Highly efficient quantum spin dynamics simulation algorithms

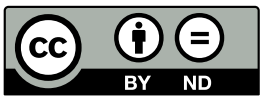
by

Luke J. Edwards

Oriel College,
University of Oxford.

A thesis submitted for the degree of
Doctor of Philosophy

Trinity term, 2014



Highly efficient quantum spin dynamics simulation algorithms

Luke J. Edwards

Oriel College, University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Trinity term, 2014

Abstract

Spin dynamics simulations are used to gain insight into important magnetic resonance experiments in the fields of chemistry, biochemistry, and physics. Presented in this thesis are investigations into how to accelerate these simulations by making them more efficient.

Chapter 1 gives a brief introduction to the methods of spin dynamics simulation used in the rest of the thesis. The 'exponential scaling problem' that formally limits the size of spin system that can be simulated is described.

Chapter 2 provides a summary of methods that have been developed to overcome the exponential scaling problem in liquid state magnetic resonance.

The possibility of utilizing the multiple processors prevalent in modern computers to accelerate spin dynamics simulations provides the impetus for the investigation found in Chapter 3. A number of different methods of parallelization leading to acceleration of spin dynamics simulations are derived and discussed.

It is often the case that the parameters defining a spin system are time-dependent. This complicates the simulation of the spin dynamics of the system. Chapter 4 presents a method of simplifying such simulations by mapping the spin dynamics into a larger state space. This method is applied to simulations incorporating mechanical spinning of the sample with powder averaging.

In Chapter 5, implementations of several magnetic resonance experiments are detailed. In so doing, use of techniques developed in Chapters 2 and 3 are exemplified. Further, specific details of these experiments are utilized to increase the efficiency of their simulation.

Acknowledgements

I would first like to thank my supervisor, Ilya Kuprov, for being a seemingly endless source of inspiration, and also for giving me the space to explore my own ideas.

Secondly, I thank my co-supervisor, Chris Timmel for her support and for arranging the many office moves that have proved necessary.

For the past two years I have been somewhat of a roaming student; I thank John Mcgrady, Lorna Smith, and Peter Hore for allowing me use of their respective group office spaces. I also especially thank Jason and Jen for making me feel at home in the Hore group office.

Thank you to the members of the Timmel group for not treating me like an outsider. In particular, I thank Cat, Claudia, Emrys, and Jon for many fun lunchtime conversations over occasionally-questionable soup; and Alice and Kelly-Anne for proving that a DPhil thesis is not an insurmountable obstacle.

During the course of my DPhil, I spent a week in Karlsruhe proselytizing about *Spinach* and learning about CLIP-HSQC. I thank Burkhard Luy for hosting me, and for explaining both the CLIP-HSQC pulse sequence and the States detection method to me. Further, I thank Martin Koos for welcoming me to KIT, helpful discussions about CLIP-HSQC, and several games of *Unreal Tournament*.

I acknowledge many miscellaneous people who have helped me: Matthew Krzystiniak, for helpful discussions about destination state screening and *Spinach* in general; Hannah Hogben, for help getting to grips with the *Spinach* codebase; Will Myers, for explaining what 'superheterodyne receivers' are—I hope that one day I will understand what all the words you used mean; Alex Nevzorov, for helpful comments on the elevation method derivation; Andi Kolmer, for various strychnine spin system parameters; and Zenawi Welderufael for inputting the spin system of ubiquitin into *Spinach*.

Finally, I thank Debbie for her love and support.

Publications arising from this work

[1] M. Krzystyniak, Luke J. Edwards, and Ilya Kuprov.
"Destination state screening of active spaces in spin dynamics simulations".
In: *Journal of Magnetic Resonance* 210 (2011), pp. 228–232.
DOI: 10.1016/j.jmr.2011.03.010

[2] Alexander Karabanov, Ilya Kuprov, G. T. P. Charnock,
Anniek van der Drift, Luke J. Edwards, and Walter Köckenberger.
"On the accuracy of the state space restriction approximation for spin dynamics simulations".
In: *The Journal of Chemical Physics* 135.8 (2011), p. 084106.
DOI: 10.1063/1.3624564

[3] Luke J. Edwards and Ilya Kuprov.
"Parallel density matrix propagation in spin dynamics simulations".
In: *The Journal of Chemical Physics* 136.4 (2012), p. 044108.
DOI: 10.1063/1.3679656

[4] Luke J. Edwards, D. V. Savostyanov, A. A. Nevzorov, M. Concistrè,
G. Pileio, and Ilya Kuprov. "Grid-free powder averages: On the applications of the Fokker–Planck equation to solid state NMR".
In: *Journal of Magnetic Resonance* 235 (2013), pp. 121–129.
DOI: 10.1016/j.jmr.2013.07.011

[5] Luke J. Edwards, D. V. Savostyanov, Z. T. Welderufael, Donghan Lee, and Ilya Kuprov. "Quantum mechanical NMR simulation algorithm for protein-size spin systems".
In: *Journal of Magnetic Resonance* 243 (2014), pp. 107–113.
DOI: 10.1016/j.jmr.2014.04.002

Contents

Chapter 1

Introduction

Elementary particles (and the compound particles constructed from them) have an internal degree of freedom known as 'spin' [6]. This thesis deals with simulating the dynamics of this degree of freedom.

The spin degree of freedom is constrained by the 'spin quantum number' of the particle. This integer or half-integer number dictates the maximum absolute value that the spin degree of freedom can take [6]. A particle with non-zero spin quantum number will be referred to colloquially as a 'spin'. All of the particles used in the simulations presented in this thesis have spin quantum number 1/2 (they are 'spin-1/2 particles'). Specifically, the spin-1/2 particles used are electrons and the nuclei ^{1}H, ^{15}N, and ^{13}C. The concepts presented will usually apply generally; it is explicitly stated when a concept does not apply to higher-spin particles.

Experimentally, spin manifests itself through an associated magnetic moment which gives rise to an energy level splitting in the presence of magnetic fields [7]. Transitions between these energy levels form the foundation of magnetic resonance spectroscopy. Probing the transitions of nuclear spins is known as 'nuclear magnetic resonance' (NMR) [7, 8], probing the transitions of electron spins is known as 'electron spin resonance' (ESR) [7]. Magnetic resonance spectroscopy finds much use: medical imaging [9], characterization of chemical structures [10], and protein structure determination [11] are just three.

The dynamics of a spin-1/2 particle are often modelled using a semi-classical

'vector model' [12]. We do not use this simplification here, as we wish to retain the generality offered by quantum mechanical models. It is because we use quantum mechanical models rather than classical models that the title of this thesis refers to *quantum* spin dynamics simulation.

The motivations for simulating magnetic resonance spectra are many-fold; here we mention just two. Simulations may be required to analyse spectra in cases where simple models break down; the presence of strong coupling [8, 12] and sample spinning [13] are two common causes of such a breakdown. Simulations can also be used to improve existing experiments, e.g. by applying optimal control theory to magnetic resonance experiments [14, 15].

This thesis considers how to render simulations of spin dynamics efficient. In the present chapter we describe how spin dynamics simulations are performed and why it is very difficult to simulate the dynamics of large spin systems (> 20 spins). Chapter 2 examines methods that have been developed that allow simulation of the dynamics of large spin systems in the liquid state. Following this, Chapter 3 examines how we can make use of the power of parallel computing to accelerate simulations. The fourth chapter presents a method for dealing with spin systems which interact in a time-varying manner, allowing use of tools developed for time-independent quantum mechanics in time-dependent problems. Contained in the last chapter are descriptions of how simulations of three magnetic resonance experiments have been implemented. We show how we can use the specifics of these experiments, as well as the tools discussed in Chapters 2 and 3, to improve the efficiency of their simulation.

It should be noted that all equations in this thesis are written in terms of units where the reduced Planck constant $\hbar = 1$.

1.1 Spin dynamics

The quantum mechanical systems that we shall deal with in this thesis are those that can be described by a 'spin Hamiltonian'. A spin Hamiltonian is a quantum

mechanical Hamiltonian [6] which consists of operators that operate exclusively on the spin degrees of freedom of the system; we shall call such operators 'spin operators'.

Spin Hamiltonians are formally obtained by a second order perturbation expansion of the full quantum mechanical Hamiltonian with respect to the spin operators, followed by thermal averaging over all other degrees of freedom in the system [16]. The approximations inherent in this definition mean that a spin Hamiltonian does not always exist. This thesis deals only with systems where assertion of a spin Hamiltonian is valid, and so in the following, 'Hamiltonian' will always mean 'spin Hamiltonian' except where otherwise stated. Similarly, 'operator' will be synonymous with 'spin operator'.

An explicit representation which we can use to express the spin Hamiltonian is the subject of Section 1.1.1, and a discussion of terms appearing in the spin Hamiltonians found in this work may be found in Section 1.1.2. Section 1.1.3 defines the matrix that we use to define the 'state' of a spin system, the 'density matrix'.

1.1.1 Spin basis

A vector with $2S_k + 1$ complex elements is sufficient to represent the state of a spin-S_k particle [17, Chapter 6]. These vectors are defined in a space which we shall call 'wavefunction space'. Because this space is a Hilbert space [18], it is commonly called 'Hilbert space' in the magnetic resonance literature [8]. However, in order to avoid confusion with the more general mathematical concept, we do not call it that here. The Hamiltonian of a system maps between elements of wavefunction space; a Hamiltonian acting on a single spin can thus be represented as a $(2S_k+1)\times(2S_k+1)$ matrix of complex elements [19].

A matrix of size $N \times N$ can be represented as a linear combination of N^2 basis operators (e.g. N^2 matrices each with a single 1). We define an inner product [18, 19] for our basis operators, the Frobenius inner product [8, 20, 21]. Given two square matrices A and B of equal size, this inner product is

$$\langle A, B\rangle = \mathrm{tr}\big\{A^\dagger B\big\}, \tag{1.1}$$

where $\langle \cdot, \cdot \rangle$ is the notation that we use to express an inner product, superscript † denotes Hermitian conjugation, and $\mathrm{tr}\{\cdot\}$ denotes taking the trace of the matrix within the braces [19]. The value of this inner product is the complex conjugate of that commonly used in linear algebra, given by $\mathrm{tr}\{B^\dagger A\}$ [20]. The choice of using Equation (1.1) is made to match the usual definition used in magnetic resonance [8, 21]. The basis operators that we utilize in this thesis are chosen to be orthogonal [19] with respect to the Frobenius inner product, i.e. given two basis operators $\hat{S}_i$ and $\hat{S}_j$

$$\left\langle \hat{S}_i, \hat{S}_j \right\rangle = \delta_{ij} \left\langle \hat{S}_i, \hat{S}_i \right\rangle, \tag{1.2}$$

where δ_{ij} is the 'Kronecker delta',

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases} \tag{1.3}$$

The Frobenius inner product induces the Frobenius norm [20],

$$\|A\| = \sqrt{\langle A, A \rangle} = \sqrt{\mathrm{tr}\{A^\dagger A\}}. \tag{1.4}$$

Our basis operators will not in general be normalized, i.e. in general $\|\hat{S}_i\| \neq 1$.

Irreducible spherical tensor operators [22, 23] are a commonly used set of basis operators in magnetic resonance simulations [8]. They are commonly used because they generalize well to simulations containing high-spin particles [8] and their various interrelations are very well tabulated [22, 23]. These operators are those used 'under the hood' of the simulations presented in this thesis [24].

It is, however, usual to express the Hamiltonian of spin-1/2 particles in terms of the longitudinal operator $\hat{S}_z$ and transverse operators $\hat{S}_x$ and $\hat{S}_y$ instead of irreducible spherical tensor operators [8]. We follow this custom to allow direct comparison with relevant literature. Linear combinations of $\hat{S}_x$ and $\hat{S}_y$,

$$\hat{S}_\pm = \hat{S}_x \pm \mathrm{i}\hat{S}_y, \tag{1.5}$$

represent the usual observable operators in phase- and frequency-sensitive) magnetic resonance [12]. These two operators are Hermitian conjugate to each other, as may

be verified from their definition and the fact that $\hat{S}_x$ and $\hat{S}_y$ are Hermitian [23]. The operators $\hat{S}_\pm$ may be used as basis operators instead of $\hat{S}_x$ and $\hat{S}_y$.

While the operators $\hat{S}_x$, $\hat{S}_y$, and $\hat{S}_z$ are sufficient to represent the Hamiltonian of a spin-1/2 particle, representation of more general spin-1/2 operators (e.g. the density matrix, Section 1.1.3) requires inclusion of a fourth operator in the operator basis, bringing the total number of operators to four. The operators $\hat{S}_z$ and $\hat{S}_\pm$ are each proportional to an element in the irreducible spherical tensor operator basis [8]; this implies that the final element in the basis is proportional to the fourth element of the irreducible spherical tensor operator basis. The fourth element of the irreducible spherical tensor operator basis is $\mathbb{I}_2/\sqrt{2}$, a 2×2 identity matrix [25] normalized with respect to the Frobenius norm [8]. We omit the normalization in order to simplify the following algebra; our last basis operator is thus $\mathbb{I}_2$.

The operator basis that we shall use for a system of N spins is the 'product operator basis'. An operator $\hat{O}$ acting upon the kth spin maps to the product operator

$$\hat{O}^{(k)} = \left(\bigotimes_{j=1}^{k-1} \mathbb{I}_{2S_j+1} \right) \otimes \hat{O} \otimes \left(\bigotimes_{l=k+1}^{N} \mathbb{I}_{2S_l+1} \right). \tag{1.6}$$

The operation $\otimes$ denotes the Kronecker product [26], and $\mathbb{I}_M$ is the $M \times M$ identity matrix. Operators defined in this way act on the wavefunction space containing all possible correlations between spin degrees of freedom of the constituent spins of the system. Instead of using the notation of different upper subscripts to denote product operators acting on different spins (e.g. $\hat{I}_z^{(1)}$ and $\hat{I}_z^{(2)}$), we shall occasionally utilize the convention common in NMR [8, 12] of denoting such distinct product operators by different letters (e.g. $\hat{I}_z$ and $\hat{S}_z$).

Product operators acting on more than one spin can be constructed by matrix multiplication of single-spin product operators acting on different spins. This can be shown by utilizing the useful relation involving Kronecker products that for four matrices A, B, C, and D,

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{1.7}$$

when the respective matrices match in size such that the matrix multiplications on each side are defined [26]. In the following, the number of non-identity single-spin operators in a product operator will be referred to as the 'spin-order' of the operator [8]. There is a unique (up to normalization) product operator with spin-order zero; this operator is constructed by taking the Kronecker product of the identity matrices of all of the spins, and will be denoted $\mathbb{I}$ below.

Assuming the spin operator basis of spin k contains $(2S_k + 1)^2$ orthogonal elements, one of which is (an operator proportional to) the identity matrix of appropriate size, $\prod_{k=1}^{N}(2S_k + 1)^2$ unique (up to a scalar pre-factor) product operators can be constructed by formal matrix–matrix multiplication of single-spin operators. The single-spin irreducible spherical tensor basis can be used to construct such a product basis [22, 23], as can the spin-1/2 basis described above.

The orthogonality of the single-spin operators implies that elements of the product operator basis are orthogonal to each other with respect to the Frobenius inner product given above. This follows because the trace of a Kronecker product of two matrices is the product of the traces of each matrix [27]. Respective orthogonality implies linear independence, and thus product operators define a complete basis for representing complex-valued matrices of size $\left(\prod_{k=1}^{N}(2S_k + 1)\right) \times \left(\prod_{k=1}^{N}(2S_k + 1)\right)$ [20]. This result will be utilized in Section 1.3.1 in order to map these product operators into vectors.

1.1.2 Spin Hamiltonian components

Within this thesis spin Hamiltonians are composed of just two sorts of terms: linear one-spin terms resulting from coupling of the magnetic moment of the spin to some external field, and bilinear two-spin terms due to coupling between two distinct spins. We do not discuss couplings between distinct electron spins as the spin systems used in this thesis do not contain more than one electron spin. 'Quadratic' coupling terms that include the same spin twice cannot appear, as we deal only with spin-1/2 particles [8, 28]. The assumptions made in defining the spin Hamiltonian mean that terms

involving three or more spins cannot be present.

We shall define our spin Hamiltonians in 'rotating frames'. This definition allows for a number of simplifications; these are described at appropriate points in this chapter. For Hamiltonians that consist of electrons and nuclei, the rotating frame transformation is only applied to the electrons. This is in keeping with the standard Hamiltonian definitions in ESR spectroscopy [28]. For Hamiltonians consisting entirely of nuclei, we will often make use of 'multiply' rotating frames. Formally, multiple rotating frame transformations are applied to the Hamiltonian, each corresponding to transformation with respect to a different set of spins. At the high magnetic fields we use in our simulations, heteronuclei can be defined in different rotating frames, as can weakly-coupled (see below) homonuclei.

A rotating frame transformation applied to the operator $\hat{O}$ with respect to a set of spins K into a frame rotating with angular frequency ω_0 takes the form [8]

$$\hat{O}_{\text{rot frame}} = \mathrm{e}^{+\mathrm{i}\omega_0 \hat{F}_z^{(K)} t} \hat{O} \mathrm{e}^{-\mathrm{i}\omega_0 \hat{F}_z^{(K)} t}, \tag{1.8}$$

where we define the operator $\hat{F}_z^{(K)}$ by

$$\hat{F}_z^{(K)} = \sum_{k \text{ in } K} \hat{S}_z^{(k)}. \tag{1.9}$$

The summation is taken over all spins of the type that are to be moved into this rotating frame. While we explicitly denote operators defined in the rotating frame by the subscript 'rot frame' here, this subscript is dropped when we define the Hamiltonian terms below; they will be defined in an appropriate rotating frame unless otherwise stated.

The transformation in Equation (1.8) contains two 'matrix exponentials'. A matrix exponential is formally defined by its Taylor series; given a square matrix A this series is [29]

$$\mathrm{e}^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}. \tag{1.10}$$

Matrix exponentials arise frequently in this thesis. This is because they are needed in the formal solution of Equation (1.30); see Section 1.5.

Detection of observables in magnetic resonance spectrometers is usually performed relative to a 'spectrometer reference frequency'; this allows for the detection of frequency differences, which are relatively small compared to the absolute frequencies [12, 30, 31]. This form of detection is equivalent to performing detection of observables in a rotating reference frame rotating at the spectrometer reference frequency [12]. When applicable, we choose ω_0 to be equal to the effective spectrometer reference frequency in the simulations in this thesis in order to agree with experimental protocols.

If $\hat{O}$ commutes with $\hat{F}_z^{(K)}$, then $\hat{O}_{\text{rot frame}} = \hat{O}$. However, in general an operator mapped into the rotating frame in the manner described by Equation (1.8) will pick up time-dependence from the transformation. If it happens that

$$\hat{O}_{\text{rot frame}} = \mathrm{e}^{+\mathrm{i}p\omega_0 t}\hat{O}, \tag{1.11}$$

where p is an integer, we say that $\hat{O}$ has coherence order p [12] with respect to the set of spins K.

Importantly, the operators $\hat{S}_z^{(k)}$ and $\hat{\mathbb{I}}$ have coherence order zero as they both commute with $\hat{F}_z^{(K)}$, the operator $\hat{S}_+^{(k)}$ has coherence order $+1$, and the operator $\hat{S}_-^{(k)}$ has coherence order -1 [12]. These four operators are sufficient for expanding general operators of spin-1/2 particles in terms of operators with defined coherence order. More generally, irreducible spherical tensor operators can be used: elements in the basis representing the kth spin have well-defined coherence order because of known commutation relations of these elements with $\hat{S}_z^{(k)}$ [22, 23].

