Advanced NMR Processing

Ulrich Günther

EuroLabCourse "Advanced Computing in NMR Spectroscopy", Florence, Sept. 2001

Contents

1 Introduction

NMR processing has developed over many years since the introduction of Fourier Transform (FT) -NMR spectroscopy by Richard Ernst. In FT-NMR spectroscopy the response to a perturbation from equilibrium is recorded during a certain amount of time. Because this response originates from the entire ensemble of spins it is an interferogram containing many different frequencies. The basis of all NMR processing is based on the fact that the interferogram can be described as a superposition of decaying complex exponentials, a free induction decay (FID). This signal is comparable to an acoustic signal , a high-frequency sound originating from nuclear spins. For this reason many of the more recent developments of acoustic signal processing are applicable to NMR signals.

Modern NMR spectroscopy benefits a lot from modern computational possibilities. The measured FID is digitized and stored as a digital signal, i.e. a series of complex data points.

NMR processing is used to convert this digital time-domain data into a frequency spectrum. This basic task is usually achieved using a discrete Fourier transform which provides a spectrum showing signal intensities versus frequency. Discrete Fourier transforms (DFT) were revolutionized in 1965 by the publication of a fast algorithm by Cooley and Tukey requiring $N \log(N)$ rather than N^2 operations.

Experimental NMR-signals contain noise besides the signals of interest. Most of this noise originates from the receiver circuitry. However, some of the noise is also a consequence of subtle instabilities during long measurements. The Fourier transform has the disadvantage that it convolutes noise with the spectrum. Several approaches to reduce the noise in spectra are commonly used in modern NMR processing. Most commonly the FID is multiplied with an apodization function which damps noise towards the end of the FID. Apodization usually broadens the original signal.

Many other steps are a part of regular NMR processing. Phase correction

is used to obtain pure absorption mode spectra. The solvent signal is often suppressed by removing on-resonance components from the FID. This can be achieved using different algorithms. The most common is the subtraction of a low-order polynomial which describes the slow variations in the FID. In protein NMR it is common to apply a convolution to the FID which also extracts low-frequency components.

Baseline correction is often applied to the final spectrum. This is particularly important for multi-dimensional NMR spectra which require a flat baseline for visualization and for peak picking. Baseline correction requires two distinct steps: differentiating between baseline and signal points and the calculation of a suitable signal to subtract from the spectrum.

Several more advanced signal processing techniques were applied to NMR spectroscopy. The most frequently used tool of this type is linear prediction, an algorithm introduced by Tufts and Kumaresan in 1982 [16]. Delsuc first introduced this tool for NMR processing [6]. Linear prediction is based on the fact that signals are periodic during the course of an FID while noise is not. If it is possible to determine coefficients which describe the intensity, decay and phase of the signal components in a FID the noise contribution can be eliminated and the course of the FID can be predicted beyond the duration during which the FID was recorded. This technique uses singular value decomposition (SVD) to calculate coefficients. Although it is computationally very demanding it is commonly used in modern NMR processing.

Many other techniques have also been used to process NMR signals. Maximum entropy reconstruction is the most well-known algorithm which is available in many NMR processing packages. MaxEnt uses a configurational entropy as a regularization function which provides a measure for the approximation of the calculated spectra to the experimental data. Bayesian analysis is a statistical method to estimate the degree to which a hypothesis is confirmed by experimental data. Bretthorst and Sibsi showed how Bayesian analysis can be used to process NMR spectra. More recently continuous wavelet transforms (CWT) were used to analyze NMR signals.

This course focuses on the use of discrete wavelet transforms (DWT) to reduce the noise level in NMR spectra and for suppression of the on-resonance solvent signal. Wavelet algorithms are being rapidly introduced in many fields of signal processing. The most common application of wavelets is probably image and sound compression (e.g. in jpeg, mpeg and mp3 file formats). Smoothing and noise suppression employing wavelet transforms was originally suggested by David Donoho [7, 8]. Other applications of wavelets include density estimation, and nonparametric regression. In the course basic principles of wavelet transforms will be presented.

All algorithms used in this course were implemented in NMRLab, a package for NMR processing in MATLAB (The Mathworks). Most of the source code of NMRLab is available. Although most of the routines were vectorized to maximize computational efficiency, a non-vectorized version is usually included as a comment. For wavelet transforms routines from WAVELAB were used [1].

Christian Ludwig made significant contributions to the software used in this course. The underlying work was supported by Prof. H. Rüterjans and the Large Scale Facility at Frankfurt.

2 Wavelets

Wavelets are a topic of applied mathematics. The mathematical theory of *ondelettes* (wavelets) was developed by Yves Meyer almost 15 years ago. The name *wavelet* originates from the the requirement for these functions to integrate to zero, “waving” above and below the *x*-axis. Wavelets chop up data into frequency components, and analyze each frequency component with a resolution matched to its scale.

General interest in wavelets has grown substantially in the past 10 years because wavelets solve basic problems in signal processing such as data approximation (smoothing), noise reduction, data compression, time-frequency analysis and image analysis. The availability of fast wavelet transform algorithms was crucial for their success in signal processing. The application of wavelets to smooth NMR signals has been inspired by David Donoho, one of the pioneers of the field, who used an NMR spectrum as an example to illustrate potential applications [7, 8]. In his book ’*NMR data processing*’ J. Hoch describes emerging methods in NMR data processing and shows an example for smoothing by wavelets following the ideas of Donoho [14]. More recent publications describe the use of wavelets in NMR processing [12, 11].

It is the aim of this course to introduce basic principles of wavelet analysis and potential applications to NMR researchers. For more advanced reading we refer to many excellent text books [17, 5, 13] and introductory texts [18, 21, 15, 9, 19].

2.1 The Haar System

There are many types of wavelets. The most simple wavelet is the Haar wavelet introduced in an appendix of the thesis of A. Haar in 1909. The Haar function is a simple step function a

$$\psi(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}[\\ -1, & x \in [\frac{1}{2,}1[\end{cases}$$

shown in Figure 2.1.

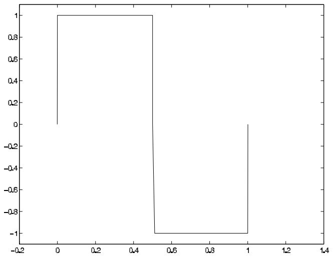


Figure 2.1: The Haar wavelet

The Haar function ψ is used to define a *mother wavelet*. From the mother wavelet a series of wavelets is derived by two dyadic operations: *dilatations* and *translations*. Dilatations compress the function on the x-axis. Translations slide it along the x-axis. For integer translation indices k and dilatation indices j wavelets are derived from the mother wavelet by

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k). \tag{2.1}$$

Haar wavelets $\psi_{j,k}$ have a support

$$\text{supp}\left(\psi_{j,k}\right) = \left[k2^{-j}, (k+1)\,2^{-j}\right[\tag{2.2}$$

i.e. they are zero outside this interval. For each Haar wavelet $\psi_{j,k}$ the integral

$$\int_{-\infty}^{\infty} \psi_{j,k}(x)dx = 0.$$

i.e. the area above the x-axis is equal to the area below the x-axis.

