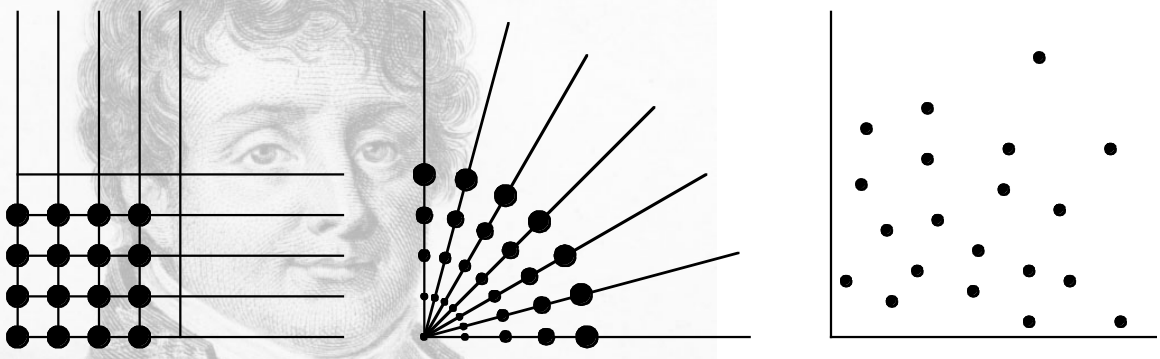
Processing NMR data sampled on a non-cartesian grid using Fourier transform

Introduction:

Acquiring multidimensional NMR spectra becomes a time-demanding process if high resolution is needed along all dimensions. Whereas all data points are sampled on-the-fly along the *direct* acquisition dimension, the same experiment has to be repeated with a modified timing for each new point along the *indirect* dimensions. The spectral width in an indirect dimension defines the dwell time of the corresponding sampling (Nyquist sampling theorem), whereas the resolution of the spectrum is related to the largest increment that can be sampled.

To speed up the acquisition, several groups have proposed new acquisition schemes, i.e. where data are not sampled on a regular Cartesian grid. Let us emphasize that, in the case of direct acquisition, nothing can beat in terms of efficiency the acquisition on a regular grid. The following discussion thus applies on indirect dimensions only. In the following figure are shown various acquisition schemes for the (t_1,t_2) planes in a 3D experiment.



Cartesian Radial Random

Only in the case of data acquired on a Cartesian grid, one can replace the 2D FT by two successive 1D FTs, which order is irrelevant. When the number of data points is a power of 2 (2^n) the discrete FT can be further replaced by the fast FT algorithm.

For the two other schemes, the time variables t_1 and t_2 are no longer separable and one cannot process first the rows and then the columns. The program described in this tutorial, which is a two-dimensional implementation of FT, can deal with such cases.

In order to acquire non-cartesian NMR data and process them using FT, two different pieces of software are needed:

(1) a program to generate the pairs of $[t_1,t_2]$ values before acquisition.
(2) a program to perform the 2D FT on the acquired data.

Preacquisition software:

As mentioned earlier, the t_1 and t_2 time variables are generally not separable. In the Cartesian case, the pulse sequence is repeated for

$t_1(i)$ (i=0,…, M-1) and $t_2(j)$ (j=0,…, N-1), leading to a MxN matrix.

In a more general case, the pulse sequence is acquired for

$t_1(k)$, $t_2(k)$ (k=0,…, L-1), leading to a vector of length L.

These values are supplied to the spectrometer to run the 3D acquisition accordingly. Note that the *apparent* dimensionality of the experiment might be different.

In the case of Varian spectrometers, delays associated with indirect dimension can be incremented *implicitly* by defining a spectral width (sw1, sw2) and a number of points (ni and ni2). There is also a way to increment *explicitly* these delays by defining an array of d2 and d3 value.

For radial sampling the computation of the increments [$t_1 \sin(\alpha)$ and $t_1 \cos(\alpha)$] is straightforward and can be performed inside the acquisition software.

In contrast for the random sampling, a stand-alone C-program is used to generate the delays: two text files where the d2 (respectively d3) are defined explicitly are created:

d2[1]=0.0000000	d3[1]=0.0000000
d2[2]=0.0004318	d3[2]=0.1038293
d2[3]=0.0019830	d3[3]=0.0742228
d2[4]=0.0044466	d3[4]=0.0982887
d2[5]=0.0037980	d3[5]=0.0507677
d2[6]=0.0099833	d3[6]=0.0580311
d2[7]=0.0042412	d3[7]=0.0122184
…	…

In order to perform the same type of experiment on a Bruker spectrometer, the user would have to write a conversion script (using *perl* or *awk*) to make these file understandable by acquisition software spectrometer.

