I. Arrays in VNMR

Vnmr allows you to quickly define experiments in which a series of spectra can be obtained as a function of any NMR parameter. For example, you can obtain an *array* of spectra where transmitter frequency, pulse width, temperature, decoupling state, sweep width, etc., are varied. You can define arrays of one dimension by varying a single parameter. Alternatively, you can define multi-dimensional arrays in which two or more parameters are varyied (on the Gemini-300 you are limited to arrays of 3 parameters).

You can array any parameter by simply typing the command **array** and answering the questions about the desired parameter and the size and range of the array. You can also define an array by manually setting the parameter to the desired values. For example, to obtain a series of 10 spectra where each successive spectra is acquired with twice the number of scans as the previous one, you would type **nt=1,2,4,8,16,32,64,256,512,1024.** To obtain spectra with the decoupler turned on and off, you would enter **dm='yyy','nnn'. NOTE**: on the Gemini-300, you need to specify whether the parameter you are varying is array 1, 2, or 3. Do so by placing a 1, 2, or 3 in parenthesis after the parameter name, e.g. **NT(1)=2,4,6,8,** etc.

The following section describes how to perform several basic NMR measurements using the standard varian 2 pulse acquisition sequence **s2pul** and arrayed acquisition of spectral data.

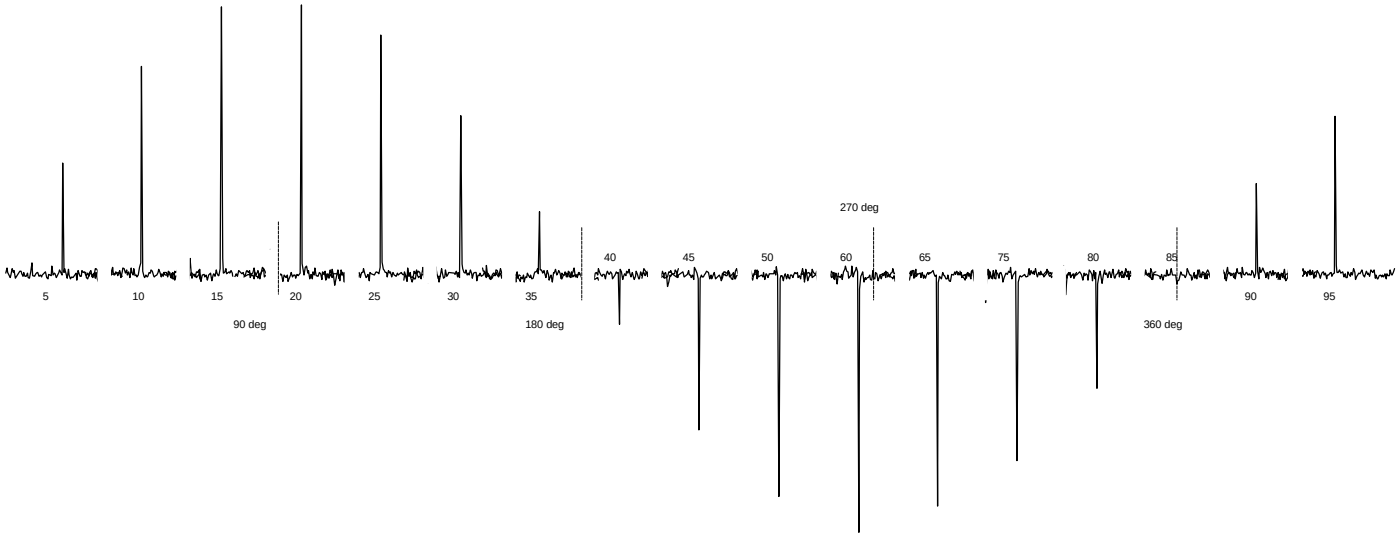
II. Simple Arrays using s2pul

A. Measuring 90 degree pulse widths

You can measure the 90 degree pulse width of the selected nucleus by making an array **pw**. The following spectra was obtained on the Gemini-300 using the array

pw=5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95 When measuring the pulse width, the parameters **p1** and **d2** are set to zero and **d1** is set to ca. 5 x T_1.

