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

API application programming interface

AWGN additive white gaussian noise

BIRD bilinear rotation decoupling

COSY correlation spectroscopy

CPMG Carl-Purcell-Meiboom-Gill

CPU central processing unit

CUPID computer-assisted undiminished-sensitivity protocol for ideal decoupling

CTP coherence transfer pathway

DFT density functional theory

DOSY diffusion-ordered spectroscopy

EYM Eckart-Young-Mirsky

FID free induction decay

FFT fast Fourier transform

FT Fourier transform

GN Gauss-Newton

GUI graphical user interface

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

TSE-PSYCHE triple spin echo PSYCHE

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:

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

$\mathcal{U}\left(l, r\right)$ Uniform distribution with bounds l and r:

$$\mathcal{U}(x \mid l, r) = \begin{cases} \frac{1}{r-l} & l \le x \le r \\ 0 & \text{otherwise} \end{cases}$$

$\mathcal{N}_{\mathcal{C}}\left(\mu, \sigma^2\right)$ 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

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

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

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

$\|A\|$ The norm of A, given by

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

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

$$\nabla f(x) = \left[\frac{\partial f}{\partial x[0]} \quad \frac{\partial f}{\partial x[1]} \quad \cdots \quad \frac{\partial f}{\partial x[n-1]} \right]^{\mathrm{T}}$$

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

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

*   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>

Since its conception almost 80 years ago, 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. Details of the means by which the raw experimental data, the FID is produced, and its structure, are given. Following on from this, an account is given of techniques used to extract quantitative information from the FID, which is the central motivation of this work.

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 $I > 1/2$ 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) magnitude:

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

where $\hbar = {}^h/_{2\pi}$ is the reduced Planck constant. Whilst it is not possible to specify multiple com-

Nucleus	I	$\gamma\,(\mathrm{rad\,T^{-1}\,s^{-1}})$	Relative Abundance (%)
^{1}H	1/2	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	1/2	6.7283×10^7	1.07
^{14}N	1	1.9338×10^7	99.636
^{15}N	1/2	-2.7126×10^7	0.364
^{16}O	0	-	99.756
^{17}O	5/2	-3.6281×10^7	0.038
^{19}F	1/2	2.5162×10^8	100
^{31}P	1/2	1.0839×10^8	100
^{113}Cd	1/2	-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.

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

$$E = -\boldsymbol{\mu} \cdot \boldsymbol{B} \tag{1.4}$$
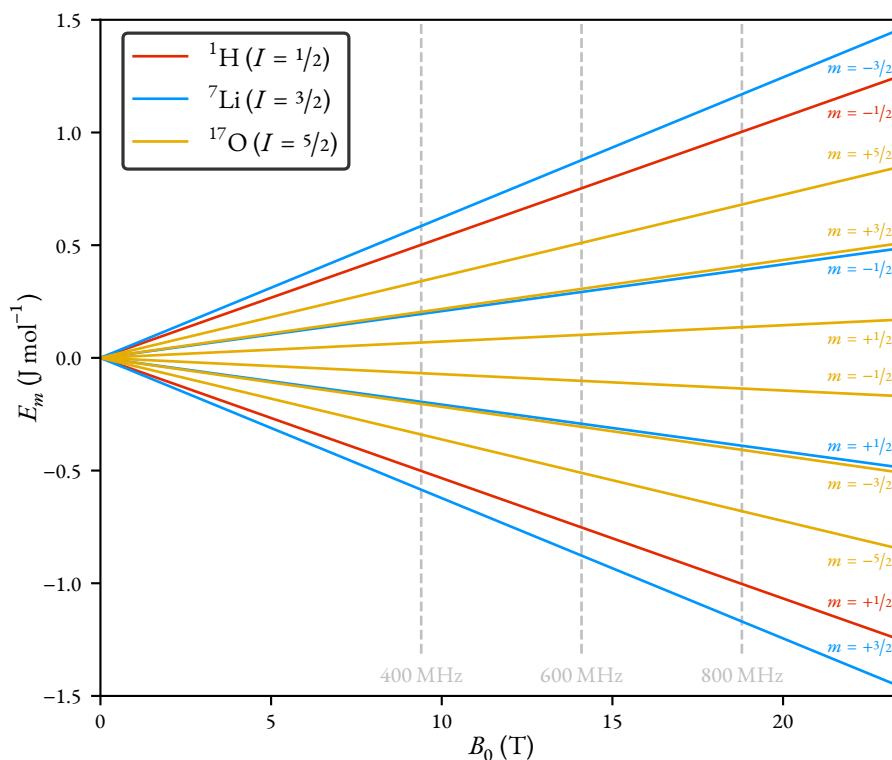
FIGURE 1.1: The variation of energy of the spin states of ^{1}H, ^{7}Li, and ^{17}O with external magnetic field strength (B_0), up to 23.5 T, which is approximately the strength of a 1 GHz NMR magnet. Three common field strengths for commercial NMR magnets are indicated: 9.40 T (400 MHz), 14.10 T (600 MHz), and 18.79 T (800 MHz).

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.

   REVISIT 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 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

---

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

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 an FID. The FID is sampled at equally-spaced time intervals Δ_t, such that it is of the form

$$\boldsymbol{y} = \Big[y(t = 0) \quad y(t = \Delta_t) \quad y(t = 2\Delta_t) \quad \cdots \quad y(t = (N-1)\Delta_t) \Big]^{\mathrm{T}} \tag{1.10}$$

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 A.5.1 for more on this).

[†]In a pulse calibration experiment for example, the goal is to rotate the spin magnetisation by exactly 360°, leading to only longitudinal (undetectable) magnetisation.
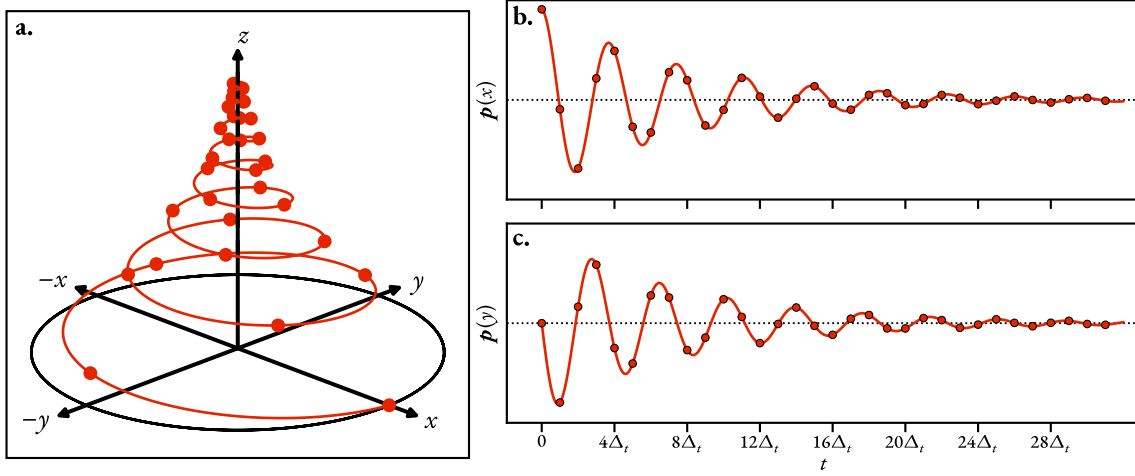
FIGURE 1.2: **If you have time, and can be bothered, the $p(x)$ and $p(y)$ axis labels should not be bold...** An illustration of quadrature detection using the vector model[25: Chapter 1]. **a.** An illustration of the evolution of the bulk magnetisation for a system comprising many identical spins, 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 $p(x) + \mathrm{i}p(y)$.

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 $\boldsymbol{y} \in \mathbb{C}^N$ therefore takes the form

$$\boldsymbol{y}[n] = \boldsymbol{x}[n] + \boldsymbol{w}[n] \quad \forall n \in \{0, 1, \cdots, N-1\}, \tag{1.11a}$$

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

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 which contributes to $\boldsymbol{x}$ is defined by four parameters:

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

✤ $\phi \in (-\pi, \pi]$ - the phase,

✤ $f \in \left[f_{\text{off}} - \frac{1}{2}f_{\text{sw}}, f_{\text{off}} + \frac{1}{2}f_{\text{sw}}\right]$ - the frequency,

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

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

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

where $a \in \mathbb{R}^{M}$ is a vector of all amplitudes, $\phi \in \mathbb{R}^{M}$ is a vector of all phases, etc.

Multidimensional experiments involve incrementing one or more experimental parameters (a common example being a particular delay period in the pulse sequence), in order to obtain an array of 1D FIDs. 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 $\theta \in \mathbb{R}^{2(1+D)M}$ is given by

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

where $f^{(D)}$ and $\eta^{(D)}$ are the frequencies and damping factors in the actively acquired (direct) dimension, and $f^{(1)} - f^{(D-1)}$ and $\eta^{(1)} - \eta^{(D-1)}$ 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 $\exp(2\pi i f t)$ or $\exp(-2\pi i f t)$ 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 $Y \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}$ can be expressed as

$$
\begin{aligned}
Y\left[n^{(1)}, \cdots, n^{(D)}\right] = {} & \sum_{m=0}^{M-1} a\left[m\right] \exp\left(i\phi\left[m\right]\right) \\
& \times \prod_{d=1}^{D} \zeta\left(2\pi\left(f^{(d)}\left[m\right] - f_{\mathrm{off}}^{(d)}\right) n^{(d)} \Delta_t^{(d)}\right) \exp\left(-\eta^{(d)}\left[m\right]n^{(d)}\Delta_t^{(d)}\right) \\
& + W\left[n^{(1)}, \cdots, n^{(D)}\right]
\end{aligned}
\tag{1.14a}
$$

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

Without any specific knowledge about the experiment, it is typical to assume that 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**

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

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

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

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 θ 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 $\boldsymbol{a}$

❖ **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

<div style="text-align: right">2</div>

This chapter provides a detailed outline of an estimation routine which has been developed for the estimation of NMR data.

2.1   Outline of the Problem

For the purposes of this work, it is always assumed that an FID to be estimated is hypercomplex in form, meaning that it obeys (1.14) with $\zeta = \exp(i\cdot) \; \forall d \in \{1, \cdots D\}$:

$$\boldsymbol{Y}\left[n^{(1)}, \cdots, n^{(D)}\right] = \boldsymbol{X}\left(\boldsymbol{\theta}\right)\left[n^{(1)}, \cdots, n^{(D)}\right] + \boldsymbol{W}\left[n^{(1)}, \cdots, n^{(D)}\right] \tag{2.1a}$$

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

$$\tag{2.1b}$$

$$\boldsymbol{W}\left[n^{(1)}, \cdots, n^{(D)}\right] \sim \mathcal{N}_{\mathcal{C}}\left(0, 2\sigma^2\right), \tag{2.1c}$$

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 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

$$p(w) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|w|^2}{2\sigma^2}\right). \tag{2.2}$$

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:

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

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

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 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 $X(\boldsymbol{\theta})$, given in Equation 2.1b, with $D = 1$:

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

(2.7)

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

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

(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 $\boldsymbol{H_{X1}}$ and $\boldsymbol{H_{X2}}$, formed by the removal of the last or first column of $\boldsymbol{H_X}$, respectively:

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

(2.9a)

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

(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:

$$\boldsymbol{H}_{\boldsymbol{X}1} = \boldsymbol{Z}_{\mathrm{L}}\boldsymbol{A}\boldsymbol{Z}_{\mathrm{R}}, \tag{2.10a}$$

$$\boldsymbol{H}_{\boldsymbol{X}2} = \boldsymbol{Z}_{\mathrm{L}}\boldsymbol{A}\boldsymbol{Z}_{\mathrm{D}}\boldsymbol{Z}_{\mathrm{R}}, \tag{2.10b}$$

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

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

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

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

Description of what a matrix pencil is The matrix pencil $\boldsymbol{H}_{\boldsymbol{X}2} - \lambda\boldsymbol{H}_{\boldsymbol{X}1}, \lambda \in \mathbb{C}$ can therefore be expressed as

$$\boldsymbol{H}_{\boldsymbol{X}2} - \lambda\boldsymbol{H}_{\boldsymbol{X}1} = \boldsymbol{Z}_{\mathrm{L}}\boldsymbol{A}\left(\boldsymbol{Z}_{\mathrm{D}} - \lambda\boldsymbol{I}_M\right)\boldsymbol{Z}_{\mathrm{R}}, \tag{2.11}$$

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

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

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]

$$\boldsymbol{z} = \left\{z \in \mathbb{C} : \det\left(\boldsymbol{H}_{\boldsymbol{X}2} - z\boldsymbol{H}_{\boldsymbol{X}1}\right) = 0\right\} \tag{2.13}$$

One means of finding the signal poles is by finding the eigenvalues of the matrix $\boldsymbol{H}_{\boldsymbol{X}1}^{+}\boldsymbol{H}_{\boldsymbol{X}2}$. Deriving the corresponding complex amplitudes can then be achieved by solving the set of linear equations

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

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

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

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

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

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

Noisy data

The presence of noise in the signal $\mathbf{Y}$ complicates the process of determining the M signal poles, as $\mathbf{H}_Y$ ($\mathbf{H}_X$'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 $\widetilde{\mathbf{H}}_Y$. 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**:

$$\widetilde{\mathbf{H}}_Y = \mathbf{U}_M \boldsymbol{\Sigma}_M \mathbf{V}_M^\dagger, \tag{2.16a}$$

$$\mathbb{C}^{(N^{(1)}-L^{(1)})\times M} \ni \mathbf{U}_M = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_M \end{bmatrix}, \tag{2.16b}$$

$$\mathbb{C}^{(L^{(1)}+1)\times M} \ni \mathbf{V}_M = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_M \end{bmatrix}, \tag{2.16c}$$

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

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

$$\widetilde{\mathbf{H}}_Y = \underset{\mathbf{A}:\ \mathrm{rank}(\mathbf{A})=M}{\arg\min} \left\|\mathbf{A} - \mathbf{H}_Y\right\| \tag{2.17}$$

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

$$\mathbf{V}_{M1} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_{M-1} \end{bmatrix}, \tag{2.18a}$$

$$\mathbf{V}_{M2} = \begin{bmatrix} \mathbf{v}_2 & \mathbf{v}_3 & \cdots & \mathbf{v}_M \end{bmatrix}. \tag{2.18b}$$

Algorithm 1 provides a pseudo-code description of the MPM as implemented in the 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:      $\boldsymbol{V}_M \leftarrow \boldsymbol{V}\left[:,:M\right]$; $\triangleright$ Retain first M right singular vectors.

7:      $\boldsymbol{V}_{M1}, \boldsymbol{V}_{M2} \leftarrow \boldsymbol{V}_M\left[:,:M-1\right], \boldsymbol{V}_M\left[:,1:\right]$; $\triangleright$ 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** $\triangleright$ 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 $E_Y \in \mathbb{C}^{\left(L^{(1)}L^{(2)}\right)\times\left(N^{(1)}-L^{(1)}+1\right)\left(N^{(2)}-L^{(2)}+1\right)}$, a block Hankel matrix of the form

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

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

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

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

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

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

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

$$
E_Y = E_{\mathrm{L}} A E_{\mathrm{R}},
\tag{2.21a}
$$

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

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

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, range $(\boldsymbol{U}_M)$ = range $(\boldsymbol{E}_L)$. This implies that there is some nonsingular matrix $\boldsymbol{T} \in \mathbb{C}^{M \times M}$ such that

$$\boldsymbol{U}_M = \boldsymbol{E}_L \boldsymbol{T}. \tag{2.23}$$

Now consider the following two matrices:

$$\boldsymbol{U}_{M1} = \boldsymbol{E}_{L1} \boldsymbol{T}, \tag{2.24a}$$

$$\boldsymbol{U}_{M2} = \boldsymbol{E}_{L1} \boldsymbol{Z}_D^{(1)} \boldsymbol{T}, \tag{2.24b}$$

$$\mathbb{C}^{L^{(1)}\left(L^{(2)}-1\right) \times M} \ni \boldsymbol{E}_{L1} = \begin{bmatrix} \boldsymbol{Z}_L^{(2)} \\ \boldsymbol{Z}_L^{(2)} \boldsymbol{Z}_D^{(1)} \\ \vdots \\ \boldsymbol{Z}_L^{(2)} \left[\boldsymbol{Z}_D^{(1)}\right]^{L^{(1)}-2} \end{bmatrix} \quad \text{(c.f. Equation 2.21b).} \tag{2.24c}$$

$\boldsymbol{U}_{M1}$ 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

$$\boldsymbol{U}_{M1} - \lambda \boldsymbol{U}_{M2} = \boldsymbol{E}_{L1} \left(\boldsymbol{Z}_D^{(1)} - \lambda \boldsymbol{I}_M\right) \boldsymbol{T}. \tag{2.25}$$

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

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

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

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

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

5:     **end for**;

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

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

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

9:     $r \leftarrow 0$

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

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

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

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

14:         $r = r + 1$;

15:       **end for**

16:    **end for**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

33: **end procedure**

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

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

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

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

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

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

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

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

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

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

As Theorem 1 indicates, a local minimum of $\mathcal{F}(\theta)$ 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:

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

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 $\theta^{(k+1)} = \theta^{(k)} - \alpha^{(k)} \boldsymbol{p}^{(k)}$

❖ *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 $\|\boldsymbol{h}\| \leq \Delta^{(k)}$. 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 typical trust region method is presented in Algorithm 3 (ignoring lines 19–22, which is a custom addition, see Section 2.3.4). An initial trust 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, a solution to the following sub-problem is sought:

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

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

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, and a truncated conjugate-gradient approach, also known as the Steihaug-Toint (ST) method[27: Chapter 7], which is employed in this work (Algorithm A1). 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.

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 such that the model is more likely to 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 $\mathcal{F}\left(\theta\,|\,Y\right):\mathbb{C}^{N^{(1)}\times\cdots\times N^{(D)}}\times\mathbb{R}^{2(1+D)M}\rightarrow$ $\mathbb{R}$ is given by

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

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

The elements of the gradient vector $g\left(\theta\,|\,Y\right)\in\mathbb{R}^{2(1+D)M}$ and the Hessian matrix $H\left(\theta\,|\,Y\right)\in$ $\mathbb{R}^{2(1+D)M\times2(1+D)M}$ are then derived by taking the first and second partial derivatives of the fidelity with respect to the elements in θ, respectively:

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

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

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

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

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

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

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

The complete set of the second derivatives is given by

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

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

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

dimensions	# 1ˢᵗ derivatives	# 2ⁿᵈ 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	$2(1+D)MN^{(1)}\cdots N^{(D)}$	$\left((1+D)(2(1+D)+1)-1\right)MN^{(1)}\cdots N^{(D)}$

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.

