

Convection in liquid-state NMR: expect the unexpected

Thaís M. Barbosa, Roberto Rittner, Cláudio F. Tormena, Gareth A. Morris and Mathias Nilsson*

Supporting Information

Table of contents:

- **Figure S1:** Representative examples of fitted experimental data over a wide range of temperatures and convection rates.
- **Table S1:** Values of decremented and incremented delays, d_5 and d_3 , used in the convection experiments.
- **Table S2:** Nominal values of gradient (G/cm) for the five spectrometers used in the convection velocity measurements.
- **Figure S2:** Flow chart of the main steps in determining convection velocities.
- Pulse program code (*IS_test_convexp_list*) for convection velocity measurements in Bruker spectrometers.
- TopSpin AU program (*convection*) used in combination with the above pulse sequence.
- TopSpin AU program (*PHC0*) for automatic phase correction.
- TopSpin AU program (*multi_pp2*) for automatic peak picking.

In the main text we demonstrate the convection velocity behavior in a range of spectrometers, probes and NMR tube geometries. The Figure S1 shows representative examples (400 MHz, BBI probe using a 5 mm NMR tube) of fitted experimental data over a wide range of temperatures (5, 9, 13, 17, 21, 25, 29, 33 and 37 °C) and convection rates.

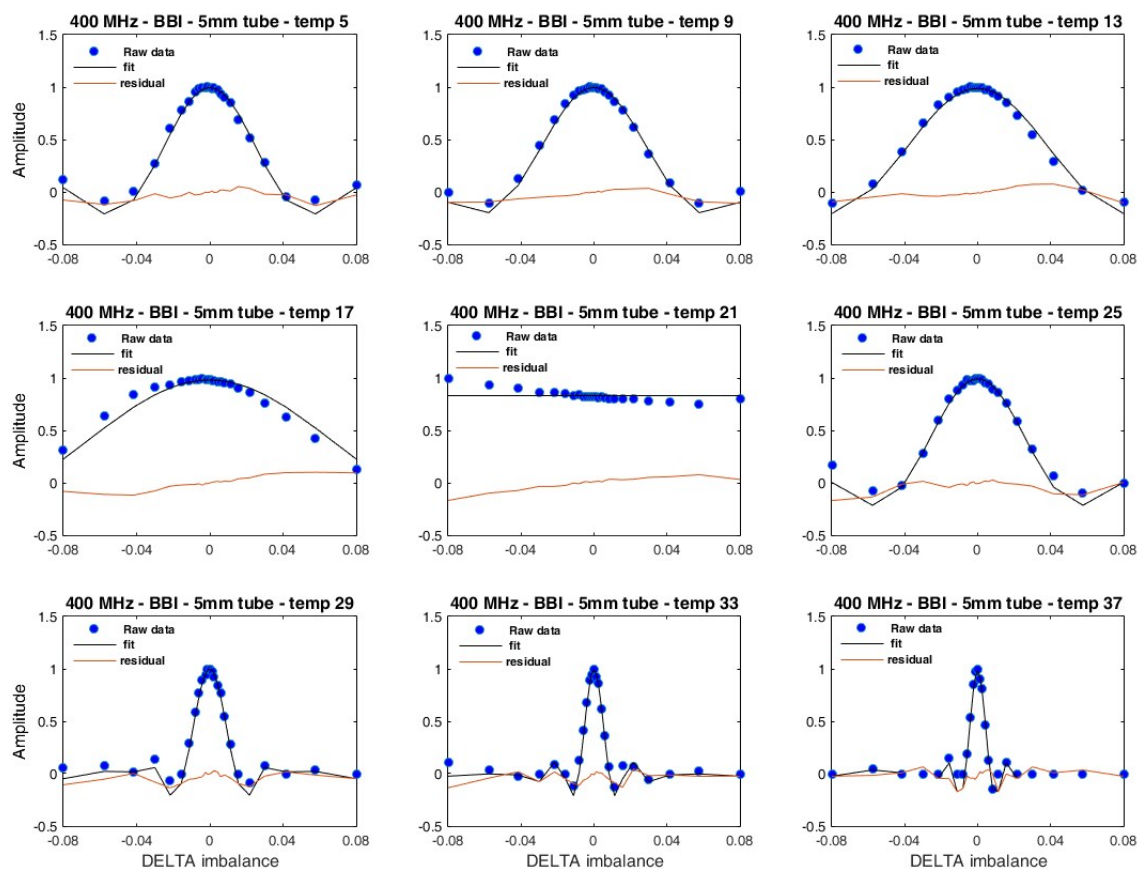


Figure S1: Experimental data points (blue circle), fit (black line), and residuals (orange line) for signal amplitude *versus* diffusion delay imbalance, at temperatures of 5, 9, 13, 17, 21, 25, 29, 33 and 37 °C, for velocity-mapping experiments performed using a standard thin-walled 5 mm NMR tube in a 400 MHz Bruker AVANCE III spectrometer with a BBI probe.

Table S1: Values of decremented and incremented delays d_5 and d_3 in the attached pulse sequence used in the convection experiments, and the corresponding diffusion delay imbalances $\Delta\Delta$.

d_5	d_3	$\Delta\Delta$
0.0435	0.0035	-0.08
