Journal of Magnetic Resonance **145,** 201–208 (2000)
doi:10.1006/jmre.2000.2071, available online at http://www.idealibrary.com on IDEAL®

NMRLAB—Advanced NMR Data Processing in Matlab

Ulrich L. Günther,[1] Christian Ludwig, and H. Rüterjans

J. W. Goethe University, Frankfurt, Institute for Biophysical Chemistry, Biocentre N230, Marie-Curie-Strasse 9, 60439 Frankfurt, Germany

Received August 18, 1999; revised February 21, 2000

NMRLAB is a toolbox for NMR data processing in MATLAB (The Mathworks). MATLAB is a matrix-oriented high-level programming environment which gives access to fast algorithms for a large number of numerical tasks on many common computer platforms. To take advantage of fast matrix operations in MATLAB most processing commands in NMRLAB have been vectorized. Data processing can be achieved either by scripts or by a user-friendly command structure. An interface to WaveLab enables spectral denoising employing wavelet transforms. The use of wavelet denoising is demonstrated for one- and two-dimensional data. © 2000 Academic Press

INTRODUCTION

With the fast development of multidimensional NMR spectroscopy the need for appropriate data processing has grown accordingly. Modern NMR data processing must not only keep pace with NMR spectroscopy but also with signal processing algorithms and computer technology. Scientific data analysis benefits from precise knowledge about processing algorithms including their individual implementation and thus from access to program sources. Even with source code available the readability of program sources is often impaired by the complexity of the programming language. This problem is significantly reduced in high-level languages which are often used in quantitative programming environments (QPEs). Because code is easily readable and because the implementation of programs is fast, QPEs have frequently been used to develop and test algorithms. More recent versions of QPEs provide access to complex graphics combined with tools to build user interfaces and the possibility to interact with C++ code. The possibility of linking code with other toolboxes is a paramount feature to build programs using state of the art algorithms for signal processing and postprocessing data manipulation. An increasing need for advanced processing and postprocessing schemes in NMR spectroscopy demands a highly flexible NMR processing package with access to other toolboxes. Such a program, to be suitable for data processing and analysis in structural biology, must be combined with a user-friendly interface and an intuitive command structure.

In the past, NMR processing packages have concentrated on different aspects of those requirements. NMRPIPE (*1*) has a superb concept using UNIX pipes to exchange streams of data between different processing steps. FELIX (Molecular Simulations) and XWINNMR (Bruker) are highly professional commercial NMR processing packages. Other NMR processing packages such as GIFA (*2*), NMR Toolkit (*3*), and PROSA (*4*) have been implemented by different NMR laboratories. Here we present NMRLAB, a MATLAB NMR data processing toolbox. MATLAB, MATCOM (Mathtools), Octave, and SciLab are high-level programing languages which share a similar programing language and give access to an immense number of numerical algorithms including fast Fourier transform, numerous matrix operations, solutions for eigenvalue problems, singular value decomposition, QR-decomposition, and various data fitting routines (Newton–Raphson, Marquardt–Levenberg). MATLAB offers a complex compiler option, MATCOM is itself a compiler for MATLAB programs, and Octave and SciLab are both open source implementations of QPEs using a MATLAB-like language.

An increasing number of MATLAB toolboxes is available for NMR data processing including several wavelet packages (MATLAB wavelet toolbox, WavBox from Computational Toolsmiths, Rice wavelet toolbox, UniWave and WaveLab (*5, 6*) toolbox), signal processing toolboxes (MATLAB signal processing toolbox, various time-frequency signal analysis toolboxes (*7, 8*), data optimization, and statistical toolboxes. Advanced graphical tools are available in MATLAB and in some add-on toolboxes. More recent versions of MATLAB give easy access to a large selection of graphical functions. An important advantage of QPEs is that basic numerical algorithms are maintained by numerical mathematics experts. NMRLAB has been written to run on any platform for which MATLAB is available including popular personal computer operating systems such as Microsoft Windows, LINUX (Free Software Foundation), Macintosh computers, and various flavors of UNIX workstations.

The current version of MATLAB does not require the initialization of variables and memory allocation. This has the inherent disadvantage that it is not possible to take advantage of data pointers to pass variables by reference. However, it is possible to allocate and free computer memory at the run-time of the QPE. The need for data pointers can often be circum-

[1] To whom correspondence should be addressed.

