Kinetics experiments with VnmrJ 2.2D and 2.3A

Eugenio Alvarado, University of Michigan
9/29/06. Last revision 5/04/07, 10/20/08, 11/16/09, 02/03/11

Setup

In a kinetics experiment, a series of spectra measured at predetermined time intervals have to be obtained over a period of time. The procedure is better explained with an example, so lets assume you want to follow a reaction overnight and want to take a spectrum every 30 minutes (1800 s) starting at 10 pm and ending at 8 am. That is 10 hours for a total of 20 spectra. You now have the choice of accumulating each spectrum for up to 30 minutes. Lets say that you want to accumulate for only 5 minutes. In Vnmrj you can't specify the actual time at which an acquisition is going to start. That is, there is no easy way of telling Vnmrj that you want the acquisition of your first spectrum to start at 10:00 pm followed by another spectrum at 10:30 pm and so on; instead, you will measure one set or array of spectra whose component spectra will be collected in succession, but with a time delay between them chosen in such a way that each spectrum will start at the desired time. For this purpose you must calculate exactly how long each spectrum will take and also calculate the time delay between spectra so that each spectrum starts at the desired time. The whole sequence of events is as follows:

pad[1] — spectrum1 —— pad[2] — spectrum2 —— pad[3] — spectrum3 —— pad[4] — spectrum4...

The time delay is called `pad`, for "pre-acquisition delay" and will be entered as an array as described below. In our example you will need delays of 25 min in order to measure the spectra every 30 minutes.

For most proton spectra, an acquisition of 16 scans is sufficient and it will take under a minute. But measuring other nuclei or dilute solutions may require longer accumulation times. Thus, you must first calculate how long each spectrum will take. In a simplified way, measuring a routine 1D spectrum takes `nt*(d1+at)` seconds. However, it is customary to include a small number of "steady state" scans prior to the acquisition of data. These steady state or dummy scans are just like the real scans except that the FID is not collected. Their purpose is to help the magnetization reach a steady state before the data is collected. In this way, the spectra in the array will be more directly comparable. By default, all 1D spectra are acquired with 2 steady state scans, `ss=2`. At this point it is important to understand that in an arrayed experiment these scans are executed only once, before the first spectrum is acquired, leaving the magnetization of the first spectrum in a different state than at the start of the other spectra. Therefore, the integration of the first spectrum may be slightly different than the rest. But the parameter `ss` can also have a negative value, in which case vnmrj will execute the steady state scans before each spectrum. This is more appropriate, and recommended for an arrayed kinetics experiment. When `ss` is negative the execution time for each spectrum will be `(nt+|ss|)*(d1+at)` compared to `nt*(d1+at)` when `ss` has a positive or zero value. Note that if you plan to use vnmrj's analysis routines (described below), you should not use `ss` with negative values because they have a bug that prevents them to calculate the acquisition times accurately.

Following with the example, next you must find the number of transients required so the experimental time per spectrum is approximately 5 minutes (300 s). After setting `ss=-2`, measuring a routine 1D spectrum will take approximately `(nt+|ss|)*(d1+at)` seconds. For a typical proton spectrum on the Inova 500, the relaxation delay `d1` is 0.5 s; the acquisition time `at` is 3.0 s and the number of transients `nt` is 16. Our two steady state scans will take 7 s, and doing a little algebra `nt` = (300-7) / 3.5 and thus `nt=83.7`. Since the number of transients must be a multiple of 4, we will choose `nt=80` (we could have chosen 84 as well) and each spectrum will take exactly 287 seconds. Therefore, the pre-acquisition delays must be 1800-287=1513 seconds. In most cases, the first spectrum can be collected immediately so the first `pad`, `pad[1]`, can be set to zero.