0.037925	0.009075	-0.0577
0.033904	0.01309595	-0.0416162
0.031004	0.015996075	-0.0300157
0.028912	0.0180878	-0.0216488
0.027404	0.01959645	-0.0156142
0.026315	0.02068455	-0.0112618
0.025531	0.021469363	-0.00812255
0.024963	0.022036618	-0.00585353
0.024533	0.022467153	-0.00413139
0.024065	0.02293486	-0.00226056
0.023827	0.023173498	-0.00130601
0.0235	0.0235	0
0.023173	0.023826503	0.00130601
0.022935	0.02406514	0.00226056
0.022467	0.024532848	0.00413139
0.022037	0.024963383	0.00585353
0.021469	0.025530638	0.00812255
0.020685	0.02631545	0.0112618
0.019596	0.02740355	0.0156142
0.018088	0.0289122	0.0216488
0.015996	0.031003925	0.0300157
0.013096	0.03390405	0.0416162
0.009075	0.037925	0.0577
0.0035	0.0435	0.08

The total delay imbalance $\Delta\Delta$ is twice the difference between $d3$ and $d5$:

$$\Delta\Delta = 2(d3 - d5) \quad (S1)$$

Table S2: Nominal values of gradient (G/cm) for the five spectrometers used.

Spectrometer	Gradient value
Bruker AVANCE III 400 MHz	55
Bruker AVANCE III 500 MHz	50.1
Bruker Ascend AVANCE III 600 MHz	50
Bruker Ascend 500 MHz	65.7
Bruker Ascend AVANCE III 800 MHz	68.5

The pulse sequence *IS_test_convexp_list* was used to perform the convection experiments in combination with the (attached) macro (AU programme) entitled *convection*. The macro sets up a series of experiments independently incrementing temperature and diffusion delay imbalance. Typically, a series of 900 1D spectra is obtained, which are subsequently processed in TopSpin (apodisation, Fourier transformation, baseline and phase correction). The macro PHC0 was then used to apply a $\Delta\Delta$ -dependent zero-order phase correction to correct for field and lock disturbances caused by the gradient pulses, and the macro pp2 used for automatic peak picking. The amplitudes were then imported to Matlab for further processing and fitting (equation 1, Fig. S2) to determine the maximum convection velocities.

