Supporting Information

Quantitative Measurement of Exchange Dynamics in Proteins via ^{13}C Relaxation Dispersion of $^{13}CHD_2$-Labeled Samples

Enrico Rennella, Anne K. Schuetz and Lewis E. Kay

Sample Preparation. All proteins were expressed in *Escherichia Coli* BL21(DE3) cells grown in M9 minimal media (~100% D_2O), containing ^{15}N-ammonium chloride (1 g/L) and [^{2}H,^{12}C]-glucose (3 g/L) as the sole nitrogen and carbon sources, respectively. Specifically labeled precursors were added one hour prior to the induction of protein overexpression with 1mM IPTG, as described previously (Goto et al. 1999). Purification of samples followed literature protocols, as summarized below.

(1) B1 domain of immunoglobulin binding protein G (Huth et al. 1997): 1.0 mM [U-^{2}H; Ileδ1-$^{13}CHD_2$; Leu, Val-$^{13}CHD_2$/$^{13}CHD_2$; Met-$^{13}CHD_2$]-labeled protein, 50 mM potassium phosphate, 100 mM NaCl, 0.1 mM NaN_3, 100% D_2O, pH 7.5.

(2) G48A Fyn SH3 domain (Bouvignies et al. 2014): 1.35 mM [U-^{2}H; Ileδ1-$^{13}CHD_2$; Leu, Val-$^{13}CHD_2$/$^{13}CHD_2$]-labeled protein, 50 mM sodium phosphate, 0.2 mM EDTA, 0.05% NaN_3, 10% D_2O, pH 7.0.

(3) The half-proteasome, $\alpha_7\alpha_7$, from *T. Acidophilum* (Sprangers and Kay 2007): 0.9 mM (monomer concentration) samples of [U-^{2}H; Ileδ1-$^{13}CH_3$; Leu,Val-$^{13}CH_3$/$^{12}CD_3$; Met-$^{13}CH_3$]- and [U-^{2}H; Ileδ1-$^{13}CHD_2$; Leu,Val-$^{13}CHD_2$/$^{13}CHD_2$; Met-$^{13}CH_3$]-labeled proteins, 25 mM potassium phosphate, 50 mM NaCl, 0.03% NaN_3, 1 mM EDTA, 100% D_2O, pH 7.5. Note that Met-$^{13}CH_3$ in both samples was used to quantify concentrations of $^{13}CH_3$- and $^{13}CHD_2$-labeled protein, as described previously (Rennella et al. 2015).

(4) DNA encoding p97 from *Mus musculus* was obtained from GenScript with an N-terminal 6xHis tag and TEV cleavage site, sub-cloned into the NdeI and XhoI sites of pET29b+ (Novagen). The truncation used for this study was p97 1-480, encompassing the N-terminal domain, the D1 nucleotide domain and the linker between D1 and D2. Point mutation R95G and a stop codon at amino acid position 481 were introduced using

Quikchange site-directed mutagenesis (Agilent). Precursors for selective labeling of Ile-δ1-[$^{13}CHD_2$], Leu/Val-γ/δ-([$^{13}CHD_2$, $^{13}CHD_2$] and Met-ε-[$^{13}CHD_2$] were added at OD600 ≈ 0.8, followed 1 h later by addition of 1 mM IPTG for induction. Expression was performed for ca. 20h overnight at 18 °C. Protein was purified as detailed previously (Chou et al. 2014) by using Ni-affinity chromatography followed by cleavage of the affinity tag using TEV protease. The cleaved proteins were concentrated using an Amicon Ultra-15 100K molecular weight cutoff filter (Millipore) and the nucleotide was digested at room temperature using apyrase (New England Biolabs, ca. 2 units per NMR sample) for at least 2 h. The apo protein was further purified on a HiLoad 16/60 Superdex 200 gel filtration column (GE Healthcare). The reducing agent BME (2 mM) was used for Ni-affinity chromatography and 2 mM DTT added during the cleavage reaction, nucleotide digestion and gel filtration. Protein was subsequently buffer exchanged using 100 kDa MWCO cutoff filters into D_2O buffer containing 25 mM HEPES, pH 7.0, 50 mM NaCl, 10 mM ADP, 5 mM TCEP at a protein concentration of 600 μM monomer (equivalent to 100 μM p97 hexamer).

NMR Spectroscopy: All experiments were recorded on Bruker AVANCE III HD 600 (GB1, Fyn SH3, $\alpha_7\alpha_7$) or 800 (p97) MHz spectrometers equipped with cryogenically cooled probes. $^{13}CH_3$-based ^{13}C-CPMG data sets were recorded as described in the literature (Korzhnev et al. 2004a; Lundstrom et al. 2007), while $^{13}CHD_2$-based ^{13}C- (GB1, Fyn SH3, $\alpha_7\alpha_7$, p97) and ^1H-CPMG (p97) spectra were measured using the pulse scheme described in Figure 1 of the text (^{13}C) and according to the protocol of Baldwin *et al* (Baldwin et al. 2010) (^1H).

$^{13}CHD_2$ ^{13}C-CPMG data sets were recorded as pseudo 3D experiments with 8 scans/FID (32 scans for p97) and a recycle delay between scans of 2 s (2.5 s for p97). Constant-time CPMG delays, T_{relax}, of 40 ms, 25 ms, 20 ms and dispersion profiles comprising 26, 26, 22 ν_{CPMG} values ranging from [25,1000], [40,1000] and [50,2000] Hz were recorded for G48A Fyn SH3, $\alpha_7\alpha_7$ and p97, respectively, giving rise to total acquisition times of 27 (SH3), 35 ($\alpha_7\alpha_7$) and 71 (p97) hours. A methyl-TROSY experiment was recorded on $^{13}CH_3$ $\alpha_7\alpha_7$ using the same parameters as for the $^{13}CHD_2$-sample. A ^{1}H CPMG data set (p97) was recorded with 32 scans/FID (recycle delay of 2.5s), T_{relax} = 21 ms and 23 ν_{CPMG} values ranging from [48,2000] Hz, for a net acquisition time of 74 hours. Analysis of the CPMG data was done using the in-house written program *chemex* (https://github.com/gbouvignies/chemex), following well established protocols (Korzhnev et al. 2004b).

