Parametric Estimation of NMR Signals in the Time Domain

SIMON HULSE

Balliol College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity Term, 2023

Supervisors:

Mohammadali Foroozandeh, Timothy Claridge, Fay Probert

Dedication goes here

Acknowldegements

I'd like to thank everyone

Acronyms

1D one-dimensional

2D two-dimensional

2DJ 2D J-resolved

ALPESTRE A Linear Predictive Estimation of Signal Time Reversal

AWGN Additive White Gaussian Noise

BIRD BIlinear Rotation Decoupling

COSY Correlation Spectroscopy

CPU Central Processing Unit

CUPID Computer-assisted Undiminished-sensitivity Protocol for Ideal Decoupling

CTP coherence transfer pathway

EYM Eckart-Young-Mirsky

FID Free Induction Decay

FFT Fast Fourier transform

FT Fourier transform

GN Gauss-Newton

HSQC Heteronuclear Single Quantum Coherence Spectroscopy

IFT Inverse Fourier transform

LM Levenberg-Marquardt

LP Linear Prediction

MDL Minimum Description Length

MEMPM Matrix Enhancement and Matrix Pencil Method

MMEMPM Modified Matrix Enhancement and Matrix Pencil Method

MPM Matrix Pencil Method

NLP non-linear programming

NMR Nuclear Magnetic Resonance Spectroscopy

NMR-EsPy NMR Estimation in Python

NOESY Nuclear Overhauser Effect Spectroscopy

pdf probability density function

PFG pulsed field gradient

PGSE Pulsed Gradient Spin Echo

PGSTE Pulsed Gradient STimulated Echo

PGSTEBP Pulsed Gradient STimulated Echo with BiPolar gradients

PSYCHE Pure Shift Yeilded by CHirp Excitation

RAM Random Access Memory

RF radio-frequency

SNR signal-to-noise ratio

ST Steihaug-Toint

SVD Singular Value Decomposition

TROSY Transverse Relaxation Optimised Spectroscopy

VE virtual echo

ZS Zangger-Sterk

NOMENCLATURE

General Mathematics

$\mathcal{A} := \mathcal{B}$ $\mathcal{A}$ is defined to be equal to $\mathcal{B}$

$\lfloor a \rfloor$ The nearest integer to a, such that $\lfloor a \rfloor \leq a$

$\lceil a \rceil$ The nearest integer to a, such that $\lceil a \rceil \geq a$

$\lfloor a \rceil$ The nearest integer to a

$a \bmod b$ a modulo b, given by

$$a \bmod b = a - \left\lfloor \frac{a}{b} \right\rfloor$$

Complex Numbers

i The imaginary unit, $\mathrm{i} := \sqrt{-1}$

$\Re(z)$ The real component of z

$\Im(z)$ The imaginary component of z

$|z|$ The absolute value of z, given by

$$|z| = \sqrt{\Re(z)^2 + \Im(z)^2}$$

Number sets

$\mathbb{C}$ The set of complex numbers

$\mathbb{N}$ The set of natural numbers with zero excluded

$\mathbb{N}_0$ The set of natural numbers with zero included

$\mathbb{R}$ The set of real numbers

$\mathbb{R}_{>0}$ The set of positive real numbers

$\mathbb{Z}$ The set of integers

Probability

$x \sim \mathcal{X}$ x behaves according to distribution $\mathcal{X}$

$A \perp\!\!\!\perp B$ A and B are conditionally independent

$\mathcal{N}\left(\mu, \sigma^2\right)$ Normal (Gaussian) distribution with mean μ and variance σ^2:

$$\mathcal{N}\left(x \mid \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$\mathcal{N}_C\left(\mu, \sigma^2\right)$ Complex normal distribution with mean μ and variance $\sigma^2/2$

Vectors, Matrices, and Arrays

$.^{\mathrm{T}}$ Transpose

$.^{\dagger}$ Conjugate transpose, equivalent to $\cdot^{*\mathrm{T}}$

$.\circlearrowright(d)$ Right circular rotation by along axis d of the array by one element. For example, for a 2D matrix $\boldsymbol{A} \in \mathbb{F}^{N^{(1)} \times N^{(2)}}$, $\boldsymbol{A}^{\circlearrowright(1)}$ is given by

$$\begin{bmatrix} \boldsymbol{A}\left[N^{(1)}-1, 0\right] & \boldsymbol{A}\left[N^{(1)}-1, 1\right] & \cdots & \boldsymbol{A}\left[N^{(1)}-1, N^{(2)}-1\right] \\ \boldsymbol{A}\left[0, 0\right] & \boldsymbol{A}\left[0, 1\right] & \cdots & \boldsymbol{A}\left[0, N^{(2)}-1\right] \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}\left[N^{(1)}-2, 0\right] & \boldsymbol{A}\left[N^{(1)}-2, 1\right] & \cdots & \boldsymbol{A}\left[N^{(1)}-2, N^{(2)}-1\right] \end{bmatrix}$$

$\boldsymbol{A}^{+}$ The Moore-Penrose pseudo inverse of $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is a matrix $\in \mathbb{C}^{n \times m}$ given by

$$\boldsymbol{A}^{+} = \left(\boldsymbol{A}^{\dagger}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\dagger}.$$

$\mathrm{diag}(\boldsymbol{v})$ Given a vector $\boldsymbol{v} \in \mathbb{F}^N$, this creates a matrix $\in \mathbb{F}^{N \times N}$ of the form

$$\mathrm{diag}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{v}[0] & 0 & \cdots & 0 \\ 0 & \boldsymbol{v}[1] & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{v}[N-1] \end{bmatrix}$$

$\langle \boldsymbol{A}, \boldsymbol{B} \rangle$ The inner product of $\boldsymbol{A}$ and $\boldsymbol{B}$, two D-dimensional arrays over the field $\mathbb{C}$ with the same dimensions, given by

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{n^{(1)}=0}^{N^{(1)}-1} \cdots \sum_{n^{(D)}=0}^{N^{(D)}-1} \left(\boldsymbol{A}\left[n^{(1)}, \cdots, n^{(D)}\right]^{*} \boldsymbol{B}\left[n^{(1)}, \cdots, n^{(D)}\right]\right)$$

$\|\boldsymbol{A}\|$ The norm of $\boldsymbol{A}$, given by

$$\|\boldsymbol{A}\| = \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle}$$

$\nabla f(\boldsymbol{x})$ The gradient of a differentiable, scalar value function $f : \mathbb{R}^n \to \mathbb{R}$, given by

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f}{\partial \boldsymbol{x}[0]} & \frac{\partial f}{\partial \boldsymbol{x}[1]} & \cdots & \frac{\partial f}{\partial \boldsymbol{x}[n-1]} \end{bmatrix}^{\mathrm{T}}$$

$\nabla^2 f(\boldsymbol{x})$ The Hessian of a twice-differentiable, scalar value function $f : \mathbb{R}^n \to \mathbb{R}$, given by

$$\nabla^2 f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial \boldsymbol{x}[0]^2} & \frac{\partial^2 f}{\partial \boldsymbol{x}[0] \partial \boldsymbol{x}[1]} & \cdots & \frac{\partial^2 f}{\partial \boldsymbol{x}[0] \partial \boldsymbol{x}[n-1]} \\ \frac{\partial^2 f}{\partial \boldsymbol{x}[1] \partial \boldsymbol{x}[0]} & \frac{\partial^2 f}{\partial \boldsymbol{x}[1]^2} & \cdots & \frac{\partial^2 f}{\partial \boldsymbol{x}[1] \partial \boldsymbol{x}[n-1]} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \boldsymbol{x}[n-1] \partial \boldsymbol{x}[0]} & \frac{\partial^2 f}{\partial \boldsymbol{x}[n-1] \partial \boldsymbol{x}[1]} & \cdots & \frac{\partial^2 f}{\partial \boldsymbol{x}[n-1]^2} \end{bmatrix}$$

*   Broadcasting notation, as is commonly performed in numerical computations. See this page of the NumPy documentation for details

Other symbols

$f_{\mathrm{off}}^{(d)}$ Transmitter offset in dimension d

$f_{\mathrm{sw}}^{(d)}$ Sweep width (spectral window) in dimension d

$\Delta_t^{(d)}$ Sampling rate in dimension d

Table of Contents

LIST OF FIGURES

List of Tables

List of Algorithms

INTRODUCTION

<div style="text-align: right">**1**</div>

The story so far: In the beginning the Universe was created. This has made a lot of people very angry and been widely regarded as a bad move.

– Douglas Adams, The Restaurant at the End of the Universe

Since its conception almost 80 years ago, Nuclear Magnetic Resonance Spectroscopy (NMR) has become a ubiquitous technique in chemistry, biochemistry and numerous other disciplines, thanks to the unique insights into structure and dynamics it can provide. In this chapter, after a very brief historical overview of the subject, an outline of the basic theory behind the experiment – without delving into the necessary quantum mechanics for a truly formal account – is given. Finally, details of the means by which the raw experimental data, the Free Induction Decay (FID) is produced, and its structure, are given. Extracting information from the FID is the central motivation of this thesis.

1.1 Nuclear Magnetic Resonance Spectroscopy

1.1.1 A Brief History of NMR

The origins of NMR can be traced back to 1945, when independent work by Felix Bloch on water[1] and Edward Purcell on parrafin[2] gave rise to the first illustrations of nuclear magnetic resonances in condensed phases. The two hadn't met before their respective papers were published with about a month's separation[3]. Both received the Nobel Prize in Physics in 1952 for their pioneering work in the field. A notable mention should also be given to Yevgeny Zavoisky – the father of Electron Paramagnetic Resonance – who probably observed NMR as far back as 1941[4]. Alas, he dismissed his results as irreproducible. In 1949 and 1950, work investigating NMR spectra from compounds containing Cu, ^{31}P, ^{14}N, and ^{19}F nuclei illustrated the concept of the chemical shift[5, 6, 7], in which nuclei in different chemical environments exhibit non-identical resonant frequencies. Chemists regarded these results with great interest, as these findings suggested that NMR could give insights into molecular structure.

Russel Varian secured the first patent for a commercial NMR machine, and a 30 MHz spectrometer followed soon after. The first spectrometers were continuous wave, which functioned by slowly sweeping the magnetic field, causing spins to come into resonance at different times. Richard Ernst and Weston Anderson, working at Varian Inc. at the time, proposed an alternative method: pulsed Fourier transform (FT) spectroscopy[8]. This was not seen as a fruitful endeavour by the company, largely because of the very long time it took to digitise the signal, and subsequently compute its FT[9]. Instead, the first commercial pulsed FT spectrometer was produced by Bruker Corp. in 1969, which revolutionised NMR. The emergence of the Cooley-Tukey's Fast Fourier transform (FFT) algorithm[10] led to vast improvements in the speed with which experiments could be conducted, which incentivised the development of the new FT approach.

The idea of 2D NMR spectroscopy was proposed by Jean Jeener in 1971[11, 12], which Ernst and co-workers showcased a few years later in the form of a Correlation Spectroscopy (COSY) experiment[13]. The use of multiple dimensions to spread out signals enabled vastly more complex structures to be studies. In 1985, a report of the first protein assigned by NMR (using COSY and Nuclear Overhauser Effect Spectroscopy (NOESY)) was presented by Kurt Wütrich and co-workers[14]. Over time, extensive developments in techniques for biomolecular systems have occurred, including the creation of 3D and 4D experiments[15, 16], as well Transverse Relaxation Optimised Spectroscopy (TROSY) experiments[17] for the study of large proteins.

NMR's significance as an analytical tool is evidenced by Nobel Prizes in Chemistry being awarded for work in the field on two separate occasions. First, Ernst received the prize in 1991 "for his contributions to the development of the methodology of high resolution Nuclear Magnetic Resonance Spectroscopy"[18]. In 2002, Wüthrich was recognised "for his development of Nuclear Magnetic Resonance Spectroscopy for determining the three-dimensional structure of biological macromolecules in solution"[19].

1.1.2   Nuclear Spin and Magnetism

NMR relies on *spin*, an intrinsic property of certain nuclei (along with other elementary particles) which, along with orbital angular momentum, is one of the two sources of angular momentum in quantum mechanics. The angular momentum associated with a nuclear spin is characterised by the quantum number I, which may be integer or half-integer. Spin-½ nuclei are the most commonly studied in NMR, as those with $I > \frac{1}{2}$ often have very short-lived excited states, due to electric quadrupole effects. While a rigorous description of the theory behind NMR requires quantum mechanics[20, 21, 22], the fundamentals can be understood using a classical description of angular momentum.

The nuclear spin angular momentum $\boldsymbol{I} \in \mathbb{R}^3$ is a vector with the following (square) magni-

Nucleus	I	$\gamma\,(\mathrm{rad\,T^{-1}\,s^{-1}})$	Relative Abundance (%)
^{1}H	½	2.6752×10^8	99.9885
^{2}H	1	4.1066×10^7	0.0115
^{6}Li	1	3.9371×10^7	7.59
^{7}Li	3/2	1.0398×10^8	92.41
^{12}C	0	-	98.93
^{13}C	½	6.7283×10^7	1.07
^{14}N	1	1.9338×10^7	99.636
^{15}N	½	-2.7126×10^7	0.364
^{16}O	0	-	99.756
^{17}O	5/2	-3.6281×10^7	0.038
^{19}F	½	2.5162×10^8	100
^{31}P	½	1.0839×10^8	100
^{113}Cd	½	-5.9609×10^7	12.26

TABLE 1.1: A table of regularly encountered nuclei in NMR, along with common nuclei which are not NMR active.

tude:

$$\boldsymbol{I}^2 = \boldsymbol{I} \cdot \boldsymbol{I} = \hbar I(I+1) \tag{1.1}$$

where $\hbar = {}^h/{}_{2\pi}$ is the reduced Planck constant. Whilst it is not possible to specify multiple components of the angular momentum simultaneously due to the uncertainty principle, it is possible to specify one along with $\boldsymbol{I}^2$. Conventionally, this is chosen to be the z-component, I_z, for which

$$I_z = \hbar m \tag{1.2}$$

where $m \in \{-I, -I+1, \cdots, +I\}$. Eqs. 1.1 & 1.2 imply that the orientation of the z-component may only adopt certain discrete values (i.e. it is quantised). A nucleus with non-zero spin has an associated *magnetic moment*, given by:

$$\boldsymbol{\mu} = \gamma \boldsymbol{I} \implies \mu_z = \gamma I_z = \gamma \hbar m \tag{1.3}$$

γ is a proportionality constant called the *gyromagnetic ratio*, which is dependent on the nucleus of interest. Table 1.1 provides the gyromagnetic ratios for a few nuclei commonly encountered in NMR.

Without the prescence of an external field, the nuclear spin states of different m are degenerate. However, when subjected to a magnetic field, the energies of the different states diverge. This
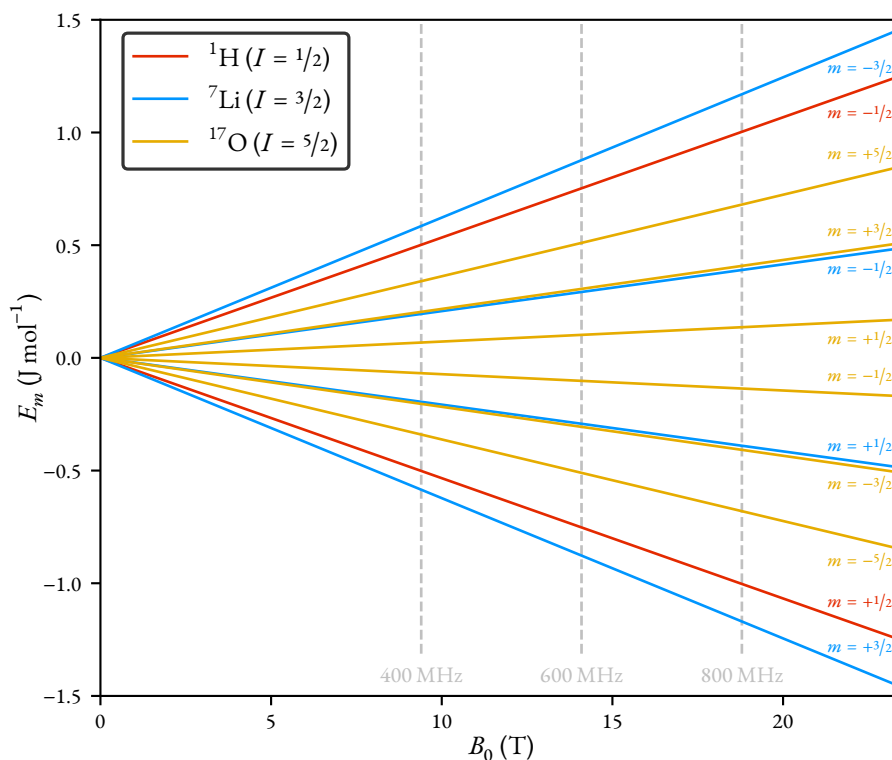
FIGURE 1.1: The variation of energy of the spin states of ^{1}H, ^{7}Li, and ^{17}O with external magnetic field strength (B_0), up to 23.5 T, which is approximately the strength of a 1 GHz NMR magnet. Three common field strengths for commercial NMR magnets are indicated: 9.40 T (400 MHz), 14.10 T (600 MHz), and 18.79 T (800 MHz).

phenomenum is known as the *Zeeman effect*. the associated energy of a given magnetic moment is given by

$$E = -\boldsymbol{\mu} \cdot \boldsymbol{B} \tag{1.4}$$

where $\boldsymbol{B}$ is the magnetic field vector. It is conventional to define the external field as directed along the laboratory z-axis, such that $B_x = B_y = 0$ and $B_z = B_0$ where B_0 is the magnetic field strength. The energies of the individual spin states are therefore

$$E_m = -\gamma I_z B_0 = -m\hbar\gamma B_0 \tag{1.5}$$

Figure 1.1 illustrates how the relative energies of different spin states vary with B_0.

REVIST THIS SECTION NMR samples comprise a vast ensemble of equivalent spin systems, and it is the macroscopic properties of the sample that are observed. Consider an ensemble of N spin-I nuclei. At thermal equilibrium, the various spin states will be disproportionately populated in accordance with the Boltzmann distribution. Assuming that $m\hbar\gamma B_0/k_{\mathrm{B}}T \gg 1$, which is the case for conventional temperatures, the proportion of spins in state m can be shown to be given

approximately by[23]

$$\frac{N_m}{N} \approx \frac{1}{2I+1}\left(1 + \frac{m\hbar\gamma B_0}{k_{\mathrm{B}}T}\right) \tag{1.6}$$

where k_{B} is the Boltzmann constant. The ensemble has an associated bulk magnetic moment $\boldsymbol{M}$, given by the summation of all the indivual spin magnetic moments:

$$\boldsymbol{M} = \gamma\boldsymbol{J} = \sum_{n=1}^{N} \boldsymbol{\mu}_n \tag{1.7}$$

where $\boldsymbol{J}$ is the spin angular momentum of the ensemble. At equilibrium, the x- and y-components of the bulk magnetisation are zero, i.e. $\sum_{n=1}^{N}\mu_{x,n} = \sum_{n=1}^{N}\mu_{y,n} = 0$, such that magnetisation is colinear with the field direction: $\boldsymbol{M} = [0, 0, M_0]^{\mathrm{T}}$, where M_0 is the magnitude of the bulk magnetism. The time evolution of the angular momentum is equivalently the torque, which is given by

$$\boldsymbol{\tau} = \frac{\mathrm{d}\boldsymbol{J}(t)}{\mathrm{d}t} = \boldsymbol{M}(t) \times \boldsymbol{B}(t) \tag{1.8}$$

Therefore, the rate of change of the bulk magnetism is

$$\frac{\mathrm{d}\boldsymbol{M}(t)}{\mathrm{d}t} = \boldsymbol{M}(t) \times \gamma\boldsymbol{B}(t), \tag{1.9}$$

which results in the following expression under the assumption that $\boldsymbol{B}(t)$ is static and directed along the z-direction:

1.1.3 The NMR Spectrometer

A modern NMR spectrometer has the capability of conducting a plethora of experiments, thanks to the availability of high magnetic fields, sophisticated electronics for pulse generation, and signal acquisition. An overview of the key components of the spectrometer is presented here.

The Magnet

The resolution of NMR data increases linearly with B_0, while sensitivity increases as $B_0^{3/2}$ [20], such that it is desirable to use magnets that produce the highest fields possible.* Conventional NMR magnets feature a superconducing solenoid immersed in liquid He, which has a boiling point of 4.2 K at atmospheric pressure. Common materials used for the solenoid include the type II superconductors Nb-Ti and Nb_3Sn. The dewar that the Helium is contained in is lined with a thermal radiation shield, which is surrounded by a larger dewar of liquid N_2, to minimise the extent of He evaporation. Passing through the magnet's z-direction is a bore, which is maintained at ambient

---

*This relation isn't perfect, when features such as a nucleus' chemical shift anisotropy are considered.

temperature, or some-other user specified temperature. Within this bore sits the sample, as well as the probe.

To maintain high spatial field homogeneity - a necessity for results of acceptable resolution - a series of coils called *shims* surround the sample. Each coil produces a weak magnetic field with a specific spatial profile, which can cancel out any inhomogeneity inherent to the main magnet.

The Probe

The probe features the coil(s) used to pulse the sample with radio-frequency (RF) radiation, as well as recieve the resultant signal. A relatively recent development in probe technology is the cryogenic probe[**Styles1984**, **Styles1989**, 24]. The transmit/recieve coil(s) and other probe circuitry are maintained at very low temperature (~ 25K), which reduces Johnson-Nyquist noise. This manifests in a higher signal-to-noise ratio (SNR), which can be as large as 4 times that achievable with conventional probes.

The Transmitter

The transmitter is the collection of components which generates RF pulses. A synthesiser acts as an RF source, producing a continuous carrier wave at or very close to the Larmor frequency. Pulses are subsequently produced by a modulator, and are then amplitfied from a power of a few mW, to daW, or even hW[**Keeler2010**].

The Reciever

The Pulse Programmer

1.1.4   The Acquisition and Structure of NMR Data

In NMR experiments, RF pulses manipulate nuclear magnetic spin states. While there is a plethora of different pulse sequences, the net effect of (virtually[†]) all of them is the generation of some magnetisation which is transverse to the field direction. The evolution of this magnetisation induces a time-varying electromotive force within the coil(s) of the probe, with the signal generated being amplified and digitised by the receiver, generating a Free Induction Decay (FID). The FID is sampled at equally-spaced time intervals Δ_t, such that it is of the form

$$\boldsymbol{y} = \begin{bmatrix} y(t = 0) & y(t = \Delta_t) & y(t = 2\Delta_t) & \cdots & y(t = (N-1)\Delta_t) \end{bmatrix}^{\mathrm{T}} \tag{1.10}$$

[†]In a pulse calibration experiment for example, the goal is to rotate the spin magnetisation by exactly 360°, leading to only longitudinal (undetectable) magnetisation.
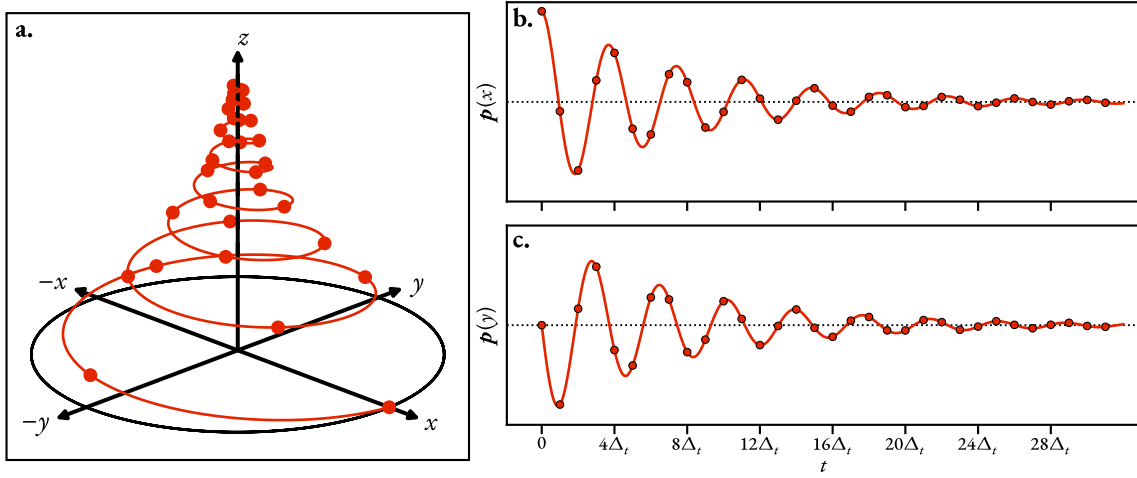
FIGURE 1.2: **If you have time, and can be bothered, the $p(x)$ and $p(y)$ axis labels should not be bold...** An illustration of quadrature detection using the vector model[25: Chapter 1]. **a.** An illustration of the evolution of the bulk magnetisation for a system comprising many identical spins, aligned initially along the laboratory x-axis (for example, immediately after a $90°_{-y}$ pulse in a pulse-acquire experiment). At regular time intervals separated by Δ_t, the signal is sampled, such that the projection onto the x-axis, $p(x)$, and onto the y-axis, $p(y)$ is measured. The projections are given in panels **b.** and **c.**. The FID at each time-point is then given by the complex value $p(x) + \mathrm{i}p(y)$.

where $y(t)$ is the (continuous) variation of the generated signal with time, and N is the number of points sampled. The inverse of the sampling rate, $1/\Delta_t$ is the sweep width or spectral window f_{sw} which defines how wide the range of samplable frequencies is (see Appendix Section B.1 for more on this).