Quadrature detection: Even with alternate sampling schemes, quadrature detection in t_1 and t_2 is still mandatory to discriminate positive and negative frequencies. Phase cycling

and/or gradient selection can be employed. The only potential difficulties that may arise are related to the mode of data storage (interleaved or not) and to the correct linear combination of data points (sensitivity-enhanced acquisition). If mirror images are present in the transformed spectrum, these two aspects should be scrutinized.

Generalized Fourier transform:

Let us start from the continuous 2D Fourier transform

$$F(\nu_1,\nu_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t_1,t_2)\exp(-2\pi i \nu_1 t_1)\exp(-2\pi \tilde{i} \nu_2 t_2)dt_1 dt_2 \qquad (1)$$

where symbol $\tilde{i}$ is used to denote the squared root of (-1) for the second dimension. This formal notation is essential for the practical implementation of the 2D FT in order to distinguish the complex pairs along the two dimensions.
In the case of a discrete 2D transform the double integral of Equ. (1) is replaced by a double *weighted* sum.

Cartesian sampling:

Starting from a (M×N) time-domain matrix sampled on a Cartesian grid, the *discrete* 2D transform is calculed as:

$$F(\nu_1^m,\nu_2^n) = \sum_{j=0}^{M-1}\sum_{k=0}^{N-1} f(t_1^j,t_2^k)\ B_{jk}^{mn}\cdot\Delta t_1^j\cdot\Delta t_2^k \qquad (2)$$

where $$B_{jk}^{mn} = \exp(-2\pi i \nu_1^m t_1^j)\cdot\exp(-2\pi\tilde{i}\,\nu_2^n t_2^k) \qquad (3)$$

Because the data are equispaced in the time domain, the area associated with each point are identical and can thus be discarded.

$$F(\nu_1^m,\nu_2^n) = \sum_{j=0}^{M-1}\sum_{k=0}^{N-1} f(t_1^j,t_2^k)\ B_{jk}^{mn} \qquad (4)$$

Polar sampling:

The running variables in the time domain are:

$$t_1^j \rightarrow t_1^j\cos(\alpha_2^k) \qquad t_2^k \rightarrow t_1^j\sin(\alpha_2^k) \qquad (5)$$

Thus, the 2D FT can be expressed as:

$$F(\nu_1^m,\nu_2^n) = \sum_{j=0}^{M-1}\sum_{k=0}^{N-1} f(t_1^j,\alpha_2^k)\ A_{jk}^{mn}\cdot\Delta t_1^j\cdot(t_1^j\Delta\alpha_2^k) \qquad (6)$$

with $$A_{jk}^{mn} = \exp(-2\pi i\cdot\nu_1^n t_1^j\cos(\alpha_2^k))\cdot\exp(-2\pi\tilde{i}\cdot\nu_2^m t_1^j\sin(\alpha_2^k)) \qquad (7)$$

The assumptions that permit to forget the area associated with each point in the cartesian case are no longer valid here. Knowing the increments in the radial and angular direction (Δt and Δα), the area can be computed *analytically* during the FT.

Random sampling:

The running variables in the time domain are given by two vectors of length L:

$$t_1^i \ , \ t_2^i \qquad (8)$$

Thus, the 2D FT can be expressed as a single sum:

$$F\left(\nu_1^m, \nu_2^n\right) = \sum_{i=0}^{L-1} f\left(t_1^i, t_2^i\right) C_i^{mn} \cdot \ S_i \qquad (9)$$