When it is well defined, the coherence order of the product of two operators is the sum of the coherence orders of the separate operators, as may be shown in the following way. Consider an operator $\hat{P}$ with coherence order p and an operator $\hat{Q}$ with coherence order q. Inserting the product of these operators into Equation (1.8) gives

$$\left(\hat{P}\hat{Q}\right)_{\text{rot frame}} = \mathrm{e}^{+\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\hat{P}\hat{Q}\mathrm{e}^{-\mathrm{i}\omega_0\hat{F}_z^{(K)}t}. \tag{1.12}$$

Noting that

$$\mathrm{e}^{-\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\mathrm{e}^{+\mathrm{i}\omega_0\hat{F}_z^{(K)}t} = \mathbb{I}, \tag{1.13}$$

the identity matrix, we can rewrite the rotating frame product as

$$\left(\hat{P}\hat{Q}\right)_{\text{rot frame}} = \mathrm{e}^{+\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\hat{P}\mathbb{I}\hat{Q}\mathrm{e}^{-\mathrm{i}\omega_0\hat{F}_z^{(K)}t} \tag{1.14}$$

$$= \left(\mathrm{e}^{+\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\hat{P}\mathrm{e}^{-\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\right)\left(\mathrm{e}^{+\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\hat{Q}\mathrm{e}^{-\mathrm{i}\omega_0\hat{F}_z^{(K)}t}\right). \tag{1.15}$$

As the two operators have a well-defined coherence order, we can rewrite this as

$$\left(\hat{P}\hat{Q}\right)_{\text{rot frame}} = \mathrm{e}^{+\mathrm{i}\omega_0 pt}\hat{P}\mathrm{e}^{+\mathrm{i}\omega_0 qt}\hat{Q} \tag{1.16}$$

$$= \mathrm{e}^{+\mathrm{i}\omega_0(p+q)t}\hat{P}\hat{Q}, \tag{1.17}$$

i.e. the product has coherence order $(p+q)$, as we set out to prove. The case in which the resulting product operator does not have a well-defined coherence order is when the product $\hat{P}\hat{Q}$ is the zero matrix. Such a product cannot contribute to a calculation in any meaningful way, however.

At the high magnetic fields usually used in magnetic resonance it is often a good approximation to restrict the rotating frame Hamiltonian to just terms of coherence order zero (with respect to all spins). This is because the magnitude of the frequency ω_0 is typically large enough that terms of the form shown in Equation (1.11) where $p \neq 0$ average to zero over the time scales probed. This restriction is known as the 'secular approximation' [30] or 'high-field approximation' [8] (we use the former designator here), and is utilized in all of the simulations in this thesis.

We define all of the Hamiltonian terms below within the secular approximation, with two exceptions. Firstly, in ESR simulations the Hamiltonian is not defined within a rotating frame with respect to any of the nuclear spins; the secular approximation with respect to the nuclear spins is thus not appropriate in these simulations. Secondly, radio- and microwave-frequency terms are usually rendered (approximately) time-independent by the rotating frame transformation (see below). The averaging argument is then not valid, and these terms are thus kept.

Zeeman interaction

The largest magnitude contributions to the spin Hamiltonians in this thesis are due to coupling of the spin magnetic moments to an external applied magnetic field.

This is the 'Zeeman interaction' [8]. In simple magnetic resonance experiments the applied magnetic field is homogeneously oriented over the entire sample; the direction of this field is conventionally taken as the z axis of the 'laboratory frame' [12]. This field is also assumed to have homogeneous magnitude, B_0, although additional 'field gradients' are commonly applied as part of magnetic resonance experiments [12].

The form of the secular Zeeman term for a spin-1/2 nucleus k in a frame rotating at frequency ω_k is [8]

$$\hat{H}^{(k)}_{\text{Zeeman}} = -B_0\left(1 - \frac{\omega_k}{B_0\gamma_k} - \sigma_{k,z}\right)\gamma_k\hat{S}^{(k)}_z. \tag{1.18}$$

Each nucleus has a specific magnetogyric ratio, γ_k [32]. The chemical shielding, $\sigma_{k,z}$, includes local contributions that modify the exact size of the field felt by the spin [7].

We shall often talk of chemical shift rather than chemical shielding in NMR simulations; chemical shift δ is defined relative to some reference frequency ω_{ref}, and expressed in the secular case as

$$\delta = 10^6\frac{B_0(1-\sigma_{k,z}) - \omega_{\text{ref}}}{\omega_{\text{ref}}}, \tag{1.19}$$

in units of parts-per-million (ppm) [12]. We use the standard NMR definitions of reference frequencies for nuclei [32].

As mentioned above, if a spin system contains both electrons and nuclei, then only the electrons will be moved into the rotating frame. This means that the secular approximation is not applicable to the nuclei. The form of the Zeeman interaction for a nucleus k is then [30]

$$\hat{H}^{(k)}_{\text{Zeeman}} = -B_0\gamma_k\Big(-\sigma_{k,x}\hat{S}^{(k)}_x - \sigma_{k,y}\hat{S}^{(k)}_y + [1-\sigma_{k,z}]\hat{S}^{(k)}_z\Big), \tag{1.20}$$

where $\sigma_{k,x}$, $\sigma_{k,y}$, and $\sigma_{k,z}$ are chemical shielding components.

A slightly different form is conventionally used for secular Zeeman terms of the Hamiltonian due to electrons [7]; for an electron l in a frame rotating at ω_l the secular Zeeman interaction is written [28]

$$\hat{H}^{(l)}_{\text{Zeeman}} = B_0\left(\mu_{\text{B}}g_l - \frac{\omega_l}{B_0}\right)\hat{S}^{(l)}_z. \tag{1.21}$$

The constant μ_{B} appearing in this equation is the Bohr magneton [28]. The 'g-factor', g_l, is similar to the chemical shift of nuclear spins in that it incorporates the effects of fields local to a given spin [7].

In general, $\sigma_{k,x}$, $\sigma_{k,y}$, $\sigma_{k,z}$, and g_l will contain both isotropic and anisotropic contributions: variation of the position and orientation in space of the particle, e.g. by rotation of the molecule which the particle is in, can vary the magnitude of the interaction. However, isotropic tumbling in solution averages all but the isotropic contribution to zero [30]. Magic angle spinning (see Section 4.4) can be used to render the anisotropic component of the secular Zeeman interaction unobservable in the solid state [33].

Radio- and microwave-frequency field interaction

Radiofrequency (NMR and ESR) and microwave-frequency (ESR) fields are used to perturb the dynamics of spin systems [8, 28]. Within the pulse sequences in this thesis, these fields are applied in the form of short 'pulses' in the xy-plane relative to the direction of the applied magnetic field.

A pulse applied at a frequency ω_1 with magnitude B_1 to spin-1/2 particles of type K defined in a frame rotating at ω_1 gives the Hamiltonian term [8]

$$\hat{H}^{(K)}_{\mathrm{pulse}} = -B_1 \sum_{k \in K} \gamma_k \Big(\cos(\phi) \hat{S}^{(k)}_x + \sin(\phi) \hat{S}^{(k)}_y \Big), \tag{1.22}$$

where γ_k should be replaced with $-g_k \mu_{\mathrm{B}}$ in the case of electron-spin excitation [28], and ϕ is the 'phase' of the pulse. Common pulse-phases include 0 (an 'x' pulse), $\pi/2$ (a 'y' pulse), π (a '$-x$' pulse) and $3\pi/2$ (a '$-y$' pulse). An additional (time-dependent) term that arises during the rotating frame transformation has been omitted as it only contributes a small frequency shift; this is standard practice in high field magnetic resonance simulations [8, 28]. We shall refer to this omission as the 'rotating frame approximation'. Use of multiple rotating frames allows us to express pulses applied at different frequencies to different sets of spins in this time-independent form.

In principle, the summation in Equation (1.22) should be over all spins in the spin

system. In practice, we do not include spins that are defined in a different rotating frame within the Hamiltonian. This is because the terms corresponding to such spins are not time-independent, and are assumed to be averaged to zero within the secular approximation.

In NMR [30] and ESR [31] experiments, if a set of spins is detected, it is usual for the excitation frequency of the pulse applied to that set of spins, ω_1, to be equal to the rotating frame frequency used for detection of that set of spins, ω_0. When appropriate, we have thus always chosen $\omega_1 = \omega_0$ in the simulations in this thesis for consistency with experimental protocols.

Nuclear spin–nuclear spin couplings

Spin–spin couplings between nuclei arise via two different mechanisms, namely J-coupling, and dipole–dipole coupling [8, 16, 30], which we shall now discuss. With the definitions of these interactions given below, care should be taken not to 'double-count' the interactions by including any pair of spins twice in the final Hamiltonian.

The J-coupling interaction between spins is mediated by the electrons in chemical bonds [30]. This coupling provides much of the information obtained in NMR spectroscopy [16, 30].

In the homonuclear case, the secular part of the J-coupling between two nuclei k and l takes the form [30]

$$\hat{H}^{(k,l)}_{J\text{ coupling}} = 2\pi J_{kl}\Big(\hat{S}^{(k)}_x\hat{S}^{(l)}_x + \hat{S}^{(k)}_y\hat{S}^{(l)}_y + \hat{S}^{(k)}_z\hat{S}^{(l)}_z\Big). \tag{1.23}$$

Between heteronuclei this reduces to [30]

$$\hat{H}^{(k,l)}_{J\text{ coupling}} = 2\pi J_{kl}\hat{S}^{(k)}_z\hat{S}^{(l)}_z \tag{1.24}$$

because $(\hat{S}^{(k)}_x\hat{S}^{(l)}_x + \hat{S}^{(k)}_y\hat{S}^{(l)}_y)$ does not have coherence order zero in the heteronuclear case, and so is discarded as part of the secular approximation (see above). The prefactor J_{kl} gives the size of the coupling between the two spins. We assume that all J_{kl} are isotropic in this thesis. While formally J_{kl} can contain anisotropic components,

we ignore them as they are often small and hard to separate from the dipole–dipole interaction described below [30].

Between every pair of nuclear spins there is a through space interaction known as 'dipole–dipole coupling'. This coupling arises from the interaction of the magnetic moment of one nucleus with the magnetic field generated by the other [16]. The coupling magnitude depends upon the distance between the two nuclei and their orientation with respect to each other; the orientation is conventionally defined in the laboratory frame.

The secular dipole–dipole coupling between two homonuclei l and k is [8, 30]

$$\hat{H}^{(k,l)}_{\text{DD coupling}} = \frac{1}{2} b_{kl}\big(3\cos^2(\theta_{kl}) - 1\big) \cdot \cdot \Big(2\hat{S}^{(k)}_z \hat{S}^{(l)}_z - \hat{S}^{(k)}_x \hat{S}^{(l)}_x - \hat{S}^{(k)}_y \hat{S}^{(l)}_y\Big), \tag{1.25}$$

and for heteronuclei it is [8, 30],

$$\hat{H}^{(k,l)}_{\text{DD coupling}} = b_{kl}\big(3\cos^2(\theta_{kl}) - 1\big)\hat{S}^{(k)}_z \hat{S}^{(l)}_z. \tag{1.26}$$

The angle θ_{kl} is the angle between the magnetic field vector and the vector joining the two nuclei. The prefactor b_{kl} is a function of the inter-spin distance, r_{kl}, and the magnitude of the magnetic moments of the two nuclei involved in the interaction. Explicitly, this factor is [30]

$$b_{kl} = -\frac{\mu_0}{4\pi}\frac{\gamma_k \gamma_l}{r^3_{kl}}. \tag{1.27}$$

The dipole–dipole coupling (and thus its secular part) has no isotropic component and is averaged to zero by isotropic tumbling in solution [30]. This coupling can, however, give rise to a mechanism of relaxation (Section 1.3.2) in solution, thus affecting the spectrum indirectly [34]. Dipole-dipole coupling can be observed directly in solid state spectra (an example is given in Figure 4.1), but can be rendered unobservable by magic-angle spinning [33]; see Section 4.4.

In the case that the chemical shifts of two homonuclei are substantially different as compared to the magnitude of the coupling between them, Equations (1.24) and (1.26) may be used to describe their interaction rather than Equation (1.23)

and (1.25). This is the 'weak coupling' regime [8] and can simplify analytical solution of the equations of motion. It is important to note that if one tries to define the Hamiltonian within separate rotating frames for two strongly-coupled spins, the Hamiltonian becomes time-dependent. However, two weakly coupled spins can be defined in different rotating frames within the Hamiltonian, while the Hamiltonian remains time-independent within the secular approximation. This is important in Section 5.3, where two different sets of ^{13}C nuclei are irradiated independently.

Hyperfine couplings

Spin–spin couplings between an electron and a nucleus are utilized in the simulations of Section 5.1. These couplings are known as 'hyperfine couplings' [28]. Details on the mechanisms of interaction may be found in References [7], [16], and [28]. The form of the hyperfine coupling term between a nuclear spin I and an electron spin S can be written as [28]

$$\hat{H}_{\mathrm{HF}}^{(I,S)} = \Big(B_x\hat{I}_x + B_y\hat{I}_y + A\hat{I}_z\Big)\hat{S}_z, \tag{1.28}$$

where we use the secular approximation for the electron spin, and A, B_x and B_y are coupling parameters. These coupling parameters will in general have both isotropic and anisotropic parts.

1.1.3 Density matrices

While the state of a single N-spin system can be described in the product basis by a vector with $\prod_{k=1}^{N}(2S_k+1)$ elements, the state of an ensemble of systems is best described by a 'density matrix' [35]. We make use of the density matrix formalism in this thesis.

The expectation value [35] of an observable operator $\hat{s}$ is obtained from the density matrix of the system, $\hat{\rho}$, using the relation [8]

$$\langle\hat{s}\rangle = \mathrm{tr}\{\hat{s}\hat{\rho}\}. \tag{1.29}$$

While density matrices can be formally defined in terms of an ensemble of wavefunctions, we choose to let this relationship formally define our density matrix, following

Reference [36].

The size of $\hat{\rho}$ is the same as the Hamiltonian of the system, it is Hermitian, and it has unit trace [36]. The equation of motion of $\hat{\rho}$ is the Liouville–von Neumann equation [8, 36]

$$\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}t} = -\mathrm{i}\Big[\hat{H}, \hat{\rho}\Big], \tag{1.30}$$

where $[\,\cdot\,,\,\cdot\,]$ is a commutator [29]; expanding the commutator this equation is written

$$\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}t} = -\mathrm{i}\Big(\hat{H}\hat{\rho} - \hat{\rho}\hat{H}\Big). \tag{1.31}$$

Solution of this equation is discussed in Section 1.5.

Usually we shall take the density matrix at the beginning of a simulation to be the density matrix representing thermal equilibrium. At a temperature T, this density matrix is given by [36]

$$\hat{\rho}_{\mathrm{eq}} = \frac{\exp\{-\beta\hat{H}\}}{\mathrm{tr}\Big\{\exp\{-\beta\hat{H}\}\Big\}}, \tag{1.32}$$

where $\beta = (\mathrm{k_B}T)^{-1}$, and $\mathrm{k_B}$ is Boltzmann's constant.

The thermal equilibrium density matrix in Equation (1.32) can be computed using the Taylor series expansion of the matrix exponential, Equation (1.10). Convergence of this expansion to machine precision ($\sim 10^{-16}$ [37]) typically occurs within only a few terms. The number of terms required is so low as a result of the smallness of the energy gaps in the spin Hamiltonian compared to β^{-1} at usual temperatures and magnetic field strengths. This observation gives rise to the 'high temperature approximation', common in magnetic resonance, that truncates the Taylor series expansion of Equation (1.32) after the second term [8, 28], i.e.

$$\hat{\rho}_{\mathrm{eq}} \approx \frac{\mathbb{I}_{\dim\{\hat{H}\}} - \beta\hat{H}}{\mathrm{tr}\Big\{\mathbb{I}_{\dim\{\hat{H}\}} - \beta\hat{H}\Big\}}. \tag{1.33}$$

The smallness of the energy gaps also means that this density matrix is dominated by the identity matrix term. Since solutions of Equation (1.30) preserve the trace, it is often convenient to omit the identity matrix from the thermal equilibrium density

matrix and formally renormalize to give the 'reduced density matrix'

$$\hat{\rho}'_{\text{eq}} = -\hat{H}. \tag{1.34}$$

This renormalization increases the numerical values of the populations as compared to those found in the density matrix of Equation (1.33), while retaining the ratios of these populations. Increasing the absolute population values is beneficial as it can help avoid numerical errors due to the finite precision arithmetic utilized in numerical calculation [25, 37]. As the Zeeman interaction is the largest magnitude interaction in the Hamiltonians used in this thesis (see Section 1.1.2), the Zeeman term will dominate the reduced density matrix.

1.2 Exponential scaling problem

It is well-known that the computational resources required to simulate general quantum mechanical systems scale exponentially with the number of particles [38]. General spin dynamics simulations are no exception.

As an example of the resources required, we can consider the storage required for a general density matrix. We first note that spin-1/2 particles have the smallest density matrix representation (a 2×2 matrix); the case of N spin-1/2 particles thus provides a lower bound on representation size. The Kronecker product of N such matrices, as required to represent an N-spin system, is a $2^N \times 2^N$ matrix, i.e. the matrix size scales exponentially with the number of spins in the system. This provides a lower bound on the size of the density matrix required to represent a general N-spin system.

This exponential scaling of computational resources with the number of spins, or 'exponential scaling problem', is severely limiting: a simulation with just 20 spins would require a density matrix with size of the order $10^6 \times 10^6$. Actual spin systems of interest can have thousands of spins (see Section 5.3).

Through the use of additional approximations and judicious choices of experiment, magnetic resonance experiments can be used to analyse spin systems with hundreds and even thousands of spins [11, 39]. These approximations are not always justified,

however, and could lead to very large errors. Chapter 2 summarizes recently developed methods that can mitigate (and even overcome) the exponential scaling problem in simulations of liquid-state magnetic resonance experiments with the introduction of only very small errors.

1.3 Liouville space

It is often convenient to work with the 'Liouville space' representation of the density matrix [8, 21]. In this representation a density matrix becomes a vector with the same number of elements as the density matrix. We refer to this vectorized density matrix, $|\hat{\rho}\rangle$, as a 'state vector'. More generally, an operator $\hat{O}$ that has been mapped into Liouville space to give $|\hat{O}\rangle$ will be referred to as either a 'vectorized operator' or, more colloquially, a 'state'. Operators which map between elements of Liouville space are known as 'superoperators' [8, 21]. They are represented by square matrices with size the square of operators on wavefunction space. Superoperators are denoted in this thesis by upper-case letters in calligraphic font, e.g. $\mathcal{L}$, $\mathcal{P}$, $\mathcal{R}$, $\mathcal{S}$.

After the discussion of Section 1.2, the concept of mapping into a space which requires even larger matrix representations may seem ridiculous. However, use of Liouville space conveys a number of benefits. Firstly, it simplifies the inclusion of relaxation theories; this is discussed in Section 1.3.2. Secondly, and most importantly, the formalism allows for the use of 'state space restriction' methodology. As detailed in Chapter 2, such methodology allows us to overcome the exponential scaling problem in special cases.

The representation of Liouville space used in spin dynamics is commonly bestowed with the following inner product [18, 19]: between any two vectorized operators $|\hat{a}\rangle$ and $|\hat{b}\rangle$

$$\left\langle \hat{a} \middle| \hat{b} \right\rangle = (|\hat{a}\rangle)^{\dagger} \left| \hat{b} \right\rangle = \sum_{k} ([|\hat{a}\rangle]_k)^{*} \left[\left| \hat{b} \right\rangle \right]_k, \tag{1.35}$$

where superscript † denotes Hermitian conjugation, superscript $*$ denotes complex conjugation, and subscripted square brackets $[\,\cdot\,]_k$ denote the kth element of the vector

contained within them [8, 21]. This inner product is defined to be equivalent to the Frobenius inner product, Equation (1.1) [8, 21]

$$\left\langle \hat{a} \middle| \hat{b} \right\rangle = \mathrm{tr}\left\{ \hat{a}^\dagger \hat{b} \right\}. \tag{1.36}$$

Unfortunately, this inner product definition differs subtly from the definition of the expectation value [8]. Comparing Equations (1.29) and (1.35), the expectation value of an observable $\hat{s}$ can be seen to be given in inner product form by

$$\langle \hat{s} \rangle = \left\langle \hat{s}^\dagger \middle| \hat{\rho} \right\rangle. \tag{1.37}$$

We cannot use the expectation value to define our inner product, as it is does not fulfil the required criterion of being non-negative [20]. In the case of Hermitian observable operators, $\hat{s} = \hat{s}^\dagger$ and thus $\langle \hat{s} \rangle = \langle \hat{s} | \hat{\rho} \rangle$ [8]: there is no difference between expectation values and inner products. However, it is usual in magnetic resonance to detect using the non-Hermitian observable operators $\hat{S}_\pm$ (Section 1.1.1), meaning that the distinction between inner products and expectation values is important [8, 12]. We shall refer to a vectorized operator $\langle \hat{s}^\dagger |$ which will be used to compute the expectation value of $\hat{s}$ as a 'detection state'.