On modern spectrometers, a quadrature acquisition system is used[23: Appendix A.5] (see Figure 1.2), such that FIDs adopt the form of a summation of M complex exponentials (oscillators), where $M \in \mathbb{N}$ is the number of resonances contributing to the signal. Each resonance will be subjected to damping due to relaxation phenomena, which is typically exponential in nature. An FID acquired with N samples therefore conventionally takes the form

$$\mathbb{C}^N \ni \boldsymbol{y}[n] = \boldsymbol{x}[n] + \boldsymbol{w}[n] \quad \forall n \in \mathbb{N}_0 : n < N, \tag{1.11a}$$

$$\boldsymbol{x}[n] = \sum_{m=0}^{M-1} \boldsymbol{a}[m] \exp(\mathrm{i}\boldsymbol{\phi}[m]) \exp([2\pi\mathrm{i}(\boldsymbol{f}[m] - f_{\mathrm{off}}) - \boldsymbol{\eta}[m]] n\Delta_t). \tag{1.11b}$$

f_{off} is included to account for the transmitter offset, which is the difference between the carrier wave frequency of the transmitter and the spectrometer basic frequency. (1.11) indicates that an FID comprises contributions from the (modellable) evolution of the spin magnetisation $\boldsymbol{x}$ and experimental noise $\boldsymbol{w}$ (*vide infra*). Each oscillator is defined by four parameters:

❖ $a \in \mathbb{R}_{>0}$ - the amplitude,

❖ $\phi \in (-\pi, \pi]$ - the phase,

❖ $f \in \left[f_{\text{off}} - {}^1\!/_2 f_{\text{sw}}, f_{\text{off}} + {}^1\!/_2 f_{\text{sw}} \right]$ - the frequency,

❖ $\eta \in \mathbb{R}_{>0}$ - the damping factor.

an FID can therefore be parameterised by the vector $\boldsymbol{\theta} \in \mathbb{R}^{4M}$:

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}} & \boldsymbol{\phi}^{\mathrm{T}} & \boldsymbol{f}^{\mathrm{T}} & \boldsymbol{\eta}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{1.12}$$

Multidimensional experiments involve incrementing one or more experimental parameters, and acquiring multiple one-dimensional (1D) FIDs with the same form as in (1.11) for each parameter/set of parameters. In a D-dimensional dataset, each contributing resonance is parameterised by an amplitude and phase as before, and also D distinct frequencies and corresponding damping factors, such that a general parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{2(1+D)M}$ is given by

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}} & \boldsymbol{\phi}^{\mathrm{T}} & \left[\boldsymbol{f}^{(1)} \right]^{\mathrm{T}} & \cdots & \left[\boldsymbol{f}^{(D)} \right]^{\mathrm{T}} & \left[\boldsymbol{\eta}^{(1)} \right]^{\mathrm{T}} & \cdots & \left[\boldsymbol{\eta}^{(D)} \right]^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{1.13}$$

where $\boldsymbol{f}^{(D)}$ and $\boldsymbol{\eta}^{(D)}$ are the frequencies and damping factors in the actively acquired (direct) dimension, and $\boldsymbol{f}^{(1)} - \boldsymbol{f}^{(D-1)}$ and $\boldsymbol{\eta}^{(1)} - \boldsymbol{\eta}^{(D-1)}$ are those for the indirect dimension(s). Indirect dimensions can exhibit different forms of evolution, depending on the precise nature of the pulse sequence. Two common functional forms exist[22: Section 4.3.4]. Signals of the form $\cos(2\pi f t)$ and $\sin(2\pi f t)$ modulate the amplitude of the direct dimension signal across iterations, while those of the form $\exp(2\pi \mathrm{i} f t)$ or $\exp(-2\pi \mathrm{i} f t)$ modulate the phase. Amplitude- or phase-modulated signals are often acquired as pairs when possible, as this ensures that frequency-discriminated spectra can be generated with absorption-mode Lorentzian lineshapes in all dimensions (*vide infra*). In general, a D-dimensional FID $\boldsymbol{Y} \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}$ can be expressed as

$$\boldsymbol{Y}\left[n^{(1)}, \cdots, n^{(D)} \right] = \sum_{m=0}^{M-1} \boldsymbol{a}\left[m \right] \exp\left(\mathrm{i}\boldsymbol{\phi}\left[m \right] \right) \times$$

$$\prod_{d=1}^{D} \zeta\left(2\pi \left(\boldsymbol{f}^{(d)}\left[m \right] - f_{\text{off}}^{(d)} \right) n^{(d)} \Delta_t^{(d)} \right) \exp\left(-\boldsymbol{\eta}^{(d)}\left[m \right] n^{(d)} \Delta_t^{(d)} \right) + \tag{1.14a}$$

$$\boldsymbol{W}\left[n^{(1)}, \cdots, n^{(D)} \right]$$

$$\boldsymbol{Y}\left[n^{(1)}, \cdots, n^{(D)} \right] = \sum_{m=0}^{M-1} \boldsymbol{\alpha}\left[m \right] \prod_{d=1}^{D} \boldsymbol{z}^{(d)}\left[m \right] + \boldsymbol{W}\left[n^{(1)}, \cdots, n^{(D)} \right], \tag{1.14b}$$

$$\zeta(\cdot) \begin{cases} = \exp(\mathrm{i}\cdot) & d = D \\ \in \{\cos(\cdot), \sin(\cdot), \exp(\mathrm{i}\cdot) \exp(-\mathrm{i}\cdot)\} & \text{otherwise} \end{cases}, \tag{1.14c}$$

Without any specific knowledge about the experiment, it is typical to assume that $\boldsymbol{W}$ is an array

of Additive White Gaussian Noise (AWGN). This means that the noise instances are assumed to be described by a complex normal distribution with mean 0, and pairs of noise instances are statistically independent, regardless of their time separation:

$$\boldsymbol{W}\left[n^{(1)}, \cdots, n^{(D)}\right] := w \sim \mathcal{N}_{\mathcal{C}}\left(0, 2\sigma^2\right)$$
$$\implies \Re\left(w\right) \perp\!\!\!\perp \Im\left(w\right), \Re\left(w\right) \sim \mathcal{N}\left(0, \sigma^2\right), \Im\left(w\right) \sim \mathcal{N}\left(0, \sigma^2\right). \tag{1.15}$$

Double check the equations are right for the SNR below... The extent by which a signal is corrupted by noise is given by the signal-to-noise ratio (SNR), the ratio of signal power and noise power: **definition of power in appendix**

$$\text{SNR}\left(\boldsymbol{Y}\right) := \frac{P_{\boldsymbol{X}}}{P_{\boldsymbol{W}}},$$
$$= \frac{1}{2\Re\sigma^2} \sum_{n^{(1)}=0}^{N^{(1)}-1} \cdots \sum_{n^{(d)}=0}^{N^{(D)}-1} \left|\boldsymbol{X}\left[n^{(1)}, \cdots, n^{(D)}\right]\right|^2, \tag{1.16}$$

where $\Re := N^{(1)} \cdots N^{(D)}$ is the total number of points the signal comprises. Due to he large dynamic range that the SNR can exhibit, it is commonplace to express it using a logarithmic scale instead, in decibels (dB):

$$\text{SNR}_{\text{dB}} := 10 \log_{10}\left(\text{SNR}\right). \tag{1.17}$$

From this, the noise variance associated with a signal a given SNR_{dB} is

$$\sigma^2 = \frac{1}{20^{\text{SNR}_{\text{dB}}/10}\Re} \sum_{n^{(1)}=0}^{N^{(1)}-1} \cdots \sum_{n^{(d)}=0}^{N^{(D)}-1} \left|\boldsymbol{X}\left[n^{(1)}, \cdots, n^{(D)}\right]\right|^2 \tag{1.18}$$

1.2 An Overview of NMR Data Analysis

To gain insights from NMR experiments on the chemical system of interest, extraction of the defining parameters $\boldsymbol{\theta}$ is necessary, though the majority of NMR users are unlikely to think about the process of NMR analysis in this way. For example:

✦ An understanding of the chemical environments of atoms in a molecule can be gained by considering the chemical shifts of the various peaks in the spectra, which are a proxy for the FID frequencies $\boldsymbol{f}$.

✦ The relative stoichiometries**does this make sense?** of a molecule can be elucidated by inspecting the integrals of spectral peaks, which are directly related to the FID amplitudes

a

- **Structure of NMR data**

- **Typical approach to analysing the data (FT, peak pick, integrate, baseline correction, window functions, zero-filling etc),**

- **Estimation techniques: LP, SVD techniques, iterative techniques (AMARES, VARPRO), Bayesian techniques (CRAFT), ML techniques**

1.2.1   Conventional NMR Analysis

1.2.2   Estimation Techniques for NMR Analysis

THEORY

The human spirit cannot be overcome. You know, as a writer, if you took away my paper,
I would write on my heart. If you take away my ink, I'd write on the wind.

It wouldn't be an ideal way to work.

— Garth Marenghi, Garth Marenghi's Darkplace

This chapter provides a detailed outline of an estimation routine which has been developed for the parametric estimation of Free Induction Decays (FIDs). **More detail?**

2.1   Outline of the Problem

$$\boldsymbol{Y}\left[n^{(1)},\cdots,n^{(D)}\right] = \boldsymbol{X}\left(\boldsymbol{\theta}\right)\left[n^{(1)},\cdots,n^{(D)}\right] + \boldsymbol{W}\left[n^{(1)},\cdots,n^{(D)}\right] \quad \forall n^{(d)} \in \mathbb{N}_0 : n^{(d)} < N^{(d)},$$

(2.1a)

$$\boldsymbol{X}\left(\boldsymbol{\theta}\right)\left[n^{(1)},\cdots,n^{(D)}\right] = \sum_{m=0}^{M-1} \boldsymbol{a}\left[m\right] \exp\left(\mathrm{i}\boldsymbol{\phi}\left[m\right]\right) \prod_{d=1}^{D} \exp\left(\left[2\pi\mathrm{i}\left(\boldsymbol{f}^{(d)}\left[m\right] - f_{\mathrm{off}}^{(d)}\right) - \boldsymbol{\eta}^{(d)}\left[m\right]\right] n^{(d)} \Delta_t^{(d)}\right),$$

(2.1b)

$$\boldsymbol{W}\left[n^{(1)},\cdots,n^{(D)}\right] \sim \mathcal{N}_{\mathcal{C}}\left(0, 2\sigma^2\right),$$

(2.1c)

where $\Delta_t^{(d)} = 1/f_{\mathrm{sw}}^{(d)}$. Under this model, it is assumed that an FID consists of a summation of M damped complex sinusoids in the presence in Additive White Gaussian Noise (AWGN). It is the goal of parametric estimation to establish the identity of all the quantities which describe the model component $\boldsymbol{X}$. These can be distilled into the vector $\boldsymbol{\theta} \in \mathbb{R}^{2(D+2)M}$:

Due to the assumed AWGN nature of the noise array, the probability density function (pdf) of an individual noise component $w := \boldsymbol{W}\left[n^{(1)},\cdots,n^{(D)}\right]$ is

$$p(w) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|w|^2}{2\sigma^2}\right).$$

(2.2)

as the elements are independent and identically distributed, the joint pdf describing the entire noise array is given by the product of each element's pdf:

$$
p\left(\boldsymbol{W}\right) = \prod_{n^{(1)}=0}^{N^{(1)}-1} \cdots \prod_{n^{(D)}=0}^{N^{(D)}-1} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|w|^2}{2\sigma^2}\right)
$$

$$
= \frac{1}{\left(2\pi\sigma^2\right)^{\mathfrak{N}}} \exp\left(-\frac{\|\boldsymbol{W}\|^2}{2\sigma^2}\right) \tag{2.3}
$$

where $\mathfrak{N} = N^{(1)} \cdots N^{(D)}$ is the total number of points in the signal. As the noise array is the difference between the data and model, the likelihood function of $\boldsymbol{\theta}$ given $\boldsymbol{Y}$, $\mathcal{L}\left(\boldsymbol{\theta}|\boldsymbol{Y}\right)$, is given by

$$
\mathcal{L}\left(\boldsymbol{\theta}|\boldsymbol{Y}\right) = \frac{1}{\left(2\pi\sigma^2\right)^{\mathfrak{N}}} \exp\left(-\frac{\|\boldsymbol{Y}-\boldsymbol{X}(\boldsymbol{\theta})\|^2}{2\sigma^2}\right). \tag{2.4}
$$

It is common to consider instead the log-likelihood function of $\boldsymbol{\theta}$ given $\boldsymbol{Y}$, $\ell\left(\boldsymbol{\theta}|\boldsymbol{Y}\right)$. As application of the logarithm is a monotonic transformation, the arguments of the maxima of $\mathcal{L}$ and ℓ are equivalent.

$$
\ell\left(\boldsymbol{\theta}|\boldsymbol{Y}\right) = -\mathfrak{N}\ln\left(2\pi\sigma^2\right) - \frac{\|\boldsymbol{Y}-\boldsymbol{X}(\boldsymbol{\theta})\|^2}{2\sigma^2}. \tag{2.5}
$$

Equation 2.5 implies that the optimal set of parameters $\boldsymbol{\theta}^{(*)}$ is that which minimises the (square) norm of the difference between the data and model

$$
\boldsymbol{\theta}^{(*)} = \underset{\boldsymbol{\theta}\in\mathbb{R}^{2(D+2)M}}{\arg\max}\ \ell\left(\boldsymbol{\theta}|\boldsymbol{Y}\right) \equiv \underset{\boldsymbol{\theta}\in\mathbb{R}^{2(D+2)M}}{\arg\min}\ \|\boldsymbol{Y}-\boldsymbol{X}(\boldsymbol{\theta})\|^2. \tag{2.6}
$$

The application of non-linear programming (NLP) is a well-established approach to solve such a problem[26, 27]. The basic principle behind NLP is to iteratively explore - in a methodical way - how a function varies with its arguments, using information about the function and optionally its derivatives, and to terminate once conditions have been achieved which imply an extremum has been found. While derivative-free approaches to NLP do exist,**(cite examples like simplex, simmulated annealing, BOBYQA?)** in scenarios where the function under consideration has well-defined, computationally tractable derivatives, the use of these can be valuable to solving optimisation problems. The problem outlined in Equation 2.6 is such an example.

2.2   Generating an Initial Guess: Matrix Pencil Method

In order for NLP to perform effectively, a large amount of *a priori* information is typically required, in the form of an initial guess $\boldsymbol{\theta}^{(0)}$. Here, a description of one such approach to achieve this is presented blah blah blah...

2.2.1 1D Matrix Pencil Method

The Matrix Pencil Method (MPM), developed by Hua and Sarkar[28, 29, 30], provides a route to extracting the signal poles of a one-dimensional (1D) dataset, based on the assumption that the number or oscillators M is known.

Noiseless data

To motivate how it works, first consider a dataset which is devoid of noise, such that it is of the form $X(\theta)$, given in Equation 2.1b, with $D = 1$:

$$
X(\theta)\left[n^{(1)}\right] = \sum_{m=0}^{M-1} \underbrace{a[m]\exp\left(\mathrm{i}\phi[m]\right)}_{\alpha[m]} \underbrace{\exp\left(\left[2\pi\mathrm{i}\left(f^{(1)}[m] - f_{\mathrm{off}}\right) - \gamma^{(1)}[m]\right]n^{(1)}\Delta_t^{(1)}\right)}_{z^{(1)}[m]^{n^{(1)}}}.
$$
(2.7)

Consider the Hankel matrix $H_X \in \mathbb{C}^{(N^{(1)}-L^{(1)})\times(L^{(1)}+1)}$:

$$
H_X = \begin{bmatrix}
X[0] & X[1] & \cdots & X\left[L^{(1)}\right] \\
X[1] & X[2] & \cdots & X\left[L^{(1)}+1\right] \\
\vdots & \vdots & \ddots & \vdots \\
X[N^{(1)}-L^{(1)}-1] & X[N^{(1)}-L^{(1)}] & \cdots & X[N^{(1)}-1]
\end{bmatrix}.
$$
(2.8)

This matrix comprises windowed segments of the FID, with each row comprising the segment shifted to the right by one point relative to the row above. $L^{(1)} \in \mathbb{N}$ is the *pencil parameter*, which dictates the size of each window Define the two matrices H_{X1} and H_{X2}, formed by the removal of the last or first column of H_X, respectively:

$$
H_{X1} = \begin{bmatrix}
X[0] & X[1] & \cdots & X\left[L^{(1)}-1\right] \\
X[1] & X[2] & \cdots & X\left[L^{(1)}\right] \\
\vdots & \vdots & \ddots & \vdots \\
X[N^{(1)}-L^{(1)}-1] & X[N^{(1)}-L^{(1)}] & \cdots & X[N^{(1)}-2]
\end{bmatrix},
$$
(2.9a)

$$
H_{X2} = \begin{bmatrix}
X[1] & X[2] & \cdots & X\left[L^{(1)}\right] \\
X[2] & X[3] & \cdots & X\left[L^{(1)}+1\right] \\
\vdots & \vdots & \ddots & \vdots \\
X[N^{(1)}-L^{(1)}] & X[N^{(1)}-L^{(1)}+1] & \cdots & X[N^{(1)}-1]
\end{bmatrix}.
$$
(2.9b)

These matrices can be deconstructed into the following forms involving matrices containing the M signal poles and complex amplitudes that the data comprises:

$$\boldsymbol{H}_{X1} = \boldsymbol{Z}_{\mathrm{L}} \boldsymbol{A} \boldsymbol{Z}_{\mathrm{R}}, \tag{2.10a}$$

$$\boldsymbol{H}_{X2} = \boldsymbol{Z}_{\mathrm{L}} \boldsymbol{A} \boldsymbol{Z}_{\mathrm{D}} \boldsymbol{Z}_{\mathrm{R}}, \tag{2.10b}$$

$$\mathbb{C}^{(N^{(1)} - L^{(1)}) \times M} \ni \boldsymbol{Z}_{\mathrm{L}} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(1)} & \boldsymbol{z}^{(1)2} & \cdots & \boldsymbol{z}^{(1)N^{(1)} - L^{(1)} - 1} \end{bmatrix}^{\mathrm{T}}, \tag{2.10c}$$

$$\mathbb{C}^{M \times L^{(1)}} \ni \boldsymbol{Z}_{\mathrm{R}} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(1)} & \boldsymbol{z}^{(1)2} & \cdots & \boldsymbol{z}^{(1)L^{(1)} - 1} \end{bmatrix}, \tag{2.10d}$$

$$\mathbb{C}^{M \times M} \ni \boldsymbol{Z}_{\mathrm{D}} = \mathrm{diag}\left(\boldsymbol{z}^{(1)}\right), \tag{2.10e}$$

$$\mathbb{C}^{M \times M} \ni \boldsymbol{A} = \mathrm{diag}\left(\boldsymbol{\alpha}\right). \tag{2.10f}$$

Description of what a matrix pencil is The matrix pencil $\boldsymbol{H}_{X2} - \lambda \boldsymbol{H}_{X1}, \lambda \in \mathbb{C}$ can therefore be expressed as

$$\boldsymbol{H}_{X2} - \lambda \boldsymbol{H}_{X1} = \boldsymbol{Z}_{\mathrm{L}} \boldsymbol{A} \left(\boldsymbol{Z}_{\mathrm{D}} - \lambda \boldsymbol{I}_M\right) \boldsymbol{Z}_{\mathrm{R}}, \tag{2.11}$$

where $\boldsymbol{I}_M \in \mathbb{C}^{M \times M}$ is the identity matrix. Assuming that the following condition is met:

$$M \leq L^{(1)} \leq N^{(1)} - M, \tag{2.12}$$

the rank of the matrix pencil will be M. Equation 2.12 must be obeyed to ensure that both the number of rows and columns of the matrix pencil are at least M. Now consider the case when the scalar λ is equal to one of the signal poles i.e. $\lambda = \boldsymbol{z}[m] \ \forall m \in \{0, \cdots, M - 1\}$. The element $\left(\boldsymbol{Z}_{\mathrm{D}} - \lambda \boldsymbol{I}_M\right)[m, m]$ will be set to 0, which will lead to the determinant of the matrix pencil being 0. The eigenvalues of the matrix pencil are the solution of the so-called *generalised eigenvalue problem*, and are defined as[31: Section 7.7]

$$\boldsymbol{z} = \left\{z \in \mathbb{C} : \det\left(\boldsymbol{H}_{X2} - z \boldsymbol{H}_{X1}\right) = 0\right\} \tag{2.13}$$

One means of finding the signal poles is by finding the eigenvalues of the matrix $\boldsymbol{H}_{X1}^+ \boldsymbol{H}_{X2}$. Deriving the corresponding complex amplitudes can then be achieved by solving the set of linear equations

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(1)} & \boldsymbol{z}^{(1)2} & \cdots & \boldsymbol{z}^{(1)N^{(1)} - 1} \end{bmatrix}^{\mathrm{T}} \boldsymbol{\alpha} \equiv \boldsymbol{Z}^{(1)} \boldsymbol{\alpha} \implies \boldsymbol{\alpha} = \boldsymbol{Z}^+ \boldsymbol{X}. \tag{2.14}$$

Extraction of the amplitudes, phases, frequencies, and damping factors from the signal poles and complex amplitudes can then take place:

$$\boldsymbol{a} = \left|\boldsymbol{\alpha}\right|, \tag{2.15a}$$

$$\phi = \arctan\left(\frac{\Im(\alpha)}{\Re(\alpha)}\right), \tag{2.15b}$$

$$f^{(1)} = \frac{f_{\text{sw}}^{(1)}}{2\pi}\Im\left(\ln z^{(1)}\right) + f_{\text{off}}, \tag{2.15c}$$

$$\eta^{(1)} = -f_{\text{sw}}^{(1)}\Re\left(\ln z^{(1)}\right). \tag{2.15d}$$

Noisy data

The presence of noise in the signal Y complicates the process of determining the M signal poles, as H_Y (H_X's equivalent with elements replaced by the noisy data) is likely to be full-rank ($\min(N^{(1)} - L^{(1)}, L^{(1)} + 1)$). To cope with this, it is necessary to generate a rank-reduced matrix $\widetilde{H}_Y$. By employing the Eckart-Young-Mirsky (EYM) theorem[31: Section 2.2], an appropriate matrix is can be obtained through Singular Value Decomposition (SVD)**Appendix description of SVD**:

$$\widetilde{H}_Y = U_M \Sigma_M V_M^\dagger, \tag{2.16a}$$

$$\mathbb{C}^{(N^{(1)}-L^{(1)})\times M} \ni U_M = \begin{bmatrix} u_1 & u_2 & \cdots & u_M \end{bmatrix}, \tag{2.16b}$$

$$\mathbb{C}^{(L^{(1)}+1)\times M} \ni V_M = \begin{bmatrix} v_1 & v_2 & \cdots & v_M \end{bmatrix}, \tag{2.16c}$$

$$\mathbb{C}^{M\times M} \ni \Sigma_M = \text{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_M\right). \tag{2.16d}$$

σ_m is the mth largest singular value of H_Y, $u_m \in \mathbb{C}^{N^{(1)}-L^{(1)}}$ and $v_m \in \mathbb{C}^{L^{(1)}+1}$ are the corresponding left and right singular vectors, respectively. The EYM proves that $\widetilde{H}_Y$ is the closest matrix of rank M to H_Y in a Frobenius norm sense, i.e.

$$\widetilde{H}_Y = \underset{A:\ \text{rank}(A)=M}{\arg\min}\ \|A - H_Y\| \tag{2.17}$$

With a rank-reduced matrix produced from the noisy matrix, the signal poles can then be derived from the eigenvalues of $\widetilde{H}_{Y1}^+ \widetilde{H}_{Y2}$, where $\widetilde{H}_{Y1}$ and $\widetilde{H}_{Y2}$ have the same relation to $\widetilde{H}_Y$ as H_{X1} and H_{X2} do to H_X. As a less expensive alternative, the same result can be achieved by computing the eigenvalues of $V_{M1}^+ V_{M2}$, with

$$V_{M1} = \begin{bmatrix} v_1 & v_2 & \cdots & v_{M-1} \end{bmatrix}, \tag{2.18a}$$

$$V_{M2} = \begin{bmatrix} v_2 & v_3 & \cdots & v_M \end{bmatrix}. \tag{2.18b}$$

Algorithm 1 provides a pseudo-code description of the MPM as implemented in the NMR Estimation in Python (NMR-EsPy) package.

Discuss complexity of MPM, talking about reducing $N^{(1)}$ and $L^{(1)}$ as a means of reducing the cost.

Algorithm 1 The Matrix Pencil Method.

1: **procedure** MPM($\boldsymbol{Y} \in \mathbb{C}^{N^{(1)}}, M \in \mathbb{N}$)

2:      $L \leftarrow \lfloor N^{(1)}/3 \rfloor$;

3:      $\boldsymbol{H}_Y \leftarrow \begin{bmatrix} \boldsymbol{Y}[0] & \boldsymbol{Y}[1] & \cdots & \boldsymbol{Y}[L^{(1)}] \\ \boldsymbol{Y}[1] & \boldsymbol{Y}[2] & \cdots & \boldsymbol{Y}[L^{(1)}+1] \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}[N^{(1)}-L^{(1)}-1] & \boldsymbol{Y}[N^{(1)}-L^{(1)}] & \cdots & \boldsymbol{Y}[N^{(1)}-1] \end{bmatrix}$;

4:      $\boldsymbol{U}, \boldsymbol{\sigma}, \boldsymbol{V}^\dagger \leftarrow \text{SVD}\left(\boldsymbol{H}_Y\right)$;

5:      $\boldsymbol{V} \leftarrow \left[\boldsymbol{V}^\dagger\right]^\dagger$;

6:      $\boldsymbol{V}_M \leftarrow \boldsymbol{V}\left[:,:M\right]$;   ▷ Retain first M right singular vectors.

