Structural methods in solid-state NMR

David A. Bennett, MSci.

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy

May 2013

Abstract

New solid-state NMR experiments for measuring internuclear distances are designed using symmetry principles. The "recoupling" sequences described here are intended to reintroduce the MAS-averaged heteronuclear dipole-dipole coupling between a spin-$\frac{1}{2}$ nucleus (e.g. ^{1}H) and a half-integer quadrupolar nucleus (e.g. ^{17}O, I = $\frac{5}{2}$). The magnitude of the dipolar interaction depends on the separation between the coupled nuclei, so the evolution of the spin system under the recoupled Hamiltonian can be used to measure the internuclear distance. Simulations of the spin dynamics are used initially to select candidate sequences and these are subsequently employed to measure both long-range and direct O–H distances in powdered L-Tyrosine.HCl (isotopically enriched with ^{17}O at 20 % to 30 % at the O$^{\eta}$ site). Improvements to existing methods for the data analysis for this type of NMR experiment are also discussed, including the restriction and/or removal of certain fit parameters and the explicit inclusion of inhomogeneous radio-frequency fields as part of the fitting procedure. The effects of processing on the uncertainty of experimentally determined distances are considered, and a new analysis method which circumvents several of these effects is presented. Similar recoupling sequences can be used to measure the anisotropy of proton chemical shifts, and some preliminary results are also presented for this application.

A systematic method for the assignment of congested spin-$\frac{1}{2}$ spectra resulting from molecules with large numbers of chemically similar sites is also described. This makes use of a comparison between the chemical shift tensor measured as usual by the 2D-PASS experiment and its principal components calculated from first principles using the density functional theory package CASTEP. The initial peak assignment is generated randomly and then varied using a steepest-ascent hill climbing algorithm with the square sum of the difference between the experimental and calculated principal values of the chemical shift tensor as the target function. The new method is tested on the ^{13}C spectrum of the anti-inflammatory drug flufenamic acid and found to be superior to simple assignments using only the isotropic chemical shift.

Declaration

The work in this thesis is based on research carried out at the solid-state NMR group, School of Chemistry, University of Nottingham, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

First and foremost, I am grateful to my supervisor Dr. Jeremy J. Titman, both for giving me the opportunity to be a part of the solid-state NMR group, but more importantly for his advice and guidance over these many years.

Next I would like to thank Dr. Dinu Iuga and Dr. Huw Williams, the facility managers of the two high-field instruments used in this work. None of the experiments in Chapter 2 would have been possible without their patient help.

I would also particularly like to acknowledge the work of my good friend Dr. David Mycock in recrystallising my samples. Working with such an expensive sample was a hair-raising task, and the quality of the final product is a testament to his skill.

Finally, I would like to thank the members of the solid-state NMR group: Dr. Andrew P. S. Jurd, Dr. Daniel F. Lee, Dr. Yanqi Wu, Dr. Andrew Powell, Gregory Martin and Habeeba Miah. Without their help, support and general good humour I would most certainly have lost my sanity.

This work was funded by the EPSRC. The UK 850 MHz solid-state NMR Facility used in this research was funded by EPSRC and BBSRC, as well as the University of Warwick including via part funding through Birmingham Science City Advanced Materials Projects 1 and 2 supported by Advantage West Midlands (AWM) and the European Regional Development Fund (ERDF).

Contents

List of Figures

List of Tables

Nomenclature

AHT	Average Hamiltonian Theory
CSA	Chemical Shift Anisotropy
DFS	Double Frequency Sweep
DFT	Density Functional Theory
GIPAW	Gauge Including Projector Augmented Waves
MAS	Magic Angle Spinning
MQMAS	Multiple-Quantum Magic-Angle Spinning
NMR	Nuclear Magnetic Resonance
NOESY	Nuclear Overhauser Effect SpectroscopY
PASS	Phase Adjusted Spinning Sidebands
RDFR	Radio-Frequency Driven Recoupling
REAPDOR	Rotational Echo Adiabatic Passage DOuble Resonance
REDOR	Rotational Echo DOuble Resonance
SEDOR	Spin Echo DOuble Resonance
ssAPT	solid-state Attached Proton Test
TEDOR	Transferred Echo DOuble Resonance
TOBSY	ThrOugh-Bond SpectroscopY
TOCSY	Total Correlation SpectroscopY
TOSS	TOtal Suppression of Sidebands
TRAPDOR	TRAnsfer of Population in DOuble Resonance
VACP	Variable Amplitude Cross-Polarization
XRD	X–Ray Diffraction

1 Introduction

Since the early days of quantum mechanics, magnetic fields and radio-frequency irradiation have been highly important methods of exploring the quantum nature of atoms and molecules. One might point to the Stern-Gerlach[1] experiment of the 1920s [1, 2], where a thin beam of silver atoms was observed to split when passed through a magnetic field; one might also consider the first resonance experiments by Rabi[2] in the late 1930s [3–5], where the magnetic moment of atoms in a beam was measured by inducing transitions with radio-frequency irradiation. The first experiments resembling modern day nuclear magnetic resonance on bulk materials were carried out independently by Purcell[3] [6] on paraffin and by Bloch [3] on water [7] in 1946 (Bloch also formulated the *Bloch equations* [8] that same year).

The range of use for NMR was greatly expanded by Ernst[4] in the mid 1960s with the introduction of pulsed Fourier transform NMR [9]; this allowed NMR spectra to be recorded much more rapidly than the continuous wave spectra which had been in use up until then, eventually making NMR indispensable to chemical research and industry as a method for the determination of chemical structures. In that vein, Ernst also introduced the concept of two-dimensional NMR [10,11] in the mid 1970s, allowing molecular connectivity to be determined through correlation methods. An example of such a method is nuclear Overhauser effect spectroscopy, developed in cooperation with Wüthrich[5] [12, 13], a technique used extensively in the determination of protein structures.

This thesis concerns methodological advances in solid-state NMR relating to the determination and evaluation of structure in solid crystals. Chapter 1 gives a brief introduction to NMR, and introduces the underlying principles using the quantum mechanical description. Chapter 2 is concerned with measurements of the

[1] Otto Stern was awarded the 1943 Nobel prize in physics.

[2] Isidor Isaac Rabi was awarded the 1944 Nobel prize in physics.

[3] Felix Bloch and Edward Mills Purcell were awarded the 1952 Nobel prize in physics.

[4] Richard R. Ernst was awarded the 1991 Nobel prize in chemistry.

[5] Kurt Wüthrich was jointly awarded the 2002 Nobel prize in chemistry with John B. Fenn and Koichi Tanaka.

distance between a spin-$\frac{1}{2}$ nucleus and a quadrupolar nucleus with half-integer spin. This is achieved by recoupling of the heteronuclear dipole-dipole interaction using a variety of symmetry based pulse sequence and a range of composite pulses. Viable sequences are tested on L-tyrosine.HCl isotopically enriched with ^{17}O. Chapter 3 demonstrates the difficulty of assigning a congested spectrum using the polymorphic, anti-inflammatory drug flufenamic acid as an example. The spectrum is then assigned by measurements of the chemical shift anisotropy parameters and a systematic least squares fitting procedure to the principal values of the chemical shielding tensor as calculated by density functional theory. The final chapter concludes with an overview of possible future work.

1.1 Quantum Mechanics

The theory of NMR begins with the time-dependent Schrödinger equation [14]:

$$\hat{H}\,|\psi\rangle(t) = i\hbar\frac{\partial\,|\psi\rangle}{\partial t} \tag{1.1.1}$$

In normal NMR usage, some of these terms have slightly narrower meanings than usual: $\hat{H}$ is the total energy operator or Hamiltonian for the nuclear spins, $|\psi\rangle(t)$ is the time-dependent wave function of the nuclear spins and $\hbar \approx 1.055 \cdot 10^{-34}\ \frac{\mathrm{J\,s}}{\mathrm{rad}}$ is the reduced Planck constant. NMR operates under the *spin Hamiltonian hypothesis* which states that the motions of electrons are sufficiently rapid that they may be averaged over the time scales of NMR experiments. In addition to this, NMR tends to operate in so-called *natural units*; we define this new Hamiltonian as $\hat{\mathscr{H}} = \frac{\hat{H}}{\hbar}$. The corresponding eigenvalues are then the angular frequencies $\omega_n = \frac{E_n}{\hbar}$ where E_n are the energy eigenvalues of $\hat{H}$. The eigenvectors $|n\rangle$ of the system remain unaffected, i.e. if $\hat{H}\,|n\rangle = E_n\,|n\rangle$, then $\hat{\mathscr{H}}|n\rangle = \omega_n\,|n\rangle$.

When working in natural units, it is also common to divide the angular momentum operators by the reduced Planck constant, giving $\hat{\boldsymbol{I}} = \frac{\hat{I}}{\hbar}$. Some common identities are:

$$\hat{\boldsymbol{I}} = \begin{pmatrix} \hat{I}_x & \hat{I}_y & \hat{I}_z \end{pmatrix} \qquad \hat{I}^2 = \hat{I}_x^2 + \hat{I}_y^2 + \hat{I}_z^2$$

$$\left[\hat{I}_x, \hat{I}_y\right] = i\hat{I}_z \qquad \circlearrowleft \qquad \hat{I}_\pm = \hat{I}_x \pm i\hat{I}_y$$

where $\circlearrowleft$ means that the equation may be permuted cyclicly. These operators are shown in Table 1.1 on page 4 for nuclei of spin $\frac{1}{2}$, $\frac{3}{2}$ and $\frac{5}{2}$. The effects of these operators on a wavefunction $\left|j, m_j\right\rangle$ with total angular momentum j and angular momentum m_j along the z axis (both in natural units, $m_j \in \{j, j-1, \ldots, 1-j, -j\}$) are:

$$\hat{I}^2 \left|j, m_j\right\rangle = j\left(j+1\right)\left|j, m_j\right\rangle \qquad \hat{I}_z \left|j, m_j\right\rangle = m_j \left|j, m_j\right\rangle$$
$$\hat{I}_\pm \left|j, m_j\right\rangle = \sqrt{j(j+1) - m_j(m_j \pm 1)}\left|j, m_j \pm 1\right\rangle$$

Hence $\hat{I}_\pm$ are known as the raising and lowering operators.

1.1.1 Hamiltonians

Now that the basics have been established, we may turn our attention to the Hamiltonians themselves. In NMR, the most important Hamiltonian by far is the Zeeman Hamiltonian $\mathscr{H}_Z$ [15]:

$$\mathscr{H}_Z = \sum_i \omega_0^i \hat{I}_z^i \tag{1.1.2}$$

where i runs over all nuclei present in the spin system, the z-axis is defined as parallel to the external magnetic field $\boldsymbol{B}_0$ and $\omega_0^i = -\gamma^i B_0$ is the Larmor frequency. γ^i is, in turn, the gyromagnetic ratio of the nucleus i; a constant for a given isotope. Since $\frac{\omega_0}{2\pi}$ is generally on the order of 100 MHz, $\hat{\mathscr{H}}_Z$ completely dominates the behaviour of the system, leading to the secular approximation. Broadly speaking[6], if the Hamiltonian can be written as $\mathscr{H} = \mathscr{H}_Z + \mathscr{H}_2$, then the off-diagonal elements in $\hat{\mathscr{H}}_2$ (when written in the Zeeman basis) can be ignored unless the energy eigenvalues of the eigenstates which they connect are degenerate, or near-degenerate.

For instance, the general form of the chemical shift Hamiltonian $\hat{\mathscr{H}}_{CS}$ is

$$\hat{\mathscr{H}}_{CS} = \sum_i \gamma^i \hat{\boldsymbol{I}}^i \boldsymbol{\sigma}^i\left(\boldsymbol{\Theta}\right) \boldsymbol{B}_0 \tag{1.1.3}$$
$$= \sum_i \gamma^i \left(\hat{I}_x^i \sigma_{xz}^i\left(\boldsymbol{\Theta}\right) + \hat{I}_y^i \sigma_{yz}^i\left(\boldsymbol{\Theta}\right) + \hat{I}_z^i \sigma_{zz}^i\left(\boldsymbol{\Theta}\right)\right) B_0 \tag{1.1.4}$$

[6] In a stricter sense, the secular approximation only applies to time-independent Hamiltonians. This means that care is sometimes required when applying RF pulses or MAS.

	$I = \frac{1}{2}$	$I = \frac{3}{2}$	$I = \frac{5}{2}$
$\hat{I}_x$	$\begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & \frac{\sqrt{3}}{2} & 0 & 0 \\ \frac{\sqrt{3}}{2} & 0 & 1 & 0 \\ 0 & 1 & 0 & \frac{\sqrt{3}}{2} \\ 0 & 0 & \frac{\sqrt{3}}{2} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & \frac{\sqrt{5}}{2} & 0 & 0 & 0 & 0 \\ \frac{\sqrt{5}}{2} & 0 & \frac{\sqrt{8}}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{8}}{2} & 0 & \frac{3}{2} & 0 & 0 \\ 0 & 0 & \frac{3}{2} & 0 & \frac{\sqrt{8}}{2} & 0 \\ 0 & 0 & 0 & \frac{\sqrt{8}}{2} & 0 & \frac{\sqrt{5}}{2} \\ 0 & 0 & 0 & 0 & \frac{\sqrt{5}}{2} & 0 \end{pmatrix}$
$\hat{I}_y$	$\begin{pmatrix} 0 & -\frac{i}{2} \\ \frac{i}{2} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -\frac{\sqrt{3}}{2}i & 0 & 0 \\ \frac{\sqrt{3}}{2}i & 0 & -i & 0 \\ 0 & i & 0 & -\frac{\sqrt{3}}{2}i \\ 0 & 0 & \frac{\sqrt{3}}{2}i & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -\frac{\sqrt{5}}{2}i & 0 & 0 & 0 & 0 \\ \frac{\sqrt{5}}{2}i & 0 & -\frac{\sqrt{8}}{2}i & 0 & 0 & 0 \\ 0 & \frac{\sqrt{8}}{2}i & 0 & -\frac{3}{2}i & 0 & 0 \\ 0 & 0 & \frac{3}{2}i & 0 & -\frac{\sqrt{8}}{2}i & 0 \\ 0 & 0 & 0 & \frac{\sqrt{8}}{2}i & 0 & -\frac{\sqrt{5}}{2}i \\ 0 & 0 & 0 & 0 & \frac{\sqrt{5}}{2}i & 0 \end{pmatrix}$
$\hat{I}_z$	$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} \end{pmatrix}$	$\begin{pmatrix} \frac{3}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & -\frac{3}{2} \end{pmatrix}$	$\begin{pmatrix} \frac{5}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{3}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{5}{2} \end{pmatrix}$
$\hat{I}^2$	$\begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} \end{pmatrix}$	$\begin{pmatrix} \frac{15}{4} & 0 & 0 & 0 \\ 0 & \frac{15}{4} & 0 & 0 \\ 0 & 0 & \frac{15}{4} & 0 \\ 0 & 0 & 0 & \frac{15}{4} \end{pmatrix}$	$\begin{pmatrix} \frac{35}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{35}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{35}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{35}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{35}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{35}{4} \end{pmatrix}$
$\hat{I}_+$	$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & \sqrt{3} & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & \sqrt{3} \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & \sqrt{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{5} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
$\hat{I}_-$	$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ \sqrt{3} & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & \sqrt{3} & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \sqrt{5} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sqrt{8} & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{5} & 0 \end{pmatrix}$

Table 1.1: Matrix forms of selected nuclear spin operators (Zeeman basis).

where $\boldsymbol{\sigma}^i(\boldsymbol{\Theta})$ is the chemical shielding tensor of the nucleus i at a given orientation $\boldsymbol{\Theta}$. Using the secular approximation, the chemical shift Hamiltonian simplifies to:

$$\hat{\mathscr{H}}_{CS}^0 = \sum_i \gamma^i \hat{I}_z^i \sigma_{zz}^i(\boldsymbol{\Theta}) B_0 \tag{1.1.5}$$

The chemical shift is caused by induced currents in the electron clouds of atoms and molecules. Since electron densities can change significantly with molecular structure, the chemical shift is highly sensitive to changes in the local chemical environment (hence the name). Even though $\boldsymbol{\sigma}$ (and therefore $\hat{\mathscr{H}}_{CS}$) is orientation dependent, it can be used for molecular identification; for sufficiently rapid, isotropic tumbling (e.g. for solution-state experiments) the chemical shift averages to the so-called *isotropic chemical shift*[7]: $\delta_{iso} = -\frac{1}{3}\operatorname{Tr}\boldsymbol{\sigma}$. This is not generally the case in solids (except for a few special cases such as buckminsterfullerene or adamantane) where $\hat{\mathscr{H}}_{CS}$ can broaden a single resonance across the entire isotropic shift range. This *Chemical Shift Anisotropy* (CSA) can be used to measure the full chemical shielding tensor for very small molecules, but spectral overlap quickly becomes an insurmountable problem. This is usually dealt with using *Magic Angle Spinning* (MAS), which is discussed in section 3.1, and the chemical shielding tensor can be measured using techniques such as 2D PASS, which are discussed in section 3.2.

Next there is the dipole-dipole coupling Hamiltonian $\hat{\mathscr{H}}_D$, a two-spin interaction which exists because the magnetic field produced by the nucleus i affects the effective magnetic field experienced by nucleus j. In the secular approximation, $\hat{\mathscr{H}}_D$ is:

$$\hat{\mathscr{H}}_D = -\frac{\mu_0\hbar}{4\pi}\sum_{i<j}\frac{\gamma_i\gamma_j\left(3\cos^2\theta_{ij}-1\right)\left(\hat{\boldsymbol{I}}^i\cdot\hat{\boldsymbol{I}}^j - 3\hat{I}_z^i\hat{I}_z^j\right)}{r_{ij}^3} \tag{1.1.6}$$

where $\mu_0 = 4\pi\cdot 10^{-7}\,\frac{\mathrm{V\,s}}{\mathrm{A\,m}}$ is the permeability of free space, r_{ij} is the distance between the nuclei i & j and θ_{ij} is the angle between the internuclear vector $\boldsymbol{r}_{ij}$ and the magnetic field $\boldsymbol{B}_0$. For the sake of brevity, it is common to define the dipole-dipole coupling constant $b_{ij} = \frac{\mu_0\hbar\gamma_i\gamma_j}{4\pi r_{ij}^3}$. $\hat{\mathscr{H}}_D$ only depends on the direction of $\boldsymbol{B}_0$, not

[7] Technically, Eqn. 1.1.4 uses the absolute shielding tensor ($\boldsymbol{\sigma}_{abs}$) which is referenced relative to the bare nucleus while the relative shielding tensor used here is derived from the chemical shift tensor $\boldsymbol{\delta}$: $\boldsymbol{\sigma}_{rel} = -\boldsymbol{\delta}$. $\boldsymbol{\sigma}_{abs}$ and $\boldsymbol{\sigma}_{rel}$ have slightly different traces, but the distinction is often ignored.

on the field strength, so its magnitude is usually given in frequency units rather than ppm. $\frac{b_{ij}}{2\pi}$ can vary across several orders of magnitude, but the cases considered here in Chapter 2 are generally on the order of 3 kHz to 15 kHz as these are typical values for ^{1}H bonded to ^{17}O via a covalent bond or a hydrogen bond.

There is also a similar interaction known as the indirect dipole-dipole interaction or J-coupling, whose Hamiltonian is:

$$\hat{\mathscr{H}}_J = 2\pi \sum_{i<j} \hat{\boldsymbol{I}}^i \boldsymbol{J}_{ij} \hat{\boldsymbol{I}}^j \tag{1.1.7}$$

$\hat{\mathscr{H}}_J$ is like $\hat{\mathscr{H}}_D$ in that it describes the effect that the magnetic field of nucleus i has on nucleus j and vice versa; in older text books the two are sometimes even presented as one Hamiltonian. The main difference is that the interaction is mediated by the electrons in the internuclear bonds. While quite useful in solution state NMR, its interaction strength is generally on the order of 100 Hz which is smaller than the linewidth of a peak in a typical solid state NMR spectrum.

The last nuclear spin Hamiltonian to discuss is the quadrupole interaction. Nuclei with $I > \frac{1}{2}$ have non-spherical charge distributions, meaning that the energy of a given state depends both on the orientation of the nuclear magnetic moment relative to $\boldsymbol{B}_0$ and the alignment of the nuclear charge distribution with respect to the local electrical environment (termed the *Electric Field Gradient*, EFG). There are more parts to the series expansion of the nuclear charge distribution, however the next non-zero term is the nuclear octopole moment, which is usually ignored. The nuclear quadrupole Hamiltonian can be written as:

$$\hat{\mathscr{H}}_Q = \frac{eQ}{2I\,(2I-1)} \hat{\boldsymbol{I}} \boldsymbol{V}(\boldsymbol{\Theta}) \hat{\boldsymbol{I}} \tag{1.1.8}$$

where $e = 1.602 \cdot 10^{-19}\,\mathrm{C}$ is the elementary charge, Q is the nuclear quadrupole moment ($Q_{^{17}\mathrm{O}} = -2.558 \cdot 10^{-30}\,\mathrm{m}^2$, $Q_{^{23}\mathrm{Na}} = 10.4 \cdot 10^{-30}\,\mathrm{m}^2$) and $\boldsymbol{V}(\boldsymbol{\Theta})$ is the EFG tensor at the nucleus for a given orientation ($\boldsymbol{\Theta}$). $\hat{\mathscr{H}}_Q$ can be simplified under the secular approximation, however it does not generally meet the "sufficiently small" criterion used in the perturbation expansion, so multiple orders may be required.

The first order Hamiltonian is shown in Equation 1.1.10:

$$\hat{\mathscr{H}}_Q = \hat{\mathscr{H}}_Q^{(1)} + \hat{\mathscr{H}}_Q^{(2)} + \cdots \tag{1.1.9}$$

$$\hat{\mathscr{H}}_Q^{(1)} = \omega_Q^{(1)} \frac{1}{6} \left(3\hat{I}_z^2 - I\left(I+1\right) \hat{\mathbf{1}} \right) \tag{1.1.10}$$

$$\omega_Q^{(1)} = \frac{3eQ\overline{V}_{zz}}{2I(2I-1)\hbar} \tag{1.1.11}$$

where $\overline{V}_{zz}$ is the average of V_{zz} across all possible orientations.

It is also common to describe the strength of the quadrupolar Hamiltonian using the so-called quadrupolar coupling constant C_Q. There doesn't seem to be a complete consensus with regards to the definition of C_Q (or $\omega_Q^{(1)}$ for that matter); $C_Q = \frac{eV_{ZZ}Q}{h}$ [16] will be used here, where V_{ZZ} is the zz component of $\boldsymbol{V}$ in the tensor's principal axis system[8].

Finally, we have the Hamiltonian of an external radio-frequency pulse.

$$\hat{\mathscr{H}}_{RF} = \sum_i -\omega_{nut}^i \left\{ \cos\left(\omega_{ref} t + \phi_p\right) \hat{I}_x^i + \sin\left(\omega_{ref} t + \phi_p\right) \hat{I}_y^i \right\} \tag{1.1.12}$$

where ω_{nut}^i is the nutation frequency of nucleus i, ω_{ref} is the frequency of the applied RF field and ϕ_p is the phase of the pulse. The nutation frequency can be calculated as $\omega_{nut}^i = \left| \frac{1}{2} \gamma_i B_{RF} \sin\theta_{RF} \right|$ with B_{RF} the maximum amplitude of the applied RF field and θ_{RF} the angle between the RF field and $\boldsymbol{B}_0$.

1.1.2 Spin Density Operator

Now that the Hamiltonians have been introduced, it is time to discuss the spins themselves. As an example, let us consider the wave function of a spin-1 nucleus:

$$|\psi\rangle = c_\oplus |\oplus\rangle + c_\odot |\odot\rangle + c_\ominus |\ominus\rangle \qquad \langle\psi| = c_\oplus^* \langle\oplus| + c_\odot^* \langle\odot| + c_\ominus^* \langle\ominus|$$

$$= \begin{pmatrix} c_\oplus \\ c_\odot \\ c_\ominus \end{pmatrix} \qquad = \begin{pmatrix} c_\oplus^* & c_\odot^* & c_\ominus^* \end{pmatrix}$$

[8] N.B. $|V_{ZZ}| \geq |V_{YY}| \geq |V_{XX}|$, $\mathrm{Tr}\,\boldsymbol{V} = 0$

where $|\psi\rangle$ is the wavefunction, $\langle\psi| = |\psi\rangle^\dagger$ is the complex conjugate of the wavefunction and the vectors are written in the Zeeman basis ordered by m_j: $\hat{\mathscr{H}}|\oplus\rangle = \omega_\oplus |\oplus\rangle + \ldots$ and $m_\oplus$, $m_\odot$ and $m_\ominus$ are 1, 0 and -1 respectively. The wavefunction is also assumed to be normalised, i.e. $\langle\psi|\psi\rangle = \sum_i c_i c_i^* = 1$.

Furthermore, if we wish to know the expectation value of some quantity Q with operator $\hat{Q}$ with respect to our wavefunction, we use:

$$\left\langle \hat{Q} \right\rangle = \langle\psi|\hat{Q}|\psi\rangle \tag{1.1.13}$$

It is simple but tedious to show that

$$\left\langle \hat{Q} \right\rangle = \mathrm{Tr}\left(|\psi\rangle\langle\psi|\hat{Q}\right) \tag{1.1.14}$$

where $|\psi\rangle\langle\psi|$ is a so-called projection operator. Its matrix form can easily be worked out:

$$|\psi\rangle\langle\psi| = \begin{pmatrix} c_\oplus \\ c_\odot \\ c_\ominus \end{pmatrix} \begin{pmatrix} c_\oplus^* & c_\odot^* & c_\ominus^* \end{pmatrix} \tag{1.1.15}$$

$$= \begin{pmatrix} c_\oplus c_\oplus^* & c_\oplus c_\odot^* & c_\oplus c_\ominus^* \\ c_\odot c_\oplus^* & c_\odot c_\odot^* & c_\odot c_\ominus^* \\ c_\ominus c_\oplus^* & c_\ominus c_\odot^* & c_\ominus c_\ominus^* \end{pmatrix} \tag{1.1.16}$$

For a single spin system this is merely a pointless exercise in algebra, but it turns out that an ensemble of $\mathbb{N}$ identical spin systems, each with wavefunction $|\psi_i\rangle$, obeys a similar equation:

$$\left\langle \hat{Q} \right\rangle = \mathrm{Tr}\left\{ \left(\sum_i |\psi_i\rangle\langle\psi_i| \right) \hat{Q} \right\} \tag{1.1.17}$$

We therefore define the *density operator* $\hat{\boldsymbol{\rho}}$:

$$\hat{\boldsymbol{\rho}} = \mathbb{N}^{-1} \sum_i |\psi_i\rangle\langle\psi_i| \tag{1.1.18}$$

$$= \overline{|\psi\rangle\langle\psi|} \tag{1.1.19}$$

and any macroscopic quantity Q may be estimated using:

$$\left\langle \hat{Q} \right\rangle = \mathbb{N}\,\mathrm{Tr}\left\{\hat{\boldsymbol{\rho}}\hat{Q}\right\} \tag{1.1.20}$$

Of course, the most interesting operators from a NMR perspective are $\hat{M}_x = \gamma\hbar\hat{I}_x$, $\hat{M}_y = \gamma\hbar\hat{I}_y$ and $\hat{M}_z = \gamma\hbar\hat{I}_z$.

The Schrödinger equation must be adapted slightly for use with $\hat{\boldsymbol{\rho}}$. The result is the Liouville-von Neumann equation:

$$\frac{\mathrm{d}\hat{\boldsymbol{\rho}}}{\mathrm{d}t} = -i\left[\hat{\mathscr{H}}(t), \hat{\boldsymbol{\rho}}(t)\right] \tag{1.1.21}$$

Assuming that $\hat{\boldsymbol{\rho}}(t=0)$ is known, this equation can in principle be used directly to obtain numerical solutions of $\hat{\boldsymbol{\rho}}(t)$. However, it is more common to work with time evolution operators called propagators (U), defined such that:

$$\hat{\boldsymbol{\rho}}(t_2) = U(t_2, t_1)\hat{\boldsymbol{\rho}}(t_1)U^{\dagger}(t_2, t_1) \tag{1.1.22}$$

if the Hamiltonian is constant over a time period from t to $t + \Delta t$, then the propagator can be calculated according to

$$U(t + \Delta t, t) = e^{-i\hat{\mathscr{H}}\Delta t} \tag{1.1.23}$$

where $e^{\hat{\mathscr{H}}}$ is expanded using a Maclaurin series such that

$$e^{\hat{\mathscr{H}}} = \sum_{j=0}^{\infty} \frac{\hat{\mathscr{H}}^j}{j!} \tag{1.1.24}$$

If the Hamiltonian is not constant over the required time period, the propagator can still be calculated as shown in Equation 1.1.26 by breaking the time period up into n short times Δt chosen such that the Hamiltonian can be considered constant over the duration of each segment.

$$U(t_2, t_1) = \hat{T}e^{-i\int_{t_1}^{t_2}\hat{\mathscr{H}}(t')\mathrm{d}t'} \tag{1.1.25}$$

$$= \hat{T}\prod_{j=0}^{n-1} e^{-i\hat{\mathscr{H}}(j\Delta t)\Delta t} \tag{1.1.26}$$

where $\hat{T}$ is the so-called *Dyson time ordering operator* which ensures that the Hamiltonians are multiplied in order of increasing time.

Of course, this is only of use when starting from a known density matrix. The most obvious starting point is the equilibrium density matrix $\hat{\boldsymbol{\rho}}^{eq}$. Given sufficient

time, a sample will relax to a state with no coherence, and populations given by the Boltzmann distribution.

$$\rho^{eq}_{p,q} = \begin{cases} \frac{e^{-E_p/k_BT}}{\sum_r e^{-E_r/k_BT}} & \text{if } p = q \\ 0 & \text{otherwise} \end{cases} \tag{1.1.27}$$

The Zeeman energy eigenvalues for the $I = 1$ case shown earlier would be:

$$\omega_\oplus = \omega_0 \qquad \omega_\odot = 0 \qquad \omega_\ominus = -\omega_0$$

with $\omega_0 = \gamma B_0$. For the sake of brevity, we define:

$$\mathbb{B} = \frac{\hbar\gamma B_0}{k_B T} \tag{1.1.28}$$

As long as[9] $\mathbb{B} \ll 1$, $e^{\mathbb{B}} \approx 1 + \mathbb{B}$, so:

$$e^{-E_\oplus/k_BT} = e^{\mathbb{B}} \approx 1 + \mathbb{B} \qquad e^{-E_\odot/k_BT} = e^0 = 1 \qquad e^{-E_\ominus/k_BT} = e^{-\mathbb{B}} \approx 1 - \mathbb{B}$$

Therefore

$$\sum_s e^{-E_s/k_BT} \approx (1 + \mathbb{B}) + 1 + (1 - \mathbb{B}) \tag{1.1.29}$$

$$= 3 \tag{1.1.30}$$

so

$$\hat{\boldsymbol{\rho}}^{eq} \approx \begin{pmatrix} \frac{1+\mathbb{B}}{3} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1-\mathbb{B}}{3} \end{pmatrix} \tag{1.1.31}$$

$$= \frac{1}{3}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \frac{1}{3}\mathbb{B}\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{1.1.32}$$

$$= \frac{1}{3}\hat{\boldsymbol{1}} + \frac{1}{3}\mathbb{B}\hat{I}_z \tag{1.1.33}$$

[9] Even ^{1}H in a 20 T magnet at liquid He temperatures has $\mathbb{B} \approx 0.01$

Since $\hat{\mathbf{1}}$, the normalisation factor $\frac{1}{2I+1}$ and $\mathbb{B}$ cannot be changed by RF irradiation or relaxation, they will sometimes be omitted. In this case, Equation 1.1.33 would be written simply as

$$\hat{\boldsymbol{\rho}}^{eq} = \hat{I}_z$$

Intuitively, we expect $\hat{\boldsymbol{\rho}}^{eq}$ to be polarised by the magnetic field; this can easily be verified using Equation 1.1.20:

$$\begin{aligned}
\left\langle \hat{M}_z \right\rangle &= \mathrm{Tr}\left\{ \hat{\boldsymbol{\rho}}^{eq} \hat{M}_z \right\} && (1.1.34)\\
&= \mathrm{Tr}\left\{ \frac{1}{2I+1} \left(\hat{\mathbf{1}} + \mathbb{B} I_z \right) \gamma\hbar \hat{I}_z \right\} && (1.1.35)\\
&= \frac{\gamma\hbar}{2I+1} \mathrm{Tr}\left\{ \hat{I}_z + \mathbb{B}\hat{I}_z^2 \right\} && (1.1.36)\\
&= \frac{\gamma\hbar\mathbb{B}}{2I+1} \mathrm{Tr}\left\{ \hat{I}_z^2 \right\} && (1.1.37)
\end{aligned}$$

Since $\hat{I}_z$ is real, $\mathrm{Tr}\,\hat{I}_z^2$ must be positive. Similarly, we can confirm that $\left\langle \hat{M}_x \right\rangle = 0$ as $\mathrm{Tr}\left\{ \hat{I}_z \hat{I}_x \right\} = 0$.

1.1.3 Time Evolution

If three operators commute $\left(\left[\hat{A}, \hat{B} \right] = i\hat{C} \right)$, then it can be shown that:

$$e^{-i\theta\hat{A}} \hat{B} e^{i\theta\hat{A}} = \hat{B}\cos\theta + \hat{C}\sin\theta \tag{1.1.38}$$

This is convenient for the time evolution of density matrices as it closely mirrors equations 1.1.22 and 1.1.23. For instance, if we consider a density matrix $\hat{\boldsymbol{\rho}} = \frac{1}{2I+1}\left(\hat{\mathbf{1}} + \mathbb{B}\hat{I}_- \right)$ for a single spin at resonant frequency ω, then the propagator can be written as

$$\begin{aligned}
U\left(t + \Delta t, t\right) &= e^{-i\hat{\mathscr{H}}\Delta t} && (1.1.39)\\
&= e^{-i\omega\Delta t \hat{I}_z} && (1.1.40)
\end{aligned}$$

We then define $R_z\left(\beta\right) = e^{-i\beta\hat{I}_z}$ with $\beta = \omega\Delta t$, and therefore:

$$U\left(t + \Delta t, t\right) = R_z\left(\beta\right) \tag{1.1.41}$$

Since the commutator for $\hat{I}_z$ and $\hat{I}_-$ is $\left[\hat{I}_z, \hat{I}_-\right] = -\hat{I}_-$, we can see that:

$$\hat{\boldsymbol{\rho}}\left(t+\Delta t\right) = U\left(t+\Delta t, t\right)\hat{\boldsymbol{\rho}}\left(t\right)U^{\dagger}\left(t+\Delta t, t\right) \tag{1.1.42}$$

$$= \frac{1}{2I+1}\left(\hat{\mathbf{1}} + \mathbb{B}R_z\left(\beta\right)\hat{I}_- R_z\left(-\beta\right)\right) \tag{1.1.43}$$

$$= \frac{1}{2I+1}\left(\hat{\mathbf{1}} + \mathbb{B}\left(\hat{I}_- \cos\beta + i\hat{I}_- \sin\beta\right)\right) \tag{1.1.44}$$

$$= \frac{1}{2I+1}\left(\hat{\mathbf{1}} + \mathbb{B}e^{i\beta}\hat{I}_-\right) \tag{1.1.45}$$

So, if relaxation is neglected, $\hat{I}_-$ will oscillate at frequency ω. Since modern spectrometers measure $\left\langle \hat{M}_x \right\rangle + i\left\langle \hat{M}_y \right\rangle = \mathrm{Tr}\left\{\hat{\boldsymbol{\rho}}\hat{I}_+\right\}$, only the $\hat{I}_-$ portion of the density matrix is relevant; this gives rise to an oscillating signal. If relaxation is included, the signal still oscillates at frequency ω, however it also decays exponentially.

Similar arguments can be applied to the RF Hamiltonian (Equation 1.1.12), though the task is greatly simplified by first transferring to the *rotating frame*, a reference frame rotated by $\Phi = \omega_{ref}t + \phi_{ref}$ about the z-axis relative to the laboratory frame. The density matrix is converted according to $\widetilde{\hat{\boldsymbol{\rho}}} = R_z\left(-\Phi\right)\hat{\boldsymbol{\rho}}R_z\left(\Phi\right)$, so the simplest density matrix to start with is $\hat{\boldsymbol{\rho}}^{eq}$ since $\hat{\mathbf{1}}$ & $\hat{I}_z$ commute with $\hat{I}_z$, meaning that $\widetilde{\hat{\boldsymbol{\rho}}}^{eq} = \hat{\boldsymbol{\rho}}^{eq}$. Hamiltonians follow a similar pattern:

$$\widetilde{\hat{\mathscr{H}}} = R_z\left(-\Phi\right)\hat{\mathscr{H}}R_z\left(\Phi\right) - \omega_{ref}\hat{I}_z \tag{1.1.46}$$

The simplest possible Hamiltonian we could use is then $\hat{\mathscr{H}} = \hat{\mathscr{H}}_{\mathcal{Z}} + \hat{\mathscr{H}}_{RF}$ with $\omega_{ref} = \omega_0$ and $\phi_p = 0$. Inserting equations 1.1.2 and 1.1.12 into Equation 1.1.46 gives:

$$\widetilde{\hat{\mathscr{H}}} = R_z\left(-\Phi\right)\left\{\omega_0\hat{I}_z - \omega_{nut}\left[\cos\left(\omega_0 t\right)\hat{I}_x + \sin\left(\omega_0 t\right)\hat{I}_y\right]\right\}R_z\left(\Phi\right) - \omega_0\hat{I}_z \tag{1.1.47}$$

$$= \omega_0\hat{I}_z + R_z\left(-\Phi\right)\left\{-\omega_{nut}\left[R_z\left(\omega_0 t\right)\hat{I}_x R_z\left(-\omega_0 t\right)\right]\right\}R_z\left(\Phi\right) - \omega_0\hat{I}_z \tag{1.1.48}$$

If $\phi_{ref} = 0$, then $\Phi = \omega_0 t$, hence

$$\widetilde{\hat{\mathscr{H}}} = \omega_{nut}\hat{I}_x \tag{1.1.49}$$

Naturally, we can now follow the same procedure as for the Zeeman interaction to show that:

$$\widetilde{\hat{\boldsymbol{\rho}}}\left(t\right) = R_x\left(\omega_{nut}t\right)\widetilde{\hat{\boldsymbol{\rho}}}^{eq}R_x\left(-\omega_{nut}t\right) \tag{1.1.50}$$

$$= \frac{1}{2I+1}\left(\hat{\mathbf{1}} + \mathbb{B}\left\{\hat{I}_z\cos\left(\omega_{nut}t\right) + \hat{I}_y\sin\left(\omega_{nut}t\right)\right\}\right) \tag{1.1.51}$$

From this we conclude that RF irradiation converts a portion of (undetectable) $\hat{I}_z$ magnetisation into (detectable) $\hat{I}_y$. Since this is occurring in the rotating frame, the laboratory frame magnetisation will actually interconvert between $\hat{I}_x$ and $\hat{I}_y$ with frequency ω_{ref}.

Unfortunately, using such a simple Hamiltonian is only valid if the spin-system is on or near resonance, with near resonance being defined as $\left|\omega_{ref} - \omega\right| \ll \omega_{nut}$. CSA and dipole-dipole interactions tend to be on the order of 10 kHz while nutation frequencies in excess of 100 kHz have been routine for decades, so *strong pulses* are generally possible for most spin-half nuclei. The quadrupole coupling, on the other hand, is usually at least several MHz, so there are only a handful of quadrupolar nuclei (e.g. ^{2}H) where strong excitation is even remotely possible[10].