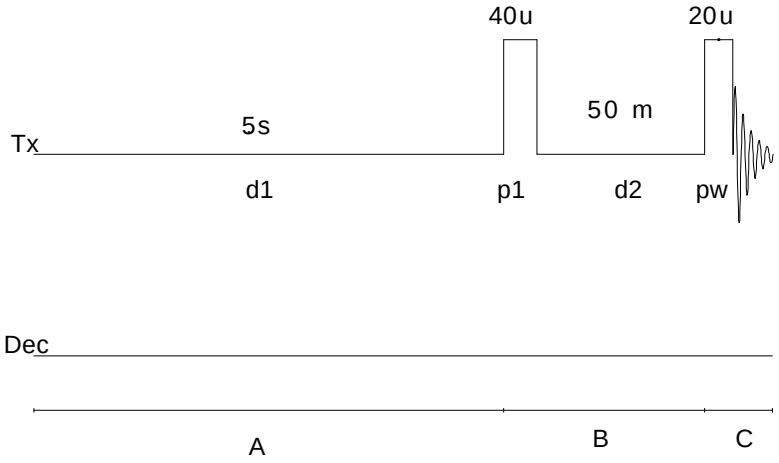
Figure 1. Peak Intensity vs. Pulse Width



B. Spin-Lattice Relaxation Times (T_1)

Arrays greatly simplify the measurement of spin-lattice relaxation times or T_1. The standard 2 pulse sequence (**s2pul**), in combination with arrays, can be used to perform T_1 measurements using inversion-recovery sequence, as shown below. The inversion (180°) pulse is performed with pulse **p1**, which is followed by the recovery delay **d2**. **d2** is arrayed to observe the relaxation of magnetization to equilibrium as a function of time.

Figure 2. Inversion Recovery T_1 Pulse Sequence Generated With s2pul.

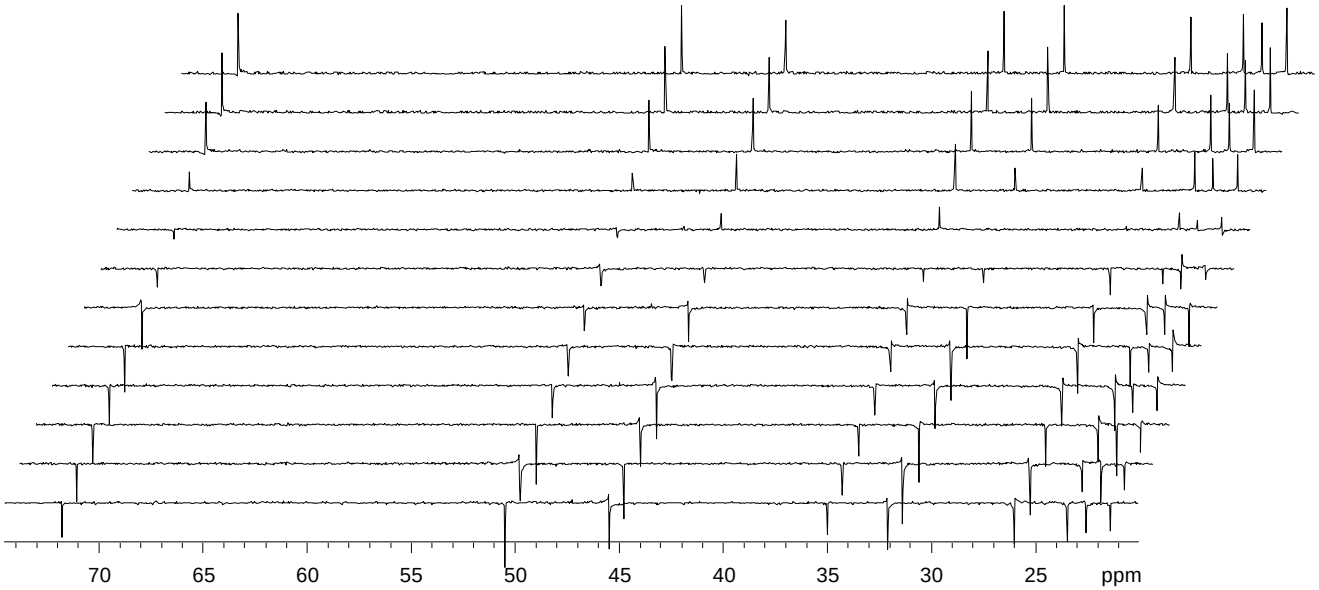


T_1 measurements require a long (5x T_1) delay of time **d1**, followed by a 180° *inversion* pulse **p1**. The magnetization is allowed to *recover* for a time **d2**. The residual magnetization is then examines by a quantitative 90° pulse **pw.** The sequence is repeated for **nt** transients at various values of **d2**.