The set $\left\{\psi_{j,k}, j,k \in Z\right\}$ constitutes a complete orthonormal basis in L_2, the space of square integrable functions[1]. This means that any square integrable function can be approximated arbitrarily well by a linear combination of these basis functions.

[1] $L_2(\mathbb{R})$ is the space of complex valued functions f on $\mathbb{R}$ with a finite L_2-norm $||f||_2 = \left(\int_{\infty}^{\infty}|f(x)|^2 dx\right) < \infty$.

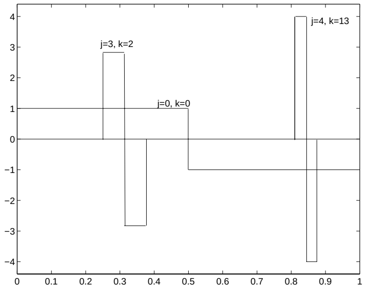
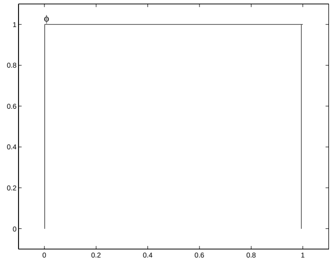


Figure 2.2: The set $\{\psi_{j,k}, j,k \in \mathbb{Z}\}$ derived from a Haar mother wavelet.

In addition to the set $\left\{\psi_{j,k}, j,k \in \mathbb{Z}\right\}$ we also need a *scaling function (father wavelet)* ϕ

$$\phi = \begin{cases} 1, & x \in [0,1[\\ 0, & x \notin [0,1[\end{cases}.$$



This is because construction of wavelets starts with the father wavelet from which the mother wavelet is derived. The reverse way is not possible. In addition the fast transform uses the father wavelet. With the scaling function ϕ we can expand our original set to $\{\phi_{j_0,k}, \psi_{j,k}, j \geq j_0, k \in \mathbb{Z}\}$. The combined set is again an orthonormal basis in L_2.

How can a data series be approximated by functions of the new set? This will be illustrated by a simple thought experiment. Any L_2-function can be approximated by a simple step function. The approximation converges for infinitely small steps. Now we must show that the same approximation can be achieved by a combination of the constant function ϕ and Haar wavelets.

Any square integrable function[2] $f(x)$ can be described by

$$f(x) = c_{00}\phi(x) + \sum_{j=0}^{n-1}\sum_{k=0}^{2^j-1} c_{j,k}\psi_{j,k}(x)\,, \tag{2.3}$$

where $c_{j,k}$ are wavelet coefficients, $\psi_{j,k}$ are wavelets derived from a mother wavelet ψ and ϕ is a scaling function (father wavelet), in the case of the Haar wavelet transform it is unity on the interval $[0,1[$. Using equation 2.4 a function f can be decomposed into a linear combination of wavelets $\psi_{j,k}$. The same is true for a data series which can be described by a function $f(x)$.

Wavelet transforms share many properties with Fourier transforms. The algorithm to determine wavelet coefficients is even faster than FFT. While the fast Fourier transform algorithm by Cooley and Tukey requires $N\log(N)$ operations, the dyadic wavelet transform gets along with only N operations.

Figure 2.3 compares the Fourier transform, the windowed Fourier transform (short-time FT) and the wavelet transform. In a time series with a high resolution in the time domain each point contains information about all frequencies. Due to the convolution properties the opposite is true for the FT of the time series. In this case every point in the frequency domain contains information from all points in the time domain. The windowed (short-time) Fourier transform divides the time-frequency plane in rectangular boxes. The resolution in time is increased at the expense of the frequency resolution. The dyadic wavelet transform (DWT) overcomes this problem by scaling the basis functions relative to their support. The WT needs more time for the detection of low frequencies than for the detection of high frequencies. Using these properties of the WT it is possible to describe an experimental signal on different frequency levels which leads to Mallats multiresolution analysis (MRA).

2.2 Mallat's Multiresolution Analysis

Mallat's multiresolution analysis provides a general framework to construct wavelet bases. The basic idea is to start from a father wavelet, derive an orthonormal mother wavelet and wavelet subspaces suitable to approximate functions with increasing resolution.

[2] For square integrable functions $||f|| = \left(\int_{\infty}^{\infty} f^2(x)dx\right)^{1/2} < \infty$ for $x \in \mathbb{R}$, i.e. $f \in L_2(\mathbb{R})$. $L_2(\mathbb{R})$ is a Banach space of square integrable functions.

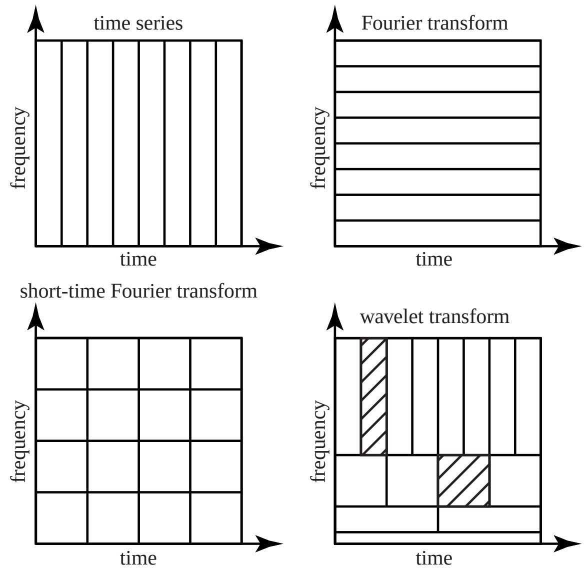


Figure 2.3: Comparison of time-frequency properties for a time series, its Fourier transform, short-time Fourier transform and wavelet transform.

1. We start with the scaling function (father wavelet)

$$\phi_{0,k} = \phi(x-k)$$

with

$$\phi(x) = \begin{cases} 1, & x \in [0,1[\\ 0, & x \notin [0,1[\end{cases}\,.$$

The set $\{\phi_{o,k}\}$ is an orthonormal basis for a reference space V_0. The function in V_0 have the form

$$f(x) = \sum_k c_k\phi(x-k)$$

which are constant functions on the interval $[k,k+1[$. Consequently we can write

$$V_0 = \{f(x) = \sum_k c_k\phi(x-k)\}.$$

2. Starting from V_0 we define linear spaces

$$V_1 = \{h(x) = f(2x) \ : \ f \in V_0\}$$

$$\vdots$$

$$V_j = \{h(x) = f(2^j x) \ : \ f \in V_0\}$$

V_1 contains all functions constant on $[\frac{k}{2}, \frac{k+1}{2}[$. The set $\{\phi_{1,k}\}$ is an orthonormal basis in $V_1$with $\phi_{1,k}(x) = \sqrt{2}\phi(2x-k)$.
Analogously, the basis functions of V_j are $\phi_{j,k} = 2^{j/2}\phi(2^j x - k)$.
ϕ generates a sequence of spaces $\{V_j,\ j \in \mathbb{Z}\}$ which are nested:

$$V_0 \subset V_1 \subset \ldots \subset V_j \subset \ldots$$

$$V_j \subset V_{j+1,}\ j \in \mathbb{Z}.$$

If in addition every square integrable function can be approximated by functions in

$$\bigcup_{j \geq 0} V_j$$

than $\{V_j,\ j \in \mathbb{Z}\}$ is a MRA[3].