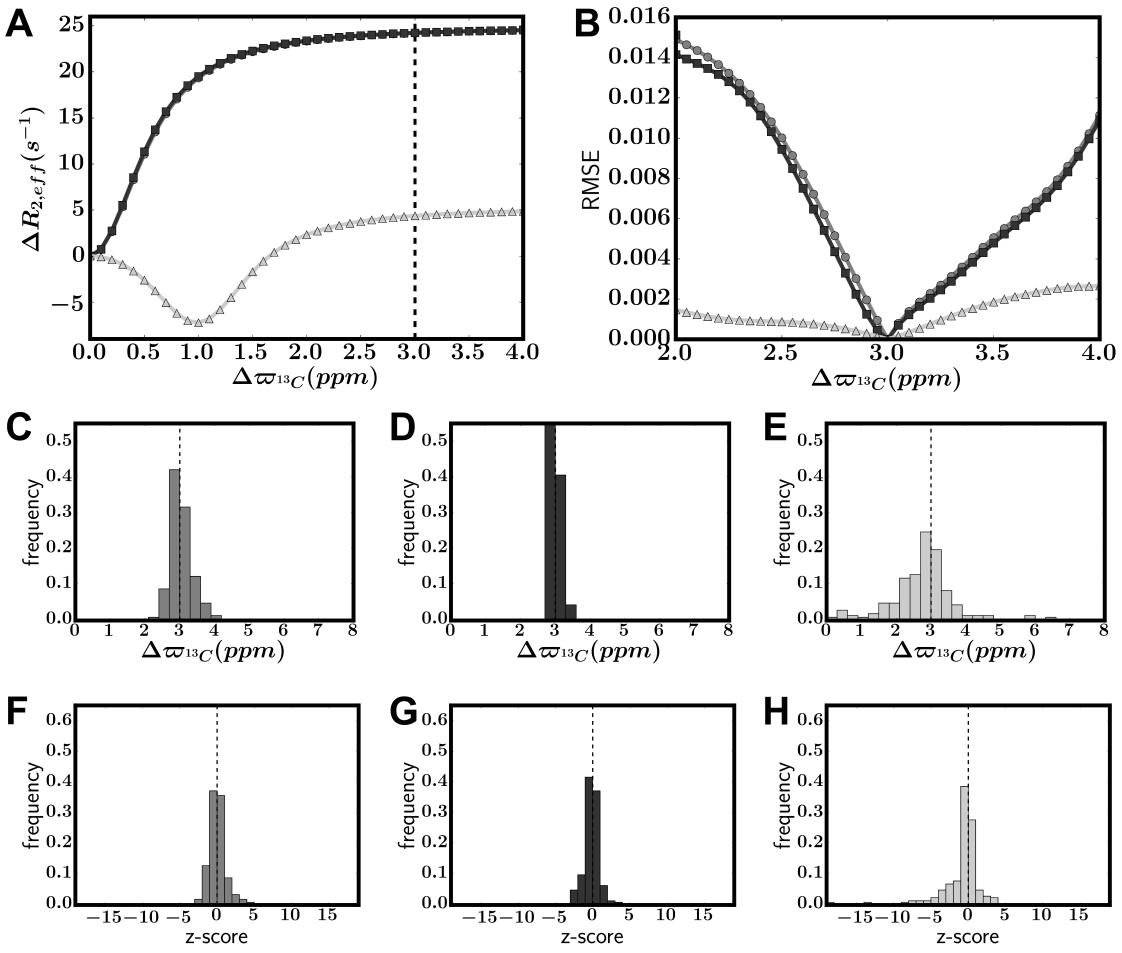

Figure S1. In some cases more robust ^{13}C $\Delta\varpi$ values can be extracted from $^{13}CHD_2$-CPMG profiles than from fits of methyl-TROSY CPMG data. (A) Size of dispersion profiles, $\Delta R_{2,eff} = R_{2,eff}(\nu_{CPMG} = 10\text{ Hz}) - R_{2,eff}(\nu_{CPMG} = 10000\text{ Hz})$, as a function of ^{13}C $\Delta\varpi$ as would be obtained in ^{13}C SQ (dark grey circles) and methyl-TROSY (black squares, ^{1}H $\Delta\varpi$=0; grey triangles ^{1}H $\Delta\varpi$=0.25 ppm) experiments. Note that black squares and grey circles overlap. Simulations were performed assuming a model of two-site chemical exchange with T_{relax} = 100 ms, k_{ex} = 500 s^{-1}, p_E = 5%, $R(C_{xy})$ = 10 s^{-1}, $R(2H_{xy}C_{xy})$ = 10 s^{-1}, where $R(C_{xy})$ and $R(2H_{xy}C_{xy})$ correspond to ^{13}C single quantum and ^{1}H-^{13}C multiple quantum relaxation rates, respectively, ^{13}C $\Delta\varpi$ = 3 ppm, ^{1}H $\Delta\varpi$ = 0 ppm (black

squares) or 0.25 ppm (grey triangles) and no noise. Unlike methyl-TROSY based profiles, ^{13}C SQ dispersion curves are independent of ^{1}H $\Delta\varpi$. (B) RMSE = $\sqrt{\sum_{j=1}^{N}\frac{1}{N}(R_{2,eff}^{E,j}(\nu_{CPMG})-R_{2,eff}^{C,j}(\nu_{CPMG}))^2}$ as a function of ^{13}C $\Delta\varpi$, obtained from fits of calculated ^{13}C SQ (dark grey circles) and methyl-TROSY (black squares or grey triangles) dispersion profiles, where N is the number of points in each dispersion profile and the superscripts E and C denote target and calculated (best fit) $R_{2,eff}$ values, respectively. Dispersions were generated with input parameters indicated above, with the exception that T_{relax} = 40 ms, and ν_{CPMG} varies between 25 and 1000 Hz. (C-H) In order to evaluate the precision and accuracy of extracted ^{13}C $\Delta\varpi$ values for the exchange parameters listed above (and including experimental noise) we have generated a series of ^{13}C SQ and methyl-TROSY CPMG profiles and fit the resulting data as per the experimental curves. Profiles were computed with ^{1}H $\Delta\varpi$ = 0 ppm (C,D,F,G) or ^{1}H $\Delta\varpi$ = 0.25 ppm (E,H). Gaussian noise was added to each theoretical curve, generating 200 simulated profiles for both $^{13}CHD_2$ and $^{13}CH_3$ classes of experiment. Noise levels were set to 3% and 1% for ^{13}C SQ CPMG and methyl TROSY-CPMG, respectively, with the difference accounting for the different average S/N values for cross peaks measured in dispersion experiments recorded on $\alpha_7\alpha_7$ (Figure 4). Each profile was fit using a pair of different starting conditions with, Case 1: (p_E,k_{ex},$R(C_{xy})$,$R(2H_{xy}C_{xy})$) set to input values, $\Delta\varpi_C$=0.1 ppm, $\Delta\varpi_H$ = 0.1 ppm; Case 2: (p_E,k_{ex}) = (2%,1000 s^{-1}), $R(C_{xy})$ = $R_{2,eff}$(1000 Hz), $R(2H_{xy}C_{xy})$= $R_{2,eff}$(1000 Hz), $\Delta\varpi_C$=0.1 ppm, $\Delta\varpi_H$ = 0.1 ppm and the parameters with the lowest χ^2 were retained. (C,D) Histograms of the distributions of the fitted ^{13}C $\Delta\varpi$ values for (C) ^{13}C SQ CPMG and (D) methyl TROSY-CPMG with ^{1}H $\Delta\varpi$ = 0 ppm; the input value of 3

ppm is shown by the dashed line. (F,G) Histograms of the *z*-score of the fitted ^{13}C $\Delta\varpi$ values for ^{13}C SQ CPMG (F) and methyl-TROSY CPMG, ^{1}H $\Delta\varpi$ = 0 ppm (G) ($z = \frac{x-u}{\sigma}$ where x is a fitted $\Delta\varpi$ value, u =3 ppm, and σ is the error in the fit of ^{13}C $\Delta\varpi$ based on a covariance matrix analysis (Press et al. 1988)). (E,H) Corresponding histograms as in (D,G) but with ^{1}H $\Delta\varpi$ = 0.25 ppm. As ^{1}H $\Delta\varpi$ values increase the methyl-TROSY dispersions decrease and the extracted shift values can become less robust, as demonstrated in (E,H). Simulations and fits were done using the in-house program *chemex* (https://github.com/gbouvignies/chemex).

References

Baldwin AJ, Religa TL, Hansen DF, Bouvignies G, Kay LE (2010) (13)CHD(2) methyl group probes of millisecond time scale exchange in proteins by (1)H relaxation dispersion: an application to proteasome gating residue dynamics J Am Chem Soc 132:10992-10995 doi:10.1021/ja104578n

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

Chou TF et al. (2014) Specific inhibition of p97/VCP ATPase and kinetic analysis demonstrate interaction between D1 and D2 ATPase domains J Mol Biol 426:2886-2899 doi:10.1016/j.jmb.2014.05.022

Goto NK, Gardner KH, Mueller GA, Willis RC, Kay LE (1999) A robust and cost-effective method for the production of Val, Leu, Ile (d1) methyl-protonated ^{15}N-,^{13}C-,^{2}H-labeled proteins J Biomol NMR 13:369-374

Huth JR, Bewley CA, Jackson BM, Hinnebusch AG, Clore GM, Gronenborn AM (1997) Design of an expression system for detecting folded protein domains and mapping

macromolecular interactions by NMR Protein Sci 6:2359-2364 doi:10.1002/pro.5560061109

Korzhnev DM, Kloiber K, Kay LE (2004a) Multiple-quantum relaxation dispersion NMR spectroscopy probing millisecond time-scale dynamics in proteins: theory and application J Am Chem Soc 126:7320-7329

Korzhnev DM, Salvatella X, Vendruscolo M, Di Nardo AA, Davidson AR, Dobson CM, Kay LE (2004b) Low-populated folding intermediates of Fyn SH3 characterized by relaxation dispersion NMR Nature 430:586-590