7:      $\boldsymbol{V}_{M1}, \boldsymbol{V}_{M2} \leftarrow \boldsymbol{V}_M\left[:,:M-1\right], \boldsymbol{V}_M\left[:,1:\right]$;   ▷ Remove last/first column

8:      $\boldsymbol{z}^{(1)} \leftarrow \text{EIGENVALUES}\left(\boldsymbol{V}_{M1}^+ \boldsymbol{V}_{M2}\right)$;

9:      $\boldsymbol{Z}^{(1)} \leftarrow \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(1)} & \boldsymbol{z}^{(1)^2} & \cdots & \boldsymbol{z}^{(1)^{N^{(1)}}} \end{bmatrix}^{\mathrm{T}}$;

10:     $\boldsymbol{\alpha} \leftarrow \boldsymbol{Z}^{(1)^+} \boldsymbol{Y}$;

11:     $\boldsymbol{a}, \boldsymbol{\phi} \leftarrow |\boldsymbol{\alpha}|, \arctan\left(\frac{\Im(\boldsymbol{\alpha})}{\Re(\boldsymbol{\alpha})}\right)$;

12:     $\boldsymbol{f}^{(1)} \leftarrow \frac{f_{\text{sw}}^{(1)}}{2\pi} \Im\left(\ln \boldsymbol{z}^{(1)}\right) + f_{\text{off}}$;

13:     $\boldsymbol{\eta}^{(1)} \leftarrow -f_{\text{sw}}^{(1)} \Re\left(\ln \boldsymbol{z}^{(1)}\right)$;

14:     **if** $\boldsymbol{\eta}^{(1)}$ contains negative values **then**   ▷ Purge any oscillators with negative damping

15:        Remove these from $\boldsymbol{\eta}^{(1)}$, and remove the corresponding values from $\boldsymbol{a}$, $\boldsymbol{\phi}$, and $\boldsymbol{f}^{(1)}$;

16:     **end if**

17:     $\theta^{(0)} \leftarrow \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}} & \boldsymbol{\phi}^{\mathrm{T}} & \left[\boldsymbol{f}^{(1)}\right]^{\mathrm{T}} & \left[\boldsymbol{\eta}^{(1)}\right]^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$;

18:     **return** $\theta^{(0)}$;

19: **end procedure**

2.2.2 2D Matrix Enhancement and Matrix Pencil Method

The MPM was extended for the consideration of two-dimensional (2D) data by Hua with the Matrix Enhancement and Matrix Pencil Method (MEMPM)[32]. The method centers around the enhanced matrix $\boldsymbol{E_Y} \in \mathbb{C}^{\left(L^{(1)}L^{(2)}\right)\times\left(N^{(1)}-L^{(1)}+1\right)\left(N^{(2)}-L^{(2)}+1\right)}$, a block Hankel matrix of the form

$$
\boldsymbol{E_Y} = \begin{bmatrix} \boldsymbol{H}_{Y,0} & \boldsymbol{H}_{Y,1} & \cdots & \boldsymbol{H}_{Y,N^{(1)}-L^{(1)}} \\ \boldsymbol{H}_{Y,1} & \boldsymbol{H}_{Y,2} & \cdots & \boldsymbol{H}_{Y,N^{(1)}-L^{(1)}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{Y,L^{(1)}-1} & \boldsymbol{H}_{Y,L^{(1)}} & \cdots & \boldsymbol{H}_{Y,N^{(1)}-1} \end{bmatrix},
\tag{2.19a}
$$

$$
\boldsymbol{H}_{Y,n^{(1)}} = \begin{bmatrix} \boldsymbol{Y}\left[n^{(1)},0\right] & \boldsymbol{Y}\left[n^{(1)},1\right] & \cdots & \boldsymbol{Y}\left[n^{(1)},N^{(2)}-L^{(2)}\right] \\ \boldsymbol{Y}\left[n^{(1)},1\right] & \boldsymbol{Y}\left[n^{(1)},2\right] & \cdots & \boldsymbol{Y}\left[n^{(1)},N^{(2)}-L^{(2)}+1\right] \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}\left[n^{(1)},L^{(2)}-1\right] & \boldsymbol{Y}\left[n^{(1)},L^{(2)}\right] & \cdots & \boldsymbol{Y}\left[n^{(1)},N^{(2)}-1\right] \end{bmatrix}.
\tag{2.19b}
$$

In a similar fashion to Equation 2.10, $\boldsymbol{H}_{Y,n^{(1)}}$ can be expressed as

$$
\boldsymbol{H}_{Y,n^{(1)}} = \boldsymbol{Z}_{\mathrm{L}}^{(2)}\boldsymbol{A}\left[\boldsymbol{Z}_{\mathrm{D}}^{(1)}\right]^{n^{(1)}}\boldsymbol{Z}_{\mathrm{R}}^{(2)},
\tag{2.20a}
$$

$$
\boldsymbol{Z}_{\mathrm{L}}^{(2)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(2)} & \boldsymbol{z}^{(2)2} & \cdots & \boldsymbol{z}^{(2)L^{(2)}-1} \end{bmatrix}^{\mathrm{T}},
\tag{2.20b}
$$

$$
\boldsymbol{Z}_{\mathrm{R}}^{(2)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(2)} & \boldsymbol{z}^{(2)2} & \cdots & \boldsymbol{z}^{(2)N^{(2)}-L^{(2)}} \end{bmatrix},
\tag{2.20c}
$$

with $\boldsymbol{Z}_{\mathrm{D}}^{(1)}$ and $\boldsymbol{A}$ given by Equations 2.10e & 2.10f, respectively. This then leads to the enhanced matrix be expressed as

$$
\boldsymbol{E_Y} = \boldsymbol{E}_{\mathrm{L}}\boldsymbol{A}\boldsymbol{E}_{\mathrm{R}},
\tag{2.21a}
$$

$$
\mathbb{C}^{L^{(1)}L^{(2)}\times M} \ni \boldsymbol{E}_{\mathrm{L}} = \begin{bmatrix} \boldsymbol{Z}_{\mathrm{L}}^{(2)} \\ \boldsymbol{Z}_{\mathrm{L}}^{(2)}\boldsymbol{Z}_{\mathrm{D}}^{(1)} \\ \vdots \\ \boldsymbol{Z}_{\mathrm{L}}^{(2)}\left[\boldsymbol{Z}_{\mathrm{D}}^{(1)}\right]^{L^{(1)}-1} \end{bmatrix},
\tag{2.21b}
$$

$$
\mathbb{C}^{M\times\left(N^{(1)}-L^{(1)}+1\right)\left(N^{(2)}-L^{(2)}+1\right)} \ni \boldsymbol{E}_{\mathrm{R}} = \begin{bmatrix} \boldsymbol{Z}_{\mathrm{R}}^{(2)} & \boldsymbol{Z}_{\mathrm{D}}^{(1)}\boldsymbol{Z}_{\mathrm{R}}^{(2)} & \cdots & \left[\boldsymbol{Z}_{\mathrm{D}}^{(1)}\right]^{N^{(1)}-L^{(1)}}\boldsymbol{Z}_{\mathrm{L}}^{(2)} \end{bmatrix}.
\tag{2.21c}
$$

As was the case in the 1D MPM, SVD can be utilised to generate a filtered matrix $\widetilde{\boldsymbol{E}}_Y$ with its rank reducd to M:

$$\widetilde{\boldsymbol{E}}_Y = \boldsymbol{U}_M \boldsymbol{\Sigma}_M \boldsymbol{V}_M^\dagger \tag{2.22}$$

If the conditions $N^{(d)} - L^{(d)} + 1 \geq M \; \forall d \in \{1, 2\}$ are met, $\mathrm{range}\,(\boldsymbol{U}_M) = \mathrm{range}\,(\boldsymbol{E}_\mathrm{L})$. This implies that there is some nonsingular matrix $\boldsymbol{T} \in \mathbb{C}^{M \times M}$ such that

$$\boldsymbol{U}_M = \boldsymbol{E}_\mathrm{L} \boldsymbol{T}. \tag{2.23}$$

Now consider the following two matrices:

$$\boldsymbol{U}_{M1} = \boldsymbol{E}_{\mathrm{L}1} \boldsymbol{T}, \tag{2.24a}$$

$$\boldsymbol{U}_{M2} = \boldsymbol{E}_{\mathrm{L}1} \boldsymbol{Z}_\mathrm{D}^{(1)} \boldsymbol{T}, \tag{2.24b}$$

$$\mathbb{C}^{L^{(1)}\left(L^{(2)}-1\right) \times M} \ni \boldsymbol{E}_{\mathrm{L}1} = \begin{bmatrix} \boldsymbol{Z}_\mathrm{L}^{(2)} \\ \boldsymbol{Z}_\mathrm{L}^{(2)} \boldsymbol{Z}_\mathrm{D}^{(1)} \\ \vdots \\ \boldsymbol{Z}_\mathrm{L}^{(2)} \left[\boldsymbol{Z}_\mathrm{D}^{(1)}\right]^{L^{(1)}-2} \end{bmatrix} \quad \text{(c.f. Equation 2.21b).} \tag{2.24c}$$

$\boldsymbol{U}_{M1}$ and $\boldsymbol{U}_{M2}$ correspond the $\boldsymbol{U}_M$ with the last and first $L^{(2)}$ rows removed, respectively. The matrix pencil for $\boldsymbol{U}_{M1}$ and $\boldsymbol{U}_{M2}$ can be expressed as

$$\boldsymbol{U}_{M1} - \lambda \boldsymbol{U}_{M2} = \boldsymbol{E}_{\mathrm{L}1} \left(\boldsymbol{Z}_\mathrm{D}^{(1)} - \lambda \boldsymbol{I}_M\right) \boldsymbol{T}. \tag{2.25}$$

As seen previously, this matrix structure implies that $\boldsymbol{z}^{(1)}$ are the solutions to the generalised eigenvalue problem, such that they are the eigenvalues of $\boldsymbol{U}_{M1}^+ \boldsymbol{U}_{M2}$.

To extract the signal poles in the other dimension, $\boldsymbol{z}^{(2)}$, the permutation matrix is defined:

$$\mathbb{R}^{L^{(1)}L^{(2)}\times L^{(1)}L^{(2)}} \ni \boldsymbol{P} = \begin{bmatrix} \boldsymbol{e}\left(0\right)^{\mathrm{T}} \\ \boldsymbol{e}\left(L^{(2)}\right)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}\left(\left(L^{(1)}-1\right)L^{(2)}\right)^{\mathrm{T}} \\ \boldsymbol{e}\left(1\right)^{\mathrm{T}} \\ \boldsymbol{e}\left(1+L^{(2)}\right)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}\left(1+\left(L^{(1)}-1\right)L^{(2)}\right)^{\mathrm{T}} \\ \vdots \\ \vdots \\ \boldsymbol{e}\left(L^{(2)}-1\right)^{\mathrm{T}} \\ \boldsymbol{e}\left(2L^{(2)}-1\right)^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}\left(L^{(1)}L^{(2)}-1\right)^{\mathrm{T}} \end{bmatrix}. \tag{2.26}$$

$\boldsymbol{e}\left(i\right) \in \mathbb{R}^{L^{(1)}L^{(2)}}$ corresponds to a unit vector comprising zeros except for $\boldsymbol{e}\left[i\right] = 1$. Multiplying $\boldsymbol{E}_{\mathrm{L}}$ by the permutation matrix leads to a matrix in which the roles of the two sets of signal poles are effectively swapped:

$$\boldsymbol{E}_{\mathrm{LP}} := \boldsymbol{P}\boldsymbol{E}_{\mathrm{L}} = \begin{bmatrix} \boldsymbol{Z}_{\mathrm{L}}^{(1)} \\ \boldsymbol{Z}_{\mathrm{L}}^{(1)}\boldsymbol{Z}_{\mathrm{D}}^{(2)} \\ \vdots \\ \boldsymbol{Z}_{\mathrm{L}}^{(1)}\left[\boldsymbol{Z}_{\mathrm{D}}^{(2)}\right]^{L^{(2)}-1} \end{bmatrix}, \tag{2.27a}$$

$$\boldsymbol{Z}_{\mathrm{L}}^{(1)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(1)} & \boldsymbol{z}^{(1)2} & \cdots & \boldsymbol{z}^{(1)L^{(1)}-1} \end{bmatrix}^{\mathrm{T}}, \tag{2.27b}$$

$$\boldsymbol{Z}_{\mathrm{D}}^{(2)} = \mathrm{diag}\left(\boldsymbol{z}^{(2)}\right). \tag{2.27c}$$

Note the similarity of Equation 2.27a with Equation 2.21b, which implies that with the same reasoning as given above, $\boldsymbol{z}^{(2)}$ can be derived by extracting the eigenvalues of $\boldsymbol{U}_{MP1}^{+}\boldsymbol{U}_{MP2}$, where $\boldsymbol{U}_{MP1}$ and $\boldsymbol{U}_{MP2}$ correspond to $\boldsymbol{P}\boldsymbol{U}_M$ with the last and first $L^{(1)}$ rows removed, respectively.

In the original account on the MEMPM, the final stage involved employing a pairing algorithm in order to assign the uncorrelated signal poles in $\boldsymbol{z}^{(1)}$ with $\boldsymbol{z}^{(2)}$ [32]. The Modified Matrix Enhancement and Matrix Pencil Method (MMEMPM) was developed in order to overcome two

issues with the pairing algorithm: (a) it is computationally expensive (b) it is fallible to return incorrect pairings[33]. As well as the generalised eigenvalues of $\boldsymbol{U}_{M1} - \lambda \boldsymbol{U}_{M2}$ ($\boldsymbol{z}^{(1)}$), the MMEMPM requires extraction of the generalised eigenvectors too ($\boldsymbol{W}^{(1)}$). Assuming that there are no repeated poles in $\boldsymbol{z}^{(1)}$, the correctly paired second set of poles is then generated via

$$\boldsymbol{z}^{(2)} = \operatorname{diag}\left(\boldsymbol{W}^{-1}\boldsymbol{U}_{MP1}^{+}\boldsymbol{U}_{MP2}\boldsymbol{W}\right) \tag{2.28}$$

Talk about case of paired eigenvalues.

    See Algorithm 2

2.2.3   Model order selection: Minimum Description Length

2.3   Non-linear programming for NMR estimation

Before discussing the details of the specific NLP problem outlined above, an general overview is given.

2.3.1   An overview of non-linear programming

In an optimisation problem, the goal is to determine the minimum (or maximum) of a function $\mathcal{F}(\boldsymbol{\theta}) : \mathbb{R}^n \to \mathbb{R}, n \in \mathbb{N}$, of called the *cost function* or *fidelity*. This is typically with the goal of determining the argument $\boldsymbol{\theta}^{(*)}$ at which the optimum is found:

$$\boldsymbol{\theta}^{(*)} = \underset{\boldsymbol{\theta} \in \mathbb{R}^n}{\arg\min}\ \mathcal{F}(\boldsymbol{\theta}). \tag{2.29}$$

The above problem is unconstrained, as there are no limitations that the parameter vector is subjected to. Unless $\mathcal{F}(\boldsymbol{\theta})$ has particular properties, such as convexity[*], it is generally only possible to determine a local minimum, rather than a global minimum.

Definition 1. $\boldsymbol{\theta}^{(*)} \in \mathbb{R}^n$ *is a local minimiser of* $\mathcal{F}(\boldsymbol{\theta})$ *if there is a neighbourhood* $N \ni \boldsymbol{\theta}^{(*)}$ *such that* $\mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right) \leq \mathcal{F}(\boldsymbol{\theta})\ \forall \boldsymbol{\theta} \in N$

    It can be shown that $\boldsymbol{\theta}^{(*)}$ is a local minimiser of $\mathcal{F}(\boldsymbol{\theta})$ if the first- and second-order necessary conditions are satisfied. Furthermore, it is a unique (strict) local minimiser if the second-order sufficient condition is also satisfied.

Theorem 1 (First-order necessary condition)**.** *If* $\mathcal{F}(\boldsymbol{\theta})$ *is continuously differentiable, then the gradient vector* $\boldsymbol{g}\left(\boldsymbol{\theta}^{(*)}\right) := \nabla\mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right)$ *is the zero vector:*

$$\boldsymbol{g}\left(\boldsymbol{\theta}^{(*)}\right) = \boldsymbol{0} \in \mathbb{R}^n \tag{2.30}$$

---

[*]A convex function is one such that a line segment through any two points of the function lies above it.

Algorithm 2 The Modified Matrix Enhancement and Matrix Pencil Method.

1: **procedure** MMEMPM($\boldsymbol{Y} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, M \in \mathbb{N}$)

2: $\quad L^{(1)}, L^{(2)} \leftarrow \lfloor N^{(1)}/2 \rfloor, \lfloor N^{(2)}/2 \rfloor;$

3: $\quad$ **for** $n^{(1)} \leftarrow \{0, \cdots, N^{(1)} - 1\}$ **do**

4: $\quad\quad \boldsymbol{H}_{Y,n^{(1)}} \leftarrow \begin{bmatrix} \boldsymbol{Y}\left[n^{(1)}, 0\right] & \boldsymbol{Y}\left[n^{(1)}, 1\right] & \cdots & \boldsymbol{Y}\left[n^{(1)}, N^{(2)} - L^{(2)}\right] \\ \boldsymbol{Y}\left[n^{(1)}, 1\right] & \boldsymbol{Y}\left[n^{(1)}, 2\right] & \cdots & \boldsymbol{Y}\left[n^{(1)}, N^{(2)} - L^{(2)} + 1\right] \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Y}\left[n^{(1)}, L^{(2)} - 1\right] & \boldsymbol{Y}\left[n^{(1)}, L^{(2)}\right] & \cdots & \boldsymbol{Y}\left[n^{(1)}, N^{(2)} - 1\right] \end{bmatrix}$

5: $\quad$ **end for**;

6: $\quad \boldsymbol{E}_Y \leftarrow \begin{bmatrix} \boldsymbol{H}_{Y,0} & \boldsymbol{H}_{Y,1} & \cdots & \boldsymbol{H}_{Y,N^{(1)} - L^{(1)}} \\ \boldsymbol{H}_{Y,1} & \boldsymbol{H}_{Y,2} & \cdots & \boldsymbol{H}_{Y,N^{(1)} - L^{(1)} + 1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_{Y,L^{(1)} - 1} & \boldsymbol{H}_{Y,L^{(1)}} & \cdots & \boldsymbol{H}_{Y,N^{(1)} - 1} \end{bmatrix};$

7: $\quad \boldsymbol{U}_M, \boldsymbol{\Sigma}_M, \boldsymbol{V}_M^\dagger \leftarrow \textsc{TruncatedSVD}\left(\boldsymbol{E}_Y, M\right);$

8: $\quad \boldsymbol{P} \leftarrow \boldsymbol{0} \in \mathbb{C}^{L^{(1)}L^{(2)} \times L^{(1)}L^{(2)}};$

9: $\quad r \leftarrow 0$

10: $\quad$ **for** $i = 0, \cdots, L^{(2)} - 1$ **do**

11: $\quad\quad$ **for** $j = 0, \cdots, L^{(1)} - 1$ **do**

12: $\quad\quad\quad c \leftarrow i + jL^{(2)};$

13: $\quad\quad\quad \boldsymbol{P}\left[r, c\right] \leftarrow 1;$

14: $\quad\quad\quad r = r + 1;$

15: $\quad\quad$ **end for**

16: $\quad$ **end for**

17: $\quad \boldsymbol{U}_{M1}, \boldsymbol{U}_{M2} \leftarrow \boldsymbol{U}_M\left[: L^{(1)}\left(L^{(2)} - 1\right)\right], \boldsymbol{U}_M\left[L^{(2)} :\right];$ $\qquad\qquad$ ▷ Last/First $L^{(2)}$ rows deleted

18: $\quad \boldsymbol{z}^{(1)}, \boldsymbol{W}^{(1)} \leftarrow \textsc{Eigendecomposition}\left(\boldsymbol{U}_{M1}^+ \boldsymbol{U}_{M2}\right);$

19: $\quad \boldsymbol{f}^{(1)}, \boldsymbol{\eta}^{(1)} \leftarrow \left(f_{\mathrm{sw}}^{(1)}/2\pi\right) \Im\left(\ln \boldsymbol{z}^{(1)}\right) + f_{\mathrm{off}}^{(1)}, -f_{\mathrm{sw}}^{(1)} \Re\left(\ln \boldsymbol{z}^{(1)}\right);$

20: $\quad \boldsymbol{U}_{MP} \leftarrow \boldsymbol{P}\boldsymbol{U}_M;$

21: $\quad \boldsymbol{U}_{MP1}, \boldsymbol{U}_{MP2} \leftarrow \boldsymbol{U}_{MP}\left[: \left(L^{(1)} - 1\right)L^{(2)}\right], \boldsymbol{U}_{MP}\left[L^{(1)} :\right];$ $\qquad$ ▷ Last/First $L^{(1)}$ rows deleted

22: $\quad \boldsymbol{z}^{(2)} \leftarrow \mathrm{diag}\left(\left[\boldsymbol{W}^{(1)}\right]^{-1} \boldsymbol{U}_{MP1}^+ \boldsymbol{U}_{MP2} \boldsymbol{W}^{(1)}\right);$

23: $\quad \boldsymbol{f}^{(2)}, \boldsymbol{\eta}^{(2)} \leftarrow \left(f_{\mathrm{sw}}^{(2)}/2\pi\right) \Im\left(\ln \boldsymbol{z}^{(2)}\right) + f_{\mathrm{off}}^{(2)}, -f_{\mathrm{sw}}^{(2)} \Re\left(\ln \boldsymbol{z}^{(2)}\right);$

24: $\quad \boldsymbol{Z}_{\mathrm{L}}^{(2)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(2)} & \boldsymbol{z}^{(2)2} & \cdots & \boldsymbol{z}^{(2)L^{(2)} - 1} \end{bmatrix}^{\mathrm{T}};$

25: $\quad \boldsymbol{Z}_{\mathrm{R}}^{(2)} \leftarrow \begin{bmatrix} \boldsymbol{1} & \boldsymbol{z}^{(2)} & \boldsymbol{z}^{(2)2} & \cdots & \boldsymbol{z}^{(2)N^{(2)} - L^{(2)}} \end{bmatrix};$

26: $\quad \boldsymbol{Z}_{\mathrm{D}}^{(1)} \leftarrow \mathrm{diag}\left(\boldsymbol{z}^{(1)}\right);$

27: $\quad \boldsymbol{E}_{\mathrm{L}} \leftarrow \begin{bmatrix} \boldsymbol{Z}_{\mathrm{L}}^{(2)} \\ \boldsymbol{Z}_{\mathrm{L}}^{(2)} \boldsymbol{Z}_{\mathrm{D}}^{(1)} \\ \vdots \\ \boldsymbol{Z}_{\mathrm{L}}^{(2)} \left[\boldsymbol{Z}_{\mathrm{D}}^{(1)}\right]^{L^{(1)} - 1} \end{bmatrix};$

28: $\quad \boldsymbol{E}_{\mathrm{R}} \leftarrow \begin{bmatrix} \boldsymbol{Z}_{\mathrm{R}}^{(2)} & \boldsymbol{Z}_{\mathrm{D}}^{(1)} \boldsymbol{Z}_{\mathrm{R}}^{(2)} & \cdots & \left[\boldsymbol{Z}_{\mathrm{D}}^{(1)}\right]^{N^{(1)} - L^{(1)}} \boldsymbol{Z}_{\mathrm{R}}^{(2)} \end{bmatrix};$

29: $\quad \boldsymbol{\alpha} \leftarrow \mathrm{diag}\left(\boldsymbol{E}_{\mathrm{L}}^+ \boldsymbol{E}_Y \boldsymbol{E}_{\mathrm{R}}^+\right);$

30: $\quad \boldsymbol{a}, \boldsymbol{\phi} \leftarrow |\boldsymbol{\alpha}|, \arctan\left(\frac{\Im(\boldsymbol{\alpha})}{\Re(\boldsymbol{\alpha})}\right);$

31: $\quad \boldsymbol{\theta}^{(0)} \leftarrow \begin{bmatrix} \boldsymbol{a}^{\mathrm{T}} & \boldsymbol{\phi}^{\mathrm{T}} & \left[\boldsymbol{f}^{(1)}\right]^{\mathrm{T}} & \left[\boldsymbol{f}^{(2)}\right]^{\mathrm{T}} & \left[\boldsymbol{\eta}^{(1)}\right]^{\mathrm{T}} & \left[\boldsymbol{\eta}^{(2)}\right]^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}};$

32: $\quad$ **return** $\boldsymbol{\theta}^{(0)}$

33: **end procedure**

Theorem 2 (Second-order necessary condition). *If $\mathcal{F}(\boldsymbol{\theta})$ and $\boldsymbol{g}(\boldsymbol{\theta})$ are continuously differentiable, and $\boldsymbol{\theta}^{(*)}$ is a local minimiser of $\mathcal{F}(\boldsymbol{\theta})$, then the Hessian $\boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right) := \nabla^2 \mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right)$ is positive semidefinite, i.e.*

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right) \boldsymbol{v} \geq 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n. \tag{2.31}$$

Theorem 3 (Second-order sufficient condition). *$\boldsymbol{\theta}^{(*)}$ is a strict local minimiser of $\mathcal{F}(\boldsymbol{\theta})$ if $\boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right)$ is positive definite:*

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right) \boldsymbol{v} > 0 \ \forall \boldsymbol{v} \in \mathbb{R}^n. \tag{2.32}$$

A plethora of approaches have been established to determine local minima of scalar functions. One of the better-known strategies is *Newton's method*, in which a quadratic approximation of the fidelity is considered. For a given iteration $k \in \mathbb{N}_0$, the fidelity is approximated using

$$\mathcal{F}_Q(\boldsymbol{\theta}) = \mathcal{F}\left(\boldsymbol{\theta}^{(k)}\right) + \boldsymbol{b}^{\mathrm{T}} \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) + \frac{1}{2} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{H}\left(\boldsymbol{\theta}^{(k)}\right) \boldsymbol{b}, \tag{2.33}$$

where $\boldsymbol{b} = \boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}$. An updated prediction of the parameter vector is derived by finding the minimum of this quadratic approximation:

$$\frac{\partial \mathcal{F}(\boldsymbol{\theta})}{\partial \boldsymbol{b}} = \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) + \boldsymbol{H}\left(\boldsymbol{\theta}^{(k)}\right) \boldsymbol{b}$$

$$\implies 0 = \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) + \boldsymbol{H}\left(\boldsymbol{\theta}^{(k)}\right)\left(\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\right)$$

$$\implies \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \left[\boldsymbol{H}\left(\boldsymbol{\theta}^{(k)}\right)\right]^{-1} \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right). \tag{2.34}$$

As Theorem 1 indicates, a local minimum of $\mathcal{F}(\boldsymbol{\theta})$ is deemed to be reached when the magnitude of the gradient vector is smaller than a particular threshold $\epsilon > 0$, implying that it is close to the zero vector:

$$\left\| \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) \right\| \leq \epsilon. \tag{2.35}$$

In practice, Equation 2.34 tends not to be used as the update formula, with certain unattractive features, including the commonly expensive requirement to compute the inverse of the Hessian, and the possibility of the Newton update not being a minimising update if the Hessian is not positive definite. Two primary strategies have emerged for numerical optimisation:

❖ *Line search methods*[27: Chapter 3] determine an appropriate direction $\boldsymbol{p}^{(k)}$ along which the updated parameter vector is sourced. After this, an appropriate step length $\alpha^{(k)}$ is determined (typically in an efficient, though not optimal manner), leading to $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha^{(k)} \boldsymbol{p}^{(k)}$

❖ *Trust region methods*[27: Chapter 4] define a radius $\Delta^{(k)} > 0$, and determine the minimum of Equation 2.33 subject to the constraint that $\|\boldsymbol{b}\| \leq \Delta^{(k)}$. This radius is typically programmed to be variable, such that if the quadratic model agrees well with the true value of

the fidelity, it is increased for the next iteration, while the converse is true if there is poor agreement.