Weak pulses have a particularly interesting effect in quadrupolar nuclei with half-integer spin. The first-order quadupole coupling commutes with $\hat{I}_z$, meaning that the rotating frame Hamiltonian for a single nucleus with RF is:

$$\widetilde{\hat{\mathscr{H}}} = \omega_{nut}\hat{I}_x + \omega_Q^{(1)}\frac{1}{6}\left(3\hat{I}_z^2 - I(I+1)\hat{\mathbf{1}}\right) \tag{1.1.52}$$

Since this Hamiltonian is time independent, we may apply the secular approximation. $\hat{I}_z^2 - \hat{\mathbf{1}}$ has many degenerate eigenvalues, but $\hat{I}_x$ only overlaps with them for $\left|m_j\right| = \frac{1}{2}$; this means that most of the density matrix remains completely unaffected by weak RF pulses except for the central portion which undergoes nutation as if it were an isolated nucleus of spin $I = \frac{1}{2}$. The details of working out the propagators for this secular Hamiltonian are long and complicated, but the upshot is that R_x^c is equivalent to $\hat{\mathbf{1}}$ with R_x for $I = \frac{1}{2}$ spliced into the centre and a new nutation frequency $\omega_{nut}^c = \left(I + \frac{1}{2}\right)\omega_{nut}$ where ω_{nut}^c is the effective nutation rate of the central transition and ω_{nut} is the nutation rate if $\omega_Q^{(1)}$ were small. This relationship can be extremely useful as the quadrupolar coupling is greatly diminished in solutions due to isotropic tumbling, meaning that pulse powers for solids with high $\omega_Q^{(1)}$ can be calibrated by dissolving some of the sample.

[10] ω_Q depends on $\boldsymbol{V}$ as well as Q, so highly symmetrical molecular environments can help.

1.1.4 Spherical Tensors

While the word tensor has been used since the middle of the 19th century, much of what we would now think of as tensors came into existence under the name of *absolute differential calculus* [17, 18]. Tensors are designed to be independent of reference frame; this makes them extremely useful for describing physical phenomena, but also initially somewhat peculiar to work with as a tensor technically stops being a tensor if it is written down numerically for a given frame. To make matters worse, it is customary to work with *spherical tensors* when discussing angular momentum in quantum mechanics [19,20]. Essentially, Cartesian tensors are converted into a new set of basis vectors whose transformation properties resemble those of Laplace's spherical harmonics $Y_l^m(\theta,\phi) = Ne^{im\phi}P_l^m(\cos\theta)$.

Generally, all interactions in NMR can be written in the form $\hat{\mathscr{H}} = c\boldsymbol{a}\boldsymbol{M}\boldsymbol{b}$ where c is a constant of proportionality, $\boldsymbol{a}$ & $\boldsymbol{b}$ are vectors of some kind and $\boldsymbol{M}$ is a rank 2 Cartesian tensor. For the chemical shift Hamiltonian, c is γ, $\boldsymbol{a}$ is $\hat{\boldsymbol{I}}$, $\boldsymbol{b}$ is $\boldsymbol{B}_0$ and $\boldsymbol{M}$ is $\boldsymbol{\sigma}$. $\boldsymbol{\sigma}$ is *reducible* according to representation theory, meaning that it consists of several components which behave differently under rotation. This becomes apparent when comparing solid-state NMR to solution-state NMR. In solids, the chemical shift of a given crystallite varies with its orientation, giving rise to a broad powder line shape. In solutions on the other hand, random rotations average the interaction to δ_{iso}.

The isotropic chemical shift is easy to find: $-\delta_{iso} = \sigma_{iso} = \frac{1}{3}\,\mathrm{Tr}\{\boldsymbol{\sigma}\}$. The remainder of the chemical shielding tensor can be broken down into an antisymmetric tensor $\sigma_{ij}^{anti} = \frac{1}{2}(\sigma_{ij} - \sigma_{ji})$ and a symmetric tensor $\sigma_{ij}^{sym} = \frac{1}{2}(\sigma_{ij} - \sigma_{ji}) - \sigma_{iso}\delta_{ij}$ where δ_{ij} is the Kronecker delta. An antisymmetric 3×3 matrix can only have three independent values, so it could be written as $\boldsymbol{\sigma}^{anti} = \sigma_1^{anti}\boldsymbol{T}_1^{anti} + \sigma_2^{anti}\boldsymbol{T}_2^{anti} + \sigma_3^{anti}\boldsymbol{T}_3^{anti}$ with:

$$\boldsymbol{T}_1^{anti} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \boldsymbol{T}_2^{anti} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \boldsymbol{T}_3^{anti} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \tag{1.1.53}$$

for instance. σ_α^{anti} and $\boldsymbol{T}_\alpha^{anti}$ ($\alpha \in \{1,2,3\}$) now form an irreducible representation

of $\boldsymbol{\sigma}^{anti}$, meaning that any rotation of $\boldsymbol{\sigma}^{anti}$ can be written as a linear combination of $\boldsymbol{T}_\alpha^{anti}$; one merely needs to work out σ_α^{anti} in the new basis. The same arguments can be applied to $\boldsymbol{\sigma}^{sym}$, except that there are 5 components[11]. The original matrix can be reconstructed according to

$$\boldsymbol{\sigma} = \sigma^{iso}\hat{\boldsymbol{1}} + \sum_\alpha \sigma_\alpha^{anti}\boldsymbol{T}_\alpha^{anti} + \sum_\beta \sigma_\beta^{sym}\boldsymbol{T}_\beta^{sym} \tag{1.1.54}$$

with $\alpha \in \{1,2,3\}$ and $\beta \in \{1,2,3,4,5\}$[12]. Of course, the choice of $\boldsymbol{T}_\alpha^{anti}$ at 1.1.53 is completely arbitrary. In quantum mechanics, it is not common to describe angular momentum with the Euclidean unit vectors $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_z$. Instead, spherical vectors are defined as follows:

$$r_1^0 = r_z \qquad r_1^{\pm 1} = \frac{1}{\sqrt{2}}\left(-ir_y \mp r_x\right) \tag{1.1.55}$$

$$\hat{\boldsymbol{e}}_1^0 = \hat{\boldsymbol{e}}_z \qquad \hat{\boldsymbol{e}}_1^{\pm 1} = \frac{1}{\sqrt{2}}\left(-i\hat{\boldsymbol{e}}_y \mp \hat{\boldsymbol{e}}_x\right) \tag{1.1.56}$$

A general position vector $\boldsymbol{r}$ is then represented as follows:

$$\boldsymbol{r} = r_x\hat{\boldsymbol{e}}_x + r_y\hat{\boldsymbol{e}}_y + r_z\hat{\boldsymbol{e}}_z \tag{1.1.57}$$

$$= -r_1^1\hat{\boldsymbol{e}}_1^{-1} + r_1^0\hat{\boldsymbol{e}}_1^0 - r_1^{-1}\hat{\boldsymbol{e}}_1^1 \tag{1.1.58}$$

$$= \sum_{\mu=-1}^{1} (-1)^\mu\, r_1^\mu \hat{\boldsymbol{e}}_1^{-\mu} \tag{1.1.59}$$

Assuming the vector $\boldsymbol{r}$ has polar co-ordinates r, θ & ϕ, it can be shown that:

$$r_1^\mu = \sqrt{\frac{4\pi}{3}} r Y_1^\mu(\theta,\phi) \tag{1.1.60}$$

Interestingly, it can also be shown that $r_1^{\mu *} = (-1)^\mu\, r_1^{-\mu}$ and that the dot product of two vectors $\boldsymbol{a}$ & $\boldsymbol{b}$ is:

$$\boldsymbol{a}\cdot\boldsymbol{b} = \sum_{\mu=-1}^{1} a_1^\mu b_1^{\mu *} \tag{1.1.61}$$

[11] Symmetric 3×3 matrices generally have 6 degrees of freedom, but the removal of σ^{iso} means $\boldsymbol{\sigma}^{sym}$ must be traceless.

[12] N.B. The Einstein summation convention will not be used in this document. NMR is sufficiently full of superscripts and subscripts that confusion might arise.

Converting a 2nd rank tensor into spherical coordinates is a little more involved; the required equations can be found on page 68 of Reference [21] and the resulting *irreducible spherical tensors* are also listed at the end of Appendix B in Reference [22]. The tensors $\hat{\boldsymbol{T}}_{mn}$ are said to have *rank* m (this rank is slightly different to what one would normally call the "rank" of a Euclidean tensor) where m ranges from 0 to 2. Each rank has a label n where $n \in \{-m, -m+1, \ldots, m-1, m\}$. As an example, one of the spherical tensors for the dipole-dipole coupling is $\hat{\boldsymbol{T}}^D_{20} = \sqrt{\frac{1}{6}}\left(3\hat{I}_z\hat{S}_z + \hat{\boldsymbol{I}}\hat{\boldsymbol{S}}\right)$. Just as before, the Hamiltonian $\hat{\mathscr{H}}_\Lambda = c\boldsymbol{a}\boldsymbol{M}\boldsymbol{b}$ can be reconstructed according to

$$\hat{\mathscr{H}}_\Lambda = c\sum_{m=0}^{2}\sum_{n=-m}^{m}(-1)^{-n}M^\Lambda_{m-n}\hat{\boldsymbol{T}}^\Lambda_{mn} \tag{1.1.62}$$

Calculating any given $\hat{\boldsymbol{T}}^\Lambda_{mn}$ is a purely mechanical matter, but the coefficients M^Λ_{mn} can offer genuine insight into the nature of NMR interactions. For instance, the antisymmetric part of most NMR interactions has no effect on single pulse experiments and little or no effect on multi-pulse experiments, so we generally set $M^\Lambda_{1n} = 0$ for all three n values. If a tensor is symmetric, it is possible to find a frame in which it is diagonal; this is known as the *principal axis frame* or PAF. Let us define $\boldsymbol{\mu} = \boldsymbol{M}^{PAF} - \frac{1}{3}\operatorname{Tr}\{\boldsymbol{M}\}\hat{\boldsymbol{1}}$ with $\boldsymbol{M}^{PAF}$ chosen such that $|\mu_{ZZ}| \geq |\mu_{YY}| \geq |\mu_{ZZ}|$ (in Cartesian coordinates). We then define the principal value $\zeta = \mu_{ZZ}$ and the asymmetry parameter $\eta = \frac{\mu_{YY}-\mu_{XX}}{\zeta}$. The coefficients for the irreducible spherical tensors are now:

$$M_{00} = M^{iso} \tag{1.1.63}$$

$$M_{10} = 0 \qquad M_{1\pm1} = 0 \tag{1.1.64}$$

$$M_{20} = \sqrt{\frac{3}{2}}\zeta \qquad M_{2\pm1} = 0 \qquad M_{2\pm2} = -\frac{1}{2}\zeta\eta \tag{1.1.65}$$

The coefficients for orientations which do not correspond to the PAF can be found using a Wigner rotation:

$$M_{ij}(\boldsymbol{\Theta}) = \sum_{k=-m}^{m} M^{PAF}_{mk}D^{(m)}_{kn}(\boldsymbol{\Theta}) \tag{1.1.66}$$

where $D^{(m)}_{kn}(\boldsymbol{\Theta})$ are the elements of the Wigner Rotation matrix:

$$D^{(m)}_{kn}(\boldsymbol{\Theta}) = e^{-ik\alpha}d^{(m)}_{kn}(\beta)\,e^{-in\gamma} \tag{1.1.67}$$

and $d_{kn}^{(m)}(\beta)$ is the Wigner small d-matrix, whose analytical formula will not be reproduced here.

This allows us to characterise various interactions. For instance, only the Zeeman and Chemical Shift Hamiltonians have isotropic components, and the dipole-dipole interaction is especially simple since its asymmetry $\eta_D = 0$ as well.

2 Recoupling Sequences

2.1 Introduction

X–ray diffraction (XRD) is one of the mainstays of crystallography; it provides accurate positions of atoms within the unit cell under two conditions:

- sufficiently large crystals can be grown
- the atoms of interest have a sufficiently large scattering cross-section

Since an atom's scattering cross-section is linked to its mass, it is common practice to assign the positions of heavy elements with XRD, and then estimate the positions of light elements using computational techniques. This has obvious drawbacks when the elements of interest are the light ones, e.g. when studying hydrogen bonding.

While neutron diffraction can often provide this missing information, NMR is generally a much cheaper alternative as it does not require neutron sources. In addition, it can also be used to investigate samples for which large crystals cannot be grown.

An early experiment used to determine internuclear distances in solid-state NMR was the *Spin Echo DOuble Resonance* (SEDOR) [23] experiment developed by Hahn and coworkers in the mid to late 50's. The experiment measures the magnitude of the dipole-dipole coupling constant b_{ij}, and the distance is then calculated according to $r_{ij} = \sqrt[3]{\frac{\mu_0 \hbar \gamma_i \gamma_j}{4\pi b_{ij}}}$. Most modern spectrometers, however, use *Magic Angle Spinning* (MAS) in order to remove the chemical shift anisotropy (this will be discussed in more depth in Section 3.1), a process which also removes the dipole-dipole interaction. With an appropriate pulse sequence, however, it is possible to reintroduce parts of a Hamiltonian which would otherwise be averaged away by MAS; this is known as *recoupling*. For instance, Gullion and Schaefer's *Rotational Echo DOuble Resonance* (REDOR) [24] experiment recouples the dipole-dipole interaction between two spin $\frac{1}{2}$ nuclei at low MAS rates, allowing the distance between them to be determined once again. There have been a number of subsequent sequences

based on REDOR, for instance the *Transferred Echo DOuble Resonance* (TEDOR [25]) sequence which reduces background signals from uncoupled nuclei, or the *TRAnsfer of Population in DOuble Resonance* (TRAPDOR [26]) and *Rotational Echo Adiabatic Passage DOuble Resonance* (REAPDOR [27]) sequences which measure the distance between a quadrupolar nucleus and a spin $\frac{1}{2}$ nucleus. The sequence has even been combined with the *Multiple-Quantum Magic-Angle Spinning* (MQMAS [28]) experiment to measure ^{19}F–^{27}Al distances [29, 30].

The main drawback of the REDOR family of sequences is that they recouple both $\hat{\boldsymbol{I}}\hat{\boldsymbol{I}}$ and $\hat{\boldsymbol{I}}\hat{\boldsymbol{S}}$ dipole-dipole couplings. The deviation can usually be kept small as REDOR's $\hat{\boldsymbol{I}}\hat{\boldsymbol{I}}$ recoupling scales with $\frac{\mathrm{pw}_{180}}{\tau_{\mathrm{MAS}}}$ where pw_{180} is the width of an $\hat{\boldsymbol{I}}$-spin 180° pulse. For hydrogen, however, $\gamma_{^1\mathrm{H}}$ is so high that MAS rates above 30 kHz are generally required for efficient decoupling, making the measurement of hydrogen bonds difficult.

From previous analysis of REDOR using symmetry based arguments [31], Andreas Brinkmann and Arno Kentgens published the pulse sequences shown in Figure 2.1 [32] in 2006, and later the same year the sequence shown in Figure 2.2 [33]. The two experiments are hitherto unnamed, to the best of my knowledge, so they will be referred to simply as the non-selective and selective experiments, respectively.

The crucial part of both these experiments is the *R-sequence*. The theory behind R-sequences is the topic of Section 2.1.1, but the notation is explained here for the sake of completeness. An R-sequence is a sequence of pulses built on the so-called *basic element* ($\mathscr{R}$). $\mathscr{R}$ must be either a $180°_x$ pulse or an equivalent composite pulse, i.e. a series of pulses which inverts $\hat{I}_y$ & $\hat{I}_z$ while leaving $\hat{I}_x$ unchanged. The full sequence then consists of N successive basic elements with an alternating phase offset of $\phi = \frac{\nu}{N}\pi$. The pulse powers must be chosen such that the the entire sequence fits exactly into n rotor periods (τ_{MAS}). N, n and ν must be integers; N must also be even. R-sequences are usually written in the following form:

$$\mathrm{R}N_n^\nu := \overbrace{\underbrace{\mathscr{R}^{\phi}\mathscr{R}^{-\phi}\cdots\mathscr{R}^{-\phi}}_{N\mathscr{R}}}^{n\tau_{\mathrm{MAS}}} \tag{2.1.1}$$

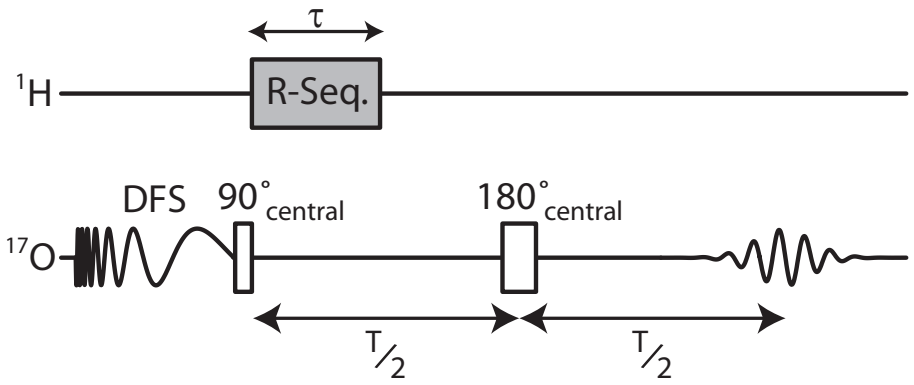


Figure 2.1: Non-selective pulse sequence.

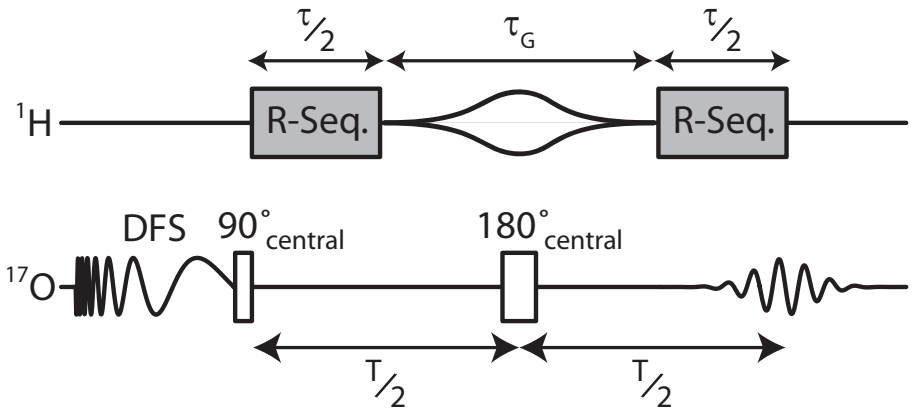


Figure 2.2: Selective pulse sequence.

As an example, let us assume $\mathscr{R}$ is a $180°_x$ pulse, then:

$$\mathrm{R}4_1^2 := 180°_{90°}180°_{-90°}180°_{90°}180°_{-90°} \tag{2.1.2}$$

where the 180° pulses are each exactly one quarter of a rotor period.

To complicate matters further, Brinkmann & Kentgens made extensive use of *supercycles* in their work. In a supercycled R-sequence, the R-sequence is repeated several times back-to-back, but in a modified form. For instance, an inversion supercycle consists of an R-sequence $\mathrm{R}N_n^\nu$, followed by the same R-sequence, but with all phases inverted. This supercycle is written as SS', or as $(\mathrm{R}N_n^\nu \mathrm{R}N_n^{-\nu})$ [1]. As

[1] Although it is not obvious from the notation, the phases of the basic element are inverted as well.

an example, the inversion supercycle for an $\mathsf{R}4_1^2$ would be:

$$\left(\mathsf{R}4_1^2\mathsf{R}4_1^{-2}\right) := 180°_{90°}180°_{-90°}180°_{90°}180°_{-90°}180°_{-90°}180°_{90°}180°_{-90°}180°_{90°} \tag{2.1.3}$$

The inversion supercycle can help reduce the effects of B_1 field inhomogeneity for some[2] R-sequences. Another, slightly broader class of supercycle are *phase-shift* supercycles, denoted by $S\mathrm{P}^{\mathrm{q}}$ or $(\mathsf{R}N_n^\nu)\,\mathrm{P}^{\mathrm{q}}$ where P & q are integers. In this case, the R-sequence is repeated P times, with a phase offset of $\frac{\mathrm{q}}{\mathrm{P}}360°$ relative to the previous sequence. Hence any $\mathsf{R}N_n^\nu$ sequence with an $S3^1$ supercycle would become:

$$(\mathsf{R}N_n^\nu)\,3^1 := [\mathsf{R}N_n^\nu]_{0°}\,[\mathsf{R}N_n^\nu]_{120°}\,[\mathsf{R}N_n^\nu]_{240°} \tag{2.1.4}$$

Phase shift supercycles with $\mathrm{q} = 1$ suppress all terms in the average Hamiltonian for which μ (see Section 2.1.1) is not an integer multiple of P.

Getting back to the experiments in Figures 2.1 & 2.2, the first increment of the non-selective experiment is essentially a spin-echo experiment on ^{17}O, but using weak pulses so as to only excite the central transition. The *Double Frequency Sweep* (DFS) [34] is used purely to enhance the signal-to-noise ratio; this is done by adiabatically inverting the satellite transitions with an amplitude modulated pulse, transferring population into the central portion of the density matrix. Once the spectrum has been recorded, the experiment is repeated with an appropriate R-sequence (see section 2.1.1) on the ^{1}H channel which recouples only the heteronuclear dipole-dipole coupling. As the recoupling time τ is increased, the ^{17}O signal evolves under the dipolar interaction of its nearest-neighbour proton; weak couplings are essentially swamped by strong couplings, and are not observed.

The selective experiment is similar, but uses a selective Gaussian pulse to invert one proton resonance in the spectrum, and a second R-sequence block is placed at the end of the experiment. Since the experiment is symmetrical, it acts as a spin-echo experiment and the effects of all dipole-dipole couplings are reversed, except for those involving the inverted proton. This allows us to measure the dipole-dipole coupling constant for any oxygen-proton pair, not just the nearest neighbours. However, the experiment suffers from a drop in signal-to-noise as

[2] This method does not work for γ-encoded sequences.

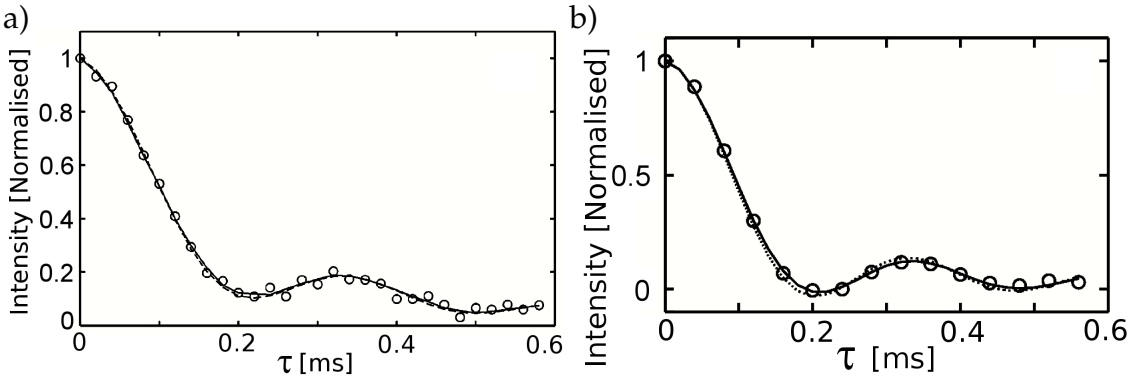


Figure 2.3: Figure a) shows a non-selective $\left(\mathrm{R12}_3^5\mathrm{R12}_3^{-5}\right)3^1$ experiment recorded, reproduced from page 16098 of reference [32]. Circles correspond to experimental data, and the black lines correspond to a two-spin (solid) and three-spin (dashed) AHT simulations with $b_{\mathrm{OH}''} = 14.27 \pm 0.38\,\mathrm{kHz}$. Figure b) shows a non-selective $\left(\mathrm{R4}_1^2\mathrm{R4}_1^{-2}\right)3^1$ (same markers, $b_{\mathrm{OH}''} = 14.77 \pm 0.26\,\mathrm{kHz}$), reproduced from page 14759 of reference [33]. Both sets of data were recorded by Brinkmann & Kentgens at 800 MHz for ^{1}H

the length of the Gaussian pulse τ_{G} must generally be several milliseconds long in order to avoid inverting multiple protons, leading to long echo times. The experiment also differs from the non-selective experiment in that the echo time T is not constant, but increases slightly as τ increases since $T = \tau_{\mathrm{G}} + \tau$.

Of course, the key to success in both cases lies in choosing good R-sequences. In their original paper, Brinkmann and Kentgens identify a set of sequences of the form $\mathrm{R4}n_n^{2n-1}$ with $n \geq 3$ which recouple ^{1}H heteronuclear dipole-dipole couplings and do not recouple any other interactions with the unavoidable exception of the ^{1}H CSA which has the same symmetry numbers as the heteronuclear dipole-dipole coupling[3]. While proton CSAs are generally on the order of 10 ppm, simulations indicate that the effects are negligible.

The results of two experiments by Brinkmann are shown in Figure 2.3. For the experiment in Figure 2.3a the supercycles simply help the long-term behaviour, but for Figure 2.3b they are vital as they remove a number of undesired interactions which would otherwise be recoupled by the $\mathrm{R4}_1^2$ sequence. The problem with supercycled sequences is that they tend to be extremely long. Assuming an MAS rate

[3]In principle, this could be avoided by a 2-channel R-sequence on ^{1}H & ^{17}O. However, this would likely require hard pulses on ^{17}O due to the high MAS rate.

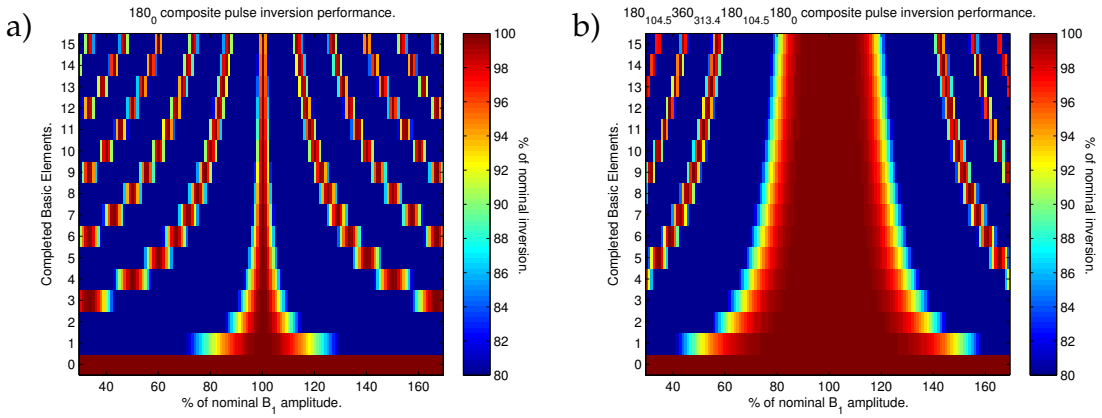


Figure 2.4: B_1 inhomogeneity behaviour of a $180°_x$ pulse (a) and the BB1(180) [35] composite pulse (b) when repeated back-to-back.

of $\nu_{MAS} = 50\,kHz$, the supercycle for the $\left(R12_3^5R12_3^{-5}\right)3^1$ sequence is 360 µs long. The $\left(R4_1^2R4_1^{-2}\right)3^1$ is shorter, but still 120 µs in length. This is clearly unacceptable as most of the signal has decayed away by 100 µs to 200 µs. Brinkmann and Kentgens get around this by sampling before the cycle is complete; the experiment in Figure 2.3a is sampled so frequently that even the R-sequences are not always complete.

Such practices are not uncommon in NMR, but they ignore a fundamental problem: the average Hamiltonian used for the R-sequence framework is only valid for complete sequences. Sampling early can cause artefacts in the signal due to unwanted interactions which have been incompletely averaged. While these artefacts are often sufficiently minor as to be ignored overall, oversampling should generally be treated with caution. On the other hand, supercycles are known to improve the robustness of R-sequences, e.g. with respect to B_1 field inhomogeneity or B_1 offset. The question therefore arises: is it possible to find unexplored R-sequences which are sufficiently short that oversampling is not required to capture the short term behaviour of a strong O–H coupling, yet also sufficiently robust that they may be used without supercycling?

In this work, the question will be addressed in part by exploring R-sequences not included in the $R4n_n^{2n-1}$ set, as well as through the use of composite pulses. Composite pulses are frequently designed with a specific aim in mind, such as com-

pensation for resonance offset effects (e.g. the BB1(180) [35]) or phase distortions (e.g. the $180°_{120°}180°_{240°}180°_{120°}$ [36]). As shown in Figure 2.4, however, some also provide a degree of robustness against B_1 inhomogeneity. In Figure 2.4a, a single $180°_x$ can achieve good inversion of $\hat{I}_z$ in spite of a 10 % offset, but the "useful zone" shrinks rapidly as more pulses are applied. In contrast, 15 back-to-back BB1(180) sets still achieve good inversion across a range of about 20 %.

Finally, there is another benefit to exploring new R-sequences which relates to the analysis. As shown in Figure 2.3, the dipole-dipole coupling is determined by fitting simulations to the experimental data. Generally, the sequences in the $\mathrm{R}4n_n^{2n-1}$ set produce simple, unstructured curves resembling exponential decays with weak oscillations. In contrast, some R-sequences possess strong oscillations, or even zero-crossing; these are expected to make the fitting "harder", resulting in narrower confidence limits if the simulations and experimental data match well.

2.1.1 Symmetry Based Recoupling

As will be discussed in Section 3.1, most NMR interactions are averaged to an isotropic value by MAS. However, it is possible to interfere with this averaging by applying RF pulses which are synchronised with the MAS rate. A theoretical framework for this comes from combining AHT and spherical tensor analysis[4] and is known as *symmetry based recoupling* [31,37,38].

There are many NMR experiments which conform to symmetry based recoupling, though many of them were discovered independently of the theory and have only been recognized with hindsight. One such example is the REDOR sequence, which has already been mentioned earlier. There are several similar REDOR schemes, but they generally conform to an $\mathrm{R}2n_n^{\nu}$ set with an SS' supercycle and a basic element $\mathscr{R}$ containing one 180° pulse and a large gap. Another example is the *Radio-*

[4]The mathematical analysis used in reference [37] differs from that of section 1.1.4, most notably in that section 1.1.4 treats the spin operators $\hat{\boldsymbol{I}}$ and $\hat{\boldsymbol{S}}$ on equal footing with the magnetic field vector $\boldsymbol{B}_0$ even though $\hat{\boldsymbol{I}}$ and $\hat{\boldsymbol{S}}$ can be rotated by RF pulses while $\boldsymbol{B}_0$ cannot. There are also differences in how the terms of the Hamiltonian combine: a spherical tensor of rank i will have $2i+1$ different $M_{i-j}\hat{\boldsymbol{T}}_{ij}$ terms, while there will be $(2l+1)\times(2\lambda+1)$ different $\mathscr{H}_{lm\lambda\mu}$ terms.

Interaction Λ	Space Rank l	Spin Rank λ	Field Rank
Isotropic Chemical Shift	0	1	1
Chemical Shift Anisotropy	2	1	1
J-Coupling	0	0	0
Dipole–Dipole Coupling	2	2	0
$\hat{\boldsymbol{I}}\hat{\boldsymbol{S}}$ Dipole–Dipole Coupling	2	1	0

Table 2.1: Interaction ranks

Frequency Driven Recoupling (RDFR [39, 40]) scheme by Griffin and co-workers, a heteronuclear dipole-dipole recoupling sequence with $\mathsf{R}4_4^1$ symmetry and an SS' supercycle which can, for instance, be used for a solid state equivalent to the ubiquitous *Nuclear Overhauser Effect SpectroscopY* (NOESY [41]). Lastely, one might also mention the *ThrOugh-Bond SpectroscopY* (TOBSY [42]) experiment which uses an $\mathsf{R}4_6^0$ sequence to perform correlation spectroscopy via J-couplings, just as with the solution state *Total Correlation SpectroscopY* (TOCSY [43]) experiment.

There are two main frameworks for symmetry based recoupling: *C-sequences* and *R-sequences*. However, only R-sequences are of relevance to the current work. The effect of an R-sequence on a given interaction Hamiltonian $\hat{\mathscr{H}}^{\Lambda}$ is highly dependent on the interaction's symmetry properties. Mathematically, $\hat{\mathscr{H}}^{\Lambda}$ is broken down into components $\hat{\mathscr{H}}^{\Lambda}_{lm\lambda\mu}$ (see Equations 2.1.5 and 2.1.6) where l is known as the *space rank*, $m \in \{-l, -l+1, \ldots, l-1, l\}$ labels the *space components*, λ is the *spin rank* and $\mu \in \{-\lambda, -\lambda+1, \ldots, \lambda-1, \lambda\}$ labels the *spin components* (l & λ for various interactions are given in Table 2.1).

$$\overline{\hat{\mathscr{H}}}\left(t^0\right) = \sum_{lm\lambda\mu} \overline{\hat{\mathscr{H}}}^{(1)}_{lm\lambda\mu}\left(t^0\right) \tag{2.1.5}$$

$$\overline{\hat{\mathscr{H}}}^{(1)}_{lm\lambda\mu}\left(t^0\right) = \frac{1}{\tau_{\mathrm{MAS}}} \int_{t^0}^{t^0+\tau_{\mathrm{MAS}}} \hat{\mathscr{H}}_{lm\lambda\mu}\left(t\right) \mathrm{d}t \tag{2.1.6}$$

The final results of all this are the following selection rules:

$$\overline{\hat{\mathscr{H}}}^{(1)}_{lm\lambda\mu} = 0 \quad \text{if} \quad mn - \mu\nu \neq \frac{N}{2} Z_{\lambda} \tag{2.1.7}$$

$$\overline{\hat{\mathscr{H}}}^{(1)}_{20\lambda\mu} = 0 \quad \text{for exact MAS.} \tag{2.1.8}$$

where Z_λ is any integer with the same parity as λ. Equation 2.1.7 is quite simple, but there are many values of m & μ for most Hamiltonians, so it is useful to display the results in so-called *spin-space selection diagrams* (see Figures 2.5 & 2.6 on page 27). In this work, blue lines correspond to mn, red lines to $mn - \mu\nu$ and the black lines block off all $\hat{\mathscr{H}}_{lm\lambda\mu}$ which match Equation 2.1.7 ($m < 0$ is ignored since the diagram is symmetric).

For instance, Figure 2.5 clearly shows that the $\mathsf{R}4_1^2$ used in reference [33] recouples most NMR interactions to a greater or lesser degree, meaning that it is only viable if it is applied in conjunction with a supercycle, e.g. the $S3^1$ which will remove all interactions with $\mu \neq 0$. On the other hand, the $\mathsf{R}10_3^1$ in Figure 2.6 only recouples the chemical shift anisotropy and the heteronuclear dipole-dipole interaction via $\hat{\mathscr{H}}_{2211}$ (and $\hat{\mathscr{H}}_{2-21-1}$), so it can be used "as is".

As an aside, the $\mathsf{R}10_3^1$ is a so-called γ-*encoded* sequence, meaning that each value of μ in the recoupled terms is associated with a unique m value and vice versa ($\{m = 2, \mu = 1\}$ and $\{m = -2, \mu = -1\}$). Conversely, the heteronuclear dipole-dipole coupling for the $\mathsf{R}4_1^2$ is not γ-encoded as it recouples on $\{m = 2, \mu = 0\}$ and $\{m = -2, \mu = 0\}$; this means that $\mu = 0$ is associated with both $m = 1$ and $m = -1$. γ-encoded sequences tend to recouple more evenly across different crystalite orientations, meaning that a larger fraction of the sample is actively contributing to the signal, but they also tend to suffer more from RF imperfections [37].

2.2 Method

2.2.1 R-Sequence Selection

In principle there are an infinite number of R-sequences, however the required criteria for our experiments reduce this to a testable quantity (see Appendix B.1 for an example program). Firstly, any given probe will have a maximum possible MAS rate $\nu_{\mathrm{MAS\,max}}$ ($\approx 60\,\mathrm{kHz}$), a minimum MAS rate we are prepared to spin at ($\nu_{\mathrm{MAS\,min}} \approx 30\,\mathrm{kHz}$) and a shortest possible 180° pulse length ($pw_{180^\circ\,\mathrm{min}} = 2\,\mu\mathrm{s}$). Secondly, the interaction we wish to measure occurs on a time scale of about 100 µs,

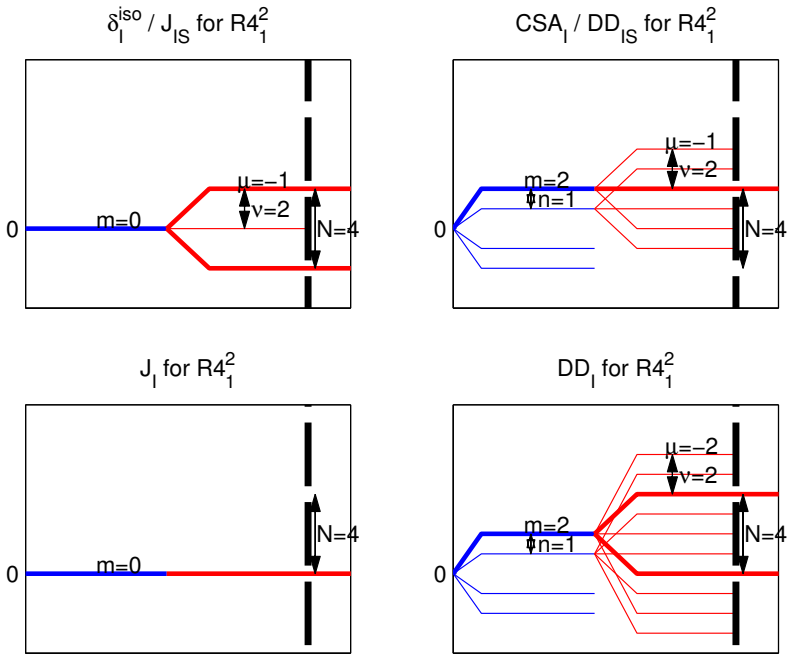


Figure 2.5: Spin-space selection diagrams for $R4_1^2$. Horizontal lines represent terms on the left hand side of Equation 2.1.7 for all $\overline{\hat{\mathscr{H}}}^{(1)}_{lm\lambda\mu}$ with $m \geq 0$ (mn is blue, $\mu\nu$ is red), and vertical ones represent the right hand side. Lines for $\overline{\hat{\mathscr{H}}}^{(1)}_{lm\lambda\mu}$ which are recoupled are drawn in bold. Interactions with identical space rank and spin rank (e.g. the isotropic chemical shift and the heteronuclear J-coupling) have been grouped in the same diagram.

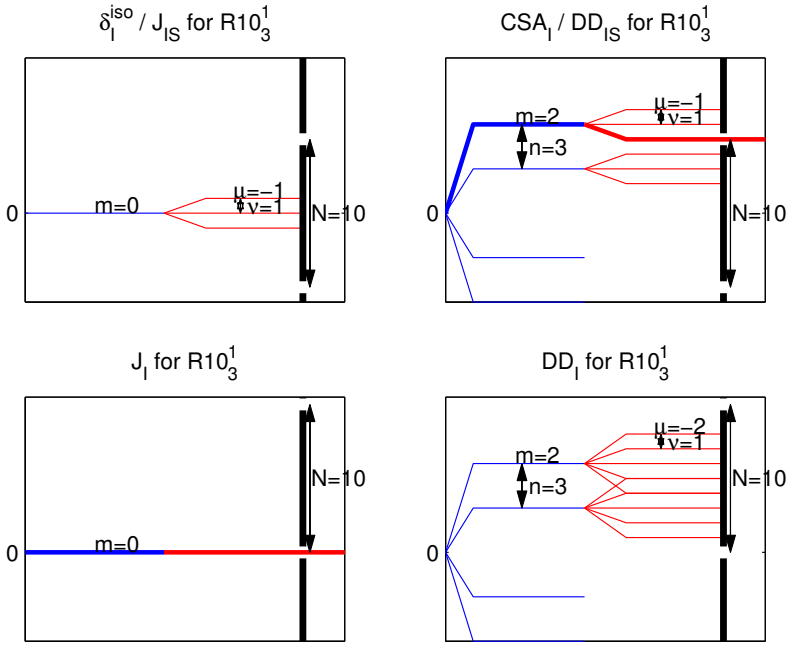


Figure 2.6: Spin-space selection diagrams for $R10_3^1$. See Figure 2.5 for details.

