Phase Cycles

Phase cycling for PE-WATERGATE (Fig. 1a of the manuscript). Phases are notated as multiples of 90° (0 = 0°, 1 = 90°, 2 = 180°, 3 = 270°), with subscripts denoting repetition; the minimum phase cycle is eight transients.

$\Phi_1 = 02$
$\Phi_2 = 2_2 3_2 0_2 1_2 + 0_{32} 2_{32}$
$\Phi_{2'} = \Phi_2 + 2 = 0_2 1_2 2_2 3_2 + 0_{32} 2_{32}$
$\Phi_3 = 0_8 1_8 2_8 3_8$
$\Phi_4 = 1_{64} 3_{64}$
$\Phi_R = \Phi_1 + 2\Phi_2 + 2\Phi_3 = 0_8 2_8 + 0_2 2_2 + 02$

Additionally, for PE-ES-WATERGATE (Fig. 1b of the manuscript).

$\Phi_{3'} = \Phi_3 + 2 = 2_8 3_8 0_8 1_8$

Excitation Profile

Wolfram Mathematica 7.0 code for calculation of excitation profile of 180_{-x} hard pulse flanked by two 90_{x} selective pulses of duration `pw90shp` using a density matrix approach for a single spin-½.

```
Id[0] = {{1, 0}, {0, 1}};
Id[1] = {{0, 0.5}, {0.5, 0}};
Id[2] = {{0, -0.5*I}, {0.5*I, 0}};
Id[3] = {{0.5, 0}, {0, -0.5}};
\[Phi]i[a_] := Sin[(Pi/2)*Mod[a, 4]]*Id[2] + Cos[(Pi/2)*Mod[a, 4]]*Id[1];
Pulse[\[Nu]i_, RF_, \[Phi]_] := -(\[Nu]i*Id[3]) - RF*\[Phi]i[\[Phi]];
CalcPulse[\[Nu]i_, RF_, \[Phi]_][t_][\[Rho]_] := MatrixExp[-(I*2*Pi*Pulse[\[Nu]i, RF,
\[Phi]]*t)].\[Rho].MatrixExp[I*2*Pi*Pulse[\[Nu]i, RF, \[Phi]]*t];
\[Omega]1[pdur_] := 1/4/pdur
pw90 = 10/10^6;
RFhp = \[Omega]1[pw90];
pw90shp = 10.5/10^3;
RFshp = \[Omega]1[pw90shp];
\[Nu]i = 0;
Offsets = 1000;
OffRange = 10000;
a = 3;
For[i = 1, i < Offsets + 2, i++, spoffs = (i - 1)*(OffRange/(Offsets - 1)) -
Offsets*(OffRange/(Offsets - 1)/2);
\[Sigma] = -2 Id[3];
\[Sigma] = CalcPulse[\[Nu]i - spoffs, RFshp, 0][pw90shp][\[Sigma]];
\[Sigma] = CalcPulse[\[Nu]i - spoffs, RFhp, 2][pw90*2][\[Sigma]];
\[Sigma] = CalcPulse[\[Nu]i - spoffs, RFshp, 0][pw90shp][\[Sigma]];
MagOut[i] = \[Sigma];
Res[i] = spoffs; ];
Print["Excitation Profile:"]
ListPlot[Table[{Res[i], Re[0.5*(1 + Tr[Id[3] . MagOut[i]])]}, {i, 1, Offsets + 1}], PlotRange
-> {{-720 + 1880, -2160 + 1880}, {0, 1}}, Joined -> True]
```