You can also use the macro `UMtime` to calculate the number of transients for you. Typing `UMtime(300)` the parameter `nt` will be calculated in such a way that the total accumulation time will be approximately 300 seconds. Typing `UMtime` without any argument will calculate the exact total experimental time with the current

parameters, including `pad[]`. Notice that the `time` macro, provided by some versions of vnmrj, or the green [Time] or [Show Time] buttons on Vnmrj's interface do not take `pad` into account in the calculation. Latest versions of vnmrj do take `pad[]` into account, but assume that all of them have the same value. In our example the experimental time calculated by `time` is 1hr 38m instead of ~10hr! Also, when measuring the spectra of some nuclei like ^{199}Hg or ^{195}Pt, where it is common to use very short acquisition and recycle times (between 5 and 100 ms), the `time` macro provided with Vnmrj is very inaccurate as it doesn't take into consideration other small delays that normally occur in the standard 1D pulse sequence. Use `UMtime` instead.

To enter the delays you could type: `pad`=0,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513,1513, 1513. You must enter 20 values in our example. But it is probably easier to type: `array('pad', 20, 1513, 0) pad[1]=0` and vnmrj will setup a `pad` array identical to the previous one. Alternatively, you can select *Acquisition>Parameter arrays...* from Vnmrj's main menu to enter your values in a window. Notice that, because `pad[1]=0`, the total experimental time will be 9 hours 35 minutes instead of 10 hours.

Now that you know how to calculate the experimental time and pre-acquisition delays, you can do it the easy way. To ease the calculation of pre-acquisition delays, the macro `UMsetupkinetics` was developed. Typing `UMsetupkinetics` asks for the total time available for the experiment, in minutes. In our example, that will be 600. Then, it asks for the time interval between spectra. Since we want to collect a spectrum every 30 minutes, we enter 30. The macro then calculates the appropriate values of `pad[]` for the experiment. The macro will also store the actual, real world time at which each spectrum ends in Unix format (the number of seconds elapsed since Jan 1, 1970). These times will be stored in the parameters `end_time[1]`, etc. But for the actual time to be stored, the acquisition must be started with `au` instead of `go` or `ga`. When the time for each spectrum is stored in this way there is no chance of miscalculation... Well, actually if the acquisition spans 2:00 am on the second Sunday in March or the first Sunday in November, when daylight saving time changes, then the actual times reported may give errors in the calculations, so beware of these dates. To display the end time of a spectrum in the array, type `write('line3', end_time[`*n*`])`, where *n* is the number of the spectrum. And to convert that number to a human-readable date, in a *Terminal Window* type `date -d @`*`1296513572`* (replace the *number* shown with the value of `end_time[n]`).

Before you can start the acquisition you must make sure that the `gain` is not set to be automatically detected (`gain='n'`) as is the default in proton spectra. Thus, if you haven't done it yet, take a normal proton spectrum in a different workspace; only 2 scans will be enough. Type `gain?` to get the value determined by the software (the number in parenthesis) and manually set `gain` to this value in the workspace where you are going to run your kinetics experiment so vnmrj won't attempt to determine it in this workspace. After that you can start the acquisition with `au`.

Some useful tips:

Usually in a kinetics experiment it is very important to carefully regulate the temperature. Before you insert your sample, set the desired temperature and use a dummy sample of similar characteristics to pre-shim the magnet and allow enough time for the probe to equilibrate, about 15 minutes. Set all the acquisition parameters while you wait. Insert your sample and, if the temperature in the probe is different from that of the sample, wait again for the sample to equilibrate, at least 5 to 10 minutes. The larger the temperature differential between room and probe, the longer the equilibration period should be.

For a quantitative integration, the spectrum should be centered in the region of interest and you should leave at least 15% of free baseline around both sides of the spectrum. For example, if the peaks in your proton spectrum go from 1.1 to 7.8 ppm you won't need the region from 10 to 14 ppm or from 0 to -2 ppm that is present in the

default parameters. Place one cursor on about 9.5 ppm and the other on -0.5 ppm and type `movesw` to select this region. Take a new spectrum to verify the new parameters.

For accurate integration the digital resolution should be very good. Each peak should be covered by at least four acquired data points in order to properly represent its area. For this reason it is usually a good idea to increase the acquisition time to 5 or 6 seconds for ^{1}H NMR. The recycle time (the time between pulses, usually `d1+at`) is also important. When using 45 degree pulses (the default) the recycle time should be at least $3T_1$ of the slowest relaxing nucleus. Medium size organic compounds have T_1 values from 0.1 to about 3 seconds. Therefore you may need to increase the relaxation delay `d1` to increase the recycle time if slowly relaxing signals like methyl groups will be used to measure your kinetics experiment. Specially if you are using some kind of internal reference for integration, like a known concentration of a small molecule like acetone, TMS or dioxane, it is important to use long recycle times as the relaxation times for these molecules can be very long (several seconds).

