

NMR NOTES #16

Using Acquisition Arrays

Virtually any acquisition parameter may be arrayed, permitting a series of spectra to be collected as a function of that parameter. Data that has been collected as an array is treated as a single data set, making it easy to plot all or some of the spectra using identical plot and processing parameters. It is also possible to array more than one parameter and to control the order in which the various arrayed parameters are incremented. Common uses for acquisition arrays are pw90 determinations, T₁ determinations, and kinetics experiments.

The only constraint on arrayed parameters is that the data table size cannot change between array elements. This means that the sweep width (**sw**) and acquisition time (**at**) parameters may not be arrayed. The other constraint on arrayed experiments is that autogain cannot be used to adjust the receiver gain. The purpose of an arrayed experiment is to be able to compare intensities in the acquired spectra as a function of the arrayed parameter, and these intensities cannot be compared if the gain is allowed to change between spectra. If you attempt to start an arrayed data acquisition with **gain='n'**, you will get an error message indicating that autogain is not permitted in arrayed experiments. You must manually choose a receiver gain value. The **gain** parameter itself however may be arrayed.

A parameter array may be simply created by giving a list of values to an array reference: **pw[1]=5,10,15,20,25,39**. This creates a pulse width array with 6 values. Any given value may be changed by simply giving it a new value, just like you would any other parameter: **pw[6]=30** to correct the typo in the first example. Additional elements may be added to the array by assigning a list of values to the first array element to be added: **pw[7]=35,40,45,50**. The array now has 10 values. Arrays may be increased in size by simply adding new values, but they cannot be reduced in size. If you want to remove the last four elements just added, all you can do is eliminate the array (give pw a single value: **pw=5**) and recreate the original array: **pw[1]=5,10,15,20,25,30**. There is also an array command that will simplify creating regularly spaced arrays: **array('pw',6,5,5)**. If you don't specify the necessary arguments to the array command, it will ask you for the necessary values: the parameter to array, the number of elements in the array, The first array value, and the array increment.

There is an **array** parameter that controls the order in which multiple arrays are incremented. If you wanted to measure pw90 at two or more different power levels, you could array both **pw** and **tpwr** parameters. Setting **array='tpwr,pw'** will result in the **pw** parameter being incremented first, and then the **tpwr** being incremented in an outer loop, changing more slowly than the **pw** parameter. This will permit you to view the variation of **pw** as a function of the **tpwr**. If you set **array='pw,tpwr'** instead, you would examine the variation of **tpwr** as a function of **pw** - probably not what you intended. The **da** command (Display Array) will display all of the arrayed parameters as well as the overall array dimension.

Once your array has been created, you can proceed with a normal data acquisition using **ga**. The system will collect one spectrum for each array element, incrementing the FID number in the acquisition status window. You can process, examine, and plot these spectra just like you would 'normal' data. There are several extra features used to implement this. The active spectrum is selected either by using the select command explicitly or simply displaying the spectrum of interest: **select(5) ds** does the same thing as **ds(5)** - it selects and displays the 5th spectrum in the array. The current, active spectrum index is shown on the top of the command line window to the right of the experiment number. Note that accessing the elements of an arrayed parameter requires the use of square brackets (**pw[5]**) whereas accessing a spectrum from the array uses parentheses (**ds(5)**). In the case of the **ds** command, the spectrum index is an argument to the command, so it is listed in parentheses like any other command argument. The plot command **pl** will also take arguments: **pl(5)** to plot the 5th spectrum, or **pl(4,6)** to plot the 4th through 6th spectra. In order to preview plots of multiple spectra, you can use the **dss** command (Display Stacked Spectra), which shows on the screen the same display that will be plotted by **pl**. It takes the same arguments as the **pl** command. The **dss** and **pl** commands use the **ho** and **vo** parameters to define the horizontal and vertical offset between successive spectra. See the online manual for more details on using these commands and the full list of arguments that are supported. The other useful display command is **dssh**, which will display the spectra stacked horizontally across the screen. In virtually all cases, when displaying arrayed data sets, you will want to use the absolute intensity mode, rather than the more common normalized intensity mode. Normalized intensity scales the spectrum so that the largest peak is **vs** mm tall. Absolute intensity does not do this scaling, so that peak intensities can be compared. Use the **ai** and **nm** commands to switch between absolute and normalized intensity display modes. Finally, the **dssl** command can be used to label a dss display with either array index values (**dssl**) or array parameter values (**dssl('values')**). If you change processing parameters and want to re-process the entire data set, use the **wft('all')** command. Just **wft** by itself will only transform the current spectrum or a few spectra in the same disk block, but not necessarily all of the spectra in the data set.