The Liouville–von Neumann equation in Liouville space is expressed in terms of a superoperator called the 'Liouvillian', denoted by $\mathcal{L}$. This Liouvillian has the property that [8]

$$\mathcal{L}|\hat{\rho}\rangle = \left| \left[\hat{H}, \hat{\rho} \right] \right\rangle. \tag{1.38}$$

Examining Equation (1.30), the Liouville–von Neumann equation in Liouville space is seen to be [21]

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}\rangle = -\mathrm{i}\mathcal{L}|\hat{\rho}\rangle. \tag{1.39}$$

One commonly used mapping into Liouville space is to define a vectorized operator $|\hat{O}\rangle$ by extending the matrix $\hat{O}$ into a column-vector row-wise. The Liouvillian operating on elements of this Liouville space can be shown to be related to the Hamiltonian of the system by [8, 21]

$$\mathcal{L} = \hat{H} \otimes \mathbb{I} - \mathbb{I} \otimes \hat{H}^{\mathrm{T}}, \tag{1.40}$$

where a superscript T denotes taking of the matrix transpose [19] and the identity matrices are the same size as the Hamiltonian. The direct correspondence to the original density matrix and Hamiltonian is convenient, but this is often not the best way to define the representation. This is especially true when making use of basis set truncation (Section 2.1). The mapping used for the Liouville space simulations presented in this thesis is sketched-out in Section 1.3.1. Use of this mapping to generate the thermal equilibrium state vector directly in Liouville space may be found in Section 1.3.3.

The vast majority of the simulations in this thesis have been computed using a Liouville space formalism. The only exceptions may be found in Section 3.3.

1.3.1 Building superoperators

While it is possible to build operators acting on wavefunction space and then map them into superoperators and state vectors [8], it is often more convenient to build superoperators and state vectors in Liouville space directly. This direct construction is especially useful when using basis set truncation (see Chapter 2), as the vectors and matrices can be built directly in the truncated basis [40].

Reference [40] contains a general scheme for building superoperators without having to build product operators that act on wavefunction space. This subsection contains application of this method to construct the superoperator that effects multiplication of $\hat{H}$ on the left of an operator $\hat{O}$. The superoperator effecting multiplication on the right can be constructed similarly; both superoperators are required to build the Liouvillian, as may be confirmed by observation of the commutator structure in Equation (1.30). The details of the construction are important for Section 1.3.3, where we construct thermal equilibrium state vectors directly in Liouville space.

We begin with expansion of $\hat{O}$ in terms of a product basis as described in Section 1.1.1. The nth element of this basis is here denoted $\hat{S}_n$, and it should be recalled that these elements are orthogonal with respect to the Frobenius inner product, Equa-

tion (1.1). Expanded in this basis $\hat{O}$ may be written

$$\hat{O} = \sum_l o_l \hat{S}_l. \tag{1.41}$$

It is convenient to use a normalized basis; this is required such that the superoperator we derive is Hermitian (see below). Normalization is performed using the Frobenius norm, Equation (1.4), to give normalized basis elements

$$\hat{S}'_l = \hat{S}_l \left\| \hat{S}_l \right\|^{-1}. \tag{1.42}$$

With the normalized basis, the expansion of $\hat{O}$, Equation (1.41), becomes

$$\hat{O} = \sum_l \left(\left\| \hat{S}_l \right\| o_l \right) \hat{S}'_l. \tag{1.43}$$

The product of $\hat{H}$ and $\hat{O}$ is then written

$$\hat{H}\hat{O} = \sum_l \hat{H} \left\| \hat{S}_l \right\| o_l \hat{S}'_l. \tag{1.44}$$

The Hamiltonian can also be expanded in the product operator basis [40], although in order to avoid unnecessary extra subscripts this expansion is not performed explicitly here.

Product operators, as defined in Section 1.1.1, provide a complete basis. Therefore Equation (1.44) can be written

$$\hat{H}\hat{O} = \sum_{kl} \hat{S}'_k h_{kl} \left\| \hat{S}_l \right\| o_l. \tag{1.45}$$

Using the orthonormality of the product operators, the coefficients h_{kl} may be found by taking inner products:

$$h_{kl} = \left\langle \hat{S}'_k, \hat{H}\hat{S}'_l \right\rangle. \tag{1.46}$$

With respect to the original (unnormalized) basis these elements are given by

$$h_{kl} = \left\| \hat{S}_k \right\|^{-1} \left\| \hat{S}_l \right\|^{-1} \mathrm{tr}\left\{ \hat{S}_k^\dagger \hat{H} \hat{S}_l \right\}. \tag{1.47}$$

The expansion in Equation (1.45) implies that we may express the product $\hat{H}\hat{O}$ in terms of a vector with kth element

$$\left[\hat{H}\hat{O} \right]_k = \sum_{kl} h_{kl} \left\| \hat{S}_l \right\| o_l. \tag{1.48}$$

This equation has the form of a matrix–vector product [25]

$$\mathcal{L}_{\text{left}}\left|\hat{O}\right\rangle = \left|\hat{H}\hat{O}\right\rangle, \tag{1.49}$$

where the (kl)th element of the matrix $\mathcal{L}_{\text{left}}$ is

$$[\mathcal{L}_{\text{left}}]_{kl} = h_{kl} \tag{1.50}$$

and the lth element of the vector $|\hat{O}\rangle$ is

$$\left[\left|\hat{O}\right\rangle\right]_l = \left\|\hat{S}_l\right\| o_l. \tag{1.51}$$

This is the desired matrix representation. Because of the Hermiticity of the Hamiltonian, it may be seen from Equation (1.47) that

$$h_{kl} = (h_{lk})^*, \tag{1.52}$$

i.e. $\mathcal{L}_{\text{left}}$ is Hermitian, as was alluded to above.

The definition in Equation (1.51) preserves inner products, as we shall now show. With that definition of the elements, the inner product (see Equation (1.35)) of vectorized operators $|\hat{a}\rangle$ and $|\hat{b}\rangle$ is given by

$$\left\langle \hat{a}\middle|\hat{b}\right\rangle = \sum_k \left\|\hat{S}_k\right\|^2 a_k^* b_k = \sum_k \text{tr}\left\{\hat{S}_k^\dagger \hat{S}_k\right\} a_k^* b_k. \tag{1.53}$$

Linearity of the trace [19] allows us to write

$$\left\langle \hat{a}\middle|\hat{b}\right\rangle = \text{tr}\left\{\sum_k \left(a_k \hat{S}_k\right)^\dagger b_k \hat{S}_k\right\}. \tag{1.54}$$

Orthogonality of the spin operators $\hat{S}_k$ with respect to the Frobenius inner product then means that we can use Equation (1.41) to re-express this as

$$\left\langle \hat{a}\middle|\hat{b}\right\rangle = \text{tr}\left\{\left[\sum_k \left(a_k \hat{S}_k\right)^\dagger\right]\left[\sum_l b_l \hat{S}_l\right]\right\} = \text{tr}\left\{\hat{a}^\dagger \hat{b}\right\}, \tag{1.55}$$

the definition of the Liouville space inner product, Equation (1.36).

The right superoperator $\mathcal{L}_{\text{right}}$ can be derived in this basis by following the above procedure beginning with the product $\hat{O}\hat{H}$ instead of $\hat{H}\hat{O}$. The Liouvillian is then defined by [40]

$$\mathcal{L} = \mathcal{L}_{\text{left}} - \mathcal{L}_{\text{right}} \tag{1.56}$$

such that

$$\mathcal{L}\Big|\hat{O}\Big\rangle = \Big|\Big[\hat{H}, \hat{O}\Big]\Big\rangle. \tag{1.57}$$

The coefficients h_{kl} can be computed efficiently and stored for later computation, as detailed in References [40], [41], and [42] This allows for efficient construction of the Liouvillian, even in the truncated basis sets discussed in Section 2.1.

In Section 1.3.3 we utilize relations found above to demonstrate how the thermal equilibrium state can be generated efficiently in this basis.

1.3.2 Relaxation

'Relaxation' is a blanket term used to denote processes that return a perturbed system to thermal equilibrium. In Liouville space it is possible to include many relaxation models by the introduction of a 'relaxation superoperator' $\mathcal{R}$ into the Liouville–von Neumann equation [8, 21, 34],

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}\rangle = (-\mathrm{i}\mathcal{L} + \mathcal{R})|\hat{\rho}\rangle. \tag{1.58}$$

This corresponds to a simple modification of the Liouvillian, leaving the form of the equation of motion unchanged. Such relaxation models cannot generally be included in treatments utilizing wavefunction space operators without complicating the form of the equation of motion, as we demonstrate using a simple example below. Being able to include relaxation is important, as it allows for the use of basis set truncation (Section 2.1) in a rigorous manner [2].

One of the simplest relaxation theories is T_1 / T_2 theory. This theory consists of the assumption that relaxation processes cause exponential decay of the state populations of spin-1/2 particles, with the population of longitudinal states decaying with

characteristic time T_1 and the population of transverse states decaying with characteristic time T_2 [43]. Using this model, we can examine why inclusion of relaxation is difficult when performing simulations using density matrices as opposed to state vectors.

The density matrix of a single spin-1/2 particle can be written

$$\hat{\rho}_t = \frac{1}{\sqrt{2}}\mathbb{I}_2 + x_t\hat{I}_x + y_t\hat{I}_y + z_t\hat{I}_z. \tag{1.59}$$

Assuming we are in a frame rotating such that the effect of the Zeeman interaction is nulled, T_1 / T_2 theory implies that the equation of motion of this density matrix is

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\rho}_t = -\frac{1}{T_2}\Big(x_t\hat{I}_x + y_t\hat{I}_y\Big) - \frac{1}{T_1}z_t\hat{I}_z. \tag{1.60}$$

The right-hand side of this equation cannot be written in the form of a commutator as in Equation (1.30); the inclusion of relaxation thus results in an equation of motion of a different form to the Liouville–von Neumann equation. Such equations will become more complicated as further spins are included in the simulation.

In contrast, in Liouville space we can write Equation (1.60) as

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}_t\rangle = &-\frac{1}{T_2}\left(\frac{|\hat{I}_x\rangle\langle\hat{I}_x|}{\langle\hat{I}_x|\hat{I}_x\rangle}x_t\Big|\hat{I}_x\Big\rangle + \frac{|\hat{I}_y\rangle\langle\hat{I}_y|}{\langle\hat{I}_y|\hat{I}_y\rangle}y_t\Big|\hat{I}_y\Big\rangle\right)\\ &-\frac{1}{T_1}\frac{|\hat{I}_z\rangle\langle\hat{I}_z|}{\langle\hat{I}_z|\hat{I}_z\rangle}z_t\Big|\hat{I}_z\Big\rangle,\end{aligned} \tag{1.61}$$

which simplifies to the form of Equation (1.58), with $\mathcal{L} = 0$ and

$$\mathcal{R} = -\frac{1}{T_2}\left(\frac{|\hat{I}_x\rangle\langle\hat{I}_x|}{\langle\hat{I}_x|\hat{I}_x\rangle} + \frac{|\hat{I}_y\rangle\langle\hat{I}_y|}{\langle\hat{I}_y|\hat{I}_y\rangle}\right) - \frac{1}{T_1}\frac{|\hat{I}_z\rangle\langle\hat{I}_z|}{\langle\hat{I}_z|\hat{I}_z\rangle}. \tag{1.62}$$

This is illustrative of the general case: in the basis described in Section 1.3.1, inclusion of T_1 / T_2 theory is effected by a diagonal relaxation superoperator that when exponentiated gives the appropriate exponential decays.

Bloch–Wangsness–Redfield theory [44, 45] represents a more sophisticated relaxation model. It can be used to represent the major relaxation mechanisms arising due to tumbling in solution [34, 46]; we use it for this purpose. The model is parameterized by an isotropic rotational correlation time, representing the ‘tumbling

rate' [34]. Simulations using Bloch–Wangsness–Redfield theory in this thesis utilize relaxation superoperators constructed using an implementation due to Ilya Kuprov, described in Reference [47]. We make use of the secular approximation (Section 1.1.2) in the definition of this superoperator, with the same justification as was used for the Hamiltonian.

The relaxation superoperators described here will damp the populations of states to zero, not the true thermal equilibrium state [48]. We ignore this effect in many simulations in this thesis, but methods to overcome this problem are discussed in Section 1.3.3.

1.3.3 Equilibrium state vector

The Hamiltonian of a system is directly related to the energy [6]. This is not true of the Liouvillian of a system. This means that it cannot be used in a manner similar to Equation (1.32) to determine the thermal equilibrium state of the system [21]. One could build this equilibrium state using Equation (1.32) and then compute the vector elements in Equation (1.51) by taking inner products, but this will not be efficient. We would also not be able to make use of basis set truncation during the construction. A better way of building this equilibrium state is through use of the method described below.

For the rest of this section, '$\mathbb{I}$' represents a product operator built from the identity matrix elements of the operator basis of each spin in the system. This operator maps into the vectorized operator $|\mathbb{I}\rangle$, which we call the 'unit state'. Acting on this state with $\mathcal{L}_{\text{left}}$ gives

$$\mathcal{L}_{\text{left}}|\mathbb{I}\rangle = \left|\hat{H}\mathbb{I}\right\rangle = \left|\hat{H}\right\rangle, \tag{1.63}$$

by construction (See Section 1.3.1). The state thus defined is the Hamiltonian represented as a vectorized operator. Further multiplication of $\mathcal{L}_{\text{left}}$ on the left of Equation (1.63) will result in $|\hat{H}^2\rangle$, a further multiplication will give $|\hat{H}^3\rangle$, and so on. This

suggests that we can compute the numerator of Equation (1.32) in Liouville space by

$$\left|\exp\{-\beta\hat{H}\}\right\rangle = \exp\{-\beta\mathcal{L}_{\text{left}}\}|\mathbb{I}\rangle. \tag{1.64}$$

The inner product of Liouville space, Equation (1.55), implies that

$$\left\langle\mathbb{I}\middle|\hat{b}\right\rangle = \text{tr}\Big\{\mathbb{I}^{\dagger}\hat{b}\Big\} = \text{tr}\Big\{\hat{b}\Big\}. \tag{1.65}$$

Therefore the denominator of Equation (1.32) can be computed using

$$\text{tr}\Big\{\exp\{-\beta\hat{H}\}\Big\} = \langle\mathbb{I}|\exp\{-\beta\mathcal{L}_{\text{left}}\}|\mathbb{I}\rangle. \tag{1.66}$$

The thermal equilibrium state in Liouville space may thus be constructed as

$$|\hat{\rho}_{\text{eq}}\rangle = \frac{\exp\{-\beta\mathcal{L}_{\text{left}}\}|\mathbb{I}\rangle}{\langle\mathbb{I}|\exp\{-\beta\mathcal{L}_{\text{left}}\}|\mathbb{I}\rangle}. \tag{1.67}$$

As was explained in Section 1.1.3, only a small number of Taylor series terms will normally be required to generate the state vector in Equation (1.67). The same arguments given there can be used to rationalize the following high temperature approximation to the equilibrium state:

$$|\hat{\rho}_{\text{eq}}\rangle \approx \frac{|\mathbb{I}\rangle - \beta\mathcal{L}_{\text{left}}|\mathbb{I}\rangle}{\langle\mathbb{I}|\mathbb{I}\rangle - \beta\langle\mathbb{I}|\mathcal{L}_{\text{left}}|\mathbb{I}\rangle}. \tag{1.68}$$

The 'reduced state vector' equivalent to the reduced density matrix of Equation (1.34) is

$$\left|\hat{\rho}'_{\text{eq}}\right\rangle = -\mathcal{L}_{\text{left}}|\mathbb{I}\rangle. \tag{1.69}$$

This thermal equilibrium 'state vector' is that which is used whenever we do not give an explicit temperature for a simulation.

The method of building the equilibrium state described in this subsection is very similar to that described in Reference [21], where the equilibrium state is built using

$$|\hat{\rho}_{\text{eq}}\rangle = \frac{\exp\{-\beta(\mathcal{L}_{\text{left}} + \mathcal{L}_{\text{right}})/2\}|\mathbb{I}\rangle}{\langle\mathbb{I}|\exp\{-\beta(\mathcal{L}_{\text{left}} + \mathcal{L}_{\text{right}})/2\}|\mathbb{I}\rangle}. \tag{1.70}$$

Use of either method must necessarily give the same result.

It was noted in Section 1.3.2 that relaxation superoperators derived using T_1 / T_2 theory and Bloch–Wangsness–Redfield theory decay state populations towards zero, not thermal equilibrium. In order to ensure that relaxation is to thermal equilibrium, Equation (1.58) is commonly modified to give [8, 48, 49]

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}\rangle = -\mathrm{i}\mathcal{L}|\hat{\rho}\rangle + \mathcal{R}(|\hat{\rho}\rangle - |\hat{\rho}_{\mathrm{eq}}\rangle). \tag{1.71}$$

Unfortunately, this modified equation is inhomogeneous, making efficient solution more difficult.

A number of methods have been described in the literature that modify $\mathcal{R}$ in order to relax to thermal equilibrium without destroying the homogeneous form of Equation (1.58) [21, 48, 49]. In this thesis we utilize a modification of the simple method suggested in Section 3 of Reference [49] whereby the column of the relaxation superoperator corresponding to the unit state (normally all zeroes) is replaced with [24]

$$-\mathcal{R}|\hat{\rho}_{\mathrm{eq}}\rangle. \tag{1.72}$$

Solutions obtained using this modified relaxation superoperator in Equation (1.58) are equivalent to solutions obtained from solving Equation (1.71) as long as the value of the unit-state element of the state vector $|\hat{\rho}\rangle$ is unity [24]. Relaxation superoperator thermalization has been used in the simulation presented in Section 2.1.

The value of the element corresponding to the unit state in the state vectors defined above will not be unity for state vectors corresponding to density matrices; instead, the element has value $\|\mathbb{I}\|^{-1}$. This may be confirmed by recalling that the density matrix must have unit trace (Section 1.1), which implies that the prefactor in front of the identity operator in the decomposition Equation (1.41) is

$$(\mathrm{tr}\{\mathbb{I}\})^{-1} = (\mathrm{tr}\{\mathbb{I}^{\dagger}\mathbb{I}\})^{-1} = \|\mathbb{I}\|^{-2}, \tag{1.73}$$

and recalling the definition of an element in the Liouville space vector, given by Equation (1.51). However, we can define a modified state vector where this element is unity:

$$|\hat{\rho}'\rangle = \|\mathbb{I}\||\hat{\rho}\rangle. \tag{1.74}$$

In order to preserve expectation values when using this modified state vector, we must also modify the definition of the expectation value, Equation (1.37), such that

$$\langle \hat{s} \rangle = \|\mathbb{I}\|^{-1} \langle \hat{s}^\dagger \big| \hat{\rho}' \rangle. \tag{1.75}$$

1.4 Sparsity

The discussion of Section 1.2 paints a grim picture. Assuming each element requires only a byte of memory (a significant underestimate), storage of the Hamiltonian of a 20 spin-1/2 system would require 4^{20} bytes of computer memory.

Fortunately, spin Hamiltonians and Liouvillians of large spin systems are sparse [3, 50, 51]: the number of non-zero elements is small as compared to the total number of elements of the Hamiltonian. We prove in Sections 1.4.1 and 1.4.2 that the restriction that only at most two-spin spin–spin couplings exist in the spin Hamiltonian (Section 1.1.2) is sufficient to explain this sparsity, and place numerical bounds on the number of non-zero elements that can be present.

Sparse matrices may be stored using much less computer memory than the size of the matrix would suggest. The large number of zero-elements allows for compression: storage of e.g. just the non-zero elements and their indices can require significantly less memory than the full matrix. A large number of possible methods of representing sparse matrices in a compressed way exist [52, 53]. The simulations in this thesis make use of that invoked by MATLAB's sparse matrix library [54, 55].

Efficient methods exist for performing common matrix operations utilizing sparse matrices [52, 53, 54]. Some of these methods are discussed in Section 2.2.3.

1.4.1 Hamiltonian sparsity

That the Hamiltonian contains only terms coupling at most two spins means that a general operator appearing in the Hamiltonian may be written

$$\mathbb{I}_{2S_1+1} \otimes \cdots \otimes \hat{S}_k \otimes \cdots \otimes \hat{S}_m \otimes \cdots \otimes \mathbb{I}_{S_N+1}, \tag{1.76}$$

where $\hat{S}_k$ and $\hat{S}_m$ are operators from the operator bases of the kth- and mth-spins, respectively. All omitted matrices are identity matrices; an omitted matrix in position j is an identity matrix of dimension $(2S_j + 1)$.