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

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

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

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

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

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

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

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

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

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 $4(1+D)^2 M^2$ 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 $2\left(1 + D\right)MN^{(1)}...N^{(D)}$ 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

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

where $\mathrm{Var_o}\left(\boldsymbol{\phi}\right) := \mathrm{Var_o}\left(\boldsymbol{\theta}[M:2M]\right)$ 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).* $\mathrm{Var_o}\left(\boldsymbol{\phi}\right)$ *is constrained to the interval* $[0, 1]$. *If the data were not normalised, it is likely that* $\|\boldsymbol{Y} - \boldsymbol{X}\|^2$ *would dominate* $\mathrm{Var_o}\left(\boldsymbol{\phi}\right)$ *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 $\mathrm{Var_o}\left(\boldsymbol{\phi}\right)$ 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

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

2 Theory

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

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

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

unsuitable for phases. Consider as a simple example a scenario where there are two oscillators with phases $\widetilde{\boldsymbol{\phi}} = \left[\pi + \delta, \pi - \delta\right]^{\mathrm{T}}$ 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 $\boldsymbol{\phi} = \left[-\pi + \delta, \pi - \delta\right]^{\mathrm{T}}$, 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]

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

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

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

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

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

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

[†]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.

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

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]:

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

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

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

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:        $\boldsymbol{p}^{(k)} \leftarrow$ STEIHAUGTOINT $\left(\boldsymbol{Y}, \boldsymbol{\theta}^{(k)}, \Delta^{(k)}\right)$;   ▷ See Algorithm A1
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:          $\boldsymbol{\theta}^{(*)}, \boldsymbol{\epsilon}^{(*)} \leftarrow$ 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:     $\boldsymbol{\epsilon}^{(*)} \leftarrow \sqrt{\frac{\mathcal{F}\left(\boldsymbol{\theta}^{(*)}\right)\operatorname{diag}\left(\left[H(\boldsymbol{\theta}^{(*)})\right]^{-1}\right)}{\left(N^{(1)}\cdots N^{(D)}\right)-1}}$
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 **??** with $D = 1$, $M = 1$, $N^{(1)} = 64$, $f_{\mathrm{sw}}^{(1)} = 5.2\,\mathrm{Hz}\,(\Delta_t^{(1)} \approx 0.192\,\mathrm{s}^{-1})$, and $f_{\mathrm{off}}^{(1)} = 0\,\mathrm{Hz}$. The resonance was parameterised by $\boldsymbol{\theta} \in \mathbb{R}^4$ comprising $a = 1$, $\phi = 0\,\mathrm{rad}$, $f^{(1)} = 1\,\mathrm{Hz}$, $\eta^{(1)} = 0.2\,\mathrm{s}^{-1}$. White Gaussian noise was
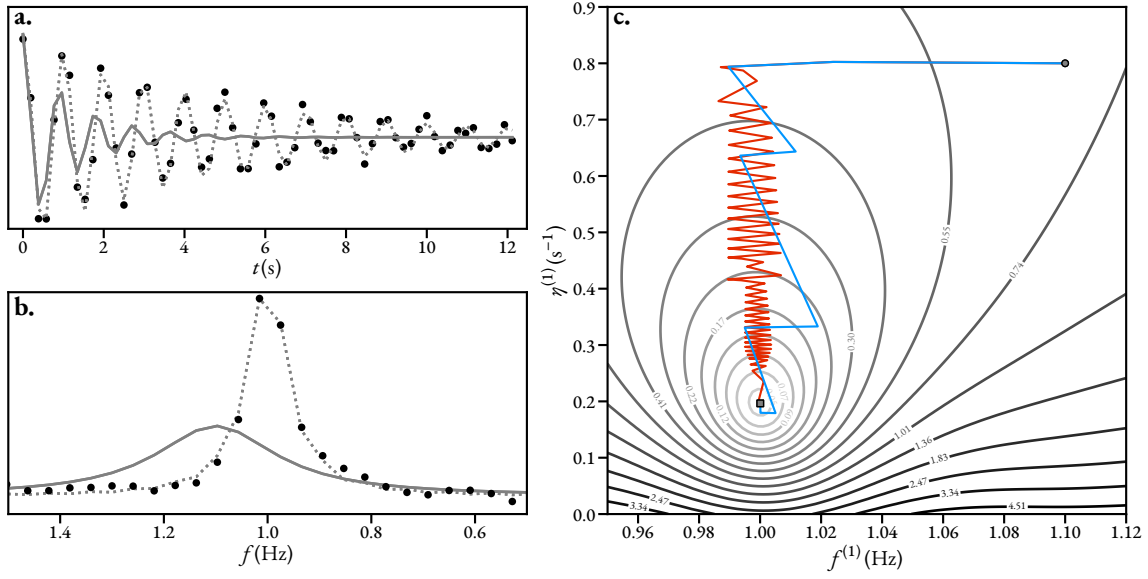
FIGURE 2.1: A visualisation of the trajectory of a 2-parameter optimisation involving a simulated FID comprising a single resonance. **a.** & **b.** Representations of the signal in the time domain and Fourier domain, respectively. Black dots: the synthetic signal Y. Solid grey line: the model generated 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 an 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 $0.8\,\text{s}^{-1}$, 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 $\epsilon \le 10^{-8}$ 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 → 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?**

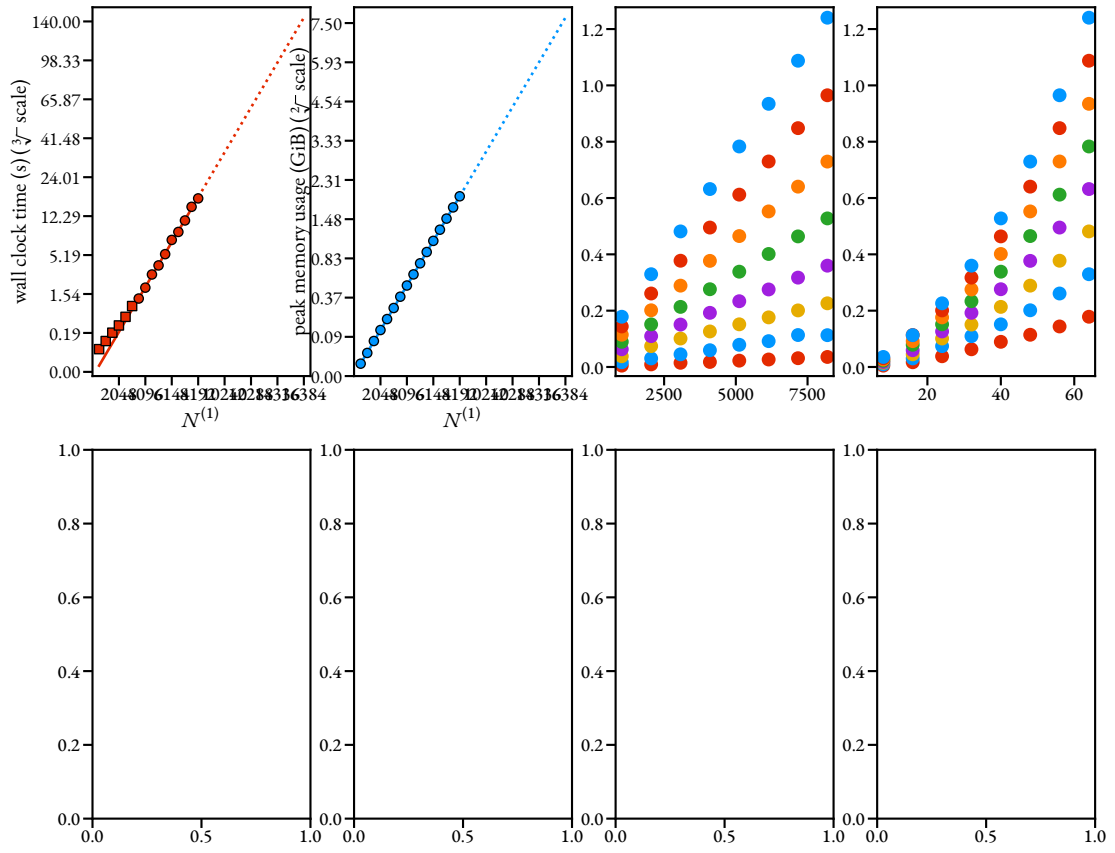❖ **Extrapolated linear fit shows that unless very big signals considered, modern PCs should be able to cope.**

FIGURE 2.2: TODO

2.5　Frequency Filtration

The previous section provides motivation for 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.

2.5.1　The virtual echo

In brief, the filtering procedure consists taking the FT of the FID, applying a band-pass filter on 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 a filtered FID to be faithfully described by the model of a summation of damped complex sinusoids, it is necessary that the spectral peaks of interest lie effectively entirely within the filter region[‡]. For absorptive Lorentzians, due to their characteristically narrow linewidths, this is straightforward. However, for broader 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.

The 1D virtual echo

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

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

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

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

$$\boldsymbol{\tau}^{(1)} = \begin{bmatrix} 0 & \Delta_t^{(1)} & \cdots & \left(N^{(1)} - 1 \right) \Delta_t^{(1)} \end{bmatrix}^{\mathrm{T}} \tag{2.55d}$$

[‡]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.

The frequency-dependence has been decomposed into its real and imaginary components. With this in mind, a conjugate pair of signals are defined:

$$\boldsymbol{\psi}_{\pm} = \boldsymbol{\gamma} \odot \left(\boldsymbol{c}^{(1)} \pm \mathrm{i} \boldsymbol{s}^{(1)} \right) + \boldsymbol{w} \equiv \Re \left(\boldsymbol{y} \right) \pm \mathrm{i} \Im \left(\boldsymbol{y} \right) \tag{2.56}$$

Two vectors $\{ \boldsymbol{t}_1, \boldsymbol{t}_2 \} \in \mathbb{C}^{2N^{(1)}}$ are constructed using the conjugate pair:

- $\boldsymbol{t}_1$ is given by $\boldsymbol{\psi}_+$ padded with zeros from below:

$$\boldsymbol{t}_1 = \begin{bmatrix} \boldsymbol{\psi}_+ \\ \boldsymbol{0} \in \mathbb{C}^{N^{(1)}} \end{bmatrix}, \tag{2.57}$$

- $\boldsymbol{t}_2$ is given by $\boldsymbol{\psi}_-$ 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 ($\cdot^{\circlearrowright(1)}$):

$$\boldsymbol{t}_2 = \begin{bmatrix} \boldsymbol{0} \in \mathbb{C}^{N^{(1)}} \\ \boldsymbol{\psi}_-^{\leftrightarrow(1)} \end{bmatrix}^{\circlearrowright(1)}, \tag{2.58}$$

The VE $\boldsymbol{y}_{\mathrm{ve}}$ is then given by $\boldsymbol{t}_1 + \boldsymbol{t}_2$, with the first element divided by 2, which is equivalent to

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

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

The 2D virtual echo

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 of 2^{D-1} signals for a D-dimensional dataset. **Maybe describe the general case in the appendix if you're bold enough?** For the 2D case, this corresponds to the pair of signals $\{ \boldsymbol{Y}_{\cos}, \boldsymbol{Y}_{\sin} \}$, given by (**??**) with $D = 2$ and $\zeta = \{ \cos(\cdot), \sin(\cdot) \}$, taking the forms (with noise neglected)

$$\boldsymbol{Y}_{\cos} = \boldsymbol{\Gamma} \odot \boldsymbol{C}^{(1)} \odot \left(\boldsymbol{C}^{(2)} + \mathrm{i} \boldsymbol{S}^{(2)} \right), \tag{2.60a}$$

$$\boldsymbol{Y}_{\sin} = \boldsymbol{\Gamma} \odot \boldsymbol{S}^{(1)} \odot \left(\boldsymbol{C}^{(2)} + \mathrm{i} \boldsymbol{S}^{(2)} \right), \tag{2.60b}$$

$$\boldsymbol{\Gamma} = \sum_m \boldsymbol{a}[m] \left(\exp \left(-\boldsymbol{\eta}^{(1)}[m] \boldsymbol{\tau}^{(1)} \right) \otimes \exp \left(-\boldsymbol{\eta}^{(2)}[m] \boldsymbol{\tau}^{(2)} \right) \right), \tag{2.60c}$$
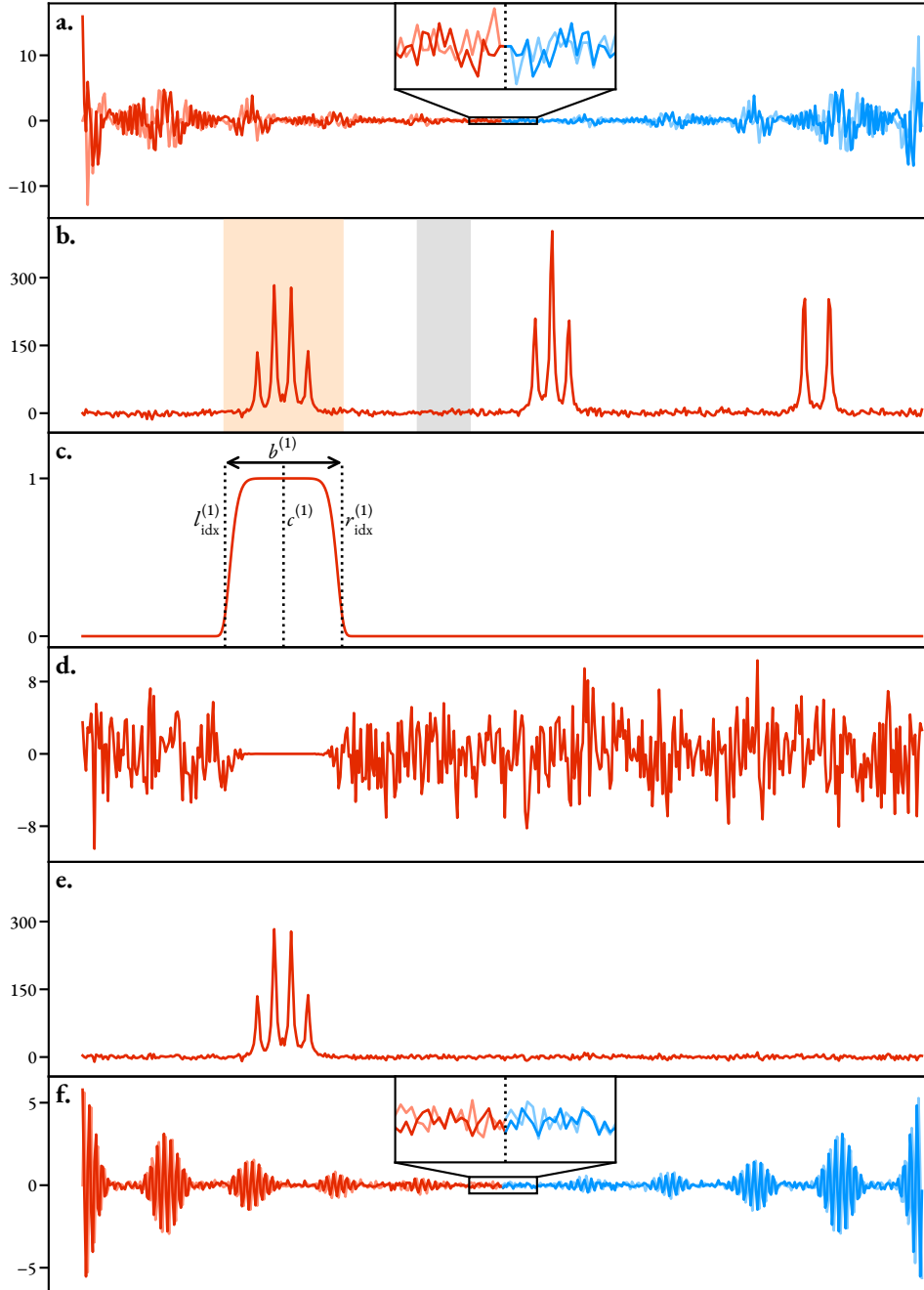
FIGURE 2.3: An illustration of the filtering procedure applied to a 1D FID. **a.** A VE $\boldsymbol{y}_{\mathrm{ve}}$, with the first and last $N^{(1)}$ points coloured red and blue, respectively. The middle of the VE is magnified to highlight its conjugate symmetry. **b.** The FT of the VE, $\boldsymbol{s}_{\mathrm{ve}}$. 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 $\widetilde{\boldsymbol{s}}_{\mathrm{ve}}$, formed by applying the super-Gaussian filter, and adding the noise vector. **f.** The IFT of the filtered spectrum, $\widetilde{\boldsymbol{y}}_{\mathrm{ve}}$, from which the final filtered signal $\widetilde{\boldsymbol{y}}$ is obtained by extracting the first $N^{(1)}$ points.

$$\boldsymbol{C}^{(1)}/\boldsymbol{S}^{(1)} = \sum_m \cos/\sin\left(2\pi\boldsymbol{f}^{(1)}\left[m\right]\boldsymbol{\tau}^{(1)}\right) \otimes \mathbf{1} \in \mathbb{C}^{N^{(2)}}, \tag{2.60d}$$

$$\boldsymbol{C}^{(2)}/\boldsymbol{S}^{(2)} = \mathbf{1} \in \mathbb{C}^{N^{(1)}} \otimes \sum_m \cos/\sin\left(2\pi\boldsymbol{f}^{(2)}\left[m\right]\boldsymbol{\tau}^{(2)}\right). \tag{2.60e}$$

Four matrices are then constructed of the form

$$\begin{aligned}
\boldsymbol{\Psi}_{\pm\pm} &= \boldsymbol{\Gamma} \odot \left(\boldsymbol{C}^{(1)} \pm^{(1)} \mathrm{i}\boldsymbol{S}^{(1)}\right) \odot \left(\boldsymbol{C}^{(2)} \pm^{(2)} \mathrm{i}\boldsymbol{S}^{(2)}\right) \\
&\equiv \Re\left(\boldsymbol{Y}_{\cos}\right) \pm^{(1)} \pm^{(2)} - \Im\left(\boldsymbol{Y}_{\sin}\right) + \mathrm{i}\left(\pm^{(1)}\Re\left(\boldsymbol{Y}_{\sin}\right) \pm^{(2)} \Im\left(\boldsymbol{Y}_{\cos}\right)\right),
\end{aligned} \tag{2.61}$$

from which the matrices $\boldsymbol{T}_{1\to4} \in \mathbb{C}^{2N^{(1)}\times 2N^{(2)}}$ are generated:

$$\boldsymbol{T}_1 = \begin{bmatrix} \boldsymbol{\Psi}_{++} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{2.62a}$$

$$\boldsymbol{T}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Psi}_{-+}^{\leftrightarrow(1)} & \mathbf{0} \end{bmatrix}^{\circlearrowright(1)}, \tag{2.62b}$$

$$\boldsymbol{T}_3 = \begin{bmatrix} \mathbf{0} & \boldsymbol{\Psi}_{+-}^{\leftrightarrow(2)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}^{\circlearrowright(2)}, \tag{2.62c}$$

$$\boldsymbol{T}_4 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Psi}_{--}^{\leftrightarrow(1,2)} \end{bmatrix}^{\circlearrowright(1,2)}. \tag{2.62d}$$

Appendix for explicit forms of the matrices $\boldsymbol{T}_{1-4}$ The virtual echo is then given by $\boldsymbol{Y}_{\mathrm{ve}} = \sum_{i=1}^4 \boldsymbol{T}_i$, with the first row and column divided by two. For a full outline of the 2D filtering procedure, see Algorithm A3.