[3] $\bigcup_{j\geq 0} V_j$ is dense in $L_2(\mathbb{R})$

3. Our set of constant functions is not a basis in L_2. To obtain a basis we must orthogonalize it. This is achieved if we find a W_0 for which

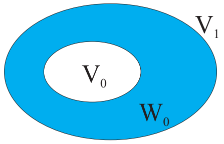


Figure 2.4: $V_1 = V_0 \bigoplus W_0$.

$$V_1 = V_0 \bigoplus W_0.$$

This means that W_0 is the orthogonal complement of V_0 in V_1. Figure 2.4 shows this relationship schematically. It requires that ϕ_{1k} must be a linear combination of ϕ_{0k} and ψ_{0k}:

$$\begin{aligned}\phi_{10}(x) = \sqrt{2}\phi(2x) \quad &= \sqrt{2}\{\phi_{00}(x) - \psi_{00}(x)\}/2 \\ &= \tfrac{1}{\sqrt{2}}\{\phi_{00}(x) - \psi_{00}(x)\}\end{aligned}$$

and similarly

$$\phi_{11}(x) = \sqrt{2}\phi(2x-1) \quad = \tfrac{1}{\sqrt{2}}\{\phi_{00}(x) + \psi_{00}(x)\}/2 \quad .$$

The system $\{\psi_{j,k}, k \in \mathbb{Z}\}$, where $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$, is an orthonormal basis in W_j.

4. The graphical representation of these functions is depicted in Figure 2.5.

The same process can be repeated for higher values of j. This leads to consecutive summation of subspaces

$$\begin{aligned} V_{j+1} &= V_j \oplus W_j \\ &= V_j \oplus W_{j-1} \oplus W_j \\ &= V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_j \\ &= V_0 \oplus \bigoplus_{l=0}^{j} W_l \end{aligned}$$

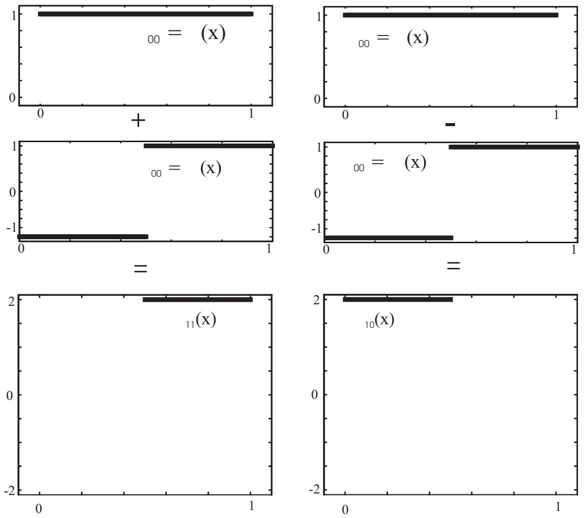


Figure 2.5: Basis functions $\phi_{j+1,k}$ can be written as linear combinations of basis functions $\psi_{0,k}$ and $\phi_{0,k}$.

Finally this sum of nested spaces spans :

$$L_2(\mathbb{R}) = V_0 \oplus \bigoplus_{l=0}^{j} W_l.$$

Consequently every square integrable function can be represented as the series

$$f(x) = \sum_k \alpha_{0,k}\phi_{0,k}(x) + \sum_{j=0}^{\infty}\sum_k \beta_{j,k}\psi_{j,k}(x).$$

The father wavelet ϕ creates a MRA. $\alpha_{0,k}$ and $\beta_{0,k}$ are coefficients for mother and father wavelets. This representation of f provides a location in time and frequency. The location in time is determined by k and the location in frequency by j. The larger j, the higher the frequency related to $\psi_{j,k}$.

It should be noted that the summation for $f(x)$ stops at 2^{j-1} for finite data sets:

$$f(x) = \alpha_{00}\phi(x) + \sum_{j=0}^{n-1}\sum_{k=0}^{2^j-1} c_{j,k}\psi_{j,k}(x), \tag{2.4}$$

Figure 2.6 depicts an example of a MRA for a synthetic NMR spectrum.

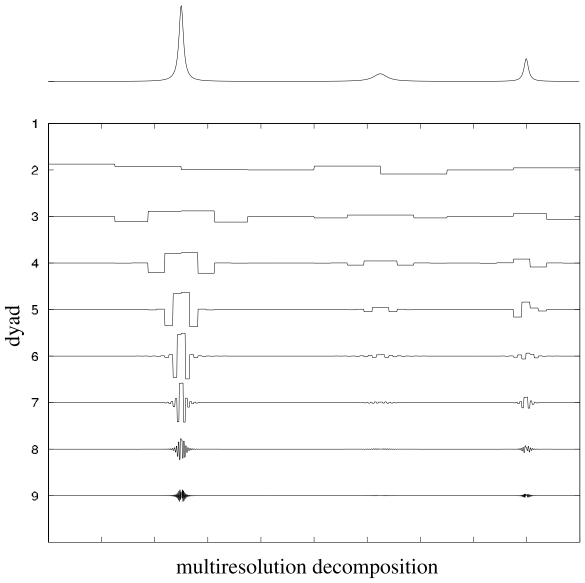


Figure 2.6: MRA of a synthetic spectrum.

It demonstrates that the approximation of the signal improves at higher values of j where higher frequencies are resolved.

2.3 Thresholding

Wavelet thresholding is the basis of wavelet based noise reduction. For a function f with Gaussian noise $y_i = f(t_i) + \sigma\varepsilon_i$ $(i \in \mathbb{N})$ this means that the function f is restored.

Two types of thresholding were proposed by Donoho and Johnstone [7]. Hard thresholding is a simple “keep or kill” selection. All wavelet coefficients below a threshold λ are zeroed.

$$c_{j,k}^{\text{hard}} = \begin{cases} 0, & |d_{j,k}| \le \lambda \\ c_{j,k}, & |c_{j,k}| > \lambda \end{cases}.$$

Soft thresholding shrinks the coefficients towards zero:

$$c_{j,k}^{\text{soft}} = \begin{cases} c_{j,k} - \lambda, & c_{j,k} > \lambda \\ 0, & |c_{j,k}| \leq \lambda \\ c_{j,k} + \lambda, & c_{j,k} < -\lambda \end{cases} .$$

The two shrinkage methods are displayed in Figure 2.7. The most important step is now a proper choice of the threshold λ. Donoho and Johnstone showed that a *universal threshold* $\lambda = \sigma\sqrt{2\log n}/\sqrt{n}$ where n is the sample size and σ the scale of the noise on a standard deviation scale[4].

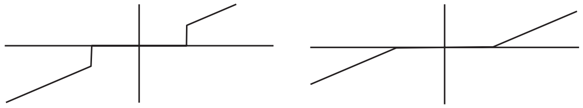


Figure 2.7: Hard and soft thresholding.