Lundstrom P, Vallurupalli P, Religa TL, Dahlquist FW, Kay LE (2007) A single-quantum methyl 13C-relaxation dispersion experiment with improved sensitivity J Biomol NMR 38:79-88

Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1988) Numerical Recipes in C. Cambridge University Press, Cambridge

Rennella E, Huang R, Velyvis A, Kay LE (2015) (13)CHD2-CEST NMR spectroscopy provides an avenue for studies of conformational exchange in high molecular weight proteins J Biomol NMR 63:187-199 doi:10.1007/s10858-015-9974-z

Sprangers R, Kay LE (2007) Quantitative dynamics and binding studies of the 20S proteasome by NMR Nature 445:618-622

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define s(x) ( (x)*(x) )
#define PI 3.141592

/* CHD2cpmg_makefiles.c
Written by LEK on Oct 1, 2015 to generate files for use with 13CHD2 cpmg with Deut decoupling.
At the moment it generates one file pwrlist0 that is by the pulse sequence to read in
the powers in WATTs
Modified to change powers to allow for power to be higher than Power level of pwd_ml

We wish to ensure that there is always an integral number of 2H pulses in each tau period of a
 tau 180 tau echo. If at the power level of tpwr_ml corresponding to pwd_ml there is a non-integral
 number of pulses we can either round up or round down. In general it is better to round up although this
 does increase power. I prefer to round up.

Two ways of doing this:

Method 1. Choose higher power if new pulse width of 2H pulse is less than 1.414 time smaller than
          nominal 2H decoupling pulse of pwd_ml. This is slightly preferred but generally involves more
power.

Method 2. Choose higher power if new pulse width is less than 1.414 of pwd_ml AND
          the difference between new power level and tpwr_ml is less than 1.1 times
          the difference between power level that is lower than tpwr_ml (but gives an integral number
          of 2H pulses during tau of the tau 180 (C) tau element) called power_0 and tpwr_ml

Method 3. Only apply 2H decoupling until 250 Hz B1
*/


  main(argc,argv)

       int argc;
       char *argv[];

  {   float  pwd_ml,tpwr_ml;
      float  time_T2,pwc_cp,tauCPMG,tauCPMG1,tauCPMG2,tauCPMG3;
      float
numpulse_0,numpulse_1,numpulse_2,numpulse_3,pw_ml_0,pw_ml_1,pw_ml_2,pw_ml_3,power_1,power_2,power_3,power_0;
      float numpulse_00,power_00,pw_ml_00,power_0a,mu_cpmg;
      int    i,ncyc,meth;
      FILE   *fi,*fi0,*fi1,*fi2,*fi3;


        if(argc != 6) {
          printf("ncyc_list_2H file must be present in this directory!!\n");
          printf("USAGE:\n");
          printf("Argument 1: Pulse width (us) of pwd_ml \n");
          printf("Argument 2: Power level of pwd_ml in WATTS NB, not dbW !\n");
          printf("Argument 3: time_T2 in units of ms \n");
          printf("Argument 4: pwc_cp in units of us \n");
          printf("Argument 5: Method 1 (1) or Method 2 (2) or Method 3 (3) Recommend Method 3  \n");
          exit(1);
        }

        pwd_ml = (float) atof(argv[1]);
        tpwr_ml = (float) atof(argv[2]);
        time_T2 = (float) atof(argv[3]);
        pwc_cp = (float) atof(argv[4]);
        meth = (int) atoi(argv[5]);

  /* keep all units in us */
        time_T2 = 1000.0*time_T2;


        fi0 = fopen("pwrlist0","w");
        fi1 = fopen("pwrlist1","w");
        fi2 = fopen("pwrlist2","w");
```

```
        fi3 = fopen("pwrlist3","w");
        fprintf(fi0,"Watt\n");
        fprintf(fi1,"Watt\n");
        fprintf(fi2,"Watt\n");
        fprintf(fi3,"Watt\n");

       fi = fopen("ncyc_list_2H","r");
         while(fscanf(fi,"%d",&ncyc) !=EOF) {
          printf("ncyc is %d\n",ncyc);

          if(ncyc != 0) {
          tauCPMG = time_T2/(4.0*(float)(ncyc)) - pwc_cp;
          tauCPMG1 = tauCPMG - pwc_cp*2.0/PI;
          tauCPMG2 = tauCPMG - pwc_cp - 2.0;
          tauCPMG3 = tauCPMG - pwc_cp*2.0/PI - 2.0;
        }

/* calculate the number of 180 pulses */
          numpulse_0 = floor((tauCPMG+pwc_cp)/(2.0*pwd_ml));
          numpulse_1 = floor((tauCPMG1)/(2.0*pwd_ml));
          numpulse_2 = floor((tauCPMG2)/(2.0*pwd_ml));
          numpulse_3 = floor((tauCPMG3)/(2.0*pwd_ml));

          numpulse_00 = ceil((tauCPMG+pwc_cp)/(2.0*pwd_ml));

/* calculate the corresponding 90o pulse widths */
          pw_ml_0 = (tauCPMG+pwc_cp)/(2.0*numpulse_0);
          pw_ml_1 = (tauCPMG1)/(2.0*numpulse_1);
          pw_ml_2 = (tauCPMG2)/(2.0*numpulse_2);
          pw_ml_3 = (tauCPMG3)/(2.0*numpulse_3);

          pw_ml_00 = (tauCPMG+pwc_cp)/(2.0*numpulse_00);

/* now calculate the appropriate watts to make the 1H cw field */
          power_0 = s((pwd_ml))*(tpwr_ml)/s((pw_ml_0));
          power_1 = s((pwd_ml))*(tpwr_ml)/s((pw_ml_1));
          power_2 = s((pwd_ml))*(tpwr_ml)/s((pw_ml_2));
          power_3 = s((pwd_ml))*(tpwr_ml)/s((pw_ml_3));

          power_00 = s((pwd_ml))*(tpwr_ml)/s((pw_ml_00));

          if(ncyc==0) {
               power_0a=0;
               power_1=0;
               power_2=0;
               power_3=0;
           }

         if(meth==1) {

         if((pwd_ml/pw_ml_00) < 1.414)
             power_0a = power_00;
          else
             power_0a = power_0;

        }

         if(meth==2) {

         if((pwd_ml/pw_ml_00 < 1.414) && (fabs(power_00 - tpwr_ml) < 1.10*fabs(power_0 - tpwr_ml)))
             power_0a = power_00;
          else
             power_0a = power_0;

        }

       if(meth==3) {
         mu_cpmg = ncyc*1000000.0/time_T2;
          if(mu_cpmg <= 250.0) {
          if((pwd_ml/pw_ml_00 < 1.414) && (fabs(power_00 - tpwr_ml) < 1.10*fabs(power_0 - tpwr_ml)))
```

```
          power_0a = power_00;
        else
          power_0a = power_0;
     }
       else
         power_0a = 0.01;
    }

       if(numpulse_0 == 0)
            power_0a=0.01;
       if(numpulse_1 == 0)
            power_1=0.01;
       if(numpulse_2 == 0)
            power_2=0.01;
       if(numpulse_3 == 0)
            power_3=0.01;

       fprintf(fi0,"%f\n",power_0a);
       fprintf(fi1,"%f\n",power_1);
       fprintf(fi2,"%f\n",power_2);
       fprintf(fi3,"%f\n",power_3);

  }
  fclose(fi);
  fclose(fi0);
  fclose(fi1);
  fclose(fi2);
  fclose(fi3);

  printf("1 file has been created; pwrlist0\n");
  printf("store pwrlist0 lists/va \n");
  printf("run executable move_lek_CHD2_CPMG to move the file to appropriate places\n");

}
```

```
/* 13CHD2_13C_CPMG_lek_600_cp

    This pulse sequence will allow one to perform the following experiment:

    2D 1H/13C to measure exchange via 13CHD2 CPMG

    Assumes that sample is specifically 13C labeled

        1H: O1 on water (4.7ppm) at end, most of the time at 0.8 ppm (methyl 1H resonances)
            pwh = p1 1H pw90 @  power level pl1 highest power (13W)

       13C: O2 centre at 20 ppm
            pwc = p2 13C pw90 @ power level pl2 highest power (175W)
            pwc_cp = p22  13C pw90 at pl22 used for 13C cpmg pulses
            power level pl21 is used for 13C decoupling.

