The Journal is © The Royal Society of Chemistry 2016

Very Broadband Diffusion-Ordered NMR Spectroscopy: ^{19}F DOSY

Electronic Supporting Information

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

Table of Contents

1. *Construction of the CHORUS Oneshot pulse sequence*

The Oneshot sequence (reference 33 of the main text) can be broken down into two parts, diffusion encoding and diffusion decoding, as shown in Fig. S1. The CHORUS Oneshot sequence (see Fig. 1 of the main text) is constructed by replacing hard pulses with appropriate chirp pulses.

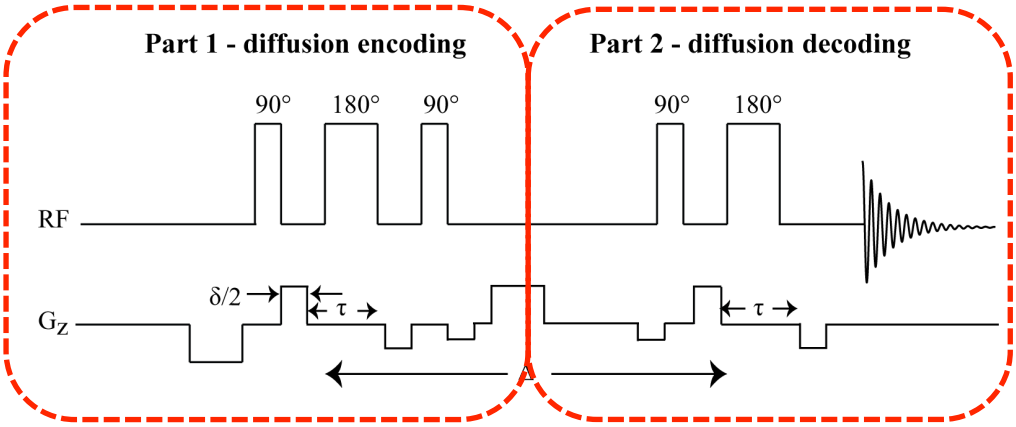

Fig. S1. The two parts of the Oneshot pulse sequence, diffusion encoding and diffusion decoding.

1.1. *Broadband diffusion encoding*

The first hard 180° pulse of Oneshot is replaced by the adiabatic composite chirp pulse (reference 7 of the main text) of Fig. S2a. This is a three-pulse sequence that is self-refocusing with no phase distortion. Uniform, constant phase refocusing over 1 MHz (more than three times that used for the experiments reported in the main text) is demonstrated in the experimental data (red dots) and simulation (green line) of Fig. S2b.

To obtain broadband diffusion encoding, the first and second 90° hard pulses are replaced with 90° chirp pulses, with the sweep direction of the second 90° pulse reversed (Fig. S3a), as described in the main text. To test this sequence an on-resonance 90° read pulse and purge gradient were added (Fig. S3a), and gradient pulses were used to enforce the echo pathway between the two 90° chirp pulses. Uniform, constant diffusion encoding over 250 kHz is demonstrated in the experimental data (dots) and simulation (solid lines) of Fig. S3b, where hard (blue) and chirp (red) pulse results are compared.

For Fig. 1 of the main text and Figs. S2b and S3b, simulations were carried out in Mathematica v.*9* using compiled analytical solutions of the Bloch equations. The profiles were constructed from experiments in which the frequency of the refocusing or the encoding was varied in equal steps but the receiver was kept on resonance to eliminate any bias caused by the receiver characteristics.