It should be possible, with a sufficiently long pre-acquisition delay, to record the spectrum of a second nucleus during this time. In this way, it is possible to follow a kinetics experiment by observing the spectra of two nuclei, for example ^{1}H and ^{31}P. The macro `UMkin2nuc` can be used to set up such experiment easily. Read the separate writeup "*Kinetics experiments with two nuclei*" for more information.

In summary:

1) Set all conditions and take a spectrum (temperature, acquisition time `at`, relaxation delay `d1`, spectral width `sw`, `gain`, dummy scans `ss=-2` or any negative value). Use `UMtime` to calculate the number of transients if necessary.
2) Use `UMsetupkinetics` to calculate the pre-acquisition delays, `pad[]`.
3) If necessary, modify acquisition parameters and repeat step 2 until you are satisfied.
4) Start the acquisition with `au`.

When the final spectrum is finished, save the array as usual. The set of spectra will be stored together in the same file and you can process it as usual and display each spectrum with `ds(1)`, `ds(2)`, etc. or you can use the → and ← buttons on vnmrj's tool bar to switch between spectra. Note that in order to compare peak integrals and intensities in different spectra, you need to have "absolute intensity mode" enabled; type `ai` to enable it.

Processing

After Fourier transformation, carefully phase a representative spectrum, usually the first or second spectrum of the array. For accurate integration and peak height measurements, a very straight baseline is essential and some sort of baseline correction is usually needed. To do this, first carefully define integral regions on *all* peaks in the spectrum, *even on those you are not interested in and those not yet present in the first few spectra*. Varian's vnmrj macro `bc` applies a baseline correction based on the integral regions defined. Unfortunately, it applies it to only one spectrum at a time. To apply it to all spectra in the array, use the `UMbc` macro (Update: recent versions of vnmrj have a new macro, `fbc`, to do baseline correction in an array). Without arguments, `UMbc` calculates a spline curve to be applied to (subtracted from) the spectrum; with one argument it calculates a polynomial function of the indicated order. For example `UMbc(5)` uses a polynomial of order 5. Some experimentation may be required to obtain the best results but a polynomial of 5th order is a good starting point.

Inspect *all* spectra to verify that the baseline correction was effective, that the phase of all the spectra is correct, that peaks in different spectra didn't move out of their integration regions, that the integrations are correctly phased (slope and bias of the integral line), etc. Sometimes one or more spectra show different baselines, phase corrections or other anomalous behavior and may need to be excluded from the calculations.

To normalize your integrals (assign a value to them), go to the Process, Integration panel, select Partial under Integral Display Mode, position vnmrj's cursor on top of one of the integrals, select Single Peak under Normalize Area To: , enter a value in Integral Area, and click on Set Integral Value.

If you have more than one set of arrayed spectra, or more than one non-arrayed spectrum that you want to use in a kinetics study, and you want to use *exactly the same integral regions, scaling and phase corrections*, the macros `UMsvir` and `UMrtir` can be helpful. First define the integral regions for the first spectrum (or the first array) and save it with `UMsvir`. Then, after loading and processing the next spectrum (or array), the regions defined previously can be loaded with the `UMrtir` macro.

Displaying arrayed spectra

There are many options, commands and parameters for displaying arrays of spectra. In interactive display mode, the "View next" and "View previous" buttons can be used to switch between spectra. Clicking on the Return button will display a new set of icons where the "Stack Spectra Vertically" and "Stack Spectra Horizontally" buttons can be used to display a stacked plot. Since Vnmrj 2.2C the appearance of these buttons has changed a little; and the "Stack Spectra" icons can now be found in the "ArrayedSpectra" vertical panel along with some other related parameters and functions.