    Pulse scheme uses higher power 1H CW during CPMG
    Can calibrate gradients under load - use C_calib

Het-CP back transfer : 51.8*p_d*4*2 = total time = 230.2 * pw , so set pw to 31.05 ms

       Written by LEK on Sept 29, 2015 from 13CHD2_13C_CEST_lek_600_v2

       Modified by LEK on Oct 5, 2015 to allow for variable power 2H
       decoupling

       Go to ../lists/ccode and run CHD2cpmg_makefiles (that uses ncyc_list_2H)
*/

#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=2m"
define pulse dly_pg2               /* Messerle purge pulse*/
              "dly_pg2=3.4m"
define pulse pwh
       "pwh=p1"                  /* 1H hard pulse at power level p1 (tpwr) */
define pulse pwc
       "pwc=p2"                 /* 13C pulse at power level pl2 (dhpwr) */
define pulse pwc_cp
       "pwc_cp=p23"             /* 13C pulse at power level pl23  */

define pulse pwh_cw
        "pwh_cw=p12"

define pulse pw_dip
        "pw_dip=p13"

define pulse p_d                /* pulse for het cp transfer back to 1H */
        "p_d = pw_dip*5.0/9.0"

define pulse pwd
       "pwd=p4"                 ; 2H pulse at power pl4

;Define delays

define delay  hscuba             /* length of 1/2 scuba delay */
              "hscuba=25m"
```

```
"in0=inf1/2"
"d11=30m"

define delay taua
       "taua=d15"        /* d15 ~ 1.8-2ms ~ 1.0s/(4*125.3)"  ~ 1 / 4J(CH) */

define delay taub
        "taub = d18"     /* d18 = 1/4JCH exactly */

define delay time_T2
       "time_T2=d13"         /* CPMG duration - <= 40 ms  */

define delay time_T2_max
       "time_T2_max=d14"      /* MAX CPMG duration - <= 40 ms */

define delay t1_aq

define delay time_eq
      "time_eq=d4"           ; time for magnetization to reach equilibrium value

define delay tauCPMG
define delay tauCPMG1
define delay tauCPMG2
define delay tauCPMG3

define list<loopcounter> ncyc_cp= <$VCLIST>
define list<power> pwrml1 = <$VALIST>

#ifdef C_comp
 define loopcounter ncyc_max ; max value of ncyc used
   "ncyc_max = l8"
#endif

;define f1180 flag

#ifdef f1180
  "d0=(in0)/2"
#else
  "d0=0.2u"
#endif

#ifndef OneDarray
define loopcounter ni
  "ni = td1/2"
#endif

; aqseq 321

/* Assign cnsts to check validity of parameter ranges */
#ifdef fsat
   "cnst10=plw10" ; tsatpwr pl10 - set max at 0.00005W
#endif

#ifdef mess_flg
   "cnst11=plw11" ; tpwrmess pl11 - set max at 1.0W
#endif

   "cnst12=plw12" ; tpwrml - power level for 1H CW decoupling during CPMG
   "cnst13=plw13" ; power level for hetcp back transfer (1H)
   "cnst17=plw17" ; power level for 1H waltz-16 decoupling during t1


   "cnst21=plw21" ; dpwr pl21 - set max at 2.0W
   "cnst23=plw23" ;      pl23 - power level for pwc_cp (CPMG 13C pulses)
   "cnst27=plw22" ; power level for het-cp back transfer (13C)
#ifdef C_comp
   "cnst26=plw26" ; power level for 13C compensation
#endif

#ifdef Ddec
```

```
   "cnst4=plw4"   ; dpwr3 pl4 - set max at 10.5W
   "cnst41=plw41" ; dpwr3D pl41 - set max at 1.5W
   "cnst42=plw42" ;        pl42 - used for 2H decoupling during CPMG
#endif

; Begin Actual Pulse Sequence

"l2 = 0" ; l2 is the counter for the ncyc_cp values for cpmg

1 ze

; check validity of parameters and assign values to some of them

#ifdef C_comp
if "l8 < 2" {
2u
print "error: l8 must be set to the max ncyc value used  !!!  "
goto HaltAcqu
}
#endif

  if "d1 < 2.0s"
  {
   2u
   print "error: d1 is too short; 2H decoupling is applied during CEST"
   goto HaltAcqu
  }

#ifdef fsat
  if " cnst10 > 0.00005"
  {
   2u
   print "error: tpwrmess pl10 too large"
   goto HaltAcqu
  }
#endif

#ifdef mess_flg
  if " cnst11 > 1.0"
  {
   2u
   print "error: tpwrmess pl11 too large"
   goto HaltAcqu
  }
#endif

  if " cnst12 > 6.0"
  {
   2u
   print "error: 1H CW decoupling during CPMG  pl12 too large < 6 W"
   goto HaltAcqu
  }

  if " cnst13 > 2.0"
  {
   2u
   print "error: 1H  power for het cp back transfer pl13 too large < 2 W"
   goto HaltAcqu
  }

  if " cnst17 > 2.0"
  {
   2u
   print "error: 1H WALTZ decoupling pl17 during t1 too large < 2 W"
   goto HaltAcqu
  }

  if "time_T2 > 40.1m"
  {
   2u
```

```
   print "error: time_T2 too long < 41ms"
   goto HaltAcqu
  }

  if "time_T2_max > 50m"
  {
   2u
   print "error: time_T2_max too long < 50ms"
   goto HaltAcqu
  }

  if " cnst21 > 1.5"
  {
   2u
   print "error: dpwr pl21 too large"
   goto HaltAcqu
  }

#ifdef C_comp
  if " cnst26 > 100.0"
  {
   2u
   print "error: 13C power for 13C compenstation is too large < 100W"
   goto HaltAcqu
  }
#endif

  if " cnst27 > 30.0"
  {
   2u
   print "error: 13C power for hetcp back transfer is too large < 30W"
   goto HaltAcqu
  }

  if " cnst23 > 80.0"
  {
   2u
   print "error: power level for  13C CPMG field is too large"
   goto HaltAcqu
  }

 if "aq > 0.065s"
  {
   2u
   print "error: aq is too long; < 65 m"
   goto HaltAcqu
  }

  if "p_d > 35u"
  {
   2u
   print "error: p_d is too long; must be < 35u"
   goto HaltAcqu
  }

#ifdef Ddec
  if "cnst4 > 15.0"
  {
   2u
   print "error: dpwr3 pl4 too large"
   goto HaltAcqu
  }

  if "cnst41 > 4.0"
  {
   2u
   print "error: dpwr3D pl41 too large"
   goto HaltAcqu
  }
```

```
  if "cnst42 > 8.0"
  {
   2u
   print "error: pl42 too large - used during 13C CPMG"
   goto HaltAcqu
  }

  if "pwd > 160u" {
   2u
   print "error: pwd is too large < 160u"
   goto HaltAcqu
  }

  if "pwd < 100u" {
   2u
   print "error: pwd is too small > 100u"
   goto HaltAcqu
  }

  if "p42 < 245u" {
   2u
   print "error: 2H pw42 during CPMG is too small < 250u"
   goto HaltAcqu
  }

;   d11 LOCKDEC_ON ; Not required for AvanceIII-HD
  50u LOCKH_ON
  d11 H2_PULSE
  2u pl41:f4      ; set power pl4 for 2H flipback pulses
#endif

2 d11 do:f2      ; loop back here NS times per fid

; continue to check run time variables

"l5 = ncyc_cp[l2]"
"COUNTER = (trunc(l5 + 0.3))"  ; used subsequently for CPMG

if "COUNTER !=0" {
   "tauCPMG = time_T2*0.25/COUNTER - pwc_cp"
 if "tauCPMG + pwc_cp < 124.0u"
   {
     2u
     print "error: tauCPMG < 124u too short for CPMG"
     goto HaltAcqu
   }
  }
else {
   "tauCPMG = time_T2*0.25 - pwc_cp"
}
0.2u

if "COUNTER > 80"
   {
     2u
     print "error: ncyc_cp must be < 81"
     goto HaltAcqu
   }

 if "pwrml1 > 9.2"
   {
     2u
     print "error: Too much 2H decoupling power during CPMG; < 9.2 W"
     goto HaltAcqu
   }