	$R4_1^0$	$R14_2^3$	$R14_2^{11}$	$R4_3^0$	
$R10_1^3$	$R10_1^7$	$R18_2^5$	$R18_2^{13}$	$R10_3^1$	$R10_3^9$
$R12_1^4$	$R12_1^8$	$R22_2^7$	$R22_2^{15}$	$R14_3^1$	$R14_3^{13}$
$R14_1^5$	$R14_1^9$	$R26_2^9$	$R26_2^{17}$	$R16_3^2$	$R16_3^{14}$
$R16_1^6$	$R16_1^{10}$	$R30_2^{11}$	$R30_2^{19}$	$R20_3^4$	$R20_3^{16}$
$R18_1^7$	$R18_1^{11}$	$R34_2^{13}$	$R34_2^{21}$	$R22_3^5$	$R22_3^{17}$
$R20_1^8$	$R20_1^{12}$	$R38_2^{15}$	$R38_2^{23}$	$R26_3^7$	$R26_3^{19}$

Table 2.2: Selected R-sequences

Name	Composite Pulse	Reference
	$180°_{0°}$	
	$90°_{90°}$ $180°_{0°}$ $90°_{90°}$	[44]
	$90°_{0°}$ $360°_{120°}$ $90°_{0°}$	[45]
	$180°_{120°}$ $180°_{240°}$ $180°_{120°}$	[36]
BB1(180)	$180°_{104.5°}360°_{313.4°}180°_{104.5°}180°_{0°}$	[35]
F1	$180°_{46.6°}$ $180°_{255.5°}180°_{0°}$ $180°_{104.5°}180°_{313.4°}$	[46]
G1	$180°_{45°}$ $180°_{270°}$ $180°_{0°}$ $180°_{90°}$ $180°_{315°}$	[46]

Table 2.3: Composite pulses.

so our sequence should increase in steps of $\Delta\tau \leq 100\,\mu s$. Therefore, we know that $\frac{n}{\nu_{MAS\,max}} \leq \Delta\tau$, that $Npw_{180°\,min} \leq \frac{n}{\nu_{MAS\,min}}$ and that $Npw_{180°\,min} \leq \Delta\tau$. And finally, sequences with $\phi \geq \frac{\pi}{2}$ are equivalent to sequences with $\phi' = \pi - \phi$ for $\mathscr{R}$=$180°_{0°}$, so $\nu \leq \frac{N}{2}$. Some sequences which fit these criteria are shown in Table 2.2.

Valid pulse sequences were then simulated in SIMPSON [47] using $\mathscr{R}$=$180°_{0°}$, as well as any of the other composite pulses shown in Table 2.3 which could be run at MAS rates of 20 kHz or above. An example program for the BB1(180) composite pulse in a non-selective experiment using typical parameters is given in Appendix A.1.

Initially, the entire FID was simulated for each experimental point and stored as a 2-dimensional data set; this was found to be unnecessary, however, as the peak shape and phase do not change. Instead, only the first point of each FID needs to be calculated and recorded. Since the intensity of the FID is proportional to

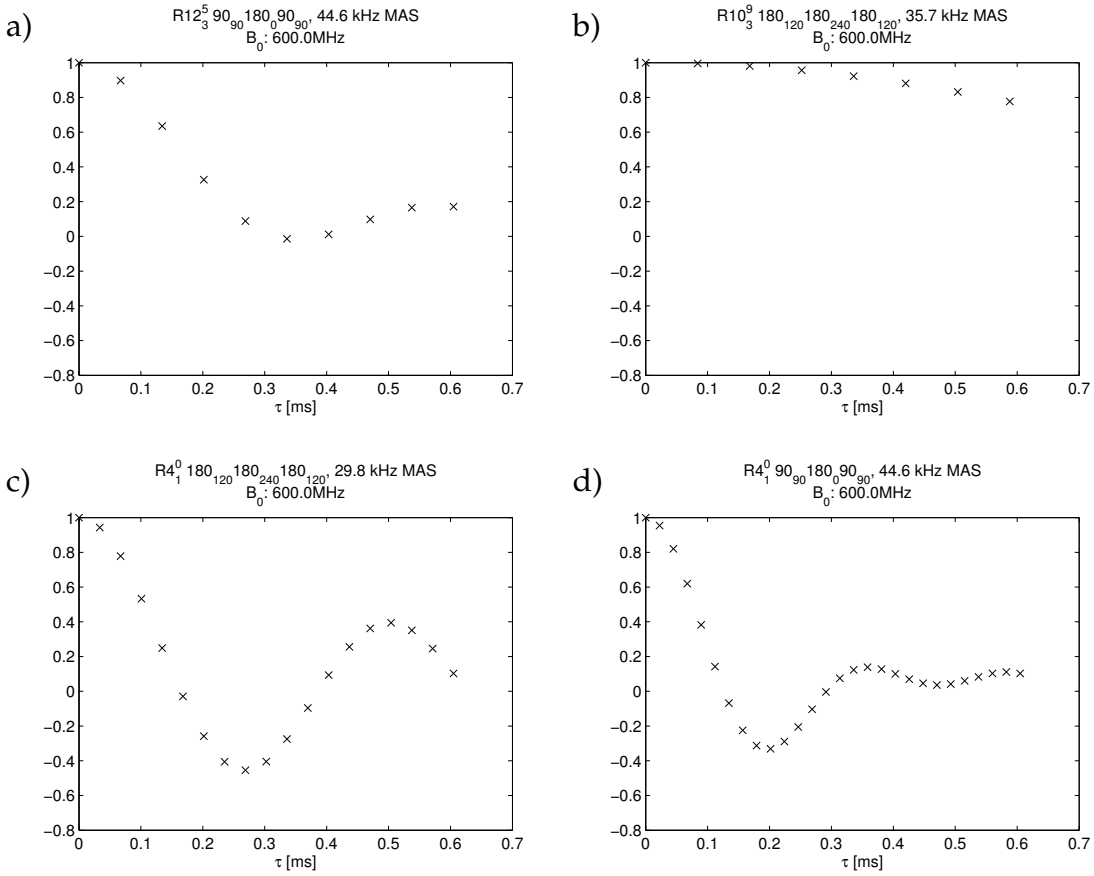


Figure 2.7: Simulated R-sequences with composite pulses for non-selective experiments at 600 MHz and maximum permissible MAS rates (assuming a proton 180° pulse length limit of 2.8 µs). Figure a) shows a viable sequence: the $R12^5_3$ with a $90°_{90°}180°_{0°}90°_{90°}$ basic element. Figure b) shows an $R10^9_3$ with a $180°_{120°}180°_{240°}180°_{120°}$ basic element; this sequence only recouples weakly and is hence not viable. Figure c) shows an $R4^0_1$ with a $180°_{120°}180°_{240°}180°_{120°}$ basic element; the sequence is viable, and has pronounced oscillations which may help constrain the fit when analysing experimental data. Figure d) shows an $R4^0_1$ with a $90°_{90°}180°_{0°}90°_{90°}$ basic element; the sequence is viable and recouples more strongly than the sequence in Figure b), but the oscillations are not as pronounced.

the intensity of the integrated spectrum, the simulated data can be used directly without the need for Fourier transforms or integration.

Non-selective experiments were assessed assuming a dipole-dipole coupling constant of about 14 kHz, selective experiments at about 3 kHz. Some non-selective examples are shown in Figure 2.7. Figure 2.7a shows an $\mathsf{R12}_3^5$ (coincidentally one of Brinkmann's sequences) with a basic element $\mathscr{R}$=$90°_{90°}180°_{0°}90°_{90°}$. In this particular case, the composite pulse has not altered the behaviour much and the graph strongly resembles Figure 2.3a on page 22. As such, it makes a good example for a viable experiment. Figure 2.7b, on the other hand, shows a common type of aberrant behaviour. Even though the $\mathsf{R10}_3^9$ has all the desired recoupling properties, the combination of $\mathsf{R10}_3^9$ and $\mathscr{R}$=$180°_{120°}180°_{240°}180°_{120°}$ has a low scaling factor, making the experiment useless. On the other hand, Figures 2.7c and d look promising. Figure 2.7c displays much stronger oscillations than the $\mathsf{R12}_3^5$ sequence; more structured signals should, in principle, provide extra constraints for the fitting process used to analyse experimental data. Figure 2.7d has oscillations intermediate between a) and c), but also has a stronger scaling factor.

As an aside, the crystal file used for the powder averaging in the simulations contained 678 orientations of α & β and was created using the REPULSION method [48]; this was found to be converged according to Figure 2.8a. Convergence was also checked with respect to the timestep Δt from Equations 1.1.23 & 1.1.26; the calculations were found to be reasonably converged at $\Delta t = 1\,\mu s$ (see Figure 2.8b). In addition, Figure 2.8c demonstrates that the ^{1}H CSA has little to no effect on the simulated results and Figure 2.8d demonstrates that a selective experiment measuring a weak O–H dipole-dipole coupling is unperturbed by the introduction of a 3rd spin with a strong O–H dipole-dipole coupling. This is convenient not only because 2-spin calculations are cheaper than 3-spin calculations, but also because it means that weak couplings can be determined independently of strong couplings.

Unfortunately, however, it was found that B_1 field inhomogeneity can have a very strong effect on these sequences. B_1 field inhomogeneity can be complicated to simulate, so the behaviour has been approximated by rerunning simulations multiple times at the same MAS rate and with the same pulse lengths, but with incor-

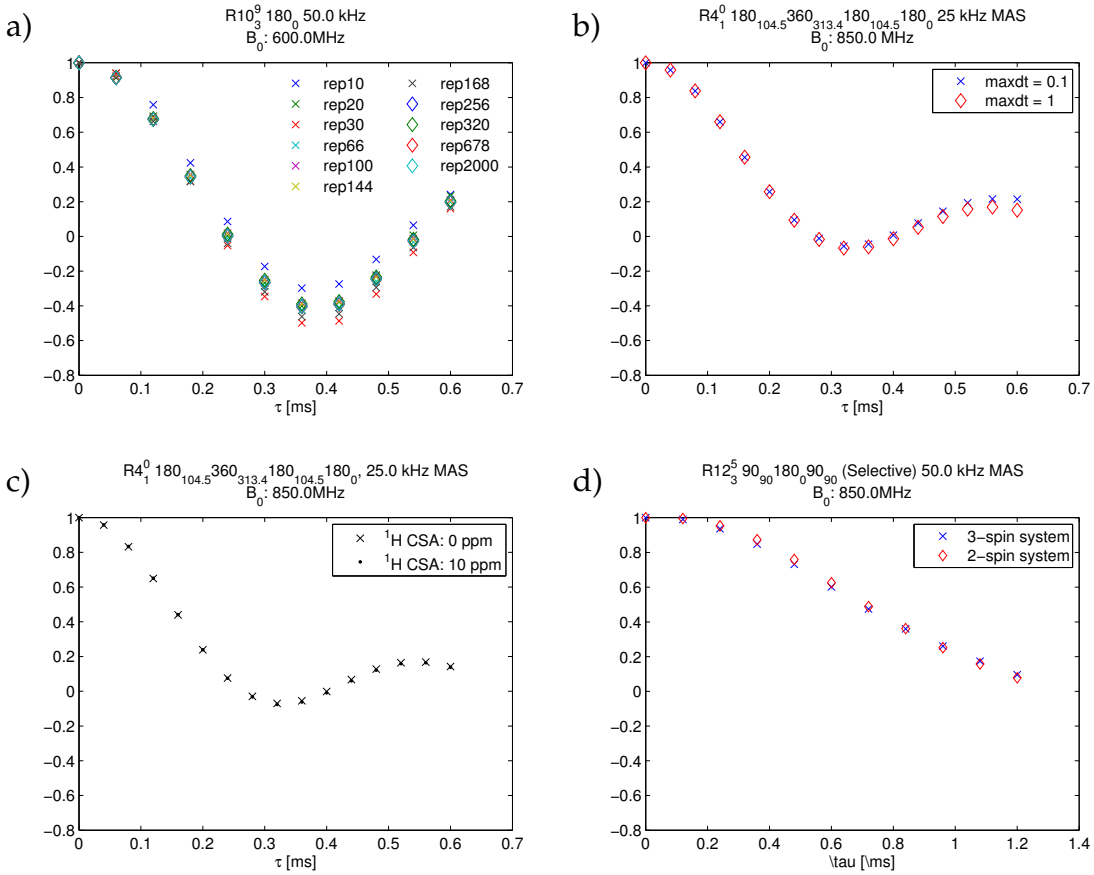


Figure 2.8: Convergence checks for SIMPSON simulations. Figure a) shows simulations of an R10$_3^9$ sequence with crystal files containing 10 to 2000 α & β orientations. The simulations are considered converged above 100 orientations. Figure b) illustrates the accuracy of the simulations w.r.t. the simulation timesteps Δt. The red diamonds show an R4$_1^0$ sequence with a BB1(180) composite pulses calculated with $\Delta t = 1\,\mu s$ (used in most of the simulations in this work) and the blue crosses mark the same simulation, but with $\Delta t = 0.1\,\mu s$. Figure c) shows that the results of a simulation without a proton CSA (blue crosses) are identical to those from a simulation using a typical proton CSA value (10 ppm, blue dots), and Figure d) shows that the results for a selective experiment using a 2-spin system (red diamonds) are very similar to the more expensive 3-spin calculations (blue crosses).

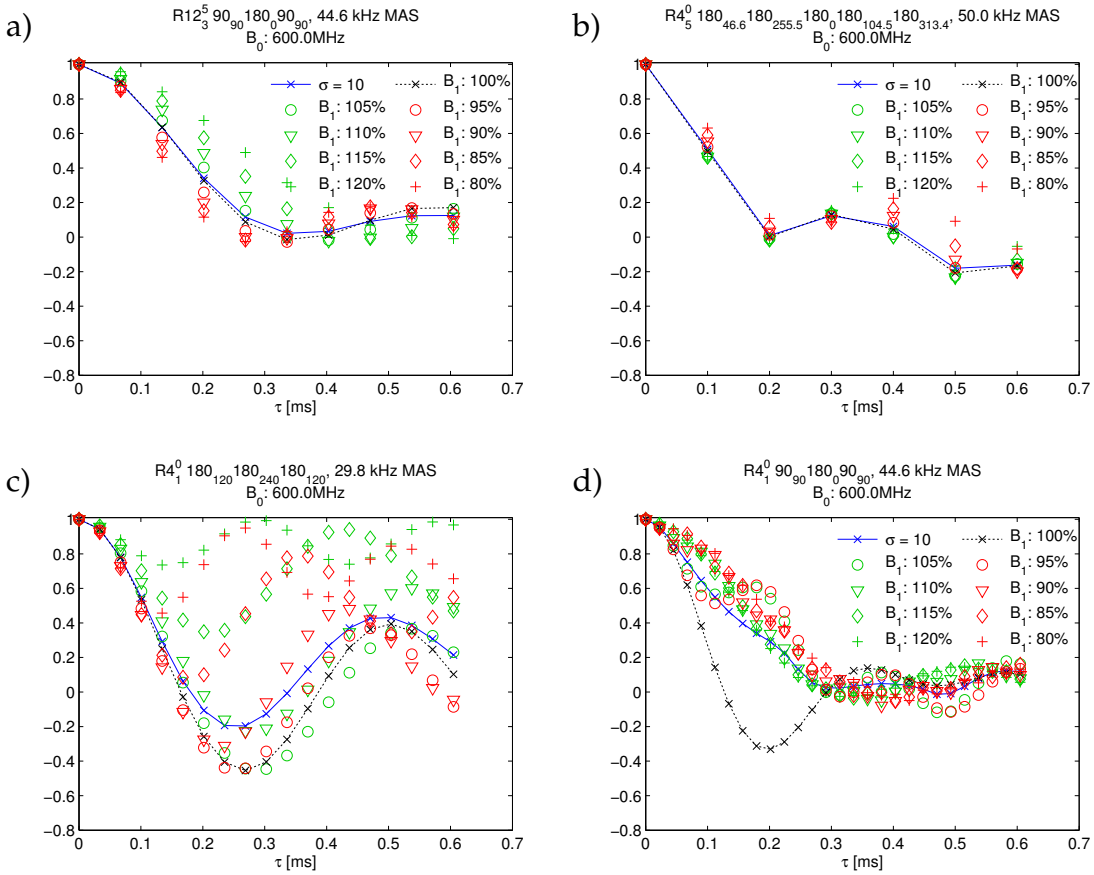


Figure 2.9: B_1 field inhomogeneity behaviour for an $R12_3^5$ with a $90°_{90°}180°_{0°}90°_{90°}$ basic element (a), an $R4_5^0$ with an F1 basic element (b), an $R4_1^0$ with a $180°_{120°}180°_{240°}180°_{120°}$ basic element (c) and an $R4_1^0$ with a $90°_{90°}180°_{0°}90°_{90°}$ basic element (d). All four simulations were run assuming a 600 MHz spectrometer and at their maximum MAS rates assuming a 180° pulse length limit of 2.8 µs and a maximum possible MAS rate of 50 kHz. The simulations with the nominal B_1 strength are marked as black crosses with black dotted lines, simulations with incorrect B_1 nutation rates are shown in green (high power) and red (low power). Simulations are labeled in percent of the nominal nutation rate. The blue crosses and blue line represent the weighted average of all simulations assuming a B_1 field inhomogeneity profile width of $\sigma = 10\,\%$. See Appendix A.1 for SIMPSON program and typical parameters.

rect ^{1}H nutation rates. In Figure 2.9, the black crosses represent a simulation with the nominal nutation rate for that sequence (labelled as "B_1: 100 %"). Simulations with low nutation rates are shown in red, and those with high nutation rates are shown in green. The blue crosses are an estimation of the overall results for the sequence; they are the Gaussian weighted average of the simulations present with weight $w = e^{\frac{(B_1-100)^2}{2\sigma^2}}$, where B_1 is the ^{1}H nutation rate in percent and σ represents the severity of the B_1 inhomogeneity for a given probe (assumed to be 10 % here).

Since the behaviour with respect to inhomogeneity was found to be different for different sequences, it was simulated for each sequence and judged by eye. Figure 2.9a shows an example of a sequence which would be accepted; the inhomogeneity causes significant deviation of the signal, but the distribution is somewhat symmetric and cancellation of errors causes the average behaviour to lie close to the nominal simulation. Figure 2.9b would also be accepted as the combination of $R4_5^0$ with an F1 composite pulse appears to be well compensated with respect to B_1 inhomogeneity. Figures c & d (incidentally the same sequences as shown in 2.7c & d) would be rejected. Figure 2.9c shows far to much deviation (especially considering that the Gaussian average is only an assumption) and the simulations in Figure 2.9d do not converge to the nominal shape.

Finally, it is worth remarking that a number of the R-sequences tested use $\nu = 0$. RN_n^0 sequences have a phase offset $\phi = 0$, meaning that any RN_n^0 with $\mathscr{R}$=180°$_{0°}$ is indistinguishable from continuous wave decoupling. Furthermore, they would be expected to suffer greatly under B_1 inhomogeneity as a slight change in B_1 strength could, for instance, turn an $R10_3^0$ into an $R4_1^0$, a sequence with completely different recoupling properties. If a composite pulse is used, however, both these ambiguities vanish. As demonstrated in Figure 2.9b, $\nu = 0$ sequences can be used for recoupling as long as an appropriate basic element $\mathscr{R}$ is chosen.

2.2.2 Processing and Analysis

All experiments were performed either on the 20 T wide bore magnet (850 MHz, Bruker Avance II spectrometer) at the "UK 850 MHz Solid-State NMR Facility",

or on the University of Nottingham's 14 T narrow bore magnet (600 MHz, Bruker Avance III spectrometer) in the "Centre for Biomolecular Sciences". The MAS probes used on both instruments could spin rotors at a maximum speed of 66 kHz, and could sustain ^{1}H nutation rates of 250 kHz and 180 kHz respectively. The experiments were generally recorded with relaxation delays of 2 s as the proton T_1 times of the tyrosine sample used were $\sim$ 0.5 s.

Non-selective experiments were recorded with echo times $T \sim 1.2$ ms, selective experiments required longer times due to τ_G, the length of the Gaussian inversion pulse. τ_G was $\sim$ 1 ms for experiments performed at 850 MHz, but was increased to $\sim$ 2 ms at 600 MHz due to the decreased separation between the H$''$ & H$^{\eta}$ peaks. Longer times would have been preferable, but the loss of ^{17}O signal from the increased echo delay was too great. Experiments at 850 MHz were generally recorded with 1000 to 4000 scans, non-selective experiments at 600 MHz required $\sim$ 10 000 as the quadrupolar coupling leads to broader ^{17}O peaks, and the selective experiments were recorded with 20 000 to 50 000 scans due to the loss of signal from the long Gaussian pulse. The DFS parameters used at 850 MHz were calibrated with the help of the facility manager Dr. D. Iuga, with good performance found for a DFS sweep length of 1 ms, a start frequency of 800 kHz and a final frequency of 100 kHz. At 600 MHz, the optimal parameters were obtained with a sweep length of 0.5 ms to 0.8 ms, a start frequency of 500 kHz and a final frequency of 100 kHz. On both spectrometers, the powers of the selective ^{17}O pulses were calibrated separately to produce a 10 µs 90° & 20 µs 180° pulse. Results were recorded as pseudo-2D data (see Figure 2.10) and then analysed according to two different methods.

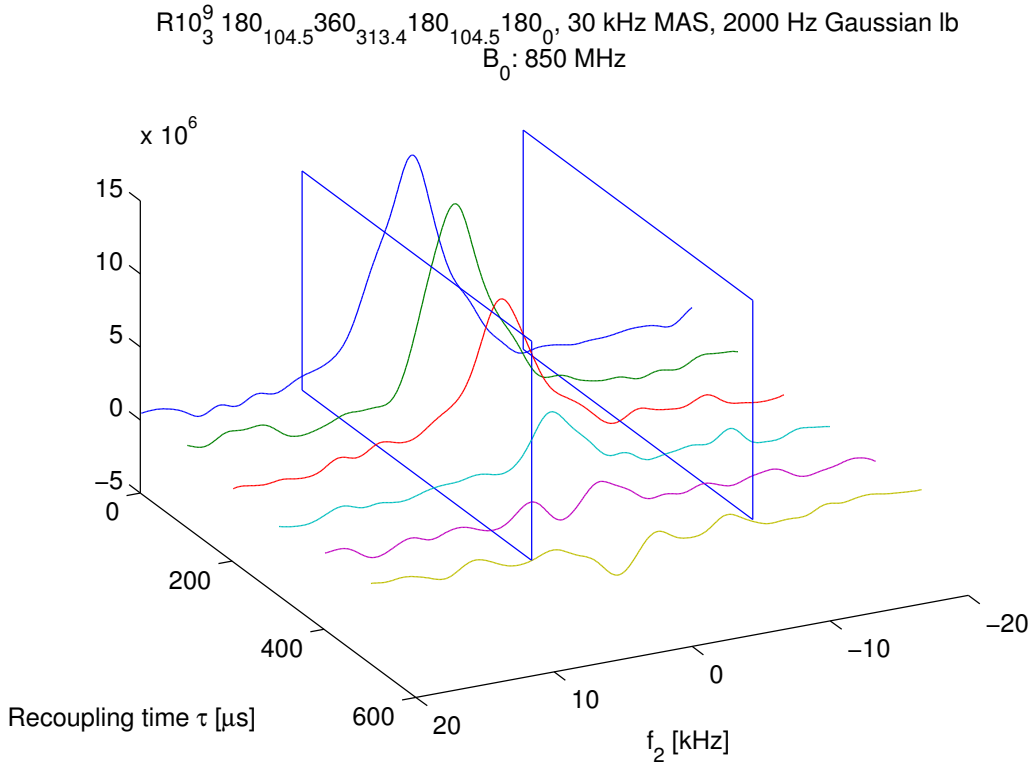


Figure 2.10: Central region of an $R10_3^9$ with BB1(180) spectrum (Fourier transformed along the direct dimension), plotted against the frequency of the direct dimension, f_2, and the recoupling time τ. The blue panels either side of the ^{17}O peak indicate the region which will be integrated in f_2.

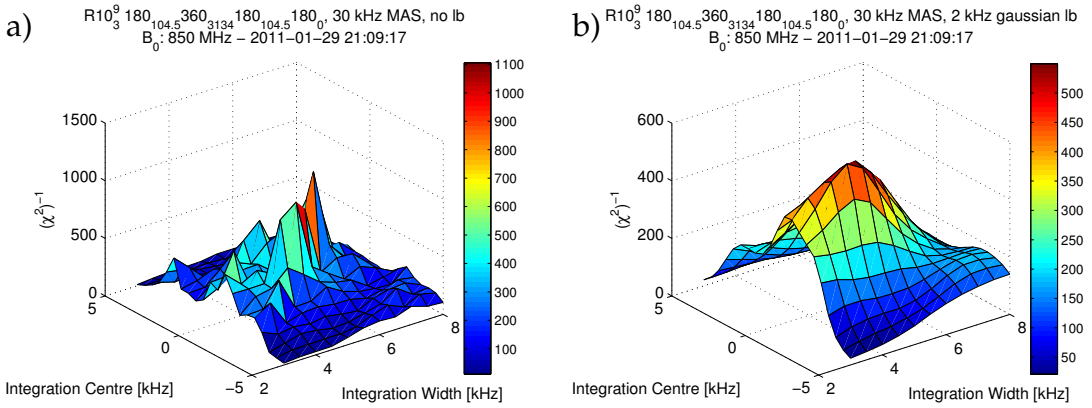


Figure 2.11: Inverse of χ^2_{min} obtained when fitting simulations to the integrated data from Figure 2.10. When no line broadening is used (Figure a), χ^2_{min} varies wildly if the integration region is changed. However, χ^2_{min} varies much more smoothly with changes to the integration width & offset when 2 kHz of Gaussian line broadening is used (Figure b).

2.2.3 Method I

The spectra were phased and then integrated within $\pm$ 8 kHz of the ^{17}O central peak. The integrals (I) were then normalised relative to the intensity at $\tau = 0$.

$$E = \sqrt{n_{\text{I}}} * \sigma_{\text{base}} \tag{2.2.1}$$

$$I_{\text{norm}} = \frac{I}{I_{(\tau=0)}} \tag{2.2.2}$$

$$E_{\text{norm}} = \left| \sqrt{\left(\frac{E}{I_{(\tau=0)}}\right)^2 + \left(\frac{I * E_{(\tau=0)}}{I^2_{(\tau=0)}}\right)^2} \right| \tag{2.2.3}$$

$$= \left| I_{\text{norm}} \sqrt{\left(\frac{E}{I}\right)^2 + \left(\frac{E_{(\tau=0)}}{I_{(\tau=0)}}\right)^2} \right| \tag{2.2.4}$$

where E is the error, n_{I} is the number of points in the integrated region and σ_{base} is the standard deviation of the baseline.

In general, the integration range was found to have an effect on the final results. The plots in Figure 2.11 show how the χ^2 value of the end result varies with the integration range. Line broadening was found to improve the reliability (Fig-

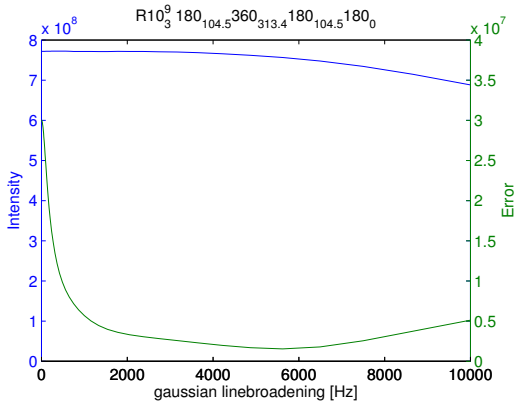


Figure 2.12: Effect of line broadening on signal-to-noise. The blue line (left axis) shows the intensity of the integrated peak at $\tau = 0$ ms as a function of the Gaussian line broadening width. The green line (right axis) shows the error associated with the integrated peak as determined from the baseline of the spectrum. Optimum signal-to-noise is found between 2 kHz and 6 kHz.

ure 2.11b is much smoother than Figure 2.11a), although it did not remove the dependence entirely. 2 kHz of Gaussian line broadening was found to give a good signal-to-noise ratio (see Figure 2.12).

In keeping with the analysis by Brinkmann & Kentgens, the integrated and normalised data I_{norm} were then fitted to a series of damped, normalised simulations ($S_{\text{norm}}(b_{\text{OH}})$) according to

$$D = fe^{\frac{\tau}{T_R}} S_{\text{norm}}(b_{\text{OH}}) + (1-f)e^{\frac{\tau}{T_Q}} \tag{2.2.5}$$

where f, T_R & T_Q are fitting parameters and b_{OH} is the hydrogen–oxygen dipole-dipole coupling constant of the simulation S. Figure 2.13 shows one such fit for an R10$_3^9$ experiment with a BB1(180) composite pulse. Experimental points are shown in red, the black diamonds represent the undamped simulation and the blue dots correspond to the fitted simulation. The dotted lines show where a simulated point of ± 1 would end up after damping, and can be therefore used to gauge the "severity" of the damping. χ^2 was then calculated using

$$\chi^2 = \sum (I_{\text{norm}} - D)^2 \tag{2.2.6}$$

The simulations are repeated for a range of dipole-dipole couplings, and χ^2 (b_{OH}) is calculated and graphed (see Figure 2.14). The most likely value for the coupling

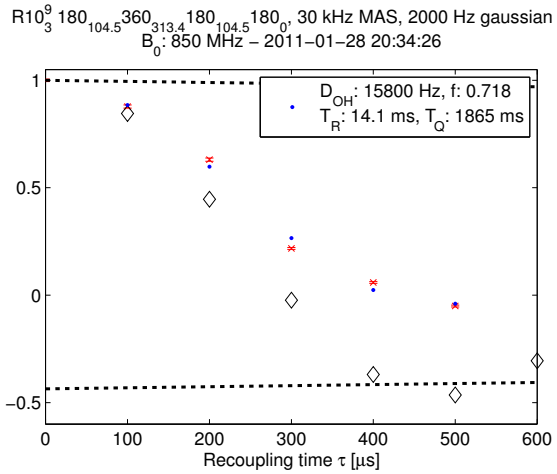


Figure 2.13: The integrated data of the R10$^9_3$ ($\mathscr{R}$=BB1(180)) from Figure 2.10 are plotted as a function of the recoupling time τ (red crosses). The best fit simulation is also shown (black diamonds: $b_{\mathrm{OH}} = 15.8\,\mathrm{kHz}$), as well as the same simulation when damping parameters are applied (blue dots: $f = 0.72$, $T_R = 14\,\mathrm{ms}$ & $T_Q = 1.8\,\mathrm{s}$). The black dotted lines indicate the maximum and minimum boundaries within which the simulation could vary: $(1-f)e^{\frac{\tau}{T_Q}} \pm fe^{\frac{\tau}{T_R}}$.

constant of the actual sample can be determined from the minimum value of the graph, and the confidence limits can be established.

According to Brinkmann [32], the confidence limits of a simulation are found at

$$\chi^2 = \chi^2_{\mathrm{min}} \left(1 + \frac{F^{\alpha}_{1,n_{\mathrm{I}}-4}}{n_{\mathrm{I}}-4} \right) \tag{2.2.7}$$

where α is the *significance level*[5] and F is the inverse of the cumulative F-distribution[6] (see Figure 2.14a). According to Ratkowsky [49], on the other hand, the confidence limits for a simulation should be found at

$$\chi^2 = \chi^2_{\mathrm{min}} \left(1 + \frac{n_{\mathrm{p}} F^{\alpha}_{n_{\mathrm{p}},n_{\mathrm{I}}-n_{\mathrm{p}}}}{n_{\mathrm{I}}-n_{\mathrm{p}}} \right) \tag{2.2.8}$$

where n_{p} is the number of parameters (see Figure 2.14b). Equation 2.2.8 resulted in a higher offset and hence broader confidence limits in all cases tested here. Ratkowsky's method also tended to match the confidence limits from Method II

[5]For the 95 % confidence limit, for instance, $\alpha = 1 - 0.95 = 0.05$.

[6]Also known as the Fisher-Snedecor distribution.

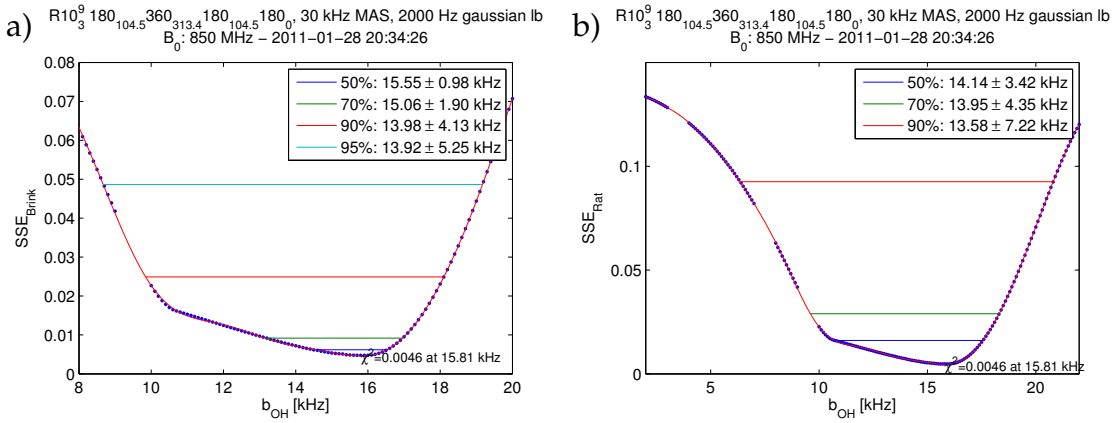


Figure 2.14: χ^2 as a function of b_{OH} (blue dots, interpolated by red line) for the integrated data of the $R10_3^9$ ($\mathscr{R}$=BB1(180)) from Figure 2.10. χ^2_{min} is labelled (simulation shown in Figure 2.13), and various confidence limits are marked by horizontal lines. Figure a) shows the limits according to Brinkmann's analysis (Eqn. 2.2.7), Figure b) shows the limits according to Ratkowsky's analysis (Eqn. 2.2.8).

(See section 2.2.4) better. However, Equation 2.2.7 will be used in most cases so as to allow the data to be compared to that of Brinkmann.

2.2.4 Method II

An early version of Method II was initially developed while working on the experiments in Section 3, and was therefore the first analysis procedure to be applied to experimental R-sequence data. It is based partly on the the manual for CERN's MINUIT fitting program [50], as well as "Statistical Methods in Experimental Physics" by Frederick James [51]. The method was temporarily abandoned over questions regarding the interpretation of E_{norm}, which will be required in Equation 2.2.9, but was later "resurrected" with modifications.

The most important revision to Method II was skipping the Fourier transformation when processing the spectra. As mentioned in Section 2.2.1 (page 28), there is only one peak in the ^{17}O spectrum, and its shape is irrelevant since it does not change. This means that the intensity of the resonance can be obtained directly from the FID. In principle, the first point of the FID would be sufficient, however the calculation of E_{norm} requires knowledge of the covariance matrix for the ex-

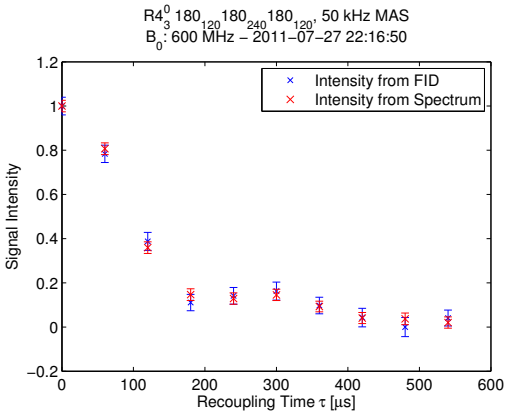


Figure 2.15: Comparison of fitted FID data (blue) vs. integrated spectral data (red).

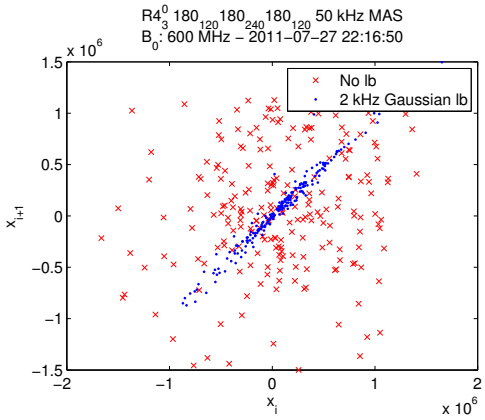


Figure 2.16: Correlation plots for a spectrum with (blue dots) and without (red crosses) line broadening.

perimental data; this was obtained by applying a 0^{th}-order phase correction to the FID and fitting the real part[7] with an exponential. As shown in Figure 2.15, the spectral intensities obtained by fitting the FID and by integrating the spectrum are comparable.

Working directly from the FID has two important advantages. Firstly, it circumvents most of the problems due to integration parameters from Figure 2.11, as well as the effects of line broadening shown in Figure 2.12. Secondly, zero filling and line broadening can cause correlation in the noise of the spectrum. This is seen readily in Figure 2.16: the baseline of a spectrum should contain only white noise, meaning that the value of a datum x_i should have no bearing on the value of its nearest neighbour x_{i+1}. This is the case when no line broadening is applied, and the data are distributed radially across the plot. When line broadening is applied, however, the Fourier transformation effectively interpolates the data, creating new points whose value lies between their left and right neighbours. This results in the blue pattern in Figure 2.16. This is not generally a problem for many practical NMR applications, but it has important mathematical consequences as many stochastic theories assume uncorrelated noise. It also means that the standard deviation for the baseline of a zero-filled spectrum may be lower than that of the same spectrum

[7] Phasing could be avoided by taking the magnitude of the FID, but the background noise would prevent the signal from decaying to 0.

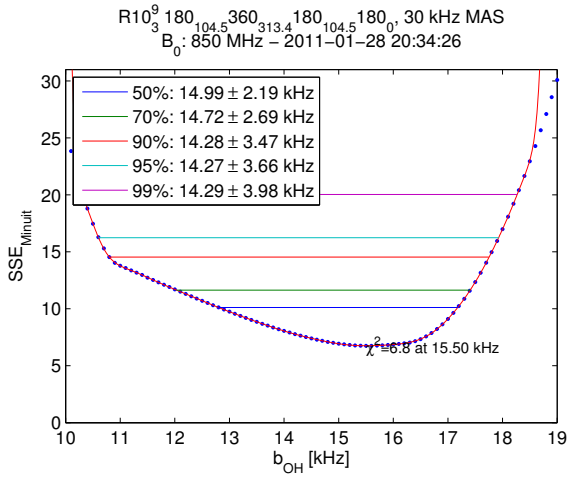


Figure 2.17: χ^2 as a function of b_{OH} for the R10$^9_3$ with $\mathscr{R}$ = BB1(180). As in Figure 2.14, blue dots correspond to fitted simulations and are interpolated by the red line. Confidence limits based on Equation 2.2.10 are shown by horizontal lines.

without zero-filling.

Once the experimental intensities were found, normalisation was performed according to equations 2.2.2–2.2.4 as before[8]. Damping also follows the same procedure (see Eqn. 2.2.6). However, χ^2 is calculated slightly differently as it is normalised by the error:

$$\chi^2 = \sum \left(\frac{I_{\text{norm}} - D}{E_{\text{norm}}} \right)^2 \tag{2.2.9}$$

The confidence limits are then found at

$$\chi^2 = \chi^2_{\text{min}} + C^{\beta}_{n_p} \tag{2.2.10}$$

where $C^{\beta}_{n_p}$ is the inverse of the chi-squared cumulative distribution function and β is the confidence level. It is worth noting at this point that Equation 2.2.10 differs from Equations 2.2.7 & 2.2.8 used by Brinkmann and Ratkowsky in that it is additive rather than multiplicative, which is made possible due to the normalisation in Equation 2.2.9.