2.5.2 The filtering process

Having constructed a virtual echo $\boldsymbol{Y}_{\mathrm{ve}} \in \mathbb{C}^{2N^{(1)}\times\cdots\times2N^{(D)}}$, a spectrum with absorption Lorentzians is produced with $\boldsymbol{S}_{\mathrm{ve}} = \mathrm{FT}\left(\boldsymbol{Y}_{\mathrm{ve}}\right)$. To filter the spectrum, it is subjected to element-wise multiplication with a super-Gaussian function. The super-Gaussian is defined by a centre $c^{(d)} \in \mathbb{R} : 0 < c^{(d)} < 2N^{(d)}$ and a bandwidth $b^{(d)} \in \mathbb{N} : b^{(d)} < 2N^{(d)}$ in each dimension (panel c of Figure 2.3):

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

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

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 $\boldsymbol{S}_{\mathrm{ve}}$ 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 $\boldsymbol{S}_{\mathrm{ve}}$ 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, $\boldsymbol{W}_{\sigma^2} \in \mathbb{R}^{2N^{(1)} \times \cdots \times 2N^{(D)}}$. The filtered spectrum is then given by

$$\widetilde{\boldsymbol{S}}_{\mathrm{ve}} = \boldsymbol{S}_{\mathrm{ve}} \odot \boldsymbol{G} + \boldsymbol{W}_{\sigma^2} \odot (\mathbf{1} - \boldsymbol{G}) . \tag{2.64}$$

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 ($\boldsymbol{G}\left[n^{(1)}, \cdots, n^{(D)}\right] = 1$), the noise is nullified. See panel d of Figure 2.3 for an example.

After filtering, $\widetilde{\boldsymbol{S}}_{\mathrm{ve}}$ 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 $\widetilde{\boldsymbol{Y}} \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}$:

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

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:

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

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

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

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

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

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).

Spectrum slicing

Thus far, the method described is able to reduce the model order of a given signal, however the signal still comprises the same number of points. However it is clear that there are a large number of points outside the region of interest in $\widetilde{\boldsymbol{S}}_{\text{ve}}$ that do not possess any meaningful information. Discarding such points should then lead to filtered FID with the same information about the resonances of interest, but with far fewer points. A slicing ratio is defined, $\xi \in \mathbb{R} : \xi > 1$, which dictates the left and right indices at which the spectrum should be sliced in each dimension:

$$l_{\text{slice}}^{(d)} = \begin{cases} c_{\text{idx}}^{(d)} - \left\lfloor \frac{b^{(d)}\xi}{2} \right\rfloor & \text{if } \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.68a}$$

$$r_{\text{slice}}^{(d)} = \begin{cases} c_{\text{idx}}^{(d)} + \left\lceil \frac{b^{(d)}\xi}{2} \right\rceil & \text{if } \leq 2N^{(d)} - 1 \\ 2N^{(d)} - 1 & \text{otherwise} \end{cases} \tag{2.68b}$$

The filtered spectrum is then sliced accordingly:

$$\widetilde{\boldsymbol{S}}_{\text{ve, slice}} = \widetilde{\boldsymbol{S}}_{\text{ve}} \left[l_{\text{slice}}^{(1)} : r_{\text{slice}}^{(1)} + 1, \cdots, l_{\text{slice}}^{(D)} : r_{\text{slice}}^{(D)} + 1\right]. \tag{2.69}$$

The equivalent process is applied to $\widetilde{\boldsymbol{S}}_{\text{ve,slice}}$ to generate the final filtered sub-FID: IFT followed by slicing in half in each dimension. It is also necessary to scale the signal by the ratio of the number of points in the sliced spectrum and it's unsliced counterpart.

$$\widetilde{\boldsymbol{Y}} = \prod_{d=1}^{D} \left(\frac{r_{\text{slice}}^{(d)} - l_{\text{slice}}^{(d)}}{2N^{(d)}}\right) \text{IFT}\left(\widetilde{\boldsymbol{S}}_{\text{ve, slice}}\right)\left[0 : N_{\text{slice}}^{(1)}, \cdots, 0 : N_{\text{slice}}^{(D)}\right] \tag{2.70a}$$

$$N_{\text{slice}}^{(d)} = \left\lfloor \frac{r_{\text{slice}}^{(d)} - l_{\text{slice}}^{(d)}}{2} \right\rfloor \tag{2.70b}$$

Finally, the effective sweep widths and transmitter offsets will have been altered by this process. The corrected values can be computed using

$$f^{(d)}_{\text{sw,slice}} = \frac{r^{(d)}_{\text{slice}} - l^{(d)}_{\text{slice}}}{2N^{(d)} - 1} f^{(d)}_{\text{sw}} \tag{2.71a}$$

$$f^{(d)}_{\text{off,slice}} = f^{(d)}_{\text{off}} + \frac{f^{(d)}_{\text{sw}}}{2}\left(1 - \frac{l^{(d)}_{\text{slice}} + r^{(d)}_{\text{slice}}}{2N^{(d)} - 1}\right) \tag{2.71b}$$

2.6   Summary

The matrix pencil-based methods described above are well established as effective procedures for parametric estimation of time-domain signals in a number of disciplines, including radar detection[42], acoustics[**TODO**], and NMR[37], with specific application to relaxometry being a recently introduced application [43, 44]. Due to considerable advances computational processing power since the introduction of the technique, estimates can be acquired from NMR signals in reasonable times. One notable downside of the technique that has been realised while assessing its effectiveness in parametrising NMR FIDs is its propensity to return oscillators with unexpected phase behaviour, especially in scenarios involving resonances with close frequencies. For this reason, estimating phase-corrected FIDs, using the result of the MPM as an initial guess to feed into a phase-variance penalised NLP routine was proposed as a means of improving parameter estimates. The theory underpinning the procedure has been explored in this chapter.

The computational burden of running the procedure is large and often intractable for complete NMR signals, which often comprise thousands of samples, and at least hundreds of contributing resonances. This has been illustrated through profiling both the CPU time and the peak memory requirements for the 1D and 2D methods. **Say more when completed?** For this reason, a method to break FIDs into frequency-filtered sub-signals, by filtering spectra derived using virtual echoes, is introduced. This method enables to formation of signals with far fewer resonances and samples as compared to the full signal.

Algorithm 4 provides an overview of the principal steps involved in the 1D and 2D estimation procedures. Detailed algorithms for each step are presented elsewhere in this text.

Having established an estimation routine, the next chapter focusses on its performance, as well as applications which are possible through parametric estimation.

Algorithm 4 An overview of the estimation procedure outlined in this work, for the consideration of 1D and 2D NMR signals.

1: **procedure** ESTIMATE1D($\boldsymbol{y} \in \mathbb{C}^{N^{(1)}}, \boldsymbol{r}_{\text{interest}} \in \mathbb{R}^2, \boldsymbol{r}_{\text{noise}} \in \mathbb{R}^2, \mathcal{M} \in \mathbb{N}_0$)
2:     $\widetilde{\boldsymbol{y}} \leftarrow$ FILTER1D $\left(\boldsymbol{y}, \boldsymbol{r}_{\text{interest}}, \boldsymbol{r}_{\text{noise}}\right)$;   ▷ Algorithm A2
3:     **if** $\mathcal{M} = 0$ **then**
4:        $\mathcal{M} \leftarrow$ MDL $\left(\widetilde{\boldsymbol{y}}\right)$;   ▷ **TODO**
5:     **end if**
6:     $\theta^{(0)} \leftarrow$ MPM $\left(\widetilde{\boldsymbol{y}}, \mathcal{M}\right)$;   ▷ Algorithm 1
7:     $\theta^{(*)}, \epsilon^{(*)} \leftarrow$ NLP $\left(\widetilde{\boldsymbol{y}}, \theta^{(*)}\right)$;   ▷ Algorithm 3
8:     **return** $\theta^{(*)}, \epsilon^{(*)}$;
9: **end procedure**

10: **procedure** ESTIMATE2D($\boldsymbol{Y}_{\cos} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{Y}_{\sin} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{R}_{\text{interest}} \in \mathbb{R}^{2 \times 2}, \boldsymbol{R}_{\text{noise}} \in \mathbb{R}^{2 \times 2}, \mathcal{M} \in \mathbb{N}_0$)
11:     $\widetilde{\boldsymbol{Y}} \leftarrow$ FILTER2D $\left(\boldsymbol{Y}_{\cos}, \boldsymbol{Y}_{\sin}, \boldsymbol{R}_{\text{interest}}, \boldsymbol{R}_{\text{noise}}\right)$;   ▷ Algorithm A3
12:     **if** $\mathcal{M} = 0$ **then**
13:        $\mathcal{M} \leftarrow$ MDL $\left(\widetilde{\boldsymbol{Y}}[:,0]\right)$;   ▷ Run the MDL on the first direct-dimension slice.
14:     **end if**
15:     $\theta^{(0)} \leftarrow$ MMEMPM $\left(\widetilde{\boldsymbol{Y}}, \mathcal{M}\right)$;   ▷ Algorithm 2
16:     $\theta^{(*)}, \epsilon^{(*)} \leftarrow$ NLP $\left(\widetilde{\boldsymbol{Y}}, \theta^{(*)}\right)$;   ▷ Algorithm 3
17:     **return** $\theta^{(*)}, \epsilon^{(*)}$;
18: **end procedure**

Results and Applications   3

3.1   1D Datasets

3.2   2D Datasets

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

Two key features which the 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 (SNR $\propto \mathrm{NS}^{1/2}$). 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[45, 46, 47], 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 FID acquisition, using well known schemes such as WALTZ-16[48, 49] and GARP[50]. 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 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, named computer-assisted undiminished-sensitivity
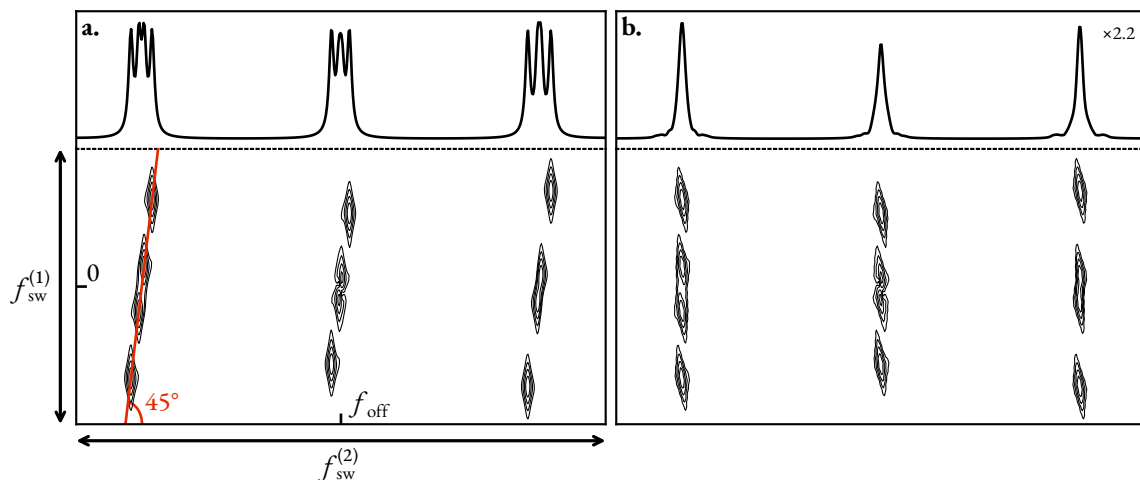
FIGURE 3.1: a. Contour plot of an absolute value mode 2DJ spectrum for an AMX spin system, produced by applying sine-bell apodisation on a hypercomplex FID before a FT in both dimensions. Each multiplet lies a long a line at 45° to the $F^{(1)}$ and $F^{(2)}$ axes. Note that this line often appears to 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.

protocol for ideal decoupling (CUPID) is presented.

3.3.1   An Overview of Pure Shift NMR

The 2D J-resolved Experiment

The 2DJ experiment[13, 51] 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 (**??**) with $D = 2$ and $\zeta = \exp(\mathrm{i}\cdot)$, i.e.

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

(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 spec-

tra with absorption mode lineshapes. In many multidimensional experiments, changing the phase of certain 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 Y_{2DJ} produces a spectrum 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 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

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

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

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

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

$$s_{PS}\left[n^{(2)}\right] = \sum_{n^{(1)}=0}^{N^{(1)}-1} S_{tilt}\left[n^{(1)}, n^{(2)}\right]. \tag{3.3}$$

Shearing and summing 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 $\Re(|S_{2DJ}|)$). 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[52: Section 3.3.5], or pseudo-echo reshaping[53], 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[55]. Such an element excites a given spin only in a narrow range of heights in the sample, as

---

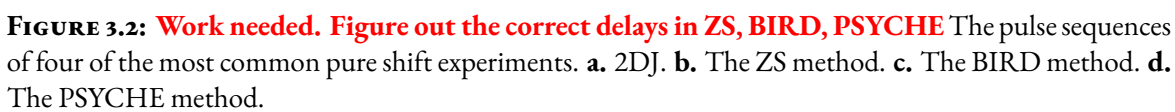*Conventionally, a R-SNOB pulse is used[54].

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.

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 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[56]. 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[57].

The BIRD Method

The bilinear rotation decoupling (BIRD) pulse sequence element[58, 59], 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[60].

PSYCHE

TODO Original paper[61], tutorial paper[62], PSYCHE-2DJ[63].

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)[64], in which the parameters of each indirect-dimension FID

[†]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).

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

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

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

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

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

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 $-t^{(2)}$ [65].

3.3.2   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 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:

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

$\forall n^{(2)} \in \{0, \cdots, N^{(2)} - 1\}$. 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.
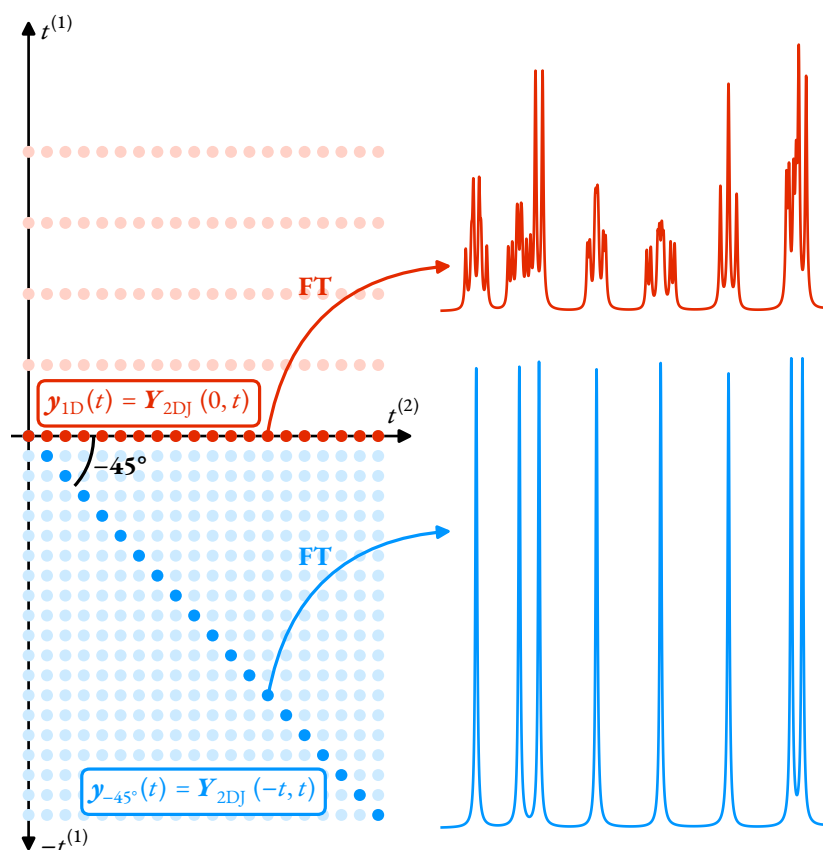
FIGURE 3.3: An illustration of the reasoning behind the name "$-45°$ signal". The pale red dots denote a typical 2DJ FID, where the amount and rate of sampling in the direct dimension is greater than in the indirect dimension (i.e. $N^{(1)} \ll N^{(2)}$ and $f_{sw}^{(1)} \ll f_{sw}^{(2)}$). The bright red dots correspond to the first direct-dimension signal $\boldsymbol{Y}_{2DJ}(0, t^{(2)})$, 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 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, i.e. $\epsilon = \min\left(f_{\mathrm{sw}}^{(1)} / 2N^{(1)}, f_{\mathrm{sw}}^{(2)} / 2N^{(2)} \right)$. 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 A5 provides a routine that can be used for multiplet prediction.