/* 1H Heating period */
 20u fq=0:f1        ;  1H carrier initially on water

if "COUNTER == 0"
```

```
  {
   (2u pl12):f1                         ;power(tpwrml) for 1H saturation for heating
   time_T2_max cw:f1 zero               ;Hcw(time_T2_max)x
   4u do:f1                             ;cw off
   (2u pl10):f1                         ;power(tsatpwr) for 1H presaturation
  }
else
  {
   (2u pl12):f1                         ;power(tpwrml) for 1H saturation for heating
   "DELTA=time_T2_max - time_T2"
   DELTA cw:f1 zero                    ; Hcw(time_T2_max - time_T2)x
   4u do:f1                             ; cw off
}
0.2u

 20u fq=0:f1        ;  1H carrier initially on water
 20u fq=0:f2

  2u pl1:f1       ; set power pl1 for 1H hard pulses
  2u pl2:f2       ; set power pl2 for 13C hard pulses
  2u pl3:f3       ; set power pl3 for 15N hard pulses
#ifdef Ddec    /* D decoupling */
  2u pl42:f4
#endif

; ensure that there is no 1H magnetization left

  (pwh zero):f1

  20u UNBLKGRAMP           ; dly 20u, unblank gradients, does not work with UNBLKGRAD

  2u
  p50:gp0*0.25            ; gradient1 gt0 @ gzlvl0
  d16                     ; dly d16 (200us)

  (pwh one):f1

  2u
  p50:gp0*0.5             ; gradient1 gt0 @ gzlvl0
  d16                     ; dly d16 (200us)

  20u BLKGRAMP            ; reblank

#ifdef Ddec    /* D decoupling */
d11 H2_LOCK    ; put lock channel in lock mode
6m LOCKH_OFF   ; turn off lock hold
#endif

/* Messerle purge */

#ifdef mess_flg      ;   zgoptn -Dmess_flg
 20u pl11:f1         ;   dly 20u, set power tpwrmess pl11
 (dly_pg1 zero):f1   ;   H(dly_pg)x
 20u                 ;   dly 20u
 (dly_pg2 one):f1    ;   H(dly_pg/1.62)y
 20u pl10:f1         ;   set power tsatpwr pl10
#endif

/* Presaturation */
#ifdef fsat              ; zgoptn -Dfsat
  4u pl10:f1             ; dly 4u, set power tsatpwr pl10 for presat
  d1 cw:f1 zero          ; cw on for d1
  4u do:f1               ; dly 4u, cw off
  2u pl1:f1              ; set power tpwr pl1 for hard pulses

# ifdef fscuba            /* Scuba pulse sequence */
   hscuba                 ;  dly hscuba
    (pwh zero):f1         ;    H90x
    (pwh*2 one):f1        ;    H180y
    (pwh zero):f1         ;    H90x
```

```
   hscuba                 ;  dly hscuba
# endif                   /* end fscuba */

#else                     /* if fsat is no */
 2u pl1:f1                ; set power tpwr pl1 for hard pulses
 d1                       ; dly d1
#endif                    /* end if fsat */

#ifdef Ddec
50u  LOCKH_ON
15u H2_PULSE
#endif

; destroy 13C magnetization
  (pwc zero):f2           ; C90x

#ifdef Ddec
  20u UNBLKGRAMP          ; dly 20u, unblank gradients, lock hold already on
#else
  20u UNBLKGRAD           ; dly 20u, unblank gradients and lock_hold
#endif

  2u
  p50:gp0                 ; gradient1 gt1 @ gzlvl1
  d16                     ; dly d16 (200us)

 20u fq=cnst1:f1          ; jump to methyl 1Hs for the full sequence

; This is the real start.

/* insert possibility for C heat comp using the Varian approach */

#ifdef C_comp
if "COUNTER - ncyc_max >0" {
     2u
     print "error: ncyc_max must be greater than or equal to ncyc_cp"
     goto HaltAcqu
}
"l6 = ncyc_max - COUNTER"
"DELTA = time_T2*l6/ncyc_max"

2u pl26:f2 ; set pl26 to be 8 db higher (ie, lower power) than pl23
DELTA cw:f2 zero
2u do:f2 ; turn decoupling off channel 2

 2u
 p51:gp0*0.5                        ; gradient 0 * 0.5
 d16

#endif

  2u pl2:f2

  (pwh zero):f1                                    ; H90x

  2u
  p51:gp1                                          ; gradient1
  d16

  "DELTA = taua  - 2u - p51 - d16"                 ; delay 1/4JCH
  DELTA

  (center (pwh*2 ph26):f1 (pwc*2 ph26):f2)         ; H180x,C180x

  "DELTA = taua  - 2u - p51 - d16 "
  DELTA                                            ; dly 1/(4*JCH)

  2u
  p51:gp1                                          ; gradient1
  d16
```

```
  (pwh one):f1                                      ; H90y

  2u
  p52:gp2                                           ; gradient2
  d16

# ifdef Ddec
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd one):f4          ; 2H 90(y)
  2u pl41:f4            ; dly 2u,  set pwr pl41 dpwr3D
  (2u cpds4 zero):f4    ; Turn on 2H decoupling - cpd4, phase x
# endif

  2u pl2:f2
  (pwc zero):f2                                     ; C90(PH1)

  taub

  (center (pwh*2 ph26):f1 (pwc*2 ph26):f2)  ; H180x,C180x

  taub

  (pwc one):f2                                     ; C90(y)
  0.2u
  (pwh zero):f1

# ifdef Ddec
  2u do:f4              ; Turn off 2H decoupling on channel 4
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd three):f4        ; 2H 90(-y)
# endif

  2u
  p53:gp3                                           ; gradient3
  d16

; now begin the CPMG period

time_eq

 if "COUNTER != 0"
  {
   "tauCPMG1=tauCPMG - pwc_cp*2.0/PI - 2u"
   "tauCPMG2=tauCPMG - pwc_cp - 2u"
   "tauCPMG3=tauCPMG - pwc_cp*2.0/PI - 2u - 2u"
  }
  else {
     "tauCPMG1 = 250u"
     "tauCPMG2 = 250u"
     "tauCPMG3 = 250u"
 }

  2u pl12:f1                        ; set power for 1H CW
  2u pl23:f2                        ; set power for 13C CPMG

# ifdef Ddec
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd one):f4          ; 2H 90(y)
  2u pwrml1:f4          ; dly 2u,  set pwr of 2H according to ncyc value
# endif

  (pwc_cp ph1):f2

#ifdef Ddec
   (2u cw ph26):f4       ; Turn on 2H decoupling - phase x
#else
    2u
#endif
```

```
 if "COUNTER == 1"                        ; ncyc_cp=1
  {
   tauCPMG1 cpds1:f1 zero              ; USE CW DECOUPLING
   (pwc_cp*2 one):f2
   tauCPMG2
  }
 if "COUNTER == 2"                        ; ncyc_cp=2
  {
   tauCPMG1 cpds1:f1 zero             ; USE CW DECOUPLING
   (pwc_cp*2 one):f2
   tauCPMG
   tauCPMG
   (pwc_cp*2 one):f2
   tauCPMG2
  }
  if "COUNTER-2 > 0"                         ; ncyc_cp > 2
  {
   "l0=COUNTER-2"
   tauCPMG1 cpds1:f1 zero
   (pwc_cp*2 one):f2
   tauCPMG
7  tauCPMG
   (pwc_cp*2 one):f2
   tauCPMG
   lo to 7 times l0
   tauCPMG
   (pwc_cp*2 one):f2
   tauCPMG2
  }

#ifdef Ddec
   2u do:f4
#else
   2u
#endif
   (pwc_cp*2 ph3):f2
   2u

#ifdef Ddec
   (2u cw ph26):f4       ; Turn on 2H decoupling - phase x
#else
    2u
#endif

 if "COUNTER == 1"                        ; ncyc_cp=1
  {
   tauCPMG2
   (pwc_cp*2 one):f2
   tauCPMG3
  }
 if "COUNTER == 2"                        ; ncyc_cp=2
  {
   tauCPMG2
   (pwc_cp*2 one):f2
   tauCPMG
   tauCPMG
   (pwc_cp*2 one):f2
   tauCPMG3
  }
  if "COUNTER-2 > 0"                         ; ncyc_cp > 2
  {
   "l0=COUNTER-2"
   tauCPMG2
   (pwc_cp*2 one):f2
   tauCPMG
8  tauCPMG
   (pwc_cp*2 one):f2
   tauCPMG
   lo to 8 times l0
   tauCPMG
```