The overall procedure of noise suppression consists of a wavelet transform (WT) which yields the wavelet coefficients $c_{j,k}$, thresholding of these coefficients followed by an inverse wavelet transform (IWT) which restores the original spectrum.

2.4 Smoother Wavelet Bases

Although the Haar wavelet is convenient to describe the basics of wavelet transforms it is not suitable for most wavelet applications because too many coefficients are required to approximate a signal. The description of the design of smoother wavelet bases is far beyond the scope of this script. Wavelet bases must form an orthonormal basis for $L_2(\mathbb{R})$. Wavelets with good smoothing properties are designed to minimize the wavelet coefficients for smooth functions. The number of such constraints applied during the design of wavelets determines the number of vanishing moments. Wavelets of the Daubechies family shown in Figure 2.8 also have compact support which is important for noise reduction.

Daubechies wavelets have vanishing moments for mother but not for father wavelets and are fairly asymmetric. Coiflets have additional vanishing moments for father wavelets. Symmlets are as close to symmetry as possible.

Basic properties of wavelets with compact support:

[4] $\sigma = \frac{\text{median}\left(\left|c_{J-1,k} - \text{median}\left(c_{J-1,k}\right)\right|\right)}{0.6745}$

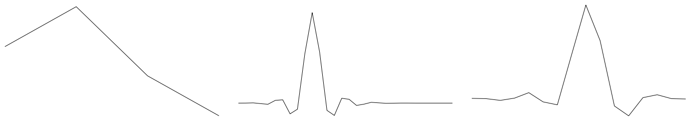


Figure 2.8: Types of wavelets from left to right: Daubechies (4), Coiflet (5) and Symmlet (8).

- Daubechies' wavelets:
  $supp\,\phi \subseteq [0, 2N-1]$
  $supp\,\psi \subseteq [-N+1, N]$
  $\int x^l \psi(x)dx = 0,\ l = 0, \ldots, N-1$
  For the D4 Daubechies wavelet: $\int \psi(x)dx = 0$, $\int x\psi(x)dx = 0$.
  Not symmetric.

- Coiflets
  $N = 2K$
  $supp\,\phi \subseteq [-2K, 4K-1]$
  $supp\,\psi \subseteq [-4K+1, 2K]$
  $\int x^l \phi(x)dx = 0,\ l = 0, \ldots, N-1$
  $\int x^l \psi(x)dx = 0,\ l = 0, \ldots, N-1$
  Coiflets are not symmetric.

- Symmlets
  $supp\,\phi \subseteq [0, 2N-1]$
  $supp\,\psi \subseteq [-N+1, N]$
  $\int x^l \psi(x)dx = 0,\ l = 0, \ldots, N-1$

- Symmlets are not symmetric.

2.5 Applications in NMR Spectroscopy

MRA and wavelet shrinkage have useful applications in NMR spectroscopy. MRA of the FID can be used to remove slow frequencies from the FID, i.e. to remove on-resonance components. Wavelet shrinkage has been used to denoise one- and multi-dimensional NMR-spectra.

2.5.1 WAVEWAT

Figure 2.9 shows a multiresolution decomposition of a FID. The high- and low-frequency components are nicely separated.

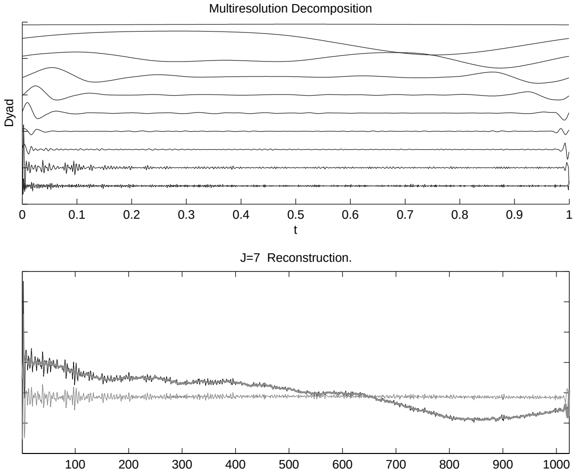
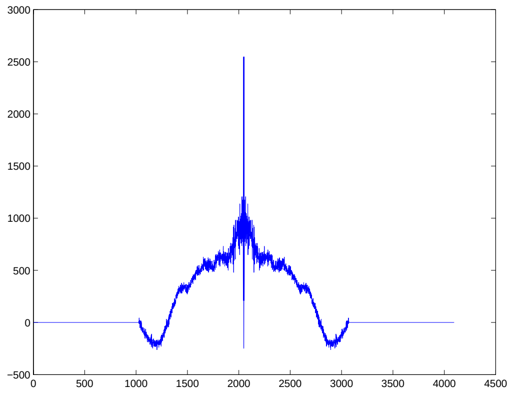


Figure 2.9: Top: Multiresolution plot of a FID from a ^{15}N-HSQC spectrum recorded at 500 MHz using a 1.2 mM protein sample. Bottom: Original FID and FID recovered from the MRA (gray) shown in Figure 1 using only using levels with $J \geq 7$.

Edge effects seen in Figure 2.9 are minimized when mirror reflections of the FID are used for the MRA. A basic limitation is the fact that the number of dyadic levels is limited by $2^J = N$ because the number of dyadic levels determines the width of the filter. A sufficiently large number of data points for a reasonably narrow filter width can be obtained by repeated zero filling.

An example of a DWT water suppression is illustrated in Figure 2.10. The signal originated from a ^{15}N-HSQC spectrum of a SH2 domain. The low intensity signal close to water was added synthetically to the experimental FID to demonstrate the effect of the filter. It can hardly be detected in Figure 2.10A underneath the strong water resonance which is not in phase with the rest of the spectrum. In Figure 2.10B the WAVEWAT-filtered spectrum (zero filling once, $ZF = 2$; reconstruction using levels ≥ 7, $J = 7$) is shown. Peak shapes and intensities are recovered perfectly. For comparison the effect of a convolution filter employing a 32 points Gaussian apodization function is shown (Figure 2.10C). This algorithm was originally proposed by Marion *et al*. [4]. Figure 2.11 demonstrates the principles of this filter.



Here the signal close to water is barely recovered and significantly distorted. Both the WAVEWAT and the convolution filter do not distort off-resonance peaks in the spectrum.