Consider the operator

$$\mathbb{I}_{2S_1+1} \otimes \cdots \otimes J_{2S_k+1} \otimes \cdots \otimes J_{2S_m+1} \otimes \cdots \otimes \mathbb{I}_{S_N+1}, \tag{1.77}$$

where J_n is a matrix of size $n \times n$ with no zeroes and the omitted matrices are again identity matrices of appropriate size. Any non-zero element of the operator in Equation (1.76) must be a non-zero element of Equation (1.77); it represents the densest form that a two-spin operator could take. The number of non-zeroes of an operator of the form Equation (1.77) is thus bounded by the number of non-zeroes in the operator of Equation (1.77), which is

$$(2S_k + 1)(2S_m + 1) \prod_{n=1}^{N} (2S_n + 1). \tag{1.78}$$

This value is derived by noting that the number of non-zeroes of the nth identity matrix is $2S_n + 1$, the number of non-zeroes of a matrix J_n is $(2S_n + 1)^2$, and the number of non-zeroes of a Kronecker product is the product of the number of non-zeroes of the constituent matrices. Simplification is then made of the product by moving one $(2S_k + 1)(2S_m + 1)$ into the product such that it ranges over all spins.

We obtain a bound on the total number of non-zeroes by summing over all possible pairs of spins:

$$\mathrm{nnz}\Big\{\hat{H}\Big\} \leq \sum_{k=1}^{N-1} \sum_{m=k+1}^{N} (2S_k + 1)(2S_m + 1) \prod_{n=1}^{N} (2S_n + 1). \tag{1.79}$$

We define the 'density' of a matrix as the number of non-zeroes divided by the number of elements, in this case $(\prod_{n=1}^{N}(2S_n + 1))^2$. The Hamiltonian density is thus bounded by

$$\mathrm{density}\Big\{\hat{H}\Big\} \leq \frac{\sum_{k=1}^{N-1} \sum_{m=k+1}^{N} (2S_k + 1)(2S_m + 1)}{\prod_{n=1}^{N} (2S_n + 1)}. \tag{1.80}$$

This bound is tighter than that derived in Reference [3].

The bound on the density becomes small exponentially fast as the number of spins increases; this is important as it means that the actual amount of memory required for storage of the Hamiltonian does not increase as quickly as would be expected from the discussion in Section 1.2. This exponential smallness may be seen explicitly in the case of N spin-1/2 particles:

$$\mathrm{nnz}\left\{\hat{H}\right\} = 2N(N-1)2^N, \tag{1.81}$$

thus

$$\mathrm{density}\left\{\hat{H}\right\} = \frac{2N(N-1)}{2^N}. \tag{1.82}$$

Consideration of Hamiltonian sparsity is used to justify the efficient method of density matrix propagation derived in Section 3.3.

1.4.2 Liouvillian sparsity

If we use the mapping into Liouville space defined by Equation (1.40), sparsity of the Liouvillian follows from sparsity of the Hamiltonian. However, we instead make use of the Liouvillian mapping described in Section 1.3.1. This mapping does not have as clear a relation to the underlying Hamiltonian, and so independent investigation of the sparsity is required.

The statement that 'only at most two-spin couplings are present' can be reformulated as 'each column of the Liouvillian can transfer population between states differing in only at most two spin operators'. Expressed in terms of explicit product operators, this statement says that a column of the Liouvillian can only make transformations of the form

$$\hat{S}_1 \otimes \cdots \otimes \hat{S}_k \otimes \cdots \otimes \hat{S}_m \otimes \cdots \otimes \hat{S}_N \tag{1.83}$$

$$\downarrow$$

$$\hat{S}_1 \otimes \cdots \otimes \hat{S}'_k \otimes \cdots \otimes \hat{S}'_m \otimes \cdots \otimes \hat{S}_N, \tag{1.84}$$

where each distinct column will correspond to a distinct state, Equation (1.83), and operators $\hat{S}_l$ and $\hat{S}'_l$ denote (not necessarily distinct) elements from the operator basis

of spin l. Each transformation corresponds to a possible non-zero element in the column.

The number of possible operators that $\hat{S}'_k$ can be is bounded by the maximum number of elements in the operator basis of spin k, which is $(2S_k+1)^2$ (Section 1.1.1). An equivalent number holds for spin m, thus the number of states that a pair of spins k and m could be mapped into is

$$(2S_k + 1)^2 (2S_m + 1)^2. \tag{1.85}$$

We must take into account all possible pairs of spins; summing over all these pairs gives the maximum number of non-zeroes in a column,

$$\sum_{k=1}^{N-1} \sum_{m=k+1}^{N} (2S_k + 1)^2 (2S_m + 1)^2. \tag{1.86}$$

There are $\prod_{n=1}^{N}(2S_n + 1)^2$ columns, therefore

$$\text{nnz}\{\mathcal{L}\} \leq \left(\sum_{k=1}^{N-1} \sum_{m=k+1}^{N} (2S_k + 1)^2 (2S_m + 1)^2 \right) \prod_{n=1}^{N} (2S_n + 1)^2 \tag{1.87}$$

which gives the density

$$\text{density}\{\mathcal{L}\} \leq \frac{\sum_{k=1}^{N-1} \sum_{m=k+1}^{N} (2S_k + 1)^2 (2S_m + 1)^2}{\prod_{n=1}^{N} (2S_n + 1)^2}. \tag{1.88}$$

As was the case for the Hamiltonian in the previous subsection, Equation (1.88) implies that the Liouvillian gets exponentially more sparse as the number of spins increases. This may be seen simply in the case of spin-1/2 particles where, from the above equations,

$$\text{nnz}\{\mathcal{L}\} \leq 8N(N-1)4^N \tag{1.89}$$

and

$$\text{density}\{\mathcal{L}\} \leq \frac{8N(N-1)}{4^N}. \tag{1.90}$$

The sparsity of the Liouvillian has an important consequence: it suggests that the Liouvillian could be compressed. This is discussed further in Chapter 2.

1.5 Spin dynamics simulation methodology

In this thesis we present methodology suitable for simulating an experiment that can be described by a 'pulse sequence' [12] consisting of a pattern of 'perturbation periods', 'precession periods', and 'detection periods'. By 'precession periods' we mean periods where the spin system is allowed to evolve under the Hamiltonian in the presence of just the applied magnetic field. 'Perturbation periods' introduce additional terms in the Hamiltonian; within this thesis these additional terms will represent pulses of radio- or microwave-frequency radiation [8, 28], or magnetic field gradients that induce magnetic field inhomogeneity [9, 12]. The 'detection periods' in the pulse sequences in this thesis are periods which vary in length. This length variation allows measurement of the final observables as a function of the length of the detection periods. We distinguish 'direct detection periods', in which expectation values are to be computed from the density matrix at each time, from 'indirect detection periods', in which we just calculate the density matrix at each time. In the pulse sequences in this thesis direct detection periods occur exclusively at the end of the pulse sequence. A simple pulse sequence is discussed in Section 1.6, and further examples are given in Chapter 5.

Simulation of a pulse sequence in the time domain involves calculation of the density matrix or state vector by solving the appropriate Liouville–von Neumann equation. There is a separate Liouville–von Neumann equation for each period of the pulse sequence, and the final state of each period is the initial state used to solve the Liouville–von Neumann equation corresponding to the next period. The experimentally measured observables can then be generated by taking expectation values using Equation (1.29) or Equation (1.37). Conversion of this time-domain data to the (conjugate) frequency domain using Fourier transformation [56] is then usually performed in order to analyse spectra [8, 28]. Methods of generating spectra directly in the frequency domain exist (see e.g. References [57, 58, 59, 60]), but will not be discussed here.

Solution of the Liouville–von Neumann equation can proceed numerically or analytically. In practice, even 'analytical' methods need to make use of some numerical techniques as the spin system size grows.

Analytical methods have the benefit of allowing greater insight into the dynamics of a simulation than can be obtained with brute-force numerical simulation. The analytical relations rapidly become too complex to reasonably compute by hand, and use of a computer algebra package is required. The spin dynamics simulation package *SpinDynamica* makes use of the algebraic capabilities of *Mathematica* to simulate spectra in this way, though it should be noted that some operations, such as powder averaging, are carried out numerically [61]. Analytical methods are not immune to the exponential scaling problem (Section 1.2), and they also have the downside that the calculations will get slower as a pulse sequence proceeds: the terms that have to be manipulated will get more complex and numerous. The MATLAB package *EasySpin* [57, 62] utilizes analytical results to accelerate numerical simulations of ESR spectra [63]. These analytical results are, however, constructed within a strict set of assumptions, and as soon as those are violated a full numerical simulation must be carried out.

Numerical solution is the method used by most general purpose spin dynamics simulation packages, e.g. *GAMMA* [64, 65], *SIMPSON* [66, 67], *Spinach* [24, 68], and *SpinEvolution* [51, 69]. Numerical solution benefits from generality and predictable computational requirements.

All of the simulations in this thesis have been computed using numerical methods of solution. The majority of simulations in this thesis make use of the numerical methodology described in Section 1.5.1. If the Hamiltonian or Liouvillian of the system is time-dependent within a period of propagation, then the methodology of Section 1.5.1 cannot be used. The reason for this, and commonly used methods in magnetic resonance to deal with this complication are discussed in Section 1.5.2; these complications are discussed further in Chapter 4. The pulse sequence that we utilize most often in this thesis is found in Section 1.6.

1.5.1 Density matrix and state vector propagation

Within a period of a pulse sequence the Hamiltonian or Liouvillian may be a function of time. The simulations in this thesis make use of the rotating frame approximation so Hamiltonian terms arising from radio- or microwave-frequency irradiation are (approximately) time-independent (see Section 1.1.2). More complicated time-dependences are discussed in Section 1.5.2 and Chapter 4; in this section we assume that Hamiltonians and Liouvillians are time-independent.

The method of simulation used in this thesis is to solve the Liouville–von Neumann equation numerically. This generates the density matrix (state vector) as a function of time, which can be used to compute expectation values as functions of time through Equation (1.29) (Equation (1.37)). Fourier transformation [56] then maps these time dependent functions into the frequency domain for further analysis [8].

Equation (1.30), the Liouville–von Neumann equation for the density matrix, can be formally solved to give [8]

$$\hat{\rho}_{t+\Delta t} = \mathrm{e}^{-\mathrm{i}\hat{H}\Delta t}\hat{\rho}_t\mathrm{e}^{+\mathrm{i}\hat{H}\Delta t}, \tag{1.91}$$

where $\hat{\rho}_t$ is the density matrix at time t. We say that $\hat{\rho}_t$ is 'propagated under $\hat{H}$', resulting in $\hat{\rho}_{t+\Delta t}$. We refer to the matrix $\exp\{-\mathrm{i}\hat{H}\Delta t\}$ as a 'propagator'. Because Hamiltonians are Hermitian [6], the matrix exponential acting on the right-hand side of the density matrix in Equation (1.91) is the Hermitian conjugate of the propagator:

$$\mathrm{e}^{+\mathrm{i}\hat{H}\Delta t} = \left(\mathrm{e}^{-\mathrm{i}\hat{H}\Delta t}\right)^{\dagger}. \tag{1.92}$$

The form of Equation (1.91) implies that we can compute $\hat{\rho}_{t+2\Delta t}$ by

$$\hat{\rho}_{t+2\Delta t} = \mathrm{e}^{-\mathrm{i}\hat{H}\Delta t}\hat{\rho}_{t+\Delta t}\mathrm{e}^{+\mathrm{i}\hat{H}\Delta t}, \tag{1.93}$$

and thus implies that

$$\mathrm{e}^{-\mathrm{i}\hat{H}N\Delta t} = \left(\mathrm{e}^{-\mathrm{i}\hat{H}\Delta t}\right)^{N}. \tag{1.94}$$

The Hamiltonians we shall deal with often consist entirely of terms with coherence order zero with respect to all spins, i.e. they are secular with respect to all spins

and contain no radio- or microwave-frequency fields (Section 1.1.2). Such a Hamiltonian conserves the coherence order of all types of spin that are present. This follows from the fact that the coherence order of a product of operators is the sum of the coherence orders of each of the operators, as shown in Section 1.1.2. The definition of the matrix exponential, Equation (1.10), means that a propagator derived from such a Hamiltonian has coherence order zero with respect to all spins, and so the action of the propagator on a state according to Equation (1.91) must conserve coherence order with respect to all spins. If this is the case, then population cannot be transferred between operators with different coherence orders by the Hamiltonian. It should be noted that this argument also applies to the Liouvillians corresponding to such Hamiltonians.

We make use of a simplification for propagators representing radio- or microwave-frequency pulses. As discussed in Section 1.1.2, the presence of radio- or microwave-frequency irradiation gives rise to additional terms in the system Hamiltonian. For brevity in the following, we shall express the additional term arising from irradiation applied to a set of spins K with magnetogyric ratio γ as

$$- B_1\gamma\hat{F}_\phi, \tag{1.95}$$

where $\hat{F}_\phi$ is given by a sum over the spins of type K:

$$\hat{F}_\phi = \sum_k \Big(\cos(\phi)\hat{S}_x^{(k)} + \sin(\phi)\hat{S}_y^{(k)}\Big); \tag{1.96}$$

the parameter ϕ is the phase of the pulse. It should be noted that if we are exciting an electron spin, then γ should be replaced with $-g\mu_\mathrm{B}$, as mentioned in Section 1.1.2. With the above notation, we can then write the full Hamiltonian as

$$\hat{H} = \hat{H}_0 - B_1\gamma\hat{F}_\phi, \tag{1.97}$$

where $\hat{H}_0$ is the Hamiltonian in the absence of irradiation. Application of a pulse of length Δt results in the density matrix

$$\hat{\rho}' = \mathrm{e}^{-\mathrm{i}\left(\hat{H}_0\Delta t + \theta\hat{F}_\phi\right)}\hat{\rho}\mathrm{e}^{+\mathrm{i}\left(\hat{H}_0\Delta t + \theta\hat{F}_\phi\right)}, \tag{1.98}$$

where $\theta = -B_1\gamma_K\Delta t$ is the 'flip angle' [8]. We shall refer to pulses by their flip angle; common values are $\pi/2$ and π. Pulse lengths are usually very short as compared to the time scales found in $\hat{H}_0$. It is thus usual to omit $\hat{H}_0$ from the propagator to give an 'ideal pulse' propagator [8, Section 4.2.2], $\exp\{-\mathrm{i}\theta\hat{F}_\phi\}$, which is applied to the initial density matrix in the same way as the 'true' propagator to give the approximation

$$\hat{\rho}' \approx \mathrm{e}^{-\mathrm{i}\theta\hat{F}_\phi}\hat{\rho}\mathrm{e}^{+\mathrm{i}\theta\hat{F}_\phi}. \tag{1.99}$$

Ideal pulses have been assumed in all simulations in this thesis.

Solution of the Liouville–von Neumann equation in Liouville space follows a very similar pattern to the above. Given the state vector at time t, $|\hat{\rho}_t\rangle$, we can obtain the state vector at time $(t + \Delta t)$, $|\hat{\rho}_{t+\Delta t}\rangle$, by solution of Equation (1.39) [8]:

$$|\hat{\rho}_{t+\Delta t}\rangle = \mathrm{e}^{-\mathrm{i}\mathcal{L}\Delta t}|\hat{\rho}_t\rangle. \tag{1.100}$$

If a relaxation superoperator is included (Section 1.3.2), then we must solve Equation (1.58), resulting in

$$|\hat{\rho}_{t+\Delta t}\rangle = \mathrm{e}^{(-\mathrm{i}\mathcal{L}+\mathcal{R})\Delta t}|\hat{\rho}_t\rangle. \tag{1.101}$$

The matrices $\exp\{-\mathrm{i}\mathcal{L}\Delta t\}$ and $\exp\{(-\mathrm{i}\mathcal{L} + \mathcal{R})\Delta t\}$ will both also be referred to as 'propagators'. Similar arguments to those leading to the relation of Equation (1.94), lead to the two relations

$$\mathrm{e}^{-\mathrm{i}\mathcal{L}N\Delta t} = \left(\mathrm{e}^{-\mathrm{i}\mathcal{L}\Delta t}\right)^N, \tag{1.102}$$

and

$$\mathrm{e}^{(-\mathrm{i}\mathcal{L}+\mathcal{R})N\Delta t} = \left(\mathrm{e}^{(-\mathrm{i}\mathcal{L}+\mathcal{R})\Delta t}\right)^N. \tag{1.103}$$

The justification of the 'ideal pulse' approximation in the Hamiltonian case is equally valid for the Liouvillian case. Defining the superoperator $\mathcal{F}_\phi$ by

$$\mathcal{F}_\phi|\hat{\rho}\rangle = \left|\left[\hat{F}_\phi, \hat{\rho}\right]\right\rangle, \tag{1.104}$$

an ideal pulse is represented by the propagator $\exp\{-\mathrm{i}\theta\mathcal{F}_\phi\}$ such that

$$|\hat{\rho}'\rangle \approx \mathrm{e}^{-\mathrm{i}\theta\mathcal{F}_\phi}|\hat{\rho}\rangle = \left|\mathrm{e}^{-\mathrm{i}\theta\hat{F}_\phi}\hat{\rho}\mathrm{e}^{+\mathrm{i}\theta\hat{F}_\phi}\right\rangle. \tag{1.105}$$

There are many ways compute the matrix exponential [70, 71]. In this thesis we use the Taylor series expansion of the exponential, Equation (1.10). For 'short-time' propagators, propagators $\exp\{A\Delta t\}$ such that $\|A\Delta t\| < 1$, where $\|\cdot\|$ is the norm chosen to determine convergence [25], the Taylor series representation of such a propagator will converge to machine precision ($\sim 10^{-16}$ [37]) in less than 20 terms. If larger time-steps are required, then we normally use a procedure known as 'scaling and squaring' [70, 71].

Scaling and squaring utilizes the property of the matrix exponential of a matrix A that [71]

$$\mathrm{e}^{A} = \left(\mathrm{e}^{A/N}\right)^{N}. \tag{1.106}$$

This property is true for time-independent Hamiltonians and Liouvillians, as demonstrated by Equations (1.94), (1.102), and (1.103). Using this property, it is possible to choose N such that $\mathrm{e}^{A/N}$,

$$\mathrm{e}^{A/N} = \sum_{k=0}^{\infty} \frac{(A/N)^{k}}{k!} \tag{1.107}$$

is a short-time propagator. We could then compute the propagator using N matrix–matrix multiplications, as implied by Equation (1.106). However, if $N = 2^{n}$, where n is some positive integer, then

$$\mathrm{e}^{A} = \left(\mathrm{e}^{A/N}\right)^{2^{n}} = \left(\left(\mathrm{e}^{A/N}\right)^{2}\right)^{2^{n-1}} = \left(\left(\left(\mathrm{e}^{A/N}\right)^{2}\right)^{2}\right)^{2^{n-2}} = \cdots, \tag{1.108}$$

which can be built from $\mathrm{e}^{A/N}$ in just n squaring steps, i.e. only n matrix–matrix multiplications. Where possible, we always choose N to be a power of two so that we can take advantage of scaling and squaring.

The combination of scaling and squaring and Taylor series expansion has been chosen for the numerical computation of propagators because it is fast, requires minimal storage, and is (relatively) simple. In principle this method can be subject to pathological behaviour ('humps') [70, 71], but the matrices involved in spin dynamics simulations seem to be well-behaved enough that we have seen no such issues.

For Liouville space simulations, computation of propagators is often the most computationally taxing part of simulating a pulse sequence. This is because whereas computing the action a propagator on a state vector requires only involve matrix–vector multiplications, generation of the propagator involves computing a matrix exponential using matrix–matrix multiplications. Methods that avoid explicit computation of the propagator are thus used for large spin systems; these are discussed in Section 2.2.3.

1.5.2 Time dependent Liouvillians

In this section we discuss complications that arise when Liouvillians are time dependent, and methods commonly used in magnetic resonance to mitigate them. Further methods may be found in Chapter 4. While the discussion in this section refers explicitly to 'Liouvillians', only minor modifications of the arguments presented are required to apply them to Hamiltonians.

