© Anne-Frances Miller, 2005

2d spectra I: gDQCOSY of Strychnine in $CDCl_3$ with CrAcAc ($T_2 \approx 90$ ms, $T_1 \approx 200$ ms)

Obtaining a respectable 2D is based entirely on having a nice 1D and good calibrations. Know pw90 (in the linear power regime if possible), have a good idea of your sample's T_2s and T_1s.

In all 2Ds, each point in the indirect dimension is acquired at the cost of a whole additional spectrum in the direct spectrum. Thus although large np and resolution cost only disk space in the direct dimension ('f2') **each ni costs approximately a minute in a traditional 2D.** A typical small molecule 2D used to take 4 hours. Using gradient coherence pathway selection to suppress artifacts and linear prediction to extend indirect dimension data set size, it is now possible to get very nice 2Ds in 10 minutes with sufficiently concentrated samples. However, we still have to compromise on the digital resolution possible in the indirect dimension (f1).

COSY: Correlation spectroscopy

COSY is the 'original' 2D, but not the easiest to understand. It nonetheless illustrates important points and is an excellent method for identifying coupled resonances and thus walking through a spin system. However Varian's COSY pulse sequence produces a magnitude mode COSY, which is low in resolution.

DQCOSY vs. gDQCOSY

We will use **g**radient **D**ouble-**Q**uantum-filtered **COSY** (gDQCOSY) because it is phase sensitive and the double quantum filter suppresses the uninformative diagonal signals and reduces their tendency to spread across the spectrum. Thus, it is preferred over even COSYps (phase sensitive) despite the theoretical loss of half the signal. Non-gradient DQCOSY relies on phase cycling (i.e. nt = some multiple of 8) to cancel out artefacts. gDQCOSY uses gradients to eliminate artefacts in the first place, so there is nothing to cancel and nt can be set to 1. This is a big advantage if you have a concentrated sample that doesn't need nt > 1 for sensitivity reasons, and gets around the problem that artefacts (or anything else) don't substrate out cleanly unless the machine, room, sample etc are absolutely stable over the course of the experiment. Our spectrometers are excellent, but the probes are so sensitive that even small variations in temperature etc result in non-cancellation. However, the use of the gradient-based artefact suppression will cost you half the signal intensity, and this may not be tolerable, depending on the sample. I still advise the use of gDQCOSY, because even with lower signal, the S/N is so much better that the signal you have is much more useful.

High resolution is required in all phase-sensitive COSY spectra, in order that antiphase resonance components do not cancel (cf. TOCSY).

Finally, because magnetization transfer is mediated by J coupling in a COSY, and antiphase components of crosspeaks are separated by J, you will need to have a value of J in mind when you set up your COSY. Your choice will depend on what couplings are resolved in your spectrum (use a J value ≥ the smallest resolved value) and the types of couplings that characterize the spin systems that interest you most. I often use a compromise value of J = 7 Hz.

With a good 1d spectrum already in an experiment (including pw90, tof, sw, solvent and phases set), type **gDQCOSY**. You may want to make changes in at (if you double it, double fn also) and ni, but recall that resolution will be bought at the cost of S/N, because at the long at and ni values needed, there will by much less signal in the FID. Similarly, d1 should be = 1.3 x T_1, at least. In fact, for COSY spectra, spurious diagonals appear when d1 is less than 3 T_1 and purge pulses are not used between scans. Always use **sspul='y'** for all COSYs without fail. This is also a good idea for multidimensional experiments in general.

Understanding the indirect dimension

During direct detection we monitor magnetization as it precesses and measure its amplitude along two axes (X and Y = real and imaginary) in 'real time' *i.e.* as it is happening. However, just as the fast reactions of colourless compounds can only be monitored by taking our samples and analyzing them, later, to yield concentration of each reactant and product as functions of the amount of time they had been allowed to react before the sample in question was taken, the indirect dimensions in multidimensional NMR experiments are collected one 'sample' at a time, and each 'sample' is a 1d spectrum representing the state of the 'reaction' after a given amount of time had been allowed to elapse, Δt = (number of increments)*(time per increment) = ni * (1/sw1). sw1 is the sweep width of the indirect dimension and as you recall from earlier lectures, and the Nyquist criterion, short (*small*) time increments of 1/sw1 must be used to collect data if high (*large*) frequencies sw1/2 are to be distinguished from smaller ones (another incidence of time α1/frequency). Thus for a large sw1 a small time increment = 1/sw1 is used. (The factor of two results from the carrier frequency being in the center of the spectrum so the biggest frequency offset is sw1/2).

