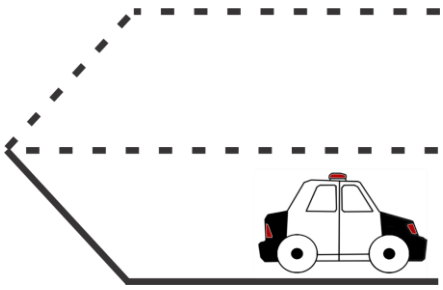
The use of the MATLAB executable implementation of Dephasing Optimization Through Coherence Order Pathway Selection (DOTCOPS)



1. Basic use for determining relative amplitude of crusher gradients

DOTCOPS is an algorithm designed to create a crusher scheme or a phase cycling for an arbitrary magnetic resonance spectroscopy pulse sequence, where the user inputs the desired coherence pathway(s) which should be selected and how many coupled spins are of interest. The output from DOTCOPS will be a crusher scheme which crushes all unwanted coherence pathway(s) and does not dephase the desired coherence pathway. We provide the MATLAB executable "DOTCOPS.exe", as well as the README which details how to ensure that the MATLAB executable can be run, and a sample .mat file of input (DOTCOPSinput.mat). After reading the README.txt file and ensuring that the appropriate MATLAB runtime environment is installed the user should navigate to the folder through the command terminal that contains DOTCOPS. The input .mat file must contain the two basic input variables named selectedPathways and maximumCoherenceOrder. To make this .mat file simply write in the MATLAB shell:

```
>>selectedPathways = [-1 1 -1]; maximumCoherenceOrder = 1;
```

Then press CTRL+s and save the file as "DOTCOPSinput.mat." One should choose their appropriate selected coherence pathway, the example given here is for PRESS and only considers a maximum coherence order of 1. In general, selectedPathways is a matrix of size MxN, where M is the number of coherence pathway(s) to be selected (often 1) and N is the number of RF pulses in your experiment (e.g., 3 for PRESS/STEAM, 5 for MEGA-PRESS/sLASER, 7 for MEGA-sLASER, 9 for MEGA-LASER, etc). For a list of the coherence pathways associated with the aforementioned pulse sequences see the publication for this algorithm. The maximumCoherenceOrder is the maximum coherence order that will be considered in the coherence pathways, this could be set to the number of coupled spins (i.e., 1 for uncoupled spins, 2 for 2 coupled spins, etc). Note that the number of coherence pathways grows rapidly with increasing this number, so for some sequences with 8+ RF pulses it may not be possible to crush all unwanted coherence pathways for coupled systems with 6 or more spins. Running DOTCOPS will output a matrix of size 3xN. The 3 is the three orthogonal gradient axes, and the 1st column would be the relative gradient amplitude in-between the 1st and 2nd RF pulse, the 2nd column is the relative gradient amplitude in-between the 2nd and 3rd RF pulse, etc, and the Nth column is the relative gradient amplitude between the Nth (last) RF pulse before signal acquisition.

Once the .mat input file has been created to run DOTCOPS simply type “DOTCOPS” either in the command terminal or MATLAB shell. Example output is given for DOTCOPS which has an input of selectedPathways = [-1 1 -1] and maximumCoherenceOrder = 1. Running DOTCOPS with this DOTCOPSinput.mat produces the result:

```
The optimized areas which should be played out are:

areas =

   -1.0000    0.0000    1.0000
   -0.0000    1.0000    1.0000
    1.0000    1.0000    0.0000


optimizedAmplitudes =

   -1.0000    0.0000    1.0000
   -0.0000    1.0000    1.0000
    1.0000    1.0000    0.0000

All the uncrushed pathways are

uncrushedPathways =

    -1     1    -1

The least crushed unwanted pathway is crushed by: 1.4142
The number of unwanted uncrushed pathways is: 0
The average unwanted pathway is crushed by: 2.2034

optimizedAmplitudes =

   -1.0000    0.0000    1.0000
   -0.0000    1.0000    1.0000
    1.0000    1.0000    0.0000
```

In this case the optimized areas (“areas”) are the same as the optimized amplitudes because all crusher durations are equal. The optimized amplitudes are what would be set in the scanner’s hardware, and it also informs the user that the least crushed pathway is crushed by $\sqrt{2}$ arbitrary units of crushing (i.e., 1 crusher has a unit of 1 in this case), and the average of the 8 unwanted coherence pathways is crushed by 2.2034.

