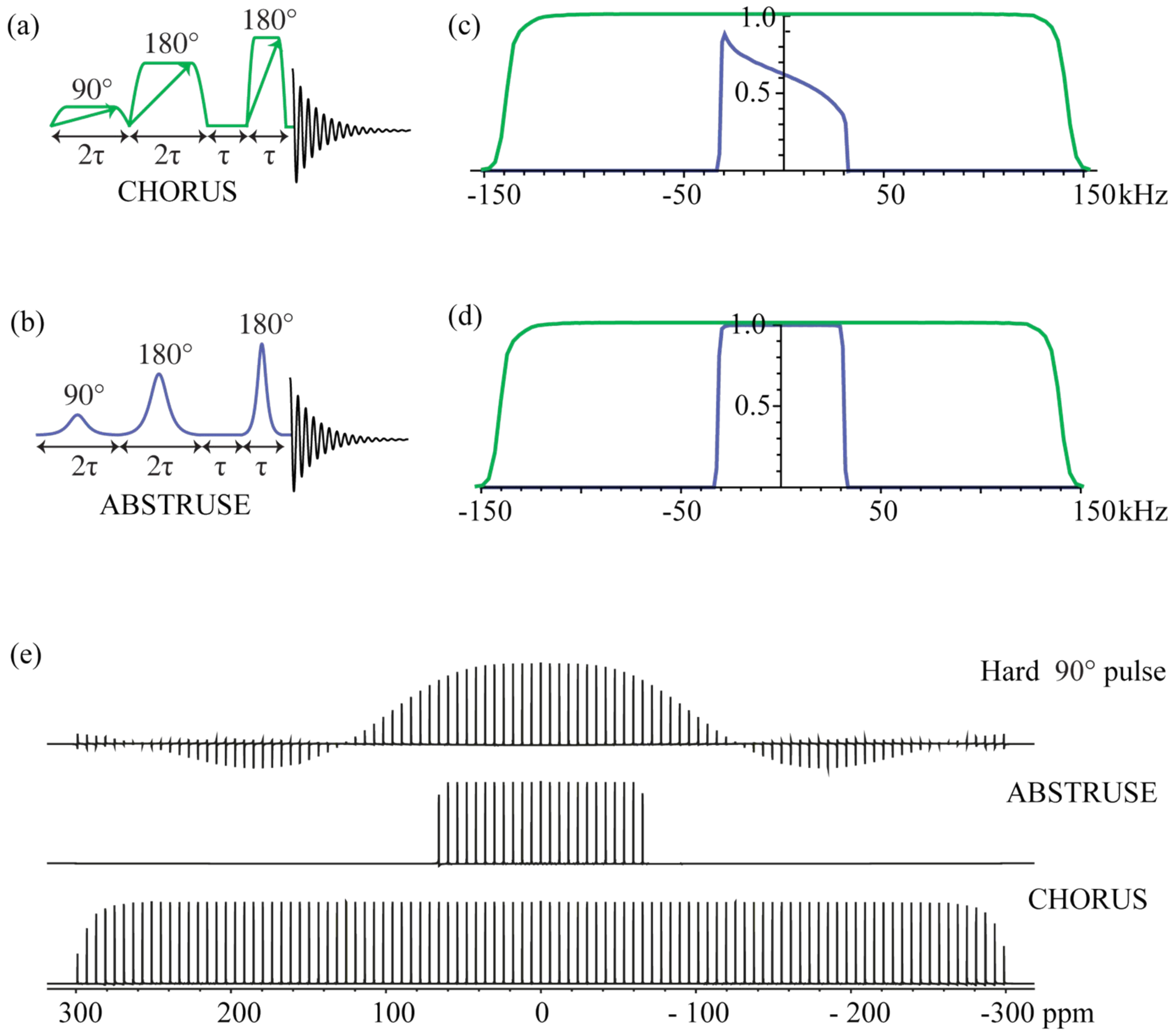


Fig. S4 (a) and (b) CHORUS and ABSTRUSE pulse sequences respectively; (c) and (d) calculated signal (c) and phase-corrected signal (d) profiles for sequences (a) (CHORUS, green) and (b) (ABSTRUSE, blue) respectively; and (e) comparison of simulated spectra as a function of offset for a hard 90° pulse (top), ABSTRUSE (middle) and CHORUS (bottom).

5. Two measures of performance

For quantitative use CHORUS should show good repeatability and robustness with respect to resonance offset; a common criterion for acceptability for these two indices of performance is ≤ 1% relative standard deviation for peak area ratios. Experimental data were acquired for both measures of performance using a sample of bicalutamide (65 mM) and 4-fluoroaniline (221 mM) in DMSO-d_6 (Fig. S5) in a 500 MHz spectrometer (Bruker Avance III, 5 mm BBO probe, QNP switch). ^{19}F{^{1}H} data were acquired with 16 scans over a spectral width (*sw*) of 234 ppm. The approximate S/N ratios of the three peaks using CHORUS and the hard pulse are shown in Table S1(a); the theoretical relative integrals of the three peaks (1:2), (1:3) and (2:3), are shown in Table S1(b). The experimental data were obtained by converting Bruker data to VnmrJ format, zero-filling to 524288 real points and using a time-domain weighting function corresponding to a 5 Hz Lorentzian. The phase for the first spectrum in each set was adjusted and integral resets set at 0.25 ppm either side of each peak, and automated polynomial baseline correction and integral determination were then carried out for all spectra.