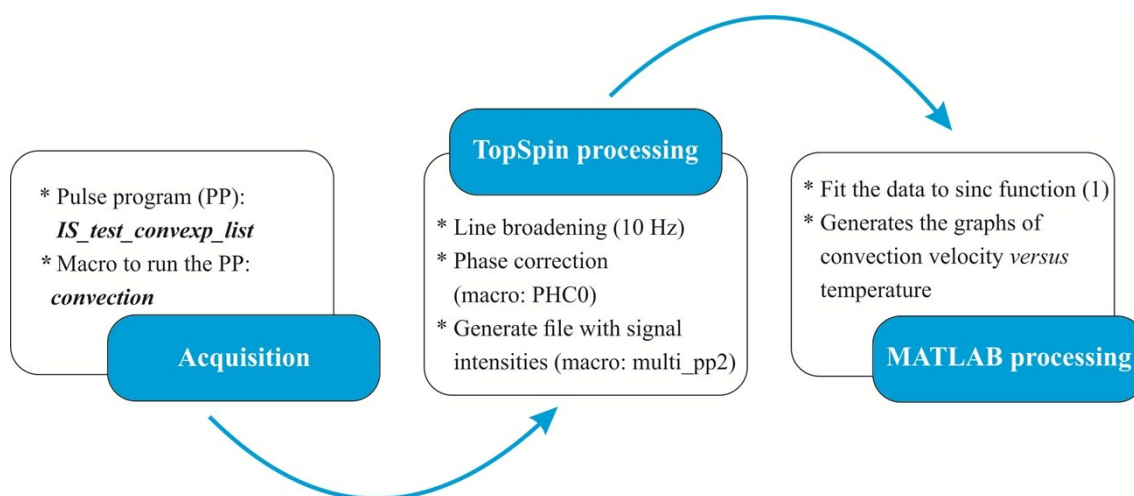


Figure S2: Flow chart of the main steps in determining convection velocities.

Pulse Program: *IS_test_convexp_list*

```
; test sequence – based on Dhom2djidosy sequence for convection compensation,
excluding 2dj and dosy.
; use with macro “convection” to increment temperature and delays
; d1 – recycle time delay (1 second for doped CHCl3)
; delay3 - incremented convection/diffusion delay
; d4 - gradient stabilisation delay
; delay5 - decremented convection/diffusion delay
; p1 - 90 deg RF duration ; ph1 phase
; p2 - 180 deg RF duration ; ph2 phase
; p16 - diffusion/convection encoding gradient duration ; gp1 amplitude
; delta imbalance ( $\Delta\Delta$ ) is twice the difference between d3 and d5:  $\Delta\Delta = 2(d3 - d5)$ 
```

```
;$OWNER=nmrsu
#include <Avance.incl>
#include <Grad.incl>
#include <Delay.incl>
```

```
"d0=3u"
```

```
;define list<delay> delay3=<ccdinc>
;define list<delay> delay5=<ccddec>
```

```
"d3=cnst3*1s"
```

```
"d5=cnst5*1s"
```

```
"p2=p1*2.0"
```

```
"d4=0.5m"
```

```
1 ze
```

```
2 d1
```

```
10u UNBLKGRAD
```

```
p1 ph1
```

```
d4
```

```
p16:gp1
```

```
d4
```

```
d3
```

```
p16:gp1*-1.0
```

```
d4
```

```
p16:gp1*-1.0
```

```
d4
```

```
d5
```

```
p16:gp1
```

d4
d4

3 p2 ph2

d4

d4

p16:gp1

d4

d5

p16:gp1*-1.0

d4

p16:gp1*-1.0

d4

d3

p16:gp1

d4

4 go=2 ph31

d1 mc #0 to 2 F0(zd)

exit

ph1=0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3

ph2=0 2 1 3 1 3 2 0 1 3 2 0 2 0 3 1

ph29=0

ph31=0 0 2 2 1 1 3 3

➤ **TOPSPIN AU program : convection**

```
/* Convection Test - use with sequence 'IS_test_convexp',  
 * Varies first del2 (diffusion delay imbalance) then temperature from 278.2 to 313.2 K at 1  
K intervals;
```

```
/* Increment Experiment number */  
int expno_incr;
```

```
/*find vd lists */  
/* My test */
```

```
char ccddecfile[PATH_MAX];  
char ccdincfile[PATH_MAX];
```

```
int numdels=25; /*number of delays in list */  
int ntemps=36; /*number of temperatures to record */  
float ndegree=1; /*number of degrees to increment between temperatures */  
float startdegree=278.2; /*starting temperature in Kelvin */
```

```
(void)sprintf(ccddecfile,"%s/stan/nmr/lists/vd/user/ccddecOrig", PathXWinNMRExp());
```

```
FILE *file = fopen(ccddecfile,"r");  
float ccddeclist[numdels];  
int i=0;  
float num;  
while(fscanf(file, "%f", &num) > 0)  
{  
    ccddeclist[i] = num;  
    i++;  
Proc_err(DEF_ERR_OPT,"import %f",ccddeclist[i-1]);  
}  
fclose(file);
```

```
(void)sprintf(ccdincfile,"%s/stan/nmr/lists/vd/user/ccdincOrig", PathXWinNMRExp());  
file = fopen(ccdincfile,"r");  
float ccdinclist[numdels];  
i=0;  
num;  
while(fscanf(file, "%f", &num) > 0)  
{  
    ccdinclist[i] = num;  
    i++;  
Proc_err(DEF_ERR_OPT,"import %f",ccdinclist[i-1]);  
}  
fclose(file);  
f1=startdegree;
```

```
TIMES(ntemps)  
STOREPAR("TE",f1);  
TESET
```

```
TEREADY(900, 0.1)
  i1=0;
  /* run experiment with the delays defined in the lists*/
  TIMES2(numdels)
    GETCURDATA
    expno_incr=expno+1;
    WRPA(name,expno_incr,procno,disk,user);
    IEXPNO
    f2=ccdinclist[i1];
    STOREPAR("CNST 3",f2)
    f2=ccddeclist[i1];
    STOREPAR("CNST 5",f2)
    Proc_err(INFO_OPT,"del value = %f + %f s",ccdinclist[i1],ccddeclist[i1]);
    i1=i1 + 1;
    ZG
    EFP
  END
  f1=f1+ndegree;
END
QUIT
```