A trust region method is applied in this work, and as such further consideration of it will now be made.

Trust Region Methods

The structure of a trust region method is presented in Algorithm 3 (ignoring lines 19–22, which is a custom addition, see Section 2.3.4). An initial radius for the region $\Delta^{(0)}$ is defined, along with a maximum permitted trust radius Δ_{max}, to ensure that excessively adventurous steps do not take place. For each iteration, one then seeks a solution to the sub-problem

$$\boldsymbol{p}^{(k)} = \underset{\boldsymbol{p} \in \mathbb{R}^n}{\arg\min} \, \mathcal{F}\left(\boldsymbol{\theta}^{(k)}\right) + \boldsymbol{p}^{\mathrm{T}} \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) + \tfrac{1}{2}\boldsymbol{p}^{\mathrm{T}} \boldsymbol{H}\left(\boldsymbol{\theta}^{(k)}\right) \boldsymbol{p}$$

$$\text{subject to } \|\boldsymbol{p}\| \le \Delta^{(k)}. \tag{2.36}$$

This sub-problem is not usually minimised exactly, but instead an efficient means of determining a sufficiently good value for $\boldsymbol{p}^{(k)}$ is used. Common approaches include computing the Cauchy point and the Dogleg method[27: Section 4.1], and a truncated conjugate-gradient approach, also known as the Steihaug-Toint (ST) method[27: Algorithm7.2], which is employed in this work. In the ST approach, iterates of the conjugate-gradient method are used, until either they go beyond the trust region, or negative curvature is discovered. **Algorithm in appendix?**

Once a step $\boldsymbol{p}^{(k)}$ is determined, a metric is considered which indicates how effectively the quadratic estimate at the proposed update $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}$ agrees with the true value of the fidelity at this point:

$$\rho^{(k)} = \frac{\mathcal{F}\left(\boldsymbol{\theta}^{(k)}\right) - \mathcal{F}\left(\boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}\right)}{\mathcal{F}_Q\left(\boldsymbol{\theta}^{(k)}\right) - \mathcal{F}_Q\left(\boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}\right)}. \tag{2.37}$$

$\rho^{(k)}$ is the ratio between the actual reduction of the fidelity caused by taking the proposed step, and the predicted reduction based on the quadratic model. If $\rho^{(k)}$ is sufficiently close to 1, the quadratic model being used to generate new iterates is deemed to be acting well enough to warrant accepting the proposed update (lines 14-15). Furthermore, if $\rho^{(k)}$ is *particulary* close to 1, and the proposed update is at the boundary of the trust radius, it is appropriate to enlarge the radius of the trust region for the next iteration in an attempt to increase the rate of convergence (lines 9-10). On the other hand, a small value of $\rho^{(k)}$ implies that the quadratic model reflects the true fidelity poorly, such that the proposed update should be rejected (lines 16-17). As well as this, the trust region's radius should be decreased to ensure the model does behave faithfully. (lines 7-8). The exact thresholds which dictate whether to accept an update, and whether to adjust the trust region radius are customisable. The hard-coded numerical values found in Algorithm 3 are the

values used in this work, and the defaults in NMR-EsPy.

2.3.2   Non-linear programming applied to FID estimation

Returning to the problem of FID estimation, the fidelity $\mathcal{F}\left(\boldsymbol{\theta}\,|\,\boldsymbol{Y}\right) : \mathbb{C}^{N^{(1)}\times\cdots\times N^{(D)}} \times \mathbb{R}^{2(D+2)M} \to$
$\mathbb{R}$ is given by

$$\mathcal{F}\left(\boldsymbol{\theta}\,|\,\boldsymbol{Y}\right) = \left\|\boldsymbol{Y} - \boldsymbol{X}\left(\boldsymbol{\theta}\right)\right\|^2. \tag{2.38}$$

Remark 1. *Prior to estimating the dataset, it is normalised, such that the signal actually under consideration is* $\boldsymbol{Y}/\|\boldsymbol{Y}\|$. *To make the result reflect the actual dataset, the final amplitudes* $\boldsymbol{a}^{(*)}$ *are multiplied by* $\|\boldsymbol{Y}\|$.

The elements of the gradient vector $\boldsymbol{g}\left(\boldsymbol{\theta}\,|\,\boldsymbol{Y}\right) \in \mathbb{R}^{2(D+2)M}$ and the Hessian matrix $\boldsymbol{H}\left(\boldsymbol{\theta}\,|\,\boldsymbol{Y}\right) \in$ $\mathbb{R}^{2(D+2)M\times 2(D+2)M}$ are then derived by taking the first and second partial derivatives of the fidelity with respect to the elements in $\boldsymbol{\theta}$, respectively:

$$\boldsymbol{g}\left[i\right] = -2\Re\left(\left(\boldsymbol{Y} - \boldsymbol{X}\right), \frac{\partial\boldsymbol{X}}{\partial\boldsymbol{\theta}\left[i\right]}\right), \tag{2.39a}$$

$$\boldsymbol{H}\left[i, j\right] = 2\Re\left(\underbrace{\left(\frac{\partial\boldsymbol{X}}{\partial\boldsymbol{\theta}\left[i\right]}, \frac{\partial\boldsymbol{X}}{\partial\boldsymbol{\theta}\left[j\right]}\right)}_{\textcircled{1}} - \underbrace{\left(\left(\boldsymbol{Y} - \boldsymbol{X}\right), \frac{\partial^2\boldsymbol{X}}{\partial\boldsymbol{\theta}\left[i\right]\partial\boldsymbol{\theta}\left[j\right]}\right)}_{\textcircled{2}}\right). \tag{2.39b}$$

$\forall i, j \in \left\{0, \cdots, 2(D+2)M - 1\right\}$. The complete set of first derivatives of a particular element of the model $x := \boldsymbol{X}\left[n^{(1)}, \cdots, n^{(D)}\right]$ (see Equation 2.1b) are as follows $\forall m \in \left\{0, \cdots, M - 1\right\}$:

$$\frac{\partial x}{\partial\boldsymbol{\theta}\left[m\right]} \equiv \frac{\partial x}{\partial\boldsymbol{a}\left[m\right]} = \frac{x}{\boldsymbol{a}\left[m\right]}, \tag{2.40a}$$

$$\frac{\partial x}{\partial\boldsymbol{\theta}\left[m + M\right]} \equiv \frac{\partial x}{\partial\boldsymbol{\phi}\left[m\right]} = \mathrm{i}x, \tag{2.40b}$$

$$\frac{\partial x}{\partial\boldsymbol{\theta}\left[m + (d+1)M\right]} \equiv \frac{\partial x}{\partial\boldsymbol{f}^{(d)}\left[m\right]} = 2\pi\mathrm{i}\Delta_t^{(d)}n^{(d)}x, \tag{2.40c}$$

$$\frac{\partial x}{\partial\boldsymbol{\theta}\left[m + (d+D+1)M\right]} \equiv \frac{\partial x}{\partial\boldsymbol{\eta}^{(d)}\left[m\right]} = -\Delta_t^{(d)}n^{(d)}x. \tag{2.40d}$$

The complete set of the second derivatives are given by

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m\right]^2} \equiv \frac{\partial^2 x}{\partial\boldsymbol{a}\left[m\right]^2} = 0, \tag{2.41a}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m\right]\partial\boldsymbol{\theta}\left[m + M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{a}\left[m\right]\partial\boldsymbol{\phi}\left[m\right]} = \frac{\mathrm{i}x}{\boldsymbol{a}\left[m\right]}, \tag{2.41b}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m\right]\partial\boldsymbol{\theta}\left[m + (d+1)M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{a}\left[m\right]\partial\boldsymbol{f}^{(d)}\left[m\right]} = \frac{2\pi\mathrm{i}\Delta_t^{(d)}n^{(d)}x}{\boldsymbol{a}\left[m\right]}, \tag{2.41c}$$

dimensions	# 1$^{\text{st}}$ derivatives	# 2$^{\text{nd}}$ derivatives
1	$4MN^{(1)}$	$9MN^{(1)}$
2	$6MN^{(1)}N^{(2)}$	$20MN^{(1)}N^{(2)}$
3	$8MN^{(1)}N^{(2)}N^{(3)}$	$35MN^{(1)}N^{(2)}N^{(3)}$
D	$2(1+D)MN^{(1)}\cdots N^{(D)}$	$\left(\left(1+D\right)\left(2\left(1+D\right)+1\right)-1\right)MN^{(1)}\cdots N^{(D)}$

TABLE 2.1: The number of first and second derivatives that are necessary to compute both the gradient vector and Hessian matrix of the fidelity for 1- 2- and 3-dimensional datasets, as well as a general D-dimensional dataset.

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m\right]\partial\boldsymbol{\theta}\left[m+(d+D+1)M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{a}\left[m\right]\partial\boldsymbol{\eta}^{(d)}\left[m\right]} = \frac{-\Delta_t^{(d)}n^{(d)}x}{\boldsymbol{a}\left[m\right]}, \tag{2.41d}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+M\right]^2} \equiv \frac{\partial^2 x}{\partial\boldsymbol{\phi}\left[m\right]^2} = -x, \tag{2.41e}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+M\right]\partial\boldsymbol{\theta}\left[m+(d+1)M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{\phi}\left[m\right]\partial\boldsymbol{f}^{(d)}\left[m\right]} = -2\pi\Delta_t^{(d)}n^{(d)}x, \tag{2.41f}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+M\right]\partial\boldsymbol{\theta}\left[m+(d+D+1)M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{\phi}\left[m\right]\partial\boldsymbol{\eta}^{(d)}\left[m\right]} = -\mathrm{i}\Delta_t^{(d)}n^{(d)}x, \tag{2.41g}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+(d+1)M\right]^2} \equiv \frac{\partial^2 x}{\partial\boldsymbol{f}^{(d)}\left[m\right]^2} = -4\pi^2\left(\Delta_t^{(d)}n^{(d)}\right)^2 x, \tag{2.41h}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+(d+1)M\right]\partial\boldsymbol{\theta}\left[m+(d+D+1)M\right]} \equiv \frac{\partial^2 x}{\partial\boldsymbol{f}^{(d)}\left[m\right]\partial\boldsymbol{\eta}^{(d)}\left[m\right]} = -2\pi\mathrm{i}\left(\Delta_t^{(d)}n^{(d)}\right)^2 x, \tag{2.41i}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[m+(d+D+1)M\right]^2} \equiv \frac{\partial^2 x}{\partial\boldsymbol{\eta}^{(d)}\left[m\right]^2} = \left(\Delta_t^{(d)}n^{(d)}\right)^2 x, \tag{2.41j}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[i\right]\partial\boldsymbol{\theta}\left[j\right]} = \frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[j\right]\partial\boldsymbol{\theta}\left[i\right]}, \tag{2.41k}$$

$$\frac{\partial^2 x}{\partial\boldsymbol{\theta}\left[i\right]\partial\boldsymbol{\theta}\left[j\right]} = 0 \ \text{ if not specified above.} \tag{2.41l}$$

Equation 2.41l indicates that any second derivative with respect to two parameters which do not belong to the same oscillator will always be 0. This, along with the symmetrical nature of the second derivatives (Equation 2.41k) drastically reduces the required number of second derivatives to compute, from $4(1+D)^2M^2$ per data-point to $(1+D)(3+2D)M$. Finally, Equation 2.41a indicates that another M second derivatives do not need to be computed. See Table 2.1 the total number of derivatives that need to be computed for 1D, 2D, and 3D signals.

2.3.3   Approximating the Hessian

Despite many of the model second derivatives being 0, computation of those that are not zero, and subsequently forming the Hessian matrix, is the most computationally expensive part of the optimisation. There are a large number of optimisation problems where this too is the case, and as such there is considerable precedent for improving the efficiency of optimisation algorithms by generating less expensive approximations of the Hessian. Examples include the Gauss-Newton (GN) method and Levenberg-Marquardt (LM) algorithm, which are specifically for residual sum-of-squares problems[27: Chapter 10], and quasi-Newton methods such as the BFGS method[27: Chapter 6].

The GN and LM approaches replace the true Hessian matrix at each iteration with the following expression:

$$\boldsymbol{H}\left[i,j\right] \approx 2\Re\left\langle \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\theta}\left[i\right]}, \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\theta}\left[j\right]}\right\rangle, \tag{2.42}$$

i.e. term ② in Equation 2.39b involving the second derivatives is neglected. The only thing that needs to be generated is the Jacobian $\boldsymbol{J} = \partial\boldsymbol{X}/\partial\boldsymbol{\theta}$, an array containing the $2\left(1+D\right)MN^{(1)}\cdots N^{(D)}$ first derivatives of $\boldsymbol{X}$ (Table 2.1). This brings a very large reduction in the computational cost, as no extra derivatives need to be computed for the Hessian at all, since the Jacobian is already required for generating the gradient vector (Equation 2.39a). In situations where the residuals between the data and model are small, term ① will tend to dominate term ②, and as such these methods often enjoy a convergence rate close to that of Newton's method close to local minima. Despite this, by invoking this approximation, the rate of convergence (i.e. the number of iterations required to reach $\boldsymbol{\theta}^{(*)}$ tends to be adversely affected. (See Section 2.3.6 for an example of this phenomenon)

2.3.4   Phase Variance Minimisation

While numerical optimisation procedures can return estimates $\boldsymbol{\theta}^{(*)}$ which can achieve highly accurate reconstructions of the original FID (i.e. $\boldsymbol{Y}$ and $\boldsymbol{X}\left(\boldsymbol{\theta}^{(*)}\right)$ are in close agreement), the estimate won't necessarily provide a faithful description of the physical phenomenon that has given rise to the data. For the purposes of FID estimation, the goal is to ensure that each resonance contributing to the FID is described by a single oscillator. In scenarios where the model order M used is greater than the true number of resonances, over-fitting of the data will occur, leading to spurious features in the estimation result, such as single resonances being fit by multiple oscillators and/or noise components being fit. To overcome this problem, it is desirable to include known information about the signal into the optimisation routine as a means of guiding the parameter vector to a more appropriate final value. One particular means of achieving this which has been to found to be effective is the incorporation of the variance of oscillator phases into the fidelity,

such that it becomes

$$\mathcal{F}_\phi(\boldsymbol{\theta} \,|\, \boldsymbol{Y}) = \|\boldsymbol{Y} - \boldsymbol{X}(\boldsymbol{\theta})\|^2 + \mathrm{Var}_\circ(\boldsymbol{\phi}), \tag{2.43}$$

where $\mathrm{Var}_\circ(\boldsymbol{\phi}) := \mathrm{Var}_\circ(\boldsymbol{\theta}[M:2M])$ is the *circular* variance of the oscillator phases (*vide infra*).

Remark 2. *The inclusion of the phase variance into the fidelity is one of the motivating reasons for normalising the data prior to estimation (see Remark 1).* $\mathrm{Var}_\circ(\boldsymbol{\phi})$ *is constrained to the interval* $[0, 1]$. *If the data were not normalised, it is likely that* $\|\boldsymbol{Y} - \boldsymbol{X}\|^2$ *would dominate* $\mathrm{Var}_\circ(\boldsymbol{\phi})$ *in Equation 2.43, such that the influence of the phase variance would be negligible.*

For the phase variance to act effectively, it is necessary to apply phase-correction to the data prior to estimation, as the assumption that the phases of all contributing resonances are equal is typically valid. **Mention phase-variation of multiplet peaks in experiments where J-coupling evolves prior to acquisition?** T_2 **for example.** The inclusion of phase variance has also been found to be effective at purging excessive oscillators that may be present in the initial guess $\boldsymbol{\theta}^{(0)}$, which be thought of in the following way:

(i) Assume that the initial guess contains more oscillators than the true number of resonances. In this circumstance, it is common to find that true resonances are fit with an acceptable oscillator, whilst extra oscillators exist with spurious phases on account of over-fitting.

(ii) Unconstrained numerical optimisation is now run, with the fidelity given by Equation 2.43. The phase variance will force oscillators with phases that are significantly different to the majority of oscillators to drastically alter their phases to match them. For some/all of these oscillators, it may be the case that the optimiser drives these to acquire a negative amplitude, such that they act as if they have a phase of π while ensuring that $\mathrm{Var}_\circ(\boldsymbol{\phi})$ is small.

(iii) This provides a criterion for detecting oscillators which are likely to be excessive. Such oscillators can be purged from $\boldsymbol{\theta}$ by periodically checking whether any amplitudes (i.e. $\boldsymbol{\theta}[:M]$) have become negative, purging these, and re-starting the optimisation.

(iv) The numerical optimisation routine is re-started each time oscillators are purged. Termination is achieved once the routine converges and no negative-amplitude oscillators exist in $\boldsymbol{\theta}$.

Circular Variance

Oscillator phases are an example of a *circular variable*, in that all phases are wrapped within an interval of width 2π, for example $(-\pi, \pi]$. Given an unconstrained (unwrapped) phase $\widetilde{\phi} \in \mathbb{R}$, the corresponding wrapped phase is given by

$$\phi = \left(\left(\widetilde{\phi}[m] + \pi\right) \bmod 2\pi\right) - \pi. \tag{2.44}$$

This makes the conventional (linear) definition of variance, given by

$$\text{Var}_{\shortmid}\left(\boldsymbol{\phi}\right) = \frac{1}{M} \sum_{m=0}^{M-1} \left(\boldsymbol{\phi}\left[m\right] - \mu\left(\boldsymbol{\phi}\right)\right)^2, \tag{2.45a}$$

$$\mu\left(\boldsymbol{\phi}\right) = \frac{1}{M} \sum_{m=0}^{M-1} \boldsymbol{\phi}\left[m\right], \tag{2.45b}$$

unsuitable for phases. Consider as a simple example a scenario where there are two oscillators with phases $\widetilde{\boldsymbol{\phi}} = \left[\pi + \delta, \pi - \delta\right]^{\text{T}}$ for some small δ.[†] The phase variance is expected to be small as these two phases are similar. However, with the inclusion of wrapping (i.e. application of Equation 2.44), these phases would actually be set to $\boldsymbol{\phi} = \left[-\pi + \delta, \pi - \delta\right]^{\text{T}}$, and the conventional definition of phase variance would be large. It is therefore apparent that a definition of variance which accounts for the periodicity of the phases is needed. The *circular variance* is given by[34: Chapter 3]

$$\left[0, 1\right] \ni \text{Var}_{\circ}\left(\boldsymbol{\phi}\right) = 1 - \frac{R}{M}, \tag{2.46a}$$

$$R = \sqrt{c_{\Sigma}^2 + s_{\Sigma}^2}, \tag{2.46b}$$

$$c_{\Sigma} = \sum_{m=0}^{M-1} \cos \boldsymbol{\phi}\left[m\right], \tag{2.46c}$$

$$s_{\Sigma} = \sum_{m=0}^{M-1} \sin \boldsymbol{\phi}\left[m\right]. \tag{2.46d}$$

R is the length of the resultant vector produced by summing M unit vectors with the angles given by $\boldsymbol{\phi}$. In the case that all the vectors have the same angle, $R = M$, leading to the variance being 0 as expected. At the other extreme, with M vectors uniformly separated about the unit circle (with an angle $2\pi/M-1$ between all pairs of adjacent vectors), the vectors will perfectly cancel, leading to $R = 0$. In this case, the maximum variance of 1 is obtained.

The first and second derivatives of the circular variance are required for the computation of the gradient vector and Hessian matrix. These are given by

$$\frac{\partial \text{Var}_{\circ}\left(\boldsymbol{\phi}\right)}{\partial \boldsymbol{\theta}\left[i\right]} = \begin{cases} \frac{1}{MR}\left(c_{\Sigma} \sin \boldsymbol{\phi}\left[i - M\right] - s_{\Sigma} \cos \boldsymbol{\phi}\left[i - M\right]\right) & M \leq i < 2M \\ 0 & \text{otherwise} \end{cases} \tag{2.47a}$$

[†]The fact that the circular phase variance was necessary rather than the linear phase variance came rather late during my PhD. It was specifically realised while the estimation of inversion recovery datasets was being considered (see Section **InvRec reference**), as the initial increment of these datasets features resonances with phases close to π, such that the linear phase variance that was originally programmed in NMR-EsPy was giving far larger values than anticipated.

$$
\frac{\partial^2 \text{Var}_\circ (\boldsymbol{\phi})}{\partial \boldsymbol{\theta}[i] \partial \boldsymbol{\theta}[j]} =
\begin{cases}
\frac{1}{RM} \left[\frac{1}{R^2} \left(c_\Sigma \sin \boldsymbol{\phi}[i-M] - s_\Sigma \cos \boldsymbol{\phi}[i-M] \right)^2 \right. \\
\quad \left. + c_\Sigma \cos \boldsymbol{\phi}[i-M] + s_\Sigma \sin \boldsymbol{\phi}[i-M] - 1 \right] & M \le i,j < 2M, i = j \\[2pt]
\frac{1}{RM} \left[\frac{1}{R^2} \left(c_\Sigma \sin \boldsymbol{\phi}[i-M] - s_\Sigma \cos \boldsymbol{\phi}[i-M] \right) \right. \\
\quad \times \left(c_\Sigma \sin \boldsymbol{\phi}[j-M] - s_\Sigma \cos \boldsymbol{\phi}[j-M] \right) & M \le i,j < 2M, i \ne j \\
\quad \left. - \cos \left(\boldsymbol{\phi}[i-M] - \boldsymbol{\phi}[j-M] \right) \right] \\[2pt]
0 & \text{otherwise}
\end{cases}
\tag{2.47b}
$$

2.3.5 Estimation Errors

The errors associated with NLP can be extracted using information about the Hessian at $\boldsymbol{\theta}^{(*)}$. The vector of standard errors is equivalent to the square root of the diagonal elements of the inverse of the observed Fisher Information matrix for $\boldsymbol{\theta}^{(*)}$ [35: Section 2.7]:

$$
\boldsymbol{\epsilon} \left(\boldsymbol{\theta}^{(*)} \right) = \sqrt{ \text{diag} \left(\boldsymbol{I} \left(\boldsymbol{\theta}^{(*)} \right)^{-1} \right) },
\tag{2.48}
$$

where the observed Fisher Information matrix contains the negative partial second derivatives of the log-likelihood with respect to $\boldsymbol{\theta}$:

$$
\boldsymbol{I} (\boldsymbol{\theta}) [i,j] = - \frac{\partial^2 \ell (\boldsymbol{\theta} \mid \boldsymbol{Y})}{\partial \boldsymbol{\theta}[i] \partial \boldsymbol{\theta}[j]}.
\tag{2.49}
$$

Recalling the assumed form of the log-likelihood (Equation 2.5), elements of $\boldsymbol{I} (\boldsymbol{\theta})$ are

$$
\boldsymbol{I} (\boldsymbol{\theta}) [i,j] = - \frac{1}{\sigma^2} \Re \left(\left\langle \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\theta}[i]}, \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\theta}[j]} \right\rangle - \left\langle (\boldsymbol{Y} - \boldsymbol{X}), \frac{\partial^2 \boldsymbol{X}}{\partial \boldsymbol{\theta}[i] \partial \boldsymbol{\theta}[j]} \right\rangle \right),
\tag{2.50}
$$

which very closely resembles the Hessian of $\boldsymbol{\theta}$:

$$
\boldsymbol{I} (\boldsymbol{\theta}) [i,j] = \frac{1}{2\sigma^2} \left(\boldsymbol{H} (\boldsymbol{\theta}) [i,j] \right).
\tag{2.51}
$$

The standard errors therefore take the form

$$
\boldsymbol{\epsilon} \left(\boldsymbol{\theta}^{(*)} \right) = \sqrt{ 2\sigma^2 \, \text{diag} \left(\left[\boldsymbol{H} \left(\boldsymbol{\theta}^{(*)} \right) \right]^{-1} \right) }.
\tag{2.52}
$$

Considering that the mean and variance of the noise $\boldsymbol{W}$ are 0 and $2\sigma^2$, respectively:

$$
2\sigma^2 = \frac{1}{\mathfrak{N} - 1} \sum_{n^{(1)}=0}^{N^{(1)}-1} \cdots \sum_{n^{(D)}=0}^{N^{(D)}-1} \left| \boldsymbol{W} \left[n^{(1)}, \cdots, n^{(D)} \right] \right|^2 = \frac{1}{\mathfrak{N} - 1} \left\| \boldsymbol{Y} - \boldsymbol{X} \left(\boldsymbol{\theta}^{(*)} \right) \right\|^2,
\tag{2.53}
$$

so that finally a useable expression for the standard errors is arrived at:

$$\epsilon\left(\boldsymbol{\theta}^{(*)}\right) = \sqrt{\frac{\mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right)\operatorname{diag}\left(\left[\boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right)\right]^{-1}\right)}{\mathfrak{N}-1}} \tag{2.54}$$

Algorithm 3 Nonlinear programming routine employed in NMR-EsPy. This makes use of Algorithms 4.1 & 7.2 in [27], with a extra check inserted to deal with any negative-amplitude oscillators which may be generated as the routine evolves.