Noise levels were calculated for all columns of a two-dimensional HSQC spectrum after convolution water suppression and WAVEWAT water suppression. For convolution water suppression the noise level is reduced over an area of at least 80 points close to the water signal (Figure 2.12A). For WAVEWAT the area in which signals are distorted is much narrower and the edge of the filter is sharper. This is demonstrated in Figure 2.12B where a symmlet wavelet with 8 vanishing moments (Symmlet(8)) has been used after zero filling to 1024 points and $J = 7$. In Figure 4C the same wavelet is used after zero filling to 1024 points using $J = 9$. In Figure 4B the area in which signals are suppressed is narrower and the edges of the water signal becomes visible. In this case signals close to water will be recovered without distortion of signal intensities. Although approximately 20 points close to the water resonance were eliminated in Figure 2.12C, the spectral area which is affected by the filter is much smaller than that

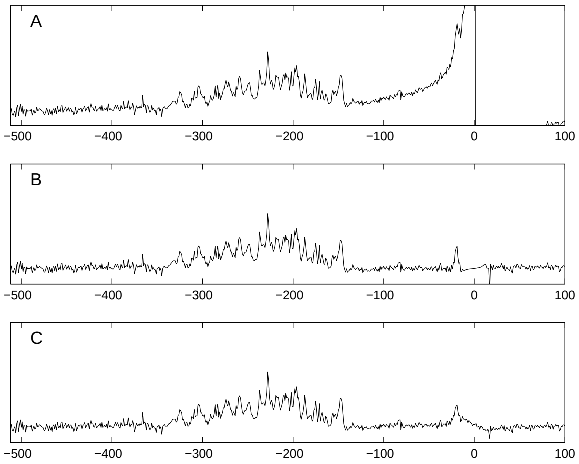


Figure 2.10: Spectrum obtained from the FID shown in Figure 2 after Fourier transformation. The small signal close to water was added synthetically to the experimental FID. A: Fourier transformed spectrum without prior water suppression. B: WAVEWAT was applied applied to the FID prior to Fourier transformation; the signal was zero-filled to 2048 points prior to MRA; the signal was recovered after rejecting 7 levels as shown in Figure 1; a symmlet (8) wavelet was used for the wavelet transform. C: water suppression was achieved by a convolution of the FID with a 32-point Gaussian window.

in the case of a convolution filter.

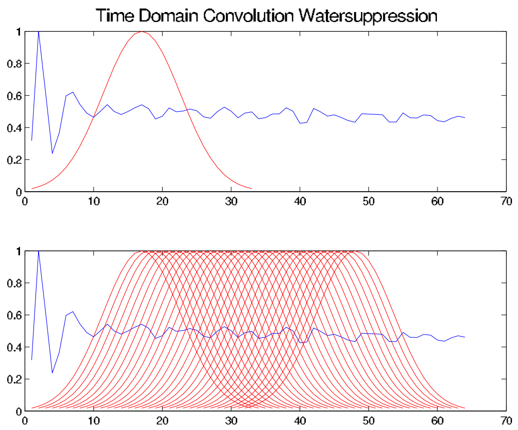


Figure 2.11: Convolution filter for noise suppression.

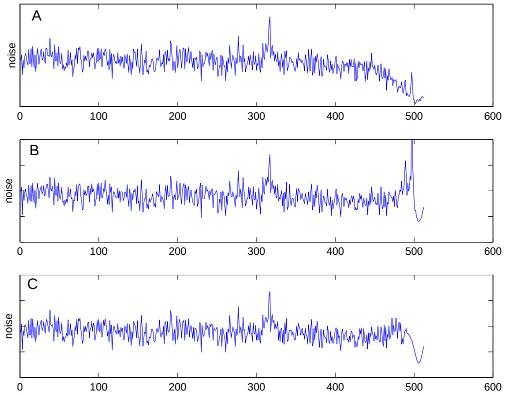


Figure 2.12: Noise levels calculated for the incremented dimension. A: after water suppression using time-domain convolution ; B: after water suppression using WAVEWAT after zero-filling to 1024 points using levels ≥ 7; C: after water suppression using WAVEWAT after zero-filling to 1024 points using levels ≥ 9.

2.5.2 Noise Suppression

The basis of noise suppression by wavelet shrinkage was described in chapter 2.3. It is achieved by performing a wavelet transform and applying a threshold to the wavelet coefficients. This is illustrated in Figure 2.13 which shows a one-dimensional spectrum before and after noise suppression together with the corresponding wavelet coefficients.

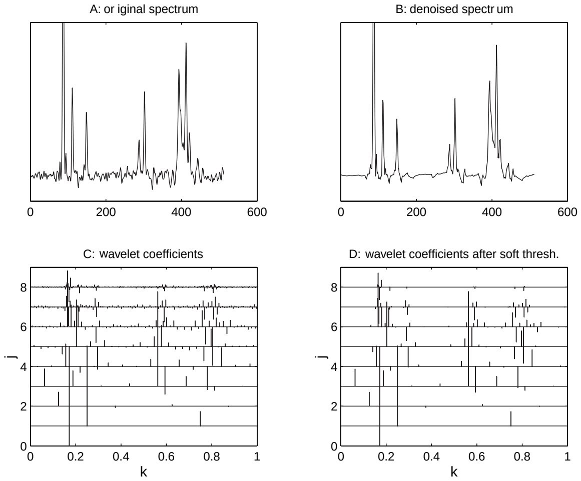


Figure 2.13: Wavelet shrinkage using a Daubechies (4) wavelet and soft thresholding.

Figure 2.14 shows an extreme case of an NMR spectrum with low signal-to-noise.

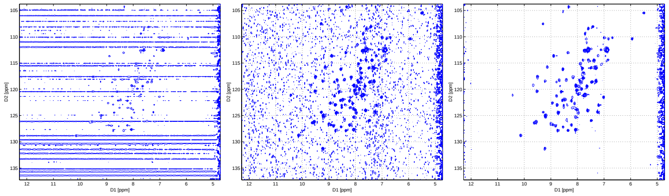


Figure 2.14: Left: Original HSQC spectrum. Middle: HSQC spectrum after baseline correction. Right: HSQC spectrum after wavelet shrinkage using a Symmlet (8) and hard-thresholding.

2.5.3 Noise suppression in three-dimensional spectra

Figure 2.15 shows the effect of wavelet shrinkage on two-dimensional slices from a three-dimensional ^{15}N-NOESY spectrum. Here a Daubechies (4) wavelet (with the smallest support among commonly used wavelets) gave the best results. Noise reduction had to be applied in both proton dimensions of the ^{15}N-NOESY-HSQC spectrum. The two spectra were plotted at comparable levels and peak picking was performed at the same levels for both spectra. In the case of the original spectrum 203 peaks were obtained whereas 101 peaks were obtained after noise reduction.

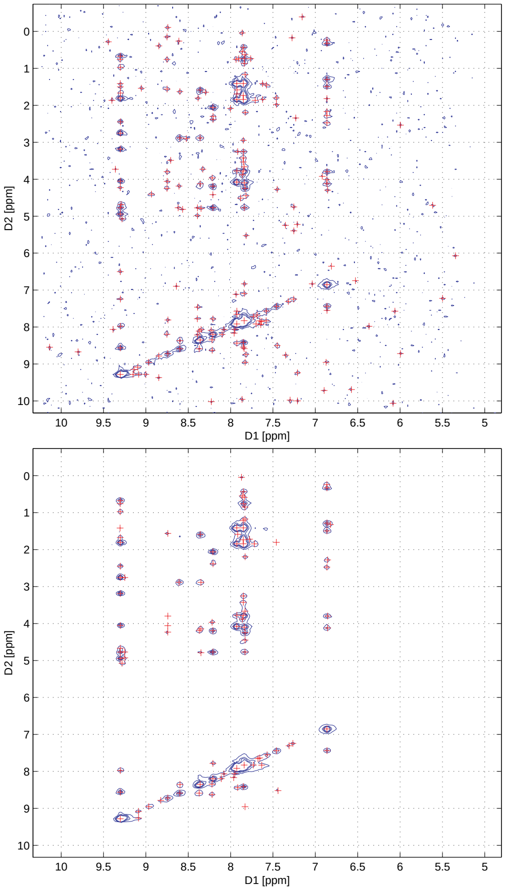


Figure 2.15: Two-dimensional slices from a three-dimensional ^{15}N-NOESY spectrum of sulfide dehydrogenase (SUD) spectrum before and after noise suppression. using a Daubechies (4) wavelet and soft-thresholding.