2. “Advanced” use of DOTCOPS for crushers

There are also a few additional input parameters which can be input to DOTCOPS. The additional inputs that can be modified are: the upper bound on the crushers, which can be used to accommodate multiple gradients or gradients of an arbitrary duration, the number of iterations, and the regularization parameter as described in the associated manuscript. This is run identically as before; the only difference is in DOTCOPSinput.mat there needs to be a struct variable called “specifiedParameters”. Specified parameters can have any of the following fields: specifiedParameters.ub, specifiedParameters.lambda, specifiedParameters.numberOfIterations, speifiedParameters.spacings. The field specifiedParameters.ub can be set to be a matrix of size 3xN which represents the relative upper bounds of the gradients. This is useful for, example, when the gradients are not all of equal duration or when multiple gradients are played out during a single inter-RF pulse delay (such as the case of PRESS where it is typical to play two crushers in-between the 2nd and 3rd RF pulse). The field

specifiedParameters.lambda is the regularization parameter as described in the publication, a value of 1 is recommended for pulse sequences which have a small number of RF pulses (less than 9), and a low maximum coherence order (3 or less). For cases with a large number of RF pulses / max coherence order it would be recommended to use a lambda = 0, however this is a parameter that the user can play with if looking to tradeoff between the average pathway crushing and the minimum pathway crushing. The field specifiedParameters.numberOfIterations is the number of iterations the algorithm runs through. For large number of RF pulses the default of 10,000 iterations can take a long time, and it may be useful to reduce this to a more feasible number like 100 or 1000. A minimum of 2 iterations is required. Lastly, the field specifiedParameters.scalings is the relative scaling or weight of crushing applied to each of the measured coherence pathways.

The default parameters for these fields are:

>>specifiedParameters.ub = ones(3,N),
>>speicfiedParameters.lambda = 1,
>>specifiedParameters.numberOfIterations = 10000,
>>specifiedParameters.weightings = ones(1,(2*p+1)^(N-1)).

Crushing with non-equal duration or amplitude of crushers

If specifiedParameters is not used or any field is not specified these default parameters are used. For example, if we set specifiedParameters.ub = [1 2 1; 1 2 1; 1 2 1] (such as the case in normal PRESS), and re-run the previous example where selectedPathways = [-1 1 -1] and maximumCoherenceOrder = 1 we get the result:

```
areas =

    1.0000    2.0000    1.0000
   -1.0000    0.0000    1.0000
    1.0000    2.0000    1.0000


optimizedAmplitudes =

    1.0000    1.0000    1.0000
   -1.0000    0.0000    1.0000
    1.0000    1.0000    1.0000

All the uncrushed pathways are

uncrushedPathways =

    -1     1    -1

The least crushed unwanted pathway is crushed by: 1.7321
The number of unwanted uncrushed pathways is: 0
The average unwanted pathway is crushed by: 3.1546

optimizedAmplitudes =

    1.0000    1.0000    1.0000
   -1.0000    0.0000    1.0000
    1.0000    1.0000    1.0000
```

In this case the optimized areas and the optimized amplitudes are not the same, because the duration of the 2nd crusher is twice that of the 1st or 3rd crusher. Once again the amplitude is what would be input into the hardware. It can also be seen that the minimum uncrushed pathway is $\sqrt{3}$ now and the average uncrushed pathway is 3.1546, about 40% larger than the case where all crusher durations were equal. Note that this is how a crushing scheme with crushers of arbitrary duration can be accommodated into DOTCOPS.

Weightings

To preferentially weight certain pathways one must provide a row vector of size 1 by $(2p+1)^{n-1}$ which provides the relative additional crushing power. To find all the pathways run the script (which is also included in the DOTCOPS.zip) in the MATLAB shell terminal:

```
[minusOnePathways] =
generateMinusOneCoherencePathways(numberOfPulses,maxCoherenceOrder)
```

For example, for uncoupled spins for 3 pulses this provides the 9 pathways:

```
>> [minusOnePathways] = generateMinusOneCoherencePathways(3,1)

minusOnePathways =

    -1    -1    -1
    -1     0    -1
    -1     1    -1
     0    -1    -1
     0     0    -1
     0     1    -1
     1    -1    -1
     1     0    -1
     1     1    -1
```

Additional weightings to the coherence pathways can be provided if the user suspects that one (or more) specific pathways are particularly prone to causing spurious echoes. To provide additional crushing to, for example, the first pathway, [-1, -1, -1], a parameter called “weightings” needs to be input as follows:

>>specifiedParameters.scalings = [2 1 1 1 1 1 1 1 1];

This indicates that the first coherence pathway will receive twice as much weighting in the crushing than the other 7 unwanted pathways. The order must be consistent with the matrix of measured coherence pathways. For example, if the user wanted instead to increase the crushing to the coherence pathway [1, 0, -1] by a factor of 2 then they would set

>>specifiedParameters.scalings = [1 1 1 1 1 1 1 2 1].