Removal of spurious oscillators

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

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

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

(ii) $\boldsymbol{Y}_{\mathrm{ve}}$ is subjected to FT along the direct dimension to produce the spectrum $\boldsymbol{S}_{\mathrm{ve}}$ (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):

$$\boldsymbol{G} = \boldsymbol{1} \otimes \boldsymbol{g}^{(2)}, \tag{3.9}$$
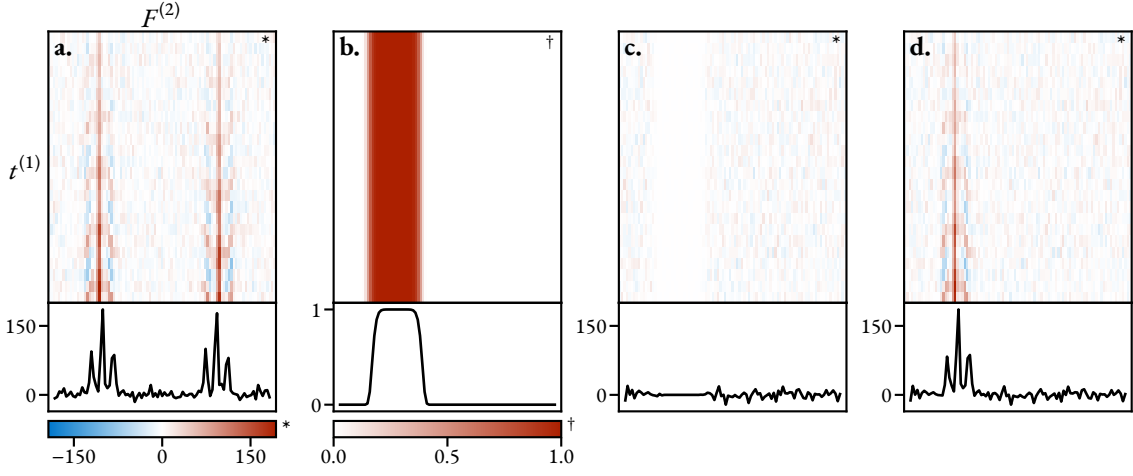
FIGURE 3.4: An illustration of the filtering procedure for 2DJ data. For each panel is a heat-map of the full 2D signal, as well as a plot underneath of the first slice of the signal in the direct dimension. **a.** The spectrum S_{ve}, **b.** Super-Gaussian filter G, **c.** Additive noise, attenuated by the super-Gaussian, $W_{\sigma^2} \odot (1 - G)$, **d.** Filtered spectrum $\widetilde{S}_{ve}$ Panels **a.–d.** are analogous to panels **b.– e.** in Figure 2.3 for the 1D case.

where $1 \in \mathbb{R}^{N^{(1)}}$ is a vector of ones, and $g^{(2)}$ 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 S_{ve} which is devoid of peaks, and generating an array $W_{\sigma^2} \in \mathbb{R}^{N^{(1)} \times 2N^{(2)}}$ 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 $\widetilde{S}_{ve}$ (panel d).

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

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

3.3.3   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[66]. 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.

For additional details relating to generation of the simulated datasets, see section A.3 in the Appendix.
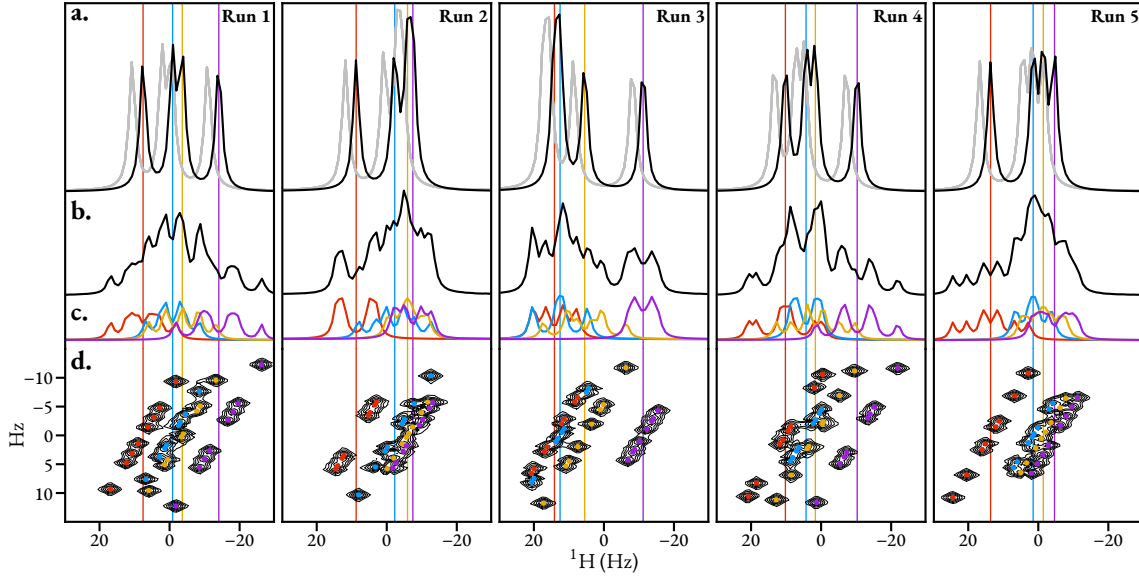
Figure 3.5: The result of applying CUPID to 5 instances of simulated 2DJ datasets with 4 heavily over-lapping multiplet structures. **a.** Black: pure shift spectrum generated by CUPID. Grey: 1D spectrum simulated with Spinach, using the same spin system as was used to produce the 2DJ dataset, but with all scalar couplings sets to 0 Hz. This has been offset slightly for clarity. **b.** 1D spectrum of the dataset. **c.** Multiplet structures predicted, using a threshold $\epsilon = f_{sw}^{(2)}/2N^{(2)} \approx 0.49$ Hz. **d.** Contour plot of the 2DJ spectrum in absolute-value mode. Coloured points denote the frequencies of oscillators in the estimation result. Coloured vertical lines denote the predicted central frequencies of each multiplet structure.

"Four Multiplets"

A series of simulated ^{1}H 2DJ datasets were generated such that within a known region of the spectrum, four ddd multiplet structures with significant overlap existed. To achieve this, a spin-system with 7 spins was formed, with the spins divided into 2 subsets:

- ❖ 4 of the spins (the "estimated spins") were assigned random resonance frequencies sampled from $\mathcal{U}(-20 \, \text{Hz}, 20 \, \text{Hz})$.

- ❖ The remaining 3 spins (the "coupling spins"), were coupled to each of the estimated spins, with the values of the couplings randomly sampled from $\mathcal{U}(-10 \, \text{Hz}, 10 \, \text{Hz})$. The coupling spins were given chemical shifts such that they lay far from the estimated spins in the spectrum (i.e. their frequencies were $\gg 20$ Hz).

AWGN noise was added to the FID, with a target SNR of 30 dB. A filtered sub-FID containing only the signals from the estimated spins was then generated using the filtering procedure described above, with $l_{\text{Hz}}^{(2)} = 30$ Hz, $r_{\text{Hz}}^{(2)} = -30$ Hz, and $\xi = 1.1$. The resulting sub-FID was expected to comprise 32 (4×2^3) oscillators. To assess the estimation procedure's ability, a random integer from the range 33 – 40 was selected as the initial number of oscillators. Hence, the initial guess from the MMEMPM would comprise an excessive number of oscillators. The FID was subjected
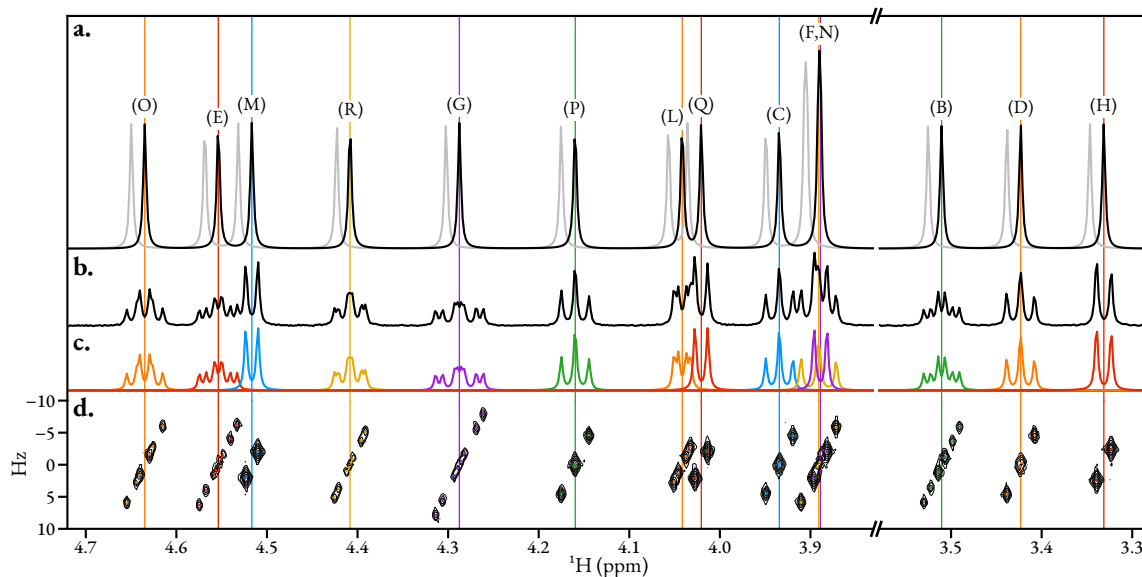
FIGURE 3.6: Application of CUPID on a simulated sucrose 2DJ dataset. **a.** Black: the spectrum generated from FT of the $-45°$ signal. Grey: the spectrum of a simulated dataset with the same chemical shifts, with all scalar couplings set to 0 Hz. **b.** Conventional 1D spectrum. **c.** Multiplet structures assigned ($\epsilon \approx 0.27$ Hz). **d.** Contour plot of the absolute value mode 2DJ spectrum, with the locations of assigned oscillators given as coloured points.

to estimation, yielding the result vector $\theta^{(*)}$. Spurious oscillators were checked for, using the criteria outlined above, with the threshold for multiplet assignment set to half the spectral resolution in the direct dimension: $\epsilon = f_{sw}^{(2)}/2N^{(2)}$. If spurious oscillators were found, these were removed, and NLP was run on the updated set of parameters.

Figure 3.5 illustrates the result achieved for 5 separate runs of the procedure described above. For each FID generated, the method was effective at producing an estimation result with 32 oscillators, as desired, despite the excessive number that were present in $\theta^{(0)}$. Most of the excessive oscillators were purged from $\theta^{(0)}$ through the NLP procedure. When spurious oscillators did remain[‡], they were then detected when checking for spurious oscillators and subsequently removed. The pure-shift spectra generated using CUPID closely agree with pure-shift 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.

"Sucrose"

As a second example of applying CUPID on simulated data, the chemical shifts and isotropic scalar couplings associated with a Gaussian[67] density functional theory (DFT) calculation of sucrose in a vacuum[§] were used to construct a 2DJ dataset. AWGN was added with a target SNR

[‡]for 2 of the 5 datasets, the result after NLP comprised 33 oscillators
[§]It is well known that isotropic chemical shift calculations using DFT are typically very inaccurate. The resulting spectrum is not typical of sucrose in the liquid state, though this doesn't really matter for assessing the performance

of 20 dB. The CUPID procedure was applied to filtered sub-FIDs such that the resonances from all 22 spins were considered, though only the regions of the dataset with the most interesting multiplet structures are presented in Figure 3.6.

The estimation technique successfully assigned multiplet structures for all 22 multiplets in the dataset, including structures derived from two spins (F & N) with a 0.6 Hz difference in resonance frequency, approaching the spectral resolution in the direct dimension (0.537 Hz). The pure-shift spectrum generated via the −45° signal again showed close agreement with a 1D spectrum simulated using the same chemical shifts, with scalar couplings set to 0 Hz. There are particular multiplets where the number of oscillators fit using the estimation routine was less than the true number. Examples of this phenomenon are exhibited in the estimates of the multiplets for spins B & O, which are both ddd structures. The scalar couplings involved meant that certain oscillators were of such similar frequencies that they were separated by significantly less than the spectral resolution, and thus deconvolving these was unrealistic. For example, there are two pairs of peaks in the spin-B multiplet which lie only 0.085 Hz apart. Under-fitting in this case had a negligible impact on the final pure shift spectrum. However there are circumstances which will be seen in the experimental examples below where more blatant cases under-fitting lead to the generation of peaks in the pure shift spectrum which are noticeably broadened (*vide infra*).

Quinine

Figure 3.7 illustrates the result of applying CUPID on a dataset generated from a sample comprising quinine in CD_3OD, with all non-aromatic protons considered. The method successfully generated a pure shift spectrum with distinct peaks for each ^{1}H environment. This example also highlights an added benefit of using CUPID: the ability to suppress unwanted signals in the final spectrum. In this example, an intense, broad singlet at around 4.89 ppm was detected (see the green peak at this frequency in panel c). The singlet was due to the presence of water in the sample and was a hindrance due to it overlapping heavily with the multiplet structure corresponding to spin D. To obtain a clean singlet for spin D in the pure shift spectrum, the oscillator corresponding to the water signal was simply neglected from $\theta^{(*)}$ in generating the −45° signal. **Find reference for work talking about using estimation for solvent suppression.**

As eluded to already, a few of the peaks in the pure-shift spectrum are rather broad on account of the estimation routine under-fitting the relevant multiplet structure. The most notable example of this phenomenon in the quinine example comes from the peak for spin G, where sig-inifcant overlap with spin F has likely compunded the task of accurately estimating the FID. With fewer than the true number of oscillators fitting a given multiplet structure, the NLP routine will compensate by giving the oscillators it does have at its disposal large amplitudes and damping fac-
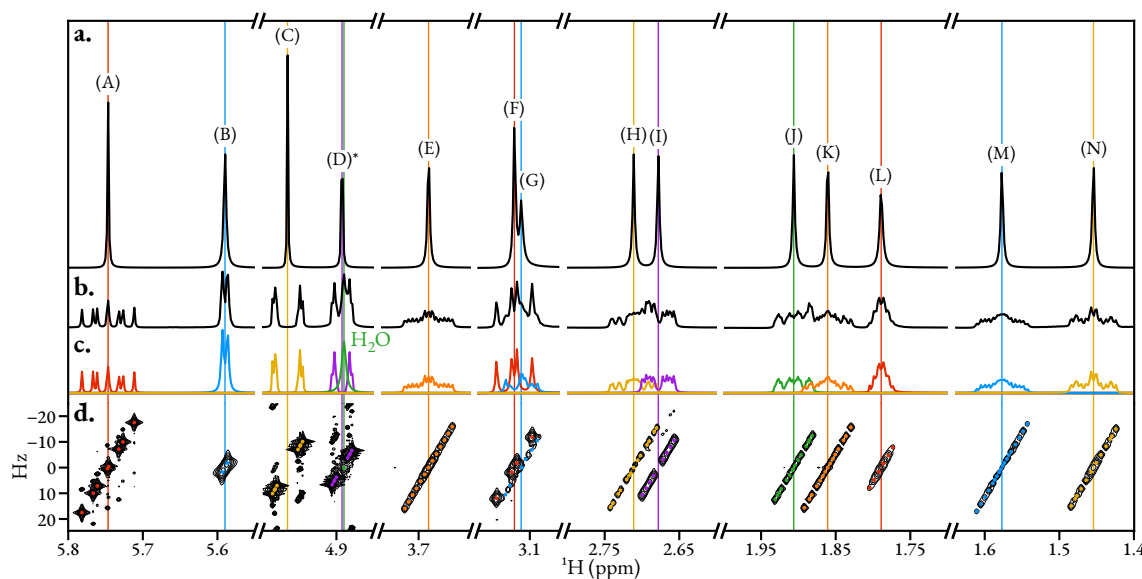
_____

of CUPID.

FIGURE 3.7: Application of CUPID on the non-aromatic regions of a quinine 2DJ dataset. **a.** The spectrum generated from FT of the −45° signal, with the green signal arising from water at about 4.89 ppm neglected. **b.** Conventional 1D spectrum. **c.** Multiplet structures assigned ($\epsilon = f_{sw}^{(2)}/N^{(2)} \approx 0.92$ Hz). **d.** Contour plot of the absolute value mode 2DJ spectrum, with the locations of assigned oscillators given as coloured points.

tors, so that they can reasonably fit multiple similar-frequency oscillators. While affecting peak linewdiths, this feature does not tend to significantly affect the integrals of the pure shift peaks. **Should probably calculate the integrals of these...**

Dexamethasone

Double check mp thold

Figure 3.8 shows the result of applying CUPID on a dataset acquired from a sample dexamethasone in DMSO-d_6. A pure-shift spectrum was also acquired using the TSE-PSYCHE experiment[62, 68] for comparison. CUPID generated a pure-shift spectrum with overall excellent agreement with the PSYCHE spectrum. Certain multiplet structures in the spectrum exhibit splitting in $F^{(2)}$, on account of heteronuclear couplings to ^{19}F. Most notable are those derived from spins D, H & O. For the D multiplet, the magnitude of $J_{1_H,19_F}$ is very small such that assigning these to separate oscillators is infeasible. For the spin N multiplet, two separate structures were successfully assigned (see the orange and green multiplets around 2.1 ppm). The estimation routine was unsuccessful at accurately estimating the structure associated with spin H, where a severe under-fitting occurred. An under-fitting of this structure even occurred when the estimation was re-run using considerable over-estimation of the model order, with most oscillators in the initial guess being purged during the NLP procedure. The result of under-fitting, as already discussed, leads to peaks in the final CUPID spectrum being broadened, with the spin H mul-
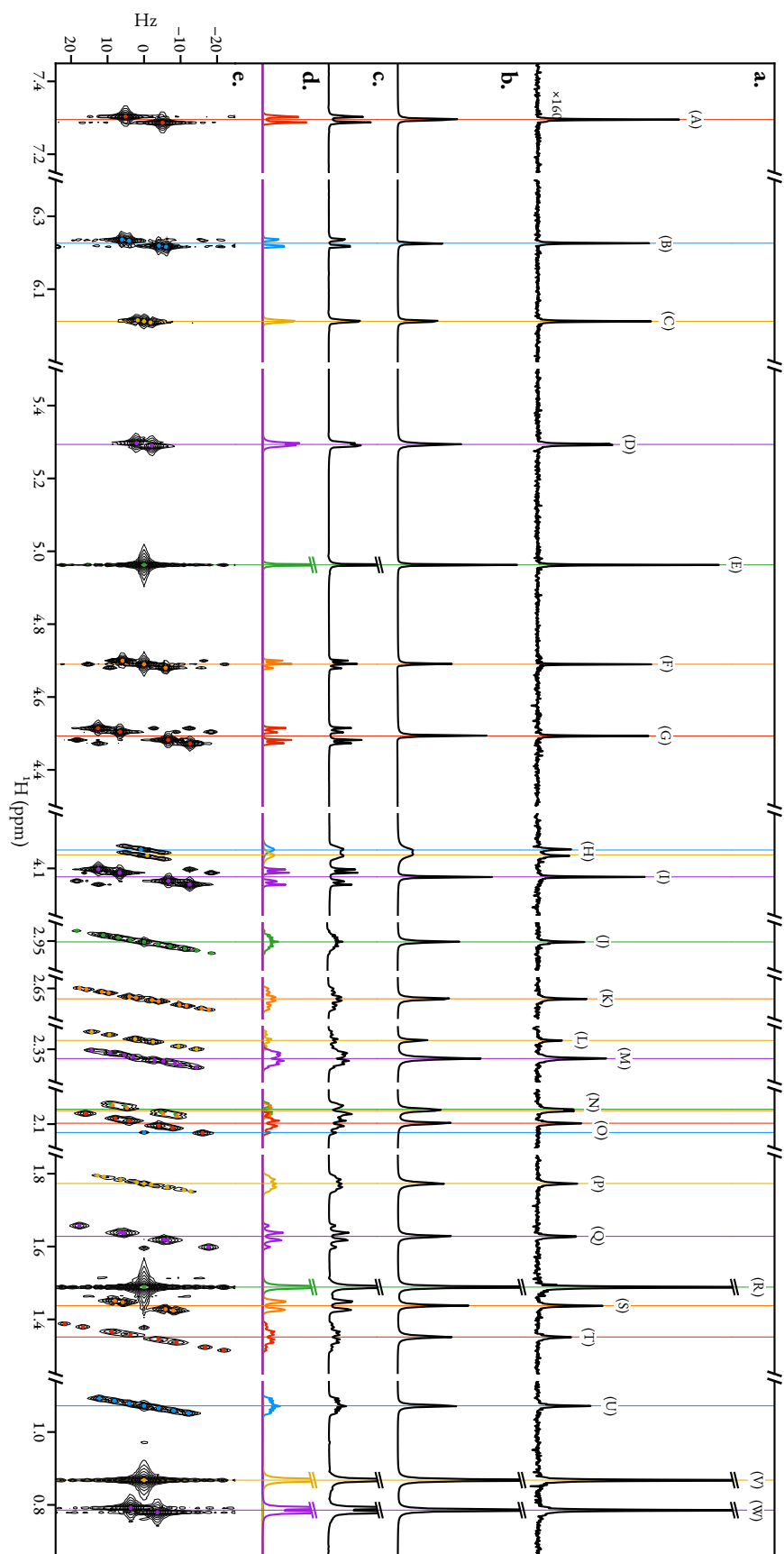
FIGURE 3.8: Application of CUPID on dexamethasone 2DJ dataset. **a.** triple spin echo PSYCHE (TSE-PSYCHE) spectrum of the sample. **b.** The spectrum generated from FT of the −45° signal. **c.** Conventional 1D spectrum. **.** Multiplet structures assigned ($\epsilon = f_{sw}^{(2)}/N^{(2)} \approx 0.92\,\text{Hz}$). **d.** Contour plot of the absolute value mode 2DJ spectrum, with the locations of assigned oscillators given as coloured points.