Figure 2.17 shows the results of analysing the data set from Figure 2.14 using Method II. The two χ^2 plots have very similar shapes, each with a kink just be-

[8]Bearing in mind that the error E is determined from the square root of the covariance matrix for the FID fit, not from σ_{base}.

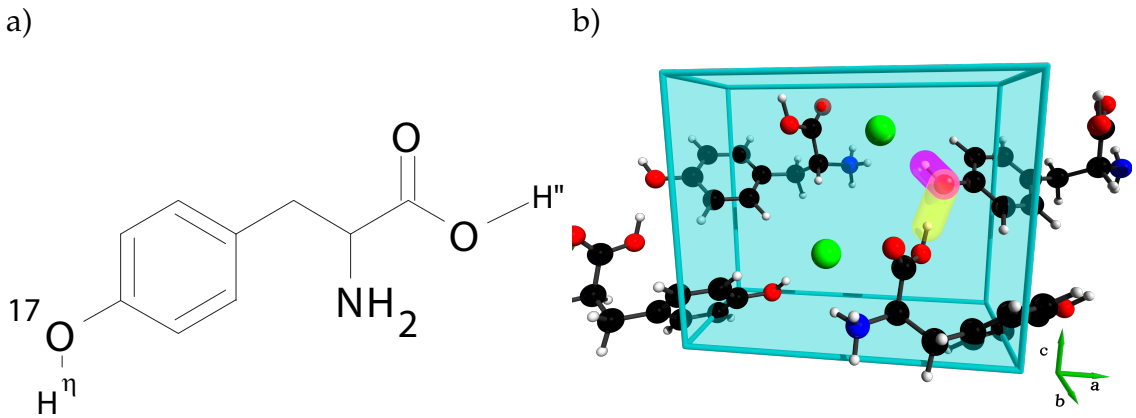


Figure 2.18: Molecular structure of tyrosine (a) and crystal structure of L-tyrosine.HCl (b). Atom colours follow the CPK scheme: hydrogen is white, carbon is black, oxygen is red, nitrogen is blue and chlorine is green. The O–H″ pair is highlighted in yellow, the O–H$^{\eta}$ in magenta. The unit cell is in the $P2_1$ space group with $a = 11.083$ Å, $b = 9.041$ Å, $c = 5.099$ Å, $\alpha = 90°$, $\beta = 91.82°$ and $\gamma = 90°$ [52].

low $b_{OH} \sim 11$ kHz and with their minima at 15.5 kHz and 15.8 kHz respectively. This is to be expected as the input data are roughly the same (though not identical), and E_{norm} is constant to a first approximation. The confidence limits, however, do not match quite so well. For instance, the Brinkmann χ^2 plot has the smallest 50 % interval at 15.6 ± 1.0 kHz, the Minuit plot is intermediate at 15 ± 2 kHz and the Ratkowsky plot lies at 14 ± 3 kHz. For the 95 % confidence limit, on the other hand, the Minuit plot is the smallest with 14 ± 4 kHz, followed by Brinkmann at 14 ± 5 kHz and the Ratkowsky 95 % confidence limit could not be reached.

2.2.5 Sample Preparation

In order to allow comparison with Reference [32] & [33], all experiments in this section were performed on ^{17}O labelled L-tyrosine (structure shown in Figure 2.18a). Tyrosine is a good model compound as it is a relatively small molecule, but simultaneously has a sufficiently asymmetric environment to give the oxygen atoms a realistic quadrupole moment. The T_1 time of protons in pure tyrosine is on the order of several seconds, but it can be reduced to about half a second by recrystallisation in hydrochloric acid. For the sample used in these experiments, the

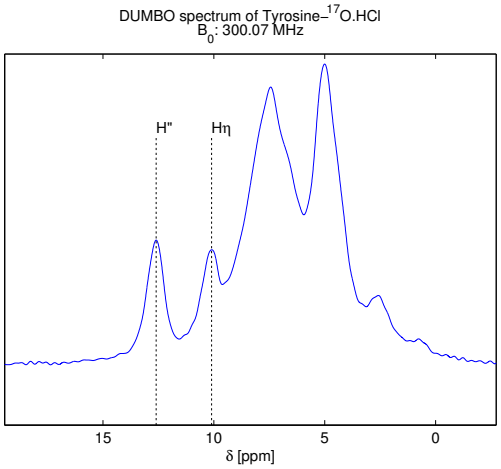


Figure 2.19: 1-pulse ^{1}H spectrum of tyrosine.HCl, recorded on a 300 MHz spectrometer at 12 kHz MAS. 100 kHz DUMBO [53] decoupling was employed during acquisition.

recrystallisation was kindly carried out by Dr. D. Mycock of the Hayes group.

The sample itself was labelled at 20 % to 30 % on the O$^{\eta}$ site (see Figure 2.18a). This site is highly convenient as it only has two neighbouring protons: the H$^{\eta}$, to which it is directly bonded, and the H" on a neighbouring molecule (crystal structure shown in Figure 2.18b). This allows us to test both the selective and non-selective sequences on the same site as the two proton signals can be resolved[9]. The two resonances are marked in Figure 2.19.

2.3 Results

To begin with, it is important to show the extent to which we are capable of reproducing the data of Brinkmann and Kentgens. Figure 2.20 shows the results of an $\mathsf{SR}4_1^2$ experiment[10] with no composite pulse, the same sequence as used in [33]. From their experiment, Brinkmann and Kentgens found a dipole-dipole coupling constant of $14\,770 \pm 260$ Hz, corresponding to a ^{1}H–^{17}O distance of 1.03 ± 0.01 Å. Using Method I with Brinkmann's χ^2 limits, we find a lower estimate: 12 ± 2 kHz

[9] This is required for the selective inversion pulse.

[10] $(\mathsf{R}4_1^2\mathsf{R}4_1^{-2})3^1$ in Brinkmann's notation.

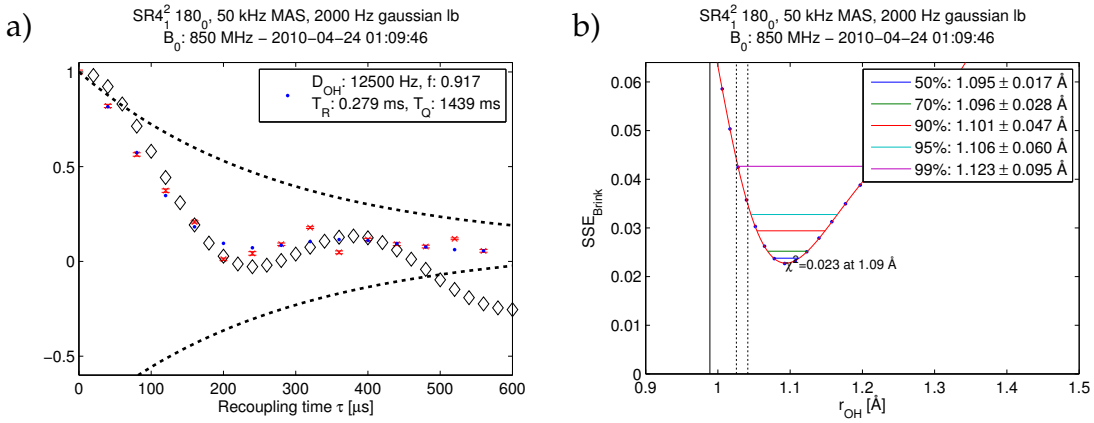


Figure 2.20: Simulated and experimental results for the SR4$_1^2$ sequence (no composite pulse) at 850 MHz and 50 kHz MAS. Figure a) shows the integrated peak intensities from the experimental data (plotted as red crosses with errorbars) against the recoupling time τ. The undamped simulation for the system with the "best" dipole-dipole coupling constant ($b_{OH} = 12.5$ kHz) is shown as black diamonds, the damped simulation is shown as blue dots. The black, dotted lines mark the maximum and minimum boundaries that the simulations could reach assuming the simulated data always lies between 1 & -1; this helps illustrate the "severity" of the damping parameters f, T_R & T_Q. Figure b) shows χ^2 against O–H$^\eta$ distance (blue dots); the simulation from a) is labeled ($b_{OH} = 12.5$ kHz corresponds to a distance of $r_{OH} = 1.09$ Å). The simulated data has been interpolated using a high-order polynomial (red line), and various confidence limits are also marked (see legend). The vertical black line shows the r_{OH} distance as recorded by neutron diffraction, and the dotted lines are 103.7 % and 105.3 % of this distance. Hence an ideal R-sequence would be centred somewhere between the dotted lines, assuming the effects of libration in tyrosine are similar to those in glycine.

or 1.11 ± 0.06 Å[11] as shown in Figure 2.20b.

The distance measured by neutron diffraction experiments is 0.989 Å (solid black line in Figure 2.20b, corresponds to $b_{OH} = 16.9$ kHz), but distances based on NMR dipole-dipole measurements tend to be slightly long; Ishii *et. al.* demonstrated computationally that molecular libration could introduce errors of 3.7 % to 5.3 % in glycine at ambient temperatures [54], and D. A. Case found discrepancies of up

[11] N.b. $\chi^2_{min} = 1.09$ Å, however the χ^2 plot is not symmetric. In this work, values will always be quoted as the centre of a given confidence limit.

to 12.6 % in other samples [55]. This means that the "best" measurements we can expect should lie between 1.035 Å and 1.041 Å (marked by black dotted lines in Figure 2.20b), but values as large as 1.114 Å can be expected. The measurement by Brinkmann is centred on the 3.7 % to 5.3 % range, while the results in Figure 2.20b are centred closer to 12 %, and have broader confidence limits.

There may be several factors which contribute to these differences. Firstly, Brinkmann and Kentgens recorded their data at a different field (800 MHz) and with a different probe, so factors such as B_1 field inhomogeneity may play a role. Secondly, the experiment in Figure 2.20 had half as many points as Brinkmann's since we did not sample between the $R4_1^2$ and $R4_1^{-2}$ supercycle steps. Finally, there are two points which appear to be anomalous, one at $\tau = 360\,\mu s$ and one at $\tau = 520\,\mu s$ which may be shifting and broadening the fit[12]. In spite of these setbacks, the experiment demonstrates that we can reproduce both the experimental data and analysis of Brinkmann and Kentgens.

As mentioned in Section 2.1.1, the $R4_1^2$ sequence does not have the correct recoupling properties when used without a supercycle, so it is unsuitable for our purposes. Instead, the $R12_3^5$ sequence from Brinkmann's earlier work will be used. In their original experiments (see [32]), Brinkmann and Kentgens reported a dipole-dipole coupling constant of 14.27 ± 0.38 kHz (1.046 ± 0.009 Å) for an $SR12_3^5$ experiment. The $SR12_3^5$ experiment shown in Figure 2.21b has a much broader confidence limit of 1.09 ± 0.11 Å. It is worth mentioning that the Brinkmann data once again had a higher sampling rate (20 µs) than the data in Figure 2.21 (60 µs). Figures 2.21c & d show the same experiment without supercycling; r_{OH} is found to be almost identical at 1.08 ± 0.14 Å. Interestingly, this suggests that the supercycle may not be providing much benefit with this sequence.

In contrast, the addition of a $90°_{90°}180°_{0°}90°_{90°}$ composite pulse greatly affects the sequence. As shown in Figure 2.21f, the 95 % confidence limit is estimated at 1.11 ± 0.03 Å. This constitutes a remarkable reduction in the confidence limits, but χ^2_{min} continues to overestimate the bond length by about 12 %. This offset was roughly consistent across all experiments so far, so the R-sequence was changed.

[12] If these points are excluded, the 95 % confidence limit shifts to 1.08 ± 0.04 Å

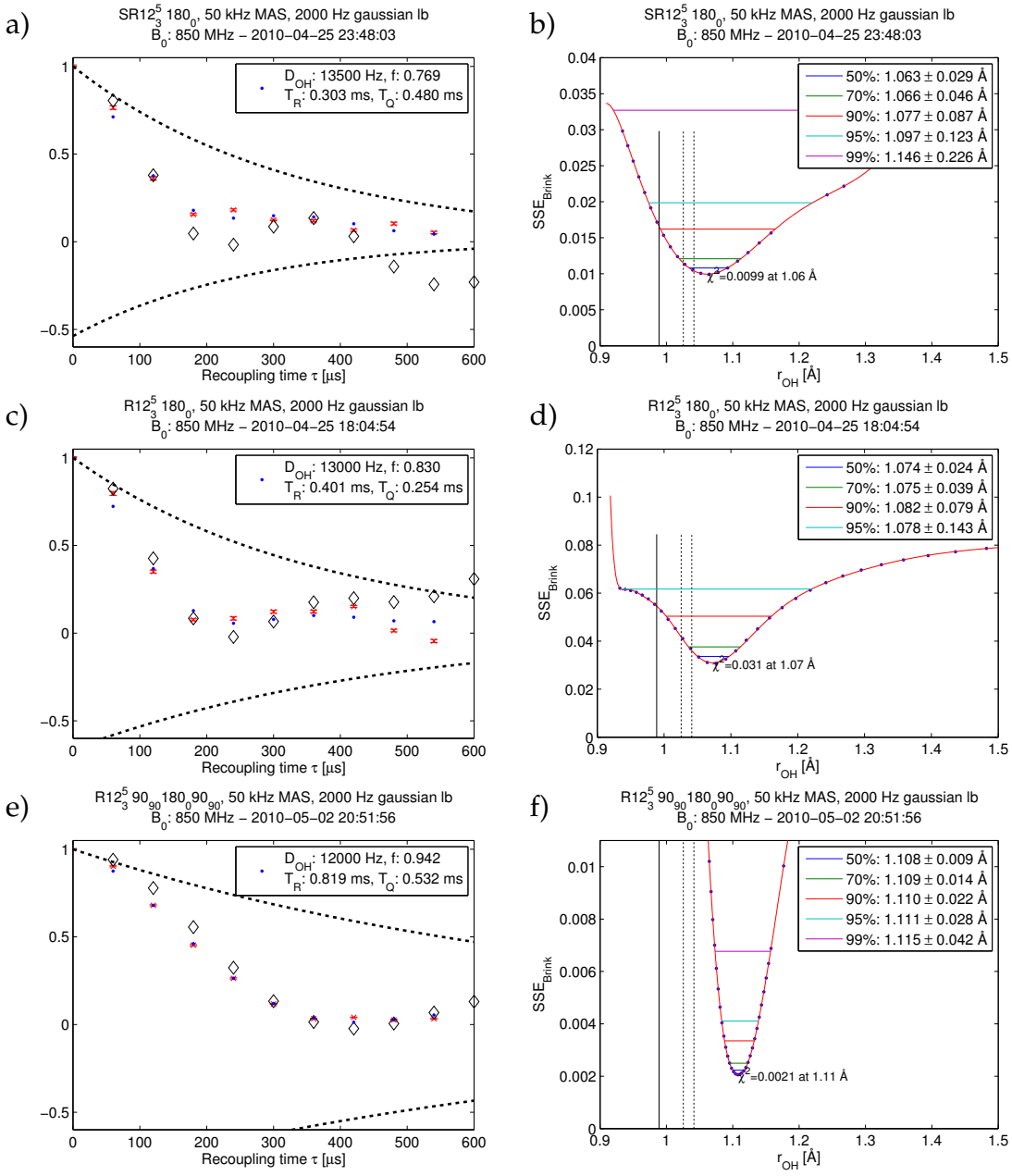


Figure 2.21: Simulated and experimental results for the SR12$_3^5$ ($\mathscr{R}$=180°$_{0°}$, Figures a & b), the R12$_3^5$ ($\mathscr{R}$=180°$_{0°}$, Figures c & d) and the R12$_3^5$ with $\mathscr{R}$=90°$_{90°}$180°$_{0°}$90°$_{90°}$ (e & f) at 850 MHz and 50 kHz MAS. See Figure 2.20 for plot details.

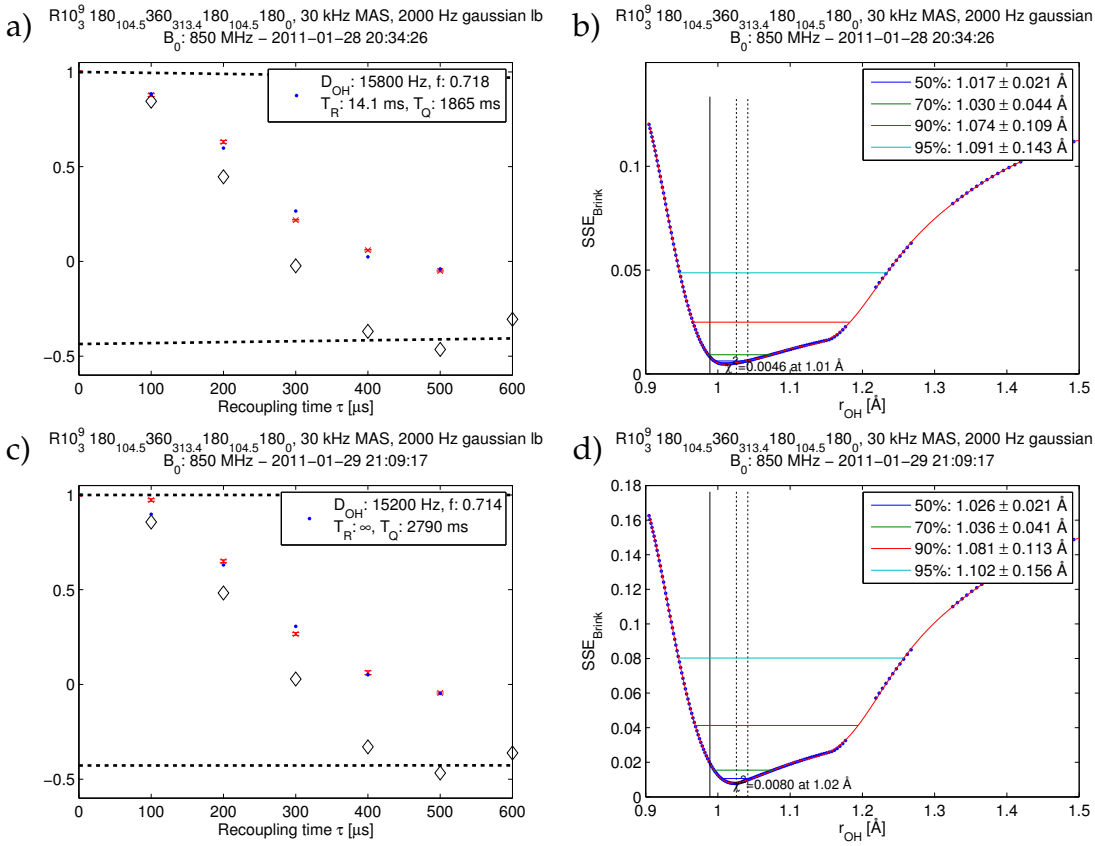


Figure 2.22: Simulated and experimental results for two separate runs of the $R10_3^9$ sequence ($\mathscr{R}$=BB1(180)) at 850 MHz and 30 kHz MAS. See Figure 2.20 for plot details.

The $R12_3^5$ recouples on $\{m = \pm 2, \mu = 0\}$, so the $R10_3^9$ was chosen as an alternative as it recouples on $\{m = 2, \mu = -1\}$ and $\{m = -2, \mu = 1\}$. This means that the $R10_3^9$ sequence recouples using an entirely different portion of the dipole-dipole Hamiltonian, but the sequence is also γ-encoded.

Figure 2.22 shows the results of an $R10_3^9$ experiment with a BB1(180) composite pulse, as well as a repetition of the experiment. Both experiments have very good χ^2_{min} values, 15.8 kHz (1.01 Å) and 15.2 kHz (1.02 Å) respectively, but the 95 % confidence limits are the worst so far at about 0.15 Å. Assuming that the difference in χ^2_{min} between the two experiments is representative, then the confidence limits are overestimating by almost an order of magnitude. Assuming that the experiments have been performed correctly, this would then imply a problem with the fitted model, or with the simulations.

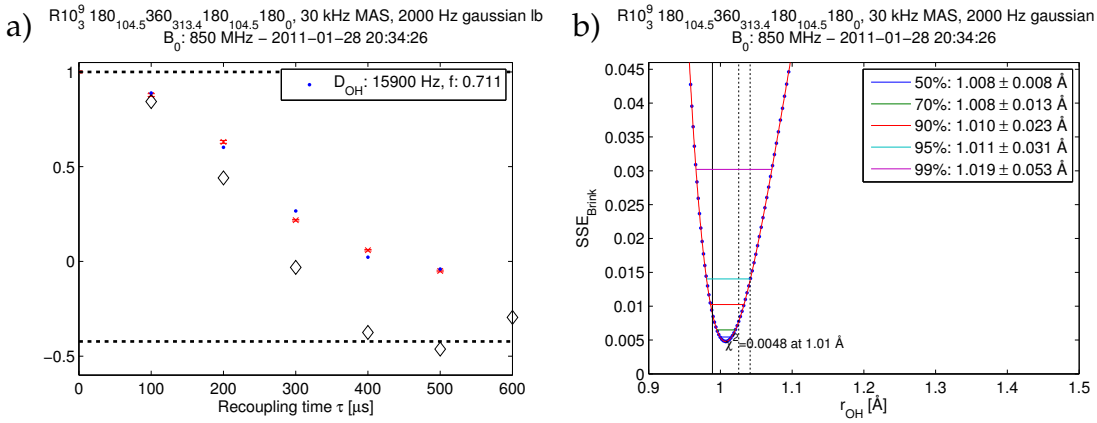


Figure 2.23: Simulated and experimental results for the experimental data from Figure 2.22a (R10$_3^9$, $\mathscr{R}$=BB1(180), $\frac{\omega_L(^1\mathrm{H})}{2\pi} = 850\,\mathrm{MHz}$, $\nu_{MAS} = 30\,\mathrm{kHz}$), but with T_R & T_Q fixed at infinity. See Figure 2.20 for plot details.

With regards to the model, it is worth noting at this point that even the smallest value of T_R (see Figure 2.22a) is 14 ms, which is much larger than $\tau_{\max} = 600\,\mu s$. For this experiment, at least, the exponential damping terms controlled by T_R & T_Q do not appear to contribute much to the overall fit. If they are excluded, then the fitted function becomes $D = fS_{\mathrm{norm}}(b_{\mathrm{OH}}) + (1-f)$, which still fits the experimental data fairly well[13]. As shown in Figure 2.23, the 95 % limit lies at 1.01 ± 0.03 Å, which constitutes an improvement over the previous results.

The simplified damping function works well for the R12$_3^5$ sequence (1.09 ± 0.06 Å) and the R12$_3^5$ with a $90°_{90°}180°_{0°}90°_{90°}$ composite pulse (1.08 ± 0.03 Å), but does not work particularly well for either of the supercycled experiments.

But, as mentioned earlier, it is also possible that the simulations can be improved. During calibration on the 1.3 mm probe, it was noticed that a ^{1}H nutation curve with a 90° pulse length of 1 µs would have the 360° zero crossing around 3.6 µs rather than at the expected 4 µs, so clearly B_1 inhomogeneity was substantial. Just as on page 33 of section 2.2.1, the simulations were rerun with varying degrees of B_1 amplitude offset. Assuming that the actual nutation frequency is $\omega_{act} = A \cdot \omega_{nom}$ where ω_{nom} is the nominal B_1 field and A is the percentage offset, then the

[13]Brinkmann's damping function was entirely empirical, so changing it does not violate any of the theory used in the analysis. On a related note, it is possible to skip the damping function completely, but the results tend to converge on the rather unsatisfying distance of 1.13 Å

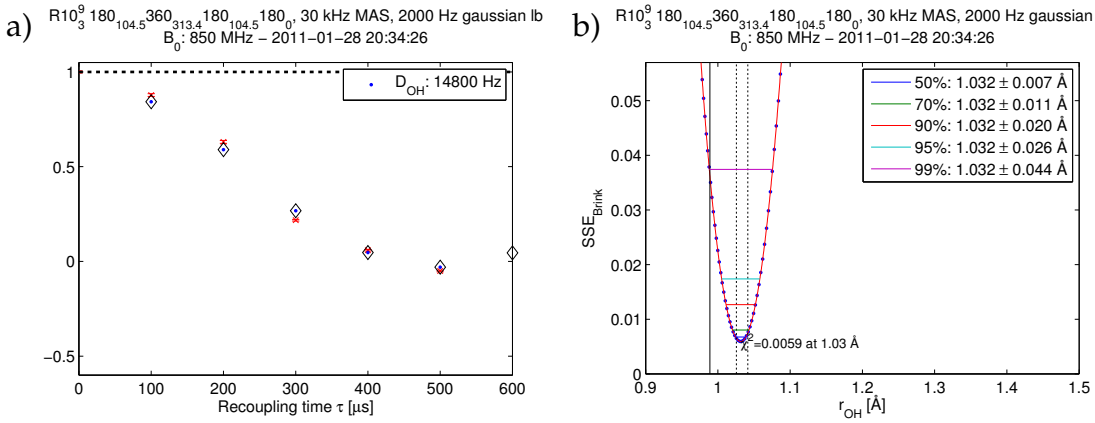


Figure 2.24: Simulated and experimental results for the experimental data from Figure 2.22a ($R10_3^9$, $\mathscr{R}$=BB1(180), $\frac{\omega_L(^1H)}{2\pi} = 850\,MHz$, $\nu_{MAS} = 30\,kHz$), but without empirical damping. B_1 inhomogeneity has been included with σ fixed at 60 %. See Figure 2.20 for plot details.

simulations were averaged with weight $w = e^{\left(\frac{A-100\,\%}{\sigma}\right)^2}$. Even without a damping function, the results for the $R10_3^9$ with a BB1(180) composite pulse are encouraging. As shown in Figure 2.24b, the 95 % limits lie at 1.032 ± 0.026 Å.

Initially, σ was included as an optional parameter in the fitting routine, but this was found to be impractical as it often tended towards unacceptably large values. As a compromise, the fitting program was instead run with several fixed values of σ, and the optimal value was chosen based on the width of the 95 % confidence limits. This awkward behaviour is perhaps not entirely surprising as the actual B_1 inhomogeneity profile is unknown; the Gaussian profile was chosen purely for simplicity. As such, the rather high value assumed by sigma (usually around 60 %) should probably be considered merely as a numerical fit parameter, rather than as any meaningful estimate of the B_1 inhomogeneity.

Non γ-encoded sequences have also been tested, although only at 600 MHz. Aside from the $R3n_n^{2n-1}$ set of sequences mentioned by Brinkmann, the only sequences which are not γ-encoded are $R4_1^0$ & $R4_3^0$. Figure 2.25 shows several such sequences, including an $R4_3^0$ with a $180°_{120°}180°_{240°}180°_{120°}$ composite pulse (Figures 2.25a & b). A sequence using Brinkmann's supercycle has also been tested, the results are shown in Figures 2.25c & d. The $R4_3^0$ gives a reasonable value (1.061 ± 0.066 Å);

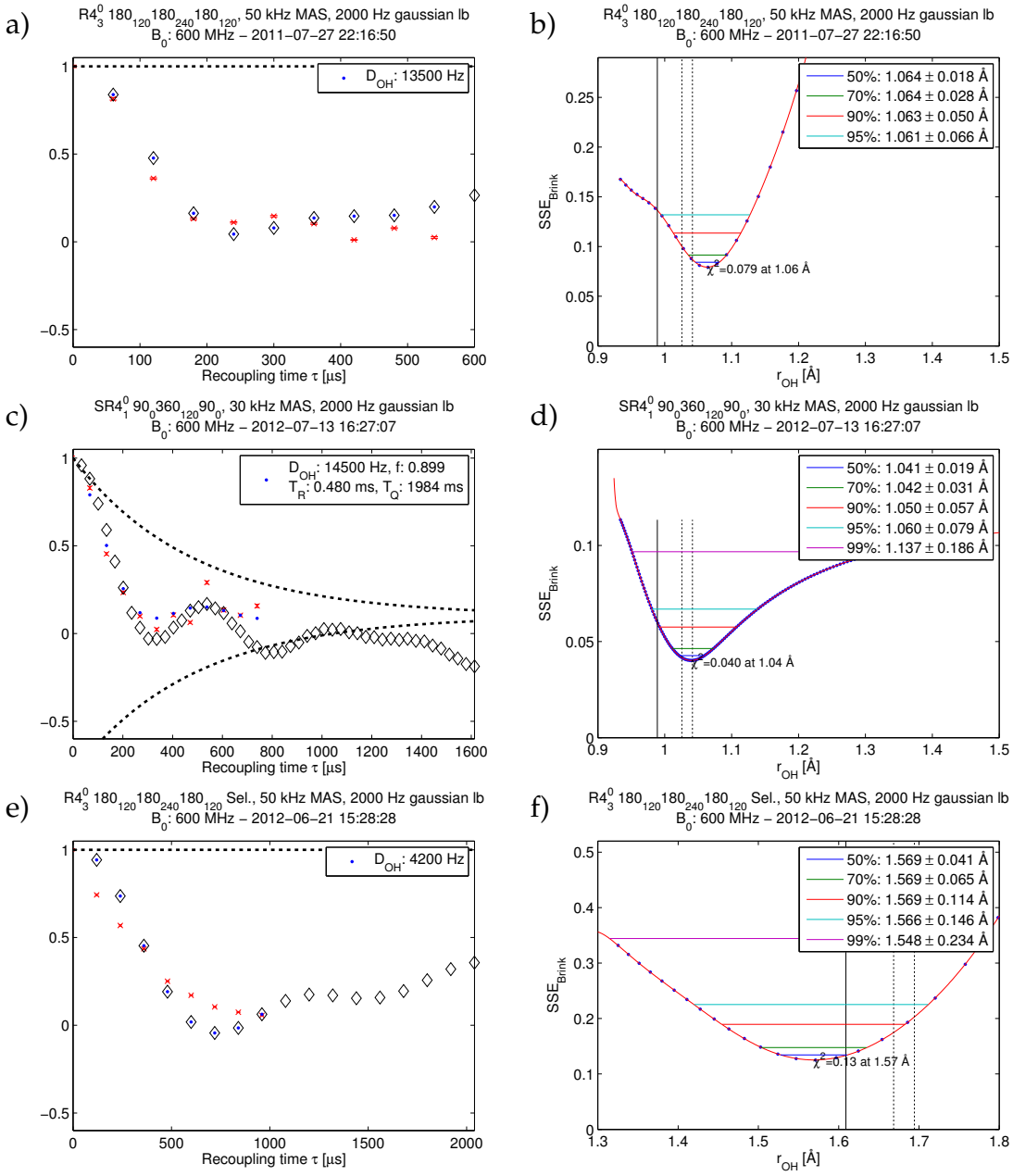


Figure 2.25: Figures a & b show simulated and experimental results for the $R4^0_3$ sequence ($\mathscr{R}$=180°$_{120°}$180°$_{240°}$180°$_{120°}$) at 600 MHz and 50 kHz MAS. The analysis was performed without damping, and with σ fixed at 60 %. Figures c & d show simulated and experimental results for the $SR4^0_1$ sequence ($\mathscr{R}$=90°$_{0°}$360°$_{120°}$90°$_{0°}$) at 600 MHz and 30 kHz MAS. The analysis was performed with damping, and without B_1 inhomogeneity. Figures e & f show a selective $R4^0_3$ sequence ($\mathscr{R}$=180°$_{120°}$180°$_{240°}$180°$_{120°}$) at 600 MHz and 50 kHz MAS. No damping was used. See Figure 2.20 for plot details.

the $SR4_1^0$ has a better χ^2_{min} location (1.04 Å), however the 95 % confidence limit is broader (1.06 ± 0.08 Å). Overall, both sequences perform pretty well, considering the field strength.

Finally, Figures 2.25e & f show the results of a selective experiment, also performed at 600 MHz. The measured ^{1}H-^{17}O distance comes out as 1.56 ± 0.15 Å[14], which is clearly different from the 1.06 ± 0.66 Å measured in the non-selective experiment with the same sequence (see Figure 2.25). This distance is also close to the O-H″ distance of 1.609 Å measured by neutron diffraction. However, once libration is taken into account, one would expect an NMR distance closer to 1.7 Å than 1.6 Å. Unfortunately, the selective inversion pulse required for the selective recoupling experiment is sufficiently long to impact on signal. This means that a compromise must be struck between the signal-to-noise ratio, and the bandwidth of the inversion pulse. It is possible, therefore, that the inversion pulse used was not sufficiently narrow for an experiment at 600 MHz and that some of the H^{η} have been inverted as well, resulting in a stronger interaction (and hence shorter distance) than would otherwise be expected. However, the experiment demonstrates that the $R4_3^0$ sequence with a $180°_{120°}180°_{240°}180°_{120°}$ composite pulse should be suitable for selective recoupling experiments, so long as they are performed at higher field, where the H^{η} and H″ peaks are more separated.

2.4 Extension to ^{1}H CSA

As shown in Table 2.1 on page 25, the CSA and the $\hat{\boldsymbol{I}}\hat{\boldsymbol{S}}$ dipole–dipole coupling have the same space rank l & spin rank λ, so both interactions will always be recoupled by the same single channel R-sequences. While this fact was treated as a nuisance in Section 2.2.1, it can be used in its own right to determine the CSA parameters of protons under rapid MAS. Older experiments for determining the CSA, such as 2D-PASS (see Section 3.2), must be performed at low MAS rates so as to conserve the side-band patterns. This means that they cannot be applied to protons as the strong homonuclear dipole-dipole couplings generally require MAS rates on the

[14] Brinkmann measured 1.73 ± 0.12 Å

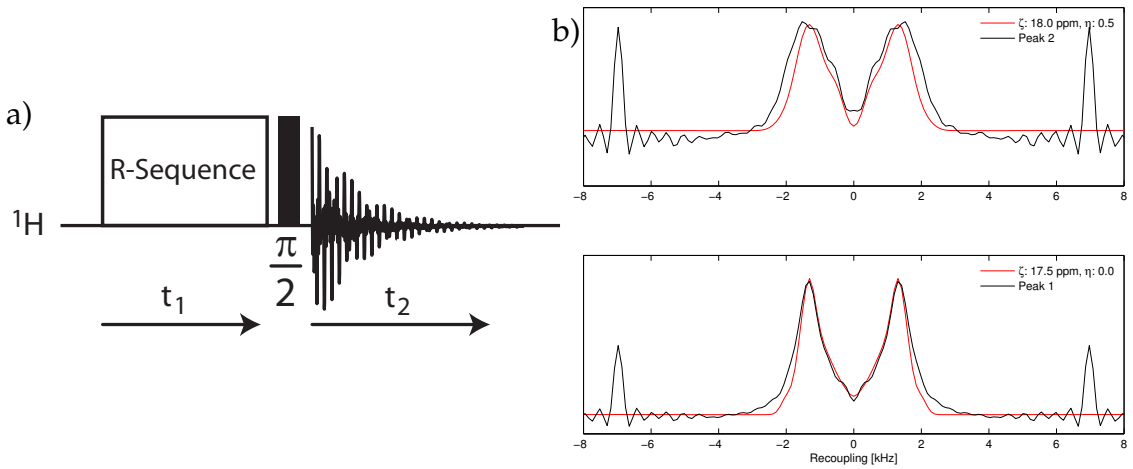


Figure 2.26: Figure 2.26a shows the pulse sequence for a Proton CSA recoupling experiment. Figure 2.26b shows the cross sections for peaks 1 & 2 (black lines) of the experiment in Figure 2.27, as well as the best fit SIMPSON simulations (red lines). CSA parameters are found to be $\zeta = 17.5$ ppm with $\eta = 0$ for peak 1 and $\zeta = 18.0$ ppm with $\eta = 0.5$ for peak 2.

order of 30 kHz or higher.

Figure 2.26a shows a pulse sequence (originally developed by Brouwer & Ripmeester using an $R18_2^5$ [56], see Table 2.2 for other viable sequences) which is, in principle, capable of recoupling at high MAS rates. Interestingly, the pulse sequence begins with the R-sequence, meaning that only $\hat{I}_z$ magnetisation is present at the start and end of the recoupling period t_1. This is in stark contrast to experiments such as 2D-PASS, which begin the evolution period t_1 with $\hat{I}_x$ magnetisation. The proton magnetization is then rotated into the x-y plane and the signal is acquired as normal. The resulting spectrum is Fourier transformed in t_2, and then the real part of the spectrum is Fourier transformed in t_1.

Unfortunately, Brouwer & Ripmeester did not actually use the pulse sequence in Figure 2.26a; they ran the sequence at 16.1 kHz MAS and used wPMLG5 [57, 58] during acquisition to decouple the protons. This works, but the use of wPMLG5 is an extra complication on the sequence, and also scales the isotropic chemical shifts. The reason for the low spinning speed lies in the power requirements of the $R18_2^5$ itself: if the probe's limit for a 180° pulse is 2.8 µs, as is the case for the Nottingham 600 MHz spectrometer's 1.3 mm probe, then an $R18_2^5$ sequence cannot

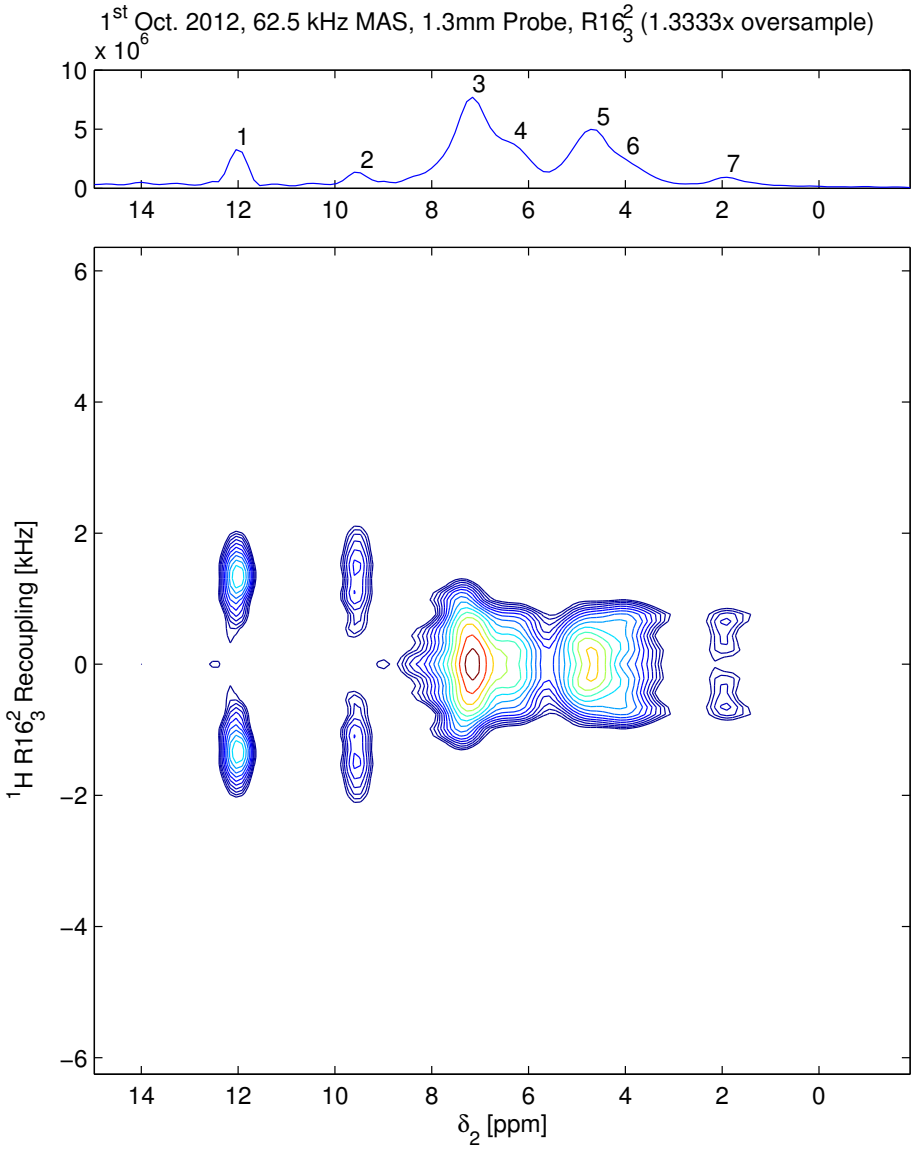


Figure 2.27: R16$_3^2$ proton CSA recoupling experiment performed on L-tyrosine.HCl at 600 MHz. The rescan delay was 4 s, the MAS rate was 62.5 kHz, the proton nutation rate $\nu_{1\mathrm{H}}$ was 167 kHz and the spectral width was 150 kHz. The experiment was oversampled by 33 % in order to remove a zero-frequency artefact. Spectrum courtesy of Habeeba Miah.

be run above 39.7 kHz. While this may be enough for some samples, in tyrosine, for instance, only two peaks in the proton spectrum are properly resolved even at 62.5 kHz (see skyline plot of Figure 2.27).

