Supporting Information

Longitudinal relaxation optimized amide ^{1}H-CEST experiments for studying slow chemical exchange processes in fully protonated proteins

Tairan Yuwen[1] and Lewis E. Kay[1,2]

[1]Departments of Molecular Genetics, Biochemistry and Chemistry, The University of Toronto, Toronto, Ontario, Canada, M5S 1A8

[2]Hospital for Sick Children, Program in Molecular Structure and Function, 555 University Avenue, Toronto, Ontario, Canada, M5G 1X8

Table of Contents

Derivation of Equations for I^{CEST}

In the discussion that follows we calculate the magnitude of the minor-state CEST dip as a function of T_{Ex}, that is given by the difference in the intensities of the ground state peak when the weak B_1 field is applied far off resonance of ground and excited state peaks of interest and when the B_1 field is applied on resonance with the excited state peak. We consider a two-state exchange reaction, $G \underset{k_{EG}}{\overset{k_{GE}}{\rightleftharpoons}} E$, as described in the text. A complete description of the evolution of the ground state magnetization is given by the Bloch–McConnell equations (McConnell 1958) but these can be simplified considerably when $\Delta\varpi_{GE} \rightarrow \infty$ and $\omega_1 \gg k_{GE}$ where ω_1 is the strength of the weak B_1 field, as described below.

Derivation of Eqs 2 and 3 of the Text

A. ^{15}N-,^{13}C-CEST

Scheme (i): Longitudinal magnetization recovers to its equilibrium value during the CEST period. When the weak B_1 field is far off resonance of ground and excited state peaks, the time evolution of the ground state X longitudinal magnetization during the CEST element is given by

$$\frac{dX_z^G}{dt} = -\rho_X(X_z^G - X_o^G) - k_{GE}X_z^G + k_{EG}X_z^E \qquad [S1]$$

where X_o is the equilibrium X magnetization and the superscripts G and E distinguish between ground and excited state magnetization. In what follows we assume that at the start of the CEST element $X_z^G/X_z^E \approx p_G/(1-p_G)$ or that the ratio M_z^G/M_z^E rapidly recovers to the equilibrium value. Under these conditions Eq [S1] becomes

$$\frac{dX_z^G}{dt} = -\rho_X(X_z^G - X_o^G) \qquad [S2]$$

and

$$X_z^G(T_{Ex}) = [(\kappa - 1)e^{-\rho_X T_{Ex}} + 1]X_o \qquad [S3]$$

with $\kappa = \{1 - \exp(-\rho_H d_1)\}|\gamma_H/\gamma_X|$ that takes into account recovery of proton magnetization during the delay between scans, d_1, and the gain in polarization from the transfer of magnetization from ^{1}H to X prior to the CEST element of duration T_{Ex}. When the weak B_1 field is applied on resonance with the excited state peak the evolution of X_z^G is given by

$$\frac{dX_z^G}{dt} = -\rho_X(X_z^G - X_o^G) - k_{GE}X_z^G \qquad [S4]$$

Note that the condition $\omega_1 \gg k_{EG}$ effectively eliminates contributions from the excited state (proportional to k_{EG}) because the time average of magnetization transferred from E to G is 0. The solution to Eq [S4] is

$$X_z^G(T_{Ex}) = \left\{ \left[\kappa - \frac{\rho_X}{\rho_X + k_{GE}} \right] e^{-(\rho_X + k_{GE})T_{Ex}} + \frac{\rho_X}{\rho_X + k_{GE}} \right\} X_o \qquad [S5]$$

The size of the minor state dip I^{CEST} is given by the difference between X_z^G values in Eqs [S3] and [S5] that produces Eq 2 of the main text.

Scheme (ii): Longitudinal magnetization decays to 0 during the CEST period. Here we consider the situation (which is used in practice for X-CEST (Bouvignies et al. 2014, Vallurupalli et al. 2012) where longitudinal magnetization is inverted in successive scans, along with the phase of the receiver, so that immediately before the CEST element the magnetization of interest is given by $\pm X_z$. Eqs [S2] and [S4] can be modified to take this 2-step phase cycle into account directly by replacing X_o by 0 (Freeman and Hill 1971, Sklenář et al. 1987, Vallurupalli et al. 2012). The solutions are,

$$X_z(T_{Ex}) = \kappa e^{-\rho_X T_{Ex}} X_o \qquad [S6]$$

and

$$X_z(T_{Ex}) = \kappa e^{-(\rho_X + k_{GE})T_{Ex}} X_o \qquad [S7]$$

respectively. The difference between X_z^G values is Eq 3 of the text.

B. L-optimized ^{1}H-CEST

A rigorous mathematical description of ^{1}H relaxation during ^{1}H-CEST is complicated by the many different dipolar relaxation pathways that are operative. Here we use a simple model that we have discussed previously (Yuwen et al. 2017) that describes the evolution of longitudinal ^{1}H magnetization by assuming that the proton spin, I, of interest is dipolar coupled to a large number, N, of spins S^i. Neglecting exchange for the moment we can write

$$\begin{aligned} \frac{dI_z^G}{dt} &= -\rho_I(I_z^G - I_o^G) - \sigma_{SI}\sum_{i=1}^{N}(S_z^i - S_o^i) \\ \frac{dS_z^i}{dt} &= -\rho_S(S_z^i - S_o^i) - \sigma_{IS}(I_z^G - I_o^G) \end{aligned} \qquad [S8]$$

where ρ and $\sigma_{SI} = \sigma_{IS}$ are auto- and cross-relaxation rates (Cavanagh et al. 2007) and all S^i spins are assumed to have the same ρ value. Summing over all N S^i spins yields

$$\begin{aligned} \frac{dI_z^G}{dt} &= -\rho_I(I_z^G - I_o^G) - \sigma_{SI}(S_z^{SS} - S_o^{SS}) \\ \frac{dS_z^{SS}}{dt} &= -\rho_S(S_z^{SS} - S_o^{SS}) - N\sigma_{IS}(I_z^G - I_o^G) \end{aligned} \qquad [S9]$$

where $S_z^{SS} = \sum_{i=1}^{N} S_z^i$. Because N is large we assume that any perturbation to spin I will not affect spin S^{SS} so that to a good approximation $(S_z^{SS} - S_o^{SS})/S_o^{SS} \approx 0$ for the duration of the CEST period. This assumption follows directly from the fact that the L-optimization approach (Pervushin et al. 2002) is used so that the non-amide protons (denoted by S^{SS} in the present discussion) are maintained at their equilibrium values throughout the experiment. Identical expressions for the time evolution of I_z are obtained as for X_z above and hence the same expressions for I^{CEST} hold. As discussed in the text, it is important to note that ρ_I is

large (certainly much larger than ρ_X) due to cross-relaxation with neighboring protons so that any perturbation to spin I will be accompanied by rapid equilibration that significantly decreases the intensity of the minor state CEST dip. This is especially the case when spin I is inverted in one of the scans for scheme (ii) that is particularly detrimental to sensitivity. Second κ is much smaller than for ^{15}N or ^{13}C that leads to large differences in the relative magnitudes of I^{CEST} in ^{1}H-CEST profiles recorded for the two approaches considered, $\frac{\Delta I^{CEST}}{I^{CEST}(\text{scheme ii})}$ (see Eq 4 of the main text and related discussion as well as Figure 5C).