3 SVD-based methods

3.1 SVD-based noise and signal suppression

Singular value decomposition (SVD) has been used for many purposes in signal processing. SVD decomposes a matrix

$$\mathbf{H} = \mathbf{U_{LL}} \mathbf{S_{LM}} \mathbf{V}^{\dagger}_{\mathbf{MM}}.$$

U and **V** are $(L \times L)$ and $(M \times M)$ unitary matrices[1], **S** is $(L \times M)$ in size and contains the singular values. The singular values correspond to signal components in **H**. Large signal components belong to large signals in **H**, small signal components are often noise related. This feature is often used in signal processing because it allows one to distinguish between small and large signal components. Noise related singular values are often nicely separated from those related to signals. This principle was first used in signal processing by Cadzow [2]. In the following paragraphs a series of applications derived from this feature of the SVD will be presented.

For all applications a matrix **H** must be derived from the FID before anything can be started. For an FID

$$f = (s_0, s_1, \ldots, s_{N-1})$$

this is achieved by forming a Hankel type matrix by using a fraction of the FID which is shifted by one point in each row of the matrix

[1] A matrix **U** is unitary if $\mathbf{V}\mathbf{V}^{\dagger} = \mathbf{E}$. For real numbered matrices this is equivalent to orthogonal matrices.

$$\mathbf{H} = \begin{pmatrix} s_0 & s_1 & s_2 & \cdots & s_{M-1} \\ s_1 & s_2 & s_3 & \cdots & s_M \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{L-1} & s_L & s_{L+1} & \cdots & s_{N-1} \end{pmatrix}.$$

For a FID which does not contain any noise the rank of this matrix is equivalent to the number of signals in the FID. However, if the signal contains noise the rank will be full (M).

The SVD picks up periodicities in the FID. Periodic components in the FID will be represented by singular values. On the diagonal of **S** the singular values are sorted by size. This makes it very easy to select signal components. An example is presented in Figure 3.1. The noise test signal contains three distinct signals which are represented by three distinct singular values. In principle it is straightforward to use this procedure to suppress noise. If all singular values below the blue line are set to zero and the FID is restored from **H** a noise-free signal is obtained. The difficulty is always to find the noise level.

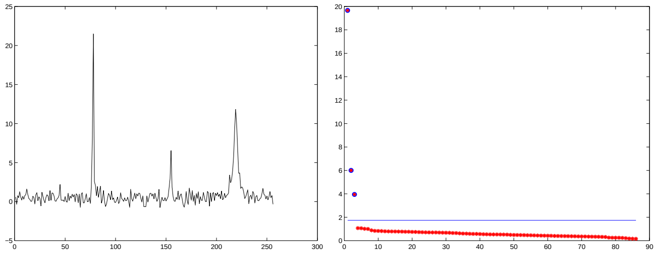


Abbildung 3.1: Left: NMR spectrum with three signals. Right: singular values sorted by size. Blue line: noise level.

A differential plot of the singular values (Figure 3.2 on the following page) shows that noise-related singular values have small positive differential numbers. These values are positive because the singular values were ordered by size and they are small because the difference between adjacent singular values is small. This differential plot can help to defined the noise level without visual inspection. If the number of data points is large compared to the number of signals in the spectrum the mean value of the last singular values $\overline{\Delta S}$ can be used to calculate the noise level by extrapolating the line back to the first singular value: $S_{thresh} = N * \overline{\Delta S}$. For the shown example the value of $\overline{\Delta S}$ was 0.0160, the noise level (blue

line) was calculated as $N = 2/3 * 128 \simeq 86 \approx 1.37$.

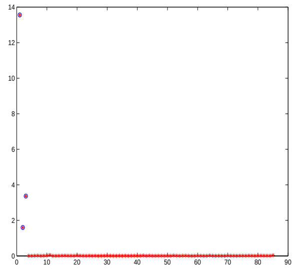


Abbildung 3.2: Differential plot of singular values.

It is important to note that the number of singular values is twice the number of signal components for real signals. If the complex points are reconstructed employing a Hilbert transform[2] this introduces erroneous signal components in the singular values shown in Figure 3.3. These errors are typically in the order of $3 \times S_{\text{thres.}}$.

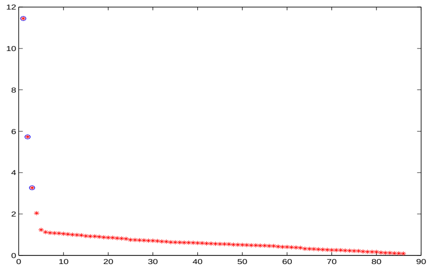


Abbildung 3.3: Singular values calculated for the spectrum from Figure 3.1 after Hilbert transform and inverse Fourier transform.

The SVD applied to a Hankel matrix derived form a FID combined with the described selection of a noise level can be used to determine the number of signal components in a FID. This is illustrated in Figure 3.4 which shows 5 spectra with complex line shapes and varying signal intensity.

Figure 3.5 depicts the corresponding signal component analysis using automatic noise level detection. Singular values above the noise level are marked

[2] The Hilbert transform must be applied in the frequency domain. This requires a Fourier transform prior to the Hilbert transform and an inverse Fourier transform to obtain the complex FID.

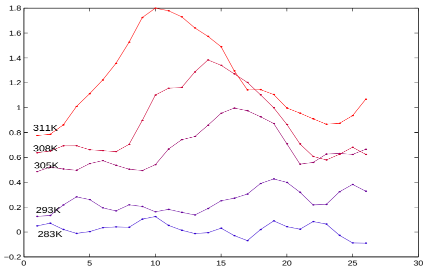


Abbildung 3.4: Test signals for signal component analysis.

by an extra circle. The results are in good agreement with the number of signal components seen by visual inspection.

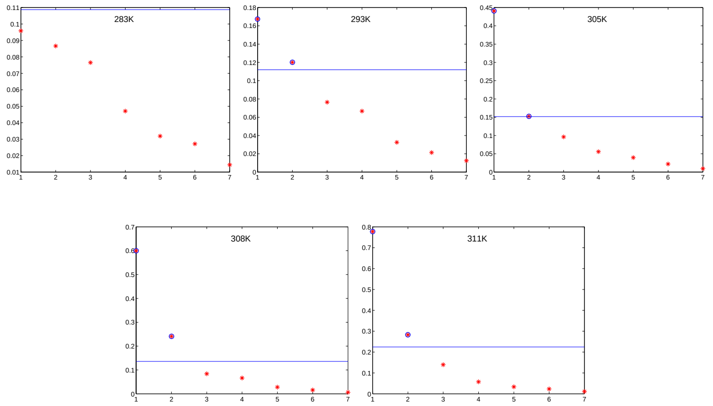


Abbildung 3.5: Singular values for the 5 spectra shown in Figure 3.4.