With an appropriate R-sequence, however, the experiment can be performed at sufficiently high MAS rates that RF decoupling is not required during acquisition. Figure 2.27 shows the results for tyrosine using an $\mathrm{R16}_3^2$ sequence at 62.5 kHz on a 600 MHz spectrometer. Just as with Brouwer's $\mathrm{R18}_2^5$, the $\mathrm{R16}_3^2$ is a γ-encoded sequence which recouples the $\{m = \pm 2, \mu = \mp 1\}$ term of the dipole-dipole coupling. The experiment was oversampled by 33 % (4 points recorded per 3 R-sequence blocks) as the non-oversampled experiment was found to have a zero-frequency artefact; the oversampled version also has artefacts, but they lie outside the pattern at about 6 kHz. As can be seen from the skyline plot of Figure 2.27, the spinning speed is sufficient to achieve a relatively high resolution spectrum in the direct dimension δ_2. It is clear from the cross sections of peak 1 (H'') & peak 2 (H^{η}) shown in Figure 2.26b that the recoupled wideline spectra produced by the experiment in ω_1 are sufficiently distinctive to be sensitive to the proton CSA; the proton chemical shift parameters have been determined through SIMPSON simulations to be $\zeta = 17.5$ ppm with $\eta = 0$ for peak 1 and $\zeta = 18.0$ ppm with $\eta = 0.5$ for peak 2.

2.5 Conclusions

In conclusion, the experiments of Brinkmann & Kentgens have been repeated on the 850 MHz magnet of the "UK 850 MHz Solid-State NMR Facility". Similar results were obtained, even with little or no oversampling, but the confidence limits were broader. Experiments can be designed and their resilience to effects such as B_1 field inhomogeneity can be tested using the SIMPSON simulation environment. Composite pulses can be used to mitigate the effects of B_1 field inhomogeneity; the $\mathrm{R12}_3^5$, for instance, is much improved by the use of $\mathscr{R}$=$90°_{90°}180°_{0°}90°_{90°}$. These effects can again be investigated in advance with SIMPSON, as demonstrated by the $\mathrm{R10}_3^9$ sequence where simulations showed it would most likely be unuseable with $\mathscr{R}$=$180°_{0°}$ or $\mathscr{R}$=$180°_{120°}180°_{240°}180°_{120°}$, but that it would work well with a BB1(180) composite pulse. Furthermore, it is shown that the analysis of the data

from such experiments can be improved. It is possible to bring the signal-to-noise ratio of the FID into the calculation of the confidence limits in a meaningful way, and the damping functions used by Brinkmann and Kentgens can be simplified for experiments which do not use supercycles, or can be left out entirely if B_1 field inhomogeneity is included explicitly in the SIMPSON simulations. It is also demonstrated that an $R4_3^0$ sequence with $\mathscr{R}$=$180°_{120°}180°_{240°}180°_{120°}$ can be used for both selective and non-selective experiments, although the non-selective experiment would probably have benefited from a B_0 higher than 600 MHz to allow better selectivity of the ^{1}H Gaussian inversion pulse. And finally, it is also shown that the R-sequences used here have utility beyond Brinkmann & Kentgens' dipolar recoupling experiments, as the sequences can equally well be employed to measure CSA parameters.

3 Spectral assignment through CSA tensor analysis

Solution-state NMR is routinely used for the determination of molecular structures, but there are times when dissolving the sample is undesirable. For instance, some substances are *polymorphic*, meaning that they can exist in several different crystal structures at standard temperature and pressure. Polymorphism is of interest to the chemical industry, particularly the pharmaceutical industry, as patents are generally issued for a particular crystal structure only. The different polymorphs may also have slightly different pharmacological properties, e.g. the rate at which they dissolve in a particular solvent or the rate at which they are metabolized by the human body.

Furthermore, some interactions are available for analysis in solids which are not present in solutions, e.g. the CSA and the dipole-dipole interaction. In this section, the assignment of a congested spectrum will be demonstrated through measurements of the full chemical shift tensor and iterative fitting to DFT calculations.

3.1 Chemical Shift Anisotropy under MAS

As already shown in Equation 1.1.66, the chemical shift of a given spin is dependent on the orientation of the molecule relative to the applied magnetic field. In solutions, the interaction is usually averaged to a single value due to rapid, isotropic molecular tumbling. In solid samples, however, *chemical shift anisotropy* (CSA) is not averaged, and the resonant frequency of a given site in a given orientation may differ significantly from δ^{iso}. In powdered samples, all possible orientations are present simultaneously, resulting in so-called *powder lineshapes* as shown in Figure 3.1.

There is an analytical equation for the powder pattern [59]:

$$S(\delta) = \begin{cases} \frac{\int_0^{\frac{\pi}{2}} \left\{1-sin^2(\phi)\cdot\left[\frac{\delta_{22}-\delta_{33}}{\delta_{11}-\delta_{22}}\frac{\delta_{11}-\delta}{\delta-\delta_{33}}\right]\right\}^{-\frac{1}{2}} d\phi}{\pi\sqrt{(\delta-\delta_{33})(\delta_{11}-\delta_{22})}} & \text{if } \delta_{33} < \delta \leq \delta_{22} \\ \frac{\int_0^{\frac{\pi}{2}} \left\{1-sin^2(\phi)\cdot\left[\frac{\delta_{11}-\delta_{22}}{\delta_{22}-\delta_{33}}\frac{\delta_{33}-\delta}{\delta_{11}-\delta}\right]\right\}^{-\frac{1}{2}} d\phi}{\pi\sqrt{(\delta_{11}-\delta)(\delta_{22}-\delta_{33})}} & \text{if } \delta_{22} < \delta < \delta_{11} \\ 0 & \text{otherwise} \end{cases} \tag{3.1.1}$$

where δ_{11}, δ_{22} & δ_{33} are the singularities of the pattern, ordered such that $\delta_{11} \geq \delta_{22} \geq \delta_{33}$. The three singularities naturally correspond to the three principal values of the chemical deshielding tensor $\boldsymbol{\delta}$, and are usually assigned to δ_{xx}, δ_{yy} & δ_{zz} in such a way[1] that $\left|\delta_{zz} - \delta^{iso}\right| \geq \left|\delta_{xx} - \delta^{iso}\right| \geq \left|\delta_{yy} - \delta^{iso}\right|$. When discussing CSAs, however, it is more common to use the parametrization described at the end of Section 1.1.4:

$$\delta^{iso} = \frac{1}{3}\sum_i \delta_i \qquad \zeta_{\boldsymbol{\delta}} = \delta_{zz} - \delta^{iso} \qquad \eta_{\boldsymbol{\delta}} = \frac{\delta_{yy} - \delta_{xx}}{\zeta_{\boldsymbol{\delta}}}$$

The corresponding chemical shielding tensor $\boldsymbol{\sigma}$ is defined by the same parameters:

$$\begin{aligned} \sigma^{iso} &= \frac{1}{3}\sum_i \sigma_i & \zeta_{\boldsymbol{\sigma}} &= \sigma_{zz} - \sigma^{iso} & \eta_{\boldsymbol{\sigma}} &= \frac{\sigma_{yy} - \sigma_{xx}}{\zeta_{\boldsymbol{\sigma}}} \\ &= -\delta^{iso} & &= -\zeta_{\boldsymbol{\delta}} & &= \eta_{\boldsymbol{\delta}} \end{aligned}$$

These parameters each have well-defined interpretations in the spectrum. δ^{iso} corresponds to the weighted average of the entire peak, the anisotropy $\zeta_{\boldsymbol{\delta}}$ controls the width of the peak and the asymmetry parameter $\eta_{\boldsymbol{\delta}}$ determines its shape[2].

As a general rule, ζ for an individual site tends to be on the same order as the range of values normally encountered in δ^{iso} for that nucleus, leading to significant spectral overlap. This makes the assignment of peaks in the static spectrum of powder samples impossible for all but the smallest molecules; instead, it is common practice to perform so-called *magic angle spinning* (MAS) experiments where the sample is packed into a rotor and spun rapidly (1 kHz to 80 kHz, depending on sample, experiment and available hardware) at the *magic angle*: $\beta_{MAS} = \arctan\sqrt{2} \approx 54.74°$.

[1] This is the Haeberlen-Mehring-Spiess convention. Others exist, e.g. Herzfeld-Berger notation.

[2] N.B. η refers to the asymmetry of the tensor, not the lineshape. $\eta = 1$ produces a lineshape which is fully symmetric about δ^{iso} with width $2\zeta_{\boldsymbol{\delta}}$

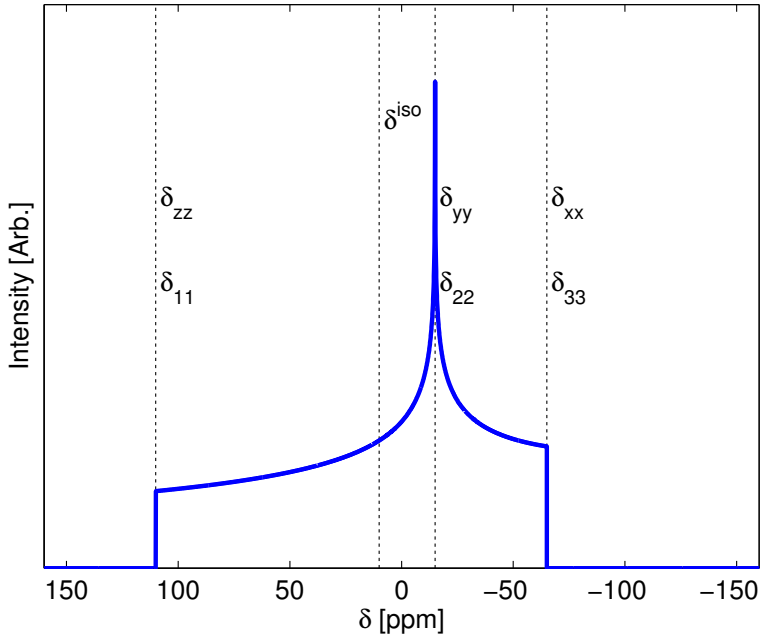


Figure 3.1: Typical powder pattern; $\delta^{iso} = 10\,\text{ppm}$, $\zeta = 100\,\text{ppm}$, $\eta = 0.5$.

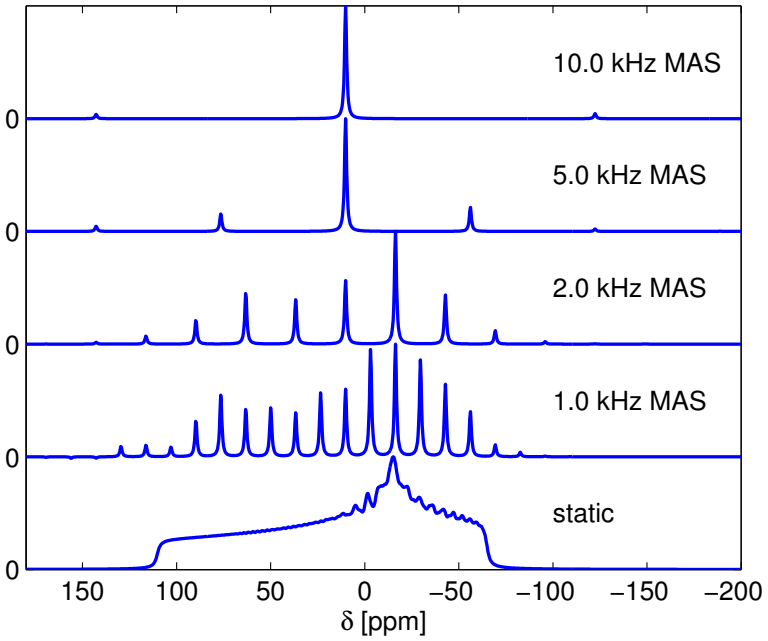


Figure 3.2: Simulated MAS spectra; $\delta^{iso} = 10\,\text{ppm}$, $\zeta = 100\,\text{ppm}$, $\eta = 0.5$.

As already mentioned in Equation 1.1.4 in Section 1.1.1, the Chemical Shift Hamiltonian for a single spin-half nucleus is

$$\hat{\mathscr{H}}_{CS} = \gamma \hat{\boldsymbol{I}} \boldsymbol{\sigma}\left(\boldsymbol{\Theta}\right) \boldsymbol{B}_0 \tag{3.1.2}$$

$$= c^{CS} \sum_{i=0}^{2} \sum_{j=-i}^{i} \left(-1\right)^{-j} M_{i-j}^{CS} \hat{\boldsymbol{T}}_{ij}^{CS} \tag{3.1.3}$$

Under the secular approximation, $\hat{\boldsymbol{T}}_{2\pm2}^{CS} = 0$ in the lab frame ($\boldsymbol{B}_0 \parallel \hat{\boldsymbol{\epsilon}}_1^0$), hence:

$$\hat{\mathscr{H}}_{CS}^{Lab} = c^{CS} \sum_{i=0}^{2} \sum_{j=-i}^{i} \left(-1\right)^{-j} M_{i-j}^{CS,Lab} \hat{\boldsymbol{T}}_{ij}^{CS,Lab} \tag{3.1.4}$$

$$= c^{CS} \left(M_{00}^{CS} \hat{\boldsymbol{T}}_{00}^{CS} + M_{20}^{CS,Lab} \hat{\boldsymbol{T}}_{20}^{CS,Lab} \right) \tag{3.1.5}$$

where $c^{CS} = \gamma$ and $c^{CS} M_{00}^{CS} \hat{\boldsymbol{T}}_{00}^{CS}$ is easily identified as the isotropic chemical shift Hamiltonian $\sigma^{iso} \mathrm{B}_0 \hat{I}_z$. It is also known from theory that $\hat{\boldsymbol{T}}_{20}^{CS,Lab} = \sqrt{\frac{2}{3}} \mathrm{B}_0 \hat{I}_z$, but $M_{20}^{CS,Lab}$ must be calculated individually for each crystal orientation.

The spherical tensor coefficients are known in the principal axis frame ($M_{ij}^{CS,PAF}$), so two rotations will be required. First, the coefficients are transformed into the rotor frame using Equation 1.1.66 where the Euler angles $\boldsymbol{\Theta}^{PAF,R}$ may take any values as the powder is assumed to be isotropic. Then, a second transformation takes the coefficients into the laboratory frame with $\boldsymbol{\Theta}^{R,Lab}\left(t\right) = \left(\omega_{MAS} t, \beta_{MAS}, 0\right)$. The final result is:

$$M_{20}^{CS,Lab} = \sum_{m'=-2}^{2} \sum_{n=-2}^{2} M_{2m'}^{CS,PAF} D_{m'n}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) D_{n0}^{(2)}\left(\boldsymbol{\Theta}^{R,Lab}\left(t\right)\right) \tag{3.1.6}$$

$$= \sum_{m'=-2}^{2} \sum_{n=-2}^{2} M_{2m'}^{CS,PAF} D_{m'n}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{n0}^{(2)}(\beta_{MAS})\, e^{-in\omega_{MAS}t} \tag{3.1.7}$$

or, if we substitute $-m$ for n:

$$M_{20}^{CS,Lab} = \sum_{m'=-2}^{2} \sum_{m=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS})\, e^{im\omega_{MAS}t} \tag{3.1.8}$$

Substituting back into Equation 3.1.5, we find:

$$\hat{\mathscr{H}}_{CS}^{Lab} = c^{CS} M_{00}^{CS} \hat{\boldsymbol{T}}_{00}^{CS} + c^{CS} M_{20}^{CS,Lab} \hat{\boldsymbol{T}}_{20}^{CS,Lab} \tag{3.1.9}$$

$$= \gamma \sigma^{iso} \mathrm{B}_0 \hat{I}_z + \gamma \sum_{m'=-2}^{2} \sum_{m=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS})\, e^{im\omega_{MAS}t} \sqrt{\frac{2}{3}} \mathrm{B}_0 \hat{I}_z \tag{3.1.10}$$

$$= \left(\omega^{iso} + \sqrt{\frac{2}{3}} \gamma \mathrm{B}_0 \sum_{m'=-2}^{2} \sum_{m=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS})\, e^{im\omega_{MAS}t} \right) \hat{I}_z \tag{3.1.11}$$

where the angular isotropic frequency $\omega^{iso} = \gamma \sigma^{iso} \mathrm{B}_0$. As the entire Hamiltonian is proportional to $\hat{I}_z$, we define the frequency of the crystal:

$$\omega\left(t; \boldsymbol{\Theta}^{PAF,R}\right) = \omega^{iso} + \sqrt{\frac{2}{3}} \gamma \mathrm{B}_0 \sum_{m'=-2}^{2} \sum_{m=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS})\, e^{im\omega_{MAS}t} \tag{3.1.12}$$

This expression can be simplified by defining a set of frequencies $\omega^{(m)}$ as follows:

$$\omega^{(m)}\left(\boldsymbol{\Theta}^{PAF,R}\right) = \delta_{m0}\omega^{iso} + \sqrt{\frac{2}{3}} \gamma \mathrm{B}_0 \sum_{m'=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\boldsymbol{\Theta}^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS}) \tag{3.1.13}$$

Hence, $\omega\left(t; \boldsymbol{\Theta}^{PAF,R}\right)$ becomes:

$$\omega\left(t; \boldsymbol{\Theta}^{PAF,R}\right) = \sum_{m=-2}^{2} \omega^{(m)}\left(\boldsymbol{\Theta}^{PAF,R}\right) e^{im\omega_{MAS}t} \tag{3.1.14}$$

In accordance with Antzukin *et al.* [60], we now group all crystals with the same values of $\alpha^{PAF,R}$ & $\beta^{PAF,R}$ into a so-called carousel, since a full rotation around the rotor axis is equivalent to sampling all possible values of $\gamma^{PAF,R}$. The frequency for a given segment of the carousel at $\gamma^{PAF,R}$ is then

$$\omega_c^{(m)}\left(\gamma^{PAF,R}\right) = \delta_{m0}\omega^{iso} + \sqrt{\frac{2}{3}} \gamma \mathrm{B}_0 \sum_{m'=-2}^{2} M_{2m'}^{CS,PAF} D_{m'-m}^{(2)}\left(\alpha_c, \beta_c, \gamma^{PAF,R}\right) d_{-m0}^{(2)}(\beta_{MAS}) \tag{3.1.15}$$

and its time evolution follows

$$\omega_c\left(t; \gamma^{PAF,R}\right) = \sum_{m=-2}^{2} \omega_c^{(m)}\left(\gamma^{PAF,R}\right) e^{im\omega_{MAS}t} \tag{3.1.16}$$

Also, note the symmetry properties of $\omega_c^{(m)}$:

$$\omega_c^{(m)}(\gamma) = \left(\omega_c^{(-m)}(\gamma)\right)^* \qquad\qquad \omega_c^{(m)}(\gamma) = \omega_c^{(m)}(0)\, e^{im\gamma}$$

Let us now consider the signal from a given carousel. Assuming that the sample is prepared in the state $\hat{\boldsymbol{\rho}} = \frac{1}{2}\left(\hat{\mathbf{1}} + \mathbb{B}\hat{I}_-\right)$ as per the arguments regarding free precession on page 12, we may determine the evolution of the density matrix according to Equation 1.1.45, resulting in the density matrix $\hat{\boldsymbol{\rho}}_c$:

$$\hat{\boldsymbol{\rho}}_c\left(t;\gamma^{PAF,R}\right) = \frac{1}{2}\left(\hat{\mathbf{1}} + \mathbb{B}e^{i\Phi_c\left(t,0;\gamma^{PAF,R}\right)}\hat{I}_-\right) \tag{3.1.17}$$

where $\Phi_c(t,0;\gamma)$ is the accumulated phase angle at time t:

$$\Phi_c(t,0;\gamma) = \int_{t'=0}^{t} \omega_c(t';\gamma)\,\mathrm{d}t' \tag{3.1.18}$$

$$= \omega_c^{(0)}t + \phi_c\left(0;\gamma\right) + \xi_c\left(t;\gamma\right) - \xi_c\left(0;\gamma\right) \tag{3.1.19}$$

where $\xi_c\left(0;\gamma\right)$ & $\phi_c\left(0;\gamma\right)$ are integration constants, and ξ_c is defined as:

$$\xi_c\left(t;\gamma\right) = \sum_{m\neq 0} \frac{\omega_c^{(m)}(\gamma)\, e^{im\omega_{MAS}t}}{im\omega_{MAS}} \tag{3.1.20}$$

Due to the symmetry properties of $\omega_c^{(m)}$, it may be shown that:

$$\xi_c\left(t;\gamma\right) = \xi_c\left(t + \frac{\gamma}{\omega_{MAS}};0\right) \qquad\qquad \xi_c\left(t;\gamma\right) = \xi_c\left(0;\gamma + \omega_{MAS}t\right)$$

Since we detect using the $\hat{I}_+$ operator, the observed signal from a given carousel segment has the following form:

$$S_c\left(t;\gamma^{PAF,R}\right) \propto e^{i\left(\omega_c^{(0)}t + \phi_c\left(0;\gamma^{PAF,R}\right) + \xi_c\left(t;\gamma^{PAF,R}\right) - \xi_c\left(0;\gamma^{PAF,R}\right)\right)} \tag{3.1.21}$$

$d^{(2)}_{-m0}(\beta) = 3\cos^2(\beta) - 1$, so $\omega_c^{(0)} = \omega^{iso}$ for $\beta_{MAS} = \arctan\sqrt{2}$, and ξ_c vanishes as ω_{MAS} tends to infinity. This means that sufficiently rapid spinning at the correct angle causes all carousels to resonate only at ω^{iso}. At finite speeds, however, more analysis is required.

Since $\xi_c(t + n\tau_{MAS};\gamma) = \xi_c(t;\gamma)$ (τ_{MAS} is the rotor period, $\frac{2\pi}{\omega_{MAS}}$, and n is any integer), it follows that $S_c(t + n\tau_{MAS};\gamma) = S_c(t;\gamma)\, e^{i\omega_c^{(0)}n\tau_{MAS}}$. The signal may therefore be broken down into a Fourier series:

$$S_c\left(t;\gamma^{PAF,R}\right) \propto \sum_{k=-\infty}^{\infty} a_c^{(k)}\left(\gamma^{PAF,R}\right) e^{i\left(\omega_c^{(0)} + k\omega_{MAS}\right)t} \tag{3.1.22}$$

The Fourier coefficients $a_c^{(k)}(\gamma)$ are calculated in the usual way:

$$a_c^{(k)}\left(\gamma^{PAF,R}\right) = \frac{1}{\tau_{MAS}} \int_0^{\tau_{MAS}} S_c(t;\gamma)\, e^{-i\left(\omega_c^{(0)} + k\omega_{MAS}\right)t} \mathrm{d}t \tag{3.1.23}$$

The average contribution of the entire carousel can then be calculated by integrating over the contributions from all possible values of $\gamma^{PAF,R}$:

$$\langle S_c(t)\rangle \propto \frac{1}{2\pi} \int_{\gamma=0}^{2\pi} S_c(t;\gamma)\, \mathrm{d}\gamma \tag{3.1.24}$$

$$\propto \sum_{k=-\infty}^{\infty} \left\langle a_c^{(k)} \right\rangle e^{i\left(\omega_c^{(0)} + k\omega_{MAS}\right)t} \tag{3.1.25}$$

$$\left\langle a_c^{(k)} \right\rangle = \frac{1}{2\pi} \int_{\gamma=0}^{2\pi} a_c^{(k)}(\gamma)\, \mathrm{d}\gamma \tag{3.1.26}$$

Finally, the signal from all carousels can be found by integrating over all possible orientations of $\alpha^{PAF,R}$ & $\beta^{PAF,R}$:

$$\langle S(t)\rangle \propto \frac{1}{4\pi} \iint\limits_c \langle S_c(t)\rangle \sin\beta\, \mathrm{d}\alpha \mathrm{d}\beta \tag{3.1.27}$$

$$\propto \sum_{k=-\infty}^{\infty} \left\langle a^{(k)} \right\rangle e^{i\left(\omega_c^{(0)} + k\omega_{MAS}\right)t} \tag{3.1.28}$$

$$\left\langle a^{(k)} \right\rangle = \frac{1}{4\pi} \iint\limits_c \left\langle a_c^{(k)} \right\rangle \sin\beta\, \mathrm{d}\alpha \mathrm{d}\beta \tag{3.1.29}$$

$\omega_c^{(0)}$ is the same across all carousels (assuming perfect β_{MAS}), so even spectra recorded at low MAS rates will have a peak at ω^{iso}. However, there will also be a set of so-called *spinning sidebands* at $\omega^{iso} + k\omega_{MAS}$ for all integer values of k. If only $\hat{I}_x$ magnetization is present at the beginning of the experiment, then $\phi_c\left(0;\gamma^{PAF,R}\right) = 0$ for all possible values of $\gamma^{PAF,R}$, and the $\left\langle a^{(k)} \right\rangle$ will be real and positive, meaning that the sidebands will be absorptive. The exact values of $\left\langle a^{(k)} \right\rangle$, however, are more difficult to predict and are generally integrated numerically. As a general rule, sidebands outside the "envelope" of the static MAS pattern tend to zero rapidly as k increases, but it is not safe to assume that the strongest peak in a MAS pattern is ω^{iso} if ω_{MAS} is small relative to the width of the static pattern.

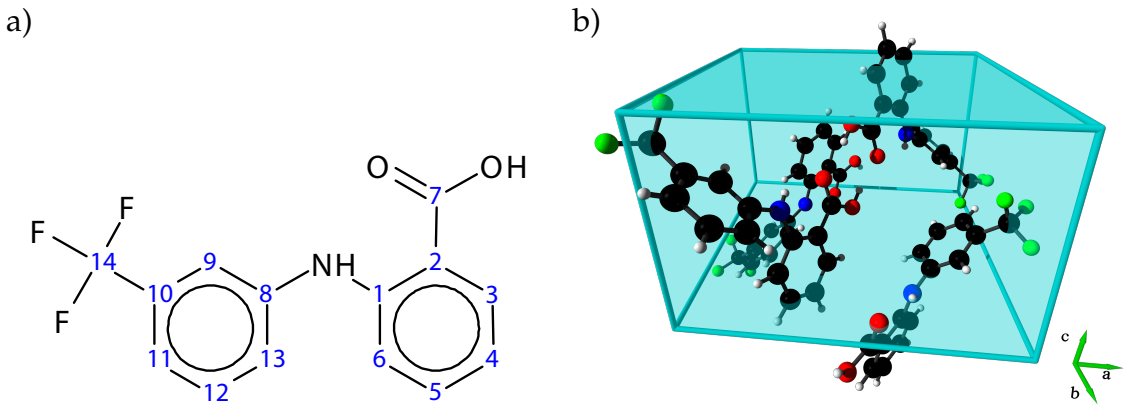


Figure 3.3: Molecular structure (a) and crystal structure (b) of flufenamic acid (Form I). Atom colours follow the CPK scheme: hydrogen is white, carbon is black, oxygen is red, nitrogen is blue and fluorine is green. The unit cell is in the $P2_1/c$ space group with $a = 12.523$ Å, $b = 7.868$ Å, $c = 12.874$ Å, $\alpha = 90°$, $\beta = 95.2°$ and $\gamma = 90°$ [61].

3.2 Assignment of Flufenamic Acid

Figure 3.3a shows the structure of 2-{[3-(Trifluoromethyl)phenyl]amino}benzoic acid, also known as flufenamic acid. The molecule strongly resembles diphenylamine, a highly symmetric molecule with only four distinct resonances in its ^{13}C spectrum. The addition of a carboxylic acid group and a trifluoromethyl group destroys this symmetry, however, meaning that many ^{13}C sites in flufenamic acid will have similar, but not identical shifts.

Flufenamic acid is known to have several polymorphic forms [62, 63]; we were provided with samples of Form I & Form III by Dr. J. Burley from the School of Pharmacy to confirm that his group had succeeded in producing two distinct polymorphs. The spectra of the two samples (recorded using *Variable Amplitude Cross-Polarization*, VACP) are shown in Figures 3.4a & 3.4b. Both spectra were recorded on a 300 MHz Varian spectrometer in a 4 mm HXY triple resonance MAS probe with $\nu_{MAS} = 12$ kHz and with 100 kHz XiX [64] proton decoupling. The contact time was long (10 ms) to ensure that all carbon sites receive similar signal enhancement, and the rescan delay was 60 s so that samples would be relaxed (proton T_1 (Form I) ≈ 15 s).

The apparent resolution of the two spectra is different; this may be due to many

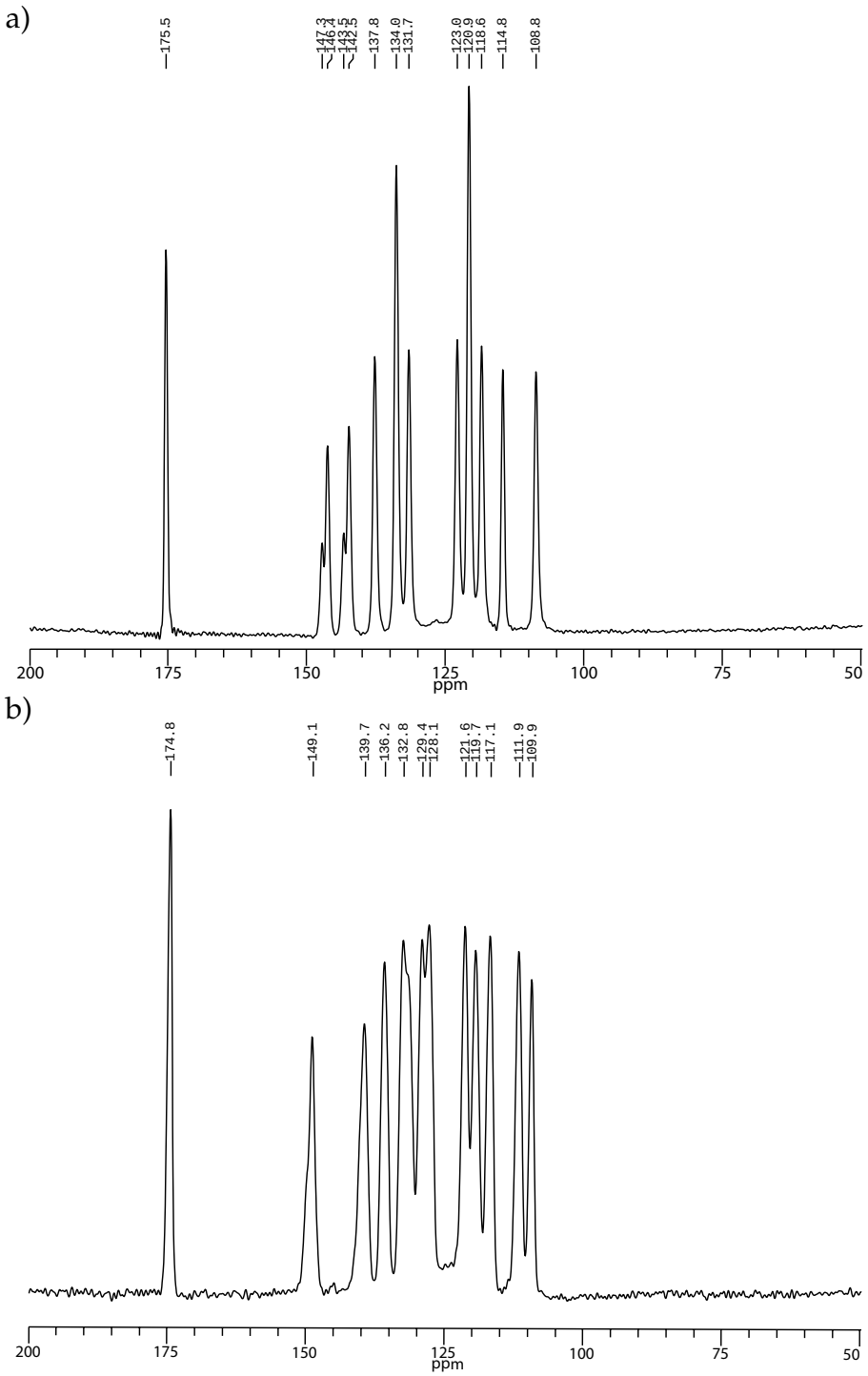


Figure 3.4: VACP ^{13}C spectra of flufenamic acid (a: Form I, b: Form III). Recorded on a 300 MHz Varian spectrometer in a 4 mm HXY triple resonance MAS probe at $\nu_{MAS} = 12$ kHz and with 100 kHz XIX [64] proton decoupling. The contact time was 10 ms and the rescan delay was 60 s.

sample specific factors such as grain size or crystallite quality, or may even be influenced by packing effects. From the small differences in the isotropic shifts, however, it is clear that the two samples are indeed different polymorphic forms of flufenamic acid.

With regards to the peak assignments, only Form I will be discussed as it has higher resolution. The analysis of Form III is completely analogous, except that the DFT calculations discussed later on in this chapter take longer as the crystal structure of Form III has twice as many molecules per unit cell as that of Form I.

The astute reader will notice first of all that the spectrum of Form I (see Figure 3.4a on page 64 for details) only has 13 peaks, yet flufenamic acid has 14 carbon sites. It is likely that the "missing" peak is that of the trifluoromethyl group; ^{19}F is 100 % abundant, and has a gyromagnetic ratio so close to ^{1}H ($\gamma_{^{19}\mathrm{F}} = 40.053\,\frac{\mathrm{MHz}}{\mathrm{T}}$, $\gamma_{^{1}\mathrm{H}} = 42.576\,\frac{\mathrm{MHz}}{\mathrm{T}}$) that only one can be decoupled at a time with the HXY probe used in these experiments. This means the signal is likely broadened by strong dipole-dipole interactions. Comparisons with trifluorobenzene in solution suggest that the likely isotropic frequency is around 124 ppm, which is close to the small bump on the baseline at 126 ppm.

Next, it is worth noting that the peaks are not all the same size. This is to be expected, at least to a degree; cross-polarization from ^{1}H to ^{13}C will not be equally efficient for all sites. With long cross-polarization times, however, they should at least be similar. Integration reveals that the peaks at 134 ppm and 120.9 ppm are almost twice the median intensity, and the group of four peaks from 140 ppm to 150 ppm are all unexpectedly small. Instead, it seems likely that the two peaks at 147.3 ppm and 146.4 ppm are actually one resonance, split into two; ditto for the peaks at 143.5 ppm and 142.5 ppm. This type of splitting is known to occur when a dipole-dipole coupling exist between a spin-half nucleus and a quadrupolar nucleus. If the quadrupole coupling constant C_Q is large enough that the ω_Q becomes an appreciable fraction of the Larmor frequency ω_0, then the quantization axis for the quadrupolar nucleus shifts slightly with the electronic field gradient and is not generally aligned with $\boldsymbol{B}_0$. This in turn means that the dipole-dipole interactions cannot be averaged perfectly, and the overall lineshapes become powder patterns.

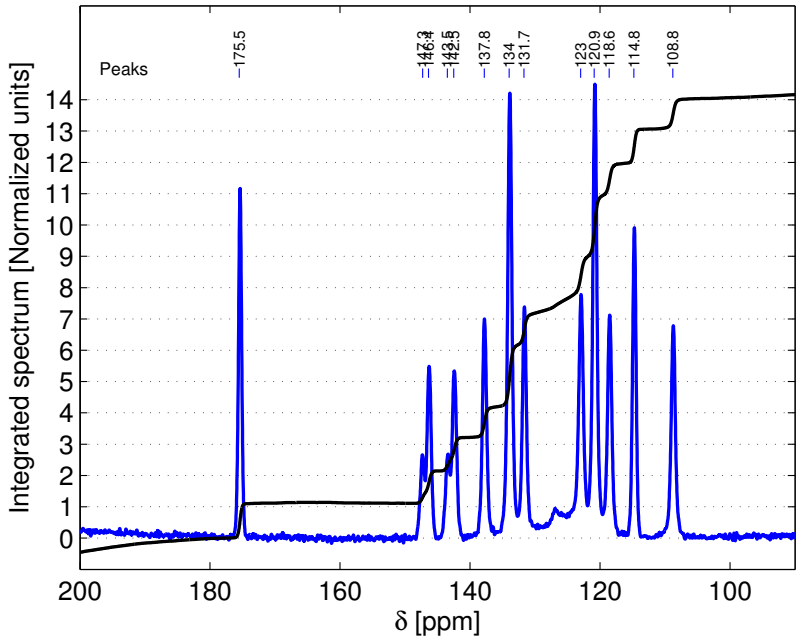


Figure 3.5: Integrated VACP spectrum of flufenamic acid, Form I.

More specifically, a ^{13}C nucleus coupled to ^{14}N will give three overlapping powder patterns since ^{14}N is spin 1. However, these patterns tend to be too narrow to resolve, so the carbon lineshape generally resembles an asymmetric doublet [65].

In summary, the peaks between 140 ppm and 150 ppm are likely from the two amine carbons, the peaks at 134 ppm and 120.9 ppm are probably each due to two carbon sites with coincident resonant frequencies, and the resonance at 175.5 ppm is, judging by its large shift, most likely linked to the carboxylic acid group.

Since one resonance has been positively identified, the next logical experiment would normally be a 2D-correlation experiment such as INADEQUATE [66,67] or SARCOSY [68]. But with a T_1 on the order of 15 s, such an experiment could take weeks at natural abundance on a 300 MHz spectrometer. Instead, CASTEP [69,70] was used to run *Density Functional Theory* (DFT) calculations on the crystal structure.

CASTEP is designed for the investigation of solid crystals rather than individual gas phase molecules, so it uses a plane wave basis set rather than the usual set of Gaussian atomic orbitals for its calculations of the electron density in a unit

cell. Unfortunately (from an NMR perspective), the calculations are accelerated by mostly calculating the density variations due to valence electrons; the core electrons (which are highly important to NMR as they are closest to the nucleus) are simulated using so-called *pseudo-potentials*. This difficulty is overcome by using *Gauge Including Projector Augmented Waves* (GIPAW) [70].

Crystal structures of both forms of flufenamic acid were obtained from Daresbury Laboratory's Chemical Database Service, and the geometries were then optimized in stages. Initially, only the positions of the hydrogen atoms were allowed to vary; then the positions of all atoms were optimized. This was done to save computation time: the available structures from the CDS were determined by XRD, meaning that the hydrogen positions were likely to require the greatest adjustments. Finally, the NMR parameters of the optimized structure were calculated in a separate run. It is important to note that CASTEP calculates $\boldsymbol{\sigma}_{\text{abs}}$, while the spectra in Figures 3.4a & 3.4b are referenced relative to TMS (using the known shifts of adamantane), so the isotropic chemical shift $\delta^{iso} = b - \boldsymbol{\sigma}_{\text{abs}}$ was determined by a least-squares-fit to the spectrum. Results are shown in Table 3.1.