	View next
	View previous
	Return
	Stack Spectra Vertically
	Stack Spectra Horizontally

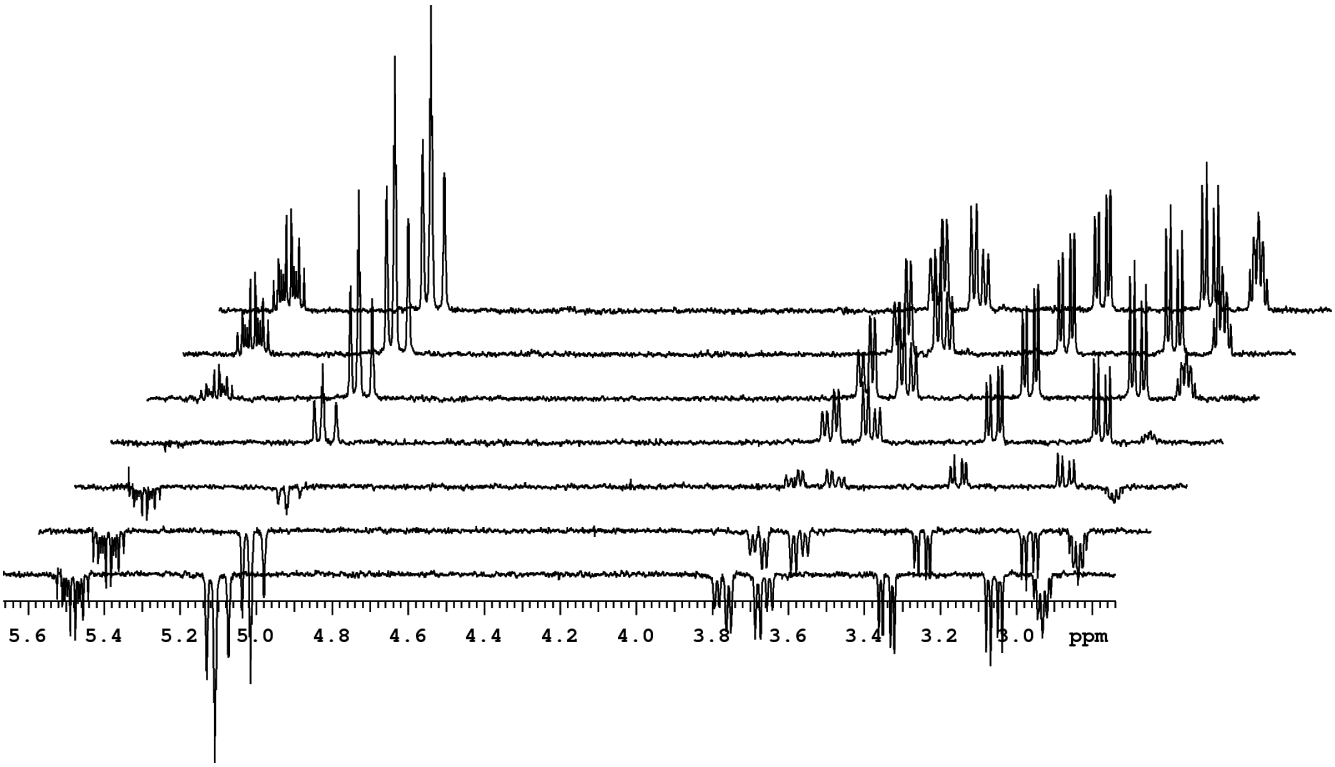
A tilted stacked plot like the one shown below can be setup by manually entering appropriate parameters as described next. But to facilitate the setup this kind of display, the macro `UMdsarray` was written. `UMdsarray(tilt, depth)` uses two arguments: *tilt* and *depth*, where *tilt* is the angle described by an imaginary line going through the left edge of each spectrum in the stacked display and the vertical axis of the page and *depth* is the fraction of the page height that the stacked spectra will occupy. For example `UMdsarray(30, 0.5)` will setup the stack so the "plane" of the spectra is at an angle of 30 degrees with the left edge of the page and the "plane" will occupy about 50% of the area on the page. Playing with different values in your own data will make the usage clearer. Without any arguments, `UMdsarray` will setup the display with default parameters (30 degrees and 0.5). The "correct" parameters depend a lot on personal taste and the features of the spectra that you want to show, so spend as much time as needed to get the right setup.