1090-7807/00 $35.00

```
================================ EDP ================================
NMR data sets in NMRDAT:
SET: 1 EXP: 1  NAME: CX_NOESY  |  SER size: 0  0 | MAT size: 1024  1024 * P
=====================================================================
TOTAL MEMORY: Matrices: 0, SER files: 1048576.
=====================================================================
  * current data set,  P selected for plot
=====================================================================
CURRENT DATA SET: 1,1.
(1)  DIMENSION 1                    (A)  ABS Baseline correction.
(2)  DIMENSION 2                    (WN) Wavelet De_N_oise.
(3)  DIMENSION 3                    (CZ) Cadzow function.
(C)  InCrementation Scheme.         -----------------------------
(D)  Display current settings.      (CS) Change current data set.
(G)  Gibbs (first point times 0.5). (U)  Update current data set.
(M)  SMooth FID.                    (SP) Save parameters.
(B)  Baseline correction (bc)       (SA) Save to all EXP in SET.
(P)  Phase correction.              -----------------------------
(L)  Linear Prediction.             (H)  HELP.
(R)  Referencing.                   (Q)  Quit and Save.
(S)  Strip Transform.               (X)  Exit without save.
(RM) Revert matrix.                 (K)  Keyboard shell.
(W)  Window function.
(Z)  ZeroFilling.
------------------------------ DIM 1 -------------------------------
NMRLAB:>
```

FIG. 1. EDP setup routine.

vented by using global (or persistent) variables. Performance penalties of QPEs are compensated by highly optimized algorithms combined with the possibility of compiling and optimizing source code.

METHODS

Concepts. The core of NMRLAB is a collection of NMR data processing routines. The basic structure of any processing command is matrix_{out} = function(matrix_{in}, parameters). This requires that the two matrices, mat_{in} and mat_{out}, must be held in the computer memory (RAM) at one time. In the case of three-dimensional (3D) data a minimum of one 3D matrix plus two two-dimensional slices must be kept in RAM. For improved performance NMRLAB can be set up to perform 3D matrix predominantly in RAM without swapping to disk. For a 2D NOESY spectrum with 1024×1024 complex data points, $2 \times 10^6 \times 8$ bytes = 16 MB of RAM must be allocated for double precision variables. For data storage a typecast to single precision can save disk space.

Although all processing commands are fully scriptable the typical usage will be through a setup routine (`edp` = edit data processing parameters) followed by `xfb` (execute Fourier transform—and associated processing commands—in both dimensions of a two-dimensional data set). Figure 1 shows the starting screen of `edp` with typical processing parameters such as phase incrementation schemes (TPPI (*9*), States (*10*), States–TPPI (*11*), and echo–antiecho sensitivity-enhanced (*12*) data), apodization functions, linear prediction, and baseline correction. Some functions allow different choices for phase deconvolution of digitally filtered data as is common on Bruker spectrometers where the FID in the fast dimension is preceded by a filter function. `Edp` contains setup routines for all processing functions of NMRLAB. Automatic processing using `edp` and `xfb` keeps track of the complete processing history. `Edp` contains routines to reference heteronuclear spectra using parameters determined by Wishart *et al.* (*13*) for calculating ^{15}N and ^{13}C chemical shift references from a temperature-dependent proton chemical shift reference.

Processing functions. All NMR processing functions in NMRLAB will automatically perform the same task on all columns of a two-dimensional data matrix. Table 1 provides a comprehensive list of the processing commands in the current version of NMRLAB. Most data processing functions have numerous options which are documented in their header (accessible by the `help` function in MATLAB). Usually a vectorized design employing matrix operations on the complete two-dimensional data matrix has been implemented. For example, `wdwf2` calculates a window function which is subsequently applied in one step to the entire input matrix. This is achieved by creating an intermediate matrix with identical copies of the window function in all matrix columns the same size as the source data matrix which is then multiplied point by point (MATLAB notation: $\text{mat}_{out} = \text{mat}_{in} \cdot \text{wdw}_{matrix}$). This type of calculation requires enough memory for two 2D matrices and has been used for phase correction, DC offset correction (`bc`), data shifts (`clsh, crsh, lsh, rsh`), and phase correction. With sufficient amounts of memory available such matrix calculations are fast owing to internal optimizations of the QPEs. Some processing functions take direct advantage of vectorized routines in the QPE such as fast discrete Fourier transform (`fft, ift`), which is implemented in MATLAB 5 as a radix-2 fast Fourier transform algorithm, singular value decomposition, or polynomial fitting. The computational speed of vectorized routines in MATLAB is demonstrated by a fast Fourier transform of a 1024×1200 point matrix which is accomplished in 2.7 s and a phase correction of the same matrix which requires 12.4 s with a preceeding Hilbert trans-