Derivation of I^{CEST} for non-L-optimized ^{1}H-CEST

In what follows we consider only scheme (i) since Figure 5C of the main text establishes that this is the preferable approach for recording ^{1}H-CEST profiles. Our starting point is Eq [S9], noting that all longitudinal magnetization is 0 at the start of the t_2 evolution period. In what follows we neglect magnetization recovery during t_2 (although this could be accounted for easily) and consider the case where all longitudinal magnetization is 0 at the start of T_{Ex}. Since N is large (see above) we further assume that S_z^{SS} recovers to its equilibrium value in a manner that is unaffected by the magnetization of spin I. The recovery of S_z^{SS} (Eq [S9]) is given by

$$S_z^{SS}(T_{Ex}) = (1 - e^{-\rho_S T_{Ex}})S_o^{SS} \qquad \text{[S10]}$$

For the case where the weak B_1 field is far from the resonance positions of ground and excited state peaks and with the same assumptions as the previous derivations above the evolution of I_z^G is

$$\begin{aligned} \frac{dI_z^G}{dt} &= -\rho_I(I_z^G - I_o^G) - \frac{\sigma}{N}(S_z^{SS} - S_o^{SS}) \\ &= -\rho_I(I_z^G - I_o^G) + \sigma e^{-\rho_S t} I_o^G \end{aligned} \qquad \text{[S11]}$$

where $\sigma_{IS} = \sigma_{SI} = \sigma/N$. Eq [S11] is solved to yield

$$I_z^G(T_{Ex}) = \left\{ \left[-1 + \frac{\sigma}{\rho_S - \rho_I} - \frac{\sigma}{\rho_S - \rho_I} e^{-(\rho_S - \rho_I)T_{Ex}} \right] e^{-\rho_I T_{Ex}} + 1 \right\} I_o^G \qquad [S12]$$

When the B_1 field is applied on-resonance with the excited state peak,

$$\frac{dI_z^G}{dt} = -\rho_I (I_z^G - I_o^G) + \sigma e^{-\rho_S t} I_o^G - k_{GE} I_z^G \qquad [S13]$$

and the solution is

$$I_z^G(T_{Ex}) = \left\{ \left[-\frac{\rho_I}{\rho_I + k_{GE}} + \frac{\sigma}{\rho_S - \rho_I - k_{GE}} - \frac{\sigma}{\rho_S - \rho_I - k_{GE}} e^{-(\rho_S - \rho_I - k_{GE})T_{Ex}} \right] e^{-(\rho_I + k_{GE})T_{Ex}} + \frac{\rho_I}{\rho_I + k_{GE}} \right\} I_o^G \qquad [S14]$$

Relaxation rates ρ_I and ρ_S can be recast as $\rho_I = \rho_I^{self} + \rho_I^{cross}$, $\rho_S = \rho_S^{self} + \rho_S^{cross}$, where ρ^{self} and ρ^{cross} are contributions that do not arise (self) and that arise directly (cross) from *I*–*S* cross-relaxation; it follows that $\rho_I^{cross} \approx -\sigma$ and $\rho_S^{cross} \approx -\sigma/N$ in the macromolecular limit. For $\rho_S \approx \rho_S^{self} \approx \rho_I^{self}$, that is assumed in the simulations in the main text (i.e., large *N*), one can write $\rho_S = \rho_I + \sigma$ and Eqs [S12] and [S14] become,

$$I_z^G(T_{Ex}) = \left\{ -e^{-(\rho_I + \sigma)T_{Ex}} + 1 \right\} I_o^G \qquad [S15]$$

$$I_z^G(T_{Ex}) = \left\{ \left[-\frac{\rho_I}{\rho_I + k_{GE}} + \frac{\sigma}{\sigma - k_{GE}} - \frac{\sigma}{\sigma - k_{GE}} e^{-(\sigma - k_{GE})T_{Ex}} \right] e^{-(\rho_I + k_{GE})T_{Ex}} + \frac{\rho_I}{\rho_I + k_{GE}} \right\} I_o^G \qquad [S16]$$

from which it follows that

$$I_z^{CEST}(T_{Ex}) = \left\{ \left[\frac{\rho_I}{\rho_I + k_{GE}} - \frac{\sigma}{\sigma - k_{GE}} \right] e^{-(\rho_I + k_{GE})T_{Ex}} + \frac{k_{GE}}{\sigma - k_{GE}} e^{-(\rho_I + \sigma)T_{Ex}} + \frac{k_{GE}}{\rho_I + k_{GE}} \right\} I_o^G \qquad [S17]$$

By analogy with Eq 5 of the main text the optimal T_{Ex} value for the non-L-optimized ^{1}H-CEST can be derived from Eq [S17], satisfying

$$[2(\rho_H+k_{GE})T_{Ex}+1]\left[\frac{\rho_H}{\rho_H+k_{GE}}-\frac{\sigma}{\sigma-k_{GE}}\right]e^{-(\rho_H+k_{GE})T_{Ex}}+[2(\rho_H+k_{GE})T_{Ex}+1]\frac{k_{GE}}{\sigma-k_{GE}}e^{-(\rho_H+\sigma)T_{Ex}}+\frac{k_{GE}}{\rho_H+k_{GE}}=0 \quad [S18]$$

where ρ_H has been substituted for ρ_I.

References

G. Bouvignies, P. Vallurupalli, and L. E. Kay. Visualizing side chains of invisible protein conformers by solution NMR. *J. Mol. Biol.*, 426(3):763–774, 2014. doi: 10.1016/j.jmb.2013.10.041.

J. Cavanagh, W. J. Fairbrother, A. G. Palmer, M. Rance, and N. Skelton. *Protein NMR Spectroscopy*. Elsevier Academic Press, London, 2 edition, 2007.

R. Freeman and H. D. W. Hill. Fourier transform study of NMR spin–lattice relaxation by "progressive saturation". *J. Chem. Phys.*, 54(8):3367–3377, 1971. doi: 10.1063/1.1675352.

H. M. McConnell. Reaction rates by nuclear magnetic resonance. *J. Chem. Phys.*, 28(3):430–431, 1958. doi: 10.1063/1.1744152.

K. Pervushin, B. Vögeli, and A. Eletsky. Longitudinal ^{1}H relaxation optimization in TROSY NMR spectroscopy. *J. Am. Chem. Soc.*, 124(43):12898–12902, 2002. doi: 10.1021/ja027149q.

V. Sklenář, D. Torchia, and A. Bax. Measurement of carbon-13 longitudinal relaxation using ^{1}H detection. *J. Magn. Reson.*, 73(2):375 – 379, 1987. doi: 10.1016/0022-2364(87)90214-9.

P. Vallurupalli, G. Bouvignies, and L. E. Kay. Studying "invisible" excited protein states in slow exchange with a major state conformation. *J. Am. Chem. Soc.*, 134(19):8148–8161, 2012. doi: 10.1021/ja3001419.

