

### COSY – COrrrelation SpectroscopY

The COSY experiment is good for determining the basic connectivity via J-couplings (through-bond).

Please note that all **bold print** is input at the prompt

#### Setting up the experiment and acquiring:

- Log into the NMR, start VNMR and join experiment #1 (type **jexp1**)
- Load the desired <sup>1</sup>H parameters including the correct solvent and type **su**
- Set the cooling gas preconditioner temperature to 15° C (if you are going to regulate at 25° C, then you can leave the preconditioner at 20° C). Make sure the VT gas is flowing 10 and 15 lpm and set the temperature to 20° C by typing **temp=20 su**. Allow between 10 and 15 minutes for the sample to reach thermal equilibrium.
- Open acqi window, insert the sample, lock, and shim as usual. As with most 2D NMR experiments, do not spin the sample. Adjust the z, z2, x, y, xz, yz, and xy shims.
- Close acqi window and tune the <sup>1</sup>H channel of the probe (channel #1).
- Collect and process a 1D proton. Type **ds** and position the cursors around all the peaks in the spectrum and type **movesw** (move spectral window according to cursors).
- Turn autogain off by typing **gain='y'** and recollect the 1D proton.
- Set the reference mark by placing the cursor on the peak then type **nl rl(7.27p)** (if your solvent is chloroform)
- Save the 1D by typing **svf** and answering the prompts. Move the fid from the current experiment #1 to a new experiment (say #2) by typing **mf(1,2)**; if experiment #2 does not exist, type **cexp(2)**
- Join experiment #2 and reprocess the data by typing **jexp2 wft**
- Load the gCOSY 2D parameters by typing **gcosy**
- Set the receiver gain by typing **set2dgain**. This macro is the same as typing **r1=nt r2=ni r3=ss wexp='set2Dgain2' nt=1 phase=1 ni=1 ss=1 gain='n' au** where 'set2Dgain' is another macro w/ **nt=r1 ni=r2 ss=r3 phase=1,2 wexp=' gain='y' if (gain > 3) then gain=gain-3 endif**.
- Typing **short2d** (macro w/ **ni=32, phase=1,2, nt=8, ss=200, gain='y', time**), **medium2d** (macro w/ **ni=128, phase=1,2, nt=16, ss=256, time**), or **long2d** (macro w/ **ni=256, phase=1,2, nt=32, ss=256, time**) will adjust nt (number of transients) and ni (number of increments) for you automatically.
- Set the following parameters: type **d1=1 fn=2k fn1=2k** – if your sample relaxes slowly, then set d1 to a larger value like 1.5, 2, or 3 (seconds).
- After you adjust your parameters, you can type **time** to give an approximate estimate of how long the experiment will run.
- If you want to use a squared, shifted sine bell for your apodization function in both dimensions, type **ssb ssb1** (otherwise you can use **wti** after the first slice has been collected). Gaussian (the default) functions are also very popular.
- Type **au** (submit experiment to acquisition and process data) to start the run.
- When the first FID has been collected (the acquisition status window will show FID: 2), transform the first FID by typing **wft(1)**.
  - If the ADC overflow light starts flashing, you may need to abort the acquisition (type **aa** to do this) and go back and reduce the receiver gain (i.e. type **gain=gain-2**) and repeat as necessary.

- Preliminary processing can be done with the help of linear prediction. When “FID:33” shows in the acquisition status window, you can begin to look at the first 16 complex points of your 2D data set with the help of linear prediction. Type **dolpav** (do linear prediction absolute value). The av at the end is a special macro to enable linear prediction for absolute value 2D data sets in order to predict the remainder of the data set based on the first 32 FIDs (16 complex points). **dolpav** makes use of the variable **celem** (completed FID elements) which keeps track of how many completed elements there are in the data set. Type **wft2da** (weight and Fourier transform phase-sensitive data) to transform the data set and type **foldt** (fold COSY-like spectrum along diagonal axis) to symmetrize it.
  - To see how much linear prediction helps, you can transform without it by typing **proc1='ft' wft2da foldt**
- To track the progress of the experiment, type **dolpav wft2da foldt** periodically.
- When the experiment is finished, the experiment should process automatically (because wexp (specify action when experiment completes) is set ; if the processing doesn't happen for some reason, type **dolpav wft2da foldt**

### Processing and manipulating the data:

- Type **dconi** this will display to plot contours and make the cursor active (or interactive). **dpc** will display the plot contours and leave the cursor active. **vs2d** can be adjusted with the middle mouse button in dconi mode (but not in dpc mode) or vs2d can be set by typing in a new value, e.g., **vs2d=200 dconi**

### Plotting the data:

- Typing **ppc** will plot, just as dpc displays (these are custom macros, the manual version of ppc is **pcon(30,1.3) dpcon(30,1.3) dconi('restart')**)
- Type **th**. This will adjust how many of the lowest contours are left off when plotting or displaying plot contours. The middle mouse button can adjust th in dconi mode.
- Type **page** to dump the contents of the plot buffer to the plotting device.
- Instead of plotting with pcon, you can run the plcosy macro. plcosy is usually run with three arguments: e.g., **plcosy(30,1.5,1)** - the first is the number of contour lines to draw, the second is the spacing between contour lines (1.5 means that each successive contour line will denote 1.5 times the intensity of the previous line), and the last number will be number of the VNMR experiment where the processed 1D spectrum resides. Note that the experiment with the processed 1D must have the exact same sweep width (sw) and transmitter offset (tof) in order to line up properly. That is why you collect the 1D and then use the movefid (mf) command to move the data including the parameters to another VNMR experiment where you then convert the parameter set to the 2D parameter set.
- Type **svf** and answer the prompts to save your file.

### Finishing up:

Be sure to turn off temperature regulation when you are done by typing **temp='n' su**. If you changed the temperature of the preconditioner, you will also need to turn the preconditioner temperature back to 20° C.