TABLE 1
NMR Processing Functions in NMRLAB

Function	Description
hft	Hilbert transform
fft[a]	Fast Fourier transform
ift[a]	Inverse fast Fourier transform
dft	Fourier transform of Bruker digital filtered data
rft	Real Fourier transform for TPPI-type data
smo	Smooth = polynomial solvent filter (*29*)
sol	Solvent filter by time-domain convolution (*30*)
wdwf2	Window functions (gm, em, sine bell, cubic sine bell)
flatten2	FLATT baseline correction (*14*) (uses flattent and chi2_flatt)
chi2_flatt	Calculate chi2 vector for FLATT
absc	Call and setup for FLATT baseline flattening (calls flatten2)
lpsvd2	SVD-based linear prediction (*15, 16*)
lpx2	Linear prediction (LPC, Prony, or Steiglitz–McBride)[b]
rev	Reverse data
csh1	Circular shift left
cshr	Circular shift right
sh1	Shift left
shr	Shift right
revm	Reverse data in one dimension
revm2	Reverse matrix in both dimensions
transm	Transpose real or complex matrix (same as ctranspose in MATLAB)
phase	Phase vector or matrix (called by uiphase)
strip	Cut a strip out of a matrix
cadzow2	Perform cadzow algorithm on a two-dimensional matrix

[a] MATLAB built-in function.
[b] Requires MATLAB signal processing toolbox.

form and 6.8 s for the actual phase correction on a 400-MHz AMD personal computer with 128 MB of RAM running the Linux operating system (kernel version 2.2). Complete two-dimensional processing of the same data matrix employing polynomial smoothing with a polynome of ninth order on the free induction decays (FIDs) in the fast dimension, a cubic sine bell apodization function in each dimension, and phase correction in either dimension is completed in 105 s on the same computer, without the polynomial smoothing in 51 s.

Although all functions in NMRLAB are written to be applied to two-dimensional matrices some cannot be vectorized because data fitting is involved (smo, flattent) or because the one-dimensional operation requires internal matrix operations. For example the χ^2 calculation of flatt (*14*) (chi2_flatt) requires a matrix operation to calculate a χ^2 vector in one step for a single line of the spectrum. In contrast to the original publication (*14*) we calculate the actual baseline preferentially by a polynomial fit to the baseline points rather than a Fourier series because the use of a Fourier series had a tendency to cause artifacts as a consequence of its periodicity. For zero-filled data an extra option (tfact) uses a reduced number of data points to calculate baseline points.

Linear prediction was implemented as originally described by Kumaresan and Tufts (*15*) employing singular value decomposition, root reflection, and mixing of forward–backward LP as proposed by Zhu and Bax (*16*). The linear prediction routine (lpsvd) can display complex roots for forward and backward linear prediction and complex roots after root reflection and mixing. Linear prediction can also be achieved employing functions in the MATLAB signal processing toolbox. Lpc (*17*) uses the maximum entropy method for spectral estimation and calls the Levinson recursion. It is much faster than singular value decomposition-based linear prediction and produces good results for common NMR data sets. Similar to lpsvd the predicted signal discriminates signal against noise. Prony's method (*18*) and an extension by Steiglitz and McBride (*19*) are also available in MATLAB. Both methods tend to fail for decaying periodic data and should only be used for constant time data.

Cadzow is a powerful filter technique which can be used to denoise spectra (*20*). It will create a matrix with Hankel structure from the free induction decay, perform a singular value decomposition, and resynthesize the FID by picking a user-specified number of singular values (the number of wanted signals). By selecting the largest singular values noise is rejected. After two to three cycles an almost noise-free spectrum is obtained. It has been shown that this technique is useful to remove noise from *in vivo* spectra with a limited number of peaks (*21*). The Cadzow algorithm has recently been used to remove large diagonal peaks from NOESY or TOCSY spectra by removing the largest singular values before the reconstruction of the data (*22*).

A series of functions which are applied to the FID (wdwf2, smo, and bc) have special modes for digital data originating from Bruker spectrometers where a filter function preceeds the actual data points of the FID. If these points are simply omitted, "frowns" or "smiles" are introduced at the edge of the spectrum. If a polynomial is fitted to the FID and later subtracted to remove on-resonance solvent peaks (smo), artifacts may be introduced if the negative time filter points are not excluded. For this reason smo, sol, and bc, which subtract a constant from all FIDs to remove DC offsets, exclude the digital filter from the calculation. A similar problem arises for the window function. Frequently the filter is moved to the end of the FID by a circular left shift (csh1) before the window function is applied. Depending on the type of window function this procedure will also eliminate the filter and cause artifacts at the edge of the spectrum. The alternative where the window function is applied to the original FID including the digital filter as negative time data points alters the window function parameters. NMRLAB offers two additional modes which are depicted in Fig. 2. Either the filter can be omitted from the window function or the digital filter points can be treated as negative time data points.