➤ **TOPSPIN AU program : *PHC0***

```

/*****
/*  multi_PH          22.04.2015                                     */
/*****
/*  Short Description :AU program for automatic phase correction of a set of 1D spectra. */
/*****
/*  Keywords : phase correction for convection experiments          */
/*****
/*  Description/Usage : AU program for automatic phase correction of a set of 1D spectra.
The zero order correction is dependent of a preset (f1) correction factor      */
/*****
/*  Author(s) :                                                 */
/*  Name       :      Thaís Mendonça Barbosa and Mike Engelhardt  */
/*  Organisation :      UNICAMP and Bruker BioSpin GmbH          */
/*  Email      : thais.unicamp@hotmail.com and michael.engelhardt@bruker.de */
/*****
/*  Name       Date      Modification:                            */
/*  TMB        22 April 2015 created                               */
/*****

/*GETINT ("Enter PHC0:", f1)*/
f1=-70.8;
/*PHC0 for the central spectrum (thirteenth spectrum) of each set of 25 experiments */
static const double PhaseInc[25]=
{-2.2,-0.8,-1.4,-1.0,-1.4,-1.4,-0.4,-1.2,0.4,-1.8,-0.2,-0.6,0.0,1.2,0.2,-1.4,-1.2,-3.2,0.0,-2.4,-
1.2,-1.0,-0.2,-1.4,0.0 };
/* INSERT THE 25 Δ of PHC0 separate by commas */
TIMES(25);
/*The whole thing happens 25 times*/
    STOREPAR("PHC0", f1 + PhaseInc[loopcount1]);
    EFP
    IEXPNO
    f1 + PhaseInc[loopcount1];
END
QUIT
```

➤ **TOPSPIN AU program : *multi_pp2***

```

/*****
/*      multi_pp          23.04.2015          */
/*****
/*      Short Description : AU program for automatic peak picking  of a series of 1D spectra.
(based on multi_decon)          */
/*****
/*      Keywords : peak picking of a series of data sets          */
/*****
/*      Description/Usage : AU program for automatic peak picking of a series of 1D. The
spectra can either be stored in a series of EXPNOs or in a series of PROCNOs.  */
/*      The AU program works as follows :          */
/*      1. Switch to the first experiment to be deconvoluted.          */
/*      2. Define the peak picking parameters with the "pp" command          */
/*      3. Start the AU program.          */
/*      The individual deconvolution results are stored in the file peak.txt in each
dataset, the summary of all results is stored in the file pp-all.txt in the first dataset.  */

/*****
/*      Author(s) :          */
/*      Name      : Mathias Nilsson          */
/*      Organisation : University of Manchester          */
/*      Email     : mathias.nilsson@manchester.ac.uk          */
/*****
/*      Name      Date   Modification:          */
/*      MN        23 April 2015 created (converted from multi_decon)          */
/*      MN        28 April 2015 changed output file structure and copies more pp
parameters from the first experiment          */
/*****

```

```

/* $Id: multi_decon,v 1.14 2007/12/05 13:43:20 gsc Exp $*/
char printname[256];
char decresult[256], decallresult[256], dummysr[256];
int iii, kk, psign, useprocnos=0;
FILE *fpnt, *fpnt2;
double f1p, f2p;
float mi, maxi, pc, cy;