T. Yuwen, A. Sekhar, and L. E. Kay. Separating dipolar and chemical exchange magnetization transfer processes in ^{1}H-CEST. *Angew. Chem. Int. Ed.*, pages n/a–n/a, 2017. doi: 10.1002/anie.201610759.

```
#!/bin/csh

# Define variables
@ nfq = 157
@ ni = 64
@ yT = $ni * $nfq * 2
@ yN = $yT * 2
set arr = `seq 1 $nfq`

# FID format conversion
bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 1664 -dspfvs 20 -grpdly 67.9841613769531  \
  -xN              1536  -yN               $yN  \
  -xT               768  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW        12019.231  -ySW         2433.100  \
  -xOBS         800.301  -yOBS          81.103  \
  -xCAR           4.664  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 1 0 -time -axis Y    \
| nmrPipe -out ./test_1.fid -verb -ov

bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 1664 -dspfvs 20 -grpdly 67.9841613769531  \
  -xN              1536  -yN               $yN  \
  -xT               768  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW        12019.231  -ySW         2433.100  \
  -xOBS         800.301  -yOBS          81.103  \
  -xCAR           4.664  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 0 1 -time -axis Y    \
| nmrPipe -out ./test_2.fid -verb -ov

# Process series of 2D spectra
pseudo3D.com -in test_1.fid -outDir fid_1 -tau $arr -time -inner -grad
pseudo3D.com -in test_2.fid -outDir fid_2 -tau $arr -time -inner -grad

foreach i (1 2)
xyz2pipe -in ./fid_$i/test%03d.fid \
| nmrPipe -fn POLY -time -ord 2 -verb                    \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 2 -auto -verb                       \
| nmrPipe -fn FT -verb                                   \
| nmrPipe -fn PS -p0 15.6 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                   \
| nmrPipe -fn LP -ord 16 -verb                          \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 3 -auto -verb                       \
| nmrPipe -fn FT -verb                                   \
| nmrPipe -fn PS -p0 -8.4 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                   \
| nmrPipe -fn POLY -auto -verb                          \
| nmrPipe -fn EXT -x1 11.0ppm -xn 5.5ppm -sw -verb      \
| pipe2xyz -ov -verb -out ./ft_$i/test%03d.ft2
end

# Fit relative peak intensities in the 2D series
peakHN.tcl -in ./ft_1/test001.ft2 -out ./test.tab -hi 2e5
autofit.tcl -specName ./ft_1/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_1.t
ab
autofit.tcl -specName ./ft_2/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_2.t
ab

# Extract autofit fitting results
getTabCol.tcl -in ./nlin_1.tab -var INDEX "Z*" > nlin_edit_1.tab
getTabCol.tcl -in ./nlin_2.tab -var INDEX "Z*" > nlin_edit_2.tab
```

```
#!/bin/csh

# Define variables
@ nfq = 102
@ ni = 64
@ yT = $ni * $nfq * 2
@ yN = $yT * 2
set arr = `seq 1 $nfq`

# FID format conversion
bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 2080 -dspfvs 20 -grpdly 67.9842071533203  \
  -xN              1280  -yN               $yN  \
  -xT               614  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW         9615.385  -ySW         1216.601  \
  -xOBS         600.253  -yOBS          60.830  \
  -xCAR           4.694  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 1 1 -time -axis Y    \
| nmrPipe -out ./test_1.fid -verb -ov

bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 2080 -dspfvs 20 -grpdly 67.9842071533203  \
  -xN              1280  -yN               $yN  \
  -xT               614  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW         9615.385  -ySW         1216.601  \
  -xOBS         600.253  -yOBS          60.830  \
  -xCAR           4.694  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 1 -1 -time -axis Y   \
| nmrPipe -out ./test_2.fid -verb -ov

# Process series of 2D spectra
pseudo3D.com -in test_1.fid -outDir fid_1 -tau $arr -time -inner -grad
pseudo3D.com -in test_2.fid -outDir fid_2 -tau $arr -time -inner -grad

xyz2pipe -in ./fid_1/test%03d.fid \
| nmrPipe -fn POLY -time -ord 2                          \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 2 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 190.6 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn LP -ord 16 -verb                            \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 1.0 -verb  \
| nmrPipe -fn ZF -zf 3 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di -verb            \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn POLY -auto -verb                            \
| nmrPipe -fn EXT -x1 10.5ppm -xn 6.0ppm -sw -verb        \
| pipe2xyz -ov -verb -out ./ft_1/test%03d.ft2

# Shift the second series by 1JHN to overlap with the first series
xyz2pipe -in ./fid_2/test%03d.fid \
| nmrPipe -fn POLY -time -ord 2                          \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 2 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 190.6 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn LP -ord 16 -verb                            \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 1.0 -verb  \
| nmrPipe -fn ZF -zf 3 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
```

```
| nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di -verb            \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn POLY -auto -verb                            \
| nmrPipe -fn CS -rs 92.8Hz -verb                         \
| nmrPipe -fn EXT -x1 10.5ppm -xn 6.0ppm -sw -verb        \
| pipe2xyz -ov -verb -out ./ft_2/test%03d.ft2

# Fit relative peak intensities in the 2D series
peakHN.tcl -in ./ft_1/test001.ft2 -out ./test.tab -hi 3e5
autofit.tcl -specName ./ft_1/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_1.t
ab
autofit.tcl -specName ./ft_2/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_2.t
ab

# Extract autofit fitting results
getTabCol.tcl -in ./nlin_1.tab -var INDEX "Z*" > nlin_edit_1.tab
getTabCol.tcl -in ./nlin_2.tab -var INDEX "Z*" > nlin_edit_2.tab
```

```
#!/bin/csh

# Define variables
@ nfq = 102
@ ni = 64
@ yT = $ni * $nfq * 2
@ yN = $yT * 2
set arr = `seq 1 $nfq`

# FID format conversion
bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 2080 -dspfvs 20 -grpdly 67.9842071533203  \
  -xN              1280  -yN               $yN  \
  -xT               614  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW         9615.385  -ySW         1216.601  \
  -xOBS         600.253  -yOBS          60.830  \
  -xCAR           4.694  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 1 1 -time -axis Y    \
| nmrPipe -out ./test_1.fid -verb -ov

bruk2pipe -in ./ser \
  -bad 0.0 -aswap -AMX -decim 2080 -dspfvs 20 -grpdly 67.9842071533203  \