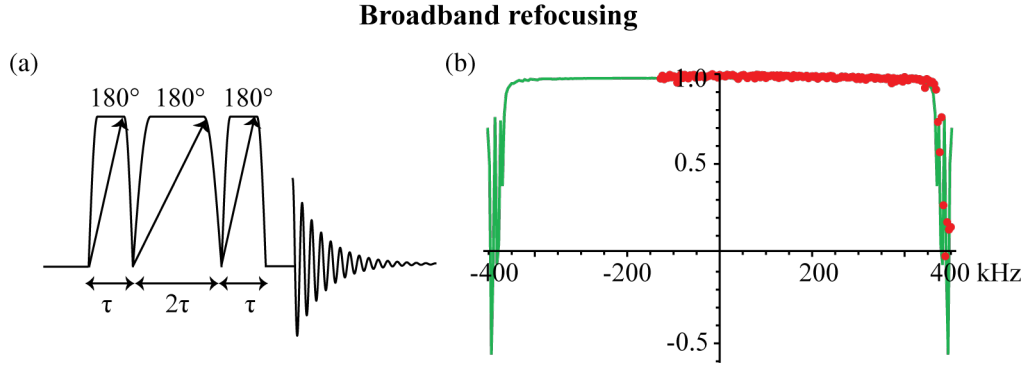

Fig. S2. (a) Adiabatic composite chirp sequence; (b) experimental results (dots) using a doped water sample, with simulations (green line), for the chirp pulse sequence of Fig. S2a. A 1 MHz offset range is covered using 20 kHz RF amplitude for each pulse element, with τ = 1 ms. The experimental data are truncated at –130 kHz because of the limited coherence range of the spectrometer synthesiser used.

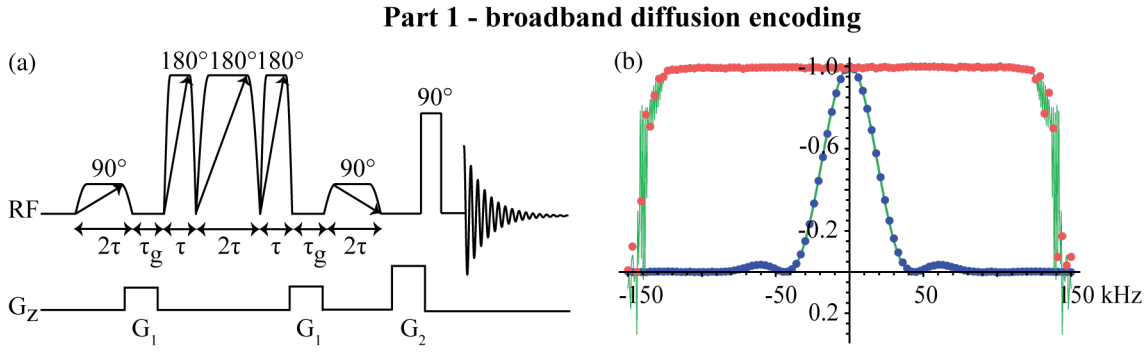

Fig. S3. (a) Broadband diffusion-encoding pulse sequence element; (b) experimental results (dots) over 300 kHz for a doped water sample, and simulations (green lines) for hard (blue) and chirp pulses (red) using the pulse sequence of Fig. S3a.

1.2. Broadband diffusion decoding

Diffusion decoding was achieved using the triple chirp sequence, CHORUS. Details of this pulse sequence and its development can be found in the text and ESI of reference 17 of the main text.

2. Experimental Details

All data were acquired non-spinning on a Bruker Avance III 500 spectrometer at 298 K with 8 transients. Oneshot used an RF amplitude of 25 kHz for the ^{1}H DOSY spectrum Fig. 2a of the main text and 20 kHz for the ^{19}F DOSY (Fig. S5). For CHORUS Oneshot τ was 1 ms, the chirp frequency range 300 kHz, and peak RF amplitude 15 kHz; both used an 8 step phase

cycle. Fig. 2a used the standard Oneshot sequence,[33] and Fig. 2b the pulse sequence of Fig. 3 with a 300 kHz sweep width. Data were acquired in 45 min using 32 gradient amplitudes from 2.7 to 34.5 G cm^{-1} in equal steps of gradient squared, a total diffusion-encoding gradient pulse duration δ of 2.0 ms and a diffusion time Δ of 0.1 s. Gaussian time-domain weighting with a time constant of 53 ms was used to avoid losing the very broad fluticasone signal at –191.52 ppm.