The simplest way to assign the spectrum is then to assume that the calculated isotropic chemical shifts are at least in the correct order, even if the precise values do not match up (see Figure 3.6). This works acceptably well for the carboxylic acid group and the amine segment of the molecule, but there are problems for the others. CASTEP predicts a gap in the spectrum between 120 ppm and 130 ppm with five resonances toward lower ppm; the gap is indeed present in the real spectrum, but there are six resonances below it in ppm (assuming the peak at 120.9 ppm corresponds to two distinct ^{13}C sites). It is also noticeable that some of the resonances are almost 5 ppm away from their assigned peaks. Assuming that these offsets are indicative of the error margin in the calculations, then some of the peaks in the spectrum are still too close together to be reliably assigned by isotropic chemical shift alone.

Furthermore, an attempt was made to determine the frequencies of the quarternary carbons of flufenamic acid using the *solid-state Attached Proton Test* (ssAPT [71,72], typical data shown in Figure 3.7). The results were inconclusive due to the short T_2

	δ^{iso} [ppm]	$\zeta_{\boldsymbol{\delta}}$ [ppm]	$\eta_{\boldsymbol{\delta}}$
1	147.6	-111.3	0.40
2	108.0	-94.4	0.29
3	132.8	-130.1	0.61
4	116.8	-116.5	0.76
5	136.4	-133.8	0.64
6	110.4	-113.2	0.63
7	176.8	-71.7	0.82
8	139.2	-103.2	0.87
9	121.5	-109.5	0.61
10	131.4	-108.7	0.73
11	119.3	-125.5	0.58
12	129.1	-130.3	0.63
13	127.7	-125.1	0.56
14	140.0	-7.7	0.49

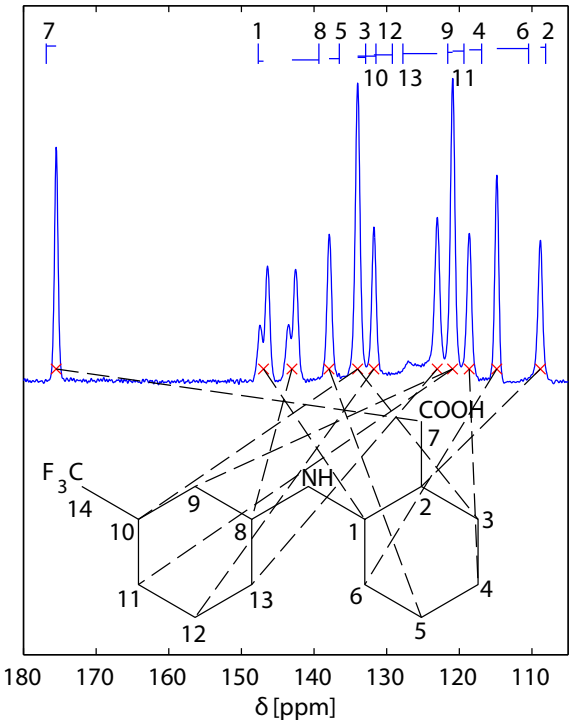
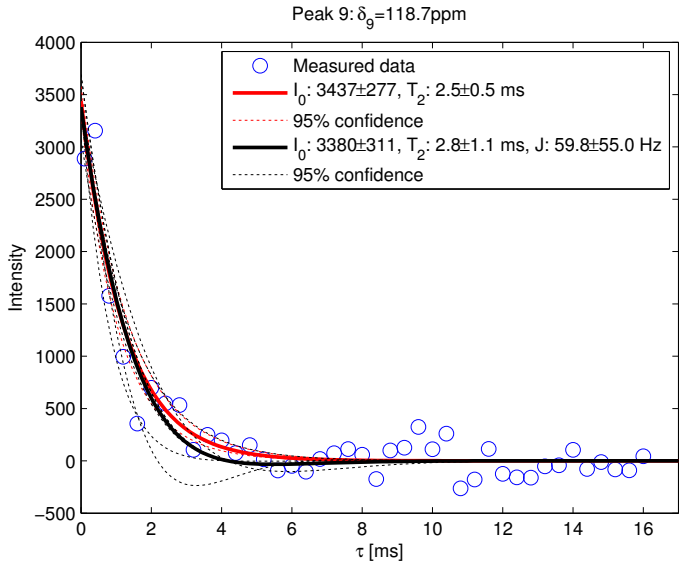


Figure 3.6 & Table 3.1: Chemical shift parameters[3] for flufenamic acid (Form I) as calculated by CASTEP. Assignments based on δ^{iso} are shown to the right.

times of flufenamic acid, but the J-coupling constants required to fit the data (see Table 3.2) suggest that the peaks at 134.0 ppm & 114.8 ppm come from quarternary carbons, meaning that ^{13}C atom 2 should be assigned to the peak at 114.8 ppm, not the one at 108.8 ppm.

It is clear that the isotropic chemical shifts calculated by CASTEP are not sufficiently accurate to unambiguously assign the spectrum of flufenamic acid. However, CASTEP calculates more than just the chemical shift anisotropies, it actually determines the full chemical shift tensor. The spectrum of flufenamic acid is far too congested to determine $\zeta_{\boldsymbol{\delta}}$ or $\eta_{\boldsymbol{\delta}}$ from a static experiment or a simple spectrum at slow MAS rates. They can, however, be measured with the 2D-PASS experiment [73]. 2D-PASS is a descendent both of Dixon's *TOtal Suppression of Sidebands*

[3]CASTEP actually displays the anisotropy slightly differently to the convention outlined on page 57. Instead, $\zeta_{\text{CASTEP}} = \sigma_{ZZ} - \frac{\sigma_{XX}+\sigma_{YY}}{2} = \frac{3}{2}\zeta_{\boldsymbol{\sigma}}$. The values shown in Table 3.1 have been adjusted to account for this.



δ^{iso} [ppm]	J-coupling [Hz]
175.5	28± 5
146.4	17±25
142.5	12±46
137.8	50±90
134.0	10± 0
131.7	100±30
123.0	56±50
120.9	87±36
118.6	60±55
114.8	14± 5
108.8	74±36

Figure 3.7 & Table 3.2: Figure 3.7 shows the integrated peak intensities for peak 9 (typical data set for 1 peak out of 11) from an ssAPT experiment against the evolution time τ. The data were recorded on a 4 mm probe at 12 kHz MAS with a 10 ms contact time, 100 kHz DUMBO decoupling during τ and 100 kHz XiX decoupling during acquisition. The spectral width was 50 kHz and the rescan delay was 30 s. The solid black line shows the best fit assuming that the site has no attached protons, and the red solid line shows the best fit assuming that the site is coupled to one proton (95 % confidence limits shown as dotted lines). Unfortunately, the two models are too similar for an unambiguous assignment. Table 3.2 shows the J-coupling strengths for all peaks in the spectrum assuming one proton is attached. The peaks from 175.5 ppm to 142.5 ppm, as well as the peaks at 134.0 ppm & 114.8 ppm, have particularly weak J-couplings, and are most likely actually quarternary.

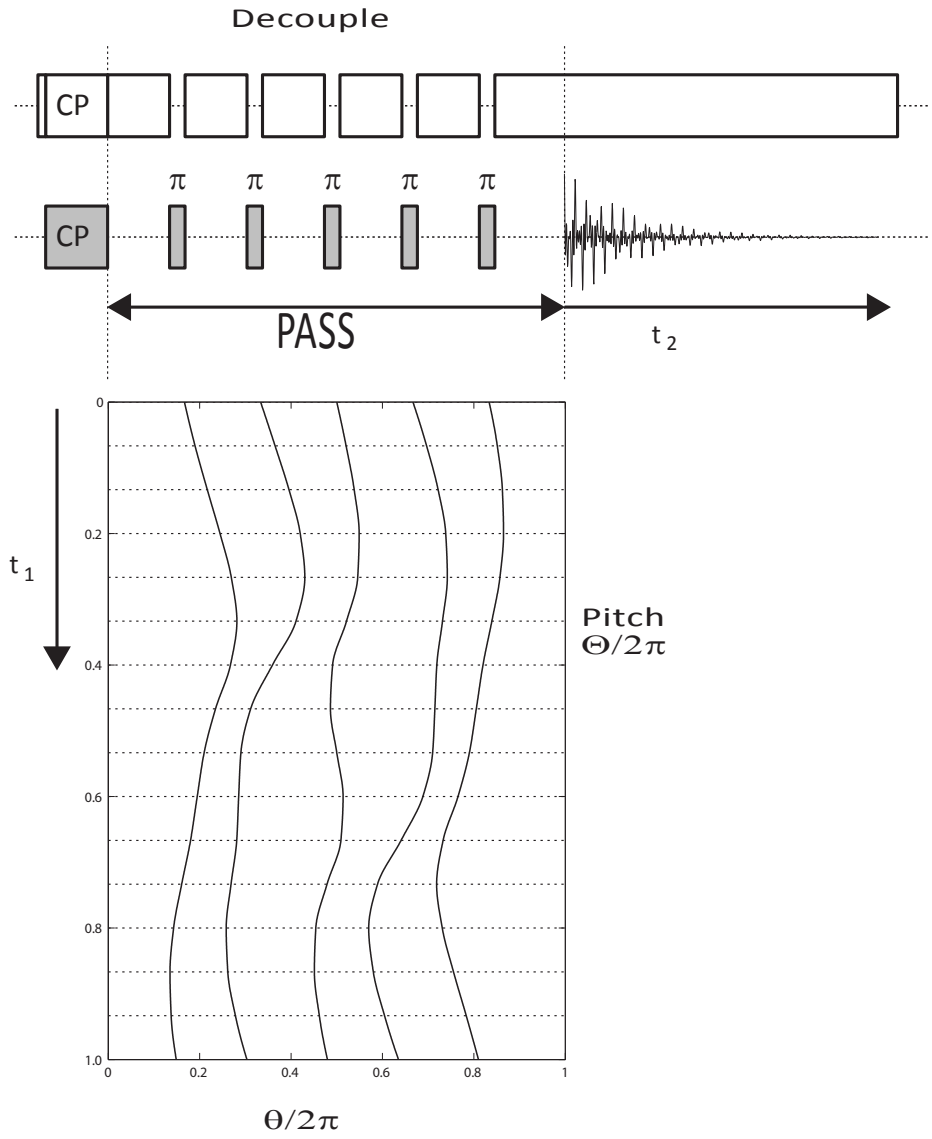


Figure 3.8: 2D-PASS pulse sequence, reproduced from page 9 of Ref. [73]. The horizontal axis is the time for each π pulse in fractions of the rotor period τ_{MAS}.

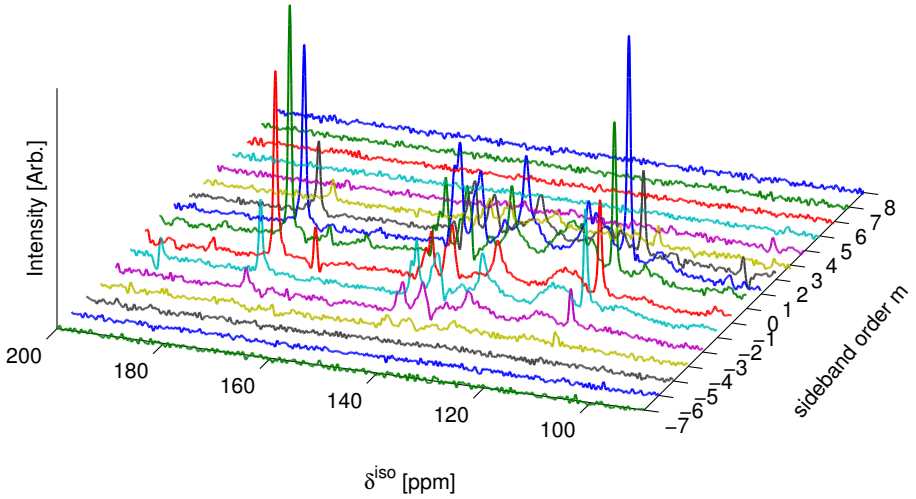


Figure 3.9: 2D-PASS spectrum of flufenamic acid (skewed by $-m\frac{\omega_{MAS}}{2\pi}$). Recorded on a 300 MHz spectrometer with a 10 ms contact time, a 50 kHz spectral width and an MAS rate of 2 kHz.

(TOSS) experiment [74], which suppresses all spinning sidebands except for the peaks at the isotropic frequency, and of the related *Phase Adjusted Spinning Sidebands* (PASS) experiment [73,75] which can select any order of spinning sideband.

The pulse sequence for Antzutkin's 2D-PASS experiment is shown in Figure 3.8. It begins with cross-polarization as usual, followed by five π pulses within one rotor period, and finishes with the acquisition of the FID. For the first increment, the π pulses are evenly spaced, resulting in a "normal" MAS spectrum. In subsequent increments, their positions are varied according to the diagram of Θ vs. θ, which modifies $\phi_c\left(0;\gamma^{PAF,R}\right)$. When the data is Fourier transformed in both dimensions, the central "strip" contains only the peaks at $\omega_c^{(0)} \propto \delta^{iso}$, and adjacent strips contain the spectra for spinning sidebands at frequencies with $\omega_c^{(m)} = \omega_c^{(0)} + m\omega_{MAS}$ (assuming perfect β_{MAS}). $\phi_c\left(0;\gamma^{PAF,R}\right)$ is chosen such that the intensities of the peak in sideband m is proportional to the Fourier coefficient $\left\langle a^{(m)} \right\rangle$ of a "normal" MAS spectrum, which can be simulated and fitted in the usual way.

Figure 3.9 shows the resulting spectrum of a 2D-PASS experiment at $\frac{\omega_{MAS}}{2\pi} = 2\,\text{kHz}$,

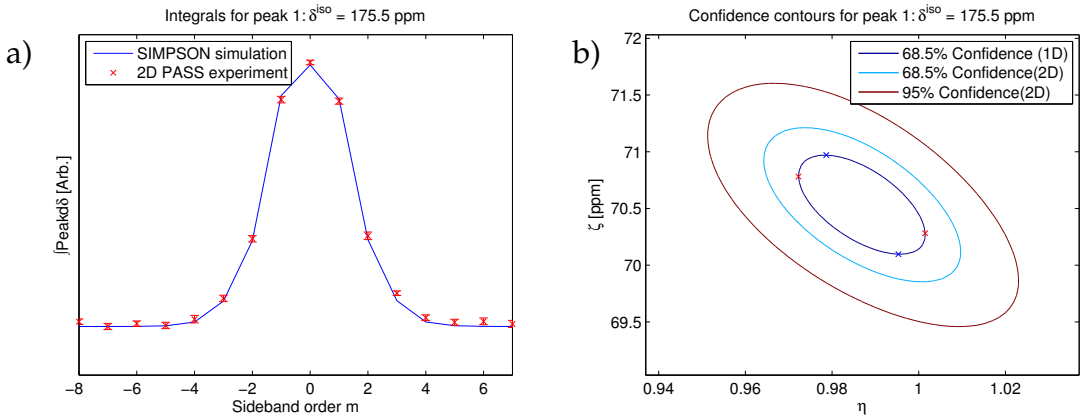


Figure 3.10: Fitting 2D-PASS data for peak 1 ($\delta^{iso} = 175.5\,\text{ppm}$). The best-fit simulated MAS pattern (blue line) is compared to the experimental MAS pattern (red crosses) in Plot a). Plot b) shows the 1σ and 2σ confidence limits for the fit.

though the data have been skewed such that the peaks for all sidebands line up at their isotropic frequencies. Each set of peaks is then fitted to a Voigt profile[4], and its intensity is fitted using SIMPSON simulations as shown in Figure 3.10a. The confidence limits can then be calculated according to Method II (see Section 2.2.4). The main difference here is that there are two parameters to be estimated (ζ&η) rather than just the dipole-dipole coupling b_{OH}, and hence two different ways to determine the confidence bounds on these parameters.

The χ^2 function can be reworked into a probability density function p. The simplest way to find confidence limits is to choose a contour c, with enclosed area A, such that $\int\int_A p\,\mathrm{d}\eta\mathrm{d}\zeta = \beta$ where β is the confidence limit (68.5 % for 1σ,95 % for 2σ). These are the light blue and dark red contours in Figure 3.10b. This method gives an excellent representation of both the confidence limits and correlation in ζ & η, however it is not well suited to representations of the form $a \pm \delta a$ since the shape of the contours is an integral part of their interpretation. In this case, we search for a region such that

$$\int_{\zeta=-\infty}^{\infty}\int_{\eta=a-\delta a}^{a+\delta a} p\,\mathrm{d}\eta\mathrm{d}\zeta = \beta \tag{3.2.1}$$

$$\int_{\zeta=b-\delta b}^{b+\delta b}\int_{\eta=-\infty}^{\infty} p\,\mathrm{d}\eta\mathrm{d}\zeta = \beta \tag{3.2.2}$$

[4] A convolution of a Gaussian and a Lorentzian peak.

	δ^{iso}[ppm]	$\zeta_{\boldsymbol{\delta}}$ [ppm]	$\eta_{\boldsymbol{\delta}}$
7	175.5	-70.5±0.4	0.99±0.01
1	146.9	-96.3±1.5	0.62±0.02
8	143.0	-100.9±2.6	0.81±0.02
5	137.8	-115.8±1.9	0.73±0.03
3	134.0	-105.6±1.1	0.71±0.01
10	134.0	-105.6±1.1	0.71±0.01
12	131.7	-113.4±1.8	0.77±0.02
9	123.0	-74.2±0.8	0.58±0.03
11	120.9	-89.5±1.9	0.66±0.02
13	120.9	-89.5±1.9	0.66±0.02
4	118.6	-104.3±2.0	0.73±0.03
2	114.8	-92.4±0.6	0.54±0.01
6	108.8	-86.1±3.8	0.56±0.09

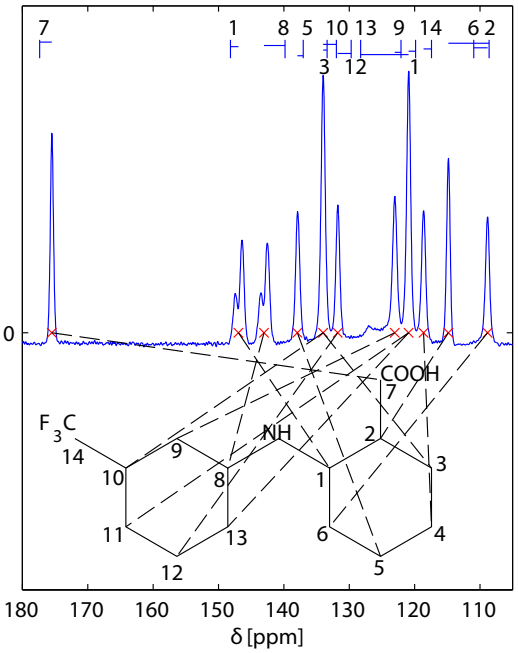


Figure 3.11 & Table 3.3: Table 3.3 shows the experimentally determined CSA parameters for flufenamic acid (Form I), as well as their assignments (first column). Figure 3.11 shows the assignment pictorially. The assignments of sites 2 and 6 have been exchanged compared to the assignments in Figure 3.6, as have those of sites 9 and 13.

It can be shown [76] that the contour line at $\chi^2_{\text{min}} + C_1^{\beta}$ (dark blue) fulfills both equations when $a \pm \delta a$ and $b \pm \delta b$ are chosen to be its extremal points along the ζ and η axes (red and blue crosses in Figure 3.10b). By this estimate, the CSA parameters for the carboxylic acid peak at 175.5 ppm are $\zeta = -70.5 \pm 0.4$ ppm and $\eta = 0.99 \pm 0.01$. The parameters for the remaining peaks are given in Table 3.3.

Even once all the CSA parameters have been calculated, there is still some ambiguity with regards to which peak should be assigned to which ^{13}C site, so an objective measure of how good a particular assignment is will be required. A least-squares approach should not be applied to δ^{iso}, ζ & η since they vary over completely different orders of magnitude. Instead, the principal axis values of the chemical shielding tensor were compared, i.e.:

$$\chi^2 = \left(\sigma_{XX}^{\text{PASS}} - \sigma_{XX}^{\text{CASTEP}}\right)^2 + \left(\sigma_{YY}^{\text{PASS}} - \sigma_{YY}^{\text{CASTEP}}\right)^2 + \left(\sigma_{ZZ}^{\text{PASS}} - \sigma_{ZZ}^{\text{CASTEP}}\right)^2 \quad (3.2.3)$$

Some small adjustments were made to ensure the "fairness" of the fit for each assignment tested: the PASS experiment measures $\boldsymbol{\sigma}_{rel}$ while CASTEP calculates $\boldsymbol{\sigma}_{abs} = \boldsymbol{\sigma}_{rel} - \sigma_{ref}\hat{\mathbf{1}}$, so σ_{ref} was fitted to produce the lowest χ^2 value, and $\zeta_{\boldsymbol{\sigma}}$ was scaled since CASTEP has been shown to overestimate ζ by about 10 % to 20 % in ^{13}C [77].

Simultaneously fitting σ_{ref} and the ζ scaling factor is not in and of itself a particularly slow process, however there are $13! = 6\,227\,020\,800$ unique assignments. Even if the known sites are grouped such that the amine carbons can only be assigned to two peaks, the quarternary carbons can only be assigned to two peaks and the carboxylic acid group assignment is fixed, there are still $8! \cdot 2! \cdot 2! \cdot 1! = 161\,280$ possible assignments. At a conservative estimate of 5 s per assignment, this would take 9 days. Instead, a steepest ascent hill climbing algorithm was used to find the best assignment. First, χ^2 was calculated for a random assignment. Then, a pair of assignments were swapped, and χ^2 was calculated for this new candidate assignment. Once all possible pairwise swaps had been checked, the candidate with the lowest χ^2 value was accepted as the current best assignment, and the algorithm was repeated until no better value of χ^2 could be found.

It is possible for hill climbing algorithms of this type to get stuck in local minima, so the program was run multiple times. The most common final assignment had the lowest energy, though some local minima were occasionally found. The best assignment is shown in Figure 3.11.

3.3 Conclusions

In conclusion, the isotropic chemical shift is not always sufficient to unambiguously assign the ^{13}C spectrum of a given molecule. Even if the longitudinal relaxation time is such that 2-dimensional correlation methods are not practical at natural abundance, experiments to measure the CSA, such as 2D-PASS, may still be viable as they can be performed at natural abundance. The results of such measurements can then be matched against the principal values of the chemical shielding tensor as calculated by DFT, giving a more reliable spectral assignment. This

procedure has been successfully carried out for Form I of the polymorphic anti-inflamatory drug flufenamic acid; the results are shown in Figure 3.11 on page 73.

4 Summary and Outlook

This thesis is concerned with new pulse sequences for measuring dipole-dipole coupling strengths, as well as a systematic approach to assigning spectra by measuring CSA parameters.

57 R-sequences which recouple only the dipole-dipole interaction have been simulated in combination with seven different basic elements $\mathscr{R}$ to determine which would be most effective in both a selective and a non-selective heteronuclear dipolar recoupling experiment. The best have been tested experimentally by measuring the H^{η}–O^{η} and H''–O^{η} distances in a sample of L-tyrosine.HCl labelled to 20 % with ^{17}O at the O^{η} position. Since a smaller number of points were recorded, a repetition of the original $SR4_1^2$ experiment by Brinkmann and Kentgens yielded error bars approximately 6 times wider than those cited by Brinkmann. It is demonstrated that R-sequences do not need to be supercycled as they are in Brinkmann's experiments, and it is found that the damping function used for the analysis should be simplified in such cases. Compared to the $SR4_1^2$ sequence, an $R10_3^9$ experiment with a BB1(180) basic element gives a two-fold reduction in the width of the 95 % confidence limit. An almost three-fold reduction in confidence limit width can be achieved by omitting the damping function entirely, and performing the simulations used to analyse the data with B_1 inhomogeneity instead. The size of the B_1 inhomogeneity required for a good fit is unexpectedly high, most likely because the B_1 inhomogeneity profile used was very simple. It is likely that a more thorough investigation of the B_1 inhomogeneity profile of the probes used would improve the simulations used for fitting, and hence further improve the experimentally obtained results for the dipole-dipole coupling. It might also be interesting to repeat the experiments on probes with less B_1 inhomogeneity, to see if good fits can be obtained without the need for damping functions or B_1 inhomogeneity simulations.

With regards to spectral assignments, it is shown that CSA parameters can be used to find an assignment even when the spectrum is too congested for DFT calculations of the isotropic chemical shift alone. This is demonstrated for the anti-inflamatory drug flufenamic acid, which has a long T_1 time and is hence not

suitable for correlation methods at natural abundance. The CSA parameters were measured by a 2D-PASS experiment, and then compared to CASTEP DFT calculations, using a steepest ascent hill climbing algorithm to find the best match for the chemical shift tensor in the principal axis frame. The final assignment is shown in Figure 3.11 on page 73. The reason for using CASTEP is that it uses a plane-wave basis set, i.e. it takes account of the translational symmetry of the crystal's unit cell. However, it has the disadvantage that pseudopotentials are used to describe the core electrons, which are closest to the nucleus and hence very important to NMR chemical shifts. A possible way to improve on the current method might, therefore, be to perform only the geometry optimisation with CASTEP, and to then calculate the chemical shift tensors using Gaussian basis sets. Such a calculation would be expensive, of course, as the Gaussian calculation would need to be performed on a system containing multiple unit cells in order to realistically approximate the electronic structure in a bulk crystal for at least one set of molecules.

References

[1] W. Gerlach and O. Stern. Der experimentelle Nachweis des magnetischen Moments des Silberatoms. *Zeitschrift für Physik*, 8(1):110–111, 1922.

[2] T. E. Phipps and O. Stern. Über die Einstellung der Richtungsquantelung. *Zeitschrift für Physik*, 73:185–191, 1932. DOI: 10.1007/BF01351212.

[3] I. I. Rabi. On the process of space quantization. *Physical Review*, 49:324–328, Feb 1936. DOI: 10.1103/PhysRev.49.324.

[4] I. I. Rabi. Space quantization in a gyrating magnetic field. *Physical Review*, 51:652–654, Apr 1937. DOI: 10.1103/PhysRev.51.652.

[5] I. I. Rabi, J. R. Zacharias, S. Millman, and P. Kusch. A new method of measuring nuclear magnetic moment. *Physical Review*, 53:318–318, Feb 1938. DOI: 10.1103/PhysRev.53.318.

[6] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review*, 69:37–38, Jan 1946. DOI: 10.1103/PhysRev.69.37.

[7] F. Bloch, W. W. Hansen, and M. Packard. Nuclear induction. *Physical Review*, 69:127–127, Feb 1946. DOI: 10.1103/PhysRev.69.127.

[8] F. Bloch. Nuclear induction. *Physical Review*, 70:460–474, Oct 1946. DOI: 10.1103/PhysRev.70.460.

[9] R. R. Ernst and W. A. Anderson. Application of fourier transform spectroscopy to magnetic resonance. *Review of Scientific Instruments*, 37(1):93–102, 1966. DOI: 10.1063/1.1719961.

[10] L. Müller, A. Kumar, and R. R. Ernst. Two-dimensional carbon-13 NMR spectroscopy. *The Journal of Chemical Physics*, 63(12):5490–5491, December 1975. DOI: 10.1063/1.431284.

[11] W. P. Aue, E. Bartholdi, and R. R. Ernst. Two-dimensional spectroscopy. Application to nuclear magnetic resonance. *The Journal of Chemical Physics*, 64(5):2229–2246, March 1976. DOI: 10.1063/1.432450.

[12] A. Kumar, G. Wagner, R. R. Ernst, and K. Wüthrich. Studies of J-connectivities and selective ^{1}H–^{1}H Overhauser effects in H_2O solutions of biological macromolecules by two-dimensional NMR experiments. *Biochemical and Biophysical Research Communications*, 96(3):1156–1163, 1980. DOI: 10.1016/0006-291X(80)90073-X.

[13] K. Nagayama and K. Wüthrich. Systematic application of two-dimensional ^{1}H nuclear-magnetic-resonance techniques for studies of proteins. *European Journal of Biochemistry*, 114(2):365–374, 1981. DOI: 10.1111/j.1432-1033.1981.tb05156.x.

[14] N. Zettili. *Quantum Mechanics: Concepts and Applications*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, England, 2nd edition, 2009. ISBN 978-0-470-02679-3.

[15] R. R. Ernst, G. Bodenhausen, and A. Wokaun. *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, volume 14 of *The International Series of Monographs on Chemistry*. Oxford University Press, Great Clarendon Dtreet, Oxford OX2 6DP, 2004. ISBN 0-19-855647-0.

[16] M. H. Levitt. *Spin dynamics*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, England, 2nd edition, 2008. ISBN 978-0-470-51117-6.

[17] M. G. Ricci. Résumé de quelques travaux sur les systèmes variables de fonctions associés à une forme différentielle quadratique. *Bulletin des Sciences Mathématiques 2*, 16:167–189, 1892. ARK: /13960/t0gt5mh0g.

[18] M. G. Ricci and T. Levi-Civita. Méthodes de calcul différentiel absolu et leurs applications. *Mathematische Annalen*, 54(1–2):125–201, March 1900. DOI: 10.1007/BF01454201.

[19] M. E. Rose. *Elementary Theory of Angular Momentum*. Structure of Matter Series. John Wiley & Sons, Inc., New York, New York, 1957. LCCN: 57008893.

[20] V. Devanathan. *Angular momentum techniques in quantum mechanics*, volume 108 of *Fundamental theories of Physics*. Kluwer Academic Publishers, New York / Boston / Dordrecht / London / Moscow, 1999. eISBN 9780792358664 / eBook ISBN 0-306-47123-X.

[21] M. Mehring. *Principles of High Resolution NMR in Solids*, volume 11 of *NMR-Basic Principles and Progress*. Springer-Verlag, New York Heidelberg Berlin, 2nd edition, 1983. ISBN 0-387-11852-7.

[22] M. J. Duer. *Introduction to Solid-State NMR Spectroscopy*. Blackwell Publishing Ltd., Oxford, 2004. ISBN 1-4051-0914-9.

[23] D. E. Kaplan and E. L. Hahn. Expériences de double irradiation en résonance magnétique par la méthode d'impulsions. *Journal de Physique et le Radium*, 19(11):821–825, Nov 1958. DOI: 10.1051/jphysrad:019580019011082100.

[24] T. Gullion and J. Schaefer. Rotational-echo double-resonance NMR. *Journal of Magnetic Resonance (1969)*, 81(1):196–200, 1989. DOI: 10.1016/0022-2364(89)90280-1.

[25] A. W. Hing, S. Vega, and J. Schaefer. Transferred-echo double-resonance NMR. *Journal of Magnetic Resonance (1969)*, 96(1):205–209, 1992. DOI: 10.1016/0022-2364(92)90305-Q.

[26] E. R. H. van Eck, R. Janssen, W. E. J. R. Maas, and W. S. Veeman. A novel application of nuclear spin-echo double-resonance to aluminophosphates and aluminosilicates. *Chemical Physics Letters*, 174(5):428–432, 1990. DOI: 10.1016/S0009-2614(90)87174-P.

[27] T. Gullion. Detecting ^{13}C–^{17}O dipolar interactions by rotational-echo, adiabatic-passage, double-resonance NMR. *Journal of Magnetic Resonance, Series A*, 117(2):326–329, 1995. DOI: 10.1006/jmra.1995.0779.

[28] A. Medek, J. S. Harwood, and L. Frydman. Multiple-quantum magic-angle spinning NMR: A new method for the study of quadrupolar nuclei in solids. *Journal of the American Chemical Society*, 117(51):12779–12787, December 1995. DOI: 10.1021/ja00156a015.

[29] C. Fernandez, D. P. Lang, J. P. Amoureux, and M. Pruski. Measurement of heteronuclear dipolar interactions between quadrupolar and spin-1/2 nuclei in solids by multiple-quantum REDOR NMR. *Journal of the American Chemical Society*, 120(11):2672–2673, Mar 1998. DOI: 10.1021/ja973311r.

[30] M. Pruski, A. Bailly, D. P. Lang, J.-P. Amoureux, and C. Fernandez. Studies of heteronuclear dipolar interactions between spin-1/2 and quadrupolar nuclei by using redor during multiple quantum evolution. *Chemical Physics Letters*, 307(1–2):35–40, 1999. DOI: 10.1016/S0009-2614(99)00490-X.

[31] A. Brinkmann and M. H. Levitt. Symmetry principles in the nuclear magnetic resonance of spinning solids: Heteronuclear recoupling by generalized Hartmann–Hahn sequences. *Journal of Chemical Physics*, 115(1):357–384, Jul 2001. DOI: 10.1063/1.1377031.

[32] A. B. and A. P. M. Kentgens. Sensitivity enhancement and heteronuclear distance measurements in biological ^{17}O solid-state NMR. *Journal of Physical Chemistry B*, 110(32):16089–16101, June 2006. DOI: 10.1021/jp062809p.

[33] A. B. and A. P. M. Kentgens. Proton-selective ^{17}O–^{1}H distance measurements in fast magic-angle-spinning solid-state NMR spectroscopy for the determination of hydrogen bond lengths. *Journal of the American Chemical Society*, 128(46):14758–14759, July 2006. DOI: 10.1021/ja065415k.

[34] D. Iuga, H. Schäfer, R. Verhagen, and A. P. M. Kentgens. Population and coherence transfer induced by double frequency sweeps in half-integer quadrupolar spin systems. *Journal of Magnetic Resonance*, 147(2):192–209, 2000. DOI: 10.1006/jmre.2000.2192.

[35] S. Wimperis. Broadband, narrowband, and passband composite pulses for use in advanced NMR experiments. *Journal of Magnetic Resonance, Series A*, 109(2):221–231, 1994. DOI: 10.1006/jmra.1994.1159.

[36] R. Tycko. Broadband population inversion. *Physical Review Letters*, 51:775–777, Aug 1983. DOI: 10.1103/PhysRevLett.51.775.

[37] M. H. Levitt. Symmetry-based pulse sequences in magic-angle spinning solid-state NMR. In David M. Grant and Robin K. Harris, editors, *Encyclopedia of Nuclear Magnetic Resonance*, volume 9, pages 165–196. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, England, 2002. ISBN 0471 49082 2.

[38] Y. K. Lee, N. D. Kurur, M. Helmle, O. G. Johannessen, N. C. Nielsen, and M. H. Levitt. Efficient dipolar recoupling in the NMR of rotating solids. A sevenfold symmetric radiofrequency pulse sequence. *Chemical Physics Letters*, 242(3):304–309, March 1995. DOI: 10.1016/0009-2614(95)00741-L.

[39] A. E. Bennett, R. G. Griffin, J. H. Ok, and S. Vega. Chemical shift correlation spectroscopy in rotating solids: Radio frequency-driven dipolar recoupling and longitudinal exchange. *Journal of Chemical Physics*, 96(11):8624–8627, March 1992. DOI: 10.1063/1.462267.

[40] D. K. Sodickson, M. H. Levitt, S. Vega, and R. G. Griffin. Broad band dipolar recoupling in the nuclear magnetic resonance of rotating solids. *Journal of Chemical Physics*, 98(9):6742–6748, May 1993. DOI: 10.1063/1.462267.

[41] K. Wüthrich. *NMR of Proteins and Nucleic Acids*. Wiley, New York, 1986.

[42] M. Baldus and B. H. Meier. Total correlation spectroscopy in the solid state. the use of scalar couplings to determine the through-bond connectivity. *Journal of Magnetic Resonance, Series A*, 121(1):65–69, 1996. DOI: 10.1006/jmra.1996.0137.

[43] L. Braunschweiler and R. R. Ernst. Coherence transfer by isotropic mixing: Application to proton correlation spectroscopy. *Journal of Magnetic Resonance (1969)*, 53(3):521–528, 1983. DOI: 10.1016/0022-2364(83)90226-3.

[44] M. H. Levitt and R. Freeman. NMR population inversion using a composite pulse. *Journal of Magnetic Resonance (1969)*, 33(2):473–476, 1979. DOI: 10.1016/0022-2364(79)90265-8.

[45] M. H Levitt and R. Freeman. Compensation for pulse imperfections in NMR spin-echo experiments. *Journal of Magnetic Resonance (1969)*, 43(1):65–80, 1981. DOI: 10.1016/0022-2364(81)90082-2.

[46] S. Wimperis. Iterative schemes for phase-distortionless composite 180° pulses. *Journal of Magnetic Resonance (1969)*, 93(1):199–206, 1991. DOI: 10.1016/0022-2364(91)90043-S.

[47] M. Bak, J. T. Rasmussen, and N. C. Nielsen. SIMPSON: A general simulation program for solid-state NMR spectroscopy. *Journal of Magnetic Resonance*, 147(2):296–330, December 2000. DOI: 10.1006/jmre.2000.2179.

[48] M. Bak and N. C. Nielsen. REPULSION, a novel approach to efficient powder averaging in solid-state NMR. *Journal of Magnetic Resonance*, 125(1):132–139, 1997. DOI: 10.1006/jmre.1996.1087.

[49] D. A. Ratkowsky. *Nonlinear Regression Modeling*. MARCEL DEKKER, INC., New York, New York, 1983. ISBN 0-8247-1907-7.

[50] F. James. *MINUIT, Function Minimization and Error Analysis, Reference Manual*. CERN, CERN Program Library Office, CERN-IT Division, CH-1211 Geneva 23, Switzerland, Aug 1998. Version 94.1, CERN Program Library entry D506.

[51] F. James. *Statistical Methods in Experimental Physics*. World Scientific Publishing Co. Pte. Ltd., 5 Toh Tuck Link Singapore 596224, 2nd edition, 2006.

[52] M. N. Frey, T. F. Koetzle, M. S. Lehmann, and W. C. Hamilton. Precision neutron diffraction structure determination of protein and nucleic acid components. x. a comparison between the crystal and molecular structures of L-tyrosine and L-tyrosine hydrochloride. *Journal of Chemical Physics*, 58(6):2547–2556, March 1973. DOI: 10.1063/1.1679537.

[53] D. Sakellariou, A. Lesage, P. Hodgkinson, and L. Emsley. Homonuclear dipolar decoupling in solid-state NMR using continuous phase modulation. *Chemical Physics Letters*, 319(3–4):253–260, March 2000. DOI: 10.1016/S0009-2614(00)00127-5.

[54] Y. Ishii, T. Terao, and S. Hayashi. Theory and simulation of vibrational effects on structural measurements by solid-state nuclear magnetic resonance. *Journal of Chemical Physics*, 107(8):2760–2774, 1997. DOI: 10.1063/1.474633.

[55] D. A. Case. Calculations of NMR dipolar coupling strengths in model peptides. *Journal of Biomolecular NMR*, 15:95–102, 1999. DOI: 10.1023/A:1008349812613.

[56] D. H. Brouwer and J. A. Ripmeester. Symmetry-based recoupling of proton chemical shift anisotropies in ultrahigh-field solid-state NMR. *Journal of Magnetic Resonance*, 185(1):173–178, 2007. DOI: 10.1016/j.jmr.2006.12.003.

[57] E. Vinogradov, P. K. Madhu, and S. Vega. High-resolution proton solid-state NMR spectroscopy by phase-modulated Lee-Goldburg experiment. *Chemical Physics Letters*, 314(5–6):443–450, December 1999. DOI: 10.1016/S0009-2614(99)01174-4.

[58] E. Vinogradov, P. K. Madhu, and S. Vega. Proton spectroscopy in solid state nuclear magnetic resonance with windowed phase modulated Lee–Goldburg decoupling sequences. *Chemical Physics Letters*, 354(3–4):193–202, 2002. DOI: 10.1016/S0009-2614(02)00060-X.

[59] H. W. Spiess. *Rotation of Molecules and Nuclear Spin Relaxation*, volume 15 of *NMR-Basic Principles and Progress*. Springer-Verlag, New York Heidelberg Berlin, 1978. ISBN 3-540-08784-2, ISBN 0-387-08784-2.