tiplet providing an extreme example of this. The most downfield peaks in the CUPID spectrum (corresponding to aromatic and hydroxyl protons) also appear to be noticeably broadened relative to their PSYCHE equivalents. This is also probably due to under-fitting of the relevant multiplet structures, though to a far less noticeable extent than for spin H. **Any other reason why this might be so?**

3.4 Amplitude-attenuated Datasets

There are a number of 2D 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 across increments. These include experiments for the determination of translational diffusion rates and relaxation properties such as longitudinal and transverse decoherence rates. Here, an extension to the 1D estimation technique is described, facilitating the determination of these properties.

After a description of relevant experiments, the estimation procedure is p

3.4.1 Relaxation experiments

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

$$\theta_1 = I_\infty = \lim_{\tau \to \infty} x \tag{3.11a}$$

$$\theta_2 = T_1 \tag{3.11b}$$

$$\boldsymbol{I}\left(\boldsymbol{\theta}, \boldsymbol{\tau}\right) = I_\infty \left[1 - 2\exp\left(-\frac{\tau}{T_1}\right) \right], \tag{3.12}$$

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

$$\frac{\partial I}{\partial I_\infty} = 1 - 2\exp\left(-\frac{\tau}{T_1}\right) \tag{3.13a}$$

$$\frac{\partial I}{\partial T_1} = -\frac{2I_\infty \tau}{T_1^2} \exp\left(-\frac{\tau}{T_1}\right) \tag{3.13b}$$

$$\frac{\partial^2 I}{\partial I_\infty^2} = 0 \tag{3.13c}$$

$$\frac{\partial^2 I}{\partial I_\infty \partial T_1} = \frac{\partial^2 I}{\partial T_1 \partial I_\infty} = -\frac{2\tau}{T_1^2} \exp\left(-\frac{\tau}{T_1}\right) \tag{3.13d}$$

$$\frac{\partial^2 I}{\partial T_1^2} = \frac{2 I_\infty \tau}{T_1^3} \exp\left(-\frac{\tau}{T_1}\right)\left(2 - \frac{\tau}{T_1}\right) \tag{3.13e}$$

3.4.2   Diffusion experiments

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[69] (Figure 3.9.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 $\omega_{\mathrm{PFG}} = -\gamma g z$, 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:

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

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 $\beta = -\gamma g\left(z + \Delta_z\right)\delta$:

$$\xrightarrow{\mathrm{PFG}} -M_y \cos(\alpha)\cos(\beta) + M_x \cos(\alpha)\sin(\beta) - M_x \sin(\alpha)\cos(\beta) - M_y \sin(\alpha)\sin(\beta)$$

$$= -M_y \cos(\gamma g \delta \Delta_z) - M_x \sin(\gamma g \delta \Delta_z),$$

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[70], 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:

$$a(g) = a_0 \exp\left(-\gamma^2 \delta^2 g^2 D \left(\Delta - \frac{\delta}{3}\right)\right), \tag{3.14}$$

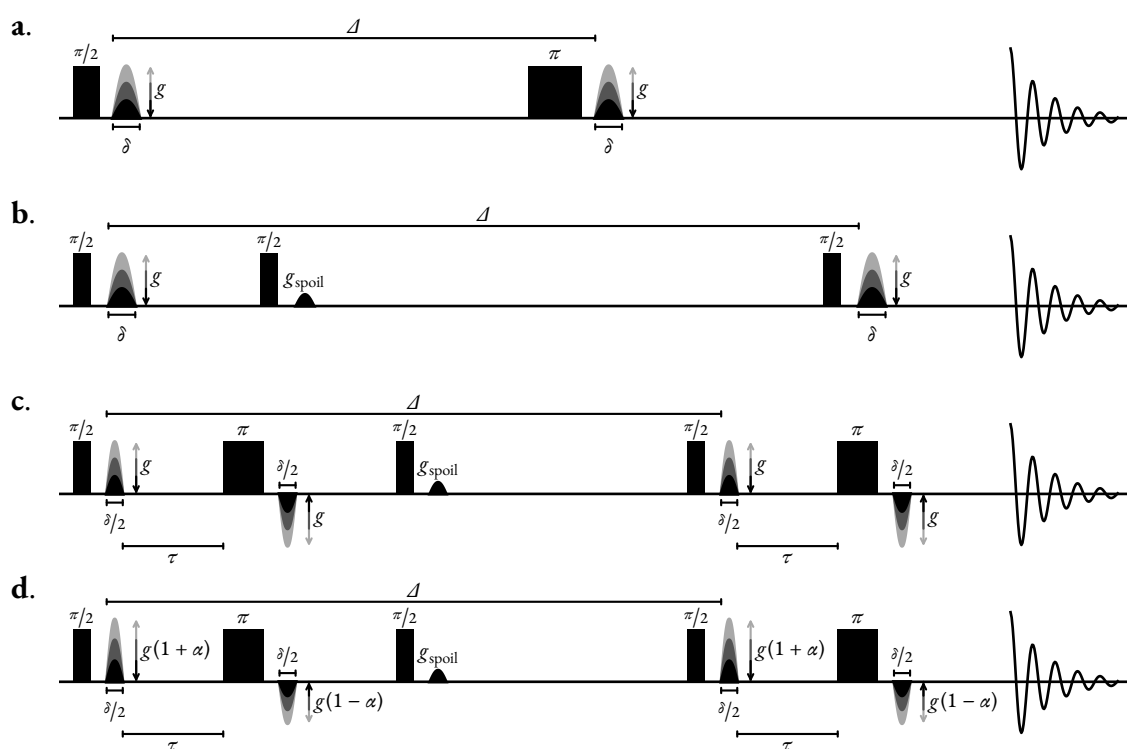**FIGURE 3.9:** Pulse sequences used for the determination of translational diffusion constants. **a.** PGSE, **b.** PGSTE, **c.** PGSTEBP, **d.** One-shot DOSY. RF pulses are denoted by solid rectangles. Diffusion-encoding gradients are denoted by sine-bell shapes with varying shades, indicating that 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), Δ 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 3.14 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)[71] (Figure 3.9.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[72, 73], 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 3.9.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[74] (Figure 3.9.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:

$$a(g) = a_0 \exp\left(-cg^2 D\right) \tag{3.15}$$

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[75]. 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:

$$c = \gamma^2 \delta^2 \sigma^2 \varDelta'. \tag{3.16}$$

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

Monopolar gradients (PGSE, PGSTE)	$\varDelta + 2\left(\kappa - \lambda\right)\delta,$	(3.17a)
Bipolar gradients (PGSTEBP)	$\varDelta + \dfrac{\left(2\kappa - 2\lambda - 1\right)\delta}{4} - \dfrac{\tau}{2},$	(3.17b)
One-shot	$\varDelta + \dfrac{\left(\kappa - \lambda\right)\left(\alpha^2 + 1\right)\delta}{2} + \dfrac{\left(\delta + 2\tau\right)\left(\alpha^2 - 1\right)}{4}.$	(3.17c)

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:

$$S(\epsilon) = \int_0^\epsilon s\left(\epsilon'\right) d\epsilon' \quad \forall \epsilon \in [0,1]. \tag{3.18}$$

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

$$S[n] = \frac{1}{n} \sum_{i=0}^{n} s[i] \quad \forall n \in \left\{0, \cdots, N_g - 1\right\}, \tag{3.19}$$

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

$$\sigma = S(1), \tag{3.20a}$$

$$\lambda = \frac{1}{\sigma} \int_0^1 S(\epsilon) d\epsilon, \tag{3.20b}$$

$$\kappa = \frac{1}{\sigma^2} \int_0^1 S^2(\epsilon) d\epsilon, \tag{3.20c}$$

with their discrete counterparts being

$$\sigma = S\left[N_g - 1\right] \tag{3.21a}$$

$$\lambda = \frac{1}{\sigma N_g} \sum_{n=0}^{N_g-1} S[n] = \frac{1}{\sigma N_g} \sum_{n=0}^{N_g-1} \frac{1}{n} \sum_{i=0}^{n} s[i] \tag{3.21b}$$

$$\kappa = \frac{1}{\sigma^2 N_g} \sum_{n=0}^{N_g-1} S^2[n] = \frac{1}{\sigma^2 N_g} \sum_{n=0}^{N_g-1} \frac{1}{n^2} \left(\sum_{i=0}^{n} s[i]\right)^2 \tag{3.21c}$$

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 3.14 comes from plugging Equation 3.17a into Equation 3.16, 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 3.16 will be extremely important for accurate measurements of D.

3.4.3   Methodology

Outline of the problem

All the experiments described above can be described in a general fashion. Suppose the experiment is run with $K \in \mathbb{N}$ increments, such that there is a vector $\boldsymbol{p} \in \mathbb{R}^K$ which gives the experimental variable for each increment. The complete FID that results from the experiment is expected to take the form $\boldsymbol{Y} \in \mathbb{C}^{K \times N^{(1)}}$ in which the value of the experimental parameter attenuates the amplitudes of the contributing resonances:

$$Y\left[k, n^{(1)}\right] = \sum_{m=0}^{M-1} \boldsymbol{A}\left[k, m\right] \exp\left(\mathrm{i}\boldsymbol{\phi}\left[m\right]\right) \exp\left(\left(2\pi\mathrm{i}\left(\boldsymbol{f}^{(1)}\left[m\right] - f_{\mathrm{off}}^{(1)}\right) - \boldsymbol{\eta}^{(1)}\left[m\right]\right) n^{(1)} \Delta_t^{(1)}\right).$$

(3.22)

$\boldsymbol{A} \in \mathbb{R}^{K \times M}$ is a matrix of the amplitudes of each oscillator across each increment. A complete parameter vector for this signal is given by $\boldsymbol{\theta} \in \mathbb{R}^{(K+3)M}$:

$$\boldsymbol{\theta} = \left[\boldsymbol{A}\left[0, :\right]^{\mathrm{T}} \cdots \quad \boldsymbol{A}\left[K-1, :\right]^{\mathrm{T}} \quad \boldsymbol{\phi}^{\mathrm{T}} \quad \left[\boldsymbol{f}^{(1)}\right]^{\mathrm{T}} \quad \left[\boldsymbol{\eta}^{(1)}\right]^{\mathrm{T}}\right]^{\mathrm{T}}.$$

(3.23)

The amplitudes are a function of the experiment parameter with the following form:

$$\boldsymbol{A}\left[k, m\right] = \boldsymbol{a}_0\left[m\right] \mathcal{A}\left(\boldsymbol{\psi}\left[m\right] \mid \boldsymbol{p}\left[k\right]\right),$$

(3.24)

where $\boldsymbol{a}_0 \in \mathbb{R}^M$ is a vector of the "maximal" amplitude for each oscillator, and $\boldsymbol{\psi} \in \mathbb{R}^M$ is a vector of the parameter of interest (diffusion coefficient, relaxation time etc.) for each oscillator . $\boldsymbol{a}_0\left[m\right]$ can be thought of as the largest possible amplitude that could be obtained for oscillator m with the given experiment:

✚ In an inversion recovery experiment, the largest amplitude is achieved as $\tau \to \infty$, as the spin system will have returned back to equilibrium prior to the 90° pulse.

✚ With Carl-Purcell-Meiboom-Gill (CPMG) experiments, the greatest amplitude will occur when $n_{\mathrm{cycles}} = 0$, as no time is designated for transverse relaxation to take place.

Experiment	p	ψ	$\mathcal{A}(\psi\|p)$	$\frac{\partial \mathcal{A}(\psi\|p)}{\partial \psi}$	$\frac{\partial^2 \mathcal{A}(\psi\|p)}{\partial \psi^2}$
Inversion Recovery	τ	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}	T_2	?	?	?
Diffusion	g	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 amplitude-attenuating NMR experiments considered, along with its first and second derivatives, which are required to extract estimates of ψ using NLP.

✦ For diffusion experiments, the largest amplitude is achieved when $g = 0$, since no diffusion-induced dephasing of spins in different positions along the z-axis will have occurred.

The function $\mathcal{A}$ describes how the amplitudes of resonances are attenuated by the experimental variable, and has a form which is intimately linked to the type of experiment. See Table 3.1 for specific examples.

Estimating amplitude-attenuated datasets

The close relationship between signals across increments means that completely estimating each signal in turn is not necessary. Instead, after the first increment is estimated from scratch, yielding $\theta_0 \in \mathbb{R}^{4M}$, the phases, frequencies and damping factors are fixed, and subsequent increments are determined by taking the parameter estimate of the previous iteration θ_{k-1} and subjecting it to NLP where only the amplitudes are allowed to be varied. Thus, determining parameter estimates for each iteration where $k \neq 0$ is reduced to the problem[††]

$$\boldsymbol{A}[k, :] = \arg\min_{\boldsymbol{a} \in \mathbb{R}^M} \mathcal{F}\left(\boldsymbol{a} \mid \boldsymbol{\phi}, \boldsymbol{f}^{(1)}, \boldsymbol{\eta}^{(1)}, \boldsymbol{Y}[k, :]\right). \tag{3.25}$$

A NLP routine can solve this very efficiently, typically in few iterations, on account of the linear dependence of the model with respect to the oscillator amplitudes. The linear dependence also means that second derivatives of the model are all zero (see (2.41a)), such that only first derivatives need to be computed to derive an exact Hessian matrix of the fidelity.

Remark 4. *Due to the linear dependence on amplitudes, an alternative means of deriving ampli-*

[††] $\mathcal{F}$ in (3.25) reads as "the fidelity with respect to the amplitudes $\boldsymbol{a}$, given phases $\boldsymbol{\phi}$, frequencies $\boldsymbol{f}^{(1)}$, damping factors $\boldsymbol{\eta}^{(1)}$, and FID $\boldsymbol{Y}[k, :]$". The expression has exactly the same mathematical form as (2.38), though it emphasises that the phases, frequencies and damping factors are no longer variables to be optimised, but fixed parameters.

Algorithm 5 Routine for estimating a sequence of 1D FIDs which exhibit variation in amplitudes across increments. NLPAMP denotes a routine which is akin to NLP (Algorithm 3), except only amplitudes are allowed to be altered, whilst phases, frequencies and damping factors are fixed.