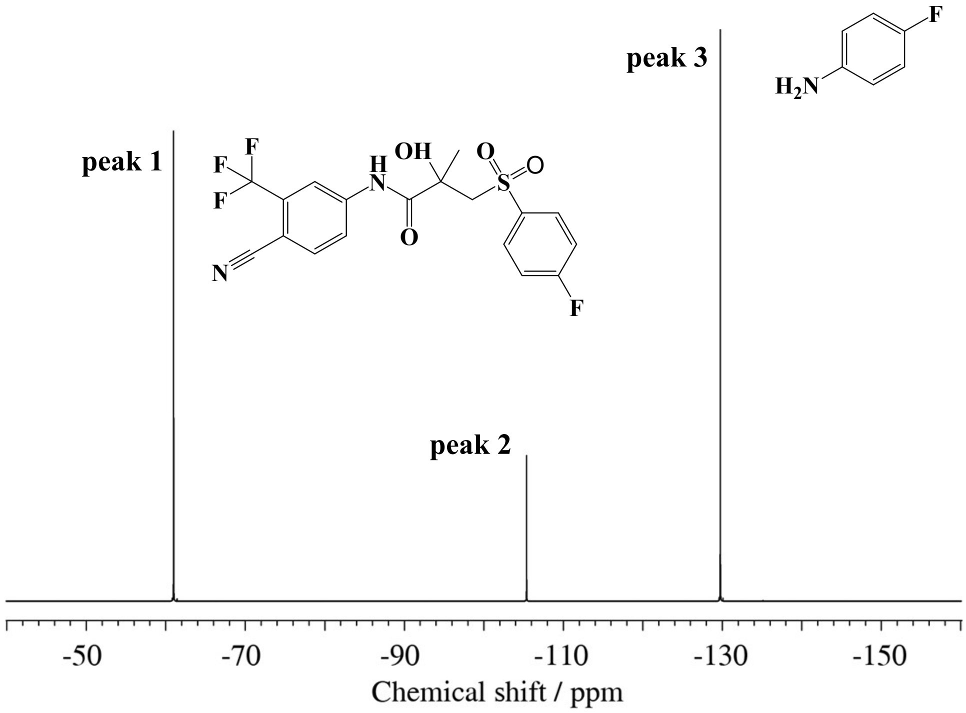


Fig. S5. ^{19}F{^{1}H} spectrum of a sample of bicalutamide (peaks 1 and 2, at -61.0 ppm and -105.4 ppm, respectively) and reference material 4-fluoroaniline (peak 3 at -129.8 ppm) in DMSO-d_6; the ^{19}F NMR spectrum was acquired using CHORUS and processed with a line broadening (*LB*) of 5 Hz.

	Approximate S/N ratio		
Table S1(a)	**peak1 (-61.0 ppm)**	**peak2 (-105.4 ppm)**	**peak3 (-129.8 ppm)**
Hard pulse (-100 ppm)	16900	5200	17200
CHORUS (-100 ppm)	17600	5400	19200

	Theoretical relative integrals		
Table S1(b)	**peaks (1:2)**	**peaks (1:3)**	**peaks (2:3)**
Theoretical relative integral ratios	3.000	0.8817	0.2939

Table S1. a) Approximate S/N ratios of the three peaks of Fig. S5. Spectra were acquired using CHORUS and a 90° pulse. The $^{19}F\{^{1}H\}$ data were processed with a line broadening (*LB*) of 5 Hz. b) Theoretical relative integrals of peaks (1:2), (1:3) and (2:3), as calculated from the mole ratios and the percentage composition by mass of the sample.

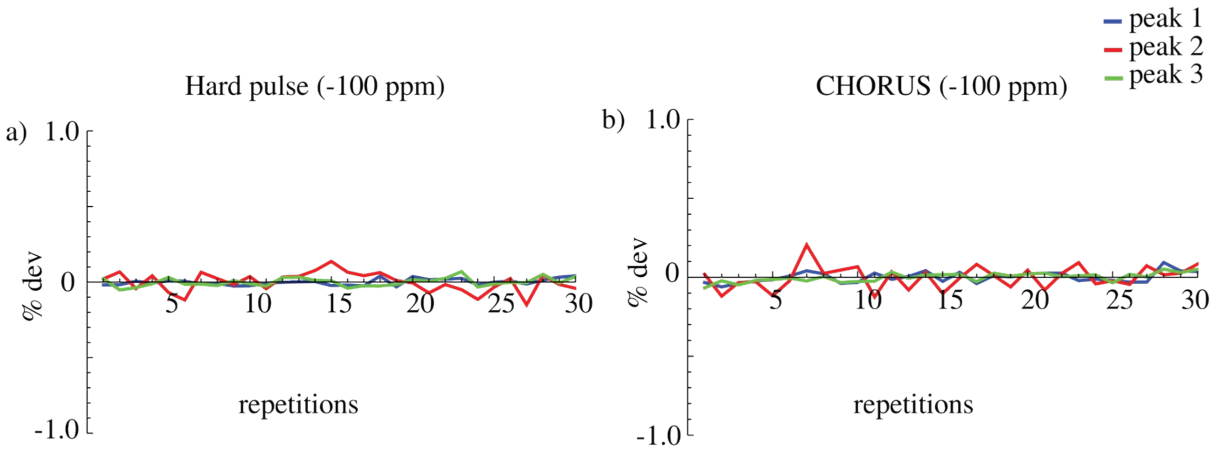
5.1. Repeatability

Repeatability of the method was investigated by comparing individual absolute integrals across 30 spectra of the same sample acquired consecutively. Acquisitions with CHORUS were interleaved with on-resonance hard pulse acquisitions, in order to monitor spectrometer drift, to give 60 acquisitions in total. The spectrometer operating frequency for both pulses was set at -100 ppm. The ratios of relative integrals peaks (1:2), (2:3) and (1:3) are summarized in Table S2, with means and relative standard deviations. Graphs of percentage deviations of absolute integrals for the three peaks are shown in Figs. S6a and S6b. For both CHORUS and 90° excitation, repeatability of peak area ratios was in both cases more than an order of magnitude better than the criterion of ≤ 1%.

Integral ratio summary

Table S2		**peaks (1:2)**	**peaks (1:3)**	**peaks (2:3)**
Hard Pulse	Mean	2.947	0.8743	0.2966
	% RSD	0.072	0.027	0.072
CHORUS	Mean	3.022	0.8798	0.2911
	% RSD	0.076	0.028	0.078

Table S2. Means and relative standard deviations for ratios of peak integrals (1:2), (2:3) and (1:3), in a test of repeatability. The data were acquired using interleaved CHORUS and hard pulse acquisitions with spectrometer operating frequencies set at -100 ppm.



Figs. S6. a) and b) Percentage deviations of absolute integrals of peaks 1, 2 and 3 for hard pulse and CHORUS, with spectrometer operating frequencies set at -100 ppm.

5.2. Robustness with respect to offset

For quantitative use, CHORUS must show uniform excitation with constant amplitude and constant (or at least linearly varying) phase over the full spectral width. In order to test this measure of performance, CHORUS measurements were, as before, interleaved with measurements using a hard pulse on resonance (o1p at -100 ppm). For CHORUS, the spectrometer operating frequency (o1p) of -100 ppm was varied from -20 ppm to -165 ppm, in steps of 5 ppm, to give 30 experiments covering an offset range of 145 ppm (68 kHz), with three repetitions. Because the frequencies of the received signals depend on o1p, signal amplitudes are affected by the characteristics of the receiver. While such effects are much smaller with modern digital signal processing than they were with the analogue filters used on older spectrometers, they are still significant. This small fixed error shows a quadratic dependence (see Figs. S9a and S9b, p. S14) on frequency, for which the CHORUS raw data can be corrected straightforwardly. Data for the relative integral ratios of peaks (1:2), (2:3) and (1:3) are summarized in Tables S3(a), S3(b) and S3(c) for the three repeats. For CHORUS, the raw data are shown (a) before correction, (b) after correction for the receiver characteristics and c) after correction for both systematic drift and receiver characteristics; for the hard pulse, the raw data are shown.

Graphs S7(a), S7(b) and S7(c) show the percentage deviations of absolute integrals for the three peaks as a function of offset, where S7(a) shows the CHORUS raw data, S7(b) the CHORUS raw data after receiver correction and S7(c) the CHORUS raw data receiver corrected after correction for systematic drift using the hard pulse data. For all data sets, repeatability of peak area ratios is significantly better than the criterion of ≤ 1%, and similar to the reproducibility data of Table S2, implying that any integral variation with respect to offset is around or below the threshold of detection in these experiments.