[60] O. N. Antzutkin, Z. Song, X. Feng, and M. H. Levitt. Suppression of sidebands in magic-angle-spinning nuclear magnetic resonance: General principles and analytical solutions. *The Journal of Chemical Physics*, 100(1):130–140, 1994. DOI: 10.1063/1.466983.

[61] H. M. Krishna Murthy, T. N. Bhat, and M. Vijayan. Structure of a new crystal form of 2-{[3-(trifluoromethyl)phenyl]amino}benzoic acid (flufenamic acid). *Acta Crystallographica Section B*, 38(1):315–317, Jan 1982. DOI: 10.1107/S0567740882002763.

[62] M. Kuhnert-Brandstätter, L. Borka, and G. Friedrich-Sander. Zur Polymorphie von Arzneimitteln: Flufenaminsäure und BL 191. *Archiv der Pharmazie*, 307(11):845–853, 1974. DOI: 10.1002/ardp.19743071107.

[63] H. M. Krishna Murthy, T. N. Bhat, and M. Vijayan. Structure of a new crystal form of 2-[3-(trifluoromethyl)phenyl]aminobenzoic acid (flufenamic acid). *Acta Crystallographica Section B*, 38(1):315–317, Jan 1982. DOI: 10.1107/S0567740882002763.

[64] A. Detken, E. H. Hardy, M. Ernst, and B. H. Meier. Simple and efficient decoupling in magic-angle spinning solid-state NMR: the XiX scheme. *Chemical Physics Letters*, 356(3–4):298–304, 2002. DOI: 10.1016/S0009-2614(02)00335-4.

[65] J. G. Hexem, M. H. Frey, and S. J. Opella. Molecular and structural information from ^{14}N–^{13}C dipolar couplings manifested in high resolution ^{13}C NMR spectra of solids. *Journal of Chemical Physics*, 77:3847–3856, 1982. DOI: 10.1063/1.444338.

[66] A. Bax, R. Freeman, and S. P. Kempsell. Natural abundance ^{13}C–^{13}C coupling observed via double-quantum coherence. *Journal of the American Chemical Society*, 102:4849–4851, 1980. DOI: 10.1021/ja00534a056.

[67] A. Bax, R. Freeman, T. A. Frenkiel, and M. H. Levitt. Assignment of carbon-13 NMR spectra via double-quantum coherence. *Journal of Magnetic Resonance (1969)*, 43(3):478–483, 1981. DOI: 10.1016/0022-2364(81)90060-3.

[68] D. Lee, J. Struppe, D. W. Elliott, L. J. Mueller, and J. J. Titman. Sensitive absorptive refocused scalar correlation NMR spectroscopy in solids. *Physical Chemistry and Chemical Physics*, 11:3547–3553, 2009. DOI: 10.1039/b818867j.

[69] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. J. Probert, K. Refson, and M. C. Payne. First principles methods using CASTEP. *Zeitschrift für Kristallographie - Crystalline Materials*, 220(5–6):567–570, 2005. DOI: 10.1524/zkri.220.5.567.65075.

[70] J. R. Yates, C. J. Pickard, and F. Mauri. Calculation of NMR chemical shifts for extended systems using ultrasoft pseudopotentials. *Physical Review B*, 76(2):024401, July 2007. DOI: 10.1103/PhysRevB.76.024401.

[71] J. N. Shoolery. Recent developments in ^{13}C- and proton-NMR. *Journal of Natural Products*, 47(2):226–259, March–April 1984. DOI: 10.1021/np50032a003.

[72] A. Lesage, S. Steuernagel, and L. Emsley. Carbon-13 spectral editing in solid-state NMR using heteronuclear scalar couplings. *Journal of the American Chemical Society*, 120(28):7095–7100, July 1998. DOI: 10.1021/ja981019t.

[73] O. N. Antzutkin, S. C. Shekar, and M. H. Levitt. Two-dimensional sideband separation in magic-angle-spinning NMR. *Journal of Magnetic Resonance, Series A*, 115(1):7–19, 1995. DOI: 10.1006/jmra.1995.1142.

[74] W. T. Dixon, J. Schaefer, M. D. Sefcik, E. O. Stejskal, and R. A. McKay. Total suppression of sidebands in CPMAS C-13 NMR. *Journal of Magnetic Resonance (1969)*, 49(2):341–345, 1982. DOI: 10.1016/0022-2364(82)90199-8.

[75] W. T. Dixon. Spinning-sideband-free and spinning-sideband-only NMR spectra in spinning samples. *Journal of Chemical Physics*, 77(4):1800–1809, Aug 1982. DOI: 10.1063/1.444076.

[76] W. T. Eadie, D. Drijard, F. E. James, M. Roos, and B. Sadoulet. *Statistical Methods in Experimental Physics*. North-Holland Publishing Co., Amsterdam, 1st edition, 1971.

[77] L. Shao, J. R. Yates, and J. J. Titman. Carbon-13 chemical shift tensors of disaccharides: Measurement, computation and assignment. *The Journal of Physical Chemistry A*, 111(50):13126–13132, 2007. DOI: 10.1021/jp075921b.

A SIMPSON

SIMPSON [47] is a general purpose simulation program for solid-state NMR; version 3.1 is currently maintained by Zdenek Tosner, Rasmus Andersen, Niels Chr. Nielsen and Thomas Vosegaard. The input files are designed to resemble a "digital spectrometer": `spinsys` corresponds to the sample and contains the parameters for the spin-system, `par` contains spectrometer parameters such as the magnet's proton frequency or the MAS rate, `proc pulseq` corresponds to the pulse sequence, and `proc main` is used for general purpose functions such as processing or saving data. The simulations begin with the density matrix $\hat{\boldsymbol{\rho}}$ defined by `start_operator`, and then calculate propagators according to Equations 1.1.23 and 1.1.26 from page 9, with $\Delta t \leq$ `maxdt`. Expectation values of measurable quantities are recorded by the `acq` command using $\langle \hat{\boldsymbol{Q}} \rangle = \mathrm{Tr}\left\{\hat{\boldsymbol{\rho}}\hat{\boldsymbol{Q}}\right\}$ where $\hat{\boldsymbol{Q}}$ is given by `detect_operator`, e.g. $\hat{I}^+$ (`Inp`) for normal measurements. Powders are treated by repeating the simulation for a number of orientations given by `crystal_file` and averaging the results.

A.1 R-Sequences

This program simulates non-selective R-sequence experiments with various dipole-dipole coupling strengths and B_1 field inhomogeneities. The values here are typical for simulations performed at 600 MHz; for 850 MHz, the proton 90° pulse limit `max_p90I` is set to 1.0. Simulations to test the viability of a given R-sequence, such as those in Figure 2.9 on Page 32, were run with `DD_Start` = 14 270.0 and `DD_Stop` = 14 270.1; simulations for analysing experimental data were initially run with `DD_Start` = 10 000.0 and `DD_Stop` = 18 000.1, then expanded as required. All simulations were run with `d6` = 600.0 unless other values were required to conform with experiments, and all simulations were run with an assumed maximum MAS rate of 50 kHz, though the simulation adjusts the MAS rate on the fly if `max_p90I` is exceeded.

```
spinsys {
  channels 17O 1H
  nuclei 17O 1H
  shift 1 0 0 0 0 0 0
  quadrupole 1 2 4.412e6 0.6 0 0 0
  shift 2 0 0 0 0 0 0
  dipole 1 2 14270 0 0 0
}

par {
  spin_rate 50E3
  proton_frequency 600e6
  variable max_p90I 1.4
  crystal_file rep678
```

```
15  gamma_angles 5
16  start_operator I1x
17  detect_operator I1c
18
19  # Timings
20  # 1 Rotor period
21  variable T_r 1.0e6/spin_rate
22  # Refocusing time T/2
23  variable d6 600.0
24
25  # R-Sequence
26  variable N 4
27  variable n 3
28  variable v 0
29
30  verbose 1001
31
32  ############################
33  # R-block pulse parameters #
34  ############################
35  # length of 1 R-block
36  variable d21 n*T_r
37  # number of 90-degree pulse equivalents in basic element
38  variable num90 10.0
39  # length of 90 degree pulse
40  variable p90I d21/(num90*N)
41  # Simulate incorrect B1 field
42  variable MissCal 1.0
43  variable MissCalStart 0.3
44  variable MissCalStop  1.71
45  # Required nutation frequency
46  variable pl90I MissCal*1.e6/(4*p90I)
47
48  variable RSeqs int(round(d6/d21))
49  np RSeqs+1
50
51  ####################################
52  # S-Spin 180 degree pulse parameters #
53  ####################################
54  # Factor for central transition (I=3/2: 2, I=5/2: 3)
55  variable  C_fact  3
56  # Pulse length in rotor periods
57  variable  p180S T_r
58  # Central nutation frequency w^c_nut
59  variable  pl180S  1E6/(2*p180S)/C_fact
60
61  # phases
62  variable phi 180.0/N*v
63  sw 1.0e6/d21
64
65  # Dipole-dipole couplings to simulate
```

```
  variable DD_Start  10000.0
  variable DD_Stop   18000.1
  variable DD_Step    1000.0
}

proc pulseq {} {
  global par
  maxdt 1

  # RSequence time
  delay $par(d21)
  store 1
  reset

  # Echo time
  prop 1 [expr int(round($par(d6)/$par(d21)))-$par(RSeqs)]
  pulse $par(p180S) $par(pl180S) y 0 0
  prop 1 [expr int(round($par(d6)/$par(d21)))]
  store 3
  reset

  # RSequence
  for {set i 0} {$i<[expr $par(N)/2]} {incr i} {
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr  104.5+$par(phi)]
    pulse [expr 4*$par(p90I)] 0 0 $par(pl90I)  [expr  313.4+$par(phi)]
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr  104.5+$par(phi)]
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr    0  +$par(phi)]
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr -104.5-$par(phi)]
    pulse [expr 4*$par(p90I)] 0 0 $par(pl90I)  [expr -313.4-$par(phi)]
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr -104.5-$par(phi)]
    pulse [expr 2*$par(p90I)] 0 0 $par(pl90I)  [expr    0  -$par(phi)]
  }
  store 5
  reset

  # Density matrix for O17 & 1H after 90 selective excitation on 17O
  matrix set start list { \
    {  2.5  0    0    0    0    0    0    0    0    0    0    0  } \
    {  0    2.5  0    0    0    0    0    0    0    0    0    0  } \
    {  0    0    1.5  0    0    0    0    0    0    0    0    0  } \
    {  0    0    0    1.5  0    0    0    0    0    0    0    0  } \
    {  0    0    0    0    0    0   -0.5  0    0    0    0    0  } \
    {  0    0    0    0    0    0    0   -0.5  0    0    0    0  } \
    {  0    0    0    0   -0.5  0    0    0    0    0    0    0  } \
    {  0    0    0    0    0   -0.5  0    0    0    0    0    0  } \
    {  0    0    0    0    0    0    0    0   -1.5  0    0    0  } \
    {  0    0    0    0    0    0    0    0    0   -1.5  0    0  } \
    {  0    0    0    0    0    0    0    0    0    0   -2.5  0  } \
    {  0    0    0    0    0    0    0    0    0    0    0   -2.5} \
  }

```

```
117   # First point has no R-sequences
118   store 6
119   prop 1 $par(RSeqs)
120   prop 3
121   acq
122
123   # points with R-sequences
124   for {set i 1} {$i<=$par(RSeqs)} {incr i} {
125     reset
126     prop 6
127     prop 5
128     store 6
129     prop 1 [expr $par(RSeqs)-$i]
130     prop 3
131     acq
132   }
133 }
134
135 proc main {} {
136   global par
137   set ExtraCommands "null.tcl"
138   set par(name) "BB1"
139   set par(SaveDir) "data"
140   if [file exists $ExtraCommands] {
141     # Allows external modification of the program, e.g. by MATLAB
142     source $ExtraCommands
143   }
144   ############################
145   # R-block pulse parameters #
146   ############################
147   # length of 1 R-block
148   set par(d21) [expr $par(n)*$par(T_r)]
149   # length of 90 degree pulse (N.B. 4 p90s per composite pulse)
150   set par(p90I) [expr $par(d21)/($par(num90)*$par(N))]
151   if {$par(p90I) < $par(max_p90I)} {
152     # Recalculate paramters if 1H powers are too high at 50 kHz MAS
153     set par(p90I) $par(max_p90I)
154     set par(d21)  [expr $par(p90I)*$par(num90)*$par(N)]
155     set par(T_r) [expr $par(d21)/$par(n)]
156     set par(spin_rate) [expr 1.0e6/$par(T_r)]
157     if {$par(spin_rate) < 25e3} {return}
158   }
159   # Required nutation frequency
160   set par(pl90I) [expr $par(MissCal)*1.e6/(4.0*$par(p90I))]
161
162   set par(RSeqs) [expr int(round($par(d6)/$par(d21)))]
163   set par(np) [expr $par(RSeqs)+1]
164
165   ######################################
166   # S-Spin 180 degree pulse parameters #
167   ######################################
```

```
168   # Pulse length in rotor periods
169   set par(p180S)  $par(T_r)
170   # Central nutation frequency w^c_nut
171   set par(pl180S) [expr 1.0E6/(2*$par(p180S))/$par(C_fact)]
172
173   # phases
174   set par(phi) [expr 180.0/$par(N)*$par(v)]
175   set par(sw) [expr 1.0e6/$par(d21)]
176   # dipole-dipole coupling loop
177   for {set par(DD) $par(DD_Start)} {$par(DD) < $par(DD_Stop)} {set ...
          par(DD) [expr $par(DD)+$par(DD_Step)]} {
178     set par(rOH) [dip2dist 1H 17O $par(DD)]
179     # B1 inhomogeneity loop
180     for {set par(MissCal) $par(MissCalStart)} {$par(MissCal) < ...
          $par(MissCalStop)} {set par(MissCal) [expr ...
          $par(MissCal)+0.05]} {
181       set par(MAS_String) [format "%4.1f" [expr $par(spin_rate)/1.0e3]]
182       set par(MCal_String) [format "%4.2f" $par(MissCal)]
183       set par(file_name) "$par(name)_R$par(N)_$par(n)_$par(v)_ ...
            $par(MAS_String)kHz_$par(DD)Hz_$par(MCal_String)"
184       set par(pl90I) [expr $par(MissCal)*1.e6/(4.0*$par(p90I))]
185       # store important simulation parameters in a text file
186       set fileId [open [format "%s%s" $par(file_name) ".txt"] "w"]
187       puts $fileId "Magnet  [expr $par(proton_frequency)/1E6]MHz"
188       puts $fileId "MAS     [expr $par(spin_rate)/1E3]kHz"
189       puts $fileId "Tau_R   $par(T_r)us"
190       puts $fileId "Seq     R$par(N)_$par(n)^$par(v)BB1"
191       puts $fileId "cmp     180_{104.5}360_{313.4}180_{104.5}180_{0}"
192       puts $fileId "N       $par(N)"
193       puts $fileId "n       $par(n)"
194       puts $fileId "nu      $par(v)"
195       puts $fileId "Phi     $par(phi)"
196       puts $fileId "H_90    $par(p90I)us"
197       puts $fileId "H_nut   [expr $par(pl90I)/1e3]kHz"
198       puts $fileId "O_180   $par(p180S)us"
199       puts $fileId "O_nut   [expr $par(pl180S)/1e3]kHz"
200       puts $fileId "O_Cent  [expr $par(pl180S)/1e3*$par(C_fact)]kHz"
201       puts $fileId "Block   $par(d21)us"
202       puts $fileId "EchoT   $par(d6)us"
203       puts $fileId "B1      [expr $par(MissCal)*100]%"
204       puts $fileId "sw      [expr $par(sw)/1E3]kHz"
205       puts $fileId "Blocks  $par(RSeqs)"
206       puts $fileId "DipDip  $par(DD)"
207       puts $fileId "Dist    $par(rOH)"
208       puts $fileId "select  0"
209       close $fileId
210       set f [fsimpson { {dipole_1_2_aniso $par(DD)} } ]
211       fsave $f "$par(SaveDir)/$par(file_name).fid"
212     }
213   }
214 }
```

B MatLab

MATLAB (MATrix LABoratory) is an interactive numerical computing environment developed and maintained by MathWorks. Due to its versatility and the ease with which data can be displayed, all the data analysis and most of the figures presented in this thesis were done with MATLAB. While it would be impractical to list all the programs, two sets have been included here. Appendix B.1 contains the program used to identify valid R-sequences which form the basis for Chapter 2; it is included because it is central to the work. Appendices B.2 through B.6 are data processing tools for NMR experiments, using the 2D-PASS experiment used in Chapter 3 as an example. The files from B.3 were also used to analyse the recoupling experiments from Chapter 2, but the top level file was larger.

Analysing NMR experiments with MATLAB is not a new idea; for instance, "MatNMR" by Jacco van Beek is available for free at `http://matnmr.sourceforge.net`. While MatNMR has many fine points, including the ability to read many common data formats and perform common processing tasks, it is not suited to the analysis of data which requires unusual processing and it is designed to be used with a GUI, meaning that its functionality cannot easily be accessed by other MATLAB programs. In contrast, the files presented from Section B.2 onwards are designed with large scale batch processing in mind. The basic design philosophy uses an object oriented approach, creating an FID class which contains all parameters which can be retrieved from a given experiment and making child classes for particular experiment.

B.1 R-Sequence Finder

This program searches for R-sequences which recouple heteronuclear dipole-dipole interactions without recoupling homonuclear dipole-dipole couplings or the isotropic chemical shift. This is done by looping over all possible values of N, n and ν that a given RN_n^ν sequence might use. Since there are technically an infinite number of sequences to check, the program uses the properties of the probe, such as the maximum MAS rate, in order to determine when to stop (see Section 2.2.1 on page 26 for details). Viable R-sequences are stored in `gamma_encoded` if the are γ-encoded, or in `non_encoded` if they are not.

```
clear all
clc

% Probes: Nottingham 1.3 mm, Warwick 1.3 mm
v_R=[66.7 66.7];     % Maximum MAS rate for probe (kHz)
v_R_min=[30 35];     % Minimum MAS rate for probe (kHz)
pw=[2.8 2];          % Shortest possible 180 degree pulse for probe (us)
Max_Sequence_length=0.1; % Maximum viable sequence length (ms)
n_max=ceil(Max_Sequence_length*v_R);  % Maximum value of n

sol = 0;

gamma_encoded = cell(0,3+numel(pw));% Store gamma encoded solutions
non_encoded = gamma_encoded;        % Store other solutions

for n = 1:max(n_max)
  CanUse=n_max≥n;
  N_max=ceil(n./pw(CanUse)./max(v_R_min(CanUse), ...
      n/Max_Sequence_length*ones(1,length(CanUse)))*1E3);
  disp(['n: ' int2str(n) ' -> N_max = ' int2str(max(N_max))])
  for N=2:2:max(N_max)
    gam_nu=[];
    non_gam_nu=[];

    for v = 0:N/2
      if (gcd(gcd(N/2,v),n)>1)% Check for duplicate R-Sequences
        continue
      end
      good = false;
      bad = false;
      m_list=[];
      mu_list=[];

      %Isotropic chemical shift: l = 0, lambda = 1;
      if any(N/2:N:(v+N)==v)
        bad = true;
      end

```

```
38        % H DD coupling: l = 2, lambda = 2
39        for m = 1:2
40          for mu = -2:2
41            tmp=n*m-v*mu;
42            if mod(2/N*tmp,2)==0
43              bad = true;
44            end
45          end
46        end
47
48        % H CSA, O-H DD coupling: l = 2, lambda = 1
49        for m = 1:2
50          for mu = -1:1
51            tmp=n*m-v*mu;
52            if mod(2/N*tmp-1,2)==0
53              good = true;
54              m_list(end+1)=m;
55              mu_list(end+1)=mu;
56            end
57          end
58        end
59
60        % Add valid values of \nu to nu_list
61        if (good && ~bad)
62          % Check gamma encoding
63          if length(unique(m_list))==length(m_list) && ...
               length(unique(mu_list))==length(mu_list) && ...
               ~any(m_list==0) && ~any(mu_list==0)
64            gam_nu(end+1)=v;
65          else
66            non_gam_nu(end+1)=v;
67          end
68        end
69      end
70      % Store viable gamma-encoded solutions
71      if ~isempty(gam_nu)
72        gamma_encoded(end+1,1:3)={N n gam_nu};
73        for i = 1:numel(pw)
74          gamma_encoded{end,3+i}=n./(N*pw(i))*1000; % Max. MAS rate (kHz)
75        end
76      end
77      % Store other viable solutions
78      if ~isempty(non_gam_nu)
79        non_encoded(end+1,1:3)={N n non_gam_nu};
80        for i = 1:numel(pw)
81          non_encoded{end,3+i}=n./(N*pw(i))*1000; % Max. MAS rate (kHz)
82        end
83      end
84    end
85  end
```

B.2 2D-PASS class file

This file creates a class for 2D-PASS experiments and was used when analysing much of the data for Chapter 3. It introduces the property `m`, corresponding to the side-band order of a given section of the spectrum, as well as an arrayed version of `m` called `M` for use in plotting. All other methods and properties are inherited directly from the FID class, except the `MakeSpec` method for creating the spectrum from the raw data, which is overwritten in lines 30 to 37, and the `plot3` method for displaying the spectrum, which is overwritten in lines 14 to 18.

```
1  classdef PASS < FID
2    properties (Dependent = true, SetAccess = private)
3      m;
4      M;
5    end
6
7    methods
8      %% Constructor
9      function obj = PASS(RawFid,varargin)
10       obj = obj@FID(RawFid,varargin{:});
11       obj.dimensionality=2;
12     end
13     %% General functions
14     function plot3(obj,varargin)
15       plot3(obj.F{1}/1E3-obj.M*obj.MAS/1E3,obj.M,real(obj.spec))
16       xlabel('Frequency [kHz]')
17       ylabel('Sideband Order')
18     end
19     %% Get functions
20     function out = get.m(obj)
21       out = (obj.n(2):-1:1) - floor(obj.n(2)/2);
22     end
23     function out = get.M(obj)
24       out = repmat(obj.m,obj.zerofill(1),1);
25     end
26   end
27
28   methods (Hidden = true)
29     function a=MakeSpec(obj)
30       switch obj.dimensionality
31         case 2
32           a=exp(1i*pi*obj.phase_array/180).* ...
                 fftshift(fft2(obj.broaden,obj.zerofill(1),obj.zerofill(2)));
33         otherwise
34           error('PASS spectra are 2D!')
35       end
36     end
37   end
38 end
```

B.3 FID class file

The FID class file is intended as a starting point for all experiments, and hence contains properties and methods which are appropriate to most, if not all, NMR experiments. The properties `nscans` (the number of scans), `fid` (contains the raw data), `pd` (the rescan delay) and `spectrometer` (the name of the spectrometer) are assumed to be invariant, and are therefore included as private properties. Technically, the FID is actually stored in the hidden property `fid_storage`; this was done in case processing of the displayed FID (such as a left shift) should be necessary. There are a number of properties which can be changed at run time by the user listed in lines 14 to 34; some such as the spectrometer frequency `sf` should be obvious, but there are a few unusual ones as well. `dimensionality` is required so that the program can distinguish between multi-dimensional experiments and arrayed experiments. `multishot` is a property which was introduced in the context of *flip-back* experiments, which have not been discussed in this thesis. `lb` is a container for line broadening functions, which have their own class files (see Appendix B.4), and `acqpars` is intentionally unassigned so that all parameters of any experiment can be included as structured arrays, e.g. from SPINSIGHT's `acq` file or from Topspin's `acqus` and `acqu2s` files.

There are also several of properties associated with fitting. For instance, `baseline` and `baselinenoise` are intended for polynomial fits of the baseline of the experiment. They take a while to calculate, however, so they are left empty by default, and can be calculated by calling the `FindBaseline` method. The `CurveFitModel` and `CurveFitErrors` properties were intended for a built-in peak fitting tool which was never implemented. `Proc2D` is intended as a flag so that processing methods for common 2D data acquisition methods could be included; current options are `'FFT'` for a simple 2D Fourier transformation, `'states'` for states data and `'TPPI'` for data acquired using TPPI. Unfortunately, these methods have not all been thoroughly tested.

Also, there are a number of dependent properties which the FID class calculates internally. The time `t` for the FID is calculated from the dwell time `dw`, which is in turn calculated from the spectral width `sw`. `f` is also calculated from the spectral width, and `ppm` is calculated from `sw`, `sf` and `RefFreq`. `spec` contains the spectrum, and `spec_basecorr` contains the spectrum with baseline corrections (`FindBaseline` is called if necessary). `spec_basecorr` is automatically truncated, and `Istart` & `Istop` are the indices of the section which is displayed. There are also a number of hidden properties which are used internally, e.g. `phase1_Bruker`, an extra 1st order phase shift which is required due to Bruker's data acquisition.

Finally, there are a large number of methods associated with the FID class file. Most are set and get methods for various properties, ensuring that only "sensible" input is accepted. The set method for `RefFreq` can accept arrayed input where the first entry is the frequency and the second entry is the ppm value of that point.

Several MATLAB functions are overwritten, e.g. the `size` command will return the size of the spectrum and the `display` function shows only a minimal subset of the overall properties; this prevents MATLAB from writing vast quantities of information to the screen when creating spectra (`disp` has not been overwritten and will display all properties). The two most important remaining methods are `ShowLB` and `ShowPhasing`, which bring up the line broadening GUI (see B.6) and phasing GUI (see B.6) respectively.

```
classdef FID < handle
  %% Properties
  properties (SetAccess = private)
    nscans      % Number of scanse
    fid
    pd          % Rescan delay
    spectrometer = '';
  end

  properties (SetAccess = private,Hidden = true)
    fid_storage
  end

  properties
    dimensionality = 1;
    multishot = 1;
    lb;
    phase0=0;       % 0th order phase correction
    phase1=0;       % 1st order phase correction
    sw=NaN;         % Spectral width
    zerofill        % Zero filling
    acqpars         % Generic acquisition parameters
    MAS = NaN;      % MAS rate (Hz)
    sf=NaN;         % Spectrometer frequency (Hz)
    RefFreq=0;      % Reference frequency
    peaklist=[];    % List of known peaks (Hz)
    CurveFitModel={};
    CurveFitErrors={};
    baseline={};
    baselinenoise={};
    Proc2D = 'FFT';
    leftshift = 0;
    states_order = 'interleaved';
  end


  properties (Dependent = true, SetAccess = private)
    dw              % Dwell time
    phase_array     % Array of phase corrections
    f               % Frequency for 1D plots of spectrum
    F               % Frequency for 2D plots of spectrum
```

```
    ppm             % Chemical shift for 1D plots of spectrum
    PPM             % Chemical shift for 2D plots of spectrum
    t               % Time for 1D plots of FID
    T               % Time for 2D plots of FID
    n               % Size of raw data
    spec            % Final spectrum
    spec_basecorr   % Final spectrum with baseline correction
    broaden         % Array of linebroadening function
    Istart          % Index for start of spectrum
    Istop           % Index for end of spectrum
    noiseI          % Indices of points for baseline noise estimate
  end

  properties (Hidden = true)
    LBFigure   =[];
    PhaseFigure=[];
    FullSpecFigure=[];
    sw_hidden=NaN;
    phase0_hidden=0;
    phase1_hidden=0;
    phase1_Bruker=0;
    RefFreq_hidden=0;
    AssumeWidth=50;
    ExcludeSpinningsidebandsInNoise=false;
  end

  properties (Dependent = true, SetAccess = private,Hidden = true)
    PhaseUnarrayed;
    LBUnarrayed;
  end
  %% Events
  events
    ChangePhase   % Event for changes to phase0 \& phase1
  end
  %% Methods
  methods
    %% Constructor
    function obj = FID(RawFid,varargin)
      if numel(varargin) == 1
        options = varargin{1};
      else
        options = struct();
        for i = 1:numel(varargin)/2
          options.(varargin{2*i-1}) = varargin{2*i};
        end
      end

      if isstruct(RawFid)
        error('This functionality is not yet provided.')
      else
        obj.fid_storage = RawFid;
```

```
93          obj.fid = obj.fid_storage;
94          obj.lb=LBdata();
95          if isfield(options,'nscans')
96            obj.nscans = options.nscans;
97          end
98          if isfield(options,'pd')
99            obj.pd = options.pd;
100         end
101         if isfield(options,'spectrometer')
102           obj.spectrometer = options.spectrometer;
103         end
104       end
105       if size(obj.fid,1) == 1
106         obj.fid=obj.fid.';
107       end
108       obj.zerofill=size(obj.fid);
109     end
110     %% overload functions
111     function a=size(obj)
112       a=size(obj.fid);
113     end
114     function plot(obj,varargin)
115       if isempty(obj.dw)
116         plot(real(obj.spec),varargin{:})
117       else
118         plot(linspace(-obj.sw/2,obj.sw/2,obj.zerofill(1))/1E3, ...
              real(obj.spec),varargin{:})
119         xlabel('Frequency [kHz]')
120       end
121       set(gca,'xdir','reverse')
122     end
123     function display(obj)
124       if ~isempty(obj.spectrometer)
125         fprintf(1,'%s\n',obj.spectrometer)
126       end
127       if ~isempty(obj.nscans)
128         fprintf(1,'\t    # scans: %i\n',obj.nscans)
129       end
130       if ~isempty(obj.MAS) && ~isnan(obj.MAS)
131         fprintf(1,'\t        MAS: %.1f kHz\n',obj.MAS/1E3)
132       end
133       for i = 1:obj.dimensionality
134         fprintf(1,'\tDimension    %i\n',i)
135         fprintf(1,'\t          n: %i\n',obj.n(i))
136         fprintf(1,'\t         dw: %.1f us\n',obj.dw(i)*1E6)
137         fprintf(1,'\tZerofilling: %i\n',obj.zerofill(i))
138         fprintf(1,'\t         sw: %.1f kHz\n',obj.sw(i)/1E3)
139         fprintf(1,'\t         LB: %s\n',obj.lb(i).type)
140         if ~ismember(obj.lb(i).type,{'flat','hann','hamming','kaiser'})
141           fprintf(1,'\t      width: %.1f Hz\n',obj.lb(i).width)
142         end
```

```
143         fprintf(1,'\t      Phase: [%.1f ...
               %.1f]\n',obj.phase0(i),obj.phase1(i))
144       end
145       if obj.dimensionality == 2
146         fprintf(1,'\t     Proc2D: %s\n',obj.Proc2D)
147       end
148     end
149     %% General functions
150     % Plot only a selection of spectra
151     function ShowMeTheSpectra(obj,n,varargin)
152       if isempty(obj.dw) || isnan(obj.dw)
153         plot(real(obj.spec(:,n)),varargin{:})
154       else
155         plot(linspace(-obj.sw/2,obj.sw/2,obj.zerofill(1))/1E3, ...
               real(obj.spec(:,n)),varargin{:})
156         xlabel('Frequency [kHz]')
157       end
158       set(gca,'xdir','reverse')
159     end
160     % Rotate FID if it's the wrong way round
161     function rotate(obj)
162       if numel(size(obj.fid)) == 2
163         obj.fid = obj.fid.';
164       end
165     end
166     % bring up linebroadening GUI
167     function ShowLB(obj,varargin)
168       MyLineBroadeningGUI(obj,varargin{:})
169     end
170     function ShowPhasing(obj,varargin)
171       MyPhasingGUI(obj,varargin{:})
172     end
173     function FindBaseline(obj)
174       if isempty(obj.peaklist)
175         warning('No peaklist present, not fitting baseline.')
176       else
177         tmp = obj.spec(obj.noiseI,:);
178         for i = 1:obj.n(2)
179           obj.baseline{i} = ...
                polyfit(obj.f{1}(obj.noiseI),real(tmp(:,i)),2);
180           obj.baselinenoise{i} = std(real(tmp(:,i))- ...
                polyval(obj.baseline{i},obj.f{1}(obj.noiseI)));
181         end
182       end
183     end
184     function a=spec_1D(obj,n)
185       if nargin < 2
186         n = 1;
187       end
188       if n > obj.n(2)
189         n = obj.n(2);
```

```
190         warning(['There are only ' int2str(obj.n(2)) ' FIds.'])
191       elseif n<1
192         n = 1;
193         warning('Showing 1st FID.')
194       end
195       a = exp(-1i*obj.PhaseUnarrayed{1}/180*pi).*fftshift(fft( ...
              obj.lb(1).broaden(obj.fid(:,n)),obj.zerofill(1) ));
196     end
197     %% Get Methods
198     function a=get.spec(obj)
199       a=MakeSpec(obj);
200     end
201     function a=get.spec_basecorr(obj)
202       if isempty(obj.peaklist)
203         a = [];
204       else
205         obj.FindBaseline;
206         start = min(obj.noiseI);
207         stop  = max(obj.noiseI);
208         a = NaN(stop-start+1,obj.n(2));
209         for i = 1:obj.n(2)
210           a(:,i) = obj.spec(start:stop,i) - ...
                polyval(obj.baseline{i},obj.f{1}(start:stop));
211         end
212       end
213     end
214     function out = get.dw(obj)
215       out=1./obj.sw;
216       if strcmp(obj.Proc2D,'TPPI') && obj.dimensionality > 1
217         out(2) = out(2)/2;
218       end
219     end
220     function out = get.phase_array(obj)
221       if obj.dimensionality == 1
222         tmp = [obj.zerofill(1) 1];
223       else
224         tmp = obj.zerofill(1:obj.dimensionality);
225       end
226       out = zeros(tmp);
227       for i = 1:obj.dimensionality
228         tmp = size(out);
229         tmp(i) = 1;
230         out = out + repmat(obj.PhaseUnarrayed{i},tmp);
231       end
232       tmp = ones(1,numel(obj.n));
233       tmp(obj.dimensionality+1:end)=obj.n(obj.dimensionality+1:end);
234       out = repmat(out,tmp);
235     end
236     function out = get.f(obj)
237       out = cell(1,obj.dimensionality);
238       for i = 1:obj.dimensionality
```

```
239         tmp = ones(1,max([2 obj.dimensionality]));
240         tmp(i) = obj.zerofill(i);
241         if isa(obj,'PASS') && i == 2
242           out{i}=reshape(obj.m,tmp);
243         else
244           out{i}=reshape(linspace(-obj.sw(i)/2, ...
                obj.sw(i)/2,obj.zerofill(i)),tmp);
245           if mod(obj.zerofill(i),2) == 0
246             out{i} = out{i} + out{i}(end/2);
247           end
248         end
249       end
250     end
251     function out = get.F(obj)
252       out = cell(1,obj.dimensionality);
253       for i = 1:obj.dimensionality
254         tmp = obj.zerofill(1:obj.dimensionality);
255         tmp(i)=1;
256         out{i} = repmat(obj.f{i},tmp);
257       end
258     end
259     function out = get.ppm(obj)
260       out = cell(1,obj.dimensionality);
261       for i = 1:obj.dimensionality
262         out{i} = (obj.f{i} - obj.RefFreq(i) )/(obj.RefFreq(i) + ...
              obj.sf(1))*1E6;
263       end
264     end
265     function out = get.PPM(obj)
266       out = cell(1,obj.dimensionality);
267       for i = 1:obj.dimensionality
268         tmp = obj.zerofill(1:obj.dimensionality);
269         tmp(i)=1;
270         out{i} = repmat(obj.ppm{i},tmp);
271       end
272     end
273     function out = get.t(obj)
274       out = cell(1,obj.dimensionality);
275       for i = 1:obj.dimensionality
276         tmp = ones(1,max([2 obj.dimensionality]));
277         tmp(i) = obj.n(i);
278         out{i}=reshape((0:obj.n(i)-1)*obj.dw(i),tmp);
279       end
280     end
281     function out = get.T(obj)
282       out = cell(1,obj.dimensionality);
283       for i = 1:obj.dimensionality
284         tmp = obj.n(1:obj.dimensionality);
285         tmp(i)=1;
286         out{i} = repmat(obj.t{i},tmp);
287       end
```

```
288     end
289     function out = get.PhaseUnarrayed(obj)
290       out = cell(1,obj.dimensionality);
291       for i = 1:obj.dimensionality
292         tmp = ones(1,max([obj.dimensionality 2]));
293         tmp(i) = obj.zerofill(i);
294         if i == 1
295           out{i}=reshape(linspace(-(obj.phase1(i)+obj.phase1_Bruker)/2, ...
                (obj.phase1(i)+obj.phase1_Bruker)/2,obj.zerofill(i))+ ...
                obj.phase0(i),tmp);
296         else
297           out{i}=reshape(linspace(-obj.phase1(i)/2,obj.phase1(i)/2, ...
                obj.zerofill(i))+obj.phase0(i),tmp);
298         end
299
300       end
301     end
302     function out = get.LBUnarrayed(obj)
303       out = cell(1,obj.dimensionality);
304       for i = 1:obj.dimensionality
305         tmp = ones(1,max([2 obj.dimensionality]));
306         %                 tmp = size(out);
307         tmp(i) = obj.n(i);
308         if i ==2 && strcmpi(obj.Proc2D ,'states')
309           tmp(i) = tmp(i)/2;
310           out{i}=reshape(obj.lb(i).broaden(ones(1,obj.n(i)/2)),tmp);
311         else
312           out{i}=reshape(obj.lb(i).broaden(ones(1,obj.n(i))),tmp);
313         end
314       end
315     end
316     function out = get.n(obj)
317       out=size(obj.fid);
318     end
319     % Return broadened FID
320     function a = get.broaden(obj)
321       tmp = ones(1,max([numel(obj.n) 2]));
322       tmp(obj.dimensionality+1:end)=obj.n(obj.dimensionality+1:end);
323       if numel(obj.LBUnarrayed) > 1
324         a = obj.fid .* repmat(mtimes(obj.LBUnarrayed{:}),tmp);
325       else
326         a = obj.fid .* repmat(obj.LBUnarrayed{1},tmp);
327       end
328     end
329     function a = get.Istop(obj)
330       if isempty(obj.peaklist)
331         a = obj.zerofill(1);
332       else
333         F = max(obj.peaklist)+max([0.05*obj.sw(1) 12*obj.AssumeWidth]);
334         [junk a] = min(abs(obj.f{1}-F));
335       end
```

```
336     end
337     function a = get.Istart(obj)
338       if isempty (obj.peaklist)
339         a = 1;
340       else
341         F = min(obj.peaklist)-max([0.05*obj.sw(1) 12*obj.AssumeWidth]);
342         [junk a] = min(abs(obj.f{1}-F));
343       end
344     end
345     function a = get.noiseI(obj)
346       % Index for 'Non-peak' areas
347       if isempty(obj.peaklist)
348         a = 1:obj.zerofill(2);
349       else
350         tmp = (obj.Istart:obj.Istop)';
351         if ~isempty(obj.peaklist)
352           for i = 1:numel(obj.peaklist)
353             tmp((abs(obj.f{1}(tmp)-obj.peaklist(i))<6*obj.AssumeWidth)) ...
                  = [];
354             if ~isempty(obj.MAS) && ~obj.ExcludeSpinningsidebandsInNoise
355               for j = 1:ceil(obj.sw/2/obj.MAS)
356                 tmp((abs(obj.f{1}(tmp)-obj.peaklist(i)+j*obj.MAS)< ...
                      2*obj.AssumeWidth)) = [];
357                 tmp((abs(obj.f{1}(tmp)-obj.peaklist(i)-j*obj.MAS)< ...
                      2*obj.AssumeWidth)) = [];
358               end
359             end
360           end
361         end
362         a = tmp;
363       end
364     end
365     %% Set Methods
366     function obj = set.multishot(obj,val)
367       if rem(size(obj.fid_storage,1)-obj.leftshift,val) == 0
368         obj.multishot = val;
369         if obj.multishot == 1
370           obj.fid = obj.fid_storage(obj.leftshift:end,:);
371         else
372           tmp1= size(obj.fid_storage);
373           tmp1= [(tmp1(1)-obj.leftshift)/obj.multishot tmp1(2:end)];
374           obj.fid = squeeze(sum(reshape(obj.fid_storage,[tmp1(1) ...
                obj.multishot tmp1(2:end)]),2));
375         end
376       else
377         warning([int2str(size(obj.fid_storage,1)) ' is not divisible ...
              by ' num2str(val)])
378       end
379     end
380     function obj = set.leftshift(obj,val)
381       if obj.multishot ≠ 1
```

```
382         warning('Leftshift not currently supported with multishot ...
                experiments.')
383       elseif val<size(obj.fid_storage,1)
384         obj.leftshift = val;
385         obj.multishot = 1;
386       else
387         warning('Leftshift cannot exceed data size.')
388       end
389     end
390     function obj = set.sw(obj,val)
391       if numel(val) == obj.dimensionality
392         obj.sw = reshape(val,1,obj.dimensionality);
393         obj.sw_hidden(1:obj.dimensionality) = [obj.sw];
394       else
395         warning(['Currently in ' int2str(obj.dimensionality) 'D mode.'])
396       end
397     end
398     function obj = set.Proc2D(obj,val)
399       if strcmpi(val(1),'F')
400         obj.Proc2D = 'FFT';
401       elseif strcmpi(val(1),'S')
402         obj.Proc2D = 'States';
403       elseif strcmpi(val(1),'T')
404         obj.Proc2D = 'TPPI';
405       else
406         warning(['2D processing method must be FFT, States or TPPI. ...
                Using ' obj.Proc2D])
407       end
408     end
409     function obj = set.states_order(obj,val)
410       if strcmpi(val(1),'i')
411         obj.states_order = 'interleaved';
412       elseif strcmpi(val(1),'b')
413         obj.states_order = 'back_to_back';
414       else
415         warning(['states_order must be "interleaved" or ...
                "back_to_back". Using ' obj.states_order])
416       end
417     end
418     function obj = set.phase0(obj,val)
419       if numel(val) == obj.dimensionality
420         obj.phase0 = reshape(val,1,obj.dimensionality);
421         obj.phase0_hidden(1:obj.dimensionality) = [obj.phase0];
422         notify(obj,'ChangePhase');
423       else
424         warning(['Currently in ' int2str(obj.dimensionality) 'D mode.'])
425       end
426     end
427     function obj = set.phase1(obj,val)
428       if numel(val) == obj.dimensionality
429         obj.phase1 = reshape(val,1,obj.dimensionality);
```

```
430         obj.phase1_hidden(1:obj.dimensionality) = [obj.phase1];
431         notify(obj,'ChangePhase');
432       else
433         warning(['Currently in ' int2str(obj.dimensionality) 'D mode.'])
434       end
435     end
436     function set.RefFreq(obj,Freq)
437       if 0
438         if nargin < 2
439           error('Insufficient input.')
440         elseif nargin == 2
441           if numel(Freq) == 1
442             obj.RefFreq = Freq;
443           elseif numel(Freq) == 2
444             obj.RefFreq = ...
                  (Freq(1)-Freq(2)*1E-6*obj.sf(1))/(1-Freq(2)*1E-6);
445           end
446         else
447           error('Too much input.')
448         end
449       else
450         % include RefFreq_hidden & multidimensional input
451         if nargin > 2
452           error('This shouldn''t be possible...')
453         end
454         if size(Freq,2) > obj.dimensionality
455           error(['Input exceeds dimensionality (' ...
                 int2str(obj.dimensionality) 'D)'])
456         end
457         if size(Freq,1) == 2
458           I = ~isnan(Freq(2,:));
459           Freq(1,I) = ...
                (Freq(1,I)-Freq(2,I)*1E-6*obj.sf(1))/(1-Freq(2,I)*1E-6);
460         end
461         obj.RefFreq(~isnan(Freq(1,:))) = Freq(1,:);
462         obj.RefFreq_hidden(1:obj.dimensionality) = ...
              obj.RefFreq(1:obj.dimensionality);
463       end
464     end
465     function obj = set.dimensionality(obj,val)
466       if ~isnumeric(val)
467         error(['Dimensionality must be numeric, was ' class(val)])
468       end
469       if obj.dimensionality > val
470         obj.dimensionality = val;
471         obj.sw     = [obj.sw_hidden(     1:min([obj.dimensionality ...
              numel(obj.sw_hidden     )])) ...
              nan(1,obj.dimensionality-numel(obj.sw_hidden     ))];
472         obj.phase0 = [obj.phase0_hidden( 1:min([obj.dimensionality ...
              numel(obj.phase0_hidden )])) ...
              nan(1,obj.dimensionality-numel(obj.phase0_hidden ))];
```

```
473         obj.phase1 = [obj.phase1_hidden( 1:min([obj.dimensionality ...
                numel(obj.phase1_hidden )])) ...
                nan(1,obj.dimensionality-numel(obj.phase1_hidden ))];
474         obj.RefFreq= [obj.RefFreq_hidden(1:min([obj.dimensionality ...
                numel(obj.RefFreq_hidden)])) ...
                nan(1,obj.dimensionality-numel(obj.RefFreq_hidden))];
475       elseif obj.dimensionality < val
476         obj.dimensionality = val;
477         obj.sw     = [obj.sw_hidden         ...
                nan(1,val-numel(obj.sw_hidden     ))];
478         obj.phase0 = [obj.phase0_hidden  ...
                zeros(1,val-numel(obj.phase0_hidden ))];
479         obj.phase1 = [obj.phase1_hidden  ...
                zeros(1,val-numel(obj.phase1_hidden ))];
480         obj.RefFreq= [obj.RefFreq_hidden ...
                zeros(1,val-numel(obj.RefFreq_hidden))];
481       end
482       if numel(obj.lb) < obj.dimensionality
483         for i = numel(obj.lb)+1 : obj.dimensionality
484           obj.lb(i) = LBdata();
485         end
486       elseif numel(obj.lb) > obj.dimensionality
487         for i = numel(obj.lb) : -1 : obj.dimensionality+1
488           obj.lb(i) = [];
489         end
490       end
491     end
492     %% Conversion methods
493     function out = PASS(obj)
494       out = PASS(obj.fid,'pd',obj.pd,'nscans', ...
              obj.nscans,'spectrometer',obj.spectrometer);
495       oub.lb      = obj.lb(1);
496       out.phase0  = obj.phase0;
497       out.phase1  = obj.phase1;
498       out.sw      = [obj.sw obj.n(2)];
499       out.acqpars = obj.acqpars;
500       out.MAS     = obj.MAS;
501       out.sf      = obj.sf;
502       out.RefFreq = obj.RefFreq;
503       out.peaklist= obj.peaklist;
504     end
505   end
506
507   methods (Hidden = true)
508     function a=MakeSpec(obj)
509       switch obj.dimensionality
510         case 1
511           a=exp(-1i*pi*obj.phase_array/180).*fftshift(fft(obj.broaden, ...
                obj.zerofill(1),1),1);
512         case 2
513           if strcmp(obj.Proc2D,'FFT')
```

```
514             disp('Using untested FFT')
515             a=exp(-1i*pi*obj.phase_array/180).*fftshift(fft(fft(obj.broaden, ...
                    obj.zerofill(1),1),obj.zerofill(2),2),1);
516           elseif strcmp(obj.Proc2D,'States')
517             tmp = exp(-1i*pi*repmat(obj.PhaseUnarrayed{1},[1 ...
                    obj.n(2:end)])/180).*fftshift(fft(repmat(obj.LBUnarrayed{1},[1 ...
                    obj.n(2:end)]).*obj.fid,obj.zerofill(1),1),1);
518             if strcmpi(obj.states_order,'interleaved')
519               cos_data = tmp(:,1:2:end);
520               sin_data = tmp(:,2:2:end);
521             elseif strcmpi(obj.states_order,'back_to_back')
522               cos_data = tmp(:,1:end/2);
523               sin_data = tmp(:,(1:end/2)+end/2);
524             end
525             a = exp(-1i*pi*repmat(obj.PhaseUnarrayed{2}, [obj.n(1) 1 ...
                    obj.n(3:end)])/180).*fftshift(fft(repmat(obj.LBUnarrayed{2}, ...
                    [obj.n(1) 1 ...
                    obj.n(3:end)]).*real(cos_data)+1i*real(sin_data), ...
                    obj.zerofill(2),2),2);
526           elseif strcmp(obj.Proc2D,'TPPI')
527             tmp= ...
                    repmat(exp(-1i*pi*obj.PhaseUnarrayed{1}/180),1,obj.n(2)).* ...
                    fftshift(fft(obj.broaden,obj.zerofill(1),1),1);
528             obj.zerofill(2) = 2*obj.zerofill(2);
529             tmp= repmat(exp(-1i*pi*obj.PhaseUnarrayed{2}/180), ...
                    obj.zerofill(1), ...
                    1).*fftshift(fft(repmat(obj.LBUnarrayed{2}, ...
                    obj.zerofill(1),1).*real(tmp), obj.zerofill(2),2),2);
530             obj.zerofill(2) = obj.zerofill(2)/2;
531             a  = real(tmp(:,end:-1:end/2+1));
532           end
533       end
534     end
535   end
536 end
```