2.1. CHORUS Oneshot sequence and phase cycling

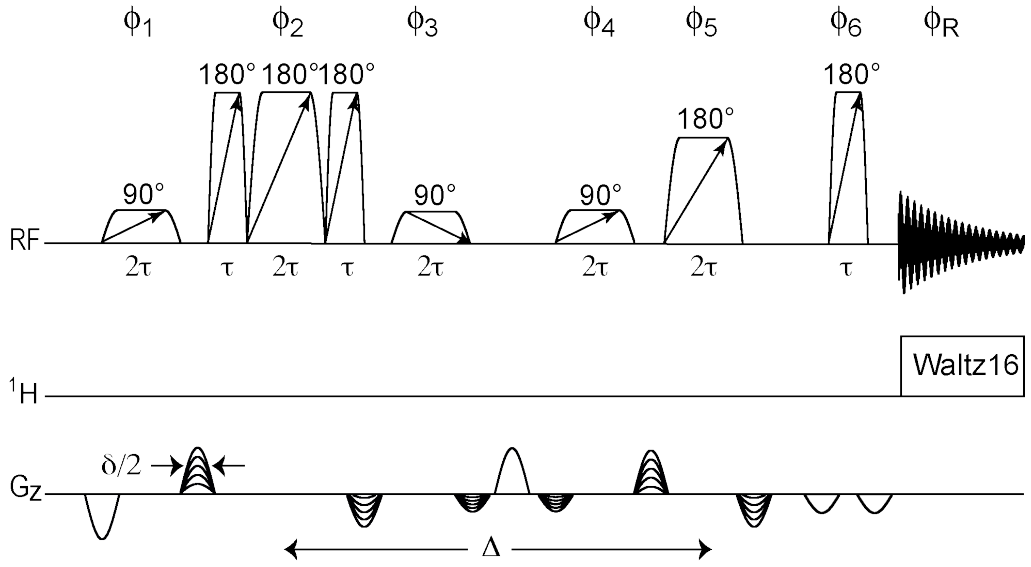

Fig. S4. CHORUS Oneshot sequence with phases indicated.

In the experiments reported, a basic 8-step phase cycle was used (Table S1); where more time averaging is needed, this can be extended to include EXORCYCLE for all 180° pulses.

Phase of Pulse	8-Step Phase Cycle
Φ_1	0 0 0 0 0 0 0 0
Φ_2	0 0 0 0 1 1 1 1
Φ_3	0 0 0 0 0 0 0 0
Φ_4	0 0 0 0 0 0 0 0
Φ_5	0 0 1 1 0 0 1 1
Φ_6	0 1 0 1 0 1 0 1
Φ_R	0 2 2 0 2 0 0 2

Table S1. Phase cycling (in units of 90°) used in the experiments reported in the main text.

3. *Generation of chirp swept-frequency pulses*

The chirp pulses were generated in the Bruker format using the Bruker TopSpin software; those used in this work had the parameters 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 below. A time-dependent phase correction was applied to the CHORUS element of CHORUS Oneshot, i.e. the third 90° (3.4) and the first 180° (3.5) chirp elements, as described in reference 17 of the main text.

A Matlab notebook for producing the three pulse shapes of the CHORUS element of CHORUS Oneshot, all of the pulse shapes themselves, and all of the raw experimental data, pulse sequence code and other materials can be downloaded from DOI: 10.15127/1.296385.

3.1. First 90° chirp element

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
Low to high field

3.2. 180°180°180° composite smoothed chirp

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Length for basic element: 1 ms
% smoothed: 5 %
Low to high field
Q (for the middle of shape): 5.0

3.3. Second 90° chirp element

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
High to low field

3.4. Third 90° chirp element (or first 90° chirp element of CHORUS)

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
Low to high field

3.5. First 180° chirp element of CHORUS

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Duration: 2 ms
% smoothed: 5 %
Low to high field
Q (for the middle of shape): 5.0

3.6. Second 180° chirp element of CHORUS

Number of points (size of shape): 10000
Total sweep-width: 300 kHz
Duration: 1 ms
% smoothed: 5 %
Low to high field
Q (for the middle of shape): 5.0

4. Sample details

The contents of the sample used are listed in Table S2.

	Sample	Chemical Shift / ppm	Conc. / mM
(1)	Rosuvastatin (supplied by AstraZeneca)	−111.91	32
(2)	*BEM (supplied by AstraZeneca)	−111.96	33
(3)	Fluticasone propionate (Sigma)	−164.19, −186.28, −191.52	13
(4)	Fluconazole (extracted from Diflucan formulation)	−107.46, −111.48	26
(5)	Hexafluorobenzene (Sigma)	−162.58	7
(6)	Sulfur hexafluoride (BOC)	59.64	3
(7)	Impurities	−58.22, −111.74, −127.95	1.1, 1.5, 1.3

*tert-butyl-E-(6-[-[4-(4-fluorophenyl)-6-isopropyl-2-[methyl(methylsulfonyl)amino]pyrimidin-5-yl]-vinyl]-((4R,6S)-2,2-dimethyl[1,3]dioxin-4yl) acetic acid).