5.3. Statistical data for both measures of performance

a) The average integral ratios for CHORUS (Tables S2, S3(a), S3(b) and S3(c)) show significant deviation from the theoretical values (Table S1(b)). This is attributable partly to T_2 relaxation, where peaks 1, 2 and 3 have approximate T_2 values of 680 ms, 470 ms and 2.9 s respectively and partly to low-level impurities present in the sample.

b) The average integral ratios for hard pulse excitation also shows significant deviation from the ideal; however, this is to be expected because of off resonance effects. When the data are corrected for the receiver characteristics and for the calculated off-resonance effects of an ideal 90° pulse (with appropriate first-order phase correction), the data for the first repeat give mean values for the ratio (1:2) of 2.990, ratio (1:3) 0.8821 and ratio (2:3) 0.2950.

Table S3(a)

Repeat (1)		peaks (1:2)	peaks (1:3)	peaks (2:3)
		Integral ratio summary		
Hard Pulse	Mean	2.951	0.8750	0.2966
	% RSD	0.051	0.030	0.040
(a) CHORUS (raw data)				
	Mean	3.024	0.8814	0.2914
	% RSD	0.14	0.22	0.12
(b) CHORUS (receiver-corrected)				
	Mean	3.024	0.8805	0.2912
	% RSD	0.070	0.024	0.065

Table S3a. Means and relative standard deviations for ratios of peak integrals (1:2), (2:3) and (1:3), in a test of robustness for repeat 1. The data were acquired using interleaved CHORUS and hard pulse acquisitions, with the latter on resonance (o1p at -100 ppm).

Table S3(b)

Integral ratio summary

Repeat (2)		**peaks (1:2)**	**peaks (1:3)**	**peaks (2:3)**
Hard Pulse	Mean	2.947	0.8744	0.2967
	% RSD	0.051	0.024	0.054
(a) CHORUS (raw data)				
	Mean	3.025	0.8812	0.2913
	% RSD	0.13	0.23	0.14
(b) CHORUS (receiver-corrected)				
	Mean	3.024	0.8804	0.2911
	% RSD	0.080	0.028	0.081

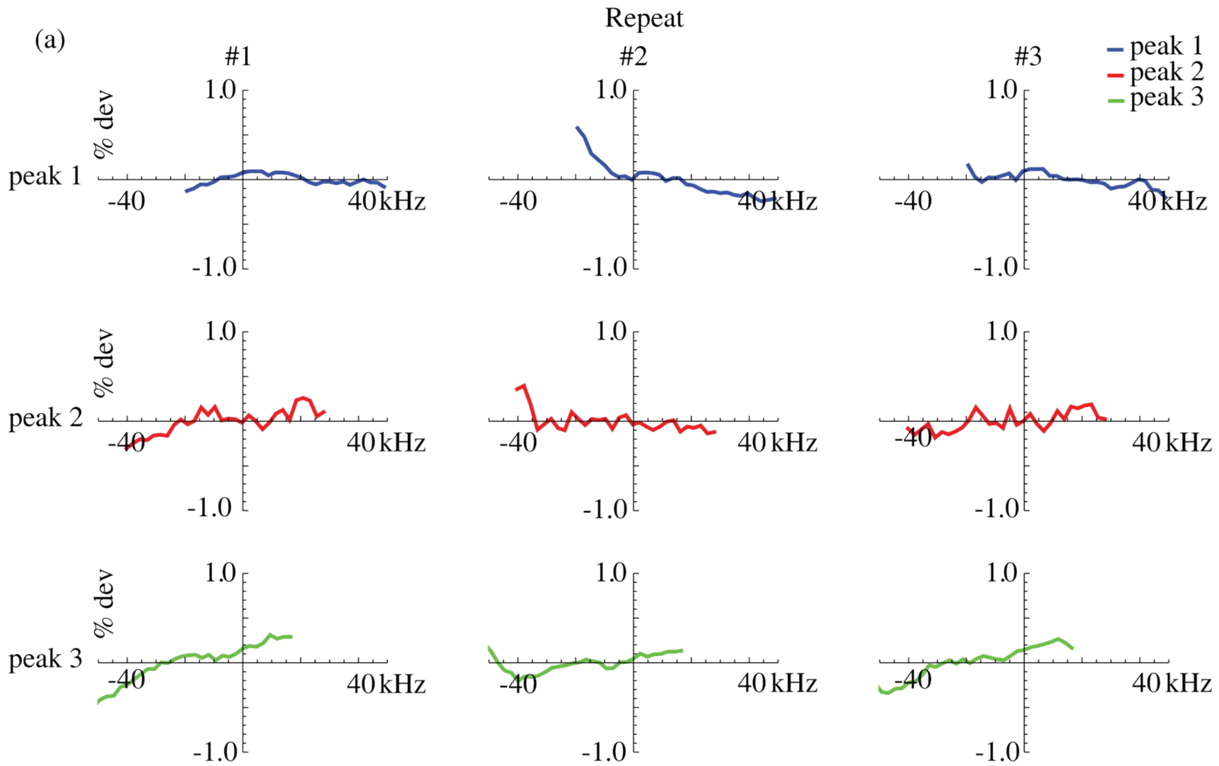
Table S3b. Means and relative standard deviations for ratios of peak integrals (1:2), (2:3) and (1:3), in a test of robustness for repeat 2. The data were acquired using interleaved CHORUS and hard pulse acquisitions, with the latter on resonance (o1p at -100 ppm).