If you were to collect one hundred samples from your reaction, 5 seconds apart, you would have sampled the reaction until 500 seconds. You would not be able to distinguish from baseline processes happening faster than 1/(5 sec), or with rates > 0.2Hz. You would also not be able to detect at all processes happening more slowly than ≈ 1/(500 sec). Thus, in NMR parlance, your sweep width would be limited to .2 Hz (faster processes would fall outside your spectrum), and your spectral resolution would be .002 Hz (slower processes would look like they weren't happening at all, and would not be resolved from 0 Hz). Finally, your total acquisition time 'at' would be 500 sec. Since we are talking about the first indirect dimension, I will refer to the longest time point sampled as at1, by analogy with at, and recognizing that at1 is related to sw1 (not sw). Although such a parameter is not defined in VNMR, it is very useful. In the example above, at1=500 sec. In your standard VNMR data set, at1 = ni/sw1.

In VNMR all the parameters pertaining to the first indirect dimension are labelled with a 1. sw1, lpext1 are the analogs of sw and lpext for the first indirect dimension. Similarly, in 3d and 4d spectra, the second and third indirect dimension's parameters are identified with 2 and 3. The exception is ni, which is the number of complex data points (samples) you will collect in the first indirect detection, ni2 and ni3 are the analogs for the second and third indirect dimensions.

In brief: two properties of the F1 dimension are its width, sw1, which you set **sw1=sw**, and its resolution, also in Hz. Each is related to the inverse of a different time. sw1 is related to 1/increment time used in extending t1 (d2). The resolution is related to the length of the indirect detection delay, just as direct dimension resolution is related to at. Thus the longest time point used in t1 is equivalent to at1 and at1 = number of steps * step size = ni*(1/sw1).

Choosing parameters pertaining to the indirect dimension

Sw1 is usually set for you, sw1=sw. You may decide to double at, in which case also double fn. Make sure d1 is sufficiently long (above) and sspul = ‘y’.

ni

In choosing our at in a 1d spectrum we weight the loss of signal amplitude due to T2 against gain in resolution in proportion with 1/at. You have seen that one can in fact collect a modest at of data and double it by linear prediction before applying a weighting function to emphasize the actual data while forcing the FID to decay all the way to zero in order to avoid sinc wiggles around the lines of slowly-relaxing resonances. In addition, zero filling by a factor of two (fn ≥ np) improves resolution.

Similar considerations apply in the generic 2d (special COSY considerations follow). T_2 relaxation proceeds during the period of sampling (sampling α 'indirectly taking data'). This period is called t_1 in texts and

pulse sequences but d2 in the VNMR parameter sets. (A second indirect detection period is called t_2 in the literature but d3 in your parameter sets.) Each successive 1d (sample) taken in your 2d will be collected after the t_1 (d2) has grown a bit longer, and the signal has decayed a bit more due to T_2. Therefore it is generally a waste to let at1 be bigger than $2T_2$. For typical samples $2T_2$ is a good limit. The cost of restraining t_1 (d2) is that you will be placing a limit on your resolution in the frequency dimension corresponding to t_1 :F1. Since the maximum t_1 before sampling is our 'at1' parameter = ni/sw1, and the resolution limit is 1/at1, we are limited to sw1/ni resolution. Check that this does indeed have units of Hz. This number is in essence the Hz per indirect dimension point. So your indirect dimension resolution improves as sw1/ni (α 1/ni) but your signal decays more with larger ni (α ni/sw1). The other disadvantage of choosing the large ni (long at1) you might be most comfortable with, is that in the indirect dimension it takes you up to a minute to sample each time point (you need another whole 1d). This is in sharp contrast to the direct dimension where a longer at comes at little or no cost in time since d1 often compensates for it. However, the need for resolution is much less severe in a 2d, since your points will be spread out in a plane, not just a line. Therefore, the best strategy is to collect relatively low resolution (low ni) indirect dimensions and linear predict and weight as necessary to suppress sinc wiggles. It will be very rare to collect ni>512, and this is usually only done for phase sensitive COSY spectra.