Table S2. Contents, chemical shifts and concentrations of the sample prepared in DMSO-d_6.

5. *Results with hard pulse Oneshot sequence*

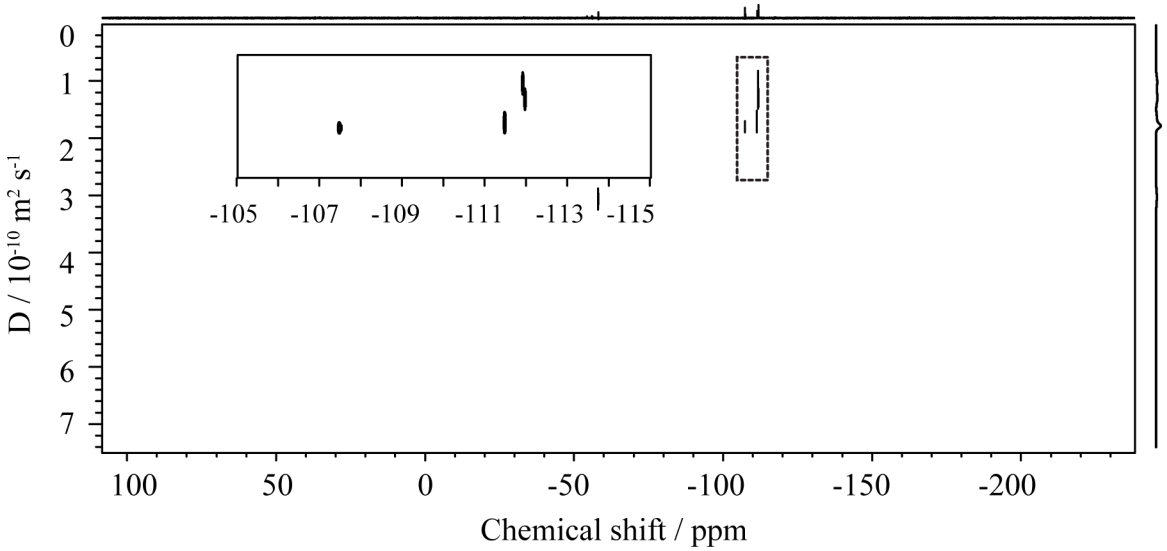

Fig. S5. Results obtained on the mixture of **1** to **6** using the hard pulse Oneshot sequence using the same conditions as Fig. 2 of the main text. The vertical scale for the top trace was expanded by a factor of 2.3 to make the signals visible. Only signals close to the transmitter frequency are excited, and even those have very low intensity.