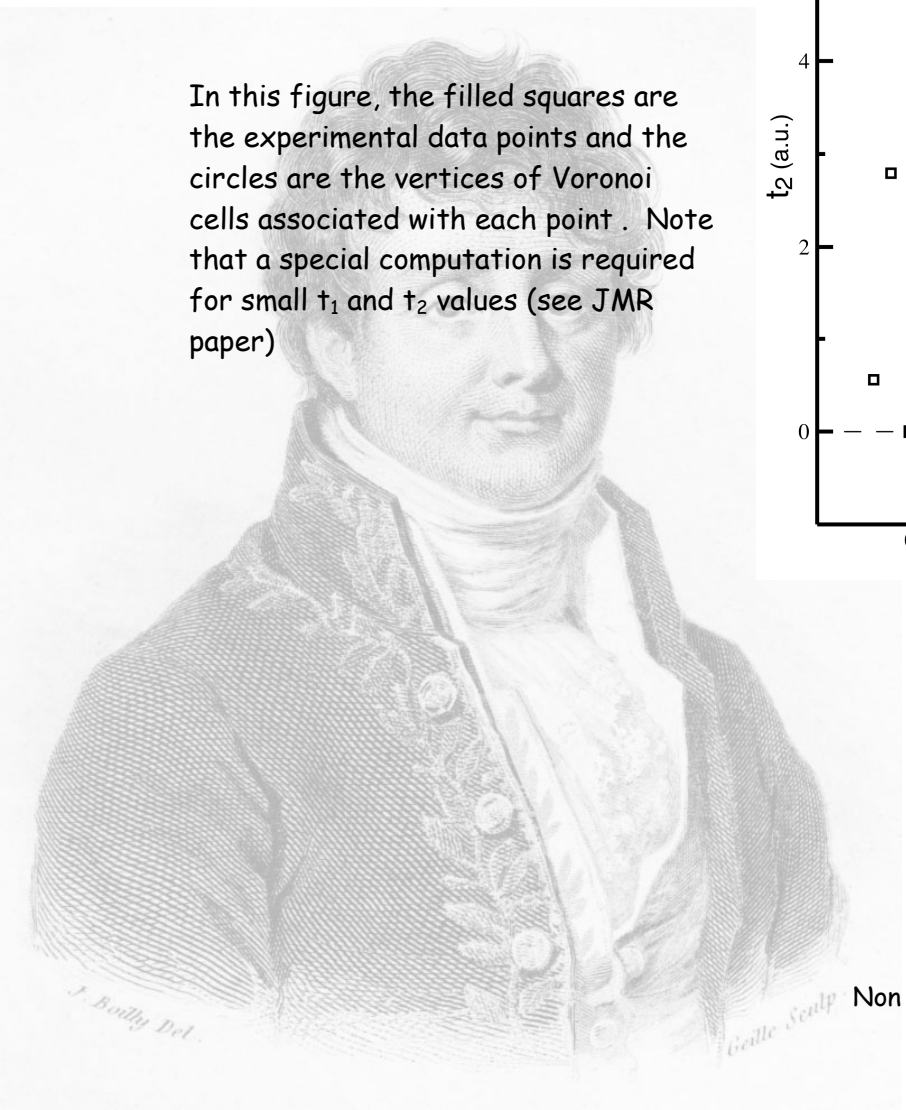
with $$C_i^{mn} = \exp\left(-2\pi i \nu_1^m t_1^i\right) \cdot \exp\left(-2\pi \tilde{i} \nu_2^n t_2^i\right) \qquad (10)$$

and S_i the integration area associated with each experimental data point. Because the points are sampled in a random manner, this surface S_i cannot be derived analytically but will be computed as a *Voronoi cell.*

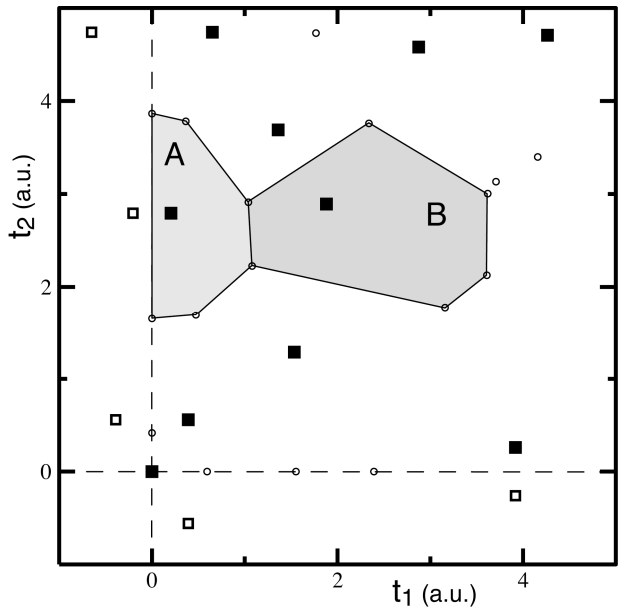
Let us consider a discrete set of points (p_a, p_b,...) in plane. The *Voronoi cell* (C_i) associated with the point s_i is made of all points x of the plane that are closer to p_i than any other points of the set S.

$$C_i = \left\{ x : \ |p_i - x| \le |p_j - x| \ , \ \forall j \neq i \right\} \qquad (11)$$

Voronoi cells are described as a convex polygon in a 2D. To evaluate these cell area for arbitrary sets of points, we have chosen "Qhull", a robust practical convex hull algorithm developed at "The Geometry Center" of the Univ. of Minnesota, USA.



In this figure, the filled squares are the experimental data points and the circles are the vertices of Voronoi cells associated with each point . Note that a special computation is required for small t_1 and t_2 values (see JMR paper)



Practical aspects:

To run these experiments, you need 3 programs:

- Coord_gen (Generation of random acquisition)
- Direct2DRFT_full (Generalized 2D FT)
- Qhull (Voronoi cells computation)

The two programs "Coord_gen" (Generation of random acquisition) and "Direct2DRFT_full" (Generalized 2D FT) are provided as executables for MacOS (Intel) and for Linux (i386). Two shells script that call these programs are given to illustrate their use; the user should modify them to suit his/her need.