B.4 Linebroadening class file

The line broadening functions have been given their own class files so that they are easy to modify. The current version includes five function types: flat (no broadening), hann, hamming, exponential and gaussian. When the class is initialised, the constructor creates a dummy version of the class, and reads the available methods to populate the `AllTypes` property; this means that `AllTypes` always contains an up-to-date list of which line broadening functions are available (there is also a `Cripple` method which permanently sets the type to flat). The line broadening class is designed to be stand alone, so it does not have access to the spectral width of the experiment to which it is attached. When broadening is applied, the functions are chosen such that the apparent width of the line broadening in Hz will be `width·sw`.

```
classdef LBdata < handle
  %% Properties
  properties (Hidden = true , SetAccess = 'private')
    AllTypes
  end

  properties
    type = 'flat';
    width=1;
  end

  properties (Hidden = true)
    shift=1;
  end

  %% Events
  events
    ChangeLB
  end

  methods
    %% Constructor
    function obj = LBdata()
      obj.CheckTypes();
    end
    %% Set & get functions
    function obj = set.type(obj,in)
      in=lower(in);
      for i = 1:length(obj.AllTypes)
        if ~isempty(strfind(obj.AllTypes{i},in))
          obj.type = obj.AllTypes{i};
          notify(obj,'ChangeLB');
          return
        end
```

```
35        end
36        error(['Unknown linebroadening type ' in])
37      end
38      function obj = set.width(obj,in)
39        obj.width = in;
40        notify(obj,'ChangeLB');
41      end
42      %% General functions
43      function a = broaden(obj,fid)
44        n=size(fid);
45        if n(1) == 1 && n(2) > 1
46          n(1:2) = [n(2) 1];
47        end
48        a=fid.*reshape(repmat(obj.(obj.type)(n(1)),1,n(2)),size(fid));
49      end
50    end
51    methods (Access  = private)
52      %% Collect linebroadening methods here
53      function a = flat(obj,n)
54        a=ones(n,1);
55      end
56      function a = hann(obj,n)
57        a = (1-cos(pi*((1:n)+(n-2*obj.shift))/(n-obj.shift)))/2;
58      end
59      function a = hamming(obj,n)
60        a = 0.54-0.46*cos(pi*((1:n)+(n-2*obj.shift))/(n-obj.shift));
61      end
62      function a = exponential(obj,n)
63        a=exp(-obj.width*pi*((1:n)-obj.shift).*sign((1:n)-obj.shift))';
64      end
65      function a = gaussian(obj,n)
66        a = exp(-2*(obj.width/sqrt(8*log(2))*pi*((1:n)-obj.shift)').^2);
67      end
68    end
69    methods (Hidden = true)
70      function CheckTypes(obj)
71        a = ? LBdata;
72        b = cell(size(a.Methods));
73        for i = 1:numel(b)
74          b{i} = a.Methods{i}.Access;
75        end
76        tmp = find(strcmp(b,'private'));
77        obj.AllTypes = cell(size(tmp));
78        for i = 1:numel(tmp)
79          obj.AllTypes{i} = a.Methods{tmp(i)}.Name;
80        end
81      end
82      function Cripple(obj)
83        obj.type = 'flat';
84        obj.AllTypes = {'flat'};
85      end
```

```
86    end
87 end
```

B.5 Phasing GUI

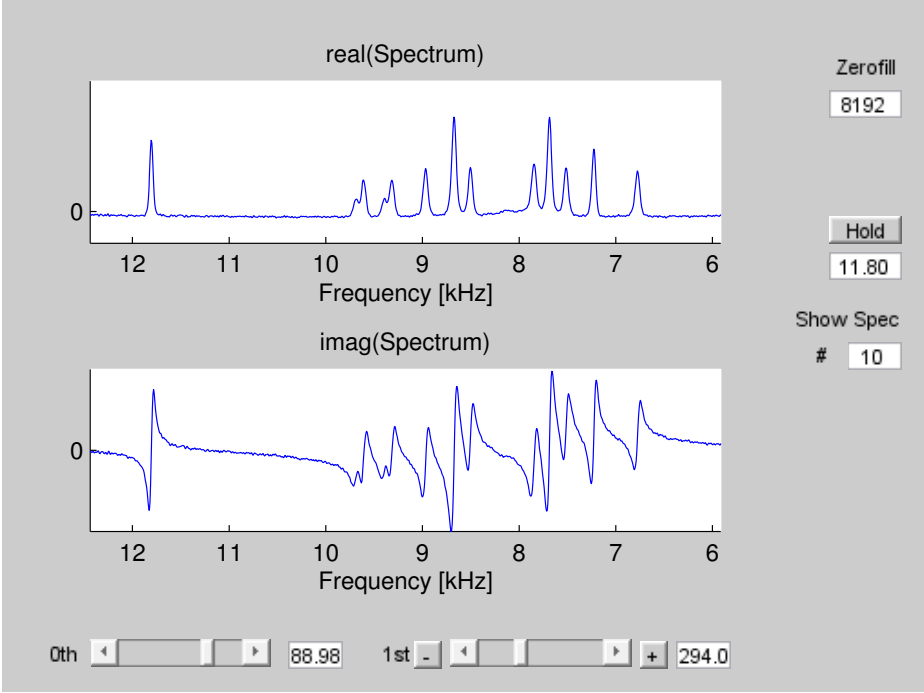


Figure B.1: Phasing GUI.

This program creates a GUI for phasing 1D spectra; the interface is shown in Figure B.1. The top plot shows the real part of the current spectrum, the bottom window shows the imaginary part. The 0^{th} and 1^{st} order phasing are controlled via the sliders at the bottom of the interface. The phase sliders automatically update the phase in the experimental data (`phase0` & `phase1`), likewise the GUI detects changes to the processing via system events (`ChangeLB` & `ChangePhase`) and replots the spectrum if, for instance, the line broadening is changed. The zero filling can be controlled from the input in the top right corner, and the pivot point of the 1^{st} order phasing can be set by clicking on the "hold" button, and then selecting a point in the spectrum. The GUI is also designed for arrayed input; the desired spectrum can be selected by typing its number into the "Show Spec" box.

By default, the GUI displays data in the direct dimension. For multi-dimensional data, however, it is possible to display slices along other dimensions. If `a` contains a 2D data set, then the command `a.ShowPhasing('dimension',2)` will display the second dimension. More importantly, multiple phasing GUIs can be open at once as long as they are linked to different dimensions. This means that phase adjust-

ments can easily be made to multidimensional data without the need to start again if the phasing in the direct dimension was not quite right the first time.

```
function MyPhasingGUI(spec,varargin)
%% Sanity Check
options = struct('dimension',1); % Structure with default options
if nargin < 1
  %% No input
  error('Input required!')
elseif nargin > 1
  %% Options
  if isstruct(varargin{1})
    fields = fieldnames(varargin{1});
    for i = 1:numel(fields)
      options.(fields{i}) = varargin{1}.(fields{i});
    end
    varargin(1) = [];
  end
  if mod(numel(varargin),2) == 0
    for i = 1:numel(varargin)/2
      options.(varargin{2*i-1}) = varargin{2*i};
    end
  else
    error('Too much information you numbskull!')
  end
end
if numel(spec.PhaseFigure) < options.dimension
  spec.PhaseFigure = [spec.PhaseFigure ...
     nan(1,options.dimension-numel(spec.PhaseFigure))];
end
if ~isa(spec,'FID')
  %% Never trust user input!
  error(['Input class must be FID, was ' class(spec) '.'])
end

%% figure
if isnan(spec.PhaseFigure(options.dimension))
  f = figure('Visible','off', 'Position',[128,512,512,384], ...
     'menubar','none', 'name',['Phasing (' ...
     int2str(options.dimension) ')'],'numbertitle','off', ...
     'toolbar','figure', 'CloseRequestFcn',@my_closefcn);
  spec.PhaseFigure(options.dimension) = f;
else
  f=spec.PhaseFigure(options.dimension);
  figure(f)
  clf
  set(f,'Visible','off', 'Position',[128,512,512,384], ...
     'menubar','none', 'name',['Phasing (' ...
     int2str(options.dimension) ')'], 'numbertitle','off', ...
     'toolbar','figure', 'CloseRequestFcn',@my_closefcn);
```

```
41 end
42 bgcol = get(gcf,'color');
43
44 % Phasing Segment
45 %% axes
46 h.RealAxes=axes('units','pixels', 'position',[50,250,350,90], ...
       'Nextplot','replacechildren', ...
       'xlim',[-spec.sw(options.dimension) ...
       spec.sw(options.dimension)]/2E3, 'ylim',[-1 1]);
47 h.ImagAxes=axes('units','pixels', 'position',[50, 90,350,90], ...
       'Nextplot','replacechildren', ...
       'xlim',[-spec.sw(options.dimension) ...
       spec.sw(options.dimension)]/2E3, 'ylim',[-1 1]);
48 linkaxes([h.RealAxes h.ImagAxes],'x');
49 %% Zero Filling
50 h.ZeroFillLbl=uicontrol('style','text', 'position',[460,340,40,15], ...
       'string','Zerofill', 'BackgroundColor',bgcol);
51 h.ZeroFill   =uicontrol('style','edit', 'position',[460,320,40,15], ...
       'Background',[1 1 1], 'value',spec.zerofill(options.dimension), ...
       'string',num2str(spec.zerofill(options.dimension)), ...
       'CallBack',{@ZeroFill_Callback});
52 %% Phase Sliders
53 FirstMultiplier=[0 0];
54 h.ZeroOrderSlider =uicontrol('style','slider', 'position',[ ...
       50,15,100,15], 'BackgroundColor',bgcol, 'min',-180, 'max',180, ...
       'value',spec.phase0(options.dimension), ...
       'CallBack',{@ZeroOrderSlider_Callback});
55 h.ZeroOrderLbl    =uicontrol('style','text', 'position',[ 20,14, ...
       30,15], 'string','0th', 'BackgroundColor',bgcol);
56 h.ZeroOrderDsp    =uicontrol('style','edit', 'position',[160,14, ...
       30,15], 'Background',[1 1 1], ...
       'val',spec.phase0(options.dimension), ...
       'string',num2str(spec.phase0(options.dimension)), ...
       'CallBack',{@ZeroOrderDsp_Callback});
57 h.FirstOrderSlider=uicontrol('style','slider', ...
       'position',[250,15,100,15], 'BackgroundColor',bgcol, ...
       'min',(ceil((spec.phase1(options.dimension)-180)/360)-0.5)*360, ...
       'max',(ceil((spec.phase1(options.dimension)-180)/360)+0.5)*360, ...
       'value',spec.phase1(options.dimension), ...
       'CallBack',{@FirstOrderSlider_Callback});
58 h.FirstOrderLbl   =uicontrol('style','text', 'position',[205,14, ...
       30,15], 'string','1st', 'BackgroundColor',bgcol);
59 h.FirstOrderDsp   =uicontrol('style','edit', 'position',[375,14, ...
       30,15], 'Background',[1 1 1], ...
       'val',spec.phase1(options.dimension), ...
       'string',num2str(spec.phase1(options.dimension)), ...
       'CallBack',{@FirstOrderDsp_Callback});
60 h.FirstOrderPlus  =uicontrol('style','pushbutton', ...
       'position',[355,14, 15,15], 'string','+', ...
       'BackgroundColor',bgcol, 'CallBack',{@FirstOrderPlus_Callback});
61 h.FirstOrderMinus =uicontrol('style','pushbutton', ...
```

```
    'position',[230,14, 15,15], 'string','-', ...
    'BackgroundColor',bgcol, 'CallBack',{@FirstOrderMinus_Callback});
ConstFreq=[0];
h.ConstFreqButton =uicontrol('style','pushbutton', ...
    'position',[460,250, 40,15], 'string','Hold', ...
    'BackgroundColor',bgcol, 'CallBack',{@ConstFreqButton_Callback});
h.ConstFreqDsp    =uicontrol('style','edit', 'position',[460,230, ...
    40,15], 'Background',[1 1 1], 'val',ConstFreq, ...
    'string',num2str(ConstFreq), 'CallBack',{@ConstFreqDsp_Callback});
%% Spectrum Chooser
h.ChooseSpectxt   =uicontrol('style','text', ...
    'position',[440,200,60,15], 'string','Show Spec', ...
    'BackgroundColor',bgcol);
h.ChooseSpecNum   =uicontrol('style','edit', 'position',[470,180, ...
    30,15], 'string','1', 'val',1, 'BackgroundColor',[1 1 1], ...
    'CallBack',{@ChooseSpecNum_Callback});
h.ChsSpecNumLbl   =uicontrol('style','text', 'position',[440,180, ...
    30,15], 'string','#', 'BackgroundColor',bgcol);

set(cell2mat(struct2cell(h)),'units','normalized')

listener(1) = addlistener(spec,'ChangePhase',@PlotPhase);
for j = 1:numel(spec.lb)
  listener(j+1) = addlistener(spec.lb(j),'ChangeLB',@PlotPhase);
end
set(f,'visible','on')
PlotPhase()

%% Callback functions
  function my_closefcn(source,eventdata)
    spec.PhaseFigure = [];
    delete(listener);
    delete(f)
  end
  function ChooseSpecNum_Callback(source,eventdata)
    val=get(source,'string');
    val=str2num(val);
    if ~isempty(val)
      val = round(val);
      switch options.dimension
        case 1
          nmax = spec.zerofill(2);
        case 2
          nmax = spec.zerofill(1);
      end
      if val ≤ nmax && val ≥ 1
        set(source,'val',val);
      end
    end
    set(source,'string',num2str(get(source,'val'),'%i'));
    PlotPhase()
```

```
102   end
103   function ZeroOrderSlider_Callback(source,eventdata)
104     spec.phase0(options.dimension) = get(source,'val');
105     set(h.ZeroOrderDsp ,'value',spec.phase0(options.dimension), ...
          'string',num2str(spec.phase0(options.dimension)))
106   end
107   function ZeroOrderDsp_Callback(source,eventdata)
108     val=str2num(get(source,'string'));
109     if isempty(val)
110       val=spec.phase0(options.dimension);
111       set(source,'string',num2str(val))
112     else
113       val=mod(val+180,360)-180;
114       set(h.ZeroOrderSlider,'val',val)
115       ZeroOrderSlider_Callback(h.ZeroOrderSlider,[])
116     end
117   end
118   function ConstFreqButton_Callback(source,eventdata)
119     axes(h.RealAxes)
120     [val,y]=ginput(1);
121     ConstFreq=val*1E3;
122     set(h.ConstFreqDsp,'string',num2str(val,'%.2f'),'value',val*1E3)
123   end
124   function ConstFreqDsp_Callback(source,eventdata)
125     val=str2num(get(source,'string'));
126     if isempty(val)
127       set(source,'string',num2str(ConstFreq/1E3,'%.2f'),'value',ConstFreq)
128     else
129       ConstFreq=val*1E3;
130       set(source,'string',num2str(val,'%.2f'),'value',val*1E3)
131     end
132   end
133   function FirstOrderSlider_Callback(source,event)
134     val = get(source,'val');     % New 1st order phasing, must set ...
          new 0th order phasing
135     listener(1).Enabled = false;
136     spec.phase0(options.dimension)=mod(spec.phase0(options.dimension)+ ...
          ConstFreq/spec.sw(options.dimension)* ...
          (spec.phase1(options.dimension)-val)+180,360)-180;
137     set(h.ZeroOrderSlider,'value',spec.phase0(options.dimension))
138     set(h.ZeroOrderDsp,'value', spec.phase0(options.dimension), ...
          'string',num2str(spec.phase0(options.dimension)))
139     listener(1).Enabled = true;
140     spec.phase1(options.dimension)=val;
141     set(h.FirstOrderDsp,'value',spec.phase1(options.dimension), ...
          'string',num2str(spec.phase1(options.dimension)))
142   end
143   function FirstOrderPlus_Callback(source,event)
144     set(h.FirstOrderSlider,'max',get(h.FirstOrderSlider,'max')+360, ...
          'min',get(h.FirstOrderSlider,'min')+360, ...
          'val',get(h.FirstOrderSlider,'val')+360)
```

```
145       FirstOrderSlider_Callback(h.FirstOrderSlider)
146     end
147     function FirstOrderMinus_Callback(source,event)
148       set(h.FirstOrderSlider,'max',get(h.FirstOrderSlider,'max')-360, ...
              'min',get(h.FirstOrderSlider,'min')-360, ...
              'val',get(h.FirstOrderSlider,'val')-360)
149       FirstOrderSlider_Callback(h.FirstOrderSlider)
150     end
151     function FirstOrderDsp_Callback(source,event)
152       val = str2num(get(source,'string'));
153       if ~isempty(val)
154         set(source,'val',val)
155         set(h.FirstOrderSlider,'max',(ceil((val-180)/360)+0.5)*360, ...
                'min',(ceil((val-180)/360)-0.5)*360,'val',val)
156         FirstOrderSlider_Callback(h.FirstOrderSlider)
157       end
158     end
159     function ZeroFill_Callback(source,event)
160       val = get(source,'string');
161       if strcmp(val(end),'k')
162         val = 1024*round(str2num(val(1:end-1)));
163       else
164         val = round(str2num(val));
165       end
166       if ~isempty(val)
167         spec.zerofill(options.dimension) = val(1);
168       end
169       set(source,'val',spec.zerofill(1), ...
              'string',num2str(spec.zerofill(options.dimension)))
170       PlotPhase();
171     end
172   %% General functions
173     function PlotPhase(source,eventdata)
174       axes(h.RealAxes)
175       i = get(h.ChooseSpecNum,'val');
176       if options.dimension == 2
177         if isempty(spec.peaklist)
178           plot(spec.f{options.dimension}/1E3, ...
                  real(spec.spec(i,:))/max(abs(real(spec.spec(i,:)))),'b')
179         else
180           plot(spec.f{options.dimension}/1E3, ...
                  real(spec.spec(i,:))/max(abs(real(spec.spec(i,:)))),'b', ...
                  spec.peaklist/1E3,repmat(0,size(spec.peaklist)),'r.')
181         end
182       else
183         if isempty(spec.peaklist)
184           plot(spec.f{options.dimension}/1E3, ...
                  real(spec.spec(:,i))/max(abs(real(spec.spec(:,i)))),'b')
185         else
186           plot(spec.f{options.dimension}/1E3, ...
                  real(spec.spec(:,i))/max(abs(real(spec.spec(:,i)))),'b', ...
```

```
            spec.peaklist/1E3,repmat(0,size(spec.peaklist)),'r.')
      end
    end
    title('real(Spectrum)')
    xlabel('Frequency [kHz]')
    set(h.RealAxes,'xdir','reverse','ytick',0)
    axes(h.ImagAxes)
    if options.dimension == 2
      plot(spec.f{options.dimension}/1E3, ...
          imag(spec.spec(i,:))/max(abs(imag(spec.spec(i,:)))),'b')
    else
      plot(spec.f{options.dimension}/1E3, ...
          imag(spec.spec(:,i))/max(abs(imag(spec.spec(:,i)))),'b')
    end
    title('imag(Spectrum)')
    xlabel('Frequency [kHz]')
    set(h.ImagAxes,'xdir','reverse','ytick',0)
  end
end
```

B.6 Linebroadening GUI

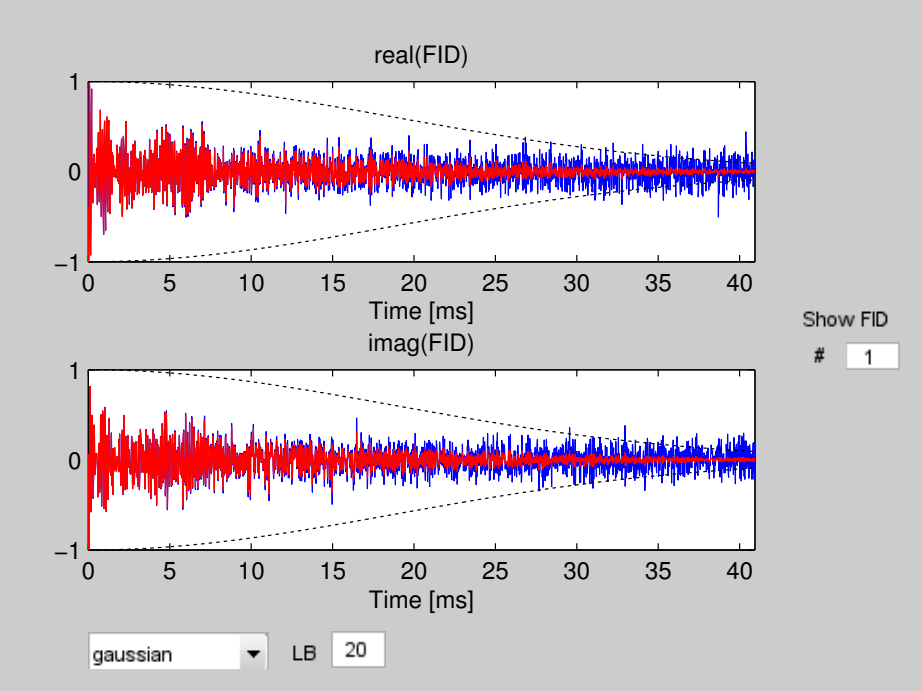


Figure B.2: Line broadening GUI.

This program creates a GUI for adjusting the line broadening of a spectrum (see Figure B.2). The top window shows the real part of the FID, the bottom window shows the imaginary part. The unbroadened FID is shown in blue, the broadened one in red and the envelope of the broadening function is shown as black dotted lines. The various line broadening functions can be selected from the drop down list at the bottom (the list is populated using `AllTypes` from the `lb` class file and is hence always up to date). As before, the GUI is designed to work with multi-dimensional data; the desired spectrum can be selected by typing the required number into the "Show FID" box, and the line broadening GUI for the indirect dimension of a 2D data set `a` can be called via `a.ShowLB('dimension',2)`.

```
function MyLineBroadeningGUI(spec,varargin)
%% Sanity Check
options = struct('dimension',1); % Structure with default options
if nargin < 1
  %% No input
  error('Input required!')
elseif nargin > 1
  %% Options
  if isstruct(varargin{1})
    fields = fieldnames(varargin{1});
    for i = 1:numel(fields)
```

```
12        options.(fields{i}) = varargin{1}.(fields{i});
13      end
14      varargin(1) = [];
15    end
16    if mod(numel(varargin),2) == 0
17      for i = 1:numel(varargin)/2
18        options.(varargin{2*i-1}) = varargin{2*i};
19      end
20    else
21      error('Too much information you numbskull!')
22    end
23  end
24  if numel(spec.LBFigure) < options.dimension
25    spec.LBFigure = [spec.LBFigure ...
          nan(1,options.dimension-numel(spec.LBFigure))];
26  end
27  if ~isa(spec,'FID')
28    %% Never trust user input!
29    error(['Input class must be FID, was ' class(spec) '.'])
30  end
31
32  %% figure
33  if isnan(spec.LBFigure(options.dimension))
34    f = figure('Visible','off', 'Position',[128,512,512,384], ...
          'menubar','none', 'name',['LineBroadening (' ...
          int2str(options.dimension) ')'], 'numbertitle','off', ...
          'toolbar','figure', 'CloseRequestFcn',@my_closefcn);
35    spec.LBFigure(options.dimension) = f;
36  else
37    f=spec.LBFigure(options.dimension);
38    figure(f)
39    clf
40    set(f,'Visible','off','Position',[128,512,512,384], ...
          'menubar','none', 'name',['LineBroadening (' ...
          int2str(options.dimension) ')'], 'numbertitle','off', ...
          'toolbar','figure', 'CloseRequestFcn',@my_closefcn);
41  end
42  bgcol = get(gcf,'color');
43
44  %% Axes
45  h.RealAxes=axes('units','pixels','position',[50,240,370,100]);
46  h.ImagAxes=axes('units','pixels','position',[50, 80,370,100]);
47  linkaxes([h.RealAxes h.ImagAxes],'x');
48  %% Fid Chooser
49  h.Choosetxt=uicontrol('style','text', 'position',[440,200,60,15], ...
      'string','Show FID', 'BackgroundColor',bgcol);
50  h.ChooseFidNum=uicontrol('style','edit', 'position',[470,180,30,15], ...
      'string','1', 'val',1, 'BackgroundColor',[1 1 1], ...
      'CallBack',{@ChooseFidNum_Callback});
51  h.ChsFidNumLbl=uicontrol('style','text', 'position',[440,180,30,15], ...
      'string','#', 'BackgroundColor',bgcol);
```

```
52
53 % Line Broadening
54 AllTypes = spec.lb(options.dimension).AllTypes;
55 [TF LOC]=ismember(spec.lb(options.dimension).type,AllTypes);
56 h.ChooseLb = uicontrol('Style','popupmenu', 'String',AllTypes, ...
       'units','pixels', 'Position',[50,15,100,20], 'value',LOC, ...
       'BackgroundColor',[1 1 1], 'Callback',{@ChooseLb_Callback});
57 h.ChseLbTxt=uicontrol('style','text', 'position',[160,10,20,20], ...
       'string','LB', 'visible','on', 'BackgroundColor',bgcol);
58 h.ChseLbVal=uicontrol('style','edit', 'position',[185,15,30,20], ...
       'string',num2str(spec.lb(options.dimension).width* ...
       spec.sw(options.dimension),'%.0f'), ...
       'val',spec.lb(options.dimension).width*spec.sw(options.dimension), ...
       'Background',[1 1 1], 'visible','on', ...
       'CallBack',{@ChseLbVal_Callback});
59 h.ChseGbTxt=uicontrol('style','text', 'position',[220,10,20,20], ...
       'string','GB', 'visible','off' ,'BackgroundColor',bgcol);
60 h.ChseGbVal=uicontrol('style','edit', 'position',[245,15,30,20], ...
       'string','10', 'val',10, 'Background',[1 1 1], 'visible','off' ...
       ,'CallBack',{@ChseGbVal_Callback});
61
62 set(cell2mat(struct2cell(h)),'units','normalized')
63 PlotLB()
64 listener = addlistener(spec.lb(options.dimension),'ChangeLB',@PlotLB);
65 set(f,'visible','on')
66
67 %% Callback functions
68   function ChooseLb_Callback(source,eventdata)
69     I = get(source,'value');
70     spec.lb(options.dimension).type = AllTypes{I};
71   end
72   function ChooseFidNum_Callback(source,eventdata)
73     val=get(source,'string');
74     val=str2num(val);
75     if ~isempty(val)
76       val = round(val);
77       switch options.dimension
78         case 1
79           if val ≤ spec.n(2) && val ≥ 1
80             set(source,'val',val);
81           end
82         case 2
83           if val ≤ spec.n(1) && val ≥ 1
84             set(source,'val',val);
85           end
86       end
87     end
88     set(source,'string',num2str(get(source,'val'),'%i'));
89     PlotLB()
90   end
91   function ChseLbVal_Callback(source,eventdata)
```

```
92      val=get(source,'string');
93      val=str2num(val);
94      if ~isempty(val)
95        set(source,'val',val);
96        spec.lb(options.dimension).width = val/spec.sw(options.dimension);
97      end
98      set(source,'string',num2str(get(source,'val'),'%.0f'));
99    end
100   function my_closefcn(source,eventdata)
101     spec.LBFigure = [];
102     delete(listener);
103     delete(gcf)
104   end
105 %% General functions
106   function PlotLB(source,eventdata)
107     axes(h.RealAxes)
108     i = get(h.ChooseFidNum,'val');
109     if options.dimension == 2
110       plot(spec.t{options.dimension}*1E3, ...
            real(spec.fid(i,:))/max(abs(real(spec.fid(i,:)))),'b', ...
            spec.t{options.dimension}*1E3, ...
            spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            -spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            real(spec.broaden(i,:))/max(abs(real(spec.broaden(i,:)))),'r')
111     else
112       plot(spec.t{options.dimension}*1E3, ...
            real(spec.fid(:,i))/max(abs(real(spec.fid(:,i)))),'b', ...
            spec.t{options.dimension}*1E3, ...
            spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            -spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            real(spec.broaden(:,i))/max(abs(real(spec.broaden(:,i)))),'r')
113     end
114     title('real(FID)')
115     xlabel('Time [ms]')
116     axes(h.ImagAxes)
117     if options.dimension == 2
118       plot(spec.t{options.dimension}*1E3, ...
            imag(spec.fid(i,:))/max(abs(imag(spec.fid(i,:)))),'b', ...
            spec.t{options.dimension}*1E3, ...
            spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            -spec.lb(options.dimension).broaden(ones( ...
```

```
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            imag(spec.broaden(i,:))/max(abs(imag(spec.broaden(i,:)))),'r')
    else
      plot(spec.t{options.dimension}*1E3, ...
            imag(spec.fid(:,i))/max(abs(imag(spec.fid(:,i)))),'b', ...
            spec.t{options.dimension}*1E3, ...
            spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            -spec.lb(options.dimension).broaden(ones( ...
            spec.n(options.dimension),1)),'k:', ...
            spec.t{options.dimension}*1E3, ...
            imag(spec.broaden(:,i))/max(abs(imag(spec.broaden(:,i)))),'r')
    end
    title('imag(FID)')
    xlabel('Time [ms]')
  end
end
```

C CASTEP

CASTEP calculations require two input files. The first is the *.cell file, which contains parameters relating to the unit cell, e.g. the lattice parameters and the atom positions within the unit cell. An example is shown in Section C.1. The other file is the *.param file which describes the calculations to be undertaken. Section C.2 shows a typical *.param file for a geometry optimisation, while Section C.3 contains typical values for the calculation of chemical shift parameters. For the data presented in this thesis, multiple calculations were performed: first a geometry optimisation of only the hydrogen atoms was performed, and then a geometry optimisation of the whole molecule (both with fixed lattice parameters). Finally, the NMR parameters for the fully relaxed unit cell were calculated.

C.1 Cell file

```
%BLOCK LATTICE_ABC
12.523000 7.868000 12.874000
90.000000 95.199997 90.000000
%ENDBLOCK LATTICE_ABC

%BLOCK POSITIONS_FRAC
        H       0.472145   0.993601   0.394113
        H       0.527855   0.006399   0.605887

  ⋮


        F       0.975100   0.533500   0.342600
        F       0.024900   0.466500   0.657400
%ENDBLOCK POSITIONS_FRAC

KPOINTS_MP_SPACING 0.02

FIX_ALL_CELL : TRUE

FIX_COM : FALSE

SYMMETRY_GENERATE

SNAP_TO_SYMMETRY
```

C.2 Parameter file (geometry optimisation)

```
comment : Test Geometry Optimisation for Flufenamic Acid
task : GeometryOptimization
xc_functional : PBE
opt_strategy : Speed
basis_precision : extreme
fix_occupancy : true
metals_method : dm
geom_method : BFGS
geom_max_iter : 120
MAX_SCF_CYCLES: 60
```

C.3 Parameter file (NMR)

```
comment : Get NMR shielding values for Fulfenamic Acid
task : MagRes
xc_functional : PBE
max_scf_cycles : 60
opt_strategy : Speed
basis_precision : extreme
fix_occupancy : true
metals_method : dm
```