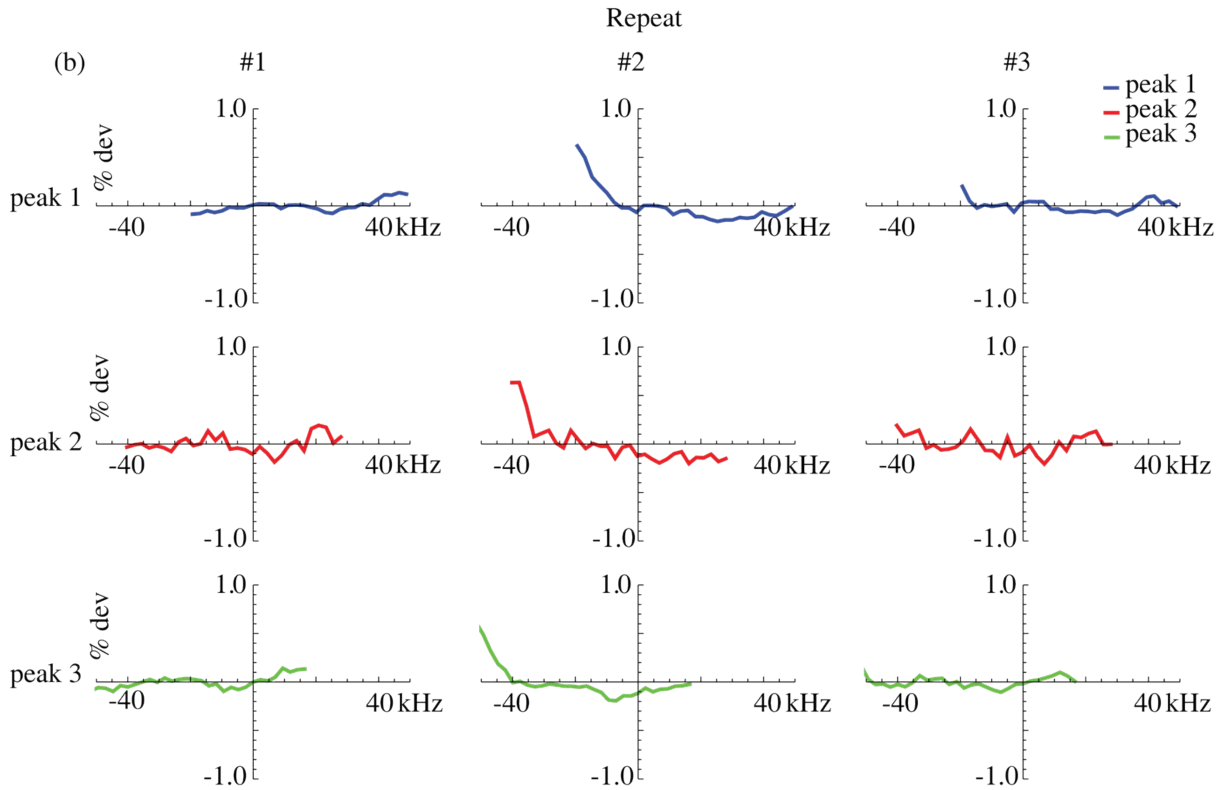
Table 3S(c)

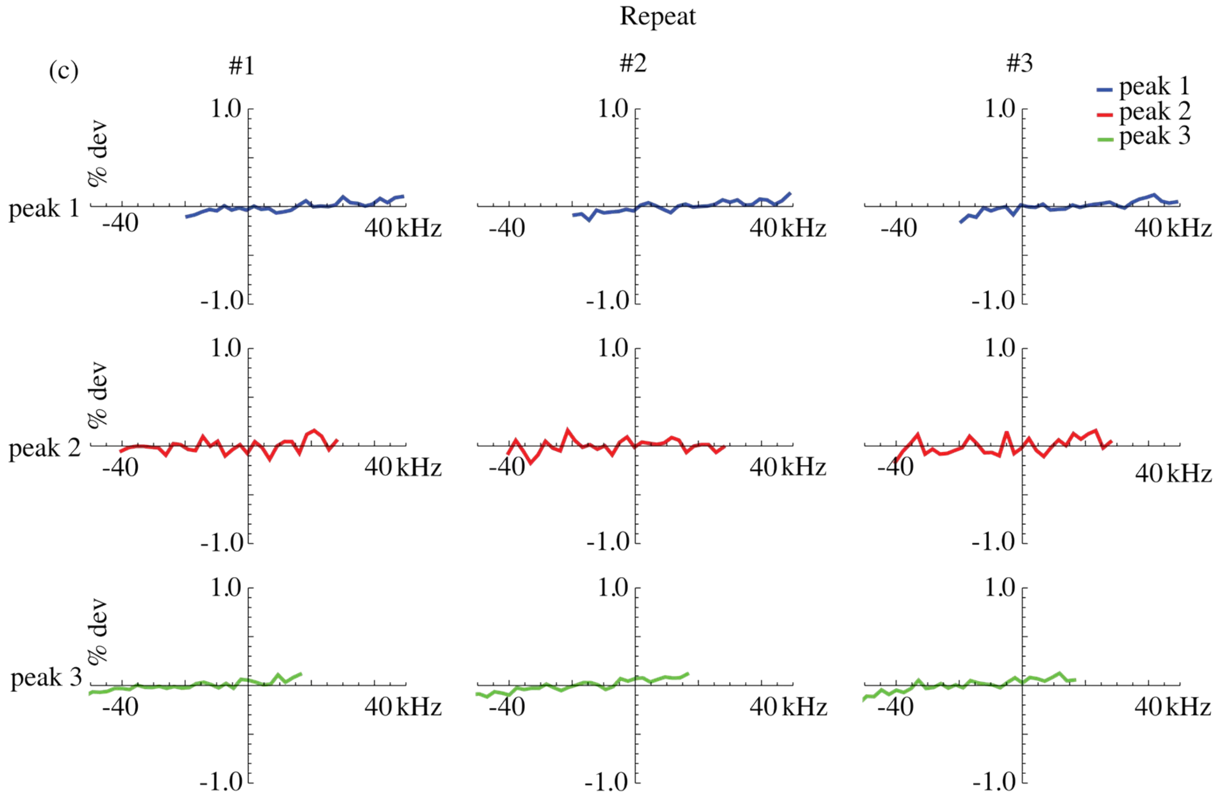
Integral ratio summary

Repeat (3)		**peaks (1:2)**	**peaks (1:3)**	**peaks (2:3)**
Hard Pulse	Mean	2.947	0.8743	0.2966
	% RSD	0.066	0.022	0.060
(a) CHORUS (raw data)				
	Mean	3.024	0.8810	0.2913
	% RSD	0.15	0.23	0.12
(b) CHORUS (receiver-corrected)				
	Mean	3.023	0.8801	0.2911
	% RSD	0.071	0.037	0.075

Table S3c. Means and relative standard deviations for ratios of peak integrals (1:2), (2:3) and (1:3), in a test of robustness for repeat 3. The data were acquired using interleaved CHORUS and hard pulse acquisitions, with the latter on resonance (o1p at -100 ppm).


(a)
Repeat
#1
#2
#3
peak 1
peak 2
peak 3
% dev
1.0
-1.0
-40
40 kHz


(b)
Repeat
#1
#2
#3
peak 1
peak 2
peak 3
% dev
1.0
-1.0
-40
40 kHz



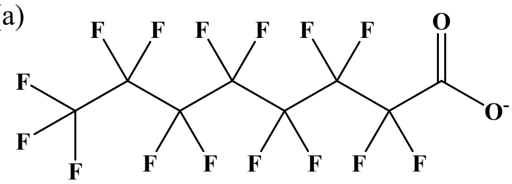
Figs. S7. a), b) and c) Percentage deviations of absolute integrals of the three peaks, as a function of offset, for three repetitions, where (a) is the CHORUS raw data, (b) the CHORUS raw data receiver-corrected and (c) the CHORUS raw data receiver-corrected after correction for systematic drift by normalisation to the on resonance hard pulse data.

6. CHORUS and J-modulation

Swept-frequency pulses are relatively long, in the order of milliseconds. For CHORUS, which has three chirp elements, for a 300 kHz bandwidth and with a τ of 1 ms, the total pulse duration is 6 ms (Fig. S1). Due to this lengthy duration, the pulse sequence can suffer from *J*-modulation if large homonuclear couplings are present.