1: **procedure** ESTIMATEAMPATTENUATED($Y \in \mathbb{C}^{K \times N^{(1)}}, r_{\text{interest}} \in \mathbb{R}^2, r_{\text{noise}} \in \mathbb{R}^2, M \in \mathbb{N}_0$)
2:      $\theta_0, \epsilon_0 \leftarrow$ ESTIMATE1D $(Y[0,:], r_{\text{interest}}, r_{\text{noise}}, M)$;
3:      $M \leftarrow {}^{\text{len}(\theta_0)}/_4$;
4:      $\theta, \epsilon \leftarrow \mathbf{0} \in \mathbb{R}^{(K+3)M}, \mathbf{0} \in \mathbb{R}^{(K+3)M}$;   ▷ Initialise complete parameter vector.
5:      $\theta[:M], \epsilon[:M] \leftarrow \theta_0[:M], \epsilon_0[:M]$;   ▷ Amplitudes for first increment.
6:      $\theta[KM:], \epsilon[KM:] \leftarrow \theta_0[M:], \epsilon_0[M:]$;   ▷ Phases, frequencies and damping factors, which are held constant across increments.
7:      **for** $k = 1, \cdots, K - 1$ **do**
8:          $\widetilde{y} \leftarrow$ FILTER1D $(Y[k,:], r_{\text{interest}}, r_{\text{noise}})$;   ▷ Algorithm A2.
9:          $\theta_k, \epsilon_k \leftarrow$ NLPAMP $(\widetilde{y}, \theta_{k-1})$;
10:         $\theta[kM:(k+1)M], \epsilon[kM:(k+1)M] \leftarrow \theta_k[:M], \epsilon_k[:M]$;   ▷ Extract amplitudes.
11:      **end for**
12:      **return** θ, ϵ;
13: **end procedure**

---

tudes for each increment is to determine the following $\forall k \in \{1, \cdots, K - 1\}$:

$$A[k,:] = \Lambda^+ Y[k,:], \tag{3.26a}$$

$$\Lambda = \begin{bmatrix} \exp(i\phi[0]) & \cdots & \exp(i\phi[M-1]) \\ \exp(i\phi[0]) \, z[0] & \cdots & \exp(i\phi[M-1]) \, z[M-1] \\ \vdots & \ddots & \vdots \\ \exp(i\phi[0]) \, z[0]^{N^{(1)}-1} & \cdots & \exp(i\phi[M-1]) \, z[M-1]^{N^{(1)}-1} \end{bmatrix}, \tag{3.26b}$$

$$z[m] = \exp\left(\left(2\pi i \left(f^{(1)}[m] - f^{(1)}_{\text{off}} \right) - \gamma^{(1)}[m] \right) \Delta_t^{(1)} \right). \tag{3.26c}$$

Determining the parameter of interest

Having generated a complete parameter estimate for the FID, focus subsequently moves to determining the parameters of interest ψ. For each oscillator, the maximal amplitude and parameter of interest are determined by solving the following problem:

$$\begin{bmatrix} a_0^{(*)}[m] \\ \psi^{(*)}[m] \end{bmatrix} = \underset{[a_0 \ \psi]^{\mathrm{T}} \in \mathbb{R}^2}{\arg\min} \left\| A[:,m] - a_0 \mathcal{A}(\psi \mid p) \right\|^2, \tag{3.27}$$

which can also be solved using an NLP routine. The gradient vector and Hessian matrix of the fidelity take very similar functional forms to those for FID estimation (see (2.39a) and (2.39b)), as both problems involve solving residual sum-of-squares problems. They are as follows $\forall i, j \in$

$\{0,1\}$:

$$g[i] = -2 \left\langle A[:,m] - a_0 \mathcal{A}(\psi|p), \frac{\partial a_0 \mathcal{A}(\psi|p)}{\partial \vartheta[i]} \right\rangle \tag{3.28a}$$

$$H[i,j] = 2 \left(\left\langle \frac{\partial a_0 \mathcal{A}(\psi|p)}{\partial \vartheta[i]}, \frac{\partial a_0 \mathcal{A}(\psi|p)}{\partial \vartheta[j]} \right\rangle - \left\langle A[:,m] - a_0 \mathcal{A}(\psi|p), \frac{\partial^2 a_0 \mathcal{A}(\psi|p)}{\partial \vartheta[i] \partial \vartheta[j]} \right\rangle \right), \tag{3.28b}$$

$$\mathbb{R}^2 \ni \vartheta = \begin{bmatrix} a_0 & \psi \end{bmatrix}^{\mathrm{T}}, \tag{3.28c}$$

with explicit expressions for the requisite first and second derivatives being

$$\frac{\partial a_0 \mathcal{A}(\psi|p)}{\partial a_0} = \mathcal{A}(\psi|p), \tag{3.29a}$$

$$\frac{\partial a_0 \mathcal{A}(\psi|p)}{\partial \psi} = a_0 \frac{\partial \mathcal{A}(\psi|p)}{\partial \psi}, \tag{3.29b}$$

$$\frac{\partial^2 a_0 \mathcal{A}(\psi|p)}{\partial a_0^2} = 0, \tag{3.29c}$$

$$\frac{\partial^2 a_0 \mathcal{A}(\psi|p)}{\partial \psi^2} = a_0 \frac{\partial^2 \mathcal{A}(\psi|p)}{\partial \psi^2}, \tag{3.29d}$$

$$\frac{\partial^2 a_0 \mathcal{A}(\psi|p)}{\partial a_0 \partial \psi} = \frac{\partial^2 a_0 \mathcal{A}(\psi|p)}{\partial \psi \partial a_0} = \frac{\partial \mathcal{A}(\psi|p)}{\partial \psi}. \tag{3.29e}$$

The functional forms of the first and second derivatives for the different experiments of interest of $\mathcal{A}$ are given in Table 3.1.

Displaying results

Visualising the results from the routine described above can be done in a similar fashion to diffusion-ordered spectroscopy (DOSY) analysis. For each oscillator in the estimation result, a 2D array is generated, corresponding to the outer product of the FT of the oscillator (s_m) and a distribution describing the predicted value of ψ (d_m).

$$\mathbb{R}^{R \times N^{(1)}} \ni S = \sum_{m=0}^{M-1} \frac{d_m}{\max(d_m)} \otimes s_m, \tag{3.30a}$$

$$s_m = \Re \left(\mathrm{FT}(x_m) \right), \tag{3.30b}$$

$$x_m \left[n^{(1)} \right] = a_0[m] \exp \left(\mathrm{i} \phi[m] \right) \exp \left(\left(2\pi \mathrm{i} \left(f^{(1)}[m] - f_{\mathrm{off}}^{(1)} \right) - \eta_m^{(1)} \right) n^{(1)} \Delta_t^{(1)} \right), \tag{3.30c}$$

where $R \in \mathbb{N}$ is the number of samples to generate the distribution from. The exact form that the distribution should take is not set in stone, though it should indicate two key pieces of information: (i) its maximum should coincide with the predicted value of ψ and (ii) its width should indicate the certainty in the prediction. In this work, the distribution used is a Gaussian distribu-

tion with mean ψ and standard deviation $c\epsilon_\psi$, where ϵ_ψ is the estimation error associated with the parameter of interest,[‡‡] and $c \in \mathbb{R}_{>0}$ is an arbitrary linewidth factor which can be chosen to ensure clear visibility of the peaks.[§§]

$$\boldsymbol{d}_m\left[r\right] = \exp\left(-\frac{\left(r - \boldsymbol{\psi}\left[m\right]\right)^2}{2\left(c\boldsymbol{\epsilon}_\psi\left[m\right]\right)^2}\right) \quad \forall r \in \{0, \cdots, R - 1\}. \tag{3.31}$$

3.4.4   Results

"Five multiplets"

Figure 3.10 shows the result of the described method in determining the T_1 values of resonances in a simulated inversion recovery datasets featuring 5 overlapping ddd multiplet structures. The spin systems used to generate the datasets were constructed in a similar way to that described for the "Four Multiplets" example in Section 3.3.3. In this case however, 5 estimated spins were present instead of 4. Some constraints were also placed on the shifts and couplings to ensure that no two oscillators would have frequencies with a difference less than $f_{sw}^{(1)}/N^{(1)}$. For each spin, a T_1 value was sampled from $\mathcal{U}\left(1\,\mathrm{s}, 5\,\mathrm{s}\right)$, and a T_2 value was sampled from $\mathcal{U}\left(0.2\,\mathrm{s}, 0.6\,\mathrm{s}\right)$. The inversion recovery experiment was simulated using Spinach, with the relaxation phenomena described by the "Extended T_1/T_2 approximation"[¶¶]. Each dataset was provided AWGN such that the target SNR of the datasets as a whole was 40 dB. The estimation routine outlined by Algorithm 5 was applied to the generate a parameter estimate of the 1.6 ppm to 1.2 ppm region. Despite heavy overlap between peaks, the routine was high successful at assigning each resonance with a T_1 value that agreed with the true value (see panel b.). As is to be expected, in scenarios where little multiplet overlap existed, T_1 predictions tended to be more accurate, with smaller assocaited errors (see for example the purple and orange multiplets in Run 1. Nevertheless, adequate estimates could still be obtained in cases of severe overlap, especially when the predicted T_1s for all oscillators associated with a given multiplet are averaged. (see the red, blue and yellow multiplets in Run 1). Particular oscillators for which the estimate of T_1 is particularly far from the true value are often reassuringly associated with large errors, with examples denoted with an asterisk.

[‡‡]The errors can be extracted from the Hessian matrix once the NLP routine has reached convergence. See section 2.3.5 for more information.

[§§]In cases where the estimation error is small, the distribution can be so sharp relative to the resolution used for plotting, that $\max(\boldsymbol{d}_m)$ can be 0. In these situations, errors arise due to attempted division by 0 (see (3.30a)). Broadening the distribution (i.e. setting $c > 1$) can resolve issues with plotting discrete samples.

[¶¶]As described in the Spinach documentation[76], longitudinal spin states are assigned the rate $1/T_1$ and transverse states are assigned the rate $1/T_2$. Multi-spin states are assigned rates which are the summation of each spin's relevant rate.
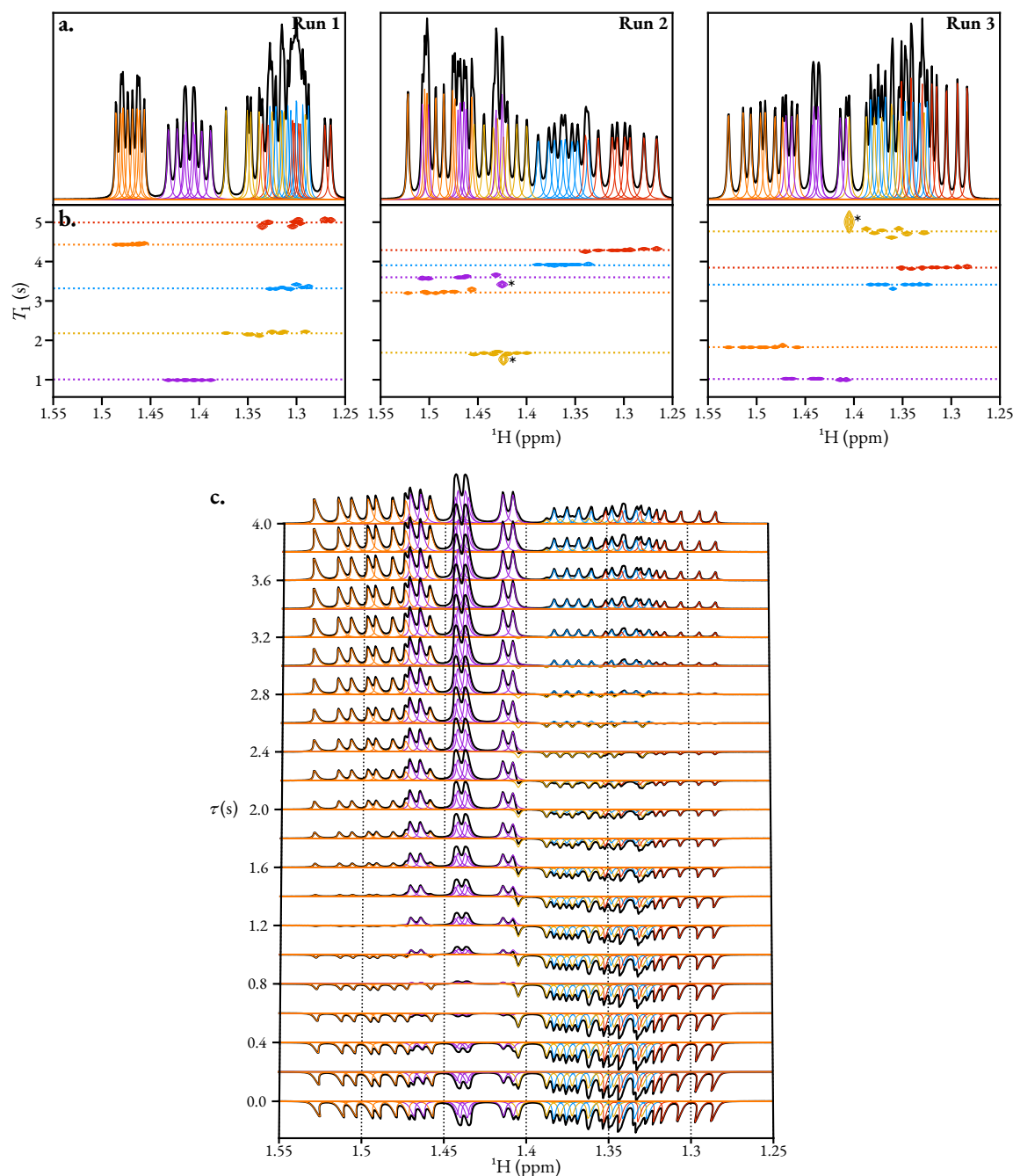
FIGURE 3.10: Three examples of results generated on simulated inversion recovery datasets comprising five ddd multiplet structures. **a.** Plot of the result generated for the first increment ($\tau = 0\,\text{s}$), with each plot multiplied by -1. Black: spectrum of the data. Coloured lines: spectra of individual oscillators generated by the estimation routine. Oscillators with the same colour are components of the same multiplet. **b.** DOSY-style contour plot of the result, with peak positions along the x-axis and predicted T_1 values along the y-axis. The broadness of the peaks along the y-axis indicate the relative errors in the T_1 estimates. The distributions were generated with the linewidth factor $c = 16$. Dashed horizontal lines denote the true T_1 values for each spin. **c.** Estimation result for each increment for Run 3, illustrating the evolution of the amplitudes of each oscillator with τ.

3.5   Miscellanea

BBQCHILI, Metabolomics (using PCA,PLS,OPLS)

The NMR-EsPy Package

<div style="text-align: right; font-size: 2em;">4</div>

The estimation routine and applications emerging from it that have been presented in the previous two chapters are accessible via the NMR-EsPy package. NMR-EsPy aims to provide a feature-rich yet simple interface in order perform estimation on datasets of interest. In this chapter, a description of the package is given, including the design choices, a basic description of the package structure, and a description of the accompanying graphical user interface (GUI).

A rigorous description of the usage of NMR-EsPy, including details on installation, walk-throughs for different data types, and a reference for the application programming interface (API) are given in the documentation for NMR-EsPy. A hard-copy of the version 2.0 documentation can be found in Section **TODO** of the Appendix. The most up-to-date HTML version of the documentation can be found at `https://foroozandehgroup.github.io/NMR-EsPy/`. The source code for NMR-EsPy is hosted on the Foroozandeh group's GitHub page at `https://github.com/foroozandehgroup/NMR-EsPy`.

4.1 The Structure of NMR-EsPy

4.1.1 Why Python?

There a number of reasons why Python was the chosen programming language for NMR-EsPy:

- It has a large user-base, particularly within the scientific community.

- The SciPy ecosystem[77], including the packages NumPy[78] and Matplotlib[79] is a powerful tool which enables high-performance scientific computation in Python

- Being a scripting language makes Python ideal for exploring datasets in a step-by-step fashion. This is useful in the context of NMR estimation, as the user may want to (a) inspect and pre-process then data, ten (b) determine the regions they wish to estimate, then (c) setup the estimation routine, and finally (d) output the estimation result. This can be achieved rather

easily by hacking and re-running Python scripts or by using "notebook" environments, such as Jupyter.

❖ It is free and open-source, as opposed to well-known scientific computing platforms such as Matlab® and Mathematica.

❖ Python supports sophisticated object-oriented programming features, such as multiple levels of inheritance.

Probably the biggest drawback of Python is its slow performance on account of it being an interpreted[*] , dynamically typed[†], language with relies on garbage collection[‡] for memory management. While NumPy provides interfaces to run fast computations with pre-compiled C-code, a significant performnace benefit would likely be realised if a low-level compiled language like C, C++, or Rust were used. However the development time in writing programs with these lower-level languages is typically a lot greater than with a language with a higher level of abstraction like Python.

4.1.2   ???

The fundamental user-facing object that NMR-EsPy provides is the `Estimator` class and its numerous inherited classes, which facilitate the estimation of different types of NMR data. A complete list of estimator objects at the time of writing is given in Table 4.1

Table 4.1: A complete list of estimator objects provided by NMR-EsPy at the time of writing. **Maybe indicate that the inv rec and diffusion estimators share similar functionality, except for the amplitude fitting. Diffusion experiments only differ in definition of fitting constant.**

Type	Description
Estimator1D	For consideration of 1D datasets.
Estimator2D	For consideration of 2D datasets comprising a pair of States (amplitude-modulated) signals.
Estimator2DJ	For consideration of 2DJ datasets. This object provides the functionality to generate pure shift spectra and assign multiplet structures using CUPID.

---

[*]With interpreted languages, the source code is processed line-by-line at the time of running by an interpreter. This differs from compiled languages, where prior to being run, the source code is converted to machine-readable byte-code.

[†]Dynamically typed languages, as opposed to statically typed languages like C, C++, Java, Rust etc. allow for a variable which is initially assigned a given type to be re-assigned to a completely different type.

[‡]Garbage collection involves a program routinely checking for any memory that has become dereferenced, and clearing this memory up.

`EstimatorInvRec`	For consideration of inversion recovery (T_1) experiments.
`EstimatorDiffusionMonopolar`	For consideration of monopolar gradient diffusion experiments.
`EstimatorDiffusionBipolar`	For consideration of bipolar gradient diffusion experiments.
`EstimatorDiffusionOneshot`	For consideration of one-shot DOSY experiments.

CONCLUSIONS AND FUTURE WORK

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] Yingbo Hua. "High resolution imaging of continuously moving object using stepped frequency radar". In: *Signal Processing* 35.1 (1994), pp. 33–40.

[43] S.N. Fricke et al. "Data processing in NMR relaxometry using the matrix pencil". In: *Journal of Magnetic Resonance* 313 (2020), p. 106704.

[44] Dennis Wörtge et al. "Quantitative stray-field T1 relaxometry with the matrix pencil method". In: *Journal of Magnetic Resonance* 351 (2023), p. 107435.

[45] 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.

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

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

[48] 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.

[49] 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.

[50] 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.

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

[52] 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.

[53] 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.

[54] 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.

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

[56] 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.

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

[58]   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.

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

[60]   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.

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

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

[63]   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.

[64]   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.

[65]   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.

[66]   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.

[67]   M. J. Frisch et al. *Gaussian03 Revision E.01*. Gaussian Inc. Wallingford CT. 2003.

[68]   M. Foroozandeh et al. "Measuring couplings in crowded NMR spectra: Pure shift NMR with multiplet analysis". eng. In: *Chemical communications (Cambridge, England)* 51.84 (2015), pp. 15410–15413.

[69]   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.

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

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

[72]   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.

[73]   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.

[74]   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.

[75]   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.

[76]   *Spinach Documentation*. `https://spindynamics.org/wiki/index.php?title=Main_Page`. Accessed 11-05-2023.

[77]   Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nat. Methods* 17 (2020), pp. 261–272.

[78]   Charles R. Harris et al. "Array programming with NumPy." In: *Nature* 585.7825 (2020), p. 357.

[79]   J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95.

[80]   Stefan Berger and Siegmar Braun. *200 and more NMR experiments : a practical course*. eng. [3rd rev. and expanded ed.] Weinheim: Wiley-VCH, 2004.

[81]   Craig P Butts, Catharine R Jones, and Jeremy N Harvey. "High precision NOEs as a probe for low level conformers-a second conformation of strychnine". eng. In: 47.4 (2011), pp. 1193–1195.