6. *Impurity signals*

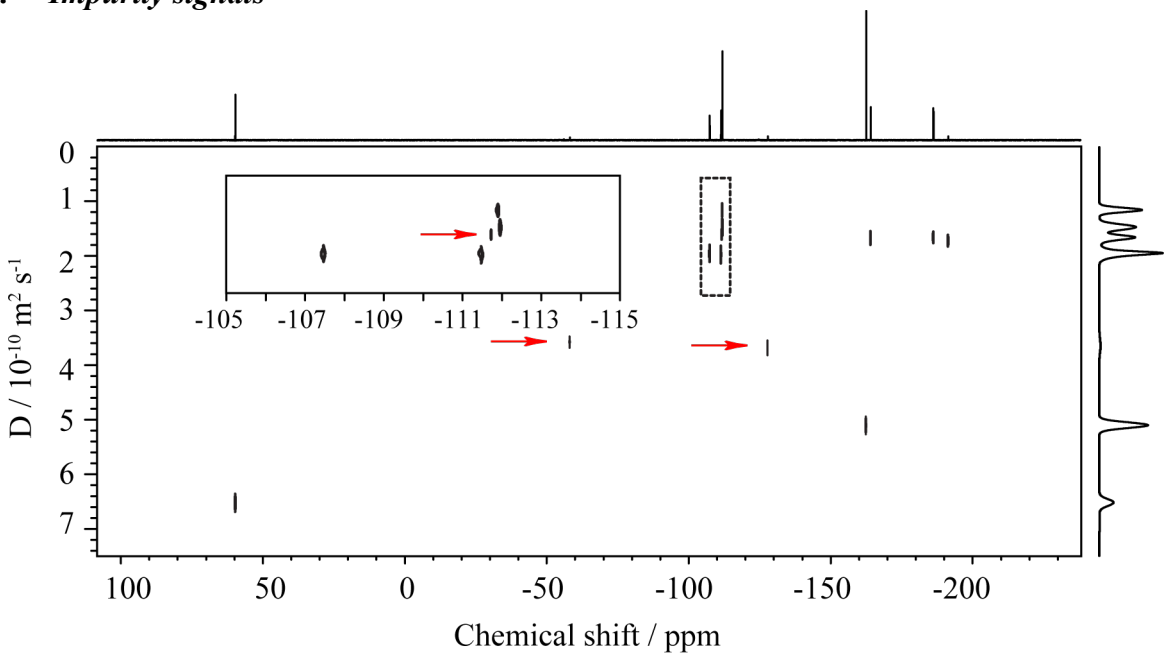

Fig. S6. Data from Fig. 2b of the main text replotted at higher vertical scale. Three impurity signals, marked by red arrows, are seen, one adjacent to the rosuvastatin and BEM signals and with a similar diffusion coefficient to BEM, and two at a diffusion coefficient of about 3.6×10^{-10} m^2 s^{-1}, consistent either with a small molecule containing two fluorine atoms, or two species of similar size each containing one fluorine.