To demonstrate the effects of *J*-modulation when using CHORUS, a sample of sodium perfluorooctanoate (NaPFO) in DMSO-d_6 was used. For a signal on resonance, the ^{19}F spectrum showed up to 20 % reduction in integral for CHORUS (red) in comparison to a hard pulse (blue) (Fig. S8). The surfactant has a relatively long T_2 of 0.4 s, so relaxation is not to blame; rather, the 6 ms duration of CHORUS is long enough for significant evolution under the ^{19}F-^{19}F couplings in the PFO anion. This causes distortion of multiplet shapes and hence reduces net signal integrals.

(a)



(b)

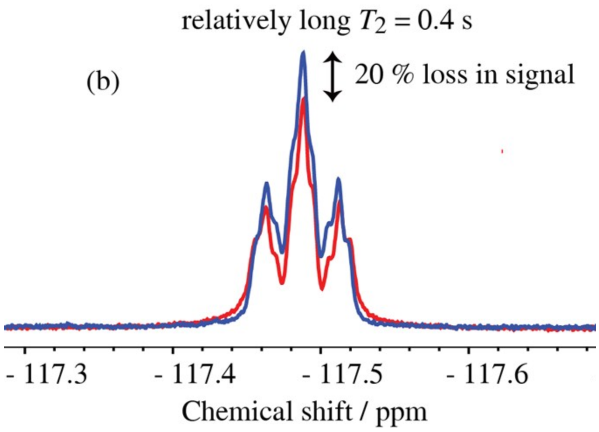


Fig. S8. a) Structure of surfactant PFO anion, and b) ^{19}F spectrum of the alpha CF_2 NaPFO signal on resonance for CHORUS (red) and for a 90° hard pulse (blue). For CHORUS, the signal shows approximately 20 % loss in integral compared to the hard pulse, due to *J*-modulation.

7. Effects of receiver signal filtration on signal amplitude

To show the effects of the analogue and digital receiver filters on signal amplitude, excitation profiles were constructed from an experiment using a doped sample of C_6F_6 in DMSO-d_6, in which the signal was excited by a 90° pulse on resonance but the receiver frequency was varied in 1 kHz steps over a spectral width of 300 kHz. To allow correction for any spectrometer drift, measurements with receiver off and on resonance were interleaved. The receiver profile (red) shows an approximately 3% reduction close to the edges of the frequency range (± 150 kHz), while the interleaved on resonance data (blue) show negligible variation in signal (Fig. S9a). Fitting the receiver characteristic to a quadratic (Fig. S9b) allowed the data for robustness with respect to offset (Table 1 of main paper) to be corrected for this source of systematic error.

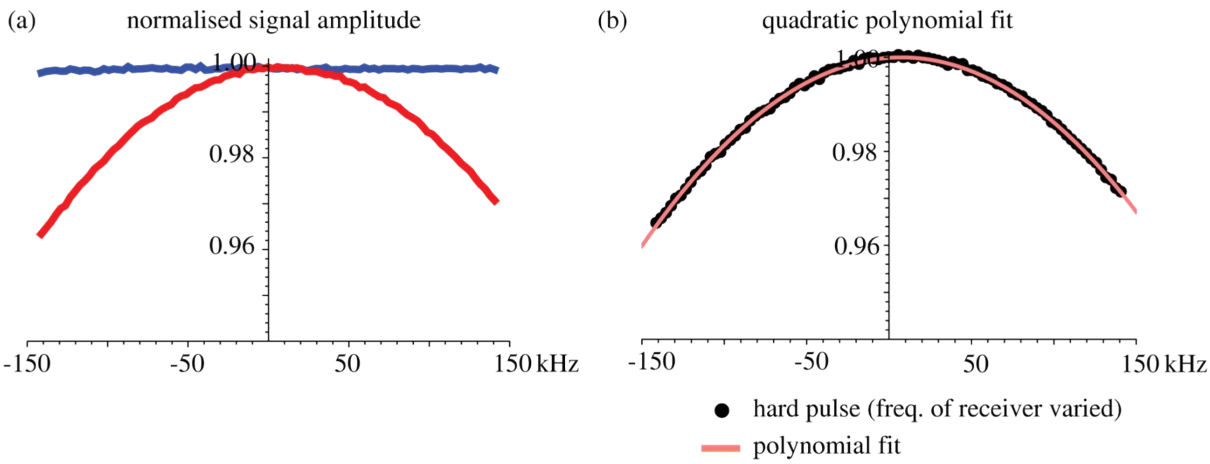


Fig. S9 a). Signal profiles constructed from a frequency-arrayed experiment in which alternate 90° pulse acquisitions were performed, with transmitter on resonance (red) and receiver offset varied, and a reference measurement (blue) with both transmitter and receiver on resonance; b) quadratic fitting (pink) of the signal amplitude variation.

8. *Pulse sequence code*

```
; CHORUSdec.mfjp
; 1H decoupled
; 1D sequence

; Mohammadali Foroozandeh
; Jane Power
; University of Manchester
; (19/08/2015)
; Avance II+/III Version

#include <Avance.incl>
#include <Delay.incl>
#include <De.incl>

"cnst50 = cnst52/4.75"
"cnst51 = cnst52/sqrt(2)"
"p30 = 1000000.0/ (cnst50*4)"
"p31 = 1000000.0/ (cnst51*4)"
"p32 = 1000000.0/ (cnst52*4)"
"cnst30 = (p30/p1)*(p30/p1)"
"cnst31 = (p31/p1)*(p31/p1)"
"cnst32 = (p32/p1)*(p32/p1)"
"spw40 = plw1/cnst30"
"spw41 = plw1/cnst31"
"spw42 = plw1/cnst32"
"d10 = p42"
"d11 = 30m"
"d12 = 20u"
"d18 = 10u"
"d19 = 10u"

1 ze
   d11 QNP_X
   d11 pl12: f2
2 30m do: f2
   d1 pl0: f1
   d12 SWITO_F
   (p40: sp40 ph1): f1
   (p41: sp41 ph2): f1
   d10
   d13
   (p42: sp42 ph3): f1
   ACQ_START(ph30,ph31) (2u SWITO_H)
   aq DWELL_GEN: f1 cpd2: f2
   rcyc = 2
   30m do: f2 mc #0 to 2 F0 (zd)
exit

ph1 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ph2 = 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3
ph3 = 0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3
ph30 = 0
ph31 = 0 2 0 2 2 0 2 0 0 2 0 2 2 0 2 0

;pl0:   f1 channel - power level for pulse (default)
;pl12:  power level for decoupling
;d1:   relaxation delay
;d13:   short delay
;p40:   duration of 1st 90-degree swept-frequency pulse
;p41:   duration of 1st 180-degree swept-frequency pulse
;p42:   duration of 2nd 180-degree swept-frequency pulse
;spw40:   RF power of 1st 90-degree swept-frequency pulse
;spw41:   RF power of 1st 180-degree swept-frequency pulse
;spw42:   RF power of 2nd 180-degree swept-frequency pulse
;spnam40:   filename for 1st 90-degree swept-frequency pulse
;spnam41:   filename for 1st 180-degree swept-frequency pulse
;spnam42:   filename for 2nd 180-degree swept-frequency pulse
;cnst52:   maximum RF power of 2nd 180-degree swept-frequency pulse
```