3D processing is accomplished by processing of series of 2D data sets. After the first two dimensions have been processed the matrix is transposed to process the third dimension as a

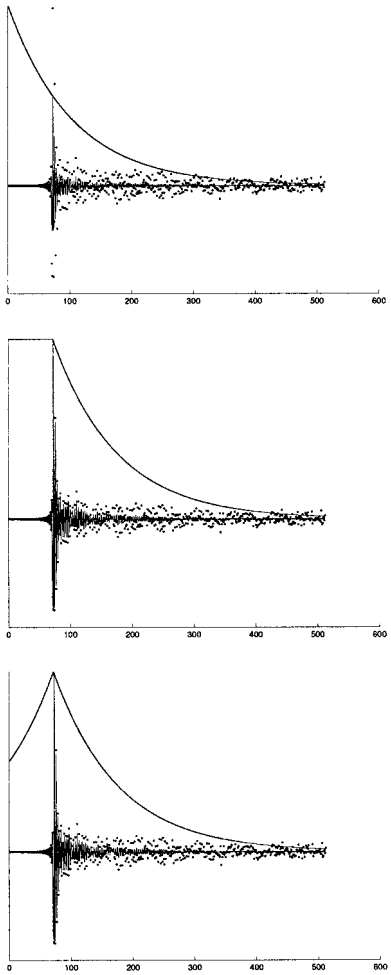

FIG. 2. Modes for an exponential window function for digitally filtered data from BRUKER spectrometers. Dots, FID before application of the window function; solid line, FID after application of the window function. Top, the window function is applied to the FID and the digital filter; middle, the digital filter has been excluded; bottom, a negative time axis is used for the digital filter.

series of 2D matrices. FELIX and XWINNMR use a matrix format with small submatrices for equally fast access to any vector or plane of a 3D matrix. Similar to NMRPIPE, NMRLAB works on series of 2D matrices which are accessed directly as 2D planes and may be combined to a MATLAB 3D matrix. To access XZ or YZ planes the 3D matrix must be transposed. Because matrix transposition is not defined for 3D matrices slices of 2D matrices must be extracted for matrix transposition. The orientation of 3D matrices is altered by a flexible user interface (`xyztranspose`) which is a user interface for the function `transpose3d` which calculates the actual matrix transposition. `Xyzop` determines the transposition (12, 13, or 23) which is required to get from one to any other matrix orientation (123, 132, 231, 213, 312, or 321). Matrix transpositions can either be calculated in RAM, which requires sufficient amounts of memory for two 3D matrices, or with intermediate swapping of 2D slices, where the RAM for one 3D matrix plus that for two 2D slices is sufficient. When 3D matrix processing is set up in `edp` the actual processing can be achieved employing `tf` (three-dimensional Fourier transform) which will automatically transpose the 3D matrix after processing the first two dimensions and return the final matrix in a 213 orientation suitable to display 12 slices.

Wavelet transform filters. NMRLAB gives access to wavelet transform (WT)-based noise suppression techniques using routines from the WaveLab toolbox (*5*). WT-based denoising is performed by a forward WT of the processed real data, the application of a filter, and a backward wavelet transform (IWT). The WT can be one- or two-dimensional. Optionally one can perform a noise normalization prior to the first WT. In the case of 2D and 3D matrices an average noise level must be calculated for the entire matrix. NMRLAB allows the choice of a wavelet transform (periodized orthogonal or translation invariant), a wavelet (Haar, Beylkin, Coiflet, Daubechies, Symmlet, Vaidyanathan), a threshold shrinkage technique (Visu, SURE, Hybrid, MinMax, MAD, Hard, Soft), and a series of parameters and options which are described in a number of publications associated with the WaveLab toolbox (*5*, *6*). Good default values are supplied with the setup routine `edp`. Wavelet shrinkage can be applied at the end of `xfb` or as a separate step after completed processing. Figure 3 shows two sections from a noisy ^{15}N–^{1}H HSQC spectrum of the N-terminal SH2 domain of the p85 subunit of PI3′-kinase (p85 N-SH2) before (left) and after (right) WT shrinkage. Both spectra were first subject to a cubic sine bell apodization in both dimensions. Although wavelet shrinkage can be applied without a preceeding window function a cubic sine bell prior to wavelet shrinkage may be necessary to avoid truncation artifacts. The results of a two-dimensional wavelet shrinkage are similar to those obtained for a one-dimensional wavelet filter applied to each row of the matrix. The two-dimensional approach is much faster (18 s compared to 38 s for the 1D mode for a 512 × 512 point matrix) but limited to square matrices. Noise reduction achieved by WT is superior to that obtained by any window function without the disadvantage of additional line broadening. This is particularly useful for noisy spectra obtained for low-concentration protein samples in SAR (structure activity relationship) by NMR studies (*23*). Wavelet shrinkage can also be combined with resolution enhancement. Figure 4 depicts a 1D ^{15}N HSQC of p85 N-SH2 processed with different options for resolution enhancement and noise reduction. The FID corresponding to spectrum A has been subjected to strong Gaussian broadening (LB = −15 Hz, GB = 0.15) to resolve shoulders in peaks with a low signal-to-noise ratio (arrow). The spectrum in Fig. 4B shows the same spectrum processed without any apodization prior to wavelet denoising with a symmlet (*8*), a periodized orthogonal WT, and soft thresholding. The resulting spectrum is almost free of noise and shows all peaks with poor resolution. Figure 4C shows the