1: **procedure** NLP($\boldsymbol{Y} \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}, \boldsymbol{\theta}^{(0)} \in \mathbb{R}^{2(D+1)M}$)
2:      $\Delta^{(0)} \leftarrow {}^{1}\!/_{10}\left\|\boldsymbol{g}\left(\boldsymbol{\theta}^{(0)} \mid \boldsymbol{Y}\right)\right\|$;
3:      $\Delta_{\max} \leftarrow 16\Delta^{(0)}$;
4:      **for** $k = 0, 1, \cdots$ **do**
5:          $\boldsymbol{p}^{(k)} \leftarrow \textsc{SteihaugToint}\left(\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}, \Delta^{(k)}\right)$;   ▷ See Algorithm **??**
6:          $\rho^{(k)} \leftarrow \frac{\mathcal{F}_{\phi}\left(\boldsymbol{\theta}^{(k)}\right) - \mathcal{F}_{\phi}\left(\boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}\right)}{\mathcal{F}_{\phi Q}\left(\boldsymbol{\theta}^{(k)}\right) - \mathcal{F}_{\phi Q}\left(\boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}\right)}$;
7:          **if** $\rho_k < {}^{1}\!/_{4}$ **then**
8:              $\Delta^{(k+1)} \leftarrow {}^{1}\!/_{4}\Delta^{(k)}$;
9:          **else if** $\rho_k > {}^{3}\!/_{4}$ **and** $\left\|\boldsymbol{p}^{(k)}\right\| = \Delta^{(k)}$ **then**
10:         $\Delta^{(k+1)} \leftarrow \min\left(2\Delta^{(k)}, \Delta_{\max}\right)$;
11:          **else**
12:         $\Delta^{(k+1)} \leftarrow \Delta^{(k)}$;
13:          **end if**
14:          **if** $\rho^{(k)} > {}^{3}\!/_{20}$ **then**
15:              $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} + \boldsymbol{p}^{(k)}$;
16:          **else**
17:              $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)}$;
18:          **end if**
19:          **if** $k \bmod 25 = 0$ **and** $\boldsymbol{\theta}^{(k+1)}$ contains negative amplitudes **then**
20:              $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}^{(k+1)}$ with negative-amplitude oscillators removed;
21:              $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)} \leftarrow \text{NLP}\left(\boldsymbol{Y}, \boldsymbol{\theta}^{(0)}\right)$;
22:          **end if**
23:          **if** $\left\|\boldsymbol{g}\left(\boldsymbol{\theta}^{(k+1)}\right)\right\| < 10^{-8}$ **then**
24:              **break**;
25:          **end if**
26:      **end for**
27:      $\boldsymbol{\theta}^{(*)} \leftarrow \boldsymbol{\theta}^{(k+1)}$
28:      $\boldsymbol{\epsilon}^{(*)} \leftarrow \sqrt{\frac{\mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right)\operatorname{diag}\left(\left[\boldsymbol{H}\left(\boldsymbol{\theta}^{(*)}\right)\right]^{-1}\right)}{\left(N^{(1)}\cdots N^{(D)}\right)-1}}$
29:      **return** $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)}$;
30: **end procedure**

2.3.6   Visualisation of a simple example

Figure 2.1 provides a visualisation of numerical optimisation applied on a simulated FID comprising a single resonance. The FID was constructed using Equation 2.1a with $D = 1$, $M = 1$, $N^{(1)} = 64$, $f_{\text{sw}}^{(1)} = 5.2\,\text{Hz}$ ($\Delta_t^{(1)} \approx 0.192\,\text{s}^{-1}$), and $f_{\text{off}}^{(1)} = 0\,\text{Hz}$. The resonance was parameterised by $\boldsymbol{\theta} \in \mathbb{R}^4$ comprising $a = 1$, $\phi = 0\,\text{rad}$, $f^{(1)} = 1\,\text{Hz}$, $\eta^{(1)} = 0.2\,\text{s}^{-1}$. White Gaussian noise was
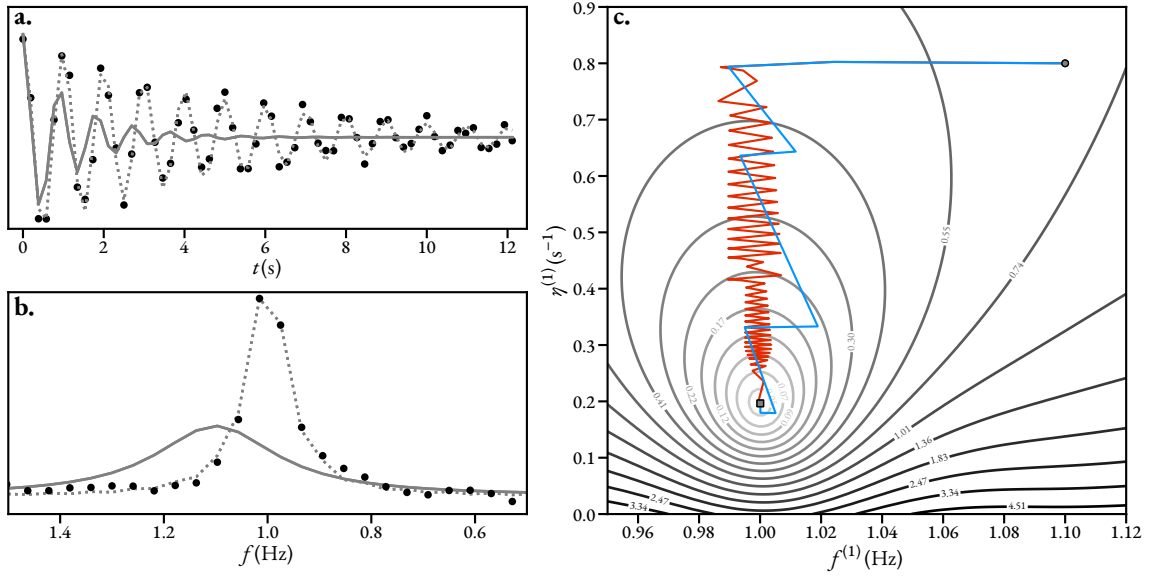
FIGURE 2.1: A visualisation of the trajectory of a 2-parameter optimisation involving a simulated FID comprising a single resonance. **a.** & **b.** Representations of the signal in the time domain and Fourier domain, respectively. Black dots: the synthetic signal Y. Solid grey line: the model generated using the initial guess $X\left(\theta^{(0)}\right)$. Dotted grey line: the model generated using the optimised result, $X\left(\theta^{(*)}\right)$. **c.** A contour plot of the fidelity. Blue line: the trajectory of the parameter vector with the true Hessian matrix used in computing each update. Red line: the analogous trajectory using the Hessian approximation in place of the true Hessian.

added to the FID to give it a signal-to-noise ratio (SNR) of approximately 10 dB. As the visualisation of 5D space is beyond the scope of this work, only two parameters, the frequency and damping factor were optimised from an initial guess, with the amplitude and phase being fixed to their true values. The initial guess comprised a freqency of 1.1 Hz, and a damping factor of 0.8 s^{-1}, with the solid grey lines in panels a & b denoting the model generated using the initial guess in the time- and Fourier-domains, respectively. $\theta^{(0)}$ was subjected to NLP twice. In the first instance, the exact Hessian matrix (Equation 2.39b) was used in order to compute each update step, while in the second the Hessian approximation given by Equation 2.42 was used. The initial radius of the trust region was set to $^1/_{10}$ of the gradient norm (≈ 0.3), which has a precedent in the literature[36]. The trajectories of the parameter vector are denoted as coloured lines in panel c. In both cases, the NLP routine successfully generated a result $\theta^{(*)}$ which agreed with the true frequency and damping factor used to construct the FID. However, it is clear that using the true Hessian matrix (blue) in this example realised a far better rate of convergence compared with the approximated analogue (red), which exhibited "zig-zagging", a phenomenon often seen in gradient search methods. 14 iterations were required to reach the convergence criterion $\epsilon \leq 10^{-8}$ when the true Hessian was used, while 81 were required for the approximated case. While an anecdotal example, this highlights that use of the true Hessian matrix tends to allow a better rate of convergence, however for FIDs comprising many resonances and far more points, the approximated form often

requires a shorter time to converge overall, as the cost of computing the Hessian second derivatives dominates the rate of convergence.

2.4   Profiling the MPM and NLP

Producing $\theta^{(*)}$ via generation of the initial guess using the MPM, and subjecting this to numerical optimisation involves operations which can be computationally demanding, both in terms of the amount of work done by the Central Processing Unit (CPU), and space requirements (i.e. the amount of Random Access Memory (RAM) required to store all the required information as the routine runs). For the MPM, the most demanding aspect is SVD calculations, which for numerical optimisation, it is generation of the Hessian matrix. Detailed accounts of the computational complexity of the MPM and MMEMPM has been described previously. However, it is useful to consider what the actual running times of these routines are. This is particularly useful, as a lot of the accounts on running the MPM are from decades before this work, and so the speed at which the estimation routine can be expected to take will have decreased a lot thanks to improvements in processing power. As an example, the account by Pines a coworkers from 1997 states that a signal with $N^{(1)}$ = 1024 would take about 4.5 min to be processed by the Minimum Description Length (MDL) and MPM, using a 100 MHz CPU[37]. On the system used for all results generated for this work (see Remark 3), an equivalent computation takes about 100 ms.

Remark 3. *All results generated in this work were acquired using a workstation featuring a Intel® Core™ i9-10900X CPU @ 3.7 GHz, and 32 GiB of RAM.*

To gain a more concrete understanding of**...**

- ❖ **1D MPM: created signal with 10 oscs, varied $N^{(1)}$ from** 512 $\to$ 8192 **in steps of 512.** $L^{(1)} = N^{(1)}/3$

- ❖ **Total time very closely agreed with tiem to compute SVD of H_Y, especially for larger $N^{(1)}$: cubic dependence found.**

- ❖ **Panel a in plot: $N^{(1)}$ vs wall clock time, with cube root scale.**

- ❖ **Linear fit to circular points is shown (for square points (small $N^{(1)}$), other parts of the algorithm (notably EVD of VM1+VM2) had an appreciable impact on time.**

- ❖ **Peak memory requirement showed perfect quadratic relationship. Why?**

- ❖ **Extrapolated linear fit shows that unless very big signals considered, modern PCs should be able to cope.**
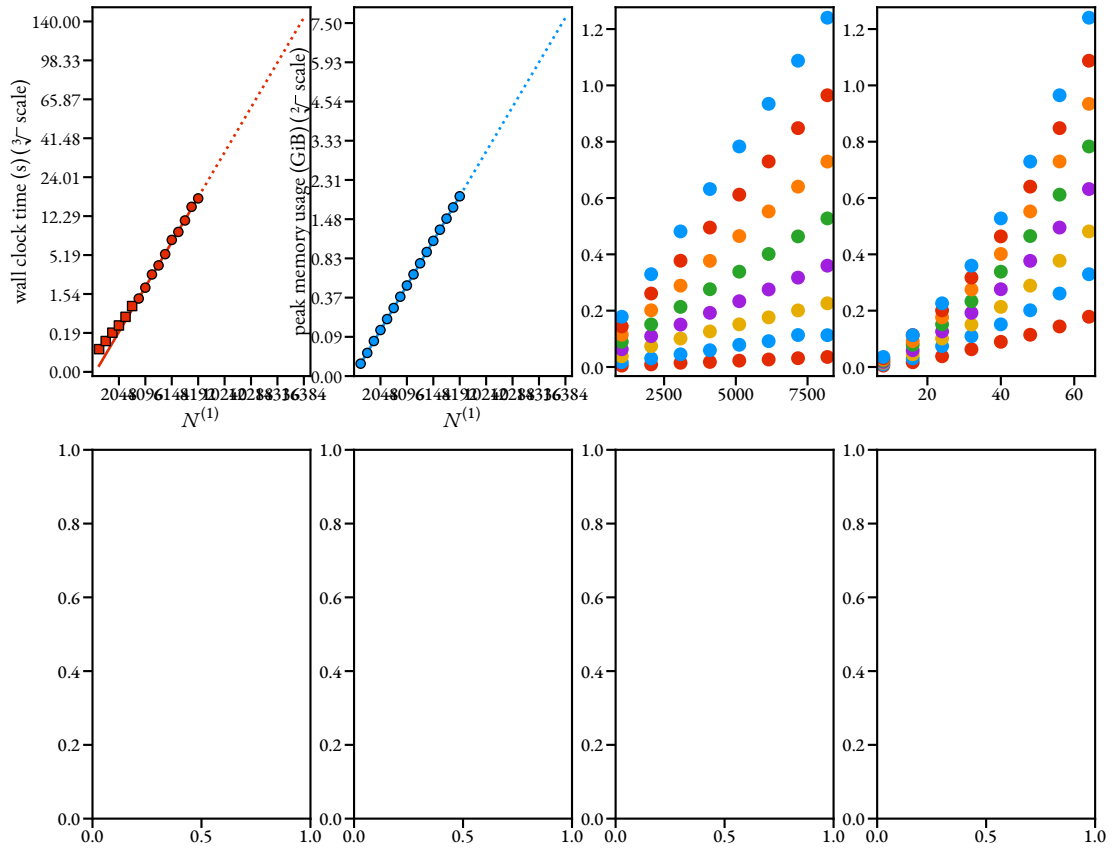
FIGURE 2.2: TODO

2.5   Frequency Filtration

The previous section provides motivation for for finding ways to reduce the number of points in the signal and also the number of oscillators that the signal contains. This has led to work on a procedure for generated frequency-filtered "sub-FIDs" from the original data. A detailed description of the filtering procedure is presented in this section.

In brief, the filtering consists of applying a Fourier transform (FT) on the FID, applying a band-pass filter to the spectral data to discard parts of not being considered, and returning the spectrum back to the time-domain by an Inverse Fourier transform (IFT). For effective filtering, more careful consideration of certain aspects is required on top of this.

For a filtered FID to be faithfully described by the model of a summation of damped complex sinusoids, it is necessary that the spectral peaks all lie entirely within the filter region[‡]. For absorptive Lorentzians, due to their characteristically narrow linewidths, this is typically straightforward. However, for dispersive Lorentzians, this is far more challenging. For this reason, generating a spectrum in which only the real component is retained is desired. Assuming the data has been phase-corrected, this will produce a spectrum comprising only absorptive Lorentzians. The virtual echo (VE) has been employed here, which has found application in the field of compressed sensing NMR[39, 40, 41]. This is a signal with double the size as the original FID, with the key characteristic that its FT has a real component which is equivalent to its counterpart derived from an unaltered FID (except it has double the points), and an imaginary component of 0s.

2.5.1   The virtual echo

Section needs work! Assuming that a 1D FID has been phase-corrected, such that $\boldsymbol{\phi} = \mathbf{0} \in \mathbb{R}^M$, it can be denoted as

$$\boldsymbol{y} = \boldsymbol{\gamma} \odot \left(\boldsymbol{c}^{(1)} + \mathrm{i}\boldsymbol{s}^{(1)} \right) + \boldsymbol{w}, \tag{2.55a}$$

$$\boldsymbol{\gamma} = \sum_m \boldsymbol{a}\left[m\right] \exp\left(-\boldsymbol{\eta}^{(1)}\left[m\right] \boldsymbol{n}^{(1)} \Delta_t^{(1)} \right), \tag{2.55b}$$

$$\boldsymbol{c}^{(1)}/\boldsymbol{s}^{(1)} = \sum_m \cos/\sin\left(2\pi \boldsymbol{f}^{(1)}\left[m\right] \boldsymbol{n}^{(1)} \Delta_t^{(1)} \right), \tag{2.55c}$$

where $\boldsymbol{n}^{(1)} = \left[0, 1, \cdots, N^{(1)} - 1 \right]^{\mathrm{T}}$. The frequency-dependence has been decomposed into its real and imaginary components. With this in mind, a conjugate pair of signals are defined:

$$\boldsymbol{\psi}_{\pm} = \Re\left(\boldsymbol{y} \right) \pm \mathrm{i}\Im\left(\boldsymbol{y} \right) \tag{2.56}$$

[‡]Lorentzian lineshapes tend to, but don't reach zero, as the distance from the maximum tends to ∞[38]. However, as long as a sufficiently wide filtering is employed, the regions of the Lorentzian which do not pass through the filter can be assumed to be negligible.

Two vectors $\{t_1, t_2\} \in \mathbb{C}^{2N^{(1)}}$ are constructed using the conjugate pair:

- t_1 is given by ψ_+ padded with zeros from below:

$$t_1 = \begin{bmatrix} \psi_+ \\ \mathbf{0} \in \mathbb{C}^{N^{(1)}} \end{bmatrix}, \tag{2.57}$$

- t_2 is given by ψ_- with its elements in reversed order ($\cdot^{\leftrightarrow^{(1)}}$), padded with zeros from above, and finally subjected to a right circular shift by one element ($\cdot^{\circlearrowright^{(1)}}$):

$$t_2 = \begin{bmatrix} \mathbf{0} \in \mathbb{C}^{N^{(1)}} \\ \psi_-^{\leftrightarrow^{(1)}} \end{bmatrix}^{\circlearrowright^{(1)}}, \tag{2.58}$$

The VE y_{ve} is then given by $t_1 + t_2$, with the first element divided by 2, which is equivalent to

$$y_{\mathrm{ve}} = \begin{bmatrix} \Re\left(y[0]\right) & y[1] & \cdots & y\left[N^{(1)} - 1\right] & 0 & y\left[N^{(1)} - 1\right]^* & \cdots & y[1]^* \end{bmatrix}^{\mathrm{T}}. \tag{2.59}$$

A signal of this form is illustrated in panel a of Figure 2.3. As eluded to already, the FT of y_{ve} produces a spectrum s_{ve} such that $\Im\left(s_{\mathrm{ve}}\right) = \mathbf{0}$, with $\Re\left(s_{\mathrm{ve}}\right)$ featuring absorption Lorentzian peaks (panel b of Figure 2.3).

The VE concept can be generalised to any number of dimensions, assuming that a pair of amplitude-modulated signals exist for each indirect-dimension, thus requiring a set 2^{D-1} signals for a D-dimensional dataset. For the 2D case, there is a pair of signals $\{Y_{\cos}, Y_{\sin}\}$, given by (1.14) with $D = 2$, and $\zeta = \{\cos(\cdot), \sin(\cdot)\}$, taking the forms

$$Y_{\cos} = \Gamma \odot C^{(1)} \odot \left(C^{(2)} + iS^{(2)}\right) \tag{2.60a}$$

$$Y_{\sin} = \Gamma \odot S^{(1)} \odot \left(C^{(2)} + iS^{(2)}\right) \tag{2.60b}$$

Four matrices are then constructed of the form

$$\Psi_{\pm\pm} = \left(\Re\left(Y_{\cos}\right) \pm i\Im\left(Y_{\cos}\right)\right)\left(\Re\left(Y_{\sin}\right) \pm i\Im\left(Y_{\sin}\right)\right) \tag{2.61}$$

$$T_1 = \begin{bmatrix} \Psi_{++} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{2.62a}$$
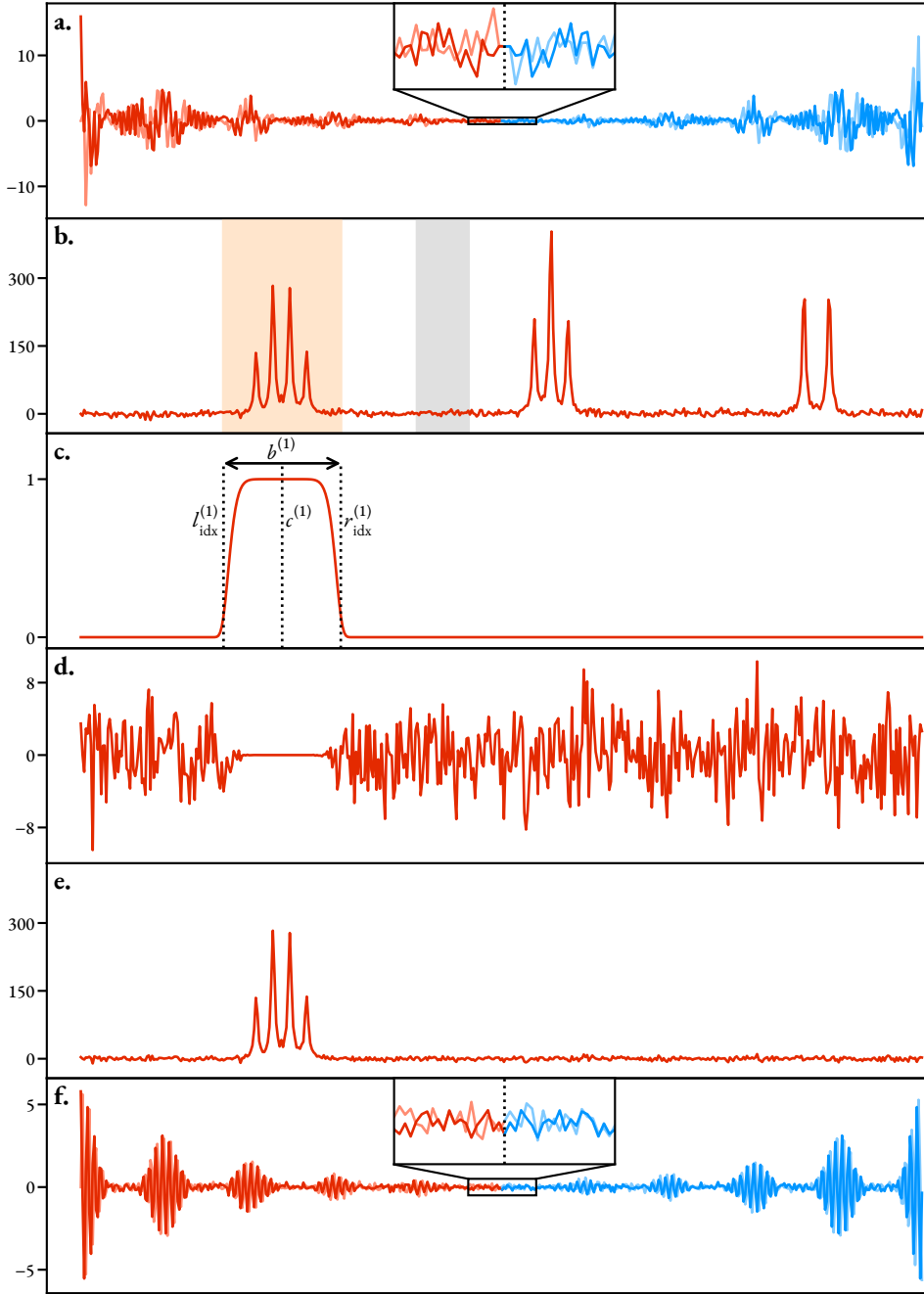
FIGURE 2.3: An illustration of the filtering procedure applied to a 1D FID. **a.** A VE $\boldsymbol{y}_{\text{ve}}$, with the first and last $N^{(1)}$ points coloured red and blue, respectively. The middle of the VE is magnified to highlight its conjugate symmetry. **b.** The FT of the VE, $\boldsymbol{s}_{\text{ve}}$. The region of interest (orange) and noise region (grey) are denoted. **c.** A super-Gaussian function used as a band-pass filter, $\boldsymbol{g}$. **d.** AWGN vector to be added to the filtered spectrum. The magnitude of the signal at each point is dependent on the corresponding super-Gaussian value. **e.** The filtered spectrum $\widetilde{\boldsymbol{s}}_{\text{ve}}$, formed by applying the super-Gaussian filter, and adding the noise vector. **f.** The IFT of the filtered spectrum, $\widetilde{\boldsymbol{y}}_{\text{ve}}$, from which the final filtered signal $\widetilde{\boldsymbol{y}}$ is obtained by extracting the first $N^{(1)}$ points.

$$T_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \overset{\leftrightarrow}{\boldsymbol{\Psi}}_{-+}^{(1)} & \mathbf{0} \end{bmatrix}^{\circlearrowleft (1)}, \tag{2.62b}$$

$$T_3 = \begin{bmatrix} \mathbf{0} & \overset{\leftrightarrow}{\boldsymbol{\Psi}}_{+-}^{(2)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}^{\circlearrowleft (2)}, \tag{2.62c}$$

$$T_4 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \overset{\leftrightarrow}{\boldsymbol{\Psi}}_{--}^{(1,2)} \end{bmatrix}^{\circlearrowleft (1,2)}. \tag{2.62d}$$

Appendix for explicit forms of the matrices T_{1-4} The virtual echo is then given by $\boldsymbol{Y}_{\text{ve}} = \sum_{i=1}^{4} \boldsymbol{T}_i$, with the first row and column divided by two.

2.5.2 The filtering process

Having constructed a virtual echo $\boldsymbol{Y}_{\text{ve}} \in \mathbb{C}^{2N^{(1)} \times \cdots \times 2N^{(D)}}$, a spectrum with absorption Lorentzians is produced with $\boldsymbol{S}_{\text{ve}} = \text{FT}(\boldsymbol{Y}_{\text{ve}})$. To filter the spectrum, it is subjected to element-wise multiplication with a super-Gaussian function. The super-Gaussian is defined by a central index $c^{(d)} \in \mathbb{R} : 0 < c^{(d)} < 2N^{(d)}$ and a bandwidth $b^{(d)} \in \mathbb{N} : b^{(d)} < 2N^{(d)}$ in each dimension, with the bandwidth denoting how many points the filter covers.

$$G = \bigotimes_{d=1}^{D} g^{(d)}, \tag{2.63a}$$

$$g^{(d)}\left[n^{(d)}\right] = \exp\left(-2^{p+1}\left(\frac{n^{(d)} - c^{(d)}}{b^{(d)}}\right)^p\right) \forall n^{(d)} \in \{0, \cdots, 2N^{(d)} - 1\}. \tag{2.63b}$$

An example of a 1D super-Gaussian is given in panel c of Figure 2.3. The scalar $p \in \mathbb{R}_{>0}$ dictates the steepness of the filter at the boundaries, with the function becoming more "box-like" as it increases. It is set to 40 in this work and as the default in NMR-EsPy. Application of the super-Gaussian filter to $\boldsymbol{S}_{\text{ve}}$ would lead to large sections of the filtered spectrum being 0. This has an undesired impact on the MDL, as noise that has passed through filter (i.e. the noise inside the region of interest) will now seem to resemble true signal, as its amplitude is infinitely greater than the zeroed regions. A massive over-estimation of model order therefore result. In order to obtain better results from the model order selection, an array of synthetic AWGN is added to the filtered spectrum. To achieve this, a region in $\boldsymbol{S}_{\text{ve}}$ is specified which contains no discernible signal peaks (referred to as the *noise region*). The variance of this region σ^2 is determined, and used to construct an array of values sampled from a normal distribution with mean 0 and variance σ^2,

$N_{0,\sigma^2} \in \mathbb{R}^{2N^{(1)} \times \cdots \times 2N^{(D)}}$. The filtered spectrum is then given by

$$\widetilde{S}_{\text{ve}} = S_{\text{ve}} \odot G + N_{0,\sigma^2} \odot (1 - G). \tag{2.64}$$

Note that the noise array's magnitude at each point is attenuated by the value of the super-Gaussian filter. Inside the region of interest ($G\left[n^{(1)}, \cdots, n^{(D)}\right] = 1$), the noise is nullified. See panel d of Figure 2.3 for an example.

After filtering, $\widetilde{S}_{\text{ve}}$ is returned to the time-domain by IFT. The IFT of a real-valued spectrum generates a conjugate-symmetric signal. This is sliced in half in each dimension, generating the final filtered sub-FID $\widetilde{Y} \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}$:

$$\widetilde{Y} = \frac{1}{2^{D-1}} \text{IFT}\left(\widetilde{S}_{\text{ve}}\right)\left[0 : N^{(1)}, \cdots, 0 : N^{(D)}\right]. \tag{2.65}$$

The scaling factor in (2.65) is to account for how many signals have been combined to generate the VE.