For complete control of the display, and later appearance of the plot, the following parameters (all in millimeters) are available. Notice that these parameters can be adjusted interactively in the "ArrayedSpectra" vertical panel, on the left side of Vnmrj 2.2C and later. Earlier versions didn't have this panel.

`wc`: (width of the chart) length of each spectrum
`sc`: (start of the chart) distance from the right edge of the first spectrum to the right edge of the display (or page)
`vo`: vertical offset between spectra

`ho`: horizontal offset between spectra; if negative, spectra are displayed from bottom-left to top-right
`vp`: (vertical position) distance from the lower edge of the display (or page) to the first spectrum
`cutoff`: peaks taller than this parameter will be cutoff in the plot or display; set to `'n'` by default.

A reasonable starting point is `wc=200, sc=50, vo=12, ho=-15`.



Plotting arrayed spectra

Once the appropriate display parameters have been found, the `pl` command will plot the array. For example, the plot in the previous figure was made with: "`pscale pl(1, celem) page`". In order to use `pl` to plot an array, at least two arguments must be included: the first indicates which spectrum to start with and the second is the last spectrum to plot. In our example we used `1` and `celem` to start with the first and end with the last stored spectrum (called `celem` for "current element"). Other arguments and options are available; please read the "Command and Parameter Reference Manual" available online from the Help menu. The commands `dsww` (display spectra in whitewash) mode and `plww` (plot spectra in whitewash mode) are also available to produce whitewashed stacked plots and are used in a similar manner.

Analysis, Vnmrj's functions

Vnmrj has built in routines to analyze simple kinetics experiments using *peak intensities*. If the signal decreases exponentially with time, the output is matched to the equation

$$I = a1 * \exp(-t/tau) + a3$$

The analysis is done by the macros `kind` or `kinds` if a short output is desired. If the signal increases exponentially with time, the output is matched to the equation

$$I = -a1 * \exp(-t/tau) + a3 - a1$$

with analysis done by the macros `kini` or by the macro `kinis` for a shorter output table. Read some comments about these calculations at the end of this document.

WARNING: As of Vnmrj 3.0, the macros `kind`, `kinds`, `kini` and `kinis` DO NOT take into account the steady states (`ss` with negative values). The end times they report for each spectrum is inaccurate and therefore their results are wrong. **Do not use these macros if you set up steady states scans with a negative value of `ss` in you spectra**. The end times reported by `UMdli` (see below) are correct.

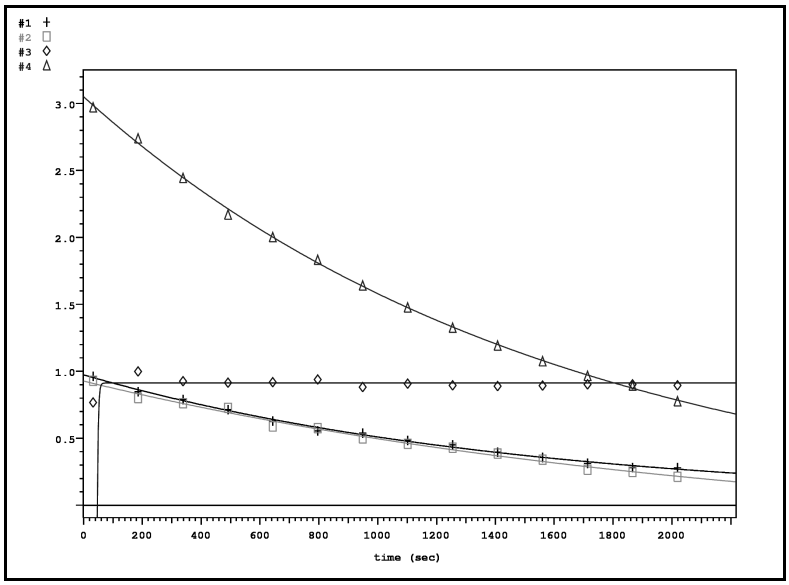
If you want to use the built in routines, follow this procedure.