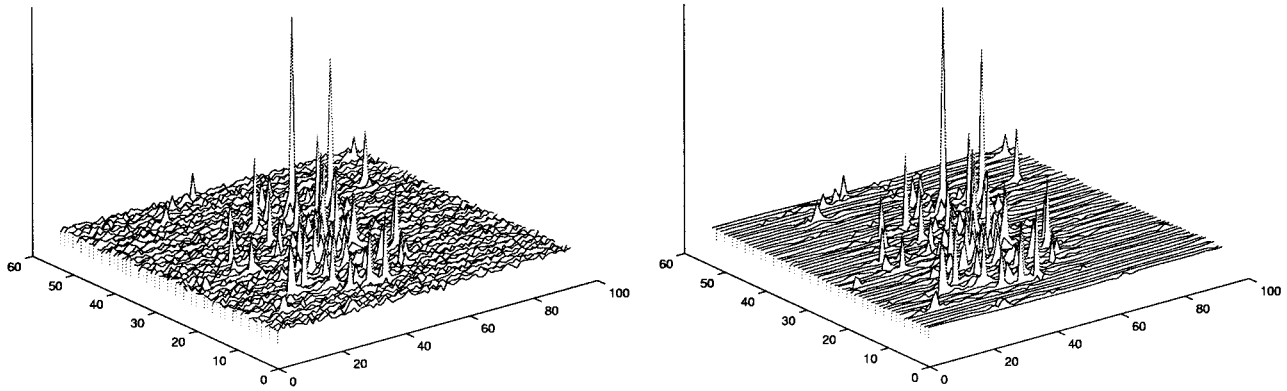

FIG. 3. Left, stack plot of a ^{15}N-HSQC-spectrum of p85 N-SH2 processed with a cubic sinebell apodization in both dimensions. Right, same spectrum treated with wavelet denoising (two-dimensional orthogonal wavelet transform, symmlet-8 wavelet, hard-thresholding, automatic normalization and calculation of the noise level, i.e., threshold = $\sqrt{2 \cdot \log(n)}$ with n = the length of a matrix row).

result of a combination of both approaches, i.e., the FID has been subjected to strong Gaussian broadening and the (real) spectrum has been treated with a wavelet filter. The effect of resolution enhancement has now been achieved at a very modest loss of signal-to-noise. Another application of wavelet transform filters is `wavwat`, which will remove on-resonance water signals similar to `smo`. `Wavwat` will apply a forward WT to the FID followed by a high threshold shrinkage and back-WT. The resulting signal is subtracted from the original FID.

Graphical routines. NMRLAB provides a graphical interface to plot 2D matrices (`uicont`). It is possible to step from slice to slice in 3D matrices or series of data sets obtained for SAR by NMR. Multiple rows or columns from 2D data sets can be picked for phase correction employing a graphical