  -xN              1280  -yN               $yN  \
  -xT               614  -yT               $yT  \
  -xMODE            DQD  -yMODE        Complex  \
  -xSW         9615.385  -ySW         1216.601  \
  -xOBS         600.253  -yOBS          60.830  \
  -xCAR           4.694  -yCAR         119.000  \
  -xLAB              HN  -yLAB             15N  \
  -ndim               2  -aq2D          States  \
| nmrPipe -fn COADD -cList 1 -1 -time -axis Y   \
| nmrPipe -out ./test_2.fid -verb -ov

# Process series of 2D spectra
pseudo3D.com -in test_1.fid -outDir fid_1 -tau $arr -time -inner -grad
pseudo3D.com -in test_2.fid -outDir fid_2 -tau $arr -time -inner -grad

xyz2pipe -in ./fid_1/test%03d.fid \
| nmrPipe -fn POLY -time -ord 2                          \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 2 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 145.6 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn LP -ord 16 -verb                            \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 1.0 -verb  \
| nmrPipe -fn ZF -zf 3 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di -verb            \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn POLY -auto -verb                            \
| nmrPipe -fn EXT -x1 10.5ppm -xn 6.0ppm -sw -verb        \
| pipe2xyz -ov -verb -out ./ft_1/test%03d.ft2

# Shift the second series by 1JHN to overlap with the first series
# Direct dimension p0 should differ by 90 degree in the two series
xyz2pipe -in ./fid_2/test%03d.fid \
| nmrPipe -fn POLY -time -ord 2                          \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 0.5 -verb  \
| nmrPipe -fn ZF -zf 2 -auto -verb                        \
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 235.6 -p1 0.0 -di -verb              \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn LP -ord 16 -verb                            \
| nmrPipe -fn SP -off 0.45 -end 0.98 -pow 2 -c 1.0 -verb  \
| nmrPipe -fn ZF -zf 3 -auto -verb                        \
```

```
| nmrPipe -fn FT -verb                                    \
| nmrPipe -fn PS -p0 -90.0 -p1 180.0 -di -verb            \
| nmrPipe -fn TP -verb                                    \
| nmrPipe -fn POLY -auto -verb                            \
| nmrPipe -fn CS -rs 92.8Hz -verb                         \
| nmrPipe -fn EXT -x1 10.5ppm -xn 6.0ppm -sw -verb        \
| pipe2xyz -ov -verb -out ./ft_2/test%03d.ft2

# Fit relative peak intensities in the 2D series
peakHN.tcl -in ./ft_1/test001.ft2 -out ./test.tab -hi 3e5
autofit.tcl -specName ./ft_1/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_1.t
ab
autofit.tcl -specName ./ft_2/test%03d.ft2 -series -inTab ./test.tab -outTab ./nlin_2.t
ab

# Extract autofit fitting results
getTabCol.tcl -in ./nlin_1.tab -var INDEX "Z*" > nlin_edit_1.tab
getTabCol.tcl -in ./nlin_2.tab -var INDEX "Z*" > nlin_edit_2.tab
```

```
/* b_1H_CEST_Iz_TRsss_lek_800_cpz

    This pulse sequence will allow one to perform the following experiment:
  1H CEST when magnetization is of the form Iz with 15N-1H HSQC readout.
  Takes the difference between 1H Z magnetization up and down

  TROSY spin state selective transfer:
    ph1=0: I(TR) ---> N(TR) ---> I(TR)
    ph1=2: I(AT) ---> N(TR) ---> I(TR)

  Pervusion TROSY based hsqc including a flag for gradient coherence transfer selectio
n

  Note: Identical trosy selection can be achieved by inverting the phase of the first
1H 90 after t1
        along with the final 15N 90 and then inverting the first 2 or last 2 receiver
phases
        Addition of coherence transfer gradients does not change the cycle

  If gd_sel is used then the min phase cycle is 2
  If gd_sel is not used then the min phase cycle is 4

  To callibrate B1 field: set -Dcal_HB1
    - calibrate by setting 1H decoupling field to 0
    - sit on one of the 1H multiplet components and evolution will be of the other com
ponent
    - calibration is done with magnetization Iz at the very beginning

  Used with N15cest_make_fq_list.c (ccode) which makes the frequency list
  Used with N15cest_cal_B1.c (ccode)
  Used with move_lek_cest_fq which moves the freq list to f1
  Used with move_lek_cest_B1 which moves the list of times to vd list
*/

;prosol relations=<triple>

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

/************************/
/*   Define phases     */
/************************/
#define zero ph=0.0
#define one ph=90.0
#define two ph=180.0
#define three ph=270.0

/*************************/
/*    Define pulses      */
/*************************/
define pulse dly_pg1     /* Messerle purge pulse */
  "dly_pg1=5m"
define pulse dly_pg2     /* Messerle purge pulse */
  "dly_pg2=dly_pg1/1.62"
define pulse pwh
  "pwh=p1"               /* 1H hard pulse at power level pl1 (tpwr) */
define pulse pwn
  "pwn=p3"               /* PW90 for N pulse at power level pl3 (dhpwr2)  */

#ifdef c_flg
  define pulse pwc_ad
    "pwc_ad=p22"         /* PW180 for adiabatic pulse at pl22 (d_ad)   */
#endif

#ifdef N_sel
  define pulse pwn_sl
    "pwn_sl=p32"
#endif
```

```
/***********************/
/*    Define lists     */
/***********************/
#ifdef cal_HB1
  define list<delay> time_cal=<t1list_calB1>
#else
#ifdef cal_HR1
  define list<delay> time_cal=<t1list_calR1>
#else
  define list<frequency> H_offset=<$FQ1LIST>
#endif
#endif

define list<gradient> EA8 = { 0.6379 0.9621 }

define loopcounter Nphase
  "Nphase=1"

/*****************************************************/
/* Assign cnsts to check validity of parameter range */
/*****************************************************/
#ifdef fsat
  "cnst10=plw10"          /* tsatpwr - set max at 0.00005W */
#endif

#ifdef mess_flg
  "cnst11=plw11"          /* tpwrmess - set max at 2.0W */
#endif

#ifdef c_flg
  "cnst22=spw22"          /* d_ad - set to 100 W */
#endif

#ifdef N_sel
  "cnst32=spw32"          /* power level for 15N selective pulse */
#endif

/*************************************************/
/* Set the weak H-amide B1 field for irradiation */
/*************************************************/
"p15=1000000.0/(4.0*cnst16)"
"plw15=plw1*(pwh)*(pwh)/((p15)*(p15))"
"cnst15=plw15"

/*****************************************************/
/* Set phase alignment and offset for shaped pulses  */
/*****************************************************/
"spoal6=0.5"
"spoal7=0.5"
"spoal8=0.5"
"spoal9=1"
"spoal19=0"

"spoff6=0"
"spoff7=bf1*(cnst5/1000000)-o1"
"spoff8=bf1*(cnst5/1000000)-o1"
"spoff9=bf1*(cnst5/1000000)-o1"
"spoff19=bf1*(cnst5/1000000)-o1"

/*****************************************/
/* Calculate parameters for shaped pulses */
/*****************************************/
#ifdef calc_SP
  /*  BIP 720,50,20.1 (p6, sp6)   */