```
   (pwc_cp*2 one):f2
   tauCPMG3
  }
#ifdef Ddec
   2u do:f4
#else
   2u
#endif
   2u do:f1                                ; turn off 1H decoupling
   (pwc_cp zero):f2

# ifdef Ddec
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd three):f4        ; 2H 90(-y)
# endif

; end of CPMG

; now purge the water

 20u pl12:f1         ;   dly 20u, set power to pl12
 (dly_pg1 zero):f1   ;   2ms
 20u                 ;   dly 20u
 (dly_pg2 one):f1    ;   3ms
 20u pl1:f1          ;   set power to high power

  2u
  p55:gp5                                          ; gradient5
  d16

#ifdef C_calib
  2u pl25:f2
  (p25 zero):f2

  2u
  p55:gp5*2.0                                      ; gradient5
  d16
#endif

; compensate for t1 dependent heating from 1H decoupling

#ifndef OneDarray
  "t1_aq = (ni-1)*in0*2.0"                        ; this is the net acquisition time

 if "ni < 2" {
  "DELTA = 2u"
}
 else {
  "DELTA = t1_aq - d0 - d0 + 2u + in0" ; in0 takes into account f1180 in the event that it is used
}
0.2u
#else
  DELTA = 2u"
#endif

  if "DELTA < 0u" {
  2u
  print "error: 1H heat compensation during t1 is negative "
  goto  HaltAcqu
}

  2u pl17:f1                                       ; power level for 1H WALTZ decoupling
  (2u cpds7 zero):f1                               ; turn on 1H decoupling

  DELTA                                            ; ensures that same amount of 1H decoupling heating
irrespective of t1

# ifdef Ddec
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd one):f4          ; 2H 90(y)
```

```
  2u pl41:f4            ; dly 2u,  set pwr pl41 dpwr3D
  (2u cpds4 zero):f4     ; Turn on 2H decoupling - cpd4, phase x
# endif

  2u pl2:f2
  (pwc ph2):f2                                  ; C90(PH1)

  d0
  d0

  2u do:f1
  2u pl1:f1

#ifndef Ddec
  "DELTA = taub - 2u - 2u - 2u - p58 - d16"
  DELTA
#else
  2u do:f4              ; Turn off 2H decoupling on channel 4
  2u pl4:f4             ; dly 2u,  set pwr pl4 dpwr3
  (pwd three):f4        ; 2H 90(-y)
  "DELTA = taub - 2u - 2u - 2u - 2u - pwd - 2u - p58 - d16"
  DELTA
# endif

  2u
  p58:gp8*EA*-1                                         ; gradient8*EA
  d16

  (center (pwh*2 ph26):f1 (pwc*2 ph26):f2)  ; H180x,C180x

  "DELTA = taub - 2u - p56 - d16 - 2u - p58 - d16"

if "DELTA < 0.0u" {
  2u
  print "error: The p56/p58 combination is too large for taub"
  goto  HaltAcqu
}

  DELTA

  2u
  p56:gp6*EA*-1                                         ; gradient6*-EA - coherence transfer gradient
  d16

  2u
  p58:gp8*EA*-1                                           ; gradient8*EA
  d16

  (center (pwh one):f1 (pwc ph4):f2)

  1u pl13:f1
  1u pl22:f2

10  (center (p_d*6.4 ph26):f1 (p_d*6.4 ph26):f2)
    (center (p_d*8.2 ph28):f1 (p_d*8.2 ph28):f2)
    (center (p_d*5.8 ph26):f1 (p_d*5.8 ph26):f2)
    (center (p_d*5.7 ph28):f1 (p_d*5.7 ph28):f2)
    (center (p_d*0.6 ph26):f1 (p_d*0.6 ph26):f2)
    (center (p_d*4.9 ph28):f1 (p_d*4.9 ph28):f2)
    (center (p_d*7.5 ph26):f1 (p_d*7.5 ph26):f2)
    (center (p_d*5.3 ph28):f1 (p_d*5.3 ph28):f2)
    (center (p_d*7.4 ph26):f1 (p_d*7.4 ph26):f2)

    (center (p_d*6.4 ph28):f1 (p_d*6.4 ph28):f2)
    (center (p_d*8.2 ph26):f1 (p_d*8.2 ph26):f2)
    (center (p_d*5.8 ph28):f1 (p_d*5.8 ph28):f2)
    (center (p_d*5.7 ph26):f1 (p_d*5.7 ph26):f2)
    (center (p_d*0.6 ph28):f1 (p_d*0.6 ph28):f2)
    (center (p_d*4.9 ph26):f1 (p_d*4.9 ph26):f2)
    (center (p_d*7.5 ph28):f1 (p_d*7.5 ph28):f2)
```

```
    (center (p_d*5.3 ph26):f1 (p_d*5.3 ph26):f2)
    (center (p_d*7.4 ph28):f1 (p_d*7.4 ph28):f2)

    (center (p_d*6.4 ph28):f1 (p_d*6.4 ph28):f2)
    (center (p_d*8.2 ph26):f1 (p_d*8.2 ph26):f2)
    (center (p_d*5.8 ph28):f1 (p_d*5.8 ph28):f2)
    (center (p_d*5.7 ph26):f1 (p_d*5.7 ph26):f2)
    (center (p_d*0.6 ph28):f1 (p_d*0.6 ph28):f2)
    (center (p_d*4.9 ph26):f1 (p_d*4.9 ph26):f2)
    (center (p_d*7.5 ph28):f1 (p_d*7.5 ph28):f2)
    (center (p_d*5.3 ph26):f1 (p_d*5.3 ph26):f2)
    (center (p_d*7.4 ph28):f1 (p_d*7.4 ph28):f2)

    (center (p_d*6.4 ph26):f1 (p_d*6.4 ph26):f2)
    (center (p_d*8.2 ph28):f1 (p_d*8.2 ph28):f2)
    (center (p_d*5.8 ph26):f1 (p_d*5.8 ph26):f2)
    (center (p_d*5.7 ph28):f1 (p_d*5.7 ph28):f2)
    (center (p_d*0.6 ph26):f1 (p_d*0.6 ph26):f2)
    (center (p_d*4.9 ph28):f1 (p_d*4.9 ph28):f2)
    (center (p_d*7.5 ph26):f1 (p_d*7.5 ph26):f2)
    (center (p_d*5.3 ph28):f1 (p_d*5.3 ph28):f2)
    (center (p_d*7.4 ph26):f1 (p_d*7.4 ph26):f2)
lo to 10 times 2

  1u pl1:f1

  (pwh one):f1

  "DELTA = 2u + p57 + d16 + 4u + 10u + 4u + de"
  DELTA

  (pwh*2 zero):f1

  2u
  p57:gp7                                          ; gradient7
  d16

#ifdef Ddec
  4u BLKGRAMP
#else
  4u BLKGRAD
#endif

  10u fq=0:f1          ; jump carrier to water
  4u pl21:f2           ; lower power for 13C decoupling

  go=2 ph31 cpds2:f2    ; acquire fid with 13C decoupling during aq

  d11 do:f2 mc #0 to 2
  F2QF(calclc(l2,+1) & calclist(pwrml1,1))
  F1EA(calgrad(EA) & calph(ph4,+180),caldel(d0,+in0) & calph(ph2,+180) & calph(ph31,+180))

#ifdef Ddec
d11 H2_LOCK
d11 LOCKH_OFF
; d11 LOCKDEC_OFF ; use statement for earlier hardware
#endif

HaltAcqu, 1m
exit

ph1=0 2
ph2=1 1 3 3
ph3=0 0 0 0 2 2 2 2
ph4=1
ph31=0 2 2 0
ph26=0
ph27=1
ph28=2
ph29=3
```