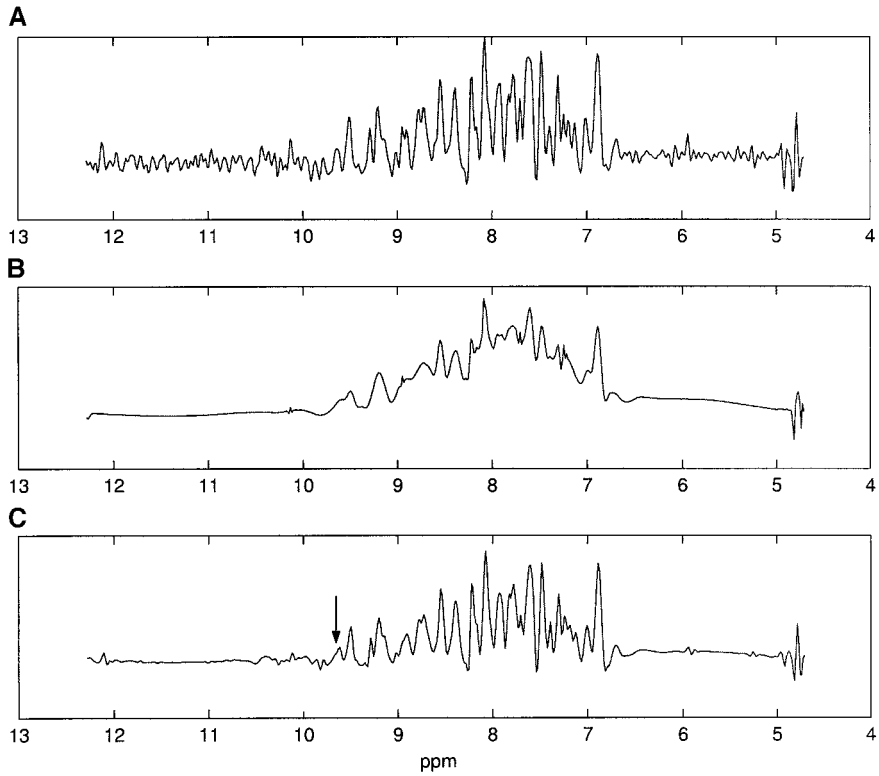

FIG. 4. ^{15}N-filtered 1D-HSQC of p85 N-SH2 processed with different options. (A) Gaussian broadening (GB = −15, LB = 0.15); (B) Application of a wavelet filter employing a Symmlet (8) wavelet and a periodized, orthogonal wavelet transform with a soft thresholding filter (threshold = $\sqrt{2 \cdot \log(n)}$ with n = 512 = number of data points in spectrum). No preceeding apodization function. (C) Gaussian apodization (as in A) followed by wavelet shrinkage (as in C).

TABLE 2
Fields in NMRDAT

ACQUS	2/3D array of structures	Acquisition parameters
PROC	2/3D array of structures	Processing parameters
DISP	Structure	Display parameters
SER	2D matrix	Raw data
MAT	2D matrix	Processed data

phasing routine (`uiphase`). For peak picking in crowded spectral series obtained by ligand titrations a graphical peak picking routine (`sartitr`) is available which plots superimposed series of spectra and allows the user to step from spectrum to spectrum to pick peaks. Peak identification is greatly facilitated by coloring schemes which help to distinguish different spectra in a spectral series and highlight the current spectrum. Peaks lists and lists of chemical shift perturbances can be generated and stored as ASCII or as spreadsheet files.

Data formats and control functions. NMRDAT is the central pool of information and data in NMRLAB. Technically it is a two-dimensional array of structures (NMRDAT(SET,EXP)) which holds data *sets* in the first and *experiments* in the second dimension. A data set can be one or multiple 2D spectra. This concept was originally designed to process (`xfall`) and access series of HSQC spectra in SAR by NMR or relaxation experiments. Within one data set all experiments can be processed with identical parameters. The same concept is also used to process three-dimensional data sets where slices of the 3D data set are stored as experiments of one set. Table 2 lists the fields in NMRDAT. All fields are accessible on the command line or with the aid of `browse` which can list, manipulate, save, import, export, and load data sets. The contents and data sizes of all data sets and experiments together with the total amount of used memory can be listed by `shownmrdat` (=`snd`). For large data sets there are optional tools to swap data to disk (`swapser, swapmat`). The processing commands (`xfb, xf2, xf1,` and `tf`) and the contour plot routine `uicont` recognize swapped fields and load their contents into RAM if necessary. Internal data storage in NMRLAB depends too much on advanced data structures on MATLAB to be compatible with Octave. However, the core processing routines (Table 1) will run in Octave and with few changes in SciLab.

Two- and three-dimensional matrices can be exported to the NMRPIPE (*1*) and NMRVIEW (*24*) file format. This format can be read by major assignment programs such as NMRVIEW and PRONTO (*25*). Import programs are currently available for NMRPIPE and FELIX. A series of commands has been written to access processing and display parameters and to process 2D and 3D data (Table 3). `Edp` and `edd` are simple tools to edit processing and display parameters, respectively. Both have options to initialize data sets. `Xfb` will process one experiment; `xfall` will process a series of experiments in one data set. `Tf` is a similar tool for 3D data processing. Baseline correction, phasing, and wavelet noise reduction can be either part of the general processing using `xfb` (`xfall` or `tf`) or alternatively in a postprocessing step (`absc, phase, denoise`).

DISCUSSION

NMRLAB is a collection of tools which gives NMR users access to the world of MATLAB functions and toolboxes. NMR data processing in a QPE has many advantages. For scientific applications the most important point is probably the high flexibility of such systems. The performance of NMR processing in NMRLAB is comparable to that of other NMR processing software packages and sometimes faster where matrix computations can be performed. Optimizations for processing of spectral series make NMRLAB a powerful tool for processing and interpretation of spectral series obtained in SAR by NMR studies. With increasing amounts of computer memory the concept of data processing in the computer memory will become the method of choice.

Although wavelet transformations have been used for years in geosciences and various fields of electronic signal processing and transmission there have been few applications to NMR spectra (*26*). There are many applications of wavelet transforms including frequency identification, noise reduction, and data compression which may be useful for NMR spectroscopy.