One additional factor enters into your choice of ni for a COSY. In this case only, the amplitude of the cross peaks (the guys you want) grows with t_1 as $\sin(\pi J t_1)$. Therefore you will get your strongest contributions from the samples (1ds) taken with $t_1 \approx 1/2J$. In practice, it is nice to go out 50% further to at1= 3/4J or ni = 3sw1/4J.

Before launching for real, test-drive yoru parameter set by setting d2=.02 and ga. After ss scans (the steady-state scans), you should see nice derivative-like signals where the signals are. **aa** to stop and RESET d2 to 0 !!!! then type time to see if you can actually afford the nt and ni you have. Cut back as needed, then launch with **ga**.

Processing

Initially, set **pmode='full'** (allows phase correction in both dimensions after the FT) and **proc='ft'**. The first increment of a COSY has very little signal in it, instead process a later odd-numbered increment. For example, **wft(201)**. Set up the window function in wti as usual. For gDQCOSY, Varian adopts the luxury of using a Gaussian, because there are so few artifacts. (Later we will also employ the sine-bell indicated by theory to best emphasize COSY crosspeaks). For now, set **gf = .5 *at**. Also, for a COSY there is no need to linear predict beyond at = 1/J, since this is where the sine bell will cut of the data (see below). Our typical at are > 1/J without linear prediction, so leave proc = 'ft'.

Process the first dimension as a block by typing **wft1da**. This will produce a whole stack of 1ds whose frequency axes are along f2. To see these, set **trace='f2'** and **dconi** to redraw, then activate trace (a button) and drag the cursor from the bottom of the box (where the first of the 1D spectra is) up, and watch the trace to see how each successive spectrum has signals in the very same places, but with growing amplitudes and with changing phases. Set **trace='f1'** to use the t1 axis as the X axis, **dconi** again, place the cursor on a streak and type **ds** to show it as a 1D of the amplitude of the selected peak vs time in t1. This is in effect an indirect dimension FID (subject to Fourier transformation, like any other FID). **wti**, and set up a window function, as usual. Again for gDQCOSY you can get away with a Gaussian just as for standard 1ds, for now. gf1 = 0.5 at1, or more correctly **gf1 = 0.5*ni/sw1**. Note that a '1' follows the gf parameter name to indicate that it pertains to the first indirect dimension.

Recall that the gf1 chosen at this point does not take into account forward linear prediction. Set up linear prediction and **proc1='lp'**. For COSY data, you would like your data (actual and calculated) to extend to 1/J in t1. Calculate how many points ('n') that would entail. You know that at1 = n*(1/sw1), and in this case you want at1 = 1/J, so n = sw1/J. (You can confirm for yourself that you will get n = 4ni/3 or n = ni + ni/3 if you used the recipe above for ni = 3 sw1/4J). Since you want n points and you have ni points, you need to predict n-ni points, so lpext1 = n-ni, (or ni/3 if you used the recipe for ni above). Use lpopt1 = 'f', lpfilt1 = 12 (a bit more than the number of crosspeaks in a typical row). Use all your data by setting lpnupts1 = ni and strtlp1 = ni strtlp1 = ni+1. Check that you have fn1 > 2* (ni+lpext1) so there is room for all the actual and calculated data points to fit into the Fourier transform. Adjust the window function if needed. Then type **wft2da** to complete the 2d transform.