If the Liouvillian is time-dependent, then we express the Liouville–von Neumann equation (Equation (1.39)) as

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}_t\rangle = -\mathrm{i}\mathcal{L}_t|\hat{\rho}_t\rangle, \tag{1.109}$$

where $\mathcal{L}_t$ is the Liouvillian at time t. We can express the solution of this equation in the form

$$|\hat{\rho}_{t'}\rangle = \mathcal{P}(t,t')|\hat{\rho}_t\rangle, \tag{1.110}$$

where $\mathcal{P}(t,t')$ is the propagator taking $|\hat{\rho}_t\rangle$ into $|\hat{\rho}_{t'}\rangle$, and $t' > t$. We cannot in general use the methods of Section 1.5.1 to construct $\mathcal{P}(t,t')$ because for $t \leq r < s \leq t'$ it may be the case that

$$\mathcal{L}_r\mathcal{L}_s - \mathcal{L}_s\mathcal{L}_r \neq 0, \tag{1.111}$$

meaning that the ordering of matrix–matrix multiplication operations in the solution is important. In order to maintain proper ordering of operations, $\mathcal{P}(t,t')$ is formally written in terms of a time-ordered exponential [72]. The time-ordering requirement results in calculation of this series being more complicated than calculation of the usual matrix exponential, defined by Equation (1.10).

If the time-dependence of the Liouvillian is discrete, i.e. the Liouvillian is piecewise constant in time, then evolution from time t to time $t+N\Delta t$ is generated by the much simpler propagator

$$\mathcal{P}(t, t+N\Delta t) = \mathrm{e}^{-\mathrm{i}\mathcal{L}_{t+N\Delta t}\Delta t} \cdots \mathrm{e}^{-\mathrm{i}\mathcal{L}_{t+2\Delta t}\Delta t} \mathrm{e}^{-\mathrm{i}\mathcal{L}_{t+\Delta t}\Delta t}, \tag{1.112}$$

where we have assumed for simplicity that the Liouvillian is piecewise constant over time-steps of size Δt. We have chosen the value of t taken by the Liouvillian during the kth time-step (from $[t+(k-1)\Delta t]$ to $[t+k\Delta t]$) to be $t+k\Delta t$ for definiteness, although it can be chosen as any time within the kth time-step. Liouvillians with continuous time-dependence can be approximated by a Liouvillian that is piecewise constant over small time steps [8], allowing us to use Equation (1.112) to generate an approximate propagator without having to consider time-ordering. This approximation is exact only in the limit as the time-step tends to zero. Unfortunately, the number of time-steps required in order to obtain convergence of the propagator below a desired tolerance may be very large, implying that a large number of propagators would have to be built. This can be very time consuming, and so the 'average Liouvillian' methods described below may be preferred.

It is useful to note that if $\mathcal{L}_t$ contains only terms of coherence order zero (with respect to a subset of spins) at all t, the expression of Equation (1.112) will conserve coherence order (of that subset of spins) via the arguments presented in Section 1.5.1. This conservation will hold as the time-step tends to zero, thus time-dependent Liouvillians containing only terms of coherence order zero (with respect to a subset of spins) will conserve coherence order (of that subset of spins). Secular Liouvillians without applied radio- or microwave-frequency fields are an important case where this conservation occurs.

Like the matrix exponential (Equation (1.10)), the time-ordered exponential is a series with an infinite number of terms. For numerical evaluation, this series must be truncated at some finite number of terms. It is often desired that such truncated series preserve properties that the true solution should have, such as unitarity [8],

the property that $[\mathcal{P}(t,t')]^{-1} = [\mathcal{P}(t,t')]^{\dagger}$ [73]. Scaling and squaring is not possible for calculation of the time-ordered series, and so preservation of such properties to a reasonable degree of accuracy is likely to require the calculation of many terms. It is possible, however, to derive various rearrangements of this series that do preserve these properties.

A useful method for deriving such a rearranged series is to think in terms of an 'average Liouvillian'. An average Liouvillian $\overline{\mathcal{L}}_{t,t'}$ is a matrix defined by the relation

$$\mathcal{P}(t,t') = \mathrm{e}^{-\mathrm{i}\overline{\mathcal{L}}_{t,t'}(t'-t)}, \tag{1.113}$$

where $t' > t$. While a given average Liouvillian will itself generally be represented by series with an infinite number of terms, methods have been devised (see below) for creating them such that truncations of the series preserve certain important properties. Average Liouvillians can also allow larger time steps to be taken than in the piecewise constant case above, reducing the number of propagators that need to be built. Once the average Liouvillian has been constructed, we can then use scaling and squaring to generate the propagator.

Average Liouvillians are normally constructed in magnetic resonance using a Magnus expansion [8, 48, 74]. Other methods are known in the literature, e.g. Krylov–Bogolioubov–Mitropolskii averaging [75, 76, 77], but will not be discussed here. The Magnus expansion is an infinite series which must be truncated for numerical use. If $\mathcal{L}_t$ is Hermitian ($\mathcal{L}_t = (\mathcal{L}_t)^{\dagger}$) for all t, the Magnus expansion has the benefit of maintaining this Hermiticity at all levels of truncation of the series [74]. A matrix $\exp\{-\mathrm{i}A\}$ is unitary if the matrix A is Hermitian; this can be shown by examining the Hermitian conjugate of the Taylor expansion of this matrix:

$$\left(\mathrm{e}^{-\mathrm{i}A}\right)^{\dagger} = \sum_{k=0}^{\infty}\left(\frac{(-\mathrm{i}A)^k}{k!}\right)^{\dagger} \tag{1.114}$$

$$= \sum_{k=0}^{\infty}\frac{(+\mathrm{i}A)^k}{k!}, \tag{1.115}$$

where the last equation is the definition of $\exp\{+\mathrm{i}A\}$, i.e. the inverse of $\exp\{-\mathrm{i}A\}$. As

they conserve Hermiticity, truncated Magnus expansions thus also maintain unitarity of the propagator.

1.6 Pulse–acquire pulse sequence

A very common pulse sequence is the 'pulse–acquire' experiment. We examine this simple pulse sequence here because this thesis includes many simulations of the spectra that arise from it.

The pulse–acquire experiment consists of a short radio- or microwave-frequency pulse with flip angle $\pi/2$ applied to the spins of interest, followed by a detection period [12]. The spin system is assumed to be at equilibrium at the beginning of the experiment; the pulse transfers population to the transverse states of the spins of interest. A direct detection period then allows detection of the population of the transverse states as a function of time; this corresponds to computing the expectation value of either $\sum_k \hat{s}_+^{(k)}$ or $\sum_k \hat{s}_-^{(k)}$, where the summation runs over all detected spins [12], as a function of time. Discrete Fourier transformation of this signal then gives the 'magnetic resonance spectrum' [8, 12, 28]; an example may be found in Figure 2.1.

There is a restriction on the largest time-steps that may be taken in the time-domain during the detection period, important for the discussion of Section 3.3.1. This restriction arises from a desire to avoid spectral artefacts due to 'peak folding'. Peak folding occurs when the time-step Δt is larger than $(2f)^{-1}$, where f is the largest frequency appearing in the time-domain signal; this criterion is given by the Nyquist–Shannon sampling theorem [8]. The peaks of an NMR spectrum arise at the eigenvalues of the Hamiltonian [12]. As the 2-norm of a matrix, $\|\cdot\|_2$, bounds the eigenvalues of a matrix [78], by taking the time-step to be less than $(2\|\hat{H}\|_2)^{-1}$ we can obtain a bound on the largest frequency that will appear in the time-domain signal, and be able to choose Δt to fulfil the sampling theorem. In practice, the 2-norm is expensive to compute. Upper bounds on its value can be obtained using other norms with lower computational cost, however [25]; these upper bounds are those usually

used to determine Δt.

More complicated pulse sequences are described in Chapter 5.

1.7 *Spinach*

All of the work presented in this thesis utilizes the *Spinach* toolbox. This toolbox, originating from the Kuprov group, provides tools for simulating general magnetic resonance spectra [24], and is implemented in the computer language MATLAB. Code arising from the concepts presented in this and subsequent chapters has been incorporated into the *Spinach* toolbox; the *Spinach* source code, available from Reference [68], at time of writing thus includes many of the files used to generate figures in this thesis.

Chapter 2

State space restriction

As discussed in Section 1.2, the sizes of matrices and vectors required for general spin dynamics simulations scale exponentially with the number of spins in the spin system. This makes general spin dynamics simulations infeasible for large spin systems.

However, it has been demonstrated that many Liouville space magnetic resonance simulations can be performed using matrices and vectors of vastly reduced size without any significant loss of accuracy [79, 80, 81, 82]. The sizes of the matrices and vectors in these simulations scale only *polynomially* with the number of spins [2]. This chapter provides descriptions of several known methods for obtaining the smaller vectors and matrices used in these simulations.

The matrix and vector sizes required for Liouville space simulation of spin dynamics are closely related to the number of operators in the product operator basis. As each product operator in the basis corresponds to an element of our state vector (Section 1.3.1), removing a product operator from the basis allows reduction of the number of elements in the state vector by one. The Liouvillian can be similarly reduced in size by removing the column and row corresponding to this element. In the following, we refer to elements of the product operator basis as 'states', and we refer to the basis as a 'state space'. We collectively refer to methods for determining which states can be removed from the basis as 'state space restriction' (SSR) methods [79].

That the full state space will not normally be required may be argued by noting the following significant features of magnetic resonance experiments:

1. only at most two-spin couplings occur in spin Hamiltonians (Section 1.1.2).

2. experiments can only be run for a finite time.

3. dynamic experiments can only be measured at discrete time points.

4. usually only a small subspace is initially populated to a significant extent.

5. relaxation is usually significant over the timescales to be simulated [83].

The first point means that Liouvillians are sparse, as discussed in Section 1.4.2. This sparsity implies that there is much room for compression of the Liouvillian. It further implies that path tracing (Section 2.2.1) and Krylov methods (Section 2.2.3) can be of use.

The second and third points together mean that the minimal number of linearly independent vectors needed for a given simulation is less than or equal to the number of time points in the simulation (Section 2.2.3). Unfortunately, in the general case, such a minimal basis can only be found exactly after the full simulation has been carried out. Zero track elimination (Section 2.2.2) attempts to approximate this minimal basis after evolving the spin system for a very short time.

The fourth point above states that the majority of magnetic resonance experiments begin with significant population of only a small subspace. This subspace consists of very low spin-order states; the validity of the high temperature approximation (Section 1.1.3) means that these are often single-spin order states, but experiments beginning with significant population of two-spin order states are known, e.g. CIDNP [84, 85], or PHIP [86]. The population to any significant extent of states of larger spin-order in the initial state is rare. This is beneficial for us: if a large number of states were initially populated to an appreciable extent, then this large number of states would be needed throughout the simulation.

The fifth point, the presence of significant relaxation in the system, can lead to states of large spin-order never becoming populated to the extent that they would

contribute to the spectrum. Such large spin-order states could thus be removed from the basis. This forms one rationale for 'basis set truncation', discussed in Section 2.1.

Basis set truncation is a 'basis level method': it is applied before the basis is built. 'Trajectory level methods', in contrast, are applied after the basis has been built, but before any considerable evolution has been made under the Liouvillian. The trajectory level methods used in the simulations in this thesis are discussed in Section 2.2. A method that can lead to even greater reductions in state space size when using trajectory level methods is presented in Section 2.3.

This chapter deals exclusively with Liouville space simulations. In simple cases state space restriction can be applied to density matrix simulations (an example may be found in Section 3.3.3). In general, however, a reduced Liouvillian will not correspond to a reduced Hamiltonian [79]. Further, relaxation, which cannot in general be included in a Hamiltonian (Section 1.3.2), provides the justification for several of the restriction methods (see Sections 2.1 and 2.2.2). We thus do not discuss Hamiltonian state space restriction here.

2.1 Basis set truncation

Reducing the number of elements in the product operator basis (Section 1.1.1) is a very important way of making the formally-exponentially scaling memory requirements of spin dynamics simulations (Section 1.2) more manageable. Truncating the basis such that it excludes all product operators of spin order (Section 1.1.1) greater than a certain value can be shown to result in a state vector with a number of elements that scales only *polynomially* with the number of spins in the spin system [79], a significant improvement over exponential scaling. Empirical investigation has found this sort of truncation to be valid in several situations in magnetic resonance [79, 80]; an example is shown in Figure 2.1.

Figure 2.1 shows the ^{1}H spectrum of strychnine, a spin system consisting of 22 ^{1}H spins, simulated using a basis truncated such that no states of spin order greater than

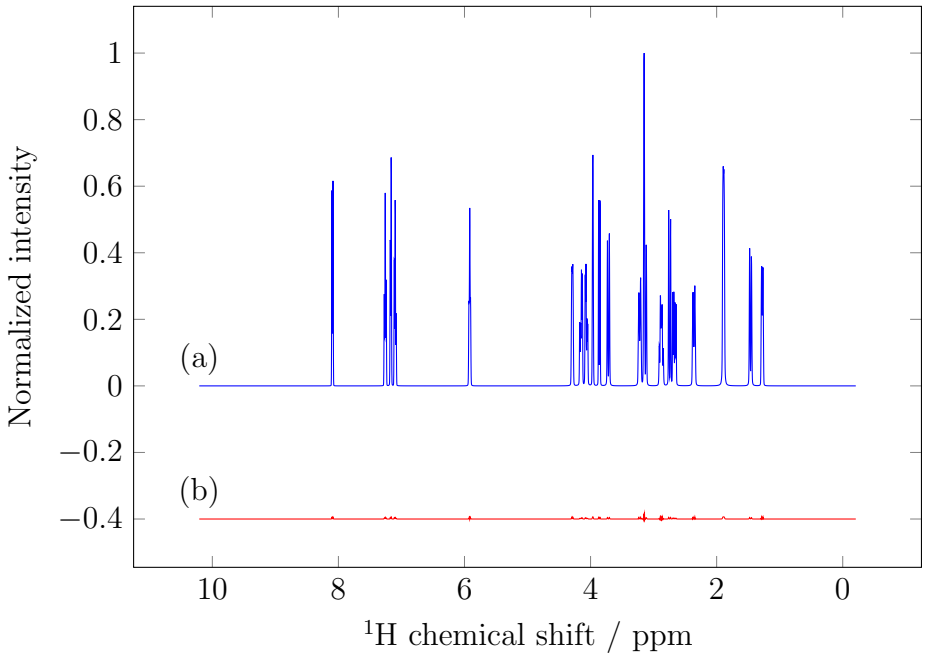


Figure 2.1: (a) NMR spectrum of the 22 ^{1}H-spin system of strychnine generated using the IK–0(5) basis set. (b) Difference between spectrum of (a) and the spectrum generated using the IK–0(4) basis set. Spectrum (b) is offset from (a) by -0.4 for clarity. Strychnine spin system parameters are taken from Reference [87], a secular Redfield relaxation superoperator [47] has been included with a correlation time [34] of 200 ps using the coordinates of the 'major' conformer from Reference [88], and the magnetic field strength is 11.74 T. The time-step was chosen to be $10/\|\mathcal{L}\|_1$, where $\|\cdot\|_1$ is the 1-norm [25], and $(2^{11}-1)$ steps were taken. Relaxation is to the thermal equilibrium state at 298 K. Trajectory level SSR was not used in this simulation.

five are included. Within *Spinach*, truncated basis sets are used by specifying the basis to be IK–0(n), where n is the maximal spin-order desired [24], i.e. this simulation was computed using the IK–0(5) basis set. A secular Bloch–Wangsness–Redfield relaxation superoperator [47] has been included (Section 1.3.2). This superoperator has been modified in the manner described in Section 1.3.3 so that it relaxes the state of the system to thermal equilibrium.

In order to see that this truncation is empirically valid, we can examine Figure 2.2, a plot of the contributions that each set of spin-order states makes to the total square norm of the state vector. Only the one spin-orders are populated to any significant extent at the beginning of the simulation, in line with the high temperature approximation (Section 1.1.3). The norm of the states of spin-order five can be seen to

be negligible for the entirety of the simulation trajectory. Omitting these and larger spin-order states from the basis results in quartic scaling of the basis set size with the number of spins [2]; while this basis is still very large for the 22-spin system of strychnine, the simulation is possible on a modern supercomputer. Calculation with the full basis would not be possible at all. A more detailed analysis of this system (with slightly different spin system parameters) may be found in Reference [2].

A rationalization of the validity of basis set truncation in spin dynamics simulations that include significant relaxation processes due to Alex Karabanov can be found in Reference [2]. The central concept is that relaxation can damp states 'far away' from the initial set of states before they are populated to any significant extent; in our case, the states 'far-away' from the initial set of states are the states of large spin-order. Relaxation explains why spin-orders greater than five never get populated to any significant extent in Figure 2.2 [2]. References [2] and [41] should be consulted for details.

Intriguingly, simulations performed by Butler, et al. demonstrating the empirical validity of basis set truncation include no relaxation processes [80]. These simulations did, however, include powder averaging and magic angle spinning. An exploration of how these inclusions could explain the validity of basis set truncation in such simulations is presented in Section 4.4.5.

Uniformly restricting the basis by spin-order is not, in general, an optimal restriction. Relatively low spin-order states could remain unpopulated if they involve spins at opposite ends of a large system, or a system may have localized clusters of strongly coupled spins requiring large spin-orders only locally. A more optimal restriction method would take into account such interaction connectivity [24].

The bases IK–1(n, k) and IK–2(k), used frequently in simulations in this thesis, are examples of connectivity-adaptive restricted basis sets. Both bases contain all product states of spins which are proximate in space (below a user-defined tolerance) up to spin-order k. They differ in how many product states between coupled spins are included: the IK–2(k) basis includes all product states between coupled spins,

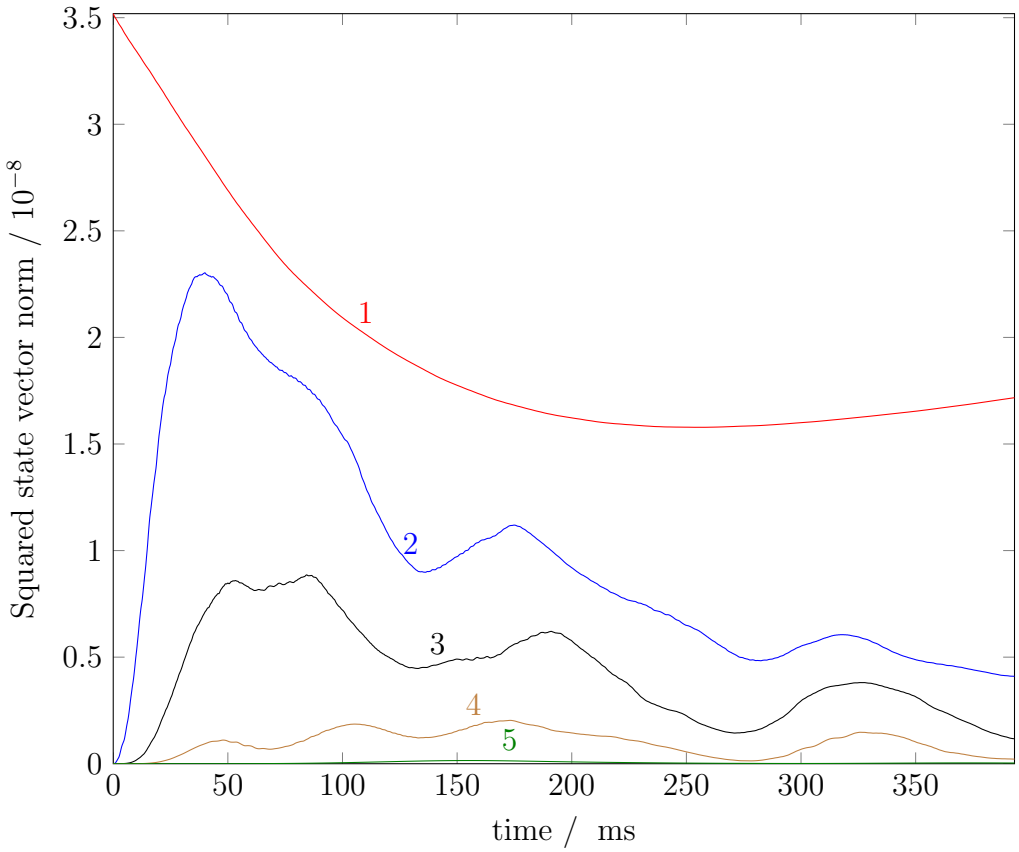


Figure 2.2: Figure demonstrating applicability of basis level state space restriction for simulation of the NMR pulse acquire spectrum shown in Figure 2.1(a). Plotted is the squared norm of the state vector used to generate Figure 2.1(a) after subtraction of the unit state (zero spin-order) part. State vector is normalized such that the unit state population before this subtraction is unity. This squared norm is further demarked by labelled lines into contributions to this total from the various spin orders. Contributions from five-spin order states are negligible for the entirety of the trajectory. The populations are tending towards equilibrium at the end of the trajectory. As the perfect pulse assumed in the pulse acquire experiment cannot cause mixing between states of different spin-order in this spin system, the equilibrium values for the various norms are those seen at time zero. This simulation thus implies that spin order states greater than five will not be populated to any appreciable extent, and so states with spin-order greater than or equal to five can be omitted from the basis with little decrease in accuracy of the simulation.

whereas IK–1(n, k) only contains product states up to spin-order n between coupled spins. For details see Reference [24].