7. *Pulse sequence code*

```
; Doneshotspdec
; by MF and JP
; CHORUS-DOSY pulse sequence
;
;
; Source citation:
; M.D. Pelta, G.A. Morris, M.J. Stchedroff, S.J. Hammond, Magn. Reson. Chem. 40 (2002)
S147-S152.
;
; J.E.Power, M. Foorzandeh, R.W. Adams, M. Nilsson, S.R. Coombes, A.R. Phillips and G.A.
Morris, Chem. Commun., 2016, 52, 2916-2919.
;
;2D Doneshot DOSY pulse sequence
;$CLASS=HighRes
;$DIM=2D
;$TYPE=
;$SUBTYPE=
;$COMMENT=


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


define list<gradient> diff=<Difframp>

"cnst17=(2*p1+d16+p30)*0.000001"; Dtau

;Assuming square gradient pulses
;"cnst18=0.000001*p30*2*0.000001*p30*2*(d20+(0.000001*p30/3)*(cnst14*cnst14-
2)+((cnst17)/2)*(cnst14*cnst14-1))" ; Dosytimecubed

;Assuming half-sine gradient pulses [most common on Bruker systems]
"cnst18=0.000001*p30*2*0.000001*p30*2*(d20 - (2*0.000001*p30*(5-3*cnst14*cnst14)/16) -
(cnst17*(1-cnst14*cnst14)/2) )" ; Dosytimecubed

"cnst15=1+cnst14"  ; 1 + alpha
"cnst16=1-cnst14"  ; 1 - alpha

"p2=p1*2"

"DELTA1=d20-4.0*p1-4.0*p30-5.0*d16-p19"
"DELTA2=p40/2"
"DELTA3=p45"

"cnst50=cnst52/4.75"
"cnst51=cnst52/sqrt(2)"
"p33=1000000.0/(cnst50*4)"
"p34=1000000.0/(cnst51*4)"
"p35=1000000.0/(cnst52*4)"
"cnst33= (p33/p1)*(p33/p1)"
"cnst34= (p34/p1)*(p34/p1)"
"cnst35= (p35/p1)*(p35/p1)"
"spw40=plw1/cnst33"
"spw41=plw1/cnst35"
"spw42=plw1/cnst33"
"spw43=plw1/cnst33"
"spw44=plw1/cnst34"
"spw45=plw1/cnst35"

"d11=30m"
"d12=20u"

;The below line should be commented out for Topspin < 2.0
;"acqt0 = -p1*2/3.1416"

1 ze
  d11 QNP_X
  d11 pl12:f2
```

```
2 30m do:f2
  d1                          ;
  d12 SWITO_F
  p19:gp2*-1.0                ;Spoiler gradient balancing pulse
  d16
  4u pl0:f1

  (p40:sp40 ph1):f1
  DELTA2
  p30:gp1*diff*cnst16                ;1 - alpha
  d16
  (p41:sp41 ph2):f1
  p30:gp1*diff*-1*cnst15             ;1 + alpha
  d16
  DELTA2
  (p42:sp42 ph3):f1

  p30:gp1*diff*2*cnst14              ;Lock refocusing pulse pulse
  d16

  p19:gp2                     ;Spoiler gradient balancing pulse
  d16

  DELTA1

  p30:gp1*diff*2*cnst14              ;Lock refocusing pulse pulse
  d16

  (p43:sp43 ph4):f1
  p30:gp1*diff*cnst16                ; 1 - alpha
  d16
  4u
  (p44:sp44 ph5):f1
  DELTA3
  p30:gp1*diff*-1*cnst15             ;1 + alpha
  d16

  p31:gp3
  d16
 (p45:sp45 ph6):f1
  p31:gp3
  d16

  ACQ_START(ph30,ph31) (2u SWITO_H)
  aq DWELL_GEN:f1 cpd2:f2
  rcyc=2

  30m do:f2 mc #0 to 2 F1QF(igrad diff)
exit


ph1 = 0 0 0 0 0 0 0 0
ph2 = 0 0 0 0 1 1 1 1
ph3 = 0 0 0 0 0 0 0 0
ph4 = 0 0 0 0 0 0 0 0
ph5 = 0 0 1 1 0 0 1 1
ph6 = 0 1 0 1 0 1 0 1
ph30= 0
ph31= 0 2 2 0 2 0 0 2


;pl0 : zero power
;pl1 : high power
;p1  : 90 degree high power pulse
;p19          gradient pulse 2 (spoil gradient)
;p30          gradient pulse (little DELTA*0.5)
;p40 : duration of 1st 90-degree chirp pulse
;p41 : duration of composite 180-degree chirp pulse
;p42 : duration of 2nd 90-degree chirp pulse
;p43 : duration of 3rd 90-degree chirp pulse
;p44 : duration of 1st 180-degree chirp pulse
;p45 : duration of 2nd 180-degree chirp pulse
;d1           relaxation delay; 1-5 * T1
;d16          delay for gradient recovery
```

```
;d20            diffusion time (big DELTA)
;spw40 : RF power of 1st 90-degree chirp pulse
;spw41 : RF power of composite 180-degree chirp pulse
;spw42 : RF power of 2nd 90-degree chirp pulse
;spw43 : RF power of 3rd 90-degree chirp pulse
;spw44 : RF power of 1st 180-degree chirp pulse
;spw45 : RF power of 2nd 180-degree chirp pulse
;spnam40: file name for 1st 90-degree chirp pulse
;spnam41: file name for composite 180-degree chirp pulse
;spnam42: file name for 2nd 90-degree chirp pulse
;spnam43: file name for 3rd 90-degree chirp pulse
;spnam44: file name for 1st 180-degree chirp pulse
;spnam45: file name 2nd 180-degree chirp pulse
;gpz1: diffusion encoding gradient (100%)
;gpz2: spoil gradient
;gpz3: CTP gradient
;gpnam1: SINE.100
;gpnam2: SINE.100
;gpnam3: SINE.100

;cnst14         alpha, typically 0.2
;cnst15         1+alpha
;cnst16         1-alpha
;cnst17         Dtau
;cnst18         dosytimecubed
;cnst52: RF amplitude for 180-degree chirp pulses (Hz)
;NS             1 * n
;DS             1 * m
;td1            number of experiments
;FnMODE         QF

; use,  gradient value  gpz1: 100  and  gpz7 : 100

;use AU-program dosy to calculate gradient ramp-file Difframp
;         use xf2 and DOSY processing
;         use "setdiffparmUoM" if 'setdiffparm' does not work
;       or  use "setdiffparm STEbp" but this gives slighly distorted diffusion
coefficients
```