Qhull should be downloaded from the following URL: http://www.qhull.org/ and compiled.
Additional shell scripts are provided on this WEB site to use this software.
NmrPipe can be downloaded from NIH.

(1) Generation of the acquisition scheme:

In the case of Cartesian and radial acquisition, the incremented t_1 and t_2 value can easily be evaluated inside the acquisition software. In the case of random sampling, a C-program "Coord_gen" was written for this purpose:

```
Coord_gen —name  [Filename (string)] \
—sw1   [SW1 (float)]  —sw2   [SW2 (float)]  \
—size1 [SIZE1 (float)]  —size2 [SIZE2 (float)] \
—n    [Number_of_points (int)] \
—sigmfact1 [Standard_Dev1 (float)]   —sigmfact2 [Standard_Dev2]
```

"Filename" is used as a root for the various output filenames. SW1 and SW2 are the spectral widths in the two dimensions. The number of points to be generated is given by "n". SIZE1 and SIZE2 refer to the corresponding length of a standard signal sampled on a cartesian grid and Standard_Dev1 and Standard_Dev2 define the width of the distribution (a narrower distribution is obtained for larger value).

"Coord_gen" uses the subroutine "gsl_ran_gaussian" of the GNU scientific library, which returns a Gaussian random variate with mean zero.

Four output files are created: two to be used as macro on the spectrometer "***_d2" and "***_d3", one with the global information "***.info" and one "***" for the evaluation of the area of the Voronoi cell. In this last file, the data points have been mirrored

with respect to the origin axes for proper computing of the area close to t_1=0 and t_2=0 (first data point scaling).

At this stage, one can compute the area of the Voronoi cells using two programs "qconvex" and "qvoronoi" which are components of the Qhull software. Not necessary for the data acquisition, these values will be read by the generalized 2D FT program. Note that they have to be evaluated only once even if several [t_1,t_2] planes are processed.

Non linear sampling algorithms introduce artifacts in the frequency domain spectra, which look like "noise" because of the random nature of the sampling scheme. At this stage one can analyze them by using simulated data (a single non decaying sinusoid) as input and computing the corresponding point spread function (PSF).

(2) Generalized 2D FT:

Let us consider a generic triple resonance experiment, where the ^{1}H are sampled during the acquisition (t_3) and two nuclei during indirect evolutions [t_1,t_2].
The data will be first transformed in a standard manner along t_3. We have written a C-program "Direct2DRFT_full" to carry out the remaining FT in the three above-mentionned sampling: Cartesian, polar and random. The first case aims at comparing with regular FFT and at debugging possible file format issues (relative location of complex pairs),

```
Direct2DRFT_full —name  [Filename] \
—mode   [cart, gauss, polar]
```

As all nmrPipe programs, this software reads its standard input for data to process and pipes it to the standard output. For each [t_1,t_2] pair, it expects 4 vectors containing sequentially the data as RR, RI, IR, and II (non interleaved data). It may be necessary to reshuffle the data points prior processing.

The size of input data is defined by the parameter NDAPOD (size of the apodization window) and that of the output data by NDSIZE. The nmrPipe "ZF" command can be used to provide space for the transform, while updating the parameters in the file header.

In the case of random sampling only, the user has to provide the values of the time increments ("***_d2" and "***_d3") as well as the Voronoi cell area ("***.area"). These files should be in the local directory and the root of the filename "***" is provided as a command-line argument.

One should remember that any implementation of FT introduces some arbitrary scaling factor and may reverse the frequency axes.

References:

❍ Pannetier N, Houben K, Blanchard L, Marion D.
J Magn Reson. 2007 *186*: 100-107
Optimized 3D-NMR sampling for resonance assignment of partially unfolded proteins.

❍ Marion D.
J Biomol NMR. 2006 *36*:45-54.
Processing of ND NMR spectra sampled in polar coordinates: a simple Fourier transform instead of a reconstruction.

❍ Marion D.
J Biomol NMR. 2005 *32*:141-50.
Fast acquisition of NMR spectra using Fourier transform of non-equispaced data.

❍ Coggins BE, Zhou P.
J Magn Reson. 2007 *184*:207-21.
Sampling of the NMR time domain along concentric rings.

❍ Coggins BE, Zhou P.
J Magn Reson. 2006 *182*:84-95.
Polar Fourier transforms of radially sampled NMR data.

❍ Kazimierczuk K, Zawadzka A, Kozmiński W, Zhukov I.
J Biomol NMR. 2006 *36*:157-68.
Random sampling of evolution time space and Fourier transform processing.

❍ Kazimierczuk K, Kozmiński W, Zhukov I.
J Magn Reson. 2006 *179*:323-8.
Two-dimensional Fourier transform of arbitrarily sampled NMR data sets.

Contact

Any further questions and comments should be directed to:
Dominique.Marion@ibs.fr