```

GETCURDATA

GETINT ("Use EXPNOs (0) or PROCNOs (1) :",useprocnos)

```

i1 = 1;
if (useprocnos)
{
  GETINT ("Enter first processed number :",i1)
  DATASET(name,expno,i1,disk,user)
}

```

```

else
{
  GETINT ("Enter first experiment number :",i1)
  DATASET(name,i1,procno,disk,user)
}
i2 = 16;
GETINT ("Enter number of experiments :",i2)

ERRORABORT
FETCHPAR("CY",&cy)
FETCHPAR("F1P",&f1p)
FETCHPAR("F2P",&f2p)
FETCHPAR("MI",&mi)
FETCHPAR("MAXI",&maxi)
FETCHPAR("PC",&pc)
FETCHPAR("PSIGN",&psign)
FETCHPAR("CURPRIN",&printname)
STOREPAR("CURPRIN","peak.txt")

PP
XCMD("sendgui convertpeaklist txt")
STOREPAR("CURPRIN",printname)

strcpy (decreresult,PROCPATH("peak.txt"));
fpnt=fopen(decreresult, "r");
if(fpnt==NULL)
{
  Proc_err(DEF_ERR_OPT, "No pp result file :\n%s\n"
           "Maybe CURPRIN is not a filename ?", decreresult);
  ABORT
}

strcpy (decallresult,PROCPATH("pp-all.txt"));
fpnt2=fopen(decallresult, "wt");
if(fpnt2==NULL)
{
  Proc_err(DEF_ERR_OPT, "No all-pp result file :\n%s",
           decallresult);
  ABORT
}
/*fprintf(fpnt2,"%f %d %s n\n",mi, mi, psign);*/
fprintf(fpnt2,"Peak picking of %d experiments \n\n",i2);

fprintf(fpnt2,"Full details of first experiment: \n");

while (fgets(dummystr, sizeof(dummystr), fpnt) != NULL)
  fputs (dummystr, fpnt2);

fprintf(fpnt2,"\n\n");

```

```

fclose (fpnt);

fpnt=fopen(decresult, "r");
if(fpnt==NULL)
{
  Proc_err(DEF_ERR_OPT, "No pp result file :\n%s\n"
           "Maybe CURPRIN is not a filename ?", decresult);
  ABORT
}
/* Discard first four lines of peak.txt result file */
for (iii = 0; iii < 4; iii++)
  fgets(dummystr, sizeof(dummystr), fpnt);

fprintf(fpnt2, "Peaks for each experiment:\n");

fprintf(fpnt2, "EXPNO/PROCNO %d \n", i1);

  while (fgets(dummystr, sizeof(dummystr), fpnt) != NULL)
    fputs (dummystr, fpnt2);
  fclose (fpnt);

kk=1;

TIMES(i2-1)
if (useprocnos)
{
  IPROCNO
}
else
{
  IEXPNO
}

STOREPAR("CY", -1.0)
STOREPAR("F1P", f1p)
STOREPAR("F2P", f2p)
STOREPAR("MI", mi)
STOREPAR("MAXI", maxi)
STOREPAR("PSIGN", psign)
STOREPAR("PC", pc)
FETCHPAR("CURPRIN", &printname)
STOREPAR("CURPRIN", "peak.txt")

PP
XCMD("sendgui convertpeaklist txt")
STOREPAR("CURPRIN", printname)
strcpy (decresult, PROCPATH("peak.txt"));
fprintf(fpnt2, "EXPNO/PROCNO %d \n", i1+kk);
kk++;
fpnt=fopen(decresult, "r");

```

```

if(fpnt==NULL)
{
  Proc_err(DEF_ERR_OPT, "No peak picking result file :\n%s\nAssuming no peaks in
spectrum",
                                                decresult);
}

else
{
  /* Discard first four lines of peak.txt result file */
  for (iii = 0; iii < 4; iii++)
    fgets(dummysr, sizeof(dummysr), fpnt);

  while (fgets(dummysr, sizeof(dummysr), fpnt) != NULL)
    fputs (dummysr, fpnt2);
}
fclose (fpnt);
END

fclose (fpnt2);

(void) sprintf (text,"--- multi_pp finished ---\n\
All results summarized in :\n%s",decallresult);
QUITMSG (text)

/* Observation : To process 19F data set it is necessary to add FETCHPAR
("PSCAL",&pscal) line and a STOREPAR ("PSCAL", pscal) line */

```