2.2 Trajectory level methods

The Liouville space calculations in this thesis make use of trajectory level SSR methods unless otherwise stated. These methods are thus described briefly here. The descriptions provided here are only intended to represent a general overview of these methods; algorithmic and implementation details can be found in the original literature, cited at appropriate points in the following.

2.2.1 Path tracing

A Liouvillian that is represented by a $K \times K$ matrix can be considered as the 'adjacency matrix' of a graph with K vertices [40, 52, 89]. The number of 'edges' of the graph corresponds to the number of non-zeroes of the Liouvillian. Well-known 'path-tracing' algorithms [40, 90] can then be used to investigate whether disconnected subspaces exist in the Liouvillian. Such subspaces can arise due to conservation laws [24, 73, 91]. The sparsity of the Liouvillian (Section 1.4) makes the application of such algorithms efficient. Path tracing was introduced into *Spinach* through the efforts of Hannah Hogben and Ilya Kuprov; proofs of the relations given below may be found in References [24] and [40].

If path tracing reveals N independent subspaces, then the state vector may be split into N sub-vectors for propagation under the Liouvillian [24]. Propagation of the sub-vector corresponding to the nth subspace, $|\hat{\rho}\rangle_n$, is performed by solving the Liouville–von Neumann equation

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}\rangle_n = -\mathrm{i}\mathcal{L}_n|\hat{\rho}\rangle_n, \tag{2.1}$$

where $\mathcal{L}_n$ is the portion of the Liouvillian mapping between elements of this subspace. The independence of the subspaces allows for parallelization; see Section 3.2. At the

end of the propagation period, the full state vector can be rebuilt by recombination of all of the individual subspaces.

A detection state $\langle\hat{s}^{\dagger}|$ can be mapped into the smaller subspaces in the same manner as the state vector to give N sub-vectors; we express the nth sub-vector as ${}_n\langle\hat{s}^{\dagger}|$. We can then compute expectation values without rebuilding the full state vector using [24]

$$\langle\hat{s}\rangle = \sum_{n=1}^{N} {}_n\langle\hat{s}^{\dagger}|\hat{\rho}\rangle_n. \tag{2.2}$$

Both the storage and time required for Tarjan's graph partitioning algorithm scale linearly with the number of edges and the number of vertices of the graph to which it is applied [90]. For a $K \times K$ Liouvillian, this means that asymptotically this algorithm scales linearly with both K and the number of non-zeroes in the Liouvillian. Sparse matrix–sparse matrix multiplication of the Liouvillian with itself, as is required to build a propagator from the Liouvillian, also asymptotically scales linearly with the number of non-zeroes of the Liouvillian (for a matrix of fixed size) [92, 93]. Assuming that the subspaces are all of equal size and that the non-zeroes are distributed equally, separation into N subspaces will reduce this scaling linearly. In general, then, use of path-tracing algorithms is not guaranteed to present any benefit; it provides linear computational savings for linear computational cost.

However, it may be the case that some of the sub-vectors are composed entirely of zeroes (or numbers very close to zero); we can drop these vectors from consideration, giving rise to a formal state space restriction. There are often many such unpopulated subspaces at the beginning of a simulation. This is due to the sparsity of the state vectors at the start of a pulse sequence, discussed at the beginning of this chapter. Not having to calculate the propagator for these subspaces could result in super-linear benefits, repaying the extra computation required in order to obtain the subspace decomposition. In Section 2.3 we show that an even greater reduction in the number of subspaces that need to be propagated can often be achieved by including analysis of the detection state.

2.2.2 Zero track elimination

'Zero track elimination' (ZTE) is a method of trajectory level SSR due to Ilya Kuprov [94]. Implementation details, etc. may be found in Reference [94]; here we just provide a brief outline of the concept.

If the population of a state remains identically zero when it is propagated for a very short time, it can be shown, via Taylor expansion of the propagator, to remain identically zero for the entirety of the simulation [94, Theorem 1]. The number of such states is likely to be small, however, so the assertion made in ZTE is that a state which remains below a population threshold $\epsilon_{\text{zte}} > 0$ over a time period t_{zte} will remain below a threshold $\delta > 0$ for the rest of the simulation (where choosing t_{zte} long enough and ϵ_{zte} small enough will be enough to make $\delta \ll 1$).

The ZTE assertion is true if the norm of $\mathcal{L}t$ is less than unity for the duration of the simulation, a simple corollary of a theorem of Kuprov [94, Theorem 2]. It is also trivially true if t_{zte} is taken to be equal to the length of the simulation. Unfortunately, neither of these cases are very useful for spin dynamics simulations.

Mazzi and Leimkuhler have shown that states could be pruned by ZTE that would have eventually reached non-negligible magnitude [95]. This possibility apparently contradicts empirical data demonstrating the validity of ZTE [94]. Mazzi and Leimkuhler posited an explanation for this: relaxation had been excluded from their analysis [95]. The simulations in Reference [94] included relaxation in an effective manner through apodization [8] of the time-domain signal. Relaxation could possibly reduce the population of these states before they reach any appreciable magnitude.

The dynamics of general spin systems are complicated; this is why we resort to numerical calculations. Unfortunately, this means that despite best efforts, analytical criteria for quantifying the error associated with a given choice of t_{zte} and ϵ_{zte} over the course of a simulation are still wanting. The implication of the work of Mazzi and Leimkuhler is, however, that ZTE should be used with care, especially in the absence of relaxation.

2.2.3 'Krylov' methods

In most cases in magnetic resonance one is not interested in a given propagator. Rather, one desires only the state vector generated by the action of this propagator on an initial state.

A propagator represented by a Taylor series truncated at the mth term acting on a state vector $|\hat{\rho}_0\rangle$ generates dynamics that remain within a well defined subspace, known as a 'Krylov subspace' [52, 96]. This Krylov subspace, K_m, is defined by

$$K_m(-\mathrm{i}\mathcal{L}\Delta t, |\hat{\rho}_0\rangle) = \\ \mathrm{span}\big\{|\hat{\rho}_0\rangle, -\mathrm{i}\mathcal{L}\Delta t|\hat{\rho}_0\rangle, (-\mathrm{i}\mathcal{L}\Delta t)^2|\hat{\rho}_0\rangle, \ldots, (-\mathrm{i}\mathcal{L}\Delta t)^{m-1}|\hat{\rho}_0\rangle\big\}. \tag{2.3}$$

The 'span' of a set of vectors is the subspace generated by taking all possible linear combinations of the vectors in the set [19]. The dimension of this space, the number of vectors required to span the subspace, is $\leq m$. If m is less than the number of elements in the state vector, then mapping into this smaller subspace would result in a reduction in state space size.

It is possible to explicitly exploit such 'natural' reduction via Krylov methods which either implicitly or explicitly provide a basis set for this Krylov subspace [96]. These methods are particularly applicable when the matrices involved are sparse [52]. Use of such methods is well-known in frequency domain solutions of the Liouville–von Neumann equation [58, 97]. Time-domain evolution using these Krylov methods may be performed in MATLAB by using the *Expokit* package [95, 98, 99].

Unfortunately, Krylov methods were found to be too slow for the sort of large-scale calculations we wish to perform. However, we can still make use of the concept of a Krylov subspace in the following way. As we are only interested in the final state, and not the matrix exponential itself, we can generate the action of the matrix exponential on the initial vector directly, and not the explicit matrix exponential. Only matrix–vector multiplications are required, as may be shown by examination of the Taylor series definition of the matrix exponential (Equation (1.10)) of a matrix

X acting on a vector v:

$$\mathrm{e}^{X}v = \sum_{k=0}^{\infty} \frac{1}{k!} X^{k} v = \sum_{k=0}^{\infty} c_{k}, \tag{2.4}$$

where we define the shorthand $c_k = (1/k!)X^k v$. Examining the kth term c_k, we see that it may be written

$$c_{k} = \frac{1}{k} X c_{k-1}, \tag{2.5}$$

i.e. as a matrix–vector multiplication.

The 'Krylov' method implied by Equations (2.4) and (2.5) can be applied to massively reduce the amount of computer memory required for propagation [24]. A similar relation holds when the matrix exponential is represented as a Chebyshev series [100]. The 'Krylov' method does, however, preclude scaling and squaring (Section 1.5.1): because the propagator is not constructed explicitly, it cannot be squared. This means that using the 'Krylov' method can result in much longer simulation times. This 'Krylov' method has been utilized to simulate the dynamics of the spin systems in Sections 2.1 and 5.3.6.

The concept of a Krylov space can also be used to prove an important result relating to the possibility of SSR, described in Reference [24]. When the Liouvillian $\mathcal{L}$ is time-independent, the repeated action of the propagator on the initial state vector $|\hat{\rho}_0\rangle$ is itself contained within a Krylov subspace. Denoting the propagator $\exp\{-\mathrm{i}\mathcal{L}\Delta t\}$ by $\mathcal{P}$, the Krylov subspace generated by taking $m-1$ steps with this propagator is given by

$$K_{m}(\mathcal{P}, |\hat{\rho}_0\rangle) = \mathrm{span}\big\{|\hat{\rho}_0\rangle, \mathcal{P}|\hat{\rho}_0\rangle, \mathcal{P}^{2}|\hat{\rho}_0\rangle, \ldots, \mathcal{P}^{m-1}|\hat{\rho}_0\rangle\big\}. \tag{2.6}$$

A corollary of this is that the number of linearly independent vectors required to represent the trajectory is always less than or equal to the number of time points, i.e. while the number of time points is less than the dimension of $\mathcal{L}$ a dimensional reduction is always possible [24].

2.3 Destination state screening

Destination state screening (DSS) is the concept that one can apply trajectory level SSR tools to the *detection state* rather than (or in addition to) applying them to the initial (source) state for the direct detection period at the end of a pulse sequence. This concept, due to Matthew Krzystyniak, can lead to greater reductions in state space size than application of these trajectory level tools to just the initial state, as we demonstrate below and in Section 5.1. DSS and several examples of its utility have been published [1].

That the application of SSR tools to the detection state is possible can be formally deduced in the following way. As discussed in Section 1.5, time-domain Liouville space simulations of the directly detected period of a pulse sequence are normally carried out by propagating the state vector at the beginning of the period, $|\hat{\rho}_0\rangle$, to some time t, followed by multiplication by a detection state $\langle\hat{s}|$ to compute the expectation value of an observable operator $\hat{s}^\dagger$. Explicitly,

$$\left\langle \hat{s}^\dagger \right\rangle_t = \langle\hat{s}|\big(\mathrm{e}^{-\mathrm{i}\mathcal{L}t}|\hat{\rho}_0\rangle\big). \tag{2.7}$$

However, equivalently, we could calculate

$$\langle\hat{s}_t| = \langle\hat{s}|\exp\{-\mathrm{i}\mathcal{L}t\}, \tag{2.8}$$

and then compute the expectation value by multiplication by $|\hat{\rho}_0\rangle$, i.e.

$$\left\langle \hat{s}^\dagger \right\rangle_t = \langle\hat{s}_t|\,\hat{\rho}_0\rangle. \tag{2.9}$$

This is the Heisenberg formulation of quantum mechanics [73]. This calculation is strictly equivalent to performing the simulation 'forwards', even in the presence of relaxation, meaning that the elements of $|\hat{\rho}_0\rangle$ that cannot be reached from $\langle\hat{s}|$ never contribute to $\langle\hat{s}^\dagger\rangle_t$. The corresponding elements of the state vector can therefore be removed, whichever way around the simulation is performed.

The detection state in magnetic resonance experiments often consists of a sum of single spin-order, single coherence-order states (the transverse states of one sort

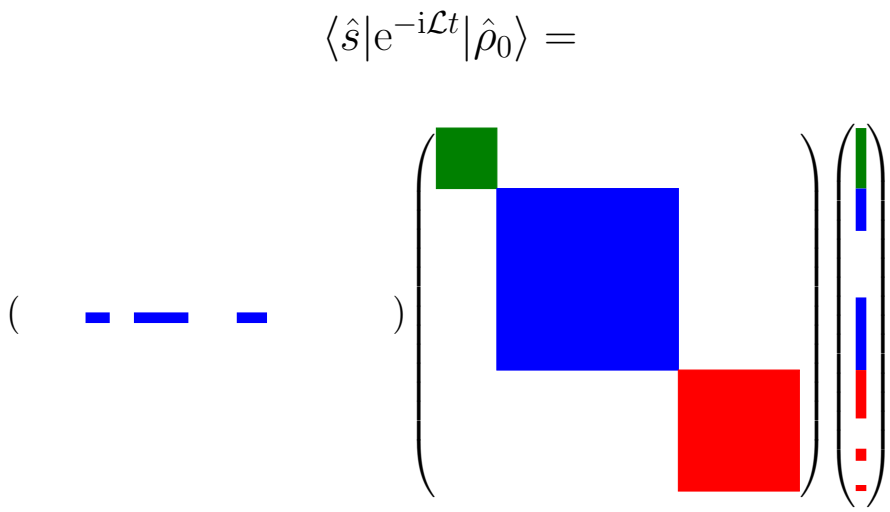


Figure 2.3: Schematic representation of the direct detection period of a spin dynamics simulation demonstrating how DSS can lead to greater reductions than source state screening. The coloured blocks in the vectors and matrices represent non-zero matrix elements. As is often the case in magnetic resonance (see Section 2.2.1), the propagator $\exp\{-\mathrm{i}\mathcal{L}t\}$ is separable into non-interacting subspaces each of which can be propagated separately. After a complicated pulse sequence, the source state $|\hat{\rho}_0\rangle$ is likely to have elements in many of these subspaces; source state screening will predict that all of those subspaces need to be propagated. However, inspection of the elements of the detection state $\langle \hat{s}|$ shows that only the central blue subspace needs to be propagated.

of nucleus) [12]; such a combination of states is sparse. The source state, $|\hat{\rho}_0\rangle$, will often have been through a complicated multi-stage pulse sequence before reaching the direct detection period; even if it were initially sparse, it is unlikely that it will have remained so. The dimension of the subspace containing $\langle \hat{s}|$ is therefore likely to be smaller than that of $|\hat{\rho}_0\rangle$. This is shown schematically in Figure 2.3. Application of existing trajectory level state space restriction tools to $\langle \hat{s}|$ instead of $|\hat{\rho}_0\rangle$ would then lead to a greater degree of reduction, speeding up the simulation.

Shown in Table 2.1 are examples of reductions found in several 2D NMR pulse sequences. Further examples may be found in Reference [1]. Multi-dimensional pulse sequences benefit greatly from the use of DSS as the pulse sequences are usually long. The density matrix is thus very dense at the beginning of the detection period and little reduction is possible using 'source-state' screening, as demonstrated in these examples. Use of DSS results in significantly smaller subspaces.

Spin system	Pulse sequence	State vector length		
		IK–2(1) state space	Source state screening	Destination state screening
Strychnine	^{1}H NOESY	12,316	9,551	1,549
Sucrose	^{1}H DQF-COSY	1,165	1,133	258

Table 2.1: Comparison of reductions using source- and destination-state screening during the direct detection period of simulations of 2D NMR pulse sequences. See e.g. Reference [10] for details of the pulse sequences, and the *Spinach* source code [68] for implementation details. The IK–2(1) truncated basis set (Section 2.1) has been used, and the trajectory level methods applied to the source- and destination-states are ZTE and path tracing. The destination state is the detection state $\sum_k |\hat{s}_+^{(k)}\rangle$, where summation is over all ^{1}H spins, in both cases. A number of state vectors are propagated through the direct detection period, sampling the indirect detection period of these pulse sequences. Source state screening is applied to the mean state vector. The data presented differ from those presented in Reference [1] in two ways. Firstly, the implementation of trajectory-level state space restriction has been improved due to the efforts of Ilya Kuprov. Secondly, only +1 coherence is retained during the indirect detection periods [12]. Sucrose spin–spin couplings taken from a vacuum DFT simulation performed by Ilya Kuprov using Gaussian 03 [101]; isotropic chemical shifts taken from Reference [102]. Strychnine spin system parameters taken from Reference [87]. Magnetic field of 5.9 T used in both simulations.

The application of DSS is not restricted to just the direct detection period at the end of a pulse sequence. An example is given in Section 5.1 of the application of DSS to an earlier precession period of a pulse sequence.

2.4 Conclusions

This chapter has described a selection of the state space restriction methods currently in use in the field of spin dynamics. The advantages of utilizing these methods cannot be overstated; they make simulations possible that would have been impossible just a few short years ago. The following chapters give numerous examples of the utility of these state space restriction techniques.

A particularly impressive example of the use of state space restriction is presented in Section 5.3.6: a simulation of the HN(CO)C_α spectrum [103] of the protein ubiquitin [11]. This protein contains over a thousand spins; the simulation would not be possible without the use of state space restriction.

Intriguingly, state space restricted Liouvillians can still retain a large amount of sparsity. This leads to the possibility that even further reductions are possible. 'Tensor trains' provide a promising avenue of investigation [4, 104]; by expressing the state vector as an explicit tensor product one can often obtain a reduction in effective state space size. This effective reduction in matrix size may lead to an increase in the speed of simulations, though obtaining such a speed increase is unfortunately not trivial [104].

Chapter 3

Parallelization

The methods of simulation presented in Chapters 1 and 2 have implicitly assumed that calculations will be performed 'serially', as if by hand. Modern computers have multiple processing units, which we refer to generically as 'processors'. Each of these processors is capable of performing computations independently of the others in 'parallel' [105].

In this chapter we discuss how we can make use of parallel computing to reduce the time taken to perform spin dynamics simulations. For the sake of brevity, we shall often refer to the 'time taken to perform' a simulation as the 'run-time' of a simulation. The rest of this section introduces terminology and concepts utilized in the rest of the chapter.

In order to take advantage of parallelism to accelerate a simulation, we must separate the simulation into independent 'tasks' which can each be computed on a separate processor [25, 106]. We shall refer to modification of an algorithm to split it up into separate tasks to be run on separate processors as 'parallelization' [105].

If a simulation takes a time T to compute serially using a single processor, then splitting the calculation between N processors is ideally expected to result in a run-time of T/N. Such an analysis provides a useful benchmark, but ignores several very important factors, the most important of which we shall now describe.

Computer memory can play an important rôle in the efficiency of a parallelization; this is because computer memory is a finite resource. Further, not all memory types

are equal: there is a hierarchy of memory types [105, 107]. We give here a model of this hierarchy sufficient to explain the methods of parallelization in this chapter. More sophisticated models can be found in References [105, 106, 107].

As a first approximation to the hierarchy, we can consider a parallel computer which has a large amount of slow 'main memory' that every processor can access, and a small amount of fast memory ('cache') associated with each processor which no other processor can access [105, 107]. While processors can access their cache independently of other processors, access to main memory provides a bottleneck, as normally only a single processor can read or write to it at once. This breaks the independence of the tasks, resulting in non-ideal scaling of the run-time with the number of processors. In order to attain better parallel performance, it is thus beneficial to place all data required by a processor in its cache. Two idioms for making efficient use of the cache are 'replication' and 'distribution', which we shall now describe.

If a parallelization requires that each processor work on a separate non-intersecting portion of a matrix or vector, e.g. in the case of a matrix, processor one needs the first column, processor two the second column, etc., then each processor only has to store this small portion of the matrix or vector in its cache. This matrix or vector is then said to be 'distributed' among the processors.