d1=5 x T_1
p1=180°
d2=array from ca. 0.01 x T_1 to 3-5 x T_1
pw=90°

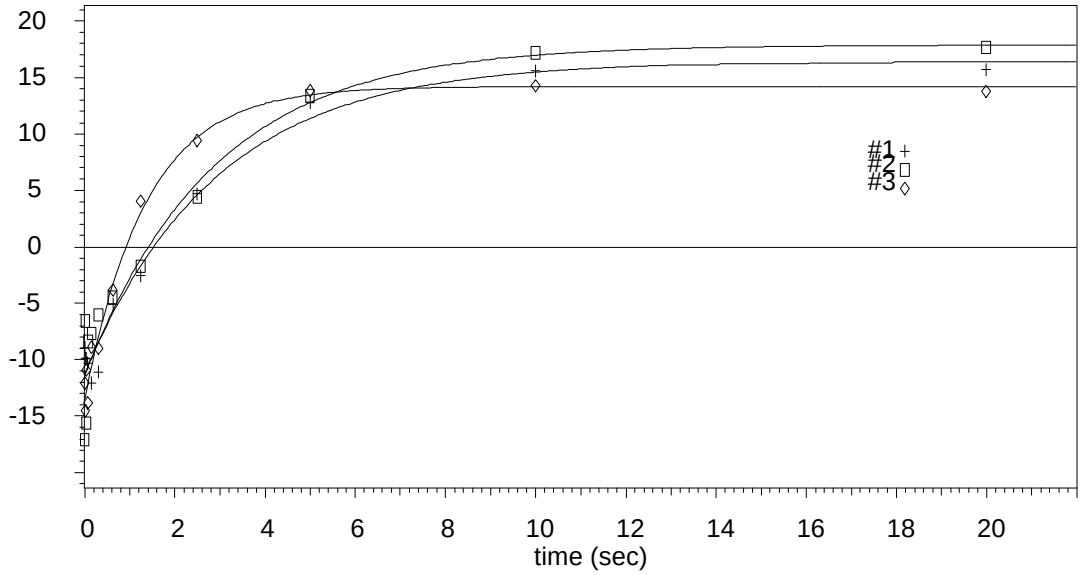
An example of an arrayed series of spectra obtained using the inversion-recovery method is shown in Figure 3. T_1 experiments can be easily acquired using **s2pul** and arrays. To acquire a T_1 data set, obtain a simple 1D spectrum. Type **dot1** and answer the questions about minimum and maximum expected T_1 values and the desired time for the experiment. The macro **dot1** will generate set **pw** to 90° and **p1** to 180°. **dot1** will set the equilibration delay **d1** to ca.5 times the longest expected T_1 value and **dot1** will then generate an array of relaxation delays **d2**. You can adjust **d1, d2** and **nt** to change the overall time required, if so desired (type **time** to find out how long the experiment will take. Finally, type **ga** to acquire the data.

Figure 3. Example of a Carbon T_1 Measurement (Menthol).



VNMR has an excellent set of tools for analyzing the results of the T_1 experiment. To process the T_1 data, display the last spectrum and set the threshold for peak picking. Type **ds(n)** where **n** is the number of the last spectrum. Phase the spectrum and set the threshold, in exactly the same way you would with a simple 1D spectrum. Adjust the display so that only six peaks are shown and type **dpf** (or **dll**). This will generate a list of peaks in the spectrum. Now type **fp**. The command **fp** will find the intensity of each of the peaks in the displayed region and will do so for all the spectra in the array. Type **t1** to calculate the T_1 value for each of the displayed peaks. The command **dl** will generate a list of peak intensities for each peak, and the quality of the fit to the estimated T_1 value. You can generate a graphical display of this data using the **expl** (Figure 4). The command **expl** will display the fit of the data. The following graph was generated by displaying three of the peaks in the T_1 example shown earlier, followed by **dpf, fp, t1,** and **expl.** The exponential analysis can be output to the plotter by typing **pexpl page**. **expl** and **pexpl** accept the peak numbers as options. For example, if you had displayed six peaks and analyzed the T_1 values with **fp**, followed by **t1**, you could generate the T_1 curves for peaks 2 and 5 (of the six) by typing **expl(2,5)**.