Determining $c^{(d)}$ and $b^{(d)}$

The central index and bandwidth of the super-Gaussian filter function are given by the following expressions:

$$c^{(d)} = \tfrac{1}{2}\left(l_{\text{idx}}^{(d)} + r_{\text{idx}}^{(d)}\right), \tag{2.66a}$$

$$b^{(d)} = l_{\text{idx}}^{(d)} - r_{\text{idx}}^{(d)}, \tag{2.66b}$$

where $l_{\text{idx}}^{(d)}$ and $r_{\text{idx}}^{(d)}$ denote the desired indices where the filter's left and right bounds are located, respectively. Array indices can be obtained from the corresponding spectral frequency $f_{\text{Hz}}^{(d)}$ via

$$f_{\text{idx}}^{(d)} = \left\lfloor \frac{\left(2N^{(d)} - 1\right)\left(f_{\text{sw}}^{(d)} + 2\left(f_{\text{off}}^{(d)} - f_{\text{Hz}}^{(d)}\right)\right)}{2f_{\text{sw}}^{(d)}} \right\rfloor \tag{2.67}$$

$$\forall f_{\text{Hz}}^{(d)} \in \left[f_{\text{off}}^{(d)} - \tfrac{1}{2}f_{\text{sw}}^{(d)}, f_{\text{off}}^{(d)} + \tfrac{1}{2}f_{\text{sw}}^{(d)}\right].$$

Conversion from ppm to array indices can be achieved by replacing f_{Hz} in (2.67) with $f_{\text{ppm}}f_{\text{sfo}}$, where f_{sfo} is the transmitter frequency (MHz).

2.6   Overview

Here, an overview summarising the proposed estimation technique is provided.

(i) Given an FID y for the 1D case, or the set of FIDs $\{Y_{\text{cos}}, Y_{\text{sin}}\}$ for the 2D case, generate a

filtered sub-FID $\widetilde{Y}$ via the VE filtering approach.

(ii) Predict the number of oscillators M using the MDL. Alternatively, provide a hard-coded estimate of M.

(iii) Generate an initial guess $\boldsymbol{\theta}^{(0)}$ using the MPM (1D) or MMEMPM (2D).

(iv) Subject $\boldsymbol{\theta}^{(0)}$ to numerical optimisation, minimising the fidelity (2.43), generating the final estimation result $\boldsymbol{\theta}^{(*)}$.

Algorithm 4 An overview of the estimation proceudre outlined in this work, for the consideration of 1D and 2D Nuclear Magnetic Resonance Spectroscopy (NMR) signals.

1: **procedure** ESTIMATE1D($\boldsymbol{y} \in \mathbb{C}^{N^{(1)}}, \boldsymbol{r}_{\text{interest}} \in \mathbb{R}^2, \boldsymbol{r}_{\text{noise}} \in \mathbb{R}^2, M \in \mathbb{N}_0$)
2:     $\widetilde{\boldsymbol{y}} \leftarrow$ FILTER1D $(\boldsymbol{y}, \boldsymbol{r}_{\text{interest}}, \boldsymbol{r}_{\text{noise}})$;   ▷ **TODO**
3:     **if** $M = 0$ **then**
4:         $M \leftarrow$ MDL $(\widetilde{\boldsymbol{y}})$;   ▷ **TODO**
5:     **end if**
6:     $\boldsymbol{\theta}^{(0)} \leftarrow$ MPM $(\widetilde{\boldsymbol{y}}, M)$;   ▷ Algorithm 1
7:     $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)} \leftarrow$ NLP $(\widetilde{\boldsymbol{y}}, \boldsymbol{\theta}^{(*)})$;   ▷ Algorithm 3
8:     **return** $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)}$;
9: **end procedure**

10: **procedure** ESTIMATE2D($\boldsymbol{Y}_{\cos} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{Y}_{\sin} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{R}_{\text{interest}} \in \mathbb{R}^{2\times2}, \boldsymbol{R}_{\text{noise}} \in \mathbb{R}^{2\times2}, M \in \mathbb{N}_0$)
11:     $\widetilde{\boldsymbol{Y}} \leftarrow$ FILTER2D $(\boldsymbol{Y}_{\cos}, \boldsymbol{Y}_{\sin}, \boldsymbol{R}_{\text{interest}}, \boldsymbol{R}_{\text{noise}})$;   ▷ **TODO**
12:     **if** $M = 0$ **then**
13:         $M \leftarrow$ MDL $(\widetilde{\boldsymbol{Y}}[:, 0])$;   ▷ Run the MDL on the first direct-dimension slice.
14:     **end if**
15:     $\boldsymbol{\theta}^{(0)} \leftarrow$ MMEMPM $(\widetilde{\boldsymbol{Y}}, M)$;   ▷ Algorithm 2
16:     $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)} \leftarrow$ NLP $(\widetilde{\boldsymbol{Y}}, \boldsymbol{\theta}^{(*)})$;   ▷ Algorithm 3
17:     **return** $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)}$;
18: **end procedure**

RESULTS AND APPLICATIONS   3

3.1   one-dimensional Datasets

3.2   Pure-Shift Spectra via 2D J-resolved Estimation

Two key features which the Nuclear Magnetic Resonance Spectroscopy (NMR) community is constantly seeking to improve are the sensitivity and resolving power of experiments. There are numerous means of improving spectral sensitivity, via technological advancements including the production of superconducting magnets with higher field strengths**CITE?** (sensitivity $\propto B_0^{7/4}$), and cryogenic probes[24], as well as simply increasing the number of scans (signal-to-noise ratio (SNR) $\propto NS^{1/2}$). There are however few means of achieving better resolution beyond increased field strengths (resolution $\propto B_0$). Significant interest has therefore been given to the development of homodecoupled ("pure shift") experiments[42, 43, 44], in which the effects of homonuclear scalar couplings are removed from the data. The presence of J-couplings, indicating the close proximity of spins in a chemical bonding sense, is a distinguishing feature of NMR, and is often valuable for structural assignment. However, in many cases their effect can lead to spectra which are too crowded for meaningful insights to be gleamed. Heteronuclear couplings, between spins with different nuclei (i.e. ^{1}H and ^{13}C) are straightforward to decouple at the point of Free Induction Decay (FID) acquisition, using well known schemes such as WALTZ-16[45, 46] and GARP[47]. Homonuclear decoupling is anything but simple on the other hand.

In this section, an overview of the key techniques which have been developed to generate pure shift spectra is given, starting with 2D J-resolved (2DJ) spectroscopy and finishing up with Pure Shift Yeilded by CHirp Excitation (PSYCHE), widely considered the most robust pure shift experiment in terms of sensitivity and tolerance to strong coupling. Subsequently, a new technique for generating pure shift spectra via parametric estimation, given the name Computer-assisted Undiminished-sensitivity Protocol for Ideal Decoupling (CUPID) is presented.
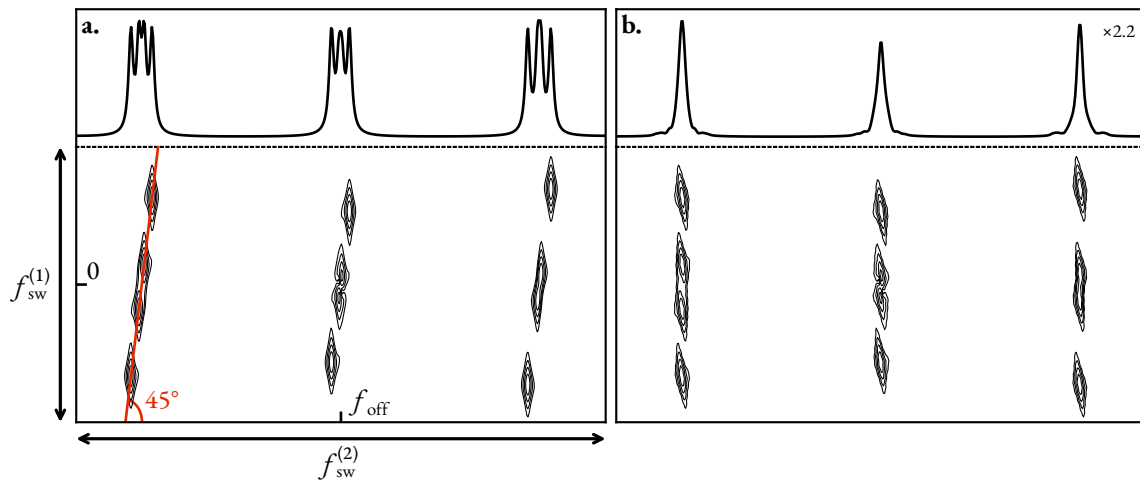
FIGURE 3.1: a. Contour plot 2DJ absolute value mode spectrum for an AMX spin system, produced by applying sine-bell apodisation on a hypercomplex FID before a Fourier transform (FT) in both dimensions. Each multiplet lies a long a line at 45° to the $F^{(1)}$ and $F^{(2)}$ axes. Note that this line often appears to make an angle that is greater than 45° with the $F^{(2)}$ axis when viewing spectra, since typically $f_{sw}^{(1)} \ll f_{sw}^{(2)}$. Above the 2DJ spectrum is a plot of its summation along the $F^{(1)}$ axis. **b.** Spectrum generated after application of a 45° shear, with its $F^{(1)}$ summation above.

3.2.1   An Overview of Pure Shift NMR

The 2D J-resolved Experiment

The 2DJ experiment[13, 48] provided the first means of achieving pure shift spectra. It has a simple pulse sequence, presented in Figure 3.2.a: after excitation of magnetisation onto the transverse plane, the indirect dimension evolution consists of a spin echo, with acquisition following immediately afterwards. Fourier transformation in both dimensions leads to a spectrum in which only scalar couplings contribute in $F^{(1)}$, as the chemical shifts are refocussed by the spin echo, while both scalar couplings and chemical shifts contribute in $F^{(2)}$. Peaks belonging to a particular multiplet lie along a line at 45° to the $F^{(1)}$ and $F^{(2)}$ axes, as seen in panel a. of Figure 3.1.

An FID generated by the 2DJ experiment is hypercomplex, taking the form of (1.14) with $D = 2$ and $\zeta = \exp(i\cdot)$, i.e.

$$
\begin{aligned}
\boldsymbol{Y}_{2DJ}\left[n^{(1)}, n^{(2)}\right] = \sum_{m=0}^{M-1} \boldsymbol{a}\left[m\right] \exp\left(i\boldsymbol{\phi}\left[m\right]\right) \exp\left(\left(2\pi i \boldsymbol{f}^{(1)}\left[m\right] - \boldsymbol{\eta}^{(1)}\left[m\right]\right) n^{(1)} \Delta_t^{(1)}\right) \times \\
\exp\left(\left(2\pi i \boldsymbol{f}^{(2)}\left[m\right] - \boldsymbol{\eta}^{(2)}\left[m\right]\right) n^{(2)} \Delta_t^{(2)}\right) + \boldsymbol{W}\left[n^{(1)}, n^{(2)}\right].
\end{aligned}
\tag{3.1}
$$

A major downside of the 2DJ experiment is there is no means of generating a pair of phase- or amplitude-modulated signals which are the conventional route to frequency-discriminated spectra with absorption mode lineshapes. In many multidimensional experiments, changing the phase of certain radio-frequency (RF) pulses or the use of pulsed field gradients (PFGs) can induce a

change in the coherence transfer pathway (CTP), generating such signal-pairs, though this unavailable with 2DJ spectroscopy as no mixing time exists. The FT of $\boldsymbol{Y}_{2DJ}$ produces a spectrum $\boldsymbol{S}_{2DJ}$ with "phase-twist" peaks, which possess contributions from both absorption and dispersion Lorentzians. There are two primary steps involved in obtaining a pure shift spectrum from $\boldsymbol{S}_{2DJ}$:

(i) Perform a 45° shear (often called a tilt) on the spectrum array, leading to the separation of chemical shifts and scalar couplings onto orthogonal axes (panel b. in Figure 3.1). Each slice in $F^{(2)}$ is subjected to a right circular rotation such that

$$\boldsymbol{S}_{\text{tilt}}\left[n^{(1)}, n^{(2)}\right] = \boldsymbol{S}_{2DJ}\left[n^{(1)}, n^{(2)}_{\text{new}}\right], \tag{3.2a}$$

$$n^{(2)}_{\text{new}} = \left(n^{(2)} + \left\lfloor \frac{f^{(1)}_{\text{sw}} N^{(2)}}{f^{(2)}_{\text{sw}} N^{(1)}} \left(\frac{N^{(1)}}{2} - n^{(1)}\right) \right\rfloor\right) \bmod N^{(2)}. \tag{3.2b}$$

This achieves the mapping $\boldsymbol{S}_{2DJ}\left(f^{(1)}, f^{(2)}\right) \rightarrow \boldsymbol{S}_{2DJ}\left(f^{(1)}, f^{(2)} - f^{(1)}\right)$. It should be noted that the effectiveness of the shear is maximised when both $f^{(2)}_{\text{sw}}/f^{(1)}_{\text{sw}}$ and $N^{(2)}/N^{(1)}$ are powers of 2**check this**.

(ii) Sum the spectrum along $F^{(1)}$:

$$\boldsymbol{s}_{\text{PS}}\left[n^{(2)}\right] = \sum_{n^{(1)}=0}^{N^{(1)}-1} \boldsymbol{S}_{\text{tilt}}\left[n^{(1)}, n^{(2)}\right]. \tag{3.3}$$

Shearing and summing $\boldsymbol{S}_{2DJ}$ actually leads to the absorptive and dispersive components of the spectrum cancelling each other out . It is necessary to display the spectrum in absolute value mode before carrying out the above steps (i.e. use $\Re(|\boldsymbol{S}_{2DJ}|)$). This still leads to undesirable pure shift spectra with broad "wings" on account of the presence of dispersive character. The dispersive component can be suppressed by appropriate processing to make the FID envelope symmetric in both dimensions, such as with sine-bell apodisation[49: Section3.3.5], or pseudo-echo reshaping[50], though this results in a significant reduction in sensitivity being incurred.

The Zangger-Sterk Method

In 1997 Zangger and Sterk introduced a pulse sequence element which achieves slice-selective excitation, by applying a low RF power (weak) 180° pulse* in the presence of a PFG along the z-axis[52]. Such an element excites a given spin only in a narrow range of heights in the sample, as the PFG induces a shift in resonance frequency according to $\Delta\omega(z) = \gamma G_z z$, where G_z is the magnitude of the PFG. By placing a hard 180° pulse adjacent to the selective pulse, the "active" spin in a

---

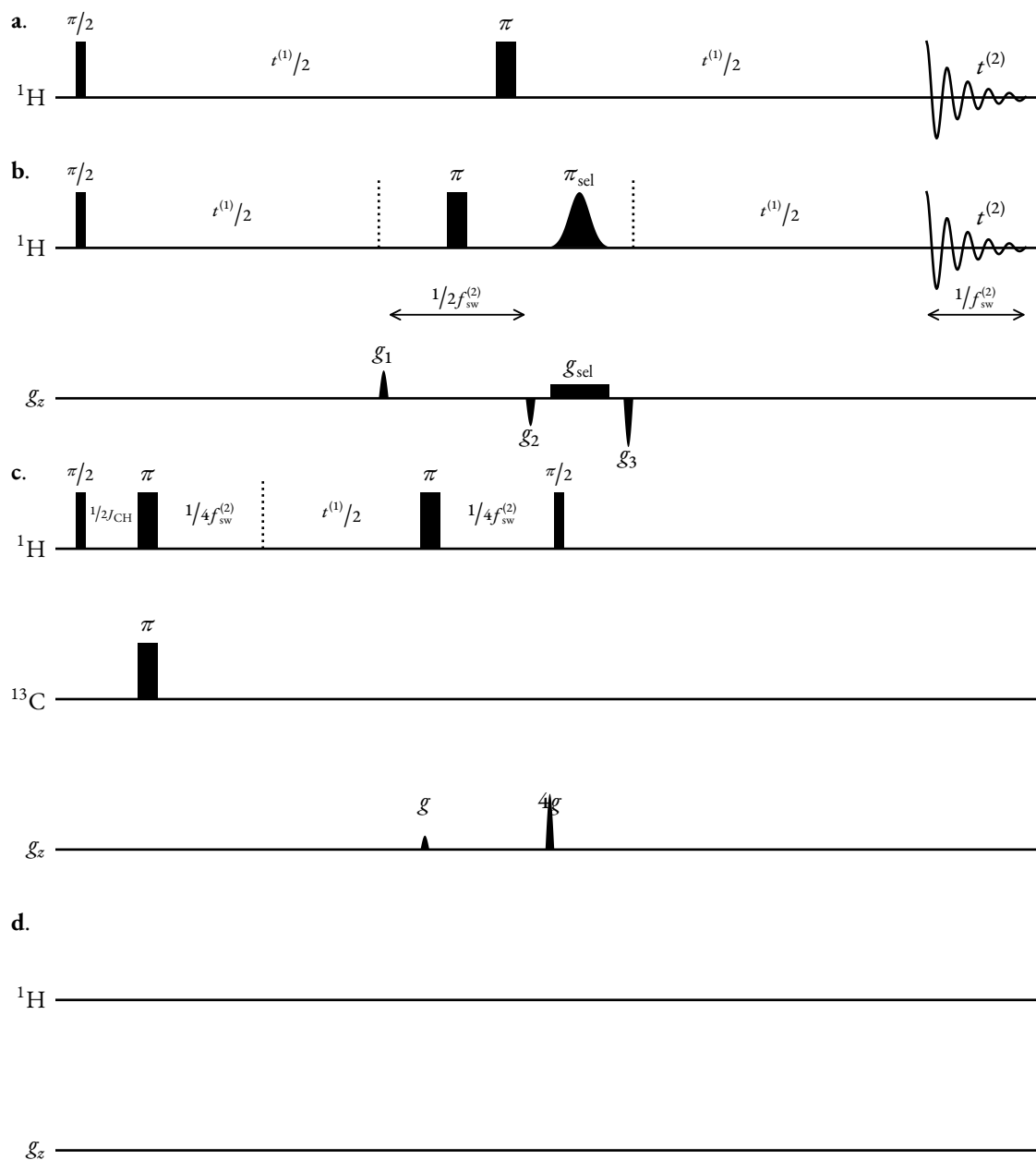*Conventionally, a R-SNOB pulse is used[51].

Figure 3.2: **Work needed. Figure out the correct delays in ZS, BIRD, PSYCHE** The pulse sequences of four of the most common pure shift experiments. **a.** 2DJ. **b.** The ZS method. **c.** The BIRD method. **d.** The PSYCHE method.

given slice is rotated by 360° (i.e. no net rotation), while all other ("passive") spins are only rotated by 180°. Placing such a element in the middle of the $t^{(1)}$ evolution therefore achieves refocussing of the J-couplings associated with the active spin[53]. In order to achieve effective decoupling of any given pair of spins, it is necessary that the bandwidth of the selective π-pulse is smaller than the difference in their chemical shifts. However, with more selective pulses, a smaller proportion of the available spin magnetisation will contribute to the final FID, and hence sensitivity will be diminished †. There is therefore a trade-off between effective decoupling of all spins, and achieving the greatest sensitivity possible. In the case of strong coupling, the Zangger-Sterk (ZS) method tends to perform poorly relative to other options for this reason. The ZS element has been applied by Keeler and Pell in order to generate 2DJ datasets comprising phase-modulated pairs, enabling the generation of pure absorption-mode spectra[54].

The BIRD Method

The BIlinear Rotation Decoupling (BIRD) pulse sequence element[55, 56], presented in Figure 3.2.c, also takes advantage of the idea of selectively inverting passive spins, while leaving active spins unaffected. However the active spins are those which are directly bound to ^{13}C nuclei, which naturally occur with an abundance of 1.1%, while the passive spins are those bound to far more abundant ^{12}C nuclei. The reduction in sensitivity of the experiment relative to a full-sensitivity experiment is therefore known and constant across samples. In scenarios where is this strong coupling, BIRD can achieve improved sensitivity over ZS, since with the latter a very weak selective pulse would be required to ensure it is of a sufficiently small bandwidth. The BIRD method is particularly attractive in scenarios where the sensitivity penalty due to the involvement of a low-abundance nucleus has already been paid, for example in a Heteronuclear Single Quantum Coherence Spectroscopy (HSQC) experiment[57].

PSYCHE

TODO Original paper[58], tutorial paper[59], PSYCHE-2DJ[60].

Pure Shift NMR via J-Resolved Post-Processing

There have been previous descriptions of acquiring pure-shift spectra from typical 2DJ datasets via more sophisticated post-processing methods. Nuzillard presented A Linear Predictive Estimation of Signal Time Reversal (ALPESTRE)[61], in which the parameters of each indirect-dimension

---

†The reduction in sensitivity is $\propto f_B/\gamma G_z l_z$, where f_B is the selective pulse bandwidth, and l_z is the length of the sample lying within the receiver coil (≈ 1.5 cm).

FID are estimated using linear prediction, such that there is a set of parameters $\boldsymbol{\Theta} \in \mathbb{R}^{N^{(2)} \times 4M}$ with

$$\boldsymbol{\Theta} \left[n^{(2)} \right] = \left[\left[\boldsymbol{a}_{n^{(2)}} \right]^{\mathrm{T}} \quad \left[\boldsymbol{\phi}_{n^{(2)}} \right]^{\mathrm{T}} \quad \left[\boldsymbol{f}^{(1)}_{n^{(2)}} \right]^{\mathrm{T}} \quad \left[\boldsymbol{\eta}^{(1)}_{n^{(2)}} \right]^{\mathrm{T}} \right]. \tag{3.4}$$

The parameters generated are used to propagate each FID backward into $-t^{(1)}$, producing a "full-echo":

$$\boldsymbol{Y}_{\text{full}} \left[n^{(1)}, n^{(2)} \right] = \sum_{m=0}^{M-1} \boldsymbol{a}_{n^{(2)}} [m] \exp\left(\mathrm{i}\boldsymbol{\phi}_{n^{(2)}} [m] \right) \exp\left(\left(2\pi \mathrm{i} \boldsymbol{f}^{(1)}_{n^{(2)}} [m] \, n^{(1)} - \boldsymbol{\eta}^{(1)}_{n^{(2)}} [m] \left| n^{(1)} \right| \right) \Delta^{(1)}_t \right)$$

$$\forall n^{(1)} \in \mathbb{Z} : -N^{(1)} < n^{(1)} < N^{(1)},$$
$$\forall n^{(2)} \in \mathbb{N}_0 : n^{(2)} < N^{(2)}. \tag{3.5}$$

FT of generates a spectrum whose real component comprises absorption-mode Lorentzian character in both dimensions. This opens up the means of producing pure-shift spectra from the 2DJ experiment with sharp lineshapes and without sensitivity loss. A similar approach proposed by Mutzenhardt et al. instead constructed full echoes via Linear Prediction (LP) of each direct-dimension FID, and generation of a full echo by propagating into $-t^{(2)}$ [62].

3.2.2   An outline of CUPID

CUPID aims to generate pure shift spectra by utilising the result of parametric estimation on 2DJ data, assumed to take the functional form of (3.1). Instead of estimating successive one-dimensional (1D) FIDs, as has been described by Nuzillard and Mutzenhardt et al., the entire 2DJ signal is estimated as a whole, giving access to the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{6M}$. Using $\boldsymbol{\theta}$, a synthetic 1D FID, named the "$-45°$ signal" (Figure 3.3), is produced:

$$\boldsymbol{y}_{-45°} (\boldsymbol{\theta}) \left[n^{(2)} \right] = \sum_{m=0}^{M-1} \boldsymbol{a} [m] \exp\left(\mathrm{i}\boldsymbol{\phi} [m] \right) \exp\left(\left(2\pi \mathrm{i} \left(\boldsymbol{f}^{(2)} [m] - \boldsymbol{f}^{(1)} [m] \right) - \boldsymbol{\eta}^{(2)} [m] \right) n^{(2)} \Delta^{(2)}_t \right). \tag{3.6}$$

$\forall n^{(2)} \in \{0, \cdots, N^{(2)} - 1\}$. The $-45°$ signal takes the same functional form as a typical 1D FID acquired by a pulse-acquire experiment, except that the frequency of each oscillator, which would usually be $f^{(2)}$, is replaced with $f^{(2)} - f^{(1)}$. In a 2DJ experiment, $f^{(1)}$ corresponds to the displacement of a given oscillator from the central frequency of the multiplet it is associated with. As such, the oscillators belonging to a given multiplet all provide a contribution with the same frequency to the $-45°$ signal, namely the chemical shift of relevant spin.
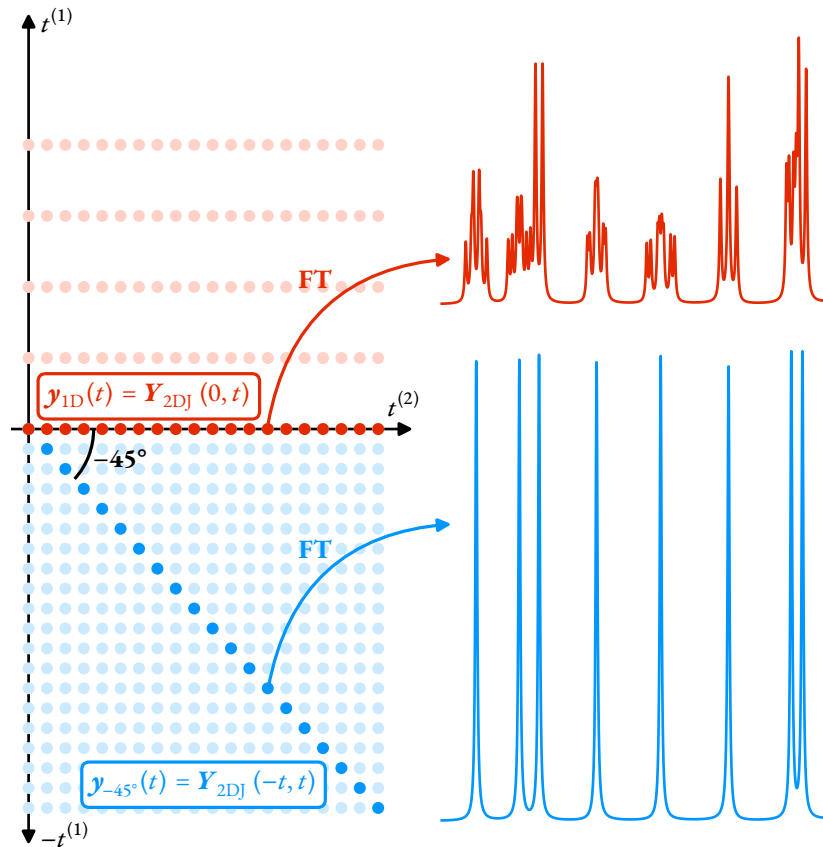
FIGURE 3.3: An illustration of the reasoning behind the name "−45° signal". The pale red dots denote a typical 2DJ FID, where the amount and rate of sampling in the direct dimension is greater than in the indirect dimension (i.e. $N^{(1)} \ll N^{(2)}$ and $f_{\text{sw}}^{(1)} \ll f_{\text{sw}}^{(2)}$). The bright red dots correspond to the first direct-dimension signal $\boldsymbol{Y}_{\text{2DJ}}(0, t^{(2)})$, which has the same form as an FID from a pulse-acquire experiment. A hypothetical signal generated by propagating the FID into $-t^{(1)}$, with the same rate of sampling in both dimensions, is denoted with pale blue dots. Taking the diagonal of this signal, such that it forms a −45° to the $t^{(2)}$ axis, yields an FID $\boldsymbol{y}_{-45°}$ which is homodecoupled. It should be noted that there is a slightly discrepancy between (3.6) and this description, in that the indirect-dimension damping factors $\boldsymbol{\eta}^{(1)}$ are neglected in the former case.