The same principle was used by Cadzow to suppress the noise level in signals [2]. The principle is shown in Figure 3.6a. By dropping the noise related singular values an almost noise-free spectrum is obtained.

The procedure can also be used in the other direction to suppress signals. This is demonstrated in Figure 3.6b where the largest signal component was zeroed. This eliminated one signal in the corresponding spectrum. For equally sized signals the choice of the peak which will be eliminated will be random. However, if the largest signal is to be eliminated the method is very stable. This algorithm has been used in NMR processing to eliminate large diagonal peaks

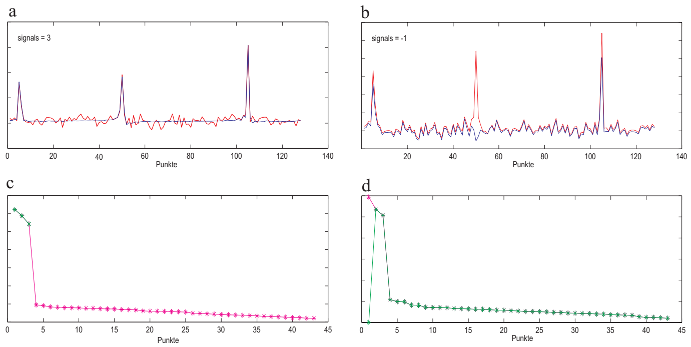


Abbildung 3.6: Cadzow noise suppression and signal suppression.

in NOESY spectra [22]. It is almost routinely used to suppress the water signal in *in vivo* MRS [20, 23]. Unfortunately this algorithm can not be used for high-resolution NMR-spectra and for large multi-dimensional spectra because the SVD is a $O((L-M)^2M)$ process.

3.2 Linear prediction

Linear prediction has in principle nothing to do with SVD. However, SVD is frequently used to determine LP coefficients. This version of linear prediction is called LP-SVD. The LP-SVD algorithm was originally described by Kumaresan and Tufts [16] and modified by Porat and Friedlander.

The forward linear prediction model assumes that a data point can be described by a linear combination of K preceding points

$$x_n = \sum_{k=1}^{K} a_k x_{n-k}, \tag{3.1}$$

the backward linear prediction model by a linear combination of the K following points:

$$x_n = \sum_{k=1}^{K} b_k x_{n+k}. \tag{3.2}$$

Equation 3.1 can be written in matrix form as

$$x = a \cdot \mathbf{Y}$$

from which the matrix coefficients can be obtained by an inversion of **Y**. Inversion of **Y** can be achieved by SVD

$$\mathbf{Y} = \mathbf{U}\Lambda\mathbf{V}^{\dagger}$$

from which **Y**′ is obtained by

$$\mathbf{Y}^{\dagger} = \mathbf{V}\Lambda^{-1}\mathbf{U}^{\dagger}.$$

Now a can be computed by simple matrix algebra

$$a = \mathbf{V}\Lambda^{-1}\mathbf{U}^{\dagger}x.$$

In the presence of noise K must be larger than the number of frequency components; it is typically set to $N/4 - N/3$.

Using these prediction coefficients a_k the frequencies and damping factors are obtained by calculating the roots of the polynomial

$$z^K - a_1 z^{K-1} - \ldots - c_K = 0.$$

The signal x_n is described by

$$x_n = \sum_{k=1}^{K} A_k z_k^{n-1}. \tag{3.3}$$

Now the A_k which encode phase and amplitude of each signal component can be determined by an additional SVD. Equation 3.3 can be rewritten in matrix form

$$x = A\mathbf{Z}.$$

A can be determined by inverting **Z** which can be achieved employing a second SVD.

These procedures are repeated for backward linear prediction. Forward and backward linear prediction yield two sets of polynomial roots. For forward linear prediction the complex roots lie inside the unit circle. For the backward linear prediction they are inside the unit circle. For the forward linear prediction the roots inside the circle are related to signal, those outside to exponentially increasing signals or to noise. Several procedures were proposed to minimize linear prediction artefacts. Root reflection replaces roots z_k by $z_k/|z_k^2|$. This ensures that all signals are within the unit circle and noise reacted signals are

eliminated. If forward and backward linear prediction coefficients were calculated it is common to use average of the reflected roots for forward and backward linear prediction.

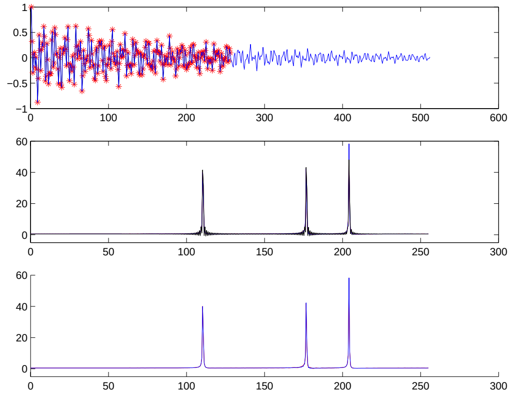


Figure 3.7: LPSVD. Top: Simulated FID (256 points, *) and predicted FID (512 point, solid line). Middle: Spectrum calculated after zero filling. Bottom: Spectrum after lpsvd.

It is important to note that the SVD steps used to calculate the linear prediction coefficients a_k (and b_k) can be used to eliminate noise related signals by truncating Λ to include solely large singular values which correspond to signals rather than noise.

Linear prediction is used to predict additional data points to improve resolution. This can be helpful to avoid truncation artefacts. It can also be applied to enhance signal-to-noise.

4 Using NMRLab

The algorithms described in this course were implemented in NMRLab [10]. NMRLab uses MATLAB (The Mathworks). Here some of the most basic tools and data structures in NMRLab are described.

NMRPAR is initialized as a global structure by `nmrlab.m` and holds all basic parameters required to run NMRLAB. The fields of NMRPAR define availability of RAM, computer type (detected by `nmrlab.m`) and other parameters.

The NMRDAT structure

NMRDAT is a global structure which is another set up by the `nmrlab.m` script usually executed from `STARTUP`. Some fields can be edited, saved and restored using `BROWSE` but NMRDAT it is also manually readable on the MATLAB command line. Table 4.1 lists the field names in NMRDAT.

Table 4.1: NMRDAT structure

`NMRDAT`	**field name**	
	`NAME`	Dataset name. Used for saving substructure to disk.
	`SER`	Converted SER file.
	`MAT`	Processed data matrix.
	`ACQUS`	Acquisition parameters (3D).
	`PROC`	Processing parameters (3D).
	`DISP`	Display parameters.

`NAME` is a string describing the name of the `.mat` file when data is saved to disk with the browse command. This is the only use of `NAME`. It has no influence on the name of the data after it is retrieved because a data set will always become an element in the `NMRDAT` structure.