Some matrices and vectors will be needed by all of the processors; in order to maintain independence of the processors, it is convenient to 'replicate' such matrices and vectors in each processor's cache, meaning that each processor will have its own local copy which it can access independently. This is infeasible for large matrices and vectors because of the large amount of memory that would be required.

The time required for 'communication' between processors must be taken into account in any thorough analysis of parallel algorithms. Communication is required between processors when one processor has information that another processor requires [106]. The algorithms utilized in this chapter result in tasks that are completely independent, in the sense that results from the calculations on one processor will not be required by another processor during the parallel calculation. However, commu-

nication will still be required to distribute tasks to the processors at the beginning of the parallel section of code, and then to combine the results of each individual processor into the final result at the end of the parallel section of code [106]. This will be seen to be important in analysis of the algorithm derived in Section 3.3.

Load balancing is also an important consideration when analysing parallel algorithms. If the set of tasks to be computed in parallel can be split evenly between the processors, the parallelization is said to be 'load balanced' [25, 106]. If this is not the case, then deviations from ideal scaling will be observed. As a simple example, we can consider a parallelization using two processors where one processor is given 99 % of the computation, and the other is given just 1 %. The time taken to carry out this computation in parallel is predicted to be 99 % of the time taken to carry out the serial calculation. If we could instead distribute the computation equally between the processors, making the parallelization load-balanced, we could theoretically finish the computation in 50 % of the time taken for the serial calculation. If load balancing becomes problematic, then load-balancing algorithms can be used [106], but we have not investigated their use in the course of this work.

When performing parallelization to improve performance, we must bear in mind Amdahl's law. Amdahl's law predicts that, unless the entirety of a computation is parallelized, the non-parallelized portions of the calculation will eventually come to dominate the run-time of the computation as the number of processors used is increased [106, 108]. If we take the run-time of the non-parallelized portion of the run-time to be t_{np} and the run-time of the serial calculation to be T, then assuming the parallelized portion is ideally parallelized the run-time of the parallel algorithm will be

$$\frac{T - t_{\mathrm{np}}}{N} + t_{\mathrm{np}} \geq \frac{T}{N}, \tag{3.1}$$

where the right hand side is the predicted run-time given in the opening paragraphs of this section. This law ultimately places a bound on the reduction in run-time that we can hope to achieve using parallel computing.

The rest of this chapter deals with specific methods of parallelizing spin dynamics

simulations. Section 3.1 deals with 'embarrassingly parallel' spin dynamics simulations: simulations that can be trivially separated into separate tasks for the purposes of parallelization. The way in which decomposition of the Liouvillian of a system can allow for parallelization is discussed in Section 3.2. Parallelization of density matrix propagation as given by Equation (1.91) to obtain expectation values is explored in Section 3.3. This latter method has been published [3].

Implementations of the parallelizations given in this chapter utilize the MATLAB Parallel Computing Toolbox [109].

3.1 Embarrassingly parallel simulations

Some computational problems are inherently structured in terms of non-interacting, approximately equally-sized tasks. Each of these tasks can be run independently on a separate processor, allowing for trivial parallelization. These problems are known as 'embarrassingly parallel' [106, 110].

Examples of spin dynamics simulations that have embarrassingly parallel structure are given below. In all of the examples the embarrassingly parallel structure emerges as a loop over independent tasks. This independence allows them to be run in parallel by using the MATLAB command `parfor` [111].

It should be noted that running tasks in parallel in this fashion will require data replication; memory can thus rapidly become a limiting factor on the number of processors that can be utilized simultaneously. If this becomes an issue then other parallelizations, such as those found in Sections 3.2 and 3.3, may be preferred.

3.1.1 Dilute nuclei

Study of a particular molecule by NMR will typically involve a sample that contains a large number of these molecules [83]. These molecules will likely not all be identical, as most nuclei have more than one naturally occurring isotope [112]. Importantly, different isotopes of a nucleus can have different spin properties such as spin quantum number and magnetogyric ratio [32].

The most commonly used nucleus in NMR [113], ^{1}H, constitutes almost all naturally occurring hydrogen (99.9885 %) [32]; in this thesis we have thus ignored any other isotopes of hydrogen. In general, though, when simulating NMR spectra we must take isotopic abundance into account, simulating each possible combination of isotopes in the molecule and combining the results with appropriate weights derived from the relative abundances of the isotopes. The spectrum arising from each combination of isotopes can be simulated independently. Simulation of natural abundance NMR spectra thus has embarrassingly parallel structure.

Because different isotopes can have different spin quantum numbers, the matrices involved in the simulations of different isotopic combinations may be different sizes (Section 1.1.1). The parallelization may thus not be well load balanced. However, load-balancing is trivially achieved in a special case which we now discuss.

It can be the case that NMR-active isotopes of an atom, those with spin quantum numbers greater than zero, have low natural abundance as compared to NMR-inactive isotopes. A common example is carbon: ^{13}C, the most abundant NMR-active carbon isotope, has a natural abundance of just 1.07 % [32]; the major isotope, constituting almost all carbon in nature, is the NMR-inactive ^{12}C [112]. We shall refer to such low natural-abundance spin-active nuclei as 'dilute' nuclei.

While in principle all possible isotopic combinations need to be simulated, in the case of dilute nuclei the proportion of molecules with large numbers of the spin-active isotopes is likely to be very small if natural-abundance samples are used. Explicitly, we can consider a molecule with N nuclei. Assuming that the probability that any one of these nuclei is spin-active is p, the probability that n of the N nuclei in the molecule will be spin-active is given by a binomial distribution [56, 114]:

$$P(n) = p^n(1-p)^{N-n}\frac{N!}{n!(N-n)!}. \tag{3.2}$$

This probability may be so small for relatively large values of n that it is a reasonable approximation to omit molecules with large numbers of the spin-active isotopes. If the molecule is 'small' and the natural abundance of spin-active nuclei very low, then

only simulations of molecules each with a single dilute nucleus may be required. As each simulation will have the same number of spin-active nuclei, the simulation is naturally load-balanced. A simulation computed taking advantage of the parallelization opportunities afforded by the presence of dilute nuclei is shown in Figure 5.3.

It should be noted that the dilute nuclei arguments above do not apply to all pulse sequences. As an example, the INADEQUATE pulse sequence [8, 115] contains a double quantum filter that is designed to remove all of the contributions to a spectrum from molecules with only a single dilute nucleus. In this case, for 'small' molecules only pairs of dilute nuclei need to be included.

3.1.2 Powder averaging grids

The Hamiltonian of a system may depend on orientation (see Section 1.1.2). 'Powder' samples are often used in spectroscopy; these are static samples with isotropically distributed crystallites [116]. This is elaborated further in Section 4.4. In order to simulate the spectra arising from such powders, we need to average over all the orientations.

This averaging can be performed numerically by evaluating the spectrum at a set of discrete points, and then combining with appropriate weights [116]. The set of points is known as a 'grid' of points. The spectrum at each point can be calculated independently of each of the others: this numerical calculation is embarrassingly parallel [117].

Section 4.4, later in this thesis, contains discussion of solid state powder-averaged spectra. Within that section, both Figure 4.3(b) and the 'static' spectrum in Figure 4.1 have been computed using parallelization over independent grid-points.

3.1.3 Dimensions in multi-dimensional experiments

Experiments that generate multidimensional data are common in NMR and ESR [8, 28]. In this section we deal with cases where extra dimensions arise as a result of generating the final observable as a function of two or more durations, labelled t_1, t_2,

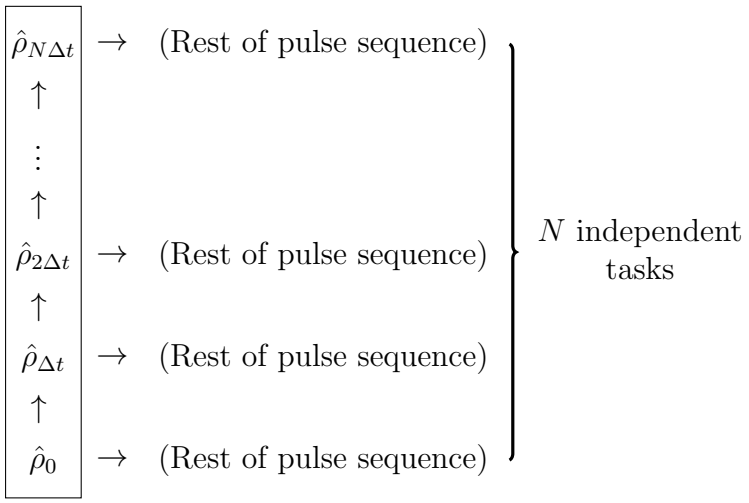


Figure 3.1: Schematic representing simulation of a multidimensional experiment. The density matrix $\hat{\rho}_0$ represents that at the beginning of an indirect detection period. Firstly, a set of density matrices corresponding to each desired value of t_1 ($t_1 = \Delta t, 2\Delta t, \ldots, N\Delta t$) must be computed by propagation; this is represented by the portion of the diagram contained within the box. The rest of the pulse sequence can then be computed independently for each of these density matrices, allowing for parallelization.

etc. Examples of such pulse sequences may be found in Section 5.2 and 5.3. We shall call the periods of the pulse sequence which correspond to these durations 'detection periods'. If there is more than one detection period, then we can make use of this to parallelize the simulation [118], as we shall now show. It should be noted that while we refer to density matrices here, this method can also be applied to state vectors.

Following the methodology of Section 1.5.1, at the end of the first detection period we shall have the density matrix at a number of discrete time points, N. Each of these density matrices must then be propagated through the rest of the pulse sequence. Propagation through the rest of the pulse sequence from each of these N density matrices can be performed independently; there is thus embarrassingly parallel structure, allowing us to run the rest of the pulse sequence in parallel. This is represented schematically in Figure 3.1.

Two multidimensional pulse sequences are discussed in Chapter 5: CLIP-HSQC (Section 5.2), and HN(CO)C$_\alpha$ (Section 5.3). This method of parallelization is not used in the simulations found in either section, however. In the CLIP-HSQC simulation of

Figure 5.3, this is because parallelization over dilute nuclei (Section 3.1.1) is more efficient in that case. For the HN(CO)C_α simulation, Figure 5.7, it is because we instead use the parallelism arising from subspace decomposition, discussed in Section 3.2. This parallelization is used because it has lower memory requirements.

3.2 Decomposition into independent subspaces

It is often the case that a Liouville space is decomposable into subspaces that are 'closed' under the action of the propagator $\exp\{-i\mathcal{L}t\}$. The subspaces are closed under this action in the sense that this propagator cannot transfer population between distinct subspaces. A general method for numerically performing such a decomposition (path tracing) was discussed in Section 2.2.1. Analytical decompositions can be derived from conservation law- and symmetry-considerations [40, 73].

The closed nature of these subspaces means that they can be propagated independently, i.e. the simulation has been rendered embarrassingly parallel by the decomposition. Unlike the cases in Section 3.1, parallelization does not theoretically require any more total memory than would be required to perform the non-decomposed simulation.

Unfortunately, the subspaces will often not be similarly sized. This results in poor load-balancing. The speed-up obtained can, however, still be significant. For this reason, unless other means of parallelization are present, *Spinach* (Section 1.7) defaults to using this method of parallelization, with the subspace decomposition performed using path-tracing.

3.3 Parallel observable computation

A general spin dynamics simulation will have no embarrassingly parallel structure, and a general Hamiltonian must be assumed to be non-decomposable. In order to parallelize a general spin dynamics simulation, then, we must look for opportunities for parallelization in the equations used to perform the simulations.

In this section, we describe investigations into the parallelization of computation of the expectation values of an observable at a series of equispaced time points computed using the density matrix formalism. Such a calculation is characteristic of the direct detection period of pulse sequences investigated in this thesis; an example is found in the pulse–acquire pulse sequence described in Section 1.6.

Computation of the expectation value of an observable at time t requires computation of the density matrix at this time (see Equation (1.29)). Given the density matrix at some initial time (we take this time to be zero for simplicity), Equation (1.91) can be used to give the density matrix at a later time t. This equation may be written in the short-hand form

$$\hat{\rho}_t = \hat{P}_t \, \hat{\rho}_0 \hat{P}_t^\dagger, \tag{3.3}$$

where we abbreviate the propagator (Section 1.5.1) as

$$\hat{P}_t = \exp\Big\{-\mathrm{i}\hat{H}t\Big\}. \tag{3.4}$$

Because the Hamiltonian commutes with itself, the propagator at time t is related to the propagator at time $(t - \Delta t)$ by (Section 1.5.1)

$$\hat{P}_t = \hat{P}_{\Delta t}\hat{P}_{t-\Delta t}. \tag{3.5}$$

In order to generate the density matrix at a series of time points, we can therefore iterate the following equation:

$$\hat{\rho}_{n\Delta t} = \hat{P}_{\Delta t}\hat{\rho}_{(n-1)\Delta t}\hat{P}_{\Delta t}^\dagger. \tag{3.6}$$

Examining Equation (3.6), we see that generation of density matrices at N time points, given the initial density matrix and the propagator, will require $2N$ matrix–matrix multiplications. We will also need to compute N expectation values, using Equation (1.29), from these density matrices. In a typical magnetic resonance experiment N will be of the order of thousands [87]. We therefore concentrate on parallelization of the multiplications in Equation (3.6), as the large number of iterations mean that such a parallelization is likely to significantly reduce the run-time.

A very general method of parallelization would be to perform each of the iterated matrix–matrix multiplications implied by Equation (3.3) using parallel matrix–matrix multiplication algorithms, such as those found in Reference [25], followed by computation of the expectation values at the end of each iteration. Unfortunately, there are a number of problems with this method. Firstly, these matrix–matrix multiplication algorithms require communication at each step, slowing the calculation. Secondly, sparse matrix–sparse matrix multiplication does not parallelize well [92]. We therefore do not use this method.

A method of parallelization that takes advantage of the symmetry found in Equation (3.3) is that due to Skinner and Glaser. They noted that if the density matrix is diagonal, then the calculation of each iterative step, Equation (3.6), splits into calculations that can be performed in parallel [119]. Diagonalization is not always possible (Section 3.3.3), and so in Section 3.3.2 we develop a generalization of this method of parallelization. This generalization can take advantage of the sparsity of the propagator, discussed in Section 3.3.1. This work has been published [3].

Also noteworthy is a method of parallelization, known in the literature, developed to efficiently compute density matrix propagation under time-dependent Hamiltonians. Computing such propagation efficiently reduces the run-time of simulations involving the application of optimal control theory to spin dynamics [120, 121]. In such simulations the Hamiltonian is assumed to be a piecewise-constant function of time (Section 1.5.2). Many propagators thus need to be built, one for each piecewise-constant step. In order to make storage requirements manageable, i.e. so that every processor does not have to store a local copy of every propagator, only one copy of each propagator is made; these propagators are then communicated between nodes in an efficient manner after each multiplication step. This communication, unnecessary for the solution of Equation (3.3), will make the method sub-optimal in our case; for this reason it has not been utilized in the course of this work. This method should be considered, however, for cases where the Hamiltonian is a function of time.

3.3.1 Sparsity considerations

As quantified in Section 1.4.1, spin Hamiltonians are very sparse. Matrix–matrix multiplications are not guaranteed to conserve sparsity; this can be shown by consideration of the definition of matrix–matrix multiplication [19]. The more multiplications, the more dense the resulting matrix is likely to be.

In Section 1.5.1, the concept of a 'short-time propagator' was introduced. There are relatively few matrix–matrix multiplications involved in the computation of such propagators, and so they are likely to retain much of the sparsity of the Hamiltonian from which they are derived. As discussed in Section 1.6, the time-steps taken in the detection period of magnetic resonance experiments often fall within this short-time regime in order to avoid spectral artefacts. This is the segment of a pulse sequence which we wish to simulate in parallel, hence our propagators are likely be very sparse. It is thus reasonable for a parallel algorithm to replicate the propagator.

The density matrix needed in order to compute expectation values using Equation (1.29) results from the repeated application of a propagator in the manner of Equation (3.6). After a few iterations the density matrix is likely to be significantly denser than the Hamiltonian, though it may be hoped for large spin systems that conservation laws [40, 73] ensure that the density matrix does not become too dense to store in main memory. This lead us to the conclusion that it is desirable that any parallelization method distribute the density matrix among the processors.

3.3.2 Parallel algorithm

In this section we develop a parallel algorithm that generates the expectation values of an observable at a number of equispaced time points. This algorithm is a generalization of that presented by Skinner and Glaser [3, 119].

We begin by decomposing the $K \times K$ density matrix at time zero, $\hat{\rho}_0$, into $2K$ separate vectors. This decomposition is shown below to permit separation of the calculation of expectation values into K independent calculations, allowing parallelization.

The form of the decomposition is

$$\hat{\rho}_0 = \sum_{k=1}^{K} p_k \big|u^{(k)}\big\rangle\big\langle v^{(k)}\big|, \tag{3.7}$$

where each p_k is a complex number, each $|u^{(k)}\rangle$ is a row-vector with K complex elements, and each $\langle v^{(k)}|$ is a column-vector with K complex elements. A decomposition of this form always exists, however we leave discussion of its computation until Section 3.3.3 because that discussion is aided by knowledge of the algorithm derived in this section. The decomposition corresponding to diagonalization is that used by Skinner and Glaser in their parallel algorithm [119]. The product between vectors is an 'outer-product' [25], that results, in this case, in a $K \times K$ matrix.

Inserting the decomposition of Equation (3.7) into Equation (3.3) we obtain

$$\hat{\rho}_t = \hat{P}_t \left(\sum_{k=1}^{K} p_k \big|u^{(k)}\big\rangle\big\langle v^{(k)}\big| \right) \hat{P}_t^\dagger \tag{3.8}$$

$$= \sum_{k=1}^{K} p_k \Big(\hat{P}_t \big|u^{(k)}\big\rangle \Big) \Big(\big\langle v^{(k)}\big| \hat{P}_t^\dagger \Big). \tag{3.9}$$

We can abstract the structure of Equation (3.9) by defining the vectors

$$\Big|u_t^{(k)}\Big\rangle = \hat{P}_t \big|u^{(k)}\big\rangle, \ \Big\langle v_t^{(k)}\Big| = \big\langle v^{(k)}\big| \hat{P}_t^\dagger, \tag{3.10}$$

such that Equation (3.9) can be written as

$$\hat{\rho}_t = \sum_{k=1}^{K} p_k \Big|u_t^{(k)}\Big\rangle\Big\langle v_t^{(k)}\Big|. \tag{3.11}$$

Inserting the decomposition of Equation (3.11) into the expectation value definition, Equation (1.29), gives

$$\big\langle \hat{s} \big\rangle_t = \sum_{k=1}^{K} p_k \operatorname{tr}\Big\{ \hat{s} \Big|u_t^{(k)}\Big\rangle\Big\langle v_t^{(k)}\Big| \Big\}, \tag{3.12}$$

where we have made use of the linearity of the trace operation [19]. The validity of cyclic permutation under the trace [56] allows us to write this as

$$\big\langle \hat{s} \big\rangle_t = \sum_{k=1}^{K} p_k \operatorname{tr}\Big\{ \Big\langle v_t^{(k)}\Big| \hat{s} \Big|u_t^{(k)}\Big\rangle \Big\} \tag{3.13}$$

$$= \sum_{k=1}^{K} p_k \Big\langle v_t^{(k)}\Big| \hat{s} \Big|u_t^{(k)}\Big\rangle, \tag{3.14}$$

where the trace operation has been dropped because it now contains a scalar. We write this scalar for brevity as

$$\left\langle \hat{s}^{(k)} \right\rangle_t = \left\langle v_t^{(k)} \middle| \hat{s} \middle| u_t^{(k)} \right\rangle, \tag{3.15}$$

such that Equation (3.14) becomes

$$\langle \hat{s} \rangle_t = \sum_{k=1}^{K} p_k \left\langle \hat{s}^{(k)} \right\rangle_t. \tag{3.16}$$

In order to compute expectation values at a discrete series of time points, Equation (3.12) shows that we require all K $|u_t^{(k)}\rangle$ and $\langle v_t^{(k)}|$ at these points. Noting Equation (3.5), the definitions of the vectors in Equation (3.10) allow us to write

$$\left| u_{n\Delta t}^{(k)} \right\rangle = \hat{P}_{\Delta t} \left| u_{(n-1)\Delta t}^{(k)} \right\rangle, \ \left\langle v_{n\Delta t}^{(k)} \right| = \left\langle v_{(n-1)\Delta t}^{(k)} \right| \hat{P}_{\Delta t}^{\dagger}, \tag{3.17}$$

where we make the identifications

$$\left| u_0^{(k)} \right\rangle = \left| u^{(k)} \right\rangle, \ \left\langle v_0^{(k)} \right| = \left\langle v^{(k)} \right|. \tag{3.18}$$

This implies that each vector can be propagated independently, i.e. we could distribute the vectors among the processors and propagate in parallel. If each processor also has a copy of $\hat{s}$, which we assume to be sparse, then the local expectation values $\langle \hat{s}^{(k)} \rangle_t$ can be generated at the end of each iteration step. After all of the required local expectation values have been generated in parallel, these values can then be combined according to Equation (3.16) to obtain the desired expectation values.