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

Appendix

A.1 Implementation Details

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

A.1.1 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
from typing import Tuple
import numpy as np

def obj_grad_hess(
    theta: np.ndarray, *args,
) -> Tuple[float, np.ndarray, np.ndarray]:
    """Compute the objective, gradient and hessian for 2D data.

    Parameters
    ----------
    theta
        1D array of parameters, of length 2(1+D)M

    args
        Contains elements in the following order:

        * data: Array of the original FID data.
        * 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.

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

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

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

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

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

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

$$\boldsymbol{X}_m = \exp\left(\boldsymbol{Z}\right) * \alpha \tag{A.2}$$

```
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.1.2 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]]
```

A.2   Additional Algorithms

Algorithm A1 Steihaug-Toint method for determining an update for nonlinear programming. This is equivalent to Algorithm 7.2 in [27].

1: **procedure** STEIHAUGTOINT($Y \in \mathbb{C}^{N^{(1)} \times \cdots \times N^{(D)}}, \theta^{(k)} \in \mathbb{R}^{2(1+D)M}, \Delta^{(k)} \in \mathbb{R}_{>0}$)

2:      $g \leftarrow \nabla \mathcal{F}_\phi \left(\theta^{(k)} \mid Y \right);$ ▷ Grad vector: (2.39a)

3:      $H \leftarrow \nabla^2 \mathcal{F}_\phi \left(\theta^{(k)} \mid Y \right);$ ▷ Hessian matrix, either exact: (2.39b) or approximate: (2.42)

4:      $\epsilon^{(k)} \leftarrow \min \left(1/2, \sqrt{\|g\|} \right) \|g\|;$

5:      $z^{(0)} \leftarrow 0 \in \mathbb{R}^{6M};$

6:      $r^{(0)} \leftarrow g;$

7:      $d^{(0)} \leftarrow -r^{(0)};$

8:      **if** $\|r^{(0)}\| < \epsilon^{(k)}$ **then**

9:          **return** $z^{(0)};$

10:      **end if**

11:      **for** $j = \{0, 1, \cdots\}$ **do**

12:          **if** $d^{(j)\,\mathrm{T}} H d^{(j)} \leq 0$ **then**

13:              Find τ such that $p^{(k)} = z^{(j)} + \tau d^{(j)}$ minimises $\mathcal{F}_{\phi Q} \left(\theta^{(k)} + p^{(k)} \right)$, subject to $\|p^{(k)}\| = \Delta^{(k)};$

14:              **return** $p^{(k)};$

15:          **end if**

16:          $\alpha^{(j)} \leftarrow \dfrac{r^{(j)\,\mathrm{T}} r^{(j)}}{d^{(j)\,\mathrm{T}} H d^{(j)}};$

17:          $z^{(j+1)} \leftarrow z^{(j)} + \alpha^{(j)} d^{(j)};$

18:          **if** $\|z^{(j+1)}\| < \epsilon^{(k)}$ **then**

19:              Find $\tau \in \mathbb{R}_{>0}$ such that $p^{(k)} = z^{(j)} + \tau d^{(j)}$ satisfies $\|p^{(k)}\| = \Delta^{(k)};$

20:              **return** $p^{(k)};$

21:          **end if**

22:          $r^{(j+1)} \leftarrow r^{(j)} + \alpha^{(j)} H d^{(j)};$

23:          **if** $\|r^{(j+1)}\| < \epsilon^{(k)}$ **then**

24:              **return** $z^{(j+1)};$

25:          **end if**

26:          $\beta^{(j+1)} \leftarrow \dfrac{r^{(j+1)\,\mathrm{T}} r^{(j+1)}}{r^{(j)\,\mathrm{T}} r^{(j)}};$

27:          $d^{(j+1)} \leftarrow -r^{(j+1)} + \beta^{(j+1)} d^{(j)};$

28:      **end for**

29: **end procedure**

Algorithm A2 Filtering procedure for 1D data. $\boldsymbol{r}_{\text{interest}}$ is a vector of length 2 containing the indices of the left and right bounds of the region of interest. These would typically be provided in units of Hz or ppm by a user. Conversion to array indices can be carried out using (2.67). $\boldsymbol{r}_{\text{noise}}$ contains the left and right bounds of the region used to estimate the noise variance. RANDOM-SAMPLE indicates taking a random sample from the given distribution.

1: **procedure** FILTER1D($\boldsymbol{y} \in \mathbb{C}^{N^{(1)}}, \boldsymbol{r}_{\text{interest}} \in \mathbb{N}_0^2, \boldsymbol{r}_{\text{noise}} \in \mathbb{N}_0^2$)
2:     $\boldsymbol{y}_{\text{ve}} \leftarrow \text{VIRTUALECHO1D}(\boldsymbol{y})$;
3:     $\boldsymbol{s}_{\text{ve}} \leftarrow \text{FT}(\boldsymbol{y}_{\text{ve}})$;
4:     $l_{\text{idx}}^{(1)}, r_{\text{idx}}^{(1)} \leftarrow \boldsymbol{r}_{\text{interest}}[0], \boldsymbol{r}_{\text{interest}}[1]$;
5:     $l_{\text{idx,noise}}^{(1)}, r_{\text{idx,noise}}^{(1)} \leftarrow \boldsymbol{r}_{\text{noise}}[0], \boldsymbol{r}_{\text{noise}}[1]$;
6:     $c_{\text{idx}}^{(1)} \leftarrow \left(l_{\text{idx}}^{(1)} + r_{\text{idx}}^{(1)} \right)/2$;
7:     $b_{\text{idx}}^{(1)} \leftarrow r_{\text{idx}}^{(1)} - l_{\text{idx}}^{(1)}$;
8:     $\boldsymbol{g} \leftarrow \text{SUPERGAUSSIAN1D}\left(N^{(1)}, c_{\text{idx}}^{(1)}, b_{\text{idx}}^{(1)} \right)$;
9:     $\boldsymbol{s}_{\text{noise}} \leftarrow \boldsymbol{s}_{\text{ve}}\left[l_{\text{idx,noise}}^{(1)} : r_{\text{idx,noise}}^{(1)} + 1 \right]$;
10:     $\sigma^2 \leftarrow \text{Var}(\boldsymbol{s}_{\text{noise}})$;
11:     $\boldsymbol{w}_{\sigma^2} \leftarrow \boldsymbol{0} \in \mathbb{R}^{2N^{(1)}}$;
12:     **for** $n^{(1)} = 0, \cdots, N^{(1)} - 1$ **do**
13:         $\boldsymbol{w}_{\sigma^2}\left[n^{(1)} \right] \leftarrow \text{RANDOMSAMPLE}\left(\mathcal{N}\left(0, \sigma^2 \right) \right)$;
14:     **end for**
15:     $\widetilde{\boldsymbol{s}}_{\text{ve}} \leftarrow \boldsymbol{s}_{\text{ve}} \odot \boldsymbol{g} + \boldsymbol{w}_{\sigma^2} \odot (\boldsymbol{1} - \boldsymbol{g})$;
16:     $\widetilde{\boldsymbol{y}}_{\text{ve}} \leftarrow \text{IFT}(\widetilde{\boldsymbol{s}}_{\text{ve}})$;
17:     $\widetilde{\boldsymbol{y}} \leftarrow \widetilde{\boldsymbol{y}}_{\text{ve}}\left[: N^{(1)} \right]$;
18:     **return** $\widetilde{\boldsymbol{y}}$;
19: **end procedure**

20: **procedure** VIRTUALECHO1D($\boldsymbol{y} \in \mathbb{C}^{N^{(1)}}$)
21:     $\boldsymbol{t}_1 \leftarrow \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \in \mathbb{C}^{N^{(1)}} \end{bmatrix}$;
22:     $\boldsymbol{t}_2 \leftarrow \begin{bmatrix} \boldsymbol{0} \in \mathbb{C}^{N^{(1)}} \\ \boldsymbol{y}^{* \leftrightarrow (1)} \end{bmatrix}^{\circlearrowright (1)}$;
23:     $\boldsymbol{y}_{\text{ve}} \leftarrow \boldsymbol{t}_1 + \boldsymbol{t}_2$;
24:     $\boldsymbol{y}_{\text{ve}}[0] \leftarrow \boldsymbol{y}_{\text{ve}}[0]/2$;
25:     **return** $\boldsymbol{y}_{\text{ve}}$;
26: **end procedure**

27: **procedure** SUPERGAUSSIAN1D($N \in \mathbb{N}, c_{\text{idx}} \in \mathbb{R}_{>0}, b_{\text{idx}} \in \mathbb{N}$)
28:     $\boldsymbol{g} \leftarrow \boldsymbol{0} \in \mathbb{R}^N$;
29:     **for** $n = 0, \cdots, N - 1$ **do**
30:         $\boldsymbol{g}[n] \leftarrow \exp\left(-2^{41} \left(\frac{n - c_{\text{idx}}}{b_{\text{idx}}} \right)^{40} \right)$; $\triangleright$ p in (2.63b) has been set to 40.
31:     **end for**
32:     **return** $\boldsymbol{g}$
33: **end procedure**

Algorithm A3 Filtering procedure for 2D data.

1: **procedure** $\textsc{Filter2D}(\boldsymbol{Y}_{\cos} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{Y}_{\sin} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{R}_{\text{interest}} \in \mathbb{N}_0^{2 \times 2}, \boldsymbol{R}_{\text{noise}} \in \mathbb{N}_0^{2 \times 2})$
2:    $\boldsymbol{Y}_{\text{ve}} \leftarrow \textsc{VirtualEcho2D}\left(\boldsymbol{Y}_{\cos}, \boldsymbol{Y}_{\sin}\right);$
3:    $\boldsymbol{S}_{\text{ve}} \leftarrow \text{FT}\left(\boldsymbol{Y}_{\text{ve}}\right);$
4:    $l_{\text{idx}}^{(1)}, r_{\text{idx}}^{(1)}, l_{\text{idx}}^{(2)}, r_{\text{idx}}^{(2)} \leftarrow \boldsymbol{R}_{\text{interest}}[0,0], \boldsymbol{R}_{\text{interest}}[0,1], \boldsymbol{R}_{\text{interest}}[1,0], \boldsymbol{R}_{\text{interest}}[1,1];$
5:    **for** $d = 1,2$ **do**
6:      $c_{\text{idx}}^{(d)} \leftarrow \left(l_{\text{idx}}^{(d)} + r_{\text{idx}}^{(d)}\right)/2;$
7:      $b_{\text{idx}}^{(d)} \leftarrow r_{\text{idx}}^{(d)} - l_{\text{idx}}^{(d)};$
8:      $\boldsymbol{g}^{(d)} \leftarrow \textsc{SuperGaussian1D}\left(2N^{(d)}, c_{\text{idx}}^{(d)}, b_{\text{idx}}^{(d)}\right);$
9:      $\boldsymbol{G} \leftarrow \boldsymbol{g}^{(1)} \otimes \boldsymbol{g}^{(2)};$
10:   **end for**
11:   $l_{\text{idx,noise}}^{(1)}, r_{\text{idx,noise}}^{(1)}, l_{\text{idx,noise}}^{(2)}, r_{\text{idx,noise}}^{(2)} \leftarrow \boldsymbol{R}_{\text{noise}}[0,0], \boldsymbol{R}_{\text{noise}}[0,1], \boldsymbol{R}_{\text{noise}}[1,0], \boldsymbol{R}_{\text{noise}}[1,1];$
12:   $\boldsymbol{S}_{\text{noise}} \leftarrow \boldsymbol{S}_{\text{ve}}\left[l_{\text{idx,noise}}^{(1)} : r_{\text{idx,noise}}^{(1)} + 1, l_{\text{idx,noise}}^{(2)} : r_{\text{idx,noise}}^{(2)} + 1\right]$
13:   $\sigma^2 \leftarrow \text{Var}\left(\boldsymbol{S}_{\text{noise}}\right);$
14:   $\boldsymbol{W}_{\sigma^2} \leftarrow \boldsymbol{0} \in \mathbb{R}^{2N^{(1)} \times 2N^{(2)}};$
15:   **for** $n^{(1)} = 0, \cdots, 2N^{(1)} - 1$ **do**
16:     **for** $n^{(2)} = 0, \cdots, 2N^{(2)} - 1$ **do**
17:       $\boldsymbol{W}_{\sigma^2}\left[n^{(1)}, n^{(2)}\right] \leftarrow \textsc{RandomSample}\left(\mathcal{N}\left(0, \sigma^2\right)\right);$
18:     **end for**
19:   **end for**
20:   $\widetilde{\boldsymbol{S}}_{\text{ve}} \leftarrow \boldsymbol{S}_{\text{ve}} \odot \boldsymbol{G} + \boldsymbol{W}_{\sigma^2} \odot \left(\boldsymbol{1} - \boldsymbol{G}\right);$
21:   $\widetilde{\boldsymbol{Y}}_{\text{ve}} \leftarrow \text{IFT}\left(\widetilde{\boldsymbol{S}}_{\text{ve}}\right);$
22:   $\widetilde{\boldsymbol{Y}} \leftarrow \widetilde{\boldsymbol{Y}}_{\text{ve}}\left[: N^{(1)}, : N^{(2)}\right];$
23:   **return** $\widetilde{\boldsymbol{Y}};$
24: **end procedure**

25: **procedure** $\textsc{VirtualEcho2D}(\boldsymbol{Y}_{\cos} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}, \boldsymbol{Y}_{\sin} \in \mathbb{C}^{N^{(1)} \times N^{(2)}})$
26:   $\boldsymbol{\Psi}_{++} \leftarrow \Re\left(\boldsymbol{Y}_{\cos}\right) - \Im\left(\boldsymbol{Y}_{\sin}\right) + i\left(\Im\left(\boldsymbol{Y}_{\cos}\right) + \Re\left(\boldsymbol{Y}_{\sin}\right)\right);$
27:   $\boldsymbol{\Psi}_{+-} \leftarrow \Re\left(\boldsymbol{Y}_{\cos}\right) + \Im\left(\boldsymbol{Y}_{\sin}\right) + i\left(\Re\left(\boldsymbol{Y}_{\sin}\right) - \Im\left(\boldsymbol{Y}_{\cos}\right)\right);$
28:   $\boldsymbol{\Psi}_{-+} \leftarrow \Re\left(\boldsymbol{Y}_{\cos}\right) + \Im\left(\boldsymbol{Y}_{\sin}\right) + i\left(\Im\left(\boldsymbol{Y}_{\cos}\right) - \Re\left(\boldsymbol{Y}_{\sin}\right)\right);$
29:   $\boldsymbol{\Psi}_{--} \leftarrow \Re\left(\boldsymbol{Y}_{\cos}\right) - \Im\left(\boldsymbol{Y}_{\sin}\right) - i\left(\Im\left(\boldsymbol{Y}_{\cos}\right) + \Re\left(\boldsymbol{Y}_{\sin}\right)\right);$
30:   $\boldsymbol{Z} \leftarrow \boldsymbol{0} \in \mathbb{C}^{N^{(1)} \times N^{(2)}}$
31:   $\boldsymbol{T}_1 \leftarrow \begin{bmatrix} \boldsymbol{\Psi}_{++} & \boldsymbol{Z} \\ \boldsymbol{Z} & \boldsymbol{Z} \end{bmatrix};$
32:   $\boldsymbol{T}_2 \leftarrow \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{\Psi}_{+-}^{\leftrightarrow(2)} \\ \boldsymbol{Z} & \boldsymbol{Z} \end{bmatrix}^{\circlearrowright(2)};$
33:   $\boldsymbol{T}_3 \leftarrow \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{Z} \\ \boldsymbol{\Psi}_{-+}^{\leftrightarrow(1)} & \boldsymbol{Z} \end{bmatrix}^{\circlearrowright(1)};$
34:   $\boldsymbol{T}_4 \leftarrow \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{Z} \\ \boldsymbol{Z} & \boldsymbol{\Psi}_{--}^{\leftrightarrow(1,2)} \end{bmatrix}^{\circlearrowright(1,2)};$
35:   $\boldsymbol{Y}_{\text{ve}} \leftarrow \boldsymbol{T}_1 + \boldsymbol{T}_2 + \boldsymbol{T}_3 + \boldsymbol{T}_4;$
36:   **for** $n^{(1)} = 0, \cdots, 2N^{(1)} - 1$ **do**
37:     $\boldsymbol{Y}_{\text{ve}}\left[n^{(1)}, 0\right] \leftarrow \boldsymbol{Y}_{\text{ve}}\left[n^{(1)}, 0\right]/2;$
38:   **end for**
39:   **for** $n^{(2)} = 0, \cdots, 2N^{(2)} - 1$ **do**
40:     $\boldsymbol{Y}_{\text{ve}}\left[0, n^{(2)}\right] \leftarrow \boldsymbol{Y}_{\text{ve}}\left[0, n^{(2)}\right]/2;$
41:   **end for**
42:   **return** $\boldsymbol{Y}_{\text{ve}};$
43: **end procedure**

Algorithm A4 Filtering procedure for 2DJ data.