9. *Matlab Notebook*

```
function [RF_max,Ix,Iy,Iz,Phase]=CHORUS_Bloch(tp_exc,sw,np,sm,path)

%%%%%%%%%%%%% inputs %%%%%%%%%%%%%%%%%

% tp_exc : duration of first pulse (90-degree) in us
% sw: bandwidth of CHORUS in Hz
% np: number of points in the first pulse (90-degree) - must be an even number
% sm: smoothing %
% path: the path used for saving final waves,
% e.g. 'C:\Bruker\TopSpin3.2\exp\stan\nmr\lists\wave\user\';

%%%%%%%%%%%%% outputs %%%%%%%%%%%%%%%%%

% RF_max: maximum RF amplitude required for this pulse
% Ix,Iy,Iz,Phase : final x-, y-, and z-magnetizations and phase

%%%%%%%%%%%%% example %%%%%%%%%%%%%%%%%

%
[RF_max,Ix,Iy,Iz,Phase]=CHORUS_Bloch(2000,300000,2000,5,'C:\Bruker\TopSpin3.2\exp\stan\nmr\lis
ts\wave\user\')

% this generates CHORUS sequence with 6 ms duration for the whole element
% (2 ms 90-degree, 2 ms 180-degree, 1 ms delay, and 1 ms 180-degree)
% for 300 kHz excitation bandwidth,
% with 2000 and 1000 points for 2 ms and 1 ms pulses, respectively.
% 5% smoothing is applied either side of each pulse
% final chirp files will be saved in the directory, specified as "path"
% as: CHORUS_1st_300kHz2m5s2000.txt, CHORUS_2nd_300kHz2m5s2000.txt, and
CHORUS_3rd_300kHz1m5s1000.txt,
%
% Mohammadali Foroozandeh
% Manchester, 19/11/2015
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

np1 = np;     % points in pulse
pw1 = tp_exc;     % pulse width us

np2 = np;     % points in pulse
pw2 = tp_exc;     % pulse width us

np3 = np/2;     % points in pulse
pw3 = tp_exc/2;     % pulse width us

f0 = -sw/2;
f1 = sw/2;

Q = 5;
tp1 = pw1*1e-6;
tp2 = pw2*1e-6;
tp3 = pw3*1e-6;

time1 = tp1 / np1;
time2 = tp2 / np2;
time3 = tp3 / np3;

RF_max = (sqrt(tp3*sw*Q/(2*pi))/tp3);

[shape1]=Chirp(pw1,sw,np1,sm);
[shape2]=Chirp(pw2,sw,np2,sm);
[shape3]=Chirp(pw3,sw,np3,sm);

phi1 = shape1(:,2);
amp1 = shape1(:,1);

phi2 = shape2(:,2);
amp2 = shape2(:,1);

phi3 = shape3(:,2);
amp3 = shape3(:,1);

RF1 = (amp1*RF_max/100)/4.75;
```

```
RF2 = 0.9661*(amp2*RF_max/100)/sqrt(2);
RF3 = (amp3*RF_max/100);

%%%%%%%%%%%%%%%%%%%%% phase cycling %%%%%%%%%%%%%%%%%%%%%%%%

ph1 = 90*[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
ph2 = 90*[0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3];
ph3 = 90*[0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3];
ph31= 90*[0 2 0 2 2 0 2 0 0 2 0 2 2 0 2 0];

noffs = 500;
offs = linspace(f0,f1,noffs);
npc=length(ph1);

%%%%%%%%%%%%%%%%%%%%%% Bloch simulation %%%%%%%%%%%%%%%%%%%%

magn_fin_pc = zeros(3,noffs,npc);

for ns = 1:length(ph1)

    parfor n=1:noffs;

        mout = [0;0;0];
        magn = [0;0;1];

        for m=1:np1;

            magn=Rtot(RF1(m),offs(n),pi*(ph1(ns)+phi1(m))/180,time1)*magn;

        end

        for m=1:np2;

            magn=Rtot(RF2(m),offs(n),pi*(ph2(ns)+phi2(m))/180,time2)*magn;

        end

        magn = Rz(2*pi*offs(n)*tp3)*magn;

        for m=1:np3;

            magn=Rtot(RF3(m),offs(n),pi*(ph3(ns)+phi3(m))/180,time3)*magn

        end

        mout =mout+(Rz(-(pi/180)*ph31(ns))*magn);
        magn_fin_pc(:,n,ns)=mout;
    end

end

magn_fin=(sum(magn_fin_pc,3))/length(ph1);

Ix = magn_fin(1,:);
Iy = magn_fin(2,:);
Iz = magn_fin(3,:);
Ixy = sqrt((Ix.^2)+(Iy.^2));
Phase=(1/pi)*angle(complex(Iy,Ix));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
subplot(5,1,1)
plot(offs,Ix);
ylim([-1 1]);
ylabel('Ix');
subplot(5,1,2)
plot(offs,Iy);
ylim([-1 1]);
ylabel('Iy');
subplot(5,1,3)
plot(offs,Iz);
ylim([-1 1]);
ylabel('Iz');
subplot(5,1,4)
plot(offs,Ixy);
ylim([-1 1]);
```

```
ylabel('Ixy');
subplot(5,1,5)
plot(offs,Phase);
ylim([-1 1]);
ylabel('Phase');

%%%%%%%%%%%%%%%%%%%%%% Polynomial fitting %%%%%%%%%%%%%%%%%%%%

figure;
phi_corr=polyfit_ph(Phase,sm,np);

[shape1]=Chirp_Ph(pw1,sw,np1,sm,phi_corr);
phi1 = shape1(:,2);

%%%%%%%%%%%%%%%%%%%%%% Bloch simulation %%%%%%%%%%%%%%%%%%%%%%

magn_fin_pc = zeros(3,noffs,npc);

for ns = 1:length(ph1)

    parfor n=1:noffs;

        mout = [0;0;0];
        magn = [0;0;1];

        for m=1:np1;

            magn=Rtot(RF1(m),offs(n),pi*(ph1(ns)+phi1(m))/180,time1)*magn;

        end

        for m=1:np2;

            magn=Rtot(RF2(m),offs(n),pi*(ph2(ns)+phi2(m))/180,time2)*magn;

        end

        magn = Rz(2*pi*offs(n)*tp3)*magn;

        for m=1:np3;

            magn=Rtot(RF3(m),offs(n),pi*(ph3(ns)+phi3(m))/180,time3)*magn

        end

        mout =mout+(Rz(-(pi/180)*ph31(ns))*magn);
        magn_fin_pc(:,n,ns)=mout;
    end

end

magn_fin=(sum(magn_fin_pc,3))/length(ph1);

Ix = magn_fin(1,:);