Multiplet Prediction

A holistic two-dimensional (2D) estimation of the FID provides access to other useful information about the dataset. One such example is the grouping of oscillators in account of which multiplet structure they belong to. As has already been established, for oscillators which are associated with the same multiplet structure, the quantity $f^{(2)} - f^{(1)}$ should be equal. This provides a criterion in order to assess whether it is likely that two oscillators in the estimation result belong to the same multiplet:

$$\left| \left(\boldsymbol{f}^{(2)}\left[m_1\right] - \boldsymbol{f}^{(1)}\left[m_1\right] \right) - \left(\boldsymbol{f}^{(2)}\left[m_2\right] - \boldsymbol{f}^{(1)}\left[m_2\right] \right) \right| < \epsilon \tag{3.7}$$

$\forall m_1, m_2 \in \{0, \cdots, M-1\}$, with $\epsilon \in \mathbb{R}_{>0}$ being a suitable threshold to account for error in the estimation. An appropriate value for ϵ would appear to be half the spectral resolution in the more poorly resolved dimension $(F^{(1)})$, i.e. $\epsilon = f_{\mathrm{sw}}^{(1)}/2N^{(1)}$. In situations involving real 2DJ signals however, it is found that ϵ sometimes has to be increased to values slightly larger than this to achieve effective groupings (*vide infra*). Algorithm **??** provides a routine that can be used for multiplet prediction.

    Removal of spurious oscillators

Algorithm 5 An algorithm for multiplet assignment of a 2DJ estimation result.

```
 1:  procedure MultipletAssign(θ ∈ ℝ^{6M}, ε ∈ ℝ_{>0})
 2:      f^{(1)}, f^{(2)} ← θ[2M : 3M], θ[3M : 4M];
 3:      Map ← HashMap[ℝ, Vector[ℕ_0]];
 4:      for m = {0, ⋯, M − 1} do
 5:          f_c = f^{(2)}[m] − f^{(1)}[m];
 6:          Assigned ← False;
 7:          for f_mp, i in Map do
 8:              if |f_c − f_mp| < ε then
 9:                  i ← [i^T  m]^T;
10:                  Assigned ← True;
11:                  break;
12:              end if
13:          end for
14:          if Assigned = False then
15:              Map.insert (f_c, [m]);
16:          end if
17:      end for
18:      return Map;
19:  end procedure
```

Filtration of 2DJ data

Unlike the direct-dimension, which can often comprise sparsely distributed peaks in the Fourier domain, the indirect dimension of 2DJ datasets tends to be rather densely populated. As such, it is typically of little use in attempting to generate filtered sub-FIDs in the indirect dimension. The

filtering procedure utilised for 2DJ data is therefore an extension of the filtration procedure for 1D data described in Section 2.5 (Figure 3.4). The filtering procedure involves the following steps:

(i) The signal $\boldsymbol{Y}_{\text{ve}} \in \mathbb{C}^{N^{(1)} \times 2N^{(2)}}$ is constructed, such that a virtual echo is formed from each direct-dimension signal:

$$
\boldsymbol{Y}_{\text{ve}}\left[n^{(1)}\right] = \left[\Re\left(\boldsymbol{Y}\left[n^{(1)},0\right]\right) \quad \boldsymbol{Y}\left[n^{(1)},1\right] \quad \cdots \right.
$$
$$
\left. \boldsymbol{Y}\left[n^{(1)},N^{(2)}-1\right] \quad 0 \quad \boldsymbol{Y}\left[n^{(1)},N^{(2)}-1\right]^{*} \quad \cdots \quad \boldsymbol{Y}\left[n^{(1)},1\right]^{*}\right] \tag{3.8}
$$

$\forall n^{(1)} \in \{0,\cdots,N^{(1)}-1\}$.

(ii) $\boldsymbol{Y}_{\text{ve}}$ is subjected to FT along the direct dimension to produce the spectrum $\boldsymbol{S}_{\text{ve}}$ (panel a of Figure 3.4). This has a imaginary component of zeros.

(iii) A super-Gaussian $\boldsymbol{G} \in \mathbb{R}^{N^{(1)} \times 2N^{(2)}}$ is constructed (panel b):

$$
\boldsymbol{G} = \mathbf{1} \otimes \boldsymbol{g}^{(2)}, \tag{3.9}
$$

where $\mathbf{1} \in \mathbb{R}^{N^{(1)}}$ is a vector of ones, and $\boldsymbol{g}^{(2)}$ is a super-Gaussian vector given by (2.63b) with $d = 2$.

(iv) A matrix of additive noise is generated by extracting the variance σ^2 of a strip of $\boldsymbol{S}_{\text{ve}}$ which is devoid of peaks, and generating an array $\boldsymbol{W}_{\sigma^2} \in \mathbb{R}^{N^{(1)} \times 2N^{(2)}}$ with values independently sampled from a normal distribution with mean 0 and variance σ^2.

(v) The spectrum is filtered according to (2.64), yielding $\widetilde{\boldsymbol{S}}_{\text{ve}}$ (panel d).

(vi) $\widetilde{\boldsymbol{S}}_{\text{ve}}$ is subjected to Inverse Fourier transform (IFT) and is slicing in half in the direct dimension, yeilding the final filtered signal $\widetilde{\boldsymbol{Y}}$:

$$
\widetilde{\boldsymbol{Y}} = \text{IFT}^{(2)}\left(\widetilde{\boldsymbol{S}}_{\text{ve}}\right)\left[:,:N^{(2)}\right]. \tag{3.10}
$$

3.2.3   Results Using CUPID

A number of examples of the application of CUPID are now provided. Initially, a few results are presented using data simulated using the Spinach MATLAB® library[63]. After this, examples are provided with experimental data. In a couple of these, comparison of the result acquired using CUPID is compared with a spectrum acquired using PSYCHE.
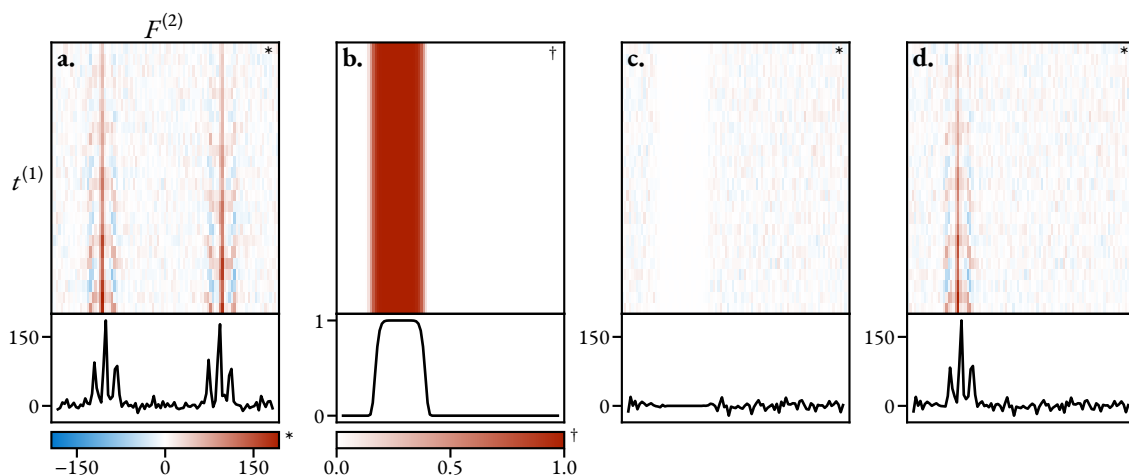
FIGURE 3.4: An illustration of the filtering procedure for 2DJ data. For each panel is a heat-map of the full 2D signal, as well as a plot underneath of the first slice of the signal in the direct dimension. **a.** The spectrum S_{ve}, **b.** Super-Gaussian filter G, **c.** Additive noise, attenuated by the super-Gaussian, $W_{\sigma^2} \odot (1 - G)$, **d.** Filtered spectrum $\widetilde{S}_{\text{ve}}$ Panels **a.–d.** are analogous to panels **b.– e.** in Figure 2.3 for the 1D case.

"Four Multiplets"

The CUPID procedure was applied on a series of datasets generated such that part of the spectrum comprised four ddd multiplet structures with significant overlap. To achieve this, a spin-system with 7 spins was constructed (Figure **??**.a). 4 of the spins (the *"estimated spins"*) were assigned random chemical shifts sampled from a uniform distribution with frequencies in the range ±20 Hz (±0.04 ppm). Each of these 4 spins were coupled with the remaining 3 spins (the *"coupling spins"*), with each coupling randomly sampled from a uniform distribution in the range ±10 Hz. The coupling spins were given chemical shifts such that they lay far from the estimated spins in the spectrum. White Gaussian noise was added to the FID, with a variance such that the SNR of the FID was approximately 30 dB. Estimation was then performed on a filtered FID comprising the signals from the estimated spins (l = 30 Hz, r = −30 Hz). Figure **??**.b illustrates the appearance of 5 such datasets generated using the approach. Table **??** provides the chemical shifts and scalar couplings present in each spin system.

Each filtered FID comprised 32 ($4{\times}2^3$) oscillators. To assess the estimation procedure's ability, for each dataset generated:

(i) A random integer from the range $[33, 40]$ was given as the initial number of oscillators. In this way, the initial guess would always comprise an excessive number of oscillators.

(ii) An initial guess was generated using the MEMP: $\theta^{(0)}$.

(iii) The initial guess was subjected to non-linear programming, generating the optimised result $\theta^{(*)}$.

(iv) Spurious oscillators were checked for, using the criteria outlined in Section **??**, with the threshold for multiplet assignment set to double the spectral resolution in the direct dimension:

$$\epsilon = 2f_{\mathrm{sw}}^{(2)}\big/N^{(2)}.$$

If spurious oscillators were found, these were removed, and non-linear programming was run on the updated set of parameters, updating $\theta^{(*)}$.

For each random spin system generated, the method was effective at producing an estimation result with 32 oscillators, as desired, despite the excessive number that the initial guess possessed. Most excessive oscillators were purged from the initial guess through the NLP procedure. Any spurious oscillators which did still remain (for 2 of the 5 datasets, the result after NLP comprised 33 oscillators) were then detected using multiplet assignment and removed. The pure-shift spectra generated using CUPID closely agree with spectra generated by running a Spinach simulation on a spin system with the same chemical shifts, but with all scalar couplings set to 0 Hz.

3.3   Sequential Datasets

There are a number of two-dimensional NMR experiments in which the variation of a parameter in the pulse sequence leads to the generation of FIDs of the same form except for an attenuation in their amplitudes. These include experiments for the determination of translational diffusion rates and relaxation properties such as T_1, T_2, and $T_{1\rho}$. Estimation of such datasets using nonlinear programming is attractive as after the first FID has been estimated without *a priori* information, subsequent FIDs can be estimated using the previous result as an initial guess. On top of this, only the amplitudes of each increment need to be optimised, as the phases, frequencies and damping factors will be unperturbed across increments.

In this chapter, a generalised approach to analysing such datasets using the estimation procedure presented in Chapter 2 is presented.

3.3.1   Methodology

Outline of the problem

There are numerous experiments in which the pulse sequence is repeated over multiple values of a certain parameter $\boldsymbol{v} \in \mathbb{R}^I$. Examples of this are various relaxation experiments (Section 3.3.2) in which this parameter is effectively the amount of time relaxation is allowed to occur, and diffusion experiments (Section 3.3.3) in which it is the strength of the diffusion-encoding PFGs. These experiments lead to an FID $\boldsymbol{Y} \in \mathbb{C}^{I \times N}$ in which the value of the experimental parameter attenuates

Experiment	v	ζ	ψ	$\mathcal{A}(v,\psi)$	$\frac{\partial \mathcal{A}(v,\psi)}{\partial \psi}$	$\frac{\partial^2 \mathcal{A}(v,\psi)}{\partial \psi^2}$
Inversion Recovery	τ	a_∞	T_1	$\left(1 - 2\exp\left(-\frac{\tau}{T_1}\right)\right)$	$-\frac{2\tau}{T_1^2}\exp\left(-\frac{\tau}{T_1}\right)$	$\frac{2\tau}{T_1^3}\exp\left(-\frac{\tau}{T_1}\right)\left(2 - \frac{\tau}{T_1}\right)$
CPMG	n_cycles	a_0	T_2	?	?	?
Diffusion	g	a_0	D	$\exp\left(-cg^2 D\right)$	$-cg^2\exp\left(-cg^2 D\right)$	$c^2 g^4 \exp\left(-cg^2 D\right)$

TABLE 3.1: The various functional forms of $\mathcal{A}$ according to the different sequential NMR experiments considered in this thesis. $\mathcal{A}$ describes how the amplitude of a resonance is affected by the experimental parameter v.

the amplitudes of the contributing resonances:

$$Y[i,n] = \sum_{m=1}^{M} a_m^{(i)} \exp\left(\mathrm{i}\phi_m\right) \exp\left(\left(2\pi\mathrm{i}f_m - \eta_m\right) n\Delta_t\right). \tag{3.11}$$

The variation of the amplitudes of all resonances is described by the parameter vector $\boldsymbol{p} \in \mathbb{R}^{2M}$:

$$\boldsymbol{p} = \left[\zeta_1, \psi_1, \zeta_2, \psi_2, \cdots, \zeta_M, \psi_M\right]^\mathsf{T}. \tag{3.12}$$

$\boldsymbol{\zeta} = \{\zeta_1, \cdots, \zeta_M\}$ are a set of scaling parameters, one per oscillator, that are not typically of interest, but that are necessary for extracting $\boldsymbol{\psi} = \{\psi_1, \cdots, \psi_M\}$, the of set parameters of interest (relaxation rates, diffusion rates, etc. associated with each oscillator). The amplitudes vary from increment to increment as a result of some function $\mathcal{A}$, whose form varies across different types of experiments (see Table 3.1). In general:

$$a_m^{(i)} = a_m\left(v_i, \zeta_m, \psi_m\right) = \zeta_m \mathcal{A}\left(v_i, \psi_m\right) \tag{3.13}$$

Determination of ψ_m, the parameter of interest for a given oscillator can be achieved by non-linear programming, via optimisation of the fidelity

$$\mathcal{F}\left(\zeta_m, \psi_m | \boldsymbol{v}\right) = \left\| \boldsymbol{a}_m - \zeta_m \mathcal{A}\left(\boldsymbol{v}, \psi_m\right) \right\|_2^2, \tag{3.14a}$$

$$\boldsymbol{a} = \left[a_m^{(1)}, a_m^{(2)}, \cdots, a_m^{(I)}\right]^\mathsf{T}. \tag{3.14b}$$

Derivatives:

$$\frac{\partial \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \zeta_m} = \mathcal{A}\left(v_i, \psi_m\right) \tag{3.15a}$$

$$\frac{\partial \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m} = \zeta_m \frac{\partial \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m} \tag{3.15b}$$

$$\frac{\partial^2 \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \zeta_m^2} = 0 \tag{3.15c}$$

$$\frac{\partial^2 \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m \partial \zeta_m} = \frac{\partial^2 \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \zeta_m \partial \psi_m} = \frac{\partial \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m} \tag{3.15d}$$

$$\frac{\partial^2 \zeta_m \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m^2} = \zeta_m \frac{\partial^2 \mathcal{A}\left(v_i, \psi_m\right)}{\partial \psi_m^2} \tag{3.15e}$$

$$\tag{3.15f}$$

It therefore is necessary to determine the amplitudes of a given oscillator for a given increment, which is desribed in the next section

Determining amplitudes

3.3.2   Relaxation experiments

Integrals of peaks modelled with two parameters $\boldsymbol{\theta} \in \mathbb{R}^2$:

$$\theta_1 = I_\infty = \lim_{\tau \to \infty} x \tag{3.16a}$$

$$\theta_2 = T_1 \tag{3.16b}$$

$$\boldsymbol{I}\left(\boldsymbol{\theta}, \boldsymbol{\tau}\right) = I_\infty \left[1 - 2\exp\left(-\frac{\tau}{T_1}\right)\right], \tag{3.17}$$

Fitting this function is achieved by minimising L2-norm (reference it in previous discussion, and refer to grad and Hessian). The first and second derivatives of the model, required to construct the grad and Hess are

$$\frac{\partial I}{\partial I_\infty} = 1 - 2\exp\left(-\frac{\tau}{T_1}\right) \tag{3.18a}$$

$$\frac{\partial I}{\partial T_1} = -\frac{2I_\infty \tau}{T_1^2}\exp\left(-\frac{\tau}{T_1}\right) \tag{3.18b}$$

$$\frac{\partial^2 I}{\partial I_\infty^2} = 0 \tag{3.18c}$$

$$\frac{\partial^2 I}{\partial I_\infty \partial T_1} = \frac{\partial^2 I}{\partial T_1 \partial I_\infty} = -\frac{2\tau}{T_1^2}\exp\left(-\frac{\tau}{T_1}\right) \tag{3.18d}$$

$$\frac{\partial^2 I}{\partial T_1^2} = \frac{2I_\infty \tau}{T_1^3}\exp\left(-\frac{\tau}{T_1}\right)\left(2 - \frac{\tau}{T_1}\right) \tag{3.18e}$$

3.3.3　Diffusion experiments

Along with their widespread use in dephasing undesired coherences in a pulse sequence, the use of PFGs is central to determining translational diffusion properties of species with NMR. The first showcase for extracting translational diffusion coefficients came from Stejskal and Tanner in 1965, in which they introduced the Pulsed Gradient Spin Echo (PGSE) pulse sequence[64] (Figure 3.5.a).

The PGSE sequence consists of a conventional spin-echo ($90° \xrightarrow{\tau} 180° \xrightarrow{\tau}$ acquire), with PFGs applied after each of the RF pulses. As a simple overview of how the pulse sequence works, consider a single spin on resonance with the transmitter (i.e. its rotating frame frequency is zero) in a sample tube at position z along the axis collinear with the main field. After the 90° pulse, the magnetisation will be $-M_y$. During the first PFG, the spin's resonance frequency will become $\omega_{PFG} = -\gamma g z$, where g is the strength of the PFG. Assuming the gradient is applied for a time δ, the spin will precess by an angle of $\alpha = -\gamma g z \delta$. After the 180° pulse, the spin's magnetisation is as follows:

$$-M_y \xrightarrow{\text{PFG}} -M_y \cos(\alpha) + M_x \sin(\alpha) \xrightarrow{180°_y} -M_y \cos(\alpha) - M_x \sin(\alpha).$$

Supposing that the spin has moved to a new position $z + \Delta_z$ between the end of the first gradient and the beginning of the second, application of the second gradient causes precession by the angle $\beta = -\gamma g (z + \Delta_z) \delta$:

$$\xrightarrow{\text{PFG}} - M_y \cos(\alpha) \cos(\beta) + M_x \cos(\alpha) \sin(\beta) - M_x \sin(\alpha) \cos(\beta) - M_y \sin(\alpha) \sin(\beta)$$
$$= -M_y \cos(\gamma g \delta \Delta_z) - M_x \sin(\gamma g \delta \Delta_z),$$

In the scenario that the spin has not translated in the z-direction between PFGs ($\Delta_z = 0$), the net effect of the pulse sequence is nothing (except for a loss of signal amplitude through T_2 relaxation). However, if translation does occur, the signal phase is adjusted, as a function of the extent of translation Δ_z. The gradients have effectively been employed to encode the change in position of the spin after a known amount of time. Extending this idea to a system of many identical spins, which will translate by different extents between the PFGs, individual spin contributions to the bulk magnetisation will become dephased, leading to an attenuation of the amplitude of the resulting FID.

Through consideration of the Bloch-Torrey equations[65], which extend the classic Bloch equations to account for the effects of diffusion on magnetisation, Stejskal and Tanner were able to derive the following equation for the variation of the amplitude of a resonance as a function of gradient strength, known widely as the Stejskal-Tanner equation:

$$a(g) = a_0 \exp\left(-\gamma^2 \delta^2 g^2 D \left(\Delta - \frac{\delta}{3}\right)\right), \tag{3.19}$$
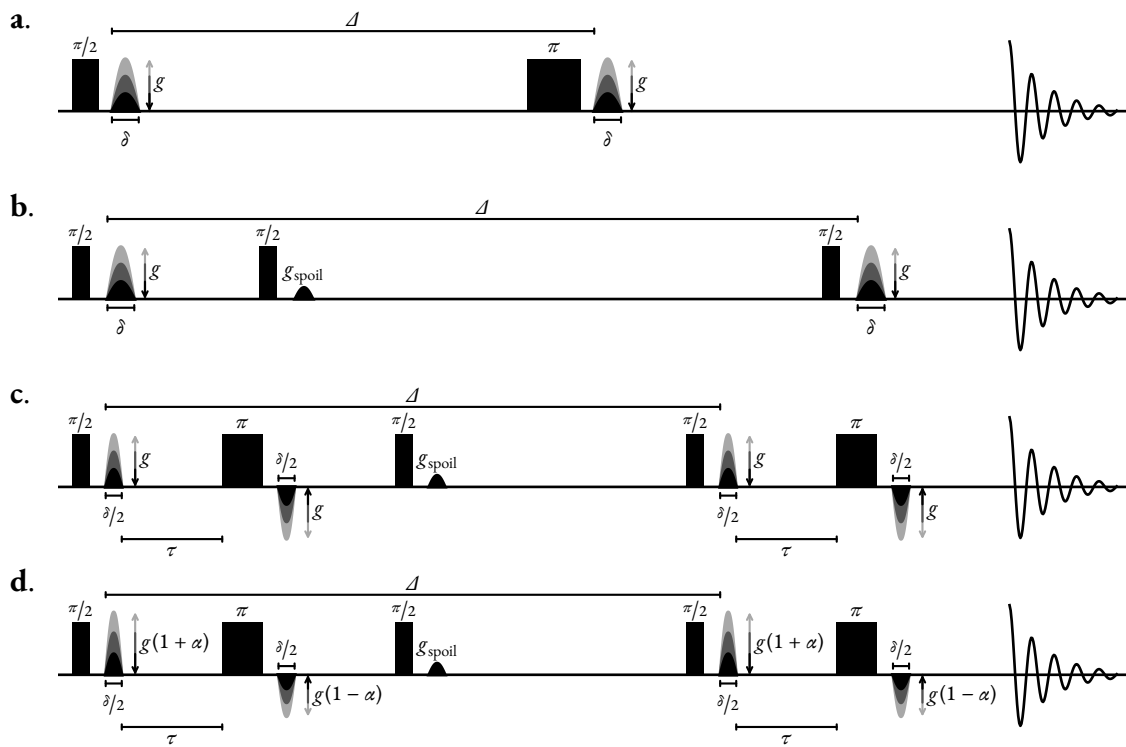
FIGURE 3.5: Pulse sequences used for the determination of translational diffusion constants. **a.** PGSE, **b.** PGSTE, **c.** PGSTEBP, **d.** One-shot DOSY. RF pulses are denoted by solid rectangles. Diffusion-encoding gradients are denoted by sine-bell shapes with varying shades, indicating that the intensity is incremented to create a 2D dataset. Spoiler gradients are denoted by solid black sine-bell shapes.

where $a_0 = \lim_{g \to 0} a$, γ is the gyromagnetic ratio of the target nucleus (MHz T^{-1}), g is the gradient strength (T m^{-1})[‡], δ is the duration of the PFGs (s), $\varDelta$ is the delay between the PFGs, often known as the diffusion time (s), and D is the translation diffusion constant of the species giving rise to the resonance (m s^{-2}). While Equation 3.19 is widely stated in the literature, it is only technically applicable to the scenario in which the PGSE sequence is used (or any other *monopolar* sequence, *vide infra*), and rectangular PFGs are applied [§].

Tanner introduced a variant on the original PGSE experiment called Pulsed Gradient STimulated Echo (PGSTE)[66] (Figure 3.5.b). Instead of the diffusion period including a 180° pulse, PGSTE has two 90° pulses, with the first being applied shortly after the initial PFG, and the second being applied just before the second PFG. The key difference between this and the PGSE experiment is that relaxation of the signal during the diffusion time is dictated by longitudinal relaxation (T_1), rather than transverse relaxation (T_2). PGSTE is therefore favoured in scenarios where $T_1 \ll T_2$, as improved sensitivity will be achieved in the resulting FID.