```
;pl1 : tpwr - power level for pwh
;pl10 : tsatpwr - power level for presat
;pl11 : tpwrmess - power level for Messerle purge
;pl12 : power level for 1H CW during CPMG
;pl13 : 1H power level for het cp back transfer
;pl17 : power level for 1H waltz decoupling during CT-t1
;pl2 : dhpwr - power level for 13C pulse pwc (p2)
;pl21 : dpwr - power level for  13C decoupling cpd2
;pl22 : power level for 13C het cp back transfer
;pl23 : power level for 13C CPMG pulses pwc_cp
;pl25 : power level for 13C calibration of p25
;pl26 : power level for 13C compensation 8 dB lower than pl23
;pl4  : power level for 2H high power pulses
;pl41 : power level for 2H waltz decoupling
;pl42 : power level for 2H for CW decoupling during 13C CPMG
;p1 : pwh
;pcpd1 : 1H pw for 1H Waltz-16 decoupling during CT-t1
;p12 : 1H pw for CW decoupling during CPMG ~ 15 us
;p13 : 1H pw90 for hetcp back transfer ~ 31.05 us 230.2*p13 = transfer time = 1/JCH
;p2 : pwc
;pcpd2 : 13C pulse width for 13C decoupling
;p23 : 13C pwc_cp
;p25 : 13C calibration 90 pulse at pl25
;pcpd4 : 2H pulse width for 2H decoupling
;p4 : 2H high power pulse
;p42 : 2H 90 for cw decoupling during CPMG
;t1_aq: net acquisition time in t1
;d1 : Repetition delay D1
;d4 : time_eq typically 3-5 ms
;d11 : delay for disk i/o, 30ms
;d13 : time_T2 CPMG duration <=40ms
;d14 : time_T2_max max <50ms
;d15 : taua ~1/(4*JCH)  ~1.8-2ms
;d16 : gradient recovery delay, 200us
;d18 : taub - set to 1/4JHC = 2.0 ms
;cpd1 : 1H CW decoupling during CPMG
;cpd2 : 13C decoupling during t2 according to program defined by cpdprg2
;cpd4 : 2H decoupling during t1
;cpd7 : 1H Waltz-16 decoupling during CT-t1
;inf1 : 1/SW(X) = 2*DW(X)
;in0:1/(2*SW(x))=DW(X)
;l0 : set internally do not adjust
;l2 : pointer to durations for 13C evolution during B1 calibn - internal do not touch
;l5 : set internally - ncyc_cp value
;l6 : set internally do not adjust
;l8 : set to ncyc_max for 13C compensation if C_comp is defined
;cpdprg1 : 1H CW decoupling during CPMG -  p12 is the 90o pulse width
;cpdprg2 : 13C decoupling during t2
;cpdprg4 : 2H decoupling during t1
;cpdprg7 : 1H Waltz-16 decoupling during CT-t1 uses pcpd1 for 90o pulse width
;cnst1 : offset of methyls from water (0.8 ppm - 4.7 ppm, in Hz)
;cnst4 : power level for 2H high power pulses
;cnst12 : power in w for 1H CW field during CPMG
;cnst13 : 1H power in w for hetCP
;cnst17 : power in w for 1H Waltz during CT-t1
;cnst21 : power in w for 13C dec
;cnst23 : power in w for pwc_cp
;cnst26 : power in w for 13C compensation - scales with ncyc
;cnst27 : power for 13C for het cp
;cnst41 : power level for 2H decoupling
;nd0=2
;zgoptns : Df1180 Dfsat Dmess_flg Dfscuba DC_calib DOneDarray DDdec
```

```
##TITLE= Parameter file, TOPSPIN                Version 3.2
##JCAMPDX= 5.0
##DATATYPE= Parameter Values
##NPOINTS= 5    $$ modification sequence number
##ORIGIN= Bruker BioSpin GmbH
##OWNER= nmrsu
$$ 2015-10-08 18:56:34.688 -0400  nmrsu@UOFT600
$$ /opt/topspin3.2/exp/stan/nmr/par/user/13CHD2_13C_CPMG_lek_600_cp/acqu
$$ process /opt/topspin3.2/prog/mod/edparproc
##$ACQT0= 0
##$AMP= (0..31)
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100
##$AMPCOIL= (0..19)
-1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$ANAVPT= 1
##$AQSEQ= 0
##$AQ_mod= 3
##$AUNM= <au_zgonly>
##$AUTOPOS= <>
##$BF1= 600.25
##$BF2= 150.93258
##$BF3= 60.822804
##$BF4= 92.14203
##$BF5= 600.25
##$BF6= 600.25
##$BF7= 600.25
##$BF8= 600.25
##$BYTORDA= 0
##$CAGPARS= (0..11)
0 0 0 0 0 0 0 0 0 0 0 0
##$CFDGTYP= 0
##$CFRGTYP= 9
##$CHEMSTR= <none>
##$CNST= (0..63)
-2400 -2340 1 1 90 1 1 1 1 1 1 1 1 1 1 1 1 1 1 25 0 0 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##$CPDPRG= (0..8)
<> <cw.cpd> <waltz16> <onepulse.sp31> <waltz16> <> <onepulse.sp26> <waltz16>
<p5m4sp180.p22>
##$D= (0..63)
0 2 0 0 0.003 0 0 0 0 0 0 0 0.005 0.04 0.041 0.0018 0.0002 0.014 0.002
0 0 0 0 0 0.002777778 0 0 0.012 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$DATE= 0
##$DE= 25
##$DECBNUC= <off>
##$DECIM= 2080
##$DECNUC= <off>
##$DECSTAT= 7
##$DIGMOD= 1
##$DIGTYP= 12
##$DQDMODE= 0
##$DR= 22
##$DS= 512
##$DSPFIRM= 0
##$DSPFVS= 0
##$DTYPA= 0
##$EXP= <13CHD2_13C_CEST_lek_600_v2>
##$F1LIST= <>
##$F2LIST= <>
##$F3LIST= <>
##$FCUCHAN= (0..9)
0 2 1 3 4 0 0 0 0 0
##$FL1= 0
##$FL2= 0
##$FL3= 0
##$FL4= 0
##$FOV= 0
##$FQ1LIST= <C_cest_fq40>
```

##$FQ2LIST= <>
##$FQ3LIST= <>
##$FQ4LIST= <>
##$FQ5LIST= <>
##$FQ6LIST= <>
##$FQ7LIST= <>
##$FQ8LIST= <>
##$FRQLO3= 0
##$FRQLO3N= 0
##$FS= (0..7)
83 83 83 83 83 83 83 83
##$FTLPGN= 0
##$FW= 125000
##$FnMODE= 0
##$FnTYPE= 0
##$GPNAM= (0..31)
<SMSQ10.32> <SMSQ10.32> <SMSQ10.32> <SMSQ10.32> <SMSQ10.32> <SMSQ10.32>
<SMSQ10.32> <SMSQ10.32> <SMSQ10.32> <> <> <> <> <SMSQ10.100> <SMSQ10.100>
<> <> <> <> <> <> <> <> <> <> <> <> <> <> <> <> <>
##$GPX= (0..31)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$GPY= (0..31)
0 0 0 0 0 0 90 90 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$GPZ= (0..31)
40 30 50 -50 -30 -30 0 0 40 0 0 0 0 15 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0
##$GRDPROG= <>
##$GRPDLY= -1
##$HDDUTY= 20
##$HDRATE= 1
##$HGAIN= (0..3)
0 0 0 0
##$HL1= 0
##$HL2= 0
##$HL3= 0
##$HL4= 0
##$HOLDER= 0
##$HPMOD= (0..7)
0 0 0 0 0 0 0 0
##$HPPRGN= 0
##$IN= (0..63)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$INF= (0..7)
0 368 37 0 0 0 0 0
##$INP= (0..63)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$INSTRUM= <>
##$L= (0..31)
1 24 16 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##$LFILTER= 10
##$LGAIN= -10
##$LINPSTP= 0
##$LOCKED= no
##$LOCKFLD= 0
##$LOCKGN= 0
##$LOCKPOW= -20
##$LOCKPPM= 0
##$LOCNUC= <2H>
##$LOCPHAS= 0
##$LOCSHFT= no
##$LOCSW= 0
##$LTIME= 0.1
##$MASR= 4200
##$MASRLST= <masrlst>
##$MULEXPNO= (0..15)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$NBL= 1
##$NC= 0