Iy = magn_fin(2,:);
Iz = magn_fin(3,:);
Ixy = sqrt((Ix.^2)+(Iy.^2));
Phase=(1/pi)*angle(complex(Iy,Ix));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure;
subplot(5,1,1)
plot(offs,Ix);
ylim([-1 1]);
ylabel('Ix');
subplot(5,1,2)
plot(offs,Iy);
ylim([-1 1]);
ylabel('Iy');
subplot(5,1,3)
plot(offs,Iz);
ylim([-1 1]);
ylabel('Iz');
subplot(5,1,4)
plot(offs,Ixy);
ylim([-1 1]);
```

```
ylabel('Ixy');
subplot(5,1,5)
plot(offs,Phase);
ylim([-1 1]);
ylabel('Phase');

%%%%%%%%%%% writing shapes in Bruker format %%%%%%%%%%%%%

head = [];
endline = [];

dlmwrite('shape.txt',shape1,'delimiter',',','precision','%.6f','newline','pc');

str = fileread('shape.txt');
str = strrep(str, ',', ', ');
fid = fopen(['new_' 'shape.txt'], 'w');
fwrite(fid, str, '*char');
fclose(fid);

dlmwrite('head.txt',head,'delimiter',',','precision','%.6f');
dlmwrite('end.txt',endline,'delimiter',',','precision','%.6f');
dlmwrite('head.txt',[...
    '##TITLE= CHORUS_1st_' num2str(sw/1000) 'kHz' num2str(pw1/1000) 'm' num2str(sm) 's'
num2str(np1) 13 10 ...
    '##JCAMP-DX= 5.00 Bruker JCAMP library' 13 10 ...
    '##DATA TYPE= Shape Data' 13 10 ...
    '##ORIGIN= Bruker BioSpin GmbH' 13 10 ...
    '##OWNER= <M.FOROOZANDEH>' 13 10 ...
    '##DATE= ' num2str(datestr(now,1)) 13 10 ...
    '##TIME= ' num2str(datestr(now,'HH:MM:SS')) 13 10 ...
    '##$SHAPE_PARAMETERS= Type: SmoothedChirp ; Total Sweep-Width [kHz] ' num2str(sw/1000) ' ;
Length of Pulse [msec] ' num2str(pw1/1000) ' ; % to be smoothed ' num2str(sm) 13 10 ...
    '##MINX= ' num2str(min(shape1(:,1))) 13 10 ...
    '##MAXX= ' num2str(max(shape1(:,1))) 13 10 ...
    '##MINY= ' num2str(min(shape1(:,2))) 13 10 ...
    '##MAXY= ' num2str(max(shape1(:,2))) 13 10 ...
    '##$SHAPE_EXMODE= Adiabatic' 13 10 ...
    '##$SHAPE_TOTROT= 1.800000E02' 13 10 ...
    '##$SHAPE_TYPE= Inversion' 13 10 ...
    '##$SHAPE_USER_DEF= ' 13 10 ...
    '##$SHAPE_REPHFAC= ' 13 10 ...
    '##$SHAPE_BWFAC= 2.735620E02' 13 10 ...
    '##$SHAPE_BWFAC50= ' 13 10 ...
    '##$SHAPE_INTEGFAC= 3.236043E-02' 13 10 ...
    '##$SHAPE_MODE= 0' 13 10 ...
    '##NPOINTS= ' num2str(np1) 13 10 ...
    '##XYPOINTS= (XY..XY)' 13 10 ...
fileread('head.txt')],'delimiter','');

dlmwrite('end.txt',['##END=' fileread('end.txt')],'delimiter','');

filename = [path 'CHORUS_1st_' num2str(sw/1000) 'kHz' num2str(pw1/1000) 'm' num2str(sm) 's'
num2str(np1) '.txt'];

if ispc==1
    system('copy head.txt+new_shape.txt+end.txt bruker_chlin.txt')
else
    system('cat head.txt new_shape.txt end.txt > bruker_chlin.txt')
end

copyfile('bruker_chlin.txt', filename);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

head = [];
endline = [];

dlmwrite('shape.txt',shape2,'delimiter',',','precision','%.6f','newline','pc');

str = fileread('shape.txt');
str = strrep(str, ',', ', ');
fid = fopen(['new_' 'shape.txt'], 'w');
fwrite(fid, str, '*char');
fclose(fid);

dlmwrite('head.txt',head,'delimiter',',','precision','%.6f');
dlmwrite('end.txt',endline,'delimiter',',','precision','%.6f');
```

```
dlmwrite('head.txt',[...
    '##TITLE= CHORUS_2nd_' num2str(sw/1000) 'kHz' num2str(pw2/1000) 'm' num2str(sm) 's'
num2str(np2) 13 10 ...
    '##JCAMP-DX= 5.00 Bruker JCAMP library' 13 10 ...
    '##DATA TYPE= Shape Data' 13 10 ...
    '##ORIGIN= Bruker BioSpin GmbH' 13 10 ...
    '##OWNER= <M.FOROOZANDEH>' 13 10 ...
    '##DATE= ' num2str(datestr(now,1)) 13 10 ...
    '##TIME= ' num2str(datestr(now,'HH:MM:SS')) 13 10 ...
    '##$SHAPE_PARAMETERS= Type: SmoothedChirp ; Total Sweep-Width [kHz] ' num2str(sw/1000) ' ;
Length of Pulse [msec] ' num2str(pw2/1000) ' ; % to be smoothed ' num2str(sm) 13 10 ...
    '##MINX= ' num2str(min(shape2(:,1))) 13 10 ...
    '##MAXX= ' num2str(max(shape2(:,1))) 13 10 ...
    '##MINY= ' num2str(min(shape2(:,2))) 13 10 ...
    '##MAXY= ' num2str(max(shape2(:,2))) 13 10 ...
    '##$SHAPE_EXMODE= Adiabatic' 13 10 ...
    '##$SHAPE_TOTROT= 1.800000E02' 13 10 ...
    '##$SHAPE_TYPE= Inversion' 13 10 ...
    '##$SHAPE_USER_DEF= ' 13 10 ...
    '##$SHAPE_REPHFAC= ' 13 10 ...
    '##$SHAPE_BWFAC= 2.735620E02' 13 10 ...
    '##$SHAPE_BWFAC50= ' 13 10 ...
    '##$SHAPE_INTEGFAC= 3.236043E-02' 13 10 ...
    '##$SHAPE_MODE= 0' 13 10 ...
    '##NPOINTS= ' num2str(np2) 13 10 ...
    '##XYPOINTS= (XY..XY)' 13 10 ...
fileread('head.txt')],'delimiter','');

dlmwrite('end.txt',['##END=' fileread('end.txt')],'delimiter','');

filename = [path 'CHORUS_2nd_' num2str(sw/1000) 'kHz' num2str(pw2/1000) 'm' num2str(sm) 's'
num2str(np2) '.txt'];

if ispc==1