`SER` is simply an array for FIDs in the order they were recorded. The fast (t2) dimension is in columns. `MAT` is the processed data matrix with dimension 1 in rows. This is not consequent but it saves time when data is displayed because the contour command logically plots matrix rows as rows, the way we usually look

at two-dimensional NMR data. `MAT` is set to ’-1’ when a matrix is saved to disk and deleted from RAM.

`ACQUS` contains the acquisition parameters which were read from the BRUKER acqus, acqu2s and acqu3s files with the readacqus command. It is itself a two or three dimensional array which contains the information for the three spectral dimensions.

`PROC` has a similar structure and holds the processing parameters (set in `EDP`). It is also a three-dimensional array of structures.

NMR processing functions in NMRLAB.	
Function	**Description**
`hft`	Hilbert Transform
`fft`*	Fast Fourier transform
`ift`*	Inverse fast Fourier transform
`dft`	Fourier transform of BRUKER digital filtered data
`rft`	Real Fourier transform for TPPI-type data
`smo`	smooth = polynomial sovent filter [3]
`sol`	solvent filter by time-domain convolution [4]
`wavewat`	WAVEWAT water suppression
`wdwf2`	Window functions (gm, em, sine bell, cubic sine bell)
`baseline2`	Different algorithms for baseline correction
`chi2_flatt`	Calculate chi2 vector for `FLATT`
`absc`	call and setup for `FLATT` baseline flattening (calls `flatten2`)
`lpsvd2`	SVD based Linear Prediction [24, 16]
`lpx2`	Linear Prediction (LPC, Prony, or Steiglitz-McBride)**
`rev`	reverse data
`cshl`	circular shift left
`cshr`	circular shift right
`shl`	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 (uses `CADZOW`)
*MATLAB built-in function. **Requires MATLAB signal processing tool box.	

Table 4.2: All processing commands can be executed from the command line. However, usually processing routines are called from the processing functions `xfb` and `tf`. Help for all commands is available by typing `help('command')`.

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
`snd`	show data sets and sizes NMRDAT
`uiphase`	interactive phase correction
`uicont`	interactive contour plotting*
`sartitr`	analyze series of 2D NMR spectra (e.g. SAR by NMR series)
`edp`	edit proocessing 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 post processing baseline correction
`phasend`	2D/3D post processing phase correction
`denoise`	2D/3D post processing wavelet denoising
`xyztranspose`	transpose 3D structures
`makespc`	utility to create synthetic spectra
*MATLAB $\geq$ 5.3 will automatically activate graphical tools and zoom options.	
For MATLAB 5.0-5.2 use the zoom and plotedit commands instead.	

Table 4.3:

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 < < J, $N = 2^J$ (*2-4*) N = number of data points
normalize	normalize noise *	
WT_type**		*periodized orthogonal* fully translation invariant
thr	threshold value	manual entry, *universal* $\sqrt{\log(n)}$, n = number of data points.

*2D and 3D version of normalization has been implemented in NMRLAB.

**Other wavelet transforms (i.e. the Meyer wt) are available in WaveLab .

Parameters which yield good results for NMR spectroscopy have been italicized.

Table 4.4:

Type of LP	**algorithm**	**parameter vector**
`lpsvd`	SVD based linear prediction	[FID, start,stop,poles,signals]
`prony`	PRONY's method	[FID,NB,NA]
`stmbc`	Steiglitz McBride	[FID,NB,NA,N]
`lpc`	LPC	[FID,N]

`Lpsvd` does not give a choice regarding forward vs. forward-backward linear prediction because forward linear prediction is prone to pick up faulty noise peaks and should therefore never be used without user interaction.

Bibliography

[1] J.B. Buckheit and D.L. Donoho. Wavelab and reproducible research. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 53–81. Springer, Berlin, 1995.

[2] J. Cadzow. Signal enhancement. A composite property mapping algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-36:49–62, 1988.

[3] P.T. Callaghan, A.L. MacKay, K.P. Pauls, O. Soderman, and M. Bloom. *J. Magn. Reson.*, 56:101–109, 1984.

[4] A. Bax M. Ikura D. Marion. Improved solvent suppression in one- and two-dimensional NMR spectra by convolution of time-domain data. *J. Magn. Reson.*, 84:425–430, 1989.

[5] I Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[6] MArc A. Delsuc, Feng Ni, and George C. Levy. Improvement of linear prediction of nmr spectra having very low signal-to-noise. *J. Magn. reson.*, 73:548–552, 1987.

[7] D.L. Donoho and I.M. Johnstone. Ideal spacial adaptaion via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[8] D.L. Donoho and I.M. Johstone. Adapting to unknown smoothness via wavelet shrinkage. *J. of Amer. Stat. Assoc.*, 90:1200–1224, 1995.

[9] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engeneering*, 2:1–18, 1995.

[10] UL Günther, C Ludwig, and H Rüterjans. NMRLAB - Advanced NMR data processing in Matlab. *J. Magn. Reson.*, 145(2):101–108, 2000.

[11] UL Günther, C Ludwig, and H Rüterjans. Improved automatic structure calculation usind wavelet denoised data. *in preparation*, 2001.

[12] UL Günther, C Ludwig, and H Rüterjans. WAVEWAT - Improved solvent suppression in NMR spectra employing wavelet transforms. *submitted to J. Magn. Reson.*, 2001.

[13] W. Härdle, G. Kerkyacharian, D. Picard, and A. Tsybakov. *Wavelets, Approximation, and Statistical Applications*. Springer, 1998.

[14] Jeffrey C. Hoch and Alan S. Stern. *NMR data processing*. 1996.

[15] B Jawerth and W Sweldens. An overview of wavelet based multiresolution analysis. *SIAM Review*, 36:377–412, 1994.

[16] R. Kumaresan and D.W. Tufts. Estimating the parametes of exponentially damped sinosoids and pole-zero modelling in noise. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-30:833–840, 1982.

[17] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.

[18] Yves Nievergelt. *Wavelets Made Easy*. Birkhäuser, 1999.

[19] Todd R Odgen. *Essential wavelets for statistical applications and data analysis*. Birkhäuser, 1997.

[20] WWF Pijnappel, A van den Boogaart, R de Beer, and D van Ormondt. SVD-based quantification of magnetic resonance signals. *J. Magn. Reson.*, 97:122–134, 1992.

[21] Brani Vidakovic and Peter Müller. Wavelets for kids. Available at http://www.isye.gatech.edu/ brani/wavelet.html, 1994.

[22] G. Zhu, W.Y. Choy, G. Song, and B.C. Sanctuary. Suppression of diagonal peaks with singular value decomposition. *J. Magn. Reson.*, 132:176–178, 1998.

[23] G Zhu, D Smith, and Y Hua. Post-acquisition solvent suppression by singular-value decomposition. *J. Magn. Reson.*, 124(1):286–9, 1997.

[24] Guang Zhu and Ad Bax. Improved linear prediction of damped nmr signals using modified "forward-backward" linear prediction. *J. Magn. Reson.*, 100:202–207, 1992.