```
##$NLOGCH= 4
##$NOVFLW= 0
##$NS= 8
##$NUC1= <1H>
##$NUC2= <13C>
##$NUC3= <15N>
##$NUC4= <2H>
##$NUC5= <off>
##$NUC6= <off>
##$NUC7= <off>
##$NUC8= <off>
##$NUCLEUS= <off>
##$NUSLIST= <automatic>
##$NusAMOUNT= 25
##$NusFPNZ= no
##$NusJSP= 0
##$NusSEED= 54321
##$NusT2= 1
##$NusTD= 0
##$O1= 2824
##$O2= 3018.65
##$O3= 7237.9
##$O4= 73.71
##$O5= 2823.24
##$O6= 2823.24
##$O7= 2823.24
##$O8= 2823.24
##$OVERFLW= 0
##$P= (0..63)
0 10.75 12.85 39.5 152 0 0 0 0 0 0 1000 15 31 7000 0 1000 0 0 0 0 120.5
14.5 15 360 14.9 23998 0 0 0 0 26325 0 0 0 0 0 0 0 0 0 0 250 0 0 0 0 0
0 0 1000 500 800 1000 700 900 1024 256 300 0 0 0 0 0
##$PACOIL= (0..15)
1 2 3 4 0 0 0 0 0 0 0 0 0 0 0 0
##$PAPS= 0
##$PARMODE= 2
##$PCPD= (0..9)
0 40 150 225 350 0 0 40 0 0
##$PEXSEL= (0..9)
1 1 1 1 1 1 1 1 1 1
##$PHCOR= (0..31)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$PHLIST= <>
##$PHP= 1
##$PH_ref= 0
##$PL= (0..63)
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120
##$PLSTEP= 0.1
##$PLSTRT= -6
##$PLW= (0..63)
0 9 90.36494 0 12.5603 0 0 0 0 0 3.162278e-05 0 4.466836 1.071519 0.0006309574
0 0 0.6309574 0 0 0 0.6854882 16.03246 68.54884 0 68.54884 1.122018 0 0
0 0 0.9036494 0 0 0 0 0 0 0 0 0 2.285599 4.570882 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
##$PLWMAX= (0..7)
0 0 0 0 0 0 0 0
##$POWMOD= 0
##$PQPHASE= 0
##$PQSCALE= 1
##$PR= 1
##$PRECHAN= (0..15)
-1 4 0 1 5 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
##$PRGAIN= 0
##$PROBHD= <>
##$PULPROG= <13CHD2_13C_CPMG_lek_600_cp>
##$PW= 0
##$PYNM= <>
```

##$QNP= 0
##$RD= 0
##$RECCHAN= (0..15)
0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$RECPH= 0
##$RECPRE= (0..15)
-1 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
##$RECPRFX= (0..15)
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$RECSEL= (0..15)
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
##$RG= 203
##$RO= 0
##$ROUTWD1= (0..23)
0 100 43 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
##$ROUTWD2= (0..23)
0 1 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$RSEL= (0..15)
0 1 2 3 7 0 0 0 0 0 0 0 0 0 0 0
##$S= (0..7)
83 83 83 83 83 83 83 83
##$SELREC= (0..9)
0 0 0 0 0 0 0 0 0 0
##$SFO1= 600.252824
##$SFO2= 150.93559865
##$SFO3= 60.8300419
##$SFO4= 92.14210371
##$SFO5= 600.25282324
##$SFO6= 600.25282324
##$SFO7= 600.25282324
##$SFO8= 600.25282324
##$SOLVENT= <D2O>
##$SOLVOLD= <off>
##$SP= (0..63)
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120
120 120 120 120 120 120 120 120 120 120
##$SPECTR= 0
##$SPINCNT= 0
##$SPNAM= (0..63)
<> <Sinc1.1000> <Q5.1000> <Q3.1000> <> <Q3.1000> <> <> <Q5tr.1000> <> <>
<> <Squa100.1000> <Squa100.1000> <eburp1.BRF> <seduce.100> <> <> <> <>
<> <> <reburp400pts.BRF> <chirp_80kHz.BRF> <reb_4p_UC.BRF> <> <UC_CO_CB.DEC>
<> <> <> <> <UC_N_RK.DEC> <> <> <> <> <> <> <> <> <> <> <> <> <> <> <>
<> <> <> <> <> <> <> <> <> <> <> <> <> <> <> <> <>
##$SPOAL= (0..63)
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0 0 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0 0 0 0.5 0 0.5 0.5 0.5 0.5 0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5
0.5 0.5 0.5 0.5 0.5 0.5
##$SPOFFS= (0..63)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$SPPEX= (0..63)
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##$SPW= (0..63)
0 0 0 0 0 0 0 0 0 0 0 0 0 0.0007943282 0.006309574 0 0 0 0 0 0 0 0 70.79458
142.8894 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
##$SUBNAM= (0..9)
<> <> <> <> <> <> <> <> <> <>
##$SW= 16.0188910921052
##$SWIBOX= (0..19)
0 1 2 4 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0
##$SW_h= 9615.38461538462
##$SWfinal= 0
##$TD= 1228
##$TD0= 1

```
##$TE= 300
##$TE1= 300
##$TE2= 300
##$TE3= 300
##$TE4= 300
##$TEG= 300
##$TE_PIDX= 0
##$TE_STAB= (0..9)
0 0 0 0 0 0 0 0 0 0
##$TL= (0..7)
120 120 120 120 120 120 120 120
##$TUNHIN= 0
##$TUNHOUT= 0
##$TUNXOUT= 0
##$USERA1= <>
##$USERA2= <>
##$USERA3= <>
##$USERA4= <>
##$USERA5= <>
##$V9= 5
##$VALIST= <pwr_tst2>
##$VCLIST= <ncyc_list_2H>
##$VD= 0
##$VDLIST= <>
##$VPLIST= <>
##$VTLIST= <>
##$WBST= 1024
##$WBSW= 2
##$XGAIN= (0..3)
0 0 0 0
##$XL= 0
##$YL= 0
##$YMAX_a= 0
##$YMIN_a= 0
##$ZGOPTNS= <-DDdec >
##$ZL1= 120
##$ZL2= 120
##$ZL3= 120
##$ZL4= 120
##END=
```

```
##TITLE= Parameter file, TOPSPIN                Version 3.2
##JCAMPDX= 5.0
##DATATYPE= Parameter Values
##NPOINTS= 5    $$ modification sequence number
##ORIGIN= Bruker BioSpin GmbH
##OWNER= nmrsu
$$ 2015-10-08 18:56:34.690 -0400  nmrsu@UOFT600
$$ /opt/topspin3.2/exp/stan/nmr/par/user/13CHD2_13C_CPMG_lek_600_cp/acqu2
$$ process /opt/topspin3.2/prog/mod/edparproc
##$ACQT0= 0
##$BF1= 600.25
##$FnMODE= 1
##$NUC1= <off>
##$NusJSP= 0
##$NusT2= 1
##$NusTD= 0
##$O1= 2823.24
##$SFO1= 600.25282324
##$SW= 45.0011551355258
##$SW_h= 6792.27629032211
##$TD= 43
##END=
```

##TITLE= Parameter file, TOPSPIN   Version 3.2
##JCAMPDX= 5.0
##DATATYPE= Parameter Values
##NPOINTS= 5 $$ modification sequence number
##ORIGIN= Bruker BioSpin GmbH
##OWNER= nmrsu
$$ 2015-10-08 18:56:34.690 -0400 nmrsu@UOFT600
$$ /opt/topspin3.2/exp/stan/nmr/par/user/13CHD2_13C_CPMG_lek_600_cp/acqu3
$$ process /opt/topspin3.2/prog/mod/edparproc
##$ACQT0= 0
##$BF1= 150.93258
##$FnMODE= 6
##$NUC1= <13C>
##$NusJSP= 0
##$NusT2= 1
##$NusTD= 0
##$O1= 3018.65
##$SFO1= 150.93559865
##$SW= 18
##$SW_h= 2716.8407757
##$TD= 100
##END=