Both PGSE and PGSTE employ *monopolar* PFGs for diffusion encoding, in the sense that they are both polarised in a single direction. Experiments also exist which employ *bipolar* gradient elements[67, 68], which consist of a PFG, followed by a 180° pulse, and then a second PFG with the opposite polarity to the first. A well-known example is the Pulsed Gradient STimulated Echo with BiPolar gradients (PGSTEBP) experiment (Figure 3.5.c). Bipolar gradient are useful in circumstances where it is important to purge the effects of static gradients in the sample, caused by field inhomogeneities. Morris and coworkers have also developed the One-shot DOSY experiment[69] (Figure 3.5.d), which requires a single transient per gradient strength (i.e. there is no requirement for a phase-cycling scheme). This is achieved through the use of bipolar gradients which comprise asymmetrical PFGs with relative powers $1 + \alpha : 1 - \alpha$ for some $\alpha > 0$ [¶]. **Cite some review articles (Johnson, Morris 2009)**

It is virtually always the case that the amplitudes of each resonance in the FID abide by the following general form of the Stejskal-Tanner equation:

$$a(g) = a_0 \exp\left(-cg^2 D\right) \tag{3.20}$$

for some constant c (T s^{-2}). It is important to note that functional form of c is highly variable dependent on the type of experiment used, and its value if affected by the shape of the diffusion-encoding PFGs. A consideration of the Bloch-Torrey equation for a given experiment is necessary, with an extensive overview provided by Sinnaeve for most diffusion NMR experiments[70]. In

---

[‡]Gradient strengths are often expressed in units of G cm^{-1}, which is equivalent to 10^{-2} T m^{-1}.
[§]Rectangular PFGs (i.e. those in which there is an infinitesimal time to rise to full strength, and to fall back to zero) are in fact impossible to achieve as they would require gradient coils with zero inductance.
[¶]α is typically set to be considerably less that 1, commonly 0.2.

general, c is as follows:

$$c = \gamma^2 \delta^2 \sigma^2 \Delta'. \tag{3.21}$$

σ is the *shape factor* of the PFGs (*vide infra*), and Δ' is the effective time that diffusion is allowed to occur. Examples of the value of Δ' include:

Monopolar gradients (PGSE, PGSTE) $\quad \Delta + 2\left(\kappa - \lambda\right)\delta, \tag{3.22a}$

Bipolar gradients (PGSTEBP) $\quad \Delta + \dfrac{(2\kappa - 2\lambda - 1)\delta}{4} - \dfrac{\tau}{2}, \tag{3.22b}$

One-shot $\quad \Delta + \dfrac{(\kappa - \lambda)(\alpha^2 + 1)\delta}{2} + \dfrac{(\delta + 2\tau)(\alpha^2 - 1)}{4}. \tag{3.22c}$

τ is the delay between the initial PFG and the 180° pulse in experiments with bipolar gradients. The factors σ, λ, and κ are related to the shape function $s(\epsilon) : \epsilon \in [0,1]$ of the PFG, which describes the variation in the intensity of the gradient as a function of its progression. For a rectangular gradient, $s(\epsilon) = 1 \forall \epsilon$, whereas for a sine-bell gradient, $s(\epsilon) = \sin(\pi\epsilon)$. The cumulative distribution of the shape function is given by:

$$S(\epsilon) = \int_0^\epsilon s\left(\epsilon'\right) \mathrm{d}\epsilon' \quad \forall \epsilon \in [0,1]. \tag{3.23}$$

The corresponding definition of S for the case of a gradient made of discrete steps with shape function $\boldsymbol{s} \in \mathbb{R}^{N_g}$ is

$$S[n] = \frac{1}{n} \sum_{i=0}^n s[i] \quad \forall n \in \left\{0, \cdots, N_g - 1\right\}, \tag{3.24}$$

where N_g is the number of points the gradient comprises. The three factors are given by

$$\sigma = S(1), \tag{3.25a}$$

$$\lambda = \frac{1}{\sigma} \int_0^1 S(\epsilon) \mathrm{d}\epsilon, \tag{3.25b}$$

$$\kappa = \frac{1}{\sigma^2} \int_0^1 S^2(\epsilon) \mathrm{d}\epsilon, \tag{3.25c}$$

with their discrete counterparts being

$$\sigma = S\left[N_g - 1\right] \tag{3.26a}$$

$$\lambda = \frac{1}{\sigma N_g} \sum_{n=0}^{N_g-1} S[n] = \frac{1}{\sigma N_g} \sum_{n=0}^{N_g-1} \frac{1}{n} \sum_{i=0}^n s[i] \tag{3.26b}$$

$$\kappa = \frac{1}{\sigma^2 N_g} \sum_{n=0}^{N_g-1} S^2[n] = \frac{1}{\sigma^2 N_g} \sum_{n=0}^{N_g-1} \frac{1}{n^2} \left(\sum_{i=0}^n s[i]\right)^2 \tag{3.26c}$$

For PFGs with a symmetrical shape, $\lambda = 1/2$. κ is typically equal to or close to $1/3$. It can now be seen that Equation 3.19 comes from plugging Equation 3.22a into Equation 3.21, with $\sigma = 1$, $\lambda = 1/2$, and $\kappa = \frac{1}{3}$. In many situations, Δ dominates in the expression of Δ', and so ensuring the correct form of c could be seen as excessive. However, especially when Δ is not orders of magnitude greater than δ, the exact form of Δ' used in Equation 3.21 will be extremely important for accurate measurements of D.

The necessary first and second derivatives required to estimate the translational diffusion constant using the **Trust NCG! (Trust NCG!)** algorithm are:

$$\frac{\partial a}{\partial a_0} = \exp\left(-cg^2 D\right) \tag{3.27a}$$

$$\frac{\partial a}{\partial D} = -a_0 c g^2 \exp\left(-cg^2 D\right) \tag{3.27b}$$

$$\frac{\partial^2 a}{\partial a_0^2} = 0 \tag{3.27c}$$

$$\frac{\partial^2 a}{\partial a_0 \partial D} = \frac{\partial^2 a}{\partial D \partial a_0} = -cg^2 \exp\left(-cg^2 D\right) \tag{3.27d}$$

$$\frac{\partial^2 a}{\partial D^2} = a_0 c^2 g^4 \exp\left(-cg^2 D\right) \tag{3.27e}$$

$$\frac{\partial^n x}{\partial D^n} = \left(-cG^2\right)^n \exp\left(-cDG^2\right) \tag{3.28}$$

3.3.4   Plotting DOSY-type results

Spectrum $S \in \mathbb{R}^{N,I}$ is given by

$$S = \sum_{m=1}^{M} \boldsymbol{s}_m \otimes \frac{\boldsymbol{d}}{\max\left(\boldsymbol{d}\right)} \tag{3.29a}$$

$$\boldsymbol{s}_m = \mathrm{FT}\left\{a_m^{(1)} \exp\left(\mathrm{i}\phi_m\right) \exp\left(\left(2\pi\mathrm{i}f_m - \eta_m\right)\boldsymbol{n}\Delta_t\right)\right\} \tag{3.29b}$$

$$\boldsymbol{d} = \exp\left(-\frac{\left(\boldsymbol{y} - \theta_2\right)^2}{2\sigma_2^2}\right) \tag{3.29c}$$

3.4   **Miscellanea**

BBQCHILI, Metabolomics (using PCA,PLS,OPLS)

THE NMR-ESPY PACKAGE

4

Conclusions and Future Work

5

Bibliography

[1]   F. Bloch, W. W. Hansen, and Martin Packard. "Nuclear Induction". In: *Phys. Rev.* 69 (3-4 Feb. 1946), pp. 127–127.

[2]   Edward M Purcell, Henry Cutler Torrey, and Robert V Pound. "Resonance absorption by nuclear magnetic moments in a solid". In: *Physical review* 69.1-2 (1946), p. 37.

[3]   Edwin D. Becker. "A brief history of Nuclear Magnetic Resonance". In: *Analytical Chemistry* 65.6 (1993), 295A–302A.

[4]   Gareth R Eaton, Sandra S Eaton, and K. M Salikhov. *Foundations of modern EPR*. Singapore ; London: World Scientific, 1998.

[5]   W. D. Knight. "Nuclear Magnetic Resonance Shift in Metals". In: *Phys. Rev.* 76 (8 Oct. 1949), pp. 1259–1260.

[6]   W. G. Proctor and F. C. Yu. "The Dependence of a Nuclear Magnetic Resonance Frequency upon Chemical Compound". In: *Phys. Rev.* 77 (5 Mar. 1950), pp. 717–717.

[7]   W. C. Dickinson. "Dependence of the ^{19}F Nuclear Resonance Position on Chemical Compound". In: *Phys. Rev.* 77 (5 Mar. 1950), pp. 736–737.

[8]   R. R. Ernst and W. A. Anderson. "Application of Fourier Transform Spectroscopy to Magnetic Resonance". In: *Review of Scientific Instruments* 37.1 (1966), pp. 93–102.

[9]   Ray Freeman and Gareth A. Morris. "The Varian story". In: *Journal of Magnetic Resonance* 250 (2015), pp. 80–84.

[10]   James W. Cooley and John W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 19.90 (1965), pp. 297–301.

[11]   J Jeener. "Ampere international summer school". In: *Basko Polje, Yugoslavia* 197.1 (1971).

[12]   J Jeener and Alewaeters G. ""Pulse pair technique in high resolution NMR" a reprint of the historical 1971 lecture notes on two-dimensional spectroscopy". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 94-95 (2016), pp. 75–80.

[13]   WP Aue, J Karhan, and RR Ernst. "Homonuclear broad band decoupling and two-dimensional J-resolved NMR spectroscopy". In: *The Journal of Chemical Physics* 64.10 (1976), pp. 4226–4227.

[14] Michael P. Williamson, Timothy F. Havel, and Kurt Wüthrich. "Solution conformation of proteinase inhibitor IIA from bull seminal plasma by 1H nuclear magnetic resonance and distance geometry". In: *Journal of Molecular Biology* 182.2 (1985), pp. 295–315.

[15] Dominique Marion et al. "Overcoming the overlap problem in the assignment of proton NMR spectra of larger proteins by use of three-dimensional heteronuclear proton-nitrogen-15 Hartmann-Hahn-multiple quantum coherence and nuclear Overhauser-multiple quantum coherence spectroscopy: application to interleukin 1.beta." In: *Biochemistry* 28.15 (1989), pp. 6150–6156.

[16] LE Kay et al. "Four-dimensional heteronuclear triple-resonance NMR spectroscopy of interleukin-1 beta in solution". In: *Science* 249.4967 (1990), pp. 411–414.

[17] Konstantin Pervushin et al. "Attenuated T2 relaxation by mutual cancellation of dipole–dipole coupling and chemical shift anisotropy indicates an avenue to NMR structures of very large biological macromolecules in solution". In: *Proceedings of the National Academy of Sciences* 94.23 (1997), pp. 12366–12371.

[18] Richard R. Ernst. "Nuclear Magnetic Resonance Fourier Transform Spectroscopy (Nobel Lecture)". In: *Angewandte Chemie International Edition in English* 31.7 (1992), pp. 805–823.

[19] Kurt Wüthrich. "NMR Studies of Structure and Function of Biological Macromolecules (Nobel Lecture)". In: *Angewandte Chemie International Edition* 42.29 (2003), pp. 3340–3363.

[20] P.A. Abragam and A. Abragam. *The Principles of Nuclear Magnetism*. Comparative Pathobiology - Studies in the Postmodern Theory of Education. Clarendon Press, 1961.

[21] M Goldman. *Quantum description of high-resolution NMR in liquids*. eng. International series of monographs on chemistry ; 15. Oxford: Clarendon Press, 1988.

[22] John Cavanagh. *Protein NMR spectroscopy : principles and practice*. eng. 2nd ed. Burlington, Mass. ; London: Academic Press, 2007.

[23] Malcolm H. Levitt. *Spin dynamics : basics of nuclear magnetic resonance*. eng. Second edition. Chichester, 2007.

[24] Helena Kovacs, Detlef Moskau, and Manfred Spraul. "Cryogenically cooled probes—a leap in NMR technology". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 46.2 (2005), pp. 131–155.

[25] P. J. Hore, J. A Jones, and Stephen Wimperis. *NMR : the toolkit : how pulse sequences work*. eng. Second edition. Oxford chemistry primers. Oxford, 2015.

[26] R Fletcher. *Practical methods of optimization*. eng. 2nd ed. Chichester: Wiley, 1987.

[27] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. 2nd ed. Springer series in operations research. New York: Springer, 2006.

[28] Y Hua and T.K Sarkar. "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise". In: *IEEE Trans. on Acoust. Speech Signal Process.* 38.5 (1990), pp. 814–824.

[29] Y Hua and T.K Sarkar. "Matrix pencil and system poles". In: *Signal Processing* 21.2 (1990), pp. 195–198.

[30] Y Hua and T.K Sarkar. "On SVD for estimating generalized eigenvalues of singular matrix pencil in noise". In: *IEEE Trans. Signal Process.* 39.4 (1991), pp. 892–900.

[31] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

[32] Y Hua. "Estimating two-dimensional frequencies by matrix enhancement and matrix pencil". eng. In: *IEEE transactions on signal processing* 40.9 (1992), pp. 2267–2280.

[33] Fang-Jiong Chen et al. "Estimation of Two-Dimensional Frequencies Using Modified Matrix Pencil Method". eng. In: *IEEE transactions on signal processing* 55.2 (2007), pp. 718–724.

[34] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1993.

[35] Y. Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford science publications. OUP Oxford, 2001.

[36] Nicholas I.M. Gould et al. "Sensitivity of trust-region algorithms to their parameters". eng. In: *4OR* 3.3 (2005), pp. 227–241.

[37] Yung-Ya Lin et al. "A Novel Detection–Estimation Scheme for Noisy NMR Signals: Applications to Delayed Acquisition Data". In: *J. Magn. Reson.* 128.1 (1997), pp. 30–41.

[38] C.G. Tang. "An Analysis of Baseline Distortion and Offset in NMR Spectra". In: *Journal of Magnetic Resonance, Series A* 109.2 (1994), pp. 232–240.

[39] M. Mayzel, K. Kazimierczuk, and V. Yu. Orekhov. "The causality principle in the reconstruction of sparse NMR spectra". In: *Chem. Commun.* 50.64 (2014), pp. 8947–8950.

[40] Dariusz Gołowicz, Paweł Kasprzak, and Krzysztof Kazimierczuk. "Enhancing Compression Level for More Efficient Compressed Sensing and Other Lessons from NMR Spectroscopy". In: *Sensors (Basel)* 20.5 (2020), p. 1325.

[41] Jie Luo et al. "Fast reconstruction of non-uniform sampling multidimensional NMR spectroscopy via a deep neural network". In: *Journal of Magnetic Resonance* 317 (2020), p. 106772.

[42]   N. Helge Meyer and Klaus Zangger. "Simplifying Proton NMR Spectra by Instant Homonuclear Broadband Decoupling". In: *Angewandte Chemie International Edition* 52.28 (2013), pp. 7143–7146.

[43]   Ralph W. Adams. "Pure Shift NMR Spectroscopy". In: *eMagRes*. John Wiley & Sons, Ltd, 2014, pp. 295–310.

[44]   Klaus Zangger. "Pure shift NMR". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 86-87 (2015), pp. 1–20.

[45]   A.J Shaka et al. "An improved sequence for broadband decoupling: WALTZ-16". In: *Journal of Magnetic Resonance (1969)* 52.2 (1983), pp. 335–338.

[46]   A.J Shaka, James Keeler, and Ray Freeman. "Evaluation of a new broadband decoupling sequence: WALTZ-16". In: *Journal of Magnetic Resonance (1969)* 53.2 (1983), pp. 313–340.

[47]   A.J Shaka, P.B Barker, and Ray Freeman. "Computer-optimized decoupling scheme for wideband applications and low-level operation". In: *Journal of Magnetic Resonance (1969)* 64.3 (1985), pp. 547–552.

[48]   Gareth A. Morris. "Two-Dimensional J-Resolved Spectroscopy". In: *eMagRes*. John Wiley & Sons, Ltd, 2009.

[49]   J.C. Lindon and A.G. Ferrige. "Digitisation and data processing in Fourier transform NMR". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 14.1 (1980), pp. 27–66.

[50]   Ad Bax, Ray Freeman, and Gareth A Morris. "A simple method for suppressing dispersion-mode contributions in NMR spectra: The "pseudo echo"". In: *Journal of Magnetic Resonance (1969)* 43.2 (1981), pp. 333–338.

[51]   E. Kupce, J. Boyd, and I.D. Campbell. "Short Selective Pulses for Biochemical Applications". In: *Journal of Magnetic Resonance, Series B* 106.3 (1995), pp. 300–303.

[52]   Klaus Zangger and Heinz Sterk. "Homonuclear Broadband-Decoupled NMR Spectra". In: *Journal of Magnetic Resonance* 124.2 (1997), pp. 486–489.

[53]   Juan A. Aguilar et al. "Pure Shift 1H NMR: A Resolution of the Resolution Problem?" In: *Angewandte Chemie International Edition* 49.23 (2010), pp. 3901–3903.

[54]   Andrew J. Pell and James Keeler. "Two-dimensional J-spectra with absorption-mode lineshapes". In: *Journal of Magnetic Resonance* 189.2 (2007), pp. 293–299.

[55]   J.R. Garbow, D.P. Weitekamp, and A. Pines. "Bilinear rotation decoupling of homonuclear scalar interactions". In: *Chemical Physics Letters* 93.5 (1982), pp. 504–509.

[56]   Ad Bax. "Broadband homonuclear decoupling in heteronuclear shift correlation NMR spectroscopy". In: *Journal of Magnetic Resonance (1969)* 53.3 (1983), pp. 517–520.

[57] Liladhar Paudel et al. "Simultaneously Enhancing Spectral Resolution and Sensitivity in Heteronuclear Correlation NMR Spectroscopy". In: *Angewandte Chemie International Edition* 52.44 (2013), pp. 11616–11619.

[58] Mohammadali Foroozandeh et al. "Ultrahigh-Resolution NMR Spectroscopy". In: *Angewandte Chemie International Edition* 53.27 (2014), pp. 6990–6992.

[59] Mohammadali Foroozandeh, Gareth A. Morris, and Mathias Nilsson. "PSYCHE Pure Shift NMR Spectroscopy". In: *Chemistry – A European Journal* 24.53 (2018), pp. 13988–14000.

[60] Peter Kiraly et al. "Anatomising proton NMR spectra with pure shift 2D J-spectroscopy: A cautionary tale". In: *Chemical Physics Letters* 683 (2017). Ahmed Zewail (1946-2016) Commemoration Issue of Chemical Physics Letters, pp. 398–403.

[61] Jean-Marc Nuzillard. "Time-Reversal of NMR Signals by Linear Prediction. Application to Phase-Sensitive HomonuclearJ-Resolved Spectroscopy". In: *Journal of Magnetic Resonance, Series A* 118.1 (1996), pp. 132–135.

[62] P. Mutzenhardt, F. Guenneau, and D. Canet. "A Procedure for Obtaining Pure Absorption 2D J-Spectra: Application to Quantitative Fully J-Decoupled Homonuclear NMR Spectra". In: *Journal of Magnetic Resonance* 141.2 (1999), pp. 312–321.

[63] H.J. Hogben et al. "Spinach – A software library for simulation of spin dynamics in large spin systems". eng. In: *Journal of magnetic resonance (1997)* 208.2 (2011), pp. 179–194.

[64] E. O. Stejskal and J. E. Tanner. "Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient". In: *The Journal of Chemical Physics* 42.1 (1965), pp. 288–292.

[65] H. C. Torrey. "Bloch Equations with Diffusion Terms". In: *Phys. Rev.* 104 (3 Nov. 1956), pp. 563–565.

[66] J. E. Tanner. "Use of the Stimulated Echo in NMR Diffusion Studies". In: *The Journal of Chemical Physics* 52.5 (1970), pp. 2523–2526.

[67] R.M Cotts et al. "Pulsed field gradient stimulated echo methods for improved NMR diffusion measurements in heterogeneous systems". In: *Journal of Magnetic Resonance (1969)* 83.2 (1989), pp. 252–266.

[68] D.H. Wu, A.D. Chen, and C.S. Johnson. "An Improved Diffusion-Ordered Spectroscopy Experiment Incorporating Bipolar-Gradient Pulses". In: *Journal of Magnetic Resonance, Series A* 115.2 (1995), pp. 260–264.

[69] Michelle D. Pelta et al. "A one-shot sequence for high-resolution diffusion-ordered spectroscopy". In: *Magnetic Resonance in Chemistry* 40.13 (2002), S147–S152.

[70] Davy Sinnaeve. "The Stejskal–Tanner equation generalized for any gradient shape—an overview of most pulse sequences measuring free diffusion". In: *Concepts in Magnetic Resonance Part A* 40A.2 (2012), pp. 39–65.

[71] C. E. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21.

IMPLEMENTATION DETAILS

<div style="text-align:right">

A

</div>

A bunch of Python code listings illustrating how certain things described in the main text are implemented.

A.1 Computing the gradient and Hessian for non-linear programming

In this section, a description of how the fidelity, its gradient, and its Hessian (Equation 2.39) are computed is provided.

```python
1  from typing import Tuple
2  import numpy as np
3
4  def obj_grad_hess(
5      theta: np.ndarray, *args,
6  ) -> Tuple[float, np.ndarray, np.ndarray]:
7      """Compute the objective, gradient and hessian for 2D data.
8
9      Parameters
10     ----------
11     theta
12         1D array of parameters, of length 2(1+D)M
13
14     args
15         Contains elements in the following order:
16
17         * data: Array of the original FID data.
18         * tp: The time-points the signal was sampled at
```

```
19          * phase_variance: If ``True``, include the oscillator phase
    ↪  variance to
20             the cost function.
21
22      Returns
23      -------
24      obj: float
25          Value of the objective.
26
27      grad: numpy.ndarray
28          Gradient of the objective.
29
30      hess: numpy.ndarray
31          Gauss-Newton Hessian of the objective.
32      """
33      # Unpack args
34      data, tp, phasevar = args
35      # Determine the data dimensionality and number of oscillators
36      D = data.ndim
37      m = theta.size // (2 * (1 + D))
```

We will start by constructing the model array x. It is necessary to know the form of each oscillator in x to compute its derivatives.

$$x = \exp(Z)\,\alpha, \tag{A.1a}$$

$$Z = \tau \otimes z, \tag{A.1b}$$

$$\tau[n] = (n-1)\Delta_t \;\; \forall n \in \{0, \cdots, N-1\} \tag{A.1c}$$

$$z[m] = 2\pi \mathrm{i} f_m - \eta_m \;\; \forall m \in \{1, \cdots, M\} \tag{A.1d}$$

$$\alpha[m] = a_m \exp(\mathrm{i}\phi_m) \tag{A.1e}$$

Therefore a $\mathbb{C}^{N \times M}$ array X_m is created, such that

$$X_m = \exp(Z) * \alpha \tag{A.2}$$

```
1 exp_Z = np.exp(
2     np.outer(
3         tp,  # τ
4         2j * np.pi * theta[2 * M : 3 * M] - theta[3 * M:],  # z
5     )
6 )
```

```
7 alpha = theta[: M] * np.exp(1j * theta[M : 2 * M]
8 X_m = exp_Z * alpha
```

A.2   Multiplet prediction

```
 1 # multiplet_prediction.py
 2 # Simon Hulse
 3 # simon.hulse@chem.ox.ac.uk
 4 # Last Edited: Tue 20 Dec 2022 17:13:57 GMT
 5
 6 from typing import Dict, Iterable
 7 import numpy as np
 8 np.set_printoptions(precision=3)
 9
10
11 def multiplet_prediction(theta, epsilon) -> Iterable[Iterable[int]]:
12     # Dictionary with central frequencies as keys, and lists of indices
          ↪  as values.
13     multiplets = {}
14     # Reshape for easier processing → θ ∈ ℝ^{M×6}
15     theta = theta.reshape((theta.size // 6, 6))
16     # Iterate over each oscillator index and parameters
17     for idx, params in enumerate(theta):
18         center = params[3] - params[2]  # f_c = f^{(2)} - f^{(1)}
19         assigned = False
20         for freq in multiplets.keys():
21             # Test whether |(f_i^{(2)} - f_i^{(1)}) - (f_j^{(2)} - f_j^{(1)})| < ε
22             if abs(center - freq) < epsilon:
23                 assigned = True
24                 multiplets[freq].append(idx)
25                 break
26         if not assigned:
27             multiplets[center] = [idx]
28
29     return list(multiplets.values())
30
31 theta = np.array(
32     [
33         [4, 0, 0, 20, 3, 3],
34         [2, 0, -7, -2, 3, 3],
35         [2, 0, 7, 12, 3, 3],
36         [1, 0, -10, -25, 3, 3],
```

```
37          [2, 0, 0, -15, 3, 3],
38          [1, 0, 10, -5, 3, 3],
39      ],
40      dtype="float64",
41  )
42  theta[:, 2:4] += np.random.uniform(-0.05, 0.05, (theta.shape[0], 2))
43  print("θ =\n{}".format(theta))
44  epsilon = 0.1
45  multiplets = multiplet_prediction(theta, epsilon)
46  print("Predicted multiplets:\n{}".format(multiplets))
47
48  # Output
49  # ======
50  # θ =
51  # [[ 4.000e+00  0.000e+00 -3.978e-02  1.997e+01  3.000e+00  3.000e+00]
52  #  [ 2.000e+00  0.000e+00 -7.044e+00 -2.044e+00  3.000e+00  3.000e+00]
53  #  [ 2.000e+00  0.000e+00  6.969e+00  1.198e+01  3.000e+00  3.000e+00]
54  #  [ 1.000e+00  0.000e+00 -9.983e+00 -2.502e+01  3.000e+00  3.000e+00]
55  #  [ 2.000e+00  0.000e+00 -9.125e-04 -1.495e+01  3.000e+00  3.000e+00]
56  #  [ 1.000e+00  0.000e+00  9.982e+00 -5.027e+00  3.000e+00  3.000e+00]]
57  # Predicted multiplets:
58  # [[0], [1, 2], [3, 4, 5]]
```

MISCELLANEOUS TOPICS OF INTEREST

<div style="float:right">B</div>

B.1 The Nyquist Frequency

The Nyquist frequency defines the highest frequency an oscillator can possess such that it is sampled at least twice per period:

$$f_N = \frac{1}{2\Delta t} \tag{B.1}$$

This quantity as at the heart of the Nyquist-Shannon Sampling Theorem, which - in the words of Claude Shannon[71] - states:

> If a function $y(t)$ contains no frequencies higher than f Hz, it is completely determined by giving its ordinates at a series of points spaced $^1/_{2f}$ seconds apart.

The implication of this is that the continuous FID $y(t)$ may be completely described by its digitisation with sampling interval Δt so long as the frequencies f_j that make it up satisfy:

$$-\frac{1}{2\Delta t} \leq f_j \leq \frac{1}{2\Delta t} \ \forall j \tag{B.2}$$

If any frequency f_j does not satisfy (??), the signal will be insufficiently sampled, and will be spuriously represented as having a frequency f_a that satisfies

$$f_a = 2mf_N - f_j \qquad (f_j > f_N) \tag{B.3}$$

$$f_a = 2mf_N + f_j \qquad (f_j < -f_N), \tag{B.4}$$

where $m \in \mathbb{Z}_+$. This phenomenon is called *aliasing*.

B.2   Hessian Validation

Talk about checking that my Hessian code is correct by using the finite difference method

B.3   Statistical Definitions

Definitions of standard error, likelihood function, ...