  "p6=120000.0/bf1"   /*  BIP pulse length   */
  "spw6=plw1*(pow((p1*2.0/p6)/0.25,2))"   /* BIP power level  */

  /*  PC9 (p7, sp7)   */
```

```
  "p7=7.2/(cnst6*bf1/1000000)"   /*  PC9  pulse length  */
  "spw7=plw1*(pow((p1/p7)/0.125,2))"   /* PC9  power level  */

  /*  REBURP (p8, sp8)   */
  "p8=4.875/(cnst6*bf1/1000000)"   /* REBURP pulse length  */
  "spw8=plw1*(pow((p1*2.0/p8)/0.07981,2))"   /* REBURP power level  */

  /*  EBURP & EBURP_TR  (p9, sp9, sp19)   */
  "p9=4.6/(cnst6*bf1/1000000)"   /*  EBURP pulse length   */
  "spw9=plw1*(pow((p1/p9)/0.06103,2))"   /* EBURP power level  */
  "spw19=plw1*(pow((p1/p9)/0.06103,2))"   /* EBURP_TR power level  */
#endif

/*************************************/
/*    Define shaped pulses on 1H     */
/*************************************/
define pulse pw_bip
  "pw_bip=p6"             /* PW180 for BIP pulses */
define pulse pw_pc9
  "pw_pc9=p7"             /* PW90 for PC9 pulses */
define pulse pw_reburp
  "pw_reburp=p8"          /* PW180 for REBURP pulses */
define pulse pw_eburp
  "pw_eburp=p9"           /* PW90 for EBURP pulses */
define pulse pw_eburptr
  "pw_eburptr=p9"         /* PW90 for EBURP_TR pulses */

/***************************/
/*    Define delays       */
/***************************/
define delay hscuba       /* length of 1/2 scuba delay */
  "hscuba=30m"
define delay taua         /* 1 / 4J(XH)                */
  "taua=d3"
define delay taub         /* 1 / 4J(XH)                */
  "taub=larger(d5,2u+p56+d16+pw_reburp*0.5+pw_eburp*0.67+0.2u)"

#ifdef gd_sel
  define delay BigT
    "BigT=larger(d15,2u+p55+d16+4u+de+0.2u)"
#endif /*gd_sel*/

#if !defined(cal_HB1) && !defined(cal_HR1)
  define delay time_T1
    "time_T1=d17"         /* CEST duration - typically 100-500 ms */
#endif

"d11=30m"
"in0=inf1/2"              /* t1/2 increment */

/*****************************************************************/
/* f1180: Start t1 at half dwell to get -90/180 phase correction */
/*        in F1 (15N) dim - set zgoptns -Df1180                  */
/*****************************************************************/
#ifdef f1180
  "d0=(in0)/2"
#else
  "d0=0.1u"
#endif

/****************************/
/* Initialize loop counters */
/****************************/
"l2=0"
"l3=0"

"acqt0=0"
;baseopt_echo
```

```
1 ze
/*****************************************/
/*   Check Validity of Parameter Range   */
/*****************************************/
#ifdef fsat
  if "cnst10 > 0.00005"
  {
    2u
    print "error:tsatpwr pl10 too large !!!  "
    goto HaltAcqu
  }
#endif

#ifdef mess_flg
  if "cnst11 > 2.0"
  {
    2u
    print "error:tpwrmess pl11 too large !!!  "
    goto HaltAcqu
  }
#endif

  if "cnst15 > 0.01"
  {
    2u
    print "error: cnst15 - power in W for 1H CEST is  too large !!!  "
    goto HaltAcqu
  }

  if "cnst16 > 250"
  {
    2u
    print "error: cnst16 - weak 1H B1 field strength in Hz must be <= 250 Hz !!!  "
    goto HaltAcqu
  }

#ifdef c_flg
  if "cnst22 > 45.0"
  {
    2u
    print "error:don't fry the probe, spw22 dpwr too large!  "
    goto HaltAcqu
  }
#endif

#ifdef N_sel
  if "cnst32 > 35"
  {
    2u
    print "error: spw32 power is too high "
    goto HaltAcqu
  }
#endif

#if !defined(cal_HB1) && !defined(cal_HR1)
  if "time_T1 > 0.5s"
  {
    2u
    print "error: time_T1 is too long"
    goto HaltAcqu
  }
#endif

2 d11
/*****************************************/
/* Continue to check real time variables */
/*****************************************/
#ifdef cal_HB1
  "DELTA = time_cal[l2]"
```

```
  if "DELTA > 500m" {
    2u
    print "error: time_cal (evolution time during B1 calib is too long < 500 m"
    goto  HaltAcqu
  }
#endif

/*************************************************/
/*   Presaturation Period                        */
/*   option for Messrlie purge zgoptn -Dmess_flg  */
/*************************************************/
#ifdef mess_flg
  20u pl11:f1
  (dly_pg1 ph26):f1
  20u
  (dly_pg2 ph27):f1
  20u pl10:f1
#endif /*mess_flg*/

#ifdef fsat
  4u pl10:f1
  d1 cw:f1 ph26
  4u do:f1
#ifdef fscuba
  hscuba
  (pwh ph26):f1
  (pwh*2 ph27):f1
  (pwh ph26):f1
  hscuba
#endif /*fscuba*/
#else
  d1
#endif /*fsat*/

  20u UNBLKGRAD

/***************************************/
/* Erase 15N equilibrium magnetization */
/***************************************/
  2u pl3:f3
  (pwn ph26):f3

  2u
  p51:gp1
  d16

/*************************************************/
/*  +/- phase cycle on 1H prior to CEST period    */
/*************************************************/
#ifdef Hsteady_flg
  2u fq=0:f1
  (pw_eburp:sp9 ph26):f1
  4u
  (pw_eburptr:sp19 ph0):f1

  2u
  p51:gp1
  d16
#endif /*Hsteady_flg*/

/***************************************/
/*   CEST portion on Hz magnetization   */
/***************************************/
  2u pl15:f1
#ifdef cal_HB1
  "DELTA=time_cal[l2]"
  2u fq=cnst1:f1
  DELTA cw:f1 ph26
  2u do:f1
```

```
#else
#ifdef cal_HR1
  "DELTA=time_cal[l2]"
  DELTA
#else
  2u fq=H_offset:f1
  time_T1 cw:f1 ph26
  2u do:f1
#endif /*cal_HR1*/
#endif /*cal_HB1*/
  20u fq=0:f1

  2u
  p52:gp2
  d16

/*******************************************/
/*   TROSY-type H-N transfer:              */
/*   l3=0: Hx(TR) ---> Nx(TR) ---> Hx(TR)  */
/*   l3=1: Hx(AT) ---> Nx(TR) ---> Hx(TR)  */
/*******************************************/
  if "l3%2 == 0" {
    (ralign (pw_bip:sp6 ph26):f1 (pwn ph26):f3)
  }
  else {
    (ralign (pw_bip:sp6 ph26):f1 (pwn ph28):f3)
  }

  2u
  p53:gp3
  d16

  "DELTA = taua - 2u - p53 - d16"
  DELTA

  (center (pw_bip:sp6 ph26):f1 (pwn*2 ph26):f3)

  2u
  p53:gp3
  d16

  DELTA

  (pwn ph27):f3
  (pw_pc9:sp7 ph26):f1

#ifdef N_sel
  "DELTA = taua - pw_pc9*0.5 - larger(pw_reburp,pwn_sl)*0.5"
#else
  "DELTA = taua - pw_pc9*0.5 - pw_reburp*0.5"
#endif
  DELTA

#ifdef N_sel
  (center (pw_reburp:sp8 ph26):f1 (pwn_sl:sp32 ph26):f3)
#else
  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)
#endif

  DELTA

  (pw_pc9:sp7 ph27):f1

  2u
  p54:gp4
  d16

/*********************/
/*   15N labeling    */
```

```
/**********************/
#ifndef Hsteady_flg
  if "Nphase %2 == 1" {
    (pwn ph1):f3
  }
  else {
    (pwn ph2):f3
  }
#else
  if "Nphase %2 == 1" {
    (pwn ph11):f3
  }
  else {
    (pwn ph12):f3
  }
#endif

#ifdef gd_sel
  2u
  p55:gp5*-1.0
  d16

  "DELTA = BigT - 2u - p55 - d16"
  DELTA
#endif /*gd_sel*/

#ifdef c_flg
  if "d0 - pwc_ad*0.5 > 2u"
  {
    "DELTA = d0 - pwc_ad*0.5"
    DELTA
    (pwc_ad:sp22 ph26):f2
    DELTA
  }
  else
  {
    d0
    d0
  }
#else
  d0
  d0
#endif /*c_flg*/

/*************************************/
/*   TROSY-type N->H back transfer   */
/*************************************/
  (pw_eburptr:sp19 ph3):f1

  2u
  p56:gp6