TABLE 3
Control Functions in NMRLAB

Function	Description
`nmrlab`	MATLAB script to setup parameters for NMRLAB
`re`	Read raw data from disk
`relist`	Read series of experiments
`readser`	Read Bruker ser files
`readacqus`	Read parameters from Bruker ser file
`shownmrdat = snd`	Show data sets and sizes in NMRDAT
`uiphase`	Interactive phase correction
`uicont`	Interactive contour plotting
`sartitr`	Analyze series of 2D NMR spectra (e.g., SAR by NMR series)
`edp`	Edit processing parameters
`edd`	Edit display parameters
`browse`	Browse, edit, save, export, and load NMRDAT contents
`xfb`	Process two-dimensional data
`xfall`	Process series of 2D data sets (e.g., SAR by NMR series)
`tf`	Process three-dimensional data
`absc`	2D/3D postprocessing baseline correction
`denoise`	2D/3D postprocessing wavelet denoising
`xyztranspose`	Transpose 3D structures
`xyzop`	Determine transpose action for 3D data sets
`makespc`	Utility to create synthetic 1D spectra

TABLE 4
Wavelet Shrinkage Parameters in NMRLAB

qmf_type	Wavelet type (quadrature mirror filter)	Haar, Beylkin, *Coiflet, Symmlet, Daubechies,* Vaidyanathan
par	QMF paramater	Coiflet: 1–5 (*3*). Daubechies: 4, 6, 8, *10,* 12, 14, 16, 18, 20. Symmlet: 4–10 (*8*).
thr_type	Type of shrinkage	*Hard, soft,* SURE, Hybrid, MinMax, MAD
L	Low-frequency cutoff for shrinkage.	Must be $\ll J$, $N = 2^J$ (*2–4*) N = number of data points
normalize	Normalize noise[a]	
WT_type[b]		*Periodized orthogonal,* fully translation invariant
thr	Threshold value	*Universal* $\sqrt{\log(n)}$, n = number of data points.

Note. Parameters which yield good results for most NMR spectra are italicized.

[a] 2D and 3D versions of normalization have been implemented in NMRLAB.

[b] Other wavelet transforms (i.e., the Meyer wt) are available in WaveLab.

The availability of fast algorithms for one- and two-dimensional wavelet transforms enables routine application of wavelet transforms from a computational point of view. One reason to use wavelet denoising carefully is the inherent danger to introduce artifacts, particularly when 20–30 parameters must be chosen by the operator. NMRLAB is an approach to introduce wavelet denoising strategies for routine use in NMR spectroscopy by providing a user-friendly interface for a wavelet toolbox and default parameters which will yield satisfactory results in most cases. The preservation of lineshapes and intensities during wavelet denoising enables the combination with resolution enhancement (Fig. 4). Wavelet transform filters also help to reduce noise in spectra recorded with low amounts of protein sample and few scans.

Table 4 lists the most important wavelet parameters. The major choices are the wavelet type and the noise reduction technique. We have found that symmlets, coiflets, and Daubechies wavelets are good choices for NMR noise reduction. The basis of noise reduction is the search for the largest "true" wavelet coefficients. A hard thresholding δ^{H} function,

$$\delta^{\mathrm{H}}(x) = \begin{cases} x, & \text{if } |x| > \lambda \\ 0, & \text{if } |x| < \lambda, \end{cases}$$

is a "keep or kill" selection where all coefficients below a level λ are zeroed. Alternatively Donoho and Johnstone (*27*) suggested a soft thresholding function,

$$\delta^{\mathrm{S}}(x) = \begin{cases} x - \lambda, & \text{if } x > \lambda \\ 0, & \text{if } |x| \le \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases},$$

where only the coefficients (in absolute value) greater than λ are included and their absolute values are reduced by an amount equal to the threshold. Both techniques yield good noise reduction for one- and two-dimensional NMR data. In either case it is a fundamental prerequisite that the data have been scaled with respect to their noise level. Donoho and Johnstone suggest two methods to select the threshold value, a *minmax* threshold which involves the calculation of a L^2 risk and *universal* thresholding where $\lambda = \sqrt{2 \log n}$ (n = number of data points). The latter method has become very common for global thresholding and provides good results for NMR data noise reduction.

A vast volume of literature has been devoted to more advanced techniques for nonuniversal threshold selection. The most popular scheme of this type is SURE (Stein's Unbiased Risk Estimate) thresholding which has also been introduced by Donoho and Johnstone (*28*), where a different threshold value is selected for each wavelet level. We found that SURE thresholding does not perform well for noisy NMR data where the majority of wavelet coefficients is essentially zero (i.e., the wavelet representation of the data is very sparse). This problem is usually addressed by a hybrid mechanism which defaults to the universal threshold $\sqrt{2 \log n}$ when the coefficient matrix is found to be sparse ("hybrid method") and otherwise uses SURE thresholding.