1: **procedure** FILTER2DJ($Y \in \mathbb{C}^{N^{(1)} \times N^{(2)}}$, $r_{\text{interest}} \in \mathbb{N}_0^2$, $r_{\text{noise}} \in \mathbb{N}_0^2$)
2:     $Y_{\text{ve}} \leftarrow \mathbf{0} \in \mathbb{C}^{N^{(1)} \times 2N^{(2)}}$;
3:     **for** $n^{(1)} = 0, \cdots, N^{(1)} - 1$ **do**
4:         $Y_{\text{ve}}\left[n^{(1)}, :\right] \leftarrow \text{VIRTUALECHO1D}\left(Y\left[n^{(1)}, :\right]\right)$;
5:     **end for**
6:     $S_{\text{ve}} \leftarrow \text{FT}^{(2)}(Y_{\text{ve}})$;
7:     $l_{\text{idx}}^{(2)}, r_{\text{idx}}^{(2)} \leftarrow r_{\text{interest}}[0], r_{\text{interest}}[1]$;
8:     $l_{\text{idx,noise}}^{(2)}, r_{\text{idx,noise}}^{(2)} \leftarrow r_{\text{noise}}[0], r_{\text{noise}}[1]$;
9:     $c_{\text{idx}}^{(2)} \leftarrow \left(l_{\text{idx}}^{(2)} + r_{\text{idx}}^{(2)}\right)/2$;
10:     $b_{\text{idx}}^{(2)} \leftarrow r_{\text{idx}}^{(2)} - l_{\text{idx}}^{(2)}$;
11:     $g^{(1)} \leftarrow \mathbf{1} \in \mathbb{R}^{N^{(1)}}$;
12:     $g^{(2)} \leftarrow \text{SUPERGAUSSIAN1D}\left(N^{(2)}, c_{\text{idx}}^{(2)}, b_{\text{idx}}^{(2)}\right)$;
13:     $G \leftarrow g^{(1)} \otimes g^{(2)}$;
14:     $S_{\text{noise}} \leftarrow S_{\text{ve}}\left[:, l_{\text{idx,noise}}^{(2)} : r_{\text{idx,noise}}^{(2)} + 1\right]$;
15:     $\sigma^2 \leftarrow \text{Var}(S_{\text{noise}})$;
16:     $W_{\sigma^2} \leftarrow \mathbf{0} \in \mathbb{R}^{N^{(1)} \times 2N^{(2)}}$;
17:     **for** $n^{(1)} = 0, \cdots, N^{(1)} - 1$ **do**
18:         **for** $n^{(2)} = 0, \cdots, 2N^{(2)} - 1$ **do**
19:             $W_{\sigma^2}\left[n^{(1)}, n^{(2)}\right] \leftarrow \text{RANDOMSAMPLE}\left(\mathcal{N}\left(0, \sigma^2\right)\right)$;
20:         **end for**
21:     **end for**
22:     $\widetilde{S}_{\text{ve}} \leftarrow S_{\text{ve}} \odot G + W_{\sigma^2} \odot (\mathbf{1} - G)$;
23:     $\widetilde{Y}_{\text{ve}} \leftarrow \text{IFT}^{(2)}\left(\widetilde{S}_{\text{ve}}\right)$;
24:     $\widetilde{Y} \leftarrow \widetilde{Y}_{\text{ve}}\left[:, : N^{(2)}\right]$;
25:     **return** $\widetilde{Y}$;
26: **end procedure**

Algorithm A5 An algorithm for multiplet assignment of a 2DJ estimation result.

1: **procedure** MULTIPLETASSIGN($\theta \in \mathbb{R}^{6M}, \varepsilon \in \mathbb{R}_{>0}$)
2:     $f^{(1)}, f^{(2)} \leftarrow \theta[2M : 3M], \theta[3M : 4M]$;
3:     MAP $\leftarrow$ HASHMAP$\left[\mathbb{R}, \text{VECTOR}\left[\mathbb{N}_0\right]\right]$;
4:     **for** $m = \{0, \cdots, M - 1\}$ **do**
5:         $f_{\text{c}} = f^{(2)}[m] - f^{(1)}[m]$;
6:         ASSIGNED $\leftarrow$ FALSE;
7:         **for** f_{mp}, i in MAP **do**
8:             **if** $\left|f_{\text{c}} - f_{\text{mp}}\right| < \varepsilon$ **then**
9:                 $i \leftarrow \left[i^\text{T} \ m\right]^\text{T}$;
10:                 ASSIGNED $\leftarrow$ TRUE;
11:                 **break**;
12:             **end if**
13:         **end for**
14:         **if** ASSIGNED $=$ FALSE **then**
15:             MAP.insert $\left(f_{\text{c}}, [m]\right)$;
16:         **end if**
17:     **end for**
18:     **return** MAP;
19: **end procedure**

A.3   Information on simulated datasets

A.3.1   Spinach Simulations

Many of the simulated datasets presented in this work were generated using the Spinach Mat-LAB® package[66]. In each case, the dataset was generated via a call to the `new_spinach` method associated with the relevant `Estimator` object in NMR-EsPy. `new_spinach` works by instantiating a Matlab engine from Python, which then runs a Spinach simulation of the relevant experiment with the specifications provided. The FID generated is then stored within in a new `Estimator` object. Table A1 provides a specification of the chemical shifts and scalar couplings that made up the spin systems used to construct simulated data with Spinach. Tables A2 and A3 specify key parameters used in each 2DJ and inversion recovery simulation, respectively.

Sucrose

The chemical shifts and couplings for sucrose were extracted from a Gaussian[67] logfile which ships with Spinach at the path `SPINACHROOT/examples/standard_systems/sucrose.log`. Isotropic chemical shifts were determined for each spin i via

$$\hat{\partial}_i = \frac{\mathrm{Tr}\left(\sigma_i\right)}{3}, \tag{A.3}$$

where $\sigma_i \in \mathbb{R}^{3\times3}$ is the computed chemical shift tensor of spin i.

Strychnine

The strychnine spin system was derived from the Spinach function `<SPINACHROOT>/etc/strychnine.m`, which returns a spin system specification using chemical shifts and scalar couplings from[80: Appendix 5], and atomic coordinates from[81: Supplementary Material]. To determine T_1 and T_2 values for each spin, the relaxation superoperator $\boldsymbol{R}$ was constructed according to Redfield theory, under the assumption that the molecule was undergoing spherical isotropic rotation with a rotational correlation time of 200 ps, in a magnetic field of 700 MHz. Individual T_1s and T_2s were then extracted using

$$T_{(1/2),i} = \frac{1}{R_{(1/2),i}}, \tag{A.4a}$$

$$R_{1,i} = -\Re\left(\boldsymbol{I}_{z,i}^{\dagger}\boldsymbol{R}\boldsymbol{I}_{z,i}\right), \tag{A.4b}$$

$$R_{2,i} = -\Re\left(\boldsymbol{I}_{+,i}^{\dagger}\boldsymbol{R}\boldsymbol{I}_{+,i}\right), \tag{A.4c}$$

where $\boldsymbol{l}_{z,i}$ is the state vector of the Hilbert-space operator $\hat{L}_z$ for spin i, and $\boldsymbol{l}_{+,i}$ is the corresponding vector for the $\hat{L}_+$ operator.

TABLE A1: The isotropic chemical shifts (δ), corresponding rotating frame frequencies (ω_{rot}), scalar couplings (J), and relaxation times (T_1, T_2, if applicable) associated with spin systems used in SPINACH simulations.

Spin	δ (ppm)	ω_{rot} (Hz)	J (Hz)	T_1 (s)	T_2 (s)
Four Multiplets, Run 1					
A	-2.78×10^{-2}	-13.93	**E:** -9.627, **F:** -8.202, **G:** 6.742	–	–
B	-7.11×10^{-3}	-3.56	**E:** -4.491, **F:** 5.333, **G:** 9.303	–	–
C	-1.63×10^{-3}	-0.81	**E:** 3.953, **F:** 5.422, **G:** 5.914	–	–
D	1.53×10^{-2}	7.66	**E:** -7.902, **F:** -4.556, **G:** 6.217	–	–
Four Multiplets, Run 2					
A	-1.48×10^{-2}	-7.38	**E:** -7.492, **F:** 0.917, **G:** 2.933	–	–
B	-1.18×10^{-2}	-5.88	**E:** -4.304, **F:** -1.815, **G:** 5.420	–	–
C	-4.32×10^{-3}	-2.16	**E:** 4.832, **F:** 7.573, **G:** 8.268	–	–
D	1.76×10^{-2}	8.80	**E:** -9.244, **F:** -1.816, **G:** -0.478	–	–
Four Multiplets, Run 3					
A	-2.23×10^{-2}	-11.17	**E:** -5.347, **F:** -1.851, **G:** 1.407	–	–
B	1.13×10^{-2}	5.66	**E:** 6.425, **F:** 7.291, **G:** 9.806	–	–
C	2.53×10^{-2}	12.64	**E:** -8.640, **F:** 0.613, **G:** 6.998	–	–
D	2.84×10^{-2}	14.21	**E:** -8.613, **F:** 0.782, **G:** 3.830	–	–
Four Multiplets, Run 4					
A	-2.03×10^{-2}	-10.16	**E:** -8.646, **F:** 6.719, **G:** 7.921	–	–
B	3.53×10^{-3}	1.77	**E:** -8.857, **F:** 4.314, **G:** 9.197	–	–
C	8.61×10^{-3}	4.30	**E:** -0.620, **F:** 1.767, **G:** 6.567	–	–
D	2.06×10^{-2}	10.30	**E:** -9.060, **F:** 2.355, **G:** 9.810	–	–
Four Multiplets, Run 5					
A	-9.16×10^{-3}	-4.58	**E:** -8.281, **F:** 1.621, **G:** 3.229	–	–
B	-2.79×10^{-3}	-1.40	**E:** 1.655, **F:** 4.219, **G:** 6.998	–	–
C	3.00×10^{-3}	1.50	**E:** -4.280, **F:** 1.045, **G:** 5.896	–	–

Spin	δ (ppm)	ω_{rot} (Hz)	J (Hz)	T_1 (s)	T_2 (s)
D	2.74×10^{-2}	13.72	**E:** -9.316, **F:** -8.322, **G:** -3.938	–	–
Sucrose					
A	6.005	1801.6	**B:** 2.285	–	–
B	3.510	1053.1	**A:** 2.285, **C:** 4.657, **H:** 4.828	–	–
C	3.934	1180.2	**B:** 4.657, **D:** 4.326	–	–
D	3.423	1027.0	**C:** 4.326, **E:** 4.851	–	–
E	4.554	1366.1	**D:** 4.851, **F:** 5.440, **G:** 2.288	–	–
F	3.891	1167.4	**E:** 5.440, **G:** −6.210	–	–
G	4.287	1286.2	**E:** 2.288, **F:** −6.210, **K:** 7.256	–	–
H	3.332	999.5	**B:** 4.828	–	–
I	1.908	572.3	–	–	–
J	1.555	466.6	–	–	–
K	0.644	193.3	**G:** 7.256	–	–
L	4.042	1212.5	**M:** −4.005, **S:** 1.460	–	–
M	4.517	1355.0	**L:** −4.005	–	–
N	3.889	1166.7	**O:** 4.253	–	–
O	4.635	1390.4	**N:** 4.253, **P:** 4.448, **U:** 3.221	–	–
P	4.160	1248.0	**O:** 4.448, **R:** 4.733	–	–
Q	4.021	1206.2	**R:** −4.182	–	–
R	4.408	1322.4	**P:** 4.733, **Q:** −4.182, **V:** 1.350	–	–
S	0.311	93.3	**L:** 1.460	–	–
T	1.334	400.2	–	–	–
U	0.893	267.9	**O:** 3.221	–	–
V	0.150	45.0	**R:** 1.350	–	–
Five Multiplets, Run 1					
A	1.33	665.99	**F:** 10.965, **G:** 12.657, **H:** 17.070	2.178	–
B	1.47	735.44	**F:** 3.610, **G:** 2.543, **H:** 8.448	4.430	–
C	1.31	653.87	**F:** 10.630, **G:** 6.282, **H:** 3.012	3.319	–
D	1.41	705.01	**F:** 8.101, **G:** 4.589, **H:** 9.068	1.007	–

Spin	δ (ppm)	ω_{rot} (Hz)	J (Hz)	T_1 (s)	T_2 (s)
E	1.30	650.23	**F:** 3.014, **G:** 16.537, **H:** 15.587	4.992	–
Five Multiplets, Run 2					
A	1.47	733.07	**F:** 19.488, **G:** 18.279, **H:** 3.147	3.600	–
B	1.36	681.25	**F:** 11.924, **G:** 8.400, **H:** 5.515	3.905	–
C	1.30	651.61	**F:** 13.672, **G:** 6.543, **H:** 16.275	4.291	–
D	1.43	713.48	**F:** 12.007, **G:** 5.141, **H:** 9.981	1.687	–
E	1.49	744.53	**F:** 8.715, **G:** 14.309, **H:** 9.805	3.214	–
Five Multiplets, Run 3					
A	1.32	658.87	**F:** 4.984, **G:** 18.119, **H:** 10.642	3.846	–
B	1.44	719.50	**F:** 13.518, **G:** 14.381, **H:** 3.074	1.018	–
C	1.37	683.34	**F:** 8.758, **G:** 16.689, **H:** 12.956	4.766	–
D	1.35	676.98	**F:** 17.648, **G:** 7.514, **H:** 3.918	3.414	–
E	1.49	746.76	**F:** 16.396, **G:** 7.455, **H:** 11.352	1.827	–
Strychinine					
A	7.167	2150.1	**B:** 7.490, **C:** 1.080, **D:** 0.230	1.74	1.26
B	7.098	2129.4	**A:** 7.490, **C:** 7.440, **D:** 0.980	2.47	1.78
C	7.255	2176.5	**A:** 1.080, **B:** 7.440, **D:** 7.900	2.50	1.80
D	8.092	2427.6	**A:** 0.230, **B:** 0.980, **C:** 7.900	5.12	3.69
E	3.860	1158.0	**I:** 10.410	1.26	0.91
F	3.132	939.6	**G:** −17.340, **H:** 3.340	0.52	0.37
G	2.670	801.0	**F:** −17.340, **H:** 8.470	0.55	0.39
H	4.288	1286.4	**F:** 3.340, **G:** 8.470, **I:** 3.300	0.89	0.64
I	1.276	382.8	**E:** 10.410, **H:** 3.300, **J:** 3.290	0.97	0.70
J	3.150	945.0	**I:** 3.290, **K:** 4.110, **L:** 1.960, **R:** 1.610, **T:** 0.470	1.07	0.77
K	2.360	708.0	**J:** 4.110, **L:** −14.350, **M:** 4.330	0.42	0.30
L	1.462	438.6	**J:** 1.960, **K:** −14.350, **M:** 2.420	0.41	0.30
M	3.963	1188.9	**K:** 4.330, **L:** 2.420	1.15	0.83
N	1.890	567.0	**O:** −13.900, **P:** 5.500, **Q:** 7.200	0.48	0.35
O	1.890	567.0	**N:** −13.900, **P:** 3.200, **Q:** 10.700	0.47	0.34

Spin	δ (ppm)	ω_{rot} (Hz)	J (Hz)	T_1 (s)	T_2 (s)
P	3.219	965.7	**N**: 5.500, **O**: 3.200, **Q**: −13.900	0.49	0.36
Q	2.878	863.4	**N**: 7.200, **O**: 10.700, **P**: −13.900	0.42	0.30
R	3.716	1114.8	**J**: 1.610, **S**: −14.800, **T**: 1.790	0.47	0.34
S	2.745	823.5	**R**: −14.800	0.45	0.33
T	5.915	1774.5	**J**: 0.470, **R**: 1.790, **U**: 7.000, **V**: 6.100	1.48	1.06
U	4.148	1244.4	**T**: 7.000, **V**: −13.800	0.48	0.34
V	4.066	1219.8	**T**: 6.100, **U**: −13.800	0.55	0.40

Parameter	Four Multiplets	Sucrose
$f_{bf}^{(1)}$ (MHz)	500	300
$f_{off}^{(2)}$ (Hz)	0	1000
$f_{off}^{(2)}$ (ppm)	0	3.333
$f_{sw}^{(1)}$ (Hz)	40	30
$f_{sw}^{(2)}$ (Hz)	1000	2200
$f_{sw}^{(2)}$ (ppm)	2	7.333
$N^{(1)}$	128	64
$N^{(2)}$	1024	4096

TABLE A2: Experiment parameters for 2DJ simulations run using SPINACH.

Parameter	Five Multiplets
$f_{bf}^{(1)}$ (MHz)	500
$f_{off}^{(1)}$ (Hz)	2500
$f_{off}^{(1)}$ (ppm)	5
$f_{sw}^{(1)}$ (Hz)	5000
$f_{sw}^{(1)}$ (ppm)	10
$N^{(1)}$	16384
K	21
τ_{max} (s)	4

TABLE A3: Experiment parameters for inversion recovery simulations run using SPINACH. K specifies the number of increments run, and τ_{max} specifies the largest delay time used. Delays were generated with linear spacings, with the first delay always being 0 s, such that the nth delay was $(n-1)\tau_{max}/K-1$.

Table of spinach experiment parameters Description of 2DJ and Invrec simulations (i.e. relaxation model used, approximations to basis used etc.
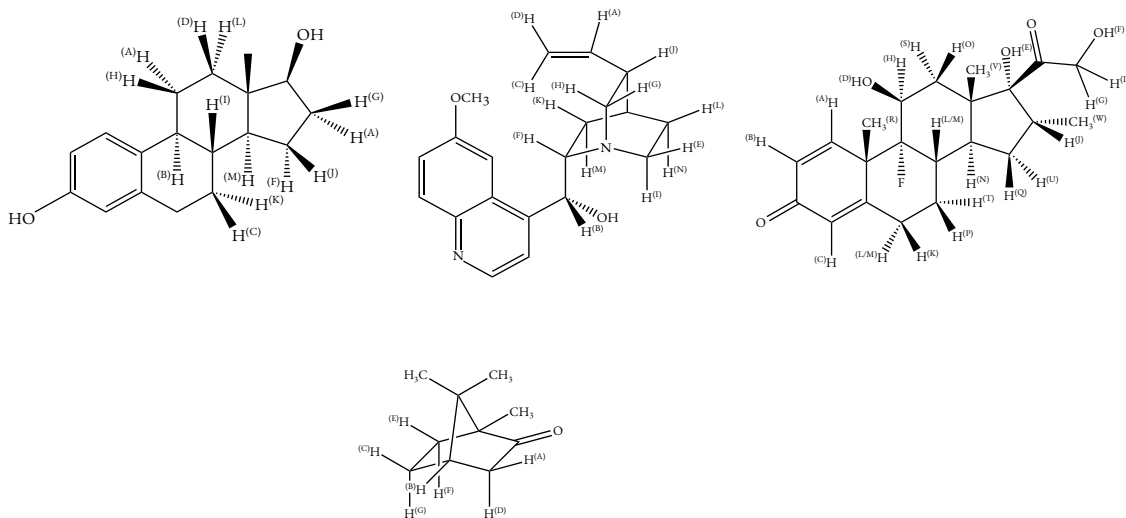
A.4   Structures

FIGURE A1: The molecular structures of species giving rise to the experimental NMR datasets considered in this work. **TODO: check assignments (esp. estradiol). Normalise fonts.**

A.5   Miscellaneous Topics of Interest

A.5.1   The Nyquist Frequency

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

$$f_N = \frac{1}{2\Delta t} \tag{A.5}$$

This quantity as at the heart of the Nyquist-Shannon Sampling Theorem, which - in the words of Claude Shannon[82] - 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:

$$-\frac{1}{2\Delta t} \leq f_j \leq \frac{1}{2\Delta t} \; \forall j \tag{A.6}$$

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

$$f_a = 2mf_N - f_j \qquad (f_j > f_N) \tag{A.7}$$

$$f_a = 2mf_N + f_j \qquad (f_j < -f_N), \tag{A.8}$$

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

A.5.2   Hessian Validation

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

A.5.3   Statistical Definitions

Definitions of standard error, likelihood function, ...