3. Use for determining phase cycling schemes

The only flag that needs to be set for DOTCOPS to designed an optimized phase cycling scheme is specifiedParameters.phaseCycleSteps. Optionally one can input specifiedParameters.lambda2, which is described in the second DOTCOPS mansuscript. For example if you wanted DOTCOPS to design a 32 step phase cycling scheme you would simply set

```
>>specifiedParameters.phaseCycleSteps = 32;
```

DOTCOPS will then spit out the following information:

```
cogString =

COG8(0,0,0,1,0,1,0;4)

The phase scheme is cogwheel (1st column is 1st RF pulse, 2nd is 2nd RF pulse, etc., last column is rec

ans =

  Columns 1 through 5

       0              0              0              0              0
       0              0              0            1/4              0
       0              0              0            1/2              0
       0              0              0            3/4              0
       0              0              0              1              0
       0              0              0            5/4              0
       0              0              0            3/2              0
       0              0              0            7/4              0

  Columns 6 through 8

       0              0              0
       1/4            0              1
       1/2            0              0
       3/4            0              1
       1              0              0
       5/4            0              1
       3/2            0              0
       7/4            0              1
```

```
The number of pathways killed by this phase cycling scheme was 711 of the 728 unwanted pathways

pathways =

    -1    -1    -1     1    -1     1    -1
    -1    -1     1    -1     1    -1    -1
    -1     0    -1     1    -1     1    -1
    -1     0     1    -1     1    -1    -1
    -1     1    -1     1    -1     1    -1
    -1     1     1    -1     1    -1    -1
     0    -1    -1     1    -1     1    -1
     0    -1     1    -1     1    -1    -1
     0     0    -1     1    -1     1    -1
     0     0     1    -1     1    -1    -1
     0     1    -1     1    -1     1    -1
     0     1     1    -1     1    -1    -1
     1    -1    -1     1    -1     1    -1
     1    -1     1    -1     1    -1    -1
     1     0    -1     1    -1     1    -1
     1     0     1    -1     1    -1    -1
     1     1    -1     1    -1     1    -1
     1     1     1    -1     1    -1    -1

All the uncrushed pathways considering both phase cycling and crushers are

uncrushedPathways =

    -1     1    -1     1    -1     1    -1

The least crushed unwanted pathway left after phase cycling is crushed by: 1.06
The number of unwanted uncrushed pathways considering both crushing and phase cycling is: 0
The average unwanted pathway left after phase cycling is crushed by: 2.829

optimizedAmplitudes =

   -1.0000   -1.0000    1.0000    0.5000   -0.5000    1.0000    1.0000
   -1.0000    0.0600   -0.9400   -1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000         0    1.0000   -1.0000   -1.0000    1.0000
```

4. Other flags to be used

Using DOTCOPS for phase cycling only

If you have a crusher scheme and you are not interested modifying it, you can input this into DOTCOPS and it will consider your crusher scheme in optimizing the phase cycling scheme. The name of the variable is “crusherScheme”, e.g.,

```
>>specifiedParameters.crusherScheme = [-1 -1 1; -1 0.06 1; 1 -0.94 0; 0.5 -1 1; -0.5 1 -1; 1 1 -1; 1 1 1]';
```

The size of crusherScheme must be 3 by number of pulses (in this case 7).

Using DOTCOPS for nested phase cycling only

If you wish to only consider only nested phase cycling schemes set this flag:

```
>>specifiedParameters.nested = 1;
```

Using DOTCOPS for cogwheel phase cycling only

Whereas if you wish to only consider cogwheel phase cycling schemes set this flag:

```
>>specifiedParameters.cogwheel = 1;
```

Make sure you do not have both specifiedParameters.cogwheel and specifiedParameters.nested equal to 1 as this makes no sense. By default, DOTCOPS considers both nested and cogwheel phase cycling.

If you want to use a modified version of DOTCOPS for something more specified, please contact the Juchem Lab. For any questions, bugs, or suggestions please email Christoph Juchem, cwj2112@columbia.edu