Figure 4. Example of T_1 Analysis.



C. Kinetics and Variable Temperature Experiments

Kinetics and Variable Temperature experiments are straightforward to set up using arrays. VNMR has a parameter called the **p**re-**a**cquisition **d**elay, **pad**. The spectrometer will wait **pad** seconds before acquiring the spectrum. For a kinetics measurement, simply array **pad** to leave the desired time between successive spectra. Alternatively, a local macro program called **kineticset** can be used to automatically defined an array of **pad** values that increase in user-definable stages that take into consideration the exponential rate of change in kinetic experiments.

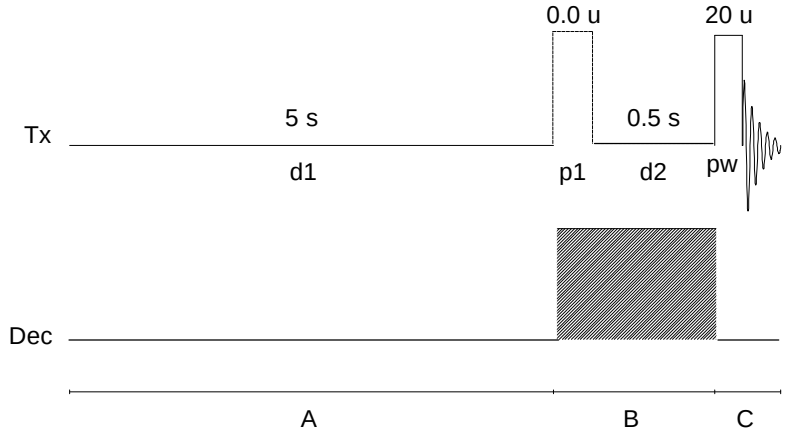
Once you have defined the array of **pad** values, type **ga** to acquire the spectra. VNMR contains tools to analyze the spectra in a manner that is very similar to that of T_1 data. First, display the last spectrum in the kinetics series and set the threshold for peak picking. Type **ds(n)** where **n** is the number of the last spectrum. Phase the spectrum and set the threshold, in exactly the same way you would with a simple 1D spectrum. Adjust the display so that only six peaks are shown and type **dpf** (or **dll**) to generate the l ist of peaks in the displayed portion of the spectrum. Next type **fp** determine the intensity of each of the peaks in the displayed region of the array. Type **kini** or **kind** to fit the data for each peak to an increasing or decreasing exponential function, respectively. The

commands **kini** and **kind** generate a list of peak intensities for each peak, and give errors to the exponential fit. As with T_1 data analysis, you can generate a graphical display of this data using the **expl** as shown in Figure 4.

D. Single Frequency Decoupling Measurements

In progress

E. Homonuclear One-Dimensional NOE Measurements



In the homonuclear NOE experiment a spectrum is acquired with single-frequency (selective) CW irradiation of one resonance. This spectrum is subtracted from a second spectra acquired without irradiation of the peak.

d1=5 x T_1 (typically 5-20 s)
d2=1 to 5 x T_1
pw=90° or less
dm='nyn'
dmm='ccc', dof=desired frequencies

The homonuclear NOE measurement requires an array of decoupler frequencies. The first spectrum is acquired with the decoupler off-resonance from any peaks. Successive spectra are acquired with irradiation of a single peak for a time **d2**. The decoupler is turned off and the transient is acquired after the pulse **pw.** The spin-system is allowed to return to equilibrium during the time **d1.** Setup, acquisition, and processing of the homonuclear NOE measurement is covered in detail in a separate handout.

IV. Interleaving Block Acquisition of Arrays Using nt, il and bs.

In progress

IV. Useful Commands Associated with the s2pul Pulse Sequence

In progress.

da
expl(n,m,etc.),
pexpl(n,m,etc.)
fp
t1
kineticset

kini, kind

pl(n,m,o), pl('all')

dssa, dssh, dss

vo, ho

sd, sda

pad

time

dot1

clradd, spadd, spsub, addi

array