Pulse Sequence

```
/*
PEWGES.c
Perfect Echo WATERGATE sequence with soft pulses and optional excitation sculpting.

Ralph W. Adams, Chloe M. Holroyd, Juan A. Aguilar, Mathias Nilsson, and Gareth A. Morris
University of Manchester, Manchester, UK. July 2012
Varian/Agilent pulse sequence code for information only,
The University of Manchester and the authors cannot be held responsible
for any damage or loss resulting from the use of this sequence.

Timing:     |<   t/4   >|<   t/4   >|<   t/4   >|<   t/4   >|
1H:      90x      [S90]180[S90]   {90y}    S90 180 S90       Acquire
PFG:        G'                 G'      G             G

pe flag: {}
es flag: []
G':      0.379*G
S90:     Long duration (c. 10 ms), low power pulse
90/180:  Short duration (c. 8,16 us), high power pulse
t:       Total echo time, tJ << 1 for true perfect echo

gstab          - gradient stabilization delay
gt2            - gradient duration for the solvent suppression echo
gzlvl1         - gradient power for the solvent suppression echo
selpw          - power of the soft pulse
phaseinc       - increments between the phases for the transmitter
stepsize       - sets the phase of the transmitter to steps of 0.5 deg
xtmrphase      - sets the phase of the transmitter to phaseinc
txphase        - sets the phase of the transmitter to phasetable
selpwr         - power for the soft pulses
selpw          - duration of soft pulses
pe             - flag to use perfect echo
es             - flag to use excitation sculpting
dse            - flag to use a double spin echo, for debugging
*/

#include <standard.h>

pulsesequence()
{
/*Set Variables*/
static int     ph1[2]   = {0,2},
                       ph2[32]  =
{0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3},
                       ph21[32] =
{2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1},
                       ph3[128] =
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3},
               ph4[32]  = {0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3},
                       ph5[64]  =
{2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3,0,0,1,1,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3
,3,0,0,1,1,2,2,3,3,0,0,1,1,2,2,3,3},
               rec[16]  = {0,2,2,0,0,2,2,0,2,0,0,2,2,0,0,2},
               rec2[16] = {0,2,0,2,0,2,0,2,2,0,2,0,2,0,2,0};
double         gstab    = getval("gstab"),
               gt2      = getval("gt2"),
                       gzlvl1   = getval("gzlvl1"),
                       selpwr   = getval("selpwr"),
                       selpw    = getval("selpw"),
                       phaseinc = getval("phaseinc");
char           pe[MAXSTR],
                       es[MAXSTR],
                       dse[MAXSTR];
rof1 = getval("rof1"); if(rof1 > 2.0e-6) rof1 = 2.0e-6;
getstr("pe",pe);
getstr("dse",dse);
getstr("es",es);
settable(t1,2,ph1);
settable(t2,32,ph2);
settable(t21,32,ph21);
settable(t3,128,ph3);
settable(t4,32,ph4);
settable(t5,64,ph5);
```

```
settable(t6,16,rec);
settable(t7,16,rec2);
/*Start Sequence*/
status(A);
        delay(d1);
status(B);
        obspower(tpwr);
        rgpulse(pw, t1, rof1, rof1);
        if (es[A] == 'y' ) {
                zgradpulse(gzlvl1*0.379,gt2);
                delay(gstab-rof1-rof1-rof1-rof1);
                if (phaseinc < 0.0) {
                        phaseinc = 1440+phaseinc;
                }
                stepsize(0.25,OBSch);
                initval(phaseinc,v1);
                obspower(selpwr);
                xmtrphase(v1);
                txphase(t21);
                rgpulse(selpw,t21,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                txphase(t2);
                rgpulse(pw*2.0,t2,rof1,rof1);
                obspower(selpwr);
                xmtrphase(v1);
                txphase(t21);
                rgpulse(selpw,t21,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                zgradpulse(gzlvl1*0.379,gt2);
                delay(gstab);
        }
        if ((pe[A] == 'y'|dse[A] == 'y') && es[A] != 'y'){
                zgradpulse(gzlvl1*0.379,gt2);
                delay(selpw+gstab-rof1-rof1);
                rgpulse(pw*2.0,t2,rof1,rof1);
                zgradpulse(gzlvl1*0.379,gt2);
                delay(selpw+gstab-rof1-rof1);
        }
        if (pe[A] == 'y'){
        /* Use 90d refocussing pulse */
                rgpulse(pw,t3,rof1,rof1);
        }
        if (pe[A] == 'y'|dse[A] == 'y'|es[A] == 'y'){
                zgradpulse(gzlvl1,gt2);
                delay(gstab-rof1-rof1-rof1-rof1);
                if (phaseinc < 0.0){
                        phaseinc = 1440+phaseinc;
                }
                stepsize(0.25,OBSch);
                initval(phaseinc,v1);
                obspower(selpwr);
                xmtrphase(v1);
                txphase(t4);
                rgpulse(selpw,t4,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                txphase(t5);
                rgpulse(pw*2.0,t5,rof1,rof1);
                obspower(selpwr);
                xmtrphase(v1);
                txphase(t4);
                rgpulse(selpw,t4,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                delay(gstab/2.0);
                zgradpulse(gzlvl1,gt2);
                delay(gstab/2.0);
                setreceiver(t6);
        }
        if (pe[A] != 'y' && dse[A] != 'y' && es[A] != 'y'){
                zgradpulse(gzlvl1,gt2);
                delay(gstab-rof1-rof1-rof1-rof1);
                if (phaseinc < 0.0){
                        phaseinc = 1440+phaseinc;
                }
```

```
                stepsize(0.25,OBSch);
                initval(phaseinc,v1);
                xmtrphase(v1);
                txphase(t21);
                rgpulse(selpw,t21,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                txphase(t2);
                rgpulse(pw*2.0,t2,rof1,rof1);
                obspower(selpwr);
                xmtrphase(v1);
                txphase(t21);
                rgpulse(selpw,t21,rof1,rof1);
                obspower(tpwr);
                xmtrphase(zero);
                delay(gstab/2.0);
                zgradpulse(gzlvl1,gt2);
                delay(gstab/2.0);
                setreceiver(t7);
        }
status(C);
}
```