  d16

  "DELTA = taub - 2u - p56 - d16 - pw_reburp*0.5 - pw_eburp*0.67"
  DELTA

  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph4):f3)

  DELTA

  2u
  p56:gp6
  d16

  (pw_eburp:sp9 ph26):f1

#ifdef gd_sel
  2u
  p55:gp5*1.0
```

```
  d16

  "DELTA = BigT - 2u - p55 - d16"
  DELTA
#endif /*gd_sel*/

  (pwn ph5):f3

  2u
  p57:gp7
  d16

  "DELTA = taub - 2u - p57 - d16 - pw_reburp*0.5"
  DELTA

  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)

#ifdef gd_sel
  "DELTA = taub - 2u - p57 - d16 - pw_reburp*0.5"
#else
  "DELTA = taub - 2u - p57 - d16 - pw_reburp*0.5 - 4u - de"
#endif /*gd_sel*/
  DELTA

  2u
  p57:gp7
  d16

#ifdef gd_sel
  (pwn ph26):f3

  2u
  p55:gp8*EA8
  d16

  "DELTA = BigT - 2u - p55 - d16 - 4u - de"
  DELTA

  4u BLKGRAD
#else
  4u BLKGRAD
  (pwn ph26):f3
#endif /*gd_sel*/

/***************************************/
/*   Test how much water left          */
/*   set d13=AQ and zgoptn -Dwater     */
/*   also use small value of rg        */
/***************************************/
#ifdef water
  if "nsdone == 0" {
    d13
    d1 pl1:f1
    (pwh*0.1 ph26):f1
  }
  else {
    d13
  }
#endif

/***************************************/
/*   Signal detection and looping      */
/***************************************/
  go=2 ph31
  d11 mc #0 to 2

  F3QF(calclc(l3,1))
#if defined(cal_HB1) || defined(cal_HR1)
  F2QF(calclc(l2,1))
```

```
#else
  F2QF(calclist(H_offset,1))
#endif /*cal_HB1, cal_HR1*/
  F1EA(calgrad(EA8) & calph(ph3, +180) & calph(ph5, +180) & calph(ph31, +180) & calclc
(Nphase, 1), caldel(d0, +in0) & calph(ph1, +180) & calph(ph2, +180) & calph(ph11, +180
) & calph(ph12, +180) & calph(ph31, +180))

HaltAcqu, 1m
exit

ph0=2 0
ph1=1 3 2 0
ph2=1 3 0 2
ph3=1
ph4=0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3
ph5=1
ph11=1 1 2 2
ph12=1 1 0 0
ph26=0
ph27=1
ph28=2
ph29=3
ph31=1 3 0 2 3 1 2 0

;d1 : Repetition delay d1
;d3 : taua ~1/4J(NH) 2.70 ms
;d5 : taub ~1/4J(NH) 2.70 ms
;d11 : delay for disk i/o, 30ms
;d13 : set to acq time for testing water preservation
;d15 : BigT set to 500 us
;d16 : gradient recovery delay, 200us
;d17 : time_T1, time for CEST, typically several 100s of ms
;pl1 : tpwr - power level for 1H pulses
;pl3 : dhpwr2 - power level for 15N pulse pwn
;pl10 : tsatpwr - power level for water presat
;pl11 : tpwrmess - power level for Messerle purge
;pl21 : dpwr - power level for  13C decoupling cpd2
;pl31 : dpwr2 - power level for 15N cpd3
;p1 : pwh
;p3 : pwn
;p6 : f1 channel - 180 degree shaped pulse for inversion
;                    Bip720,50,20.1            (200us at 600.13 MHz)
;p7 : f1 channel -  90 degree shaped pulse for excitation
;                    Pc9_4_90.1000             (3.0ms at 600.13 MHz)
;p8 : f1 channel - 180 degree shaped pulse for refocusing
;                    Reburp.1000               (2.0ms at 600.13 MHz)
;p9 : f1 channel -  90 degree shaped pulse for excitation
;                    Eburp2.1000/Eburp2tr.1000 (1.92ms at 600.13 MHz)
;p22 : pwc_ad
;p32 : pwn_sl for selective 15N pulse on amides
;p51 : gradient pulse 51                                [400 usec]
;p52 : gradient pulse 52                                [1000 usec]
;p53 : gradient pulse 53                                [1000 usec]
;p54 : gradient pulse 54                                [1500 usec]
;p55 : gradient pulse 55                                [256 usec]
;p56 : gradient pulse 56                                [256 usec]
;p57 : gradient pulse 57                                [256 usec]
;spnam6 : Bip720,50,20.1
;spnam7 : Pc9_4_90.1000
;spnam8 : Reburp.1000
;spnam9 : Eburp2.1000
;spnam19 : Eburp2tr.1000
;spnam22 : shape for pwc_ad
;spnam32 : shape for pwn_sl
;cnst1 : offset (Hz) for B1 calibration from o1
;cnst5 : H(N) chemical shift (offset, in ppm)
;cnst6 : H(N) excitation band width (in ppm)
;cnst16 : weak CEST 1H B1 field in Hz
;l2 : pointer to delay value for calibration of B1
```

```
;l3 : pointer to switch between TROSY transfer pathways
;inf1 : 1/SW(X) = 2*DW(X)
;in0 : 1/(2*SW(x))=DW(X)
;nd0 : 2
;ns : 2*n
;FnMODE : echo-antiecho

;use gradient ratio:    gp 5 : gp 8
;                         40 :   50

;for z-only gradients:
;gpz1: -25%
;gpz2: 15%
;gpz3: -60%
;gpz4: -70%
;gpz5: 40%
;gpz6: 60%
;gpz7: 15%
;gpz8: 50%

;use gradient files:
;gpnam1: SMSQ10.32
;gpnam2: SMSQ10.32
;gpnam3: SMSQ10.32
;gpnam4: SMSQ10.32
;gpnam5: SMSQ10.32
;gpnam6: SMSQ10.32
;gpnam7: SMSQ10.32
;gpnam8: SMSQ10.32

;zgoptns : Dc_flg, Dfsat, Dfscuba, Dmess_flg, Df1180, DN_sel, Dcal_HB1, Dcal_HR1, Dgd_
sel, Dcalc_SP, DHsteady_flg
```

```
/* b_1H_CEST_Iz_noTRsss_lek_800_cpz

  Used for measurement of amide 1H CEST in small proteins so dont use TROSY

  This pulse sequence will allow one to perform the following experiment:
  1H CEST when magnetization is of the form Iz with 15N-1H HSQC readout.
  Takes the difference between 1H Z magnetization up and down

  Spin state selective transfer:
    l3=0: I(TR/AT) ---> N(TR/AT) ---> I(TR/AT) ---> IP
    l3=1: I(TR/AT) ---> N(TR/AT) ---> I(TR/AT) ---> AP

  To callibrate B1 field: set -Dcal_HB1

  Min phase cycle is 2

  Used with N15cest_make_fq_list.c (ccode) which makes the frequency list
  Used with N15cest_cal_B1.c (ccode)
  Used with move_lek_cest_fq which moves the freq list to f1
  Used with move_lek_cest_B1 which moves the list of times to vd list

  s3_flg is used for the alternative shorter version of IPAP scheme
*/

;prosol relations=<triple>

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

/***********************/
/*   Define phases     */
/***********************/
#define zero ph=0.0
#define one ph=90.0
#define two ph=180.0
#define three ph=270.0

/*************************/
/*    Define pulses      */
/*************************/
define pulse dly_pg1     /* Messerle purge pulse */
  "dly_pg1=5m"
define pulse dly_pg2     /* Messerle purge pulse */
  "dly_pg2=dly_pg1/1.62"
define pulse pwh
  "pwh=p1"               /* 1H hard pulse at power level pl1 (tpwr) */
define pulse pwn
  "pwn=p3"               /* PW90 for N pulse at power level pl3 (dhpwr2)  */

#ifdef c_flg
  define pulse pwc_ad
    "pwc_ad=p22"         /* PW180 for adiabatic pulse at pl22 (d_ad)   */
#endif

#ifdef N_sel
  define pulse pwn_sl
    "pwn_sl=p32"
#endif

/***********************/
/*    Define lists     */
/***********************/
#ifdef cal_HB1
  define list<delay> time_cal=<t1list_calB1>
#else
#ifdef cal_HR1
  define list<delay> time_cal=<t1list_calR1>
#else
```

```
  define list<frequency> H_offset=<$FQ1LIST>
#endif
#endif

/*****************************************************/
/* Assign cnsts to check validity of parameter range */
/*****************************************************/
#ifdef fsat