In addition to the choice of wavelet parameters, denoising can be accomplished by two-dimensional wavelet transforms on the two-dimensional data matrix or by applying a one-dimensional wavelet transform row by row to the entire spectrum. The two-dimensional approach is significantly faster but limited to quadratic matrices and prone to introduce artifacts such as small negative distortions next to the peaks in the spectrum. With one-dimensional wavelet transforms we found that soft thresholding yields lower residual noise than hard thresholding.

The availability of NMR processing in MATLAB should encourage spectroscopists to explore the world of advanced signal processing. The availability of sources is in our eyes crucial to produce reproducable results using sophisticated signal processing algorithms. For this reason NMRLAB is distributed as an open source system and is available by internet at http://www.bpc.uni-frankfurt.de/~nmrlab.

REFERENCES

1. F. Delaglio, S. Grzesiek, G. Vuister, G. Zhu, J. Pfeifer, and A. Bax, *J. Biomol. NMR* **6**(3), 277–293 (1995).
2. N. Delsuc, M. A. F., and G. Levy, *J. Magn. Reson.* **73,** 548–552 (1987).
3. J. Hoch, Rowland Institute for Science Technical Memorandum, **RIS-18t** (1985).
4. P. Güntert, V. Doetsch, G. Wider, and K. Wüthrich, *J. Biomol. NMR* **2,** 619–629 (1992).
5. J. Buckheit and D. Donoho, *in* "Wavelets and Statistics" (A. Antoniadis and G. Oppenheim, Eds.), pp. 53–81, Springer, Berlin (1995).

6. S. Mallat, A Wavelet Tour of Signal Processing. Academic Press, San Diego (1998).
7. F. Auger and P. Flandrin, IEEE International Symposium on Time-Frequency and Time-Scale Analysis, pp. 197–200 (1994).
8. F. Auger and P. Flandrin, *IEEE Trans. Sig. Proc.* **43**(5), 1068–1089 (1995).
9. D. Marion and K. Wüthrich, *Biochem. Biophys. Res. Commun.* **113**(3), 967–974 (1983).
10. D. States, R. Haberkorn, and D. Ruben, *J. Magn. Reson.* **48,** 286–292 (1982).
11. D. Marion, M. Ikura, R. Tschudin, and A. Bax, *J. Magn. Reson.* **85,** 393–399 (1989).
12. D. Muhandiram and L. Kay, *J. Magn. Reson. B* **103,** 203–216 (1994).
13. D. S. Wishart, C. G. Bigam, J. Yao, F. Abildgaard, H. J. Dyson, E. Oldfield, J. L. Markley, and B. D. Sykes, *J. Biomol. NMR* **6**(2), 135–140 (1995).
14. P. Güntert and K. Wüthrich, *J. Magn. Reson.* **96,** 403–407 (1992).
15. R. Kumaresan and D. Tufts, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-30,** 833–840 (1982).
16. G. Zhu and A. Bax, *J. Magn. Reson.* **100,** 202–207 (1992).
17. L. Jackson, Digital Filters and Signal Processing, 2nd ed., pp. 255–257, Kluwer Academic, Dordrecht (1989).
18. T. Parks and C. Burrus, Digital Filter Design, pp. 226–228, Wiley, New York (1987).
19. K. Steiglitz and L. McBride, *IEEE Trans. Automatic Contr.* **AC-10,** 461–464 (1965).
20. J. Cadzow, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-36,** 49–62 (1988).
21. A. Diop, A. Briguet, and D. Graveron-Demilly, *Magn. Reson. Med.* **27,** 318–328 (1992).
22. G. Zhu, W. Choy, G. Song, and B. Sanctuary, *J. Magn. Reson.* **132,** 176–178 (1998).
23. S. B. Shuker, P. J. Hajduk, R. P. Meadows, and S. W. Fesik, *Science,* **274,** 1531–1534 (1996). (Using Smart Source Parsing Nov 29).
24. B. A. Johnson and R. Blevins, *J. Biomol. NMR 4,* **4,** 603–614 (1994).
25. K. Kjær, M. Andersen, and F. Poulsen, *Methods Enzymol.* **239,** 288–308 (1994).
26. H. Serrai, L. Senhadji, J. DeCertaines, and J. Coatrieux, *J. Magn. Reson.* **124,** 20–34 (1997).
27. D. Donoho and I. Johnstone, *Biometrika,* **81,** 425–455 (1994).
28. D. Donoho and I. Johstone, *J. Am. Stat. Assoc.* **90,** 1200–1224 (1995).
29. P. Callaghan, A. MacKay, K. Pauls, O. Soderman, and M. Bloom, *J. Magn. Reson.* **56,** 101–109 (1984).
30. A. D. Marion, *J. Magn. Reson.* **84,** 425–430 (1989).