    system('copy head.txt+new_shape.txt+end.txt bruker_chlin.txt')
else
    system('cat head.txt new_shape.txt end.txt > bruker_chlin.txt')
end

copyfile('bruker_chlin.txt', filename);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

head = [];
endline = [];

dlmwrite('shape.txt',shape3,'delimiter',',','precision','%.6f','newline','pc');

str = fileread('shape.txt');
str = strrep(str, ',', ', ');
fid = fopen(['new_' 'shape.txt'], 'w');
fwrite(fid, str, '*char');
fclose(fid);

dlmwrite('head.txt',head,'delimiter',',','precision','%.6f');
dlmwrite('end.txt',endline,'delimiter',',','precision','%.6f');
dlmwrite('head.txt',[...
    '##TITLE= CHORUS_3rd_' num2str(sw/1000) 'kHz' num2str(pw3/1000) 'm' num2str(sm) 's'
num2str(np3) 13 10 ...
    '##JCAMP-DX= 5.00 Bruker JCAMP library' 13 10 ...
    '##DATA TYPE= Shape Data' 13 10 ...
    '##ORIGIN= Bruker BioSpin GmbH' 13 10 ...
    '##OWNER= <M.FOROOZANDEH>' 13 10 ...
    '##DATE= ' num2str(datestr(now,1)) 13 10 ...
    '##TIME= ' num2str(datestr(now,'HH:MM:SS')) 13 10 ...
    '##$SHAPE_PARAMETERS= Type: SmoothedChirp ; Total Sweep-Width [kHz] ' num2str(sw/1000) ' ;
Length of Pulse [msec] ' num2str(pw3/1000) ' ; % to be smoothed ' num2str(sm) 13 10 ...
    '##MINX= ' num2str(min(shape3(:,1))) 13 10 ...
    '##MAXX= ' num2str(max(shape3(:,1))) 13 10 ...
    '##MINY= ' num2str(min(shape3(:,2))) 13 10 ...
    '##MAXY= ' num2str(max(shape3(:,2))) 13 10 ...
    '##$SHAPE_EXMODE= Adiabatic' 13 10 ...
    '##$SHAPE_TOTROT= 1.800000E02' 13 10 ...
    '##$SHAPE_TYPE= Inversion' 13 10 ...
    '##$SHAPE_USER_DEF= ' 13 10 ...
    '##$SHAPE_REPHFAC= ' 13 10 ...
    '##$SHAPE_BWFAC= 2.735620E02' 13 10 ...
```

```
    '##$SHAPE_BWFAC50= ' 13 10 ...
    '##$SHAPE_INTEGFAC= 3.236043E-02' 13 10 ...
    '##$SHAPE_MODE= 0' 13 10 ...
    '##NPOINTS= ' num2str(np3) 13 10 ...
    '##XYPOINTS= (XY..XY)' 13 10 ...
fileread('head.txt')],'delimiter','');

dlmwrite('end.txt',['##END=' fileread('end.txt')],'delimiter','');

filename = [path 'CHORUS_3rd_' num2str(sw/1000) 'kHz' num2str(pw3/1000) 'm' num2str(sm) 's'
num2str(np3) '.txt'];

if ispc==1
    system('copy head.txt+new_shape.txt+end.txt bruker_chlin.txt')
else
    system('cat head.txt new_shape.txt end.txt > bruker_chlin.txt')
end

copyfile('bruker_chlin.txt', filename);

end

function out=Rx(phi)

out = [1 0 0; 0 cos(phi) -sin(phi);0 sin(phi) cos(phi)];

end

function out=Ry(phi)

out = [cos(phi) 0 sin(phi);0 1 0;-sin(phi) 0 cos(phi)];

end

function out=Rz(phi)

out = [cos(phi) -sin(phi) 0;sin(phi) cos(phi) 0; 0 0 1];

end

function out = Rtot(Omega,offs,phi,time)

Omega_eff = sqrt(((2*pi*Omega)^2)+((2*pi*offs)^2));

theta=atan2(Omega,offs);

alpha=time*Omega_eff;

out=Rz(phi)*Ry(theta)*Rz(alpha)*Ry(-theta)*Rz(-phi);

end

function [shape]=Chirp(pw,sw,np,sm)

f0 = -sw/2;
f1 = sw/2;
t = linspace(0,pw/1000000,np);

h = sw/max(t);
ch_lin = f0 + h*t;

np_sm = (np * sm) / 100;
w = np - (2*np_sm);
t_mid = t((np_sm + 1) : (w + np_sm));
amp_mid = rectangularPulse(-0.5,0.5,t_mid);
amp_mid = 100*amp_mid;
kk = linspace(0,pi/2,np_sm);
apod_b = 100*(sin(kk));
apod_e = fliplr(apod_b);
pulse = horzcat(apod_b, amp_mid, apod_e );
int_lin = (h*t.^2)/2 + f0*t;
phi_lin = 2*pi*int_lin;
phi_lin_norm = phi_lin-min(phi_lin);
phi_lin_norm = 360*phi_lin_norm/(2*pi);
phi_lin_mod = mod(phi_lin_norm,360);
phase = phi_lin_mod;
amplitude = pulse;
```

```
shape(:,1)=amplitude;
shape(:,2)=phase;
end

function [shape]=Chirp_Ph(pw,sw,np,sm,phi_corr)

f0 = -sw/2;
f1 = sw/2;
t = linspace(0,pw/1000000,np);

h = sw/max(t);
ch_lin = f0 + h*t;

np_sm = (np * sm) / 100;
w = np - (2*np_sm);
t_mid = t((np_sm + 1) : (w + np_sm));
amp_mid = rectangularPulse(-0.5,0.5,t_mid);
amp_mid = 100*amp_mid;
kk = linspace(0,pi/2,np_sm);
apod_b = 100*(sin(kk));
apod_e = fliplr(apod_b);
pulse = horzcat(apod_b, amp_mid, apod_e );
int_lin = (h*t.^2)/2 + f0*t;
phi_lin = 2*pi*int_lin;
phi_lin_norm = phi_lin-min(phi_lin);
phi_lin_norm = 360*phi_lin_norm/(2*pi);
phi_lin_norm = phi_lin_norm + phi_corr;
phi_lin_mod = mod(phi_lin_norm,360);
phase = phi_lin_mod;
amplitude = pulse;
shape(:,1)=amplitude;
shape(:,2)=phase;
end

function phi_corr=polyfit_ph(Phase,sm,np)

a=length(Phase)*sm/100;
xdata = a+1:length(Phase)-a;
phisim = ( Phase * 180 ) + 180 ;
phisim_r = phisim(a+1:length(Phase)-a);
ydata = phisim_r;

p = polyfit(xdata,ydata,10);
f = polyval(p,xdata);
plot(xdata,ydata,xdata,f,'r')
figure;

X_data = linspace(1,length(Phase),np);
phi_corr = polyval(p,X_data);
plot(X_data,phi_corr)

end
```