  "cnst10=plw10"         /* tsatpwr - set max at 0.00005W */
#endif

#ifdef mess_flg
  "cnst11=plw11"         /* tpwrmess - set max at 2.0W */
#endif

#ifdef c_flg
  "cnst22=spw22"         /* d_ad - set to 100 W */
#endif

#ifdef N_sel
  "cnst32=spw32"       /* power level for 15N selective pulse */
#endif

/*************************************************/
/* Set the weak H-amide B1 field for irradiation */
/*************************************************/
"p15=1000000.0/(4.0*cnst16)"
"plw15=plw1*(pwh)*(pwh)/((p15)*(p15))"
"cnst15=plw15"

/*******************************************************/
/* Set phase alignment and offset for shaped pulses  */
/*******************************************************/
"spoal6=0.5"
"spoal7=0.5"
"spoal8=0.5"
"spoal9=1"
"spoal19=0"

"spoff6=0"
"spoff7=bf1*(cnst5/1000000)-o1"
"spoff8=bf1*(cnst5/1000000)-o1"
"spoff9=bf1*(cnst5/1000000)-o1"
"spoff19=bf1*(cnst5/1000000)-o1"

/*****************************************/
/* Calculate parameters for shaped pulses */
/*****************************************/
#ifdef calc_SP
  /*  BIP 720,50,20.1 (p6, sp6)   */
  "p6=120000.0/bf1"   /*  BIP pulse length   */
  "spw6=plw1*(pow((p1*2.0/p6)/0.25,2))"   /* BIP power level  */

  /*  PC9 (p7, sp7)   */
  "p7=7.2/(cnst6*bf1/1000000)"   /*  PC9  pulse length  */
  "spw7=plw1*(pow((p1/p7)/0.125,2))"   /* PC9  power level  */

  /*  REBURP (p8, sp8)   */
  "p8=4.875/(cnst6*bf1/1000000)"   /* REBURP pulse length  */
  "spw8=plw1*(pow((p1*2.0/p8)/0.07981,2))"   /* REBURP power level  */

  /*  EBURP & EBURP_TR  (p9, sp9, sp19)   */
  "p9=4.6/(cnst6*bf1/1000000)"   /*  EBURP pulse length   */
  "spw9=plw1*(pow((p1/p9)/0.06103,2))"   /* EBURP power level  */
  "spw19=plw1*(pow((p1/p9)/0.06103,2))"   /* EBURP_TR power level  */
#endif

/*************************************/
/*    Define shaped pulses on 1H     */
```

```
/*************************************/
define pulse pw_bip
  "pw_bip=p6"             /* PW180 for BIP pulses */
define pulse pw_pc9
  "pw_pc9=p7"             /* PW90 for PC9 pulses */
define pulse pw_reburp
  "pw_reburp=p8"          /* PW180 for REBURP pulses */
define pulse pw_eburp
  "pw_eburp=p9"           /* PW90 for EBURP pulses */
define pulse pw_eburptr
  "pw_eburptr=p9"         /* PW90 for EBURP_TR pulses */

/*************************/
/*    Define delays       */
/*************************/
define delay hscuba       /* length of 1/2 scuba delay */
  "hscuba=30m"
define delay taua         /* 1 / 4J(XH)                */
  "taua=d3"
define delay taub         /* 1 / 4J(XH)                */
  "taub=larger(d5,2u+p57+d16+pw_reburp*0.5+pw_eburp*0.67+0.2u)"

define delay tauc         /* 1 / 4J(XH) or 1 / 8J(XH)  */
#ifndef s3_flg
  "tauc=2.7m"
#else
#ifndef cal_HB1
  "tauc=1.39m"
#else
  "tauc=5.4m"
#endif /*cal_HB1*/
#endif /*s3_flg*/

#if !defined(cal_HB1) && !defined(cal_HR1)
  define delay time_T1
    "time_T1=d17"         /* CEST duration - typically 100-500 ms */
#endif

"d11=30m"
"in0=inf1/2"              /* t1/2 increment */

/*****************************************************************/
/* f1180: Start t1 at half dwell to get -90/180 phase correction */
/*        in F1 (15N) dim - set zgoptns -Df1180                  */
/*****************************************************************/
#ifdef f1180
  "d0=(in0)/2"
#else
  "d0=0.1u"
#endif

/*****************************/
/* Initialize loop counters */
/*****************************/
"l2=0"
"l3=0"

"acqt0=0"
;baseopt_echo

1 ze
/*****************************************/
/*   Check Validity of Parameter Range   */
/*****************************************/
#ifdef fsat
  if "cnst10 > 0.00005"
  {
    2u
    print "error:tsatpwr pl10 too large !!!  "
```

```
    goto HaltAcqu
  }
#endif

#ifdef mess_flg
  if "cnst11 > 2.0"
  {
    2u
    print "error:tpwrmess pl11 too large !!!  "
    goto HaltAcqu
  }
#endif

  if "cnst15 > 0.01"
  {
    2u
    print "error: cnst15 - power in W for 1H CEST is  too large !!!  "
    goto HaltAcqu
  }

  if "cnst16 > 250"
  {
    2u
    print "error: cnst16 - weak 1H B1 field strength in Hz must be <= 250 Hz !!!  "
    goto HaltAcqu
  }

#ifdef c_flg
  if "cnst22 > 45.0"
  {
    2u
    print "error:don't fry the probe, spw22 dpwr too large!  "
    goto HaltAcqu
  }
#endif

#ifdef N_sel
  if "cnst32 > 35"
  {
    2u
    print "error: spw32 power is too high "
    goto HaltAcqu
  }
#endif

#if !defined(cal_HB1) && !defined(cal_HR1)
  if "time_T1 > 0.5s"
  {
    2u
    print "error: time_T1 is too long"
    goto HaltAcqu
  }
#endif

#ifdef gd_sel
  if "tauc - 2u - p58 - d16 - 2u - p59 - d16 - pwn - pw_reburp*0.5 < 0.2u"
  {
    2u
    print "error: tauc is too short "
    goto HaltAcqu
  }
#else
  if "tauc - 2u - p59 - d16 - 4u - de - pwn*2 - pw_reburp*0.5 < 0.2u"
  {
    2u
    print "error: tauc is too short "
    goto HaltAcqu
  }
#endif /*gd_sel*/
```

```
2 d11
/*****************************************/
/* Continue to check real time variables */
/*****************************************/
#ifdef cal_HB1
  "DELTA = time_cal[l2]"
  if "DELTA > 500m" {
    2u
    print "error: time_cal (evolution time during B1 calib is too long < 500 m"
    goto  HaltAcqu
  }
#endif

/*************************************************/
/*   Presaturation Period                        */
/*   option for Messrlie purge zgoptn -Dmess_flg  */
/*************************************************/
#ifdef mess_flg
  20u pl11:f1
  (dly_pg1 ph26):f1
  20u
  (dly_pg2 ph27):f1
  20u pl10:f1
#endif

#ifdef fsat
  4u pl10:f1
  d1 cw:f1 ph26
  4u do:f1
#ifdef fscuba
  hscuba
  (pwh ph26):f1
  (pwh*2 ph27):f1
  (pwh ph26):f1
  hscuba
#endif /*fscuba*/
#else
  d1
#endif /*fsat*/

  20u UNBLKGRAD

/*****************************************/
/* Erase 15N equilibrium magnetization */
/*****************************************/
  2u pl3:f3
  (pwn ph26):f3

  2u
  p51:gp1
  d16

/*************************************************/
/*  +/- phase cycle on 1H prior to CEST period    */
/*************************************************/
#ifdef Hsteady_flg
  2u fq=0:f1
  (pw_eburp:sp9 ph26):f1
  4u
  (pw_eburptr:sp19 ph0):f1

  2u
  p51:gp1
  d16
#endif /*Hsteady_flg*/

/*****************************************/
/*   CEST portion on Hz magnetization   */
```

```
/*****************************************/
  2u pl15:f1
#ifdef cal_HB1
  "DELTA=time_cal[l2]"
  2u fq=cnst1:f1
  DELTA cw:f1 ph26
  2u do:f1
#else
#ifdef cal_HR1
  "DELTA=time_cal[l2]"
  DELTA
#else
  2u fq=H_offset:f1
  time_T1 cw:f1 ph26
  2u do:f1
#endif /*cal_HR1*/
#endif /*cal_HB1*/
  20u fq=0:f1

  2u
  p52:gp2
  d16

/*********************************/
/*   TROSY-type H->N transfer    */
/*********************************/