1. Expand the region with the peak(s) of interest. Click the Threshold button in vnmrj's tool bar and move the horizontal yellow line to define the minimum height for peak selection. 
2. Go to the *Process, Cursors/Line lists* panel and click [Display Line List] and in [Find Peaks in Array] or just type `dll fp` in vnmrj's command line to find the peak intensities in all your spectra. If you want to print a listing of the line intensities to use with a different program, use the command `printon shell('cat '+curexp+'/fp.out') printoff page` (be careful with the spaces). Vnmrj can also analyze the line list as follows.
3. Enter the command `kind`, `kinds`, `kini`, `kinis` as desired. The output from the calculation is shown in the *Process, Text Output* panel. If you want a printed copy type for example "`printon kind printoff page`". If you want to exclude some spectra from the calculation use the `dels(#)` command, where # is the number of the offending spectrum, and redo the calculation.
4. Enter the command `expl` for a graphical display of the exponential fit. This can be plotted with the `pexpl page` command.

The figure below shows the output from the `expl` command and partial results of the calculation for an experiment where peaks decrease over time.

If your peaks do not change in position, shape or line width over time, then using peak intensities is probably ok. But for better results, integrals should be used. Unfortunately, vnmrj's routines use only line intensities. The macro `UMdli` generates a list of integrals for all spectra in the array and writes its output to a file called fp.out, used by vnmrj's kinetics analysis routines, to trick vnmrj into using integrals instead of intensities. If you want to do the analysis using integrals, after running the `UMdli`, continue with steps 3 and 4 described above.

If the position of the peaks drift over time, and peak picking in some of the spectra fails, try the following. In vnmrj's command line type: `create('npoint', 'integer') npoint=50`. Then try getting the list of peaks again with `dll`. The value for npoint can be adjusted to your spectra and it is the maximum number of points that peaks may drift over the array.




```
Exponential data analysis:

peak             tau         error
   1             1490        156.3
   2             2248        543.4
   3            3.153        805.3  solution not found
   4             1592        78.61

peak number 1
 tau =     1.49e+03     error =          156
      time       observed   calculated   difference
      32.8          0.963        0.954      0.00863
     185.6          0.849        0.864      -0.0151
     338.4           0.79        0.782      0.00816
     491.2          0.715        0.708       0.0068
```

If peak picking fails to pick all peaks or picks too many peaks (spectrum is noisy or peaks are very broad), try this. In the command line type: `dll('pos', 1)` and try peak picking again. The number 1 in that macro is the "noise multiplier": small numbers pick more peaks, the default is 3. To plot the picked peaks use for example: `ppf('pos', 1)`.

Analysis, manual method (recommended)

For more complex analysis, you will need to enter your table of integrals and times into an external data analysis program like Qtiplot for Linux and Windows or Origin, Scientist, SigmaPlot, etc., for Windows, and perform a line fitting routine. Vnmr'js macro `dli` creates a list of the integrals in the current spectrum, but it does not give the full listing for all spectra in the array. The macro `UMdli` will generate a full listing of integrals for all the spectra in the array, along with their calculated start times, and will ask if you want to print it or send it by email. This last option can be convenient if you want to import the data directly into another program for analysis. You can just copy the integral table in the email message and paste it into the spreadsheet of your line fitting program for analysis. If `UMsetupkinetics` was used to setup the experiment and calculate the `pad[]` parameters, and if the acquisition was started with `au`, `UMdli` will also report the actual time in seconds recorded when each spectrum in the array was completed.

Relevant Custom-made macros

`UMtime`	Calculates the total experimental time, including all `pad[]`. With a time in seconds as an argument, calculates the number of transients.
`UMsetupkinetics`	Calculates the pre-acquisition delay, `pad[]`, in a kinetics experiment. Sets up a new `end_time` parameter to store the actual end time of each spectrum.
`UMdsarray`	Easy display of an array of spectra in whitewashed stacked mode.
`UMbc`	Baseline correction of all spectra in an array.
`UMdli`	Display list of integrals of all spectra in an array.
`UMunarray`	Extracts the spectra from an array into individual files, allowing separate processing of the component spectra of the array.
`UMsvir`	Saves the integral regions defined in the current spectrum in a file.
`UMrtir`	Retrieves the integral regions saved in a file with `UMsvir` and applies them to the current spectrum.
`UMkin2nuc`	Sets up a kinetics experiment with sequential acquisition of 2 nuclei. Read the separate writeup "*Kinetics experiments with two nuclei*".
`UMarrayfids`	Converts a series of individual fids into a single arrayed file. To be used with `UMkin2nuc`.