Phase up your spectrum, in each of the two dimensions, one at a time. It is best to work on a strong and simple cross peaks, which you know should have positive intensity in two diagonal corners (eg. North-East and South-West) and negative intensity in the other two diagonal corners (eg. North-West and South-East). Beginning with the f1 dimension, which may be what is already horizontal (if not, type **trace = 'f1' dconi**); place the horizontal cursor line across the peak and trough that make up the top half of your cross peak (begin with one in the right-hand side of the spectrum), and type **ds** to get a 1d. Phase this to get a negative peak followed by a positive one of equal magnitude, and flat level baseline on both sides using zero-order phase (rp1) (it shouldn't need much). Do this without releasing the mouse button or repositioning the mouse, you ONLY want to adjust rp1. Then choose another simple strong cross peak in the left portion of your spectrum and adjust the first order phase (lp1). As before, place your cursor across the top half of the peak and type **ds**. Click once on the phase button then click ONCE on the right-hand side of the spectrum WITHOUT CHANGING THE rp. Then move the mouse to the cross peak you selected, press down on the mouse button and adjust lp WITHOUT MOVING THE CURSOR ANOTHER TIME. Then click out of phase mode.

Select and correct f2 phasing if necessary by setting **trace = 'f2' dconi** and repeating the above exercise for rp and lp. Again, your objective is to produce peaks that are antiphase absorptive in both dimensions. They should be little squares with positive amplitude at the top right and bottom left, and negative amplitude at the bottom right and top left.. If you just need to phase f1, **pmode='partial'** will suffice but if you want to phase both dimensions you will need to set **pmode = 'full'** before you begin the Fourier transformations. For gDQCOSY, theoretical phases are rp=0, lp=0, rp1=90, lp1=0, but 15 degree deviations from these are not unusual after optimization. Once the phases are optimized, set **pmode = ''** and repeat the wft2da to produce a much smaller data set which now lacks all the imaginary numbers you don't see anyways (the data you collected are all saved, only the Fourier transformed spectrum is culled).

If your gDQCOSY was based on a 1d that already had a referenced chemical shift axis, and you used a macro to convert your 1d parameters into a gDQCOSY parameter set, then the f2 axis will retain the chemical shift calibration. This may or may not be transferred cleanly onto the f1 axis. The parameters responsible for the chemical shift calibration are rfl and rfp for the f2 axis. Typing will **rfl1 = rfl** and **rfp1 = rfp** apply the same calibration to the f1 axis. You can also calibrate a spectrum based on a line of known chemical shift. With trace = 'f2', place the vertical cursor on the line (in the centre of a COSY cross peak) and type **rl(4.7p)** for the example of a known chemical shift of 4.7 ppm. Note that you will use the **rl1** command instead of **rl** to set a reference chemical shift when calibrating the f1 axis..

To get a pretty looking contour plot, analogous to the printout but in colour, type **dconi('dpcon')**. **dconi('dpcon', 10, 1.3)** plots the spectrum with 10 contours spaced by 1.3

plcosy(10,1.3,2) page plots the COSY using 10 contours spaced by 1.3, and the spectrum in experiment 2 along one edge. This macro does not require that the sequence used actually be a COSY.

Special treatment for COSY spectra

For COSY-type spectra, the diagonal peaks contribute intensity to the FID even at early times, but the cross peaks grow in with sin (π J t). Thus, in order to emphasize the all-important crosspeaks, it is customary to use sine bell window functions. For the J value you want to emphasize, set <u>sb</u> = 1/2J and <u>sb1</u> = 1/2J, and turn off all other window functions. You can look in wti to see what this function looks like on your data, to be sure that you have a long enough <u>at</u> (likely) and your linear prediction results in a long enough set of data in t1. You will need <u>at</u> $\geq$ 1/J and n/<u>sw1</u> $\geq$1/J where n is the total number of points in t1, including actual and predicted ones (above). With sine bell windows in place, repeat the wft2da. The overall intensity of the spectrum will decrease, but this will mainly correspond to loss of diagonal peak intensity. The result is more emphasis on the (informative) crosspeaks. (Another less optimal but operationally simple choice of sine bell filters is <u>sb</u> = -<u>at</u>, <u>sb1</u> = -1*at1, where the - sign indicates the use of a squared sine instead of a simple sine, which suppresses f1 noise a bit more.)

Read chapter 5 of Claridge's book for this time, chapter 8 for next time.

Be prepared for a quiz question requiring you to predict the COSY spectrum for a spin system, and the corresponding TOCSY.