  (ralign (pw_bip:sp6 ph26):f1 (pwn ph26):f3)

  2u
  p53:gp3
  d16

  "DELTA = taua - 2u - p53 - d16"
  DELTA

  (center (pw_bip:sp6 ph26):f1 (pwn*2 ph26):f3)

  2u
  p53:gp3
  d16

  DELTA

  (pwn ph27):f3
  (pw_pc9:sp7 ph26):f1

#ifdef N_sel
  "DELTA = taua - pw_pc9*0.5 - larger(pw_reburp,pwn_sl)*0.5"
#else
  "DELTA = taua - pw_pc9*0.5 - pw_reburp*0.5"
#endif
  DELTA

#ifdef N_sel
  (center (pw_reburp:sp8 ph26):f1 (pwn_sl:sp32 ph26):f3)
#else
  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)
#endif

  DELTA

  (pw_pc9:sp7 ph27):f1

  2u
  p54:gp4
  d16

/*********************/
/*   15N labeling    */
```

```
/*********************/
#ifndef Hsteady_flg
  (pwn ph1):f3
#else
  (pwn ph11):f3
#endif

  4u
#ifdef gd_sel
  2u
  p55:gp5
  d16

  "DELTA1 = pw_eburp*(0.5+0.67*0.5) - 2u - p55 - d16 - 32u - 100u"
#else
  "DELTA1 = pw_eburp*(0.5+0.67*0.5) - 32u - 100u"
#endif
  "DELTA2 = -1.0*DELTA1"

  if "DELTA1 > 0" {
    DELTA1
  }

  32u gron0*-1.0
  d0
  100u groff

#ifdef c_flg
  (center (pw_bip:sp6 ph26):f1 (pwc_ad:sp22 ph26):f2)
#else
  (pw_bip:sp6 ph26):f1
#endif

  32u gron0*1.0
  d0
  100u groff

  "DELTA = pw_eburp*(0.5-0.67*0.5)"
  DELTA

  (pwn*2 ph2):f3

  "DELTA = 32u + 100u"
  DELTA

#ifdef c_flg
  (center (pw_bip:sp6 ph26):f1 (pwc_ad:sp22 ph26):f2)
#else
  (pw_bip:sp6 ph26):f1
#endif

  if "DELTA2 > 0" {
    DELTA2
  }

  4u
/***************************************/
/*   Planar mixing N->H back transfer   */
/***************************************/
  (pw_eburp:sp9 ph26):f1
  (pwn ph3):f3

  2u
  p56:gp6
  d16

  "DELTA = taub - 2u - p56 - d16 - pw_reburp*0.5"
  DELTA
```

```
  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)

  DELTA

  2u
  p56:gp6
  d16

  (pwn ph29):f3
  (pw_eburptr:sp19 ph27):f1

  2u
  p57:gp7
  d16

  "DELTA = taub - 2u - p57 - d16 - pw_reburp*0.5 - pw_eburp*0.67"
  DELTA

  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)

  DELTA

  2u
  p57:gp7
  d16

  (pw_eburp:sp9 ph26)

  if "l3%2 == 0" {
    (pwn ph28):f3
  }
  else {
    (pwn ph26):f3
  }

/*********************************************************/
/*   IPAP/S3E and decoding coherence selection gradient  */
/*********************************************************/
#ifdef gd_sel
  2u
  p58:gp8*EA
  d16
#endif

  2u
  p59:gp9*EA
  d16

#ifdef gd_sel
  "DELTA = tauc - 2u - p58 - d16 - 2u - p59 - d16 - pwn - pw_reburp*0.5"
#else
  "DELTA = tauc - 2u - p59 - d16 - pwn - pw_reburp*0.5"
#endif /*gd_sel*/
  DELTA

#ifdef s3_flg
  (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)
#else
  if "l3%2 == 0" {
    (center (pw_reburp:sp8 ph26):f1 (pwn ph26 pwn ph28):f3)
  }
  else {
    (center (pw_reburp:sp8 ph26):f1 (pwn*2 ph26):f3)
  }
#endif /*s3_flg*/

  2u
  p59:gp9*EA
  d16
```

```
#ifdef s3_flg
  "DELTA = tauc - 2u - p59 - d16 - 4u - de - pwn*2 - pw_reburp*0.5"
#else
  "DELTA = tauc - 2u - p59 - d16 - 4u - de - pw_reburp*0.5"
#endif /*s3_flg*/
  DELTA

  4u BLKGRAD

#ifdef s3_flg
 if "l3%2 == 0" {
   (pwn ph26 pwn ph26):f3
 }
 else {
   (pwn ph26 pwn ph28):f3
 }
#endif

/*************************************/
/*   Test how much water left        */
/*   set d13=AQ and zgoptn -Dwater   */
/*   also use small value of rg      */
/*************************************/
#ifdef water
  if "nsdone == 0" {
    d13
    d1 pl1:f1
    (pwh*0.1 ph26):f1
  }
  else {
    d13
  }
#endif

/*************************************/
/*   Signal detection and looping    */
/*************************************/
  if "l3%2 == 0" {
    go=2 ph30
  }
  else {
    go=2 ph31
  }
  d11 mc #0 to 2

  F3QF(calclc(l3,1))
#if defined(cal_HB1) || defined(cal_HR1)
  F2QF(calclc(l2,1))
#else
  F2QF(calclist(H_offset,1))
#endif /*cal_HB1, cal_HR1*/
  F1EA(calgrad(EA) & calph(ph3, +180), caldel(d0, +in0) & calph(ph1, +180) & calph(ph1
1, +180) & calph(ph30, +180) & calph(ph31, +180))

HaltAcqu, 1m
exit

ph0=2 0
ph1=0 2
ph2=0 0 1 1 2 2 3 3
ph3=2
ph11=0
ph26=0
ph27=1
ph28=2
ph29=3
ph30=0 2 2 0
ph31=1 3 3 1
```

```
;d1 : Repetition delay d1
;d3 : taua ~1/4J(NH) 2.70 ms
;d5 : taub ~1/4J(NH) 2.70 ms
;d11 : delay for disk i/o, 30ms
;d13 : set to acq time for testing water preservation
;d16 : gradient recovery delay, 200us
;d17 : time_T1, time for CEST, typically several 100s of ms
;pl1 : tpwr - power level for 1H pulses
;pl3 : dhpwr2 - power level for 15N pulse pwn
;pl10 : tsatpwr - power level for water presat
;pl11 : tpwrmess - power level for Messerle purge
;p1 : pwh
;p3 : pwn
;p6 : f1 channel - 180 degree shaped pulse for refocussing
;                    Bip720,50,20.1            (200us at 600.13 MHz)
;p7 : f1 channel -  90 degree shaped pulse for excitation
;                    Pc9_4_90.1000             (3.0ms at 600.13 MHz)
;p8 : f1 channel - 180 degree shaped pulse for refocussing
;                    Reburp.1000               (2.0ms at 600.13 MHz)
;p9 : f1 channel -  90 degree shaped pulse for excitation
;                    Eburp2.1000/Eburp2tr.1000 (1.92ms at 600.13 MHz)
;p22 : pwc_ad
;p32 : pwn_sl for selective 15N pulse on amides
;p51 : gradient pulse 51                                [400 usec]
;p52 : gradient pulse 52                                [1000 usec]
;p53 : gradient pulse 53                                [1000 usec]
;p54 : gradient pulse 54                                [1500 usec]
;p55 : gradient pulse 55                                [1250 usec]
;p56 : gradient pulse 56                                [256 usec]
;p57 : gradient pulse 57                                [256 usec]
;p58 : gradient pulse 58                                [256 usec]
;p59 : gradient pulse 59                                [400 usec]
;spnam6 : Bip720,50,20.1
;spnam7 : Pc9_4_90.1000
;spnam8 : Reburp.1000
;spnam9 : Eburp2.1000
;spnam19 : Eburp2tr.1000
;spnam22 : shape for pwc_ad
;spnam32 : shape for pwn_sl
;cnst1 : offset (Hz) for B1 calibration from o1
;cnst5 : H(N) chemical shift (offset, in ppm)
;cnst6 : H(N) excitation band width (in ppm)
;cnst16 : weak CEST 1H B1 field in Hz
;l2 : pointer to the delay value for calibration of B1
;l3 : pointer to switch between IP and AP
;inf1 : 1/SW(X) = 2*DW(X)
;in0 : 1/(2*SW(x))=DW(X)
;nd0 : 2
;ns : 2*n
;FnMODE : echo-antiecho

;use gradient ratio:    gp 5 : gp 8
;                         80 : -39.5

;for z-only gradients:
;gpz0: 0.25%
;gpz1: -25%
;gpz2: 15%
;gpz3: -60%
;gpz4: -70%
;gpz5: 80%
;gpz6: 60%
;gpz7: 15%
;gpz8: -39.5%
;gpz9: 30%

;use gradient files:
;gpnam1: SMSQ10.32
```

```
;gpnam2: SMSQ10.32
;gpnam3: SMSQ10.32
;gpnam4: SMSQ10.32
;gpnam5: SMSQ10.32
;gpnam6: SMSQ10.32
;gpnam7: SMSQ10.32
;gpnam8: SMSQ10.32
;gpnam9: SMSQ10.32

;zgoptns: Dfsat, Dfscuba, Dmess_flg, Df1180, Dgd_sel, Dc_flg, DN_sel, Dcal_HB1, Dcal_H
R1, Ds3_flg, Dcalc_SP, DHsteady_flg
```