For a detailed explanation of these macros please review the document "Custom macros at U of M" available in the Documentation section of our web site.

Please read the online Vnmrj Manuals "*Liquids NMR*", sections 9.2 to 9.4 and "*Command and Parameter Reference*" for more information and even more commands and options.

Appendix

The following email thread, which appeared in an NMR user's group, may help understand how the calculations are performed by Vnmrj.

Date: Thu, 19 Apr 2007 12:54:38 -0500
From: Shawn Carter
Subject:

We are performing a series of kinetics experiments with the NMR that involve increasing exponential signals. According to page 138 of the VnmrJ Liquids NMR User Guide we can use the software to evaluate the data for us. According to the guide, the software uses the following equation:

I=A1*EXP(-T/TAU)+A3

Are questions are:

What are A1 and A3?
How does the software find TAU?

If we analyze 119 spectrum of an array experiment, we seem to be getting surprisingly different answers (using the kinetics program built into the Varian software) than if we do a full analysis of 120 spectrum. Seemingly, to us, there is not much difference in the 120th spectrum. At least not enough to change our answer.

We appreciate the help,

Shawn
****
Shawn M. Carter, Ph.D.
Dept. of Chemistry and Biochemistry
University of Oklahoma
620 Parrington Oval
Norman, OK 73019
Tel: 405-325-2378
Fax: 405-325-6111
email: <mailto:scarter @ ou.edu>scarter @ ou.edu
****

Date: Fri, 20 Apr 2007 22:46:28 -0400
From: "Krzysztof P. Wroblewski" <krzyszto @ mail.med.upenn.edu>
Subject: AMMRL: Kinetic experiments

The equation I=A1*EXP(-T/TAU)+A3 is generally used for calculation of spin-lattice relaxation times. It was introduced by Sass and Ziessow J.Magn.Reson. 25,263(1977). This paper also explains how the parameters are calculated.
Latter Ejchart et all. J.Magn.Reson. 59,446(1986) proved that using this equation for T1 calculation could give wrong results, and proposed a correct method. For some reason both Bruker and Varian still use the older algorithm.

Krzysztof P. Wroblewski, Ph.D.
University of Pennsylvania School of Medicine
Room 96 John Morgan Building
Philadelphia, PA 19104-6055
phone: (215) 898-6396 fax: (215) 898-5642

Date: Sat, 21 Apr 2007 06:13:57 -0500
From: "William C. Stevens"
Subject: AMMRL: Kinetic experiments

How, indeed, does one find tau? 'Tis a consummation devoutly to be wished. To die, to sleep-- to sleep--perchance to dream of tau: ay, there's the rub, whence cometh tau? This must give us pause. There's the respect that this was but one of the despised pangs of learning in undergraduate chemistry.

Is tau for me the same as tau for others? And what of A1 and A3? Perish the thought that they be consigned to the realm of arbitrariness by mere mortals such as us. Much worse the prospect that tau may be "found" by such a mere thing as "software," for what is that thing but ware that is soft, with this regard their currents turn awry and lose the name of action.

We must all seek tau. It resides in our hearts and souls, not just in some simple table of numbers, such as we may determine by our mortal choices as being a measurement of man's worth or some interval between a pulse inverse to most and one of rectitude to the same stage.

And what of A1 and A3? We know not even their dimensions. Thus conscience does make cowards of us all, but it matters not - for it is neither A1 nor A3 that we seek, but rather the revelation to us of tau - -

- - which is different for all of us who will shuffle off this mortal coil. Different especially if we fail to degas our samples to constant T1.

William S.

William C. Stevens, Ph.D. Director
Nuclear Magnetic Resonance Facility
Southern Illinois University
Carbondale, IL 62901
618-453-6498 voice / -6408 fax / 521-9892 cell
http://opie.nmr.siu.edu