A parallel algorithm utilizing the above ideas is shown in Algorithm 1. Section 3.3.4 contains tests of this algorithm. It should be noted that, in order to minimize memory requirements, only a single set of vectors is stored. These are overwritten with the new vectors at each iteration step. Further, the propagator is replicated on all processors. This is only optimal if this propagator fits within each processor's cache. The discussion of Section 3.3.1 suggests that 'short-time' propagators, as commonly used for the detection periods of magnetic resonance pulse sequences, are likely to be sparse; such propagators thus likely allow good scaling of this algorithm with the number of processors. If the propagators are not sparse, then the algorithm is likely to be non-optimal as compared to dense-matrix parallelization methods.

Algorithm 1 Observable algorithm.

1. Compute $\hat{P}_{\Delta t}$.
2. Compute decomposition of $\hat{\rho}_0$, Equation (3.7).
3. Distribute $|u^{(k)}\rangle$ and $\langle v^{(k)}|$ to processors.
4. Replicate $\hat{P}_{\Delta t}$ and $\hat{s}$ on each processor.
5. In parallel over k:
   for n from 1 to N **do**
   $|u^{(k)}\rangle \leftarrow \hat{P}_{\Delta t}|u^{(k)}\rangle$.
   $\langle v^{(k)}| \leftarrow \langle v^{(k)}|\hat{P}^{\dagger}_{\Delta t}$.
   $\langle \hat{s}^{(k)} \rangle_{n\Delta t} = \langle v^{(k)}|\hat{s}|u^{(k)}\rangle$.
   end for
6. Generate $\langle \hat{s} \rangle_t$ for each t by recombining the appropriate $\langle \hat{s}^{(k)} \rangle_t$ values according to Equation (3.16).

3.3.3 Decomposition methods

Algorithm 1 is written so as to be agnostic to the particular method used to compute the decomposition of the density matrix, Equation (3.7). Possible methods of decomposition include diagonalization [119], singular value decomposition (SVD), *LU* decomposition, and Cholesky factorization [25].

A particularly simple decomposition, which avoids all pre-computation, is to take $|u^{(k)}\rangle$ to be the kth column of the initial density matrix, and to define $\langle v^{(k)}|$ such that the mth element is δ_{km}. With this decomposition, which we shall refer to as the 'decomposition requiring no pre-computation', p_k is unity for all k [3]. As an example, the decomposition of the matrix

$$\begin{pmatrix} 0 & 1/2 \\ 1/2 & 0 \end{pmatrix} \tag{3.19}$$

is

$$\left|u^{(1)}\right\rangle = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix},\ \left\langle v^{(1)}\right| = \begin{pmatrix} 1 & 0 \end{pmatrix},\ p_1 = 1; \tag{3.20}$$

$$\left|u^{(2)}\right\rangle = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix},\ \left\langle v^{(2)}\right| = \begin{pmatrix} 0 & 1 \end{pmatrix},\ p_2 = 1. \tag{3.21}$$

Since a decomposition which does not require pre-computation exists, it may naïvely seem that this decomposition should be used—time spent on pre-computation would appear unnecessary. However, decompositions requiring pre-computation could result in faster calculation if they reduce the time required for each parallel iteration sufficiently. Two decompositions involving pre-computation which could fulfill such a criterion are discussed below. Cholesky factorization and LU decomposition, two decompositions that have efficient sparse implementations [54], do not provide any such benefits [3], and so will not be discussed here.

Diagonalization is one way of computing decompositions of the form found in Equation (3.7). This special case was used by Skinner and Glaser [119]. With diagonalization, p_k are eigenvalues, and the respective eigenvectors are $|u^{(k)}\rangle = \left(\langle v^{(k)}|\right)^\dagger$, as may be confirmed by comparing Equation (3.7) with [119, Equation (3)]. The relationship between the two sets of vectors means that only one of each pair $|u^{(k)}\rangle$, $\langle v^{(k)}|$ needs to be stored and propagated, reducing both memory requirements and computational requirements in the parallel stage of Algorithm 1 as compared to using the method of decomposition requiring no pre-computation. The number of vectors can be further reduced by at least one by suitable preconditioning of the density matrix [119].

Despite the benefit of reducing the number of vectors that have to be propagated, with concomitant reductions in the time required for the latter stages of the algorithm, diagonalization has two large problems if the density matrix to be decomposed is not initially in diagonal form.

Firstly, not every matrix we would be interested in propagating is diagonalizable. In principle every density matrix is Hermitian and thus diagonalizable [36], but in magnetic resonance simulations it is often useful to deal with non-diagonalizable 'density matrices' for reasons of computational efficiency. To demonstrate this, we shall now give a simple example using a single spin-1/2 particle.

It is common in magnetic resonance to compute the expectation value of $\hat{s}_+$ at the end of a pulse sequence [12]. Expanded in terms of spin operators (Section 1.1.1),

the density matrix of a spin-1/2 particle can be written in the form (Section 1.1.3)

$$\hat{\rho} = \mathbb{I}_2/\sqrt{2} + a_+\hat{s}_+ + a_-\hat{s}_- + a_z\hat{s}_z. \tag{3.22}$$

The expectation value of $\hat{s}_+$ is (Equation (1.29))

$$\langle \hat{s}_+ \rangle = \mathrm{tr}\{\hat{s}_+\hat{\rho}\}. \tag{3.23}$$

Because $\hat{s}_+ = \hat{s}_-^\dagger$ (Section 1.1.1), we can write this in terms of a Frobenius inner product (Equation (1.1)):

$$\langle \hat{s}_+ \rangle = \mathrm{tr}\left\{\hat{s}_-^\dagger\hat{\rho}\right\} \tag{3.24}$$

$$= \langle \hat{s}_-, \hat{\rho} \rangle. \tag{3.25}$$

The spin operators are orthogonal with respect to this inner product (Section 1.1.1), and so upon inserting Equation (3.22) into Equation (3.25) and utilizing the linearity of the trace operation [19] we obtain

$$\langle \hat{s}_+ \rangle = a_-\hat{s}_-\langle \hat{s}_-, \hat{s}_- \rangle, \tag{3.26}$$

i.e. only a_- contributes to the desired expectation value.

In the absence of perturbing irradiation, the secular Hamiltonian cannot mix the subspace corresponding to $\hat{s}_-$ with that corresponding to any of the other spin operators; this is because they have different coherence orders (Section 1.5.1). We can therefore omit all but the $\hat{s}_-$ term from the density matrix; this is the density matrix equivalent of destination state screening as found in Section 2.3. The matrix representations of the spin operators are [23]:

$$\mathbb{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \hat{s}_+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},\ \hat{s}_- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix},\ \hat{s}_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.27}$$

This means that the above omission results in a reduction in the number of non-zeroes. While this is not particularly important for the simple case here, when a large number of spins are present in the spin system this reduction can be substantial. A reduction in the number of non-zeroes can result in a reduction in the time required for sparse matrix–sparse matrix multiplication [93], giving rise to better performance.

Unfortunately, $\hat{s}_-$ is not diagonalizable; the reduction in the number of non-zeroes has come at the cost of losing diagonalizability. Reducing the number of non-zeroes in this manner could decrease the computation required to such an extent that it overcomes the reduction in computation that would be gained from using diagonalization as the decomposition method.

A second problem with diagonalization is that even when the density matrix is theoretically diagonalizable, the computation and memory required in order to accurately compute the diagonalization scale rapidly with an increase in spin system size. Matrix sparsity can be utilized using iterative methods if only a few eigenvalues of a sparse matrix [122, 123, 124] are desired, but such methods rapidly run into problems with accuracy arising from a lack of orthogonality of the computed eigenvectors [122]. As such, the only general way to obtain accurate eigenvalues and eigenvectors is using dense matrix arithmetic. This, however, suffers from large computation and storage requirements: for a $K \times K$ matrix, the number of operations required by diagonalization scales asymptotically as K^3, and several dense arrays of size K^2 are required to store intermediate results [125]. These computation and storage requirements restrict the applicability of full diagonalization to very small spin systems.

SVD solves the problem of non-diagonalizability: every matrix has an SVD [78]. SVD could also hold promise in reducing the amount of computation: some of the scalars, p_k, (the 'singular values') could be small enough that the vectors corresponding to these scalars can be dropped from the subsequent computation [25, 78]. Iterative sparse methods exist to compute the SVD of a matrix, but they are plagued by similar issues with accuracy of the computed vectors as in the iterative diagonalization case [126]. Dense methods to compute the SVD scale in the same asymptotic way with matrix dimension as diagonalization [25]. This decomposition is thus also limited to small spin systems.

Because both diagonalization and SVD are incomputable for all but small spin systems, in the following we utilize the decomposition requiring no pre-computation. Benchmarking of Algorithm 1 using this decomposition may be found in Section 3.3.4.

3.3.4 Tests and benchmarking

This section presents tests and benchmarks of Algorithm 1. We utilize the decomposition that requires no pre-computation discussed in Section 3.3.3. The spin system used is that of the protons of 3-phenylmethylene-1H,3H-naphtho-[1,8-c,d]-pyran-1-one, with NMR parameters taken from Reference [127].

Figure 3.2(a) shows the ^{1}H spectrum of 3-phenylmethylene-1H,3H-naphtho-[1,8-c,d]-pyran-1-one simulated using Algorithm 1. While the transformations of Equation (3.3) leading to Algorithm 1 involve no approximations, the re-ordering of operations could in theory result in different results due to the finite precision arithmetic involved in the computation [25, 37]. Figure 3.2(b), a plot of the difference as compared to a simulation of the same spectrum using serial density matrix propagation (Equation (1.91)), shows that such effects are entirely negligible.

Table 3.1 contains the number of propagation steps performed per second when applying Algorithm 1 to the detection period of a pulse–acquire experiment on 3-phenylmethylene-1H,3H-naphtho-[1,8-c,d]-pyran-1-one. This metric is computed by dividing the total number of steps taken ($35,378$, a number of the same order of magnitude as used in high-resolution magnetic resonance [87]) by the amount of time (as measured by an external clock figuratively 'on the wall') taken to compute all of the steps (the wall-clock time [105, 128]). If the time taken for a single step is t_{step}, the time taken for N steps can be assumed in a simple analysis to be Nt_{step}. Ideally, by using n_{proc} processors we could hope to reduce this to $Nt_{\text{step}}/n_{\text{proc}}$. In the ideal case, then, the metric should scale as $n_{\text{proc}}/t_{\text{step}}$, i.e. we should observe linear scaling with the number of processors.

In Table 3.1, doubling the number of processors used results in almost double the number of steps per second up to 8 processors; this is the ideal linear scaling. After this point the speed-up becomes sub-linear with increasing processor count. While not ideal, the scaling still remains good all the way up to the largest number of processors examined, 128.

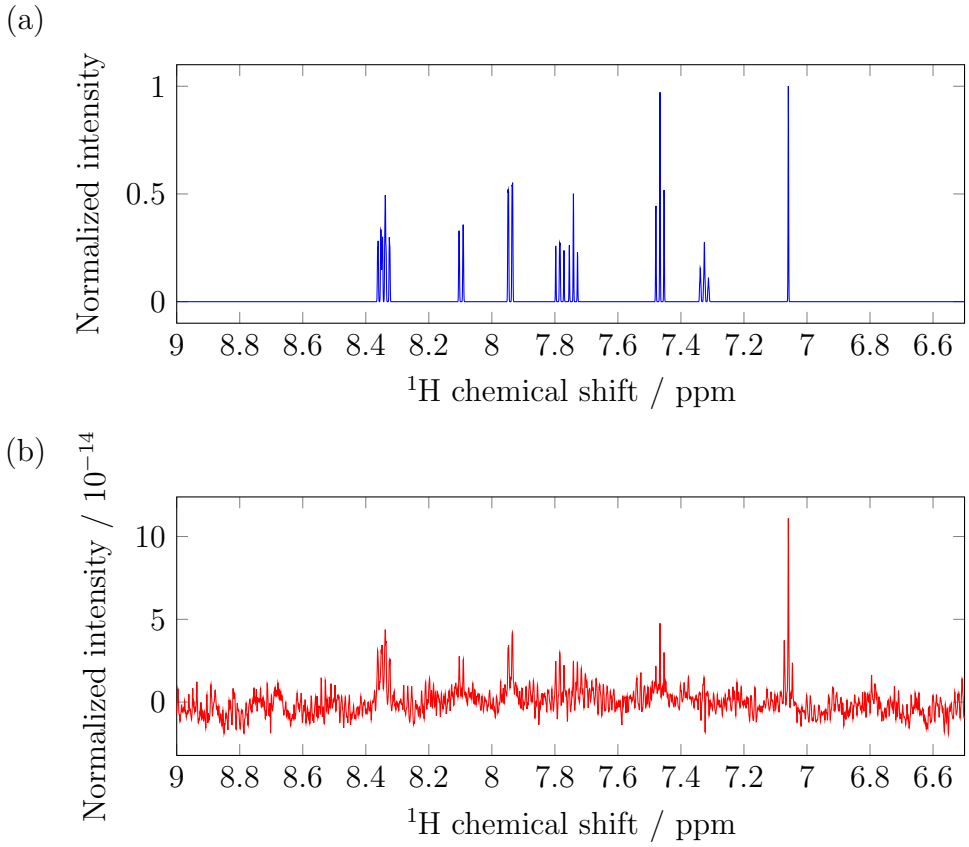


Figure 3.2: (a) ^{1}H spectrum of 3-phenylmethylene-1H,3H-naphtho-[1,8-c,d]-pyran-1-one simulated using the parallel algorithm described in Section 3.3.2. The spin system contains 12 spins. Using four Intel Xeon CPUs, on a computer with 32 of these CPUs and 700 GB of RAM, this calculation took about 4.0 minutes. (b) Difference between the spectrum (a) and the spectrum simulated using traditional serial density matrix propagation. The serial calculation took about 15.5 minutes on the same computer. The spectral difference is entirely negligible because it is much lower than the tolerance of 10^{-9} used when computing the propagator. NMR parameters taken from Reference [127], and the magnetic field strength is 14.1 T. The calculation used 2048 time steps, and the time-step was chosen to obtain the spectral width observed in the plots.

Number of CPU cores	1	2	4	8	16	32	64	128
Time steps per wall clock second	1.2	2.5	4.9	9.9	18.9	29.4	48.4	68.0

Table 3.1: Demonstration of the scaling of Algorithm 1. Bigger numbers are better. Benchmarking was performed using a MATLAB cluster running on 128 Intel Nehalem processors (16 nodes, 8 processors each) with 1.09 TB of RAM. This computing power was provisioned from the Amazon EC2 cloud service [129]. The conditions represent those used to compute the detection period of a ^{1}H pulse–acquire experiment on 3-phenylmethylene-1H,3H-naphtho-[1,8-c,d]-pyran-1-one: the initial state is $\sum_k \hat{L}_+^{(k)}$, where the summation runs over all spins. NMR parameters taken from Reference [127], and the magnetic field strength is 14.1 T. $35,378$ steps were taken, and the time-step was chosen to be $\|\hat{H}\|^{-1}$. These data were collected by Ilya Kuprov and have been published previously in Reference [3].

The non-linearity can be explained qualitatively using Amdahl's law, described in the opening section of this chapter. In order to aid the qualitative application of Amdahl's law to this case, Figure 3.3 shows a schematic decomposition of the wall clock time for Algorithm 1. We have omitted computation of the decomposition (Step 2 of Algorithm 1) as we use the decomposition that requires no pre-computation (Section 3.3.3). Calculation of the propagator represents a constant time cost as it is performed serially. Communication requirements in Algorithm 1, both to distribute/replicate the matrices and vectors before the parallel portion (Steps 3 and 4), and to generate the expectation values from the local expectation values at the end of the parallel portion (Step 6), will increase as the number of processors used increases because increasing the number of processors increases the number of messages that have to be sent. It is possible to make use of some parallelism during communication (see e.g. Reference [130]), but at best this will make communication a constant time cost. As the number of processors increases, the contribution of the parallel portion of the calculation to the total run-time will decrease such that the run-time will come to be dominated by the propagator construction and communication portions of the calculation, leading to sub-linear scaling in accordance with Amdahl's law.

While the above implies that non-linearity will occur eventually as we increase the number of processors, we can pinpoint why in this benchmark it happens when we

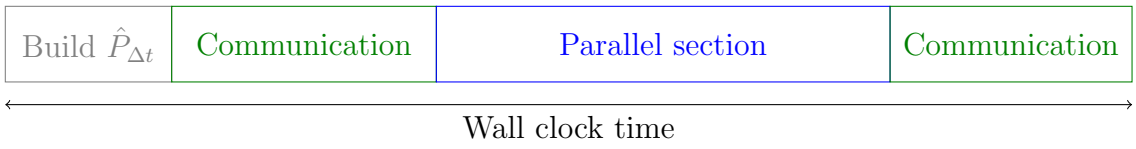


Figure 3.3: Schematic decomposition of wall clock times arising from use of Algorithm 1, used for analysis of the data in Table 3.1. Evaluation of the computation runs from left to right. The first communication step represents distribution of the $|u^{(k)}\rangle$ and $\langle v^{(k)}|$ and replication of $\hat{P}_{\Delta t}$. The second communication step represents construction of all of $\langle S\rangle_t$ at each detected time from components on the separate processors.

increase the number of processors past eight. The processors in the cluster used in this benchmarking are grouped into nodes of eight processors. Communication between nodes will be relatively slow as compared to communication between processors within a node; while the processors within a node will all be on the same chip, the nature of the Amazon EC2 cloud service from which the computing power was provisioned means that there is no guarantee of locality for separate nodes [129]. This means that the communication portions of Figure 3.3 increase sharply in size as we increase the number of processors past eight.

3.3.5 Discussion

With Algorithm 1, we have developed an efficient parallel method to generate expectation values of an observable at a set of equispaced time points.

Examining Algorithm 1, it may be observed that at the end of the parallel section, Step 5, we have a set of $|u^{(k)}\rangle$ and $\langle v^{(k)}|$ that can be used to compute the density matrix $\hat{\rho}_{N\Delta t}$ using Equation (3.11). This implies that, as discussed in Reference [3], we could omit computation of the expectation values within the parallel section of Algorithm 1 and replace Step 6 with recombination using Equation (3.11); in this way we would have an algorithm that computes $\hat{\rho}_{N\Delta t}$ from $\hat{\rho}_0$ in parallel. The algorithm derived in this way is shown in Algorithm 2.

While Algorithm 2 shares the parallel scaling characteristics of Algorithm 1 [3], it is not a very efficient way of computing $\hat{\rho}_{N\Delta t}$. This may be seen when we consider the use case for this algorithm. In order for this algorithm to be effective, the

Algorithm 2 Final state algorithm.

1. Compute $\hat{P}_{\Delta t}$.
2. Compute decomposition of $\hat{\rho}_0$, Equation (3.7).
3. Distribute $|u^{(k)}\rangle$ and $\langle v^{(k)}|$ to processors.
4. Replicate $\hat{P}_{\Delta t}$ on each processor.
5. In parallel over k:
   for n from 1 to N **do**
   $|u^{(k)}\rangle \leftarrow \hat{P}_{\Delta t}|u^{(k)}\rangle$.
   $\langle v^{(k)}| \leftarrow \langle v^{(k)}|\hat{P}^{\dagger}_{\Delta t}$.
   end for
6. Generate $\hat{\rho}_{N\Delta t}$ by recombining all $|u^{(k)}\rangle$, $\langle v^{(k)}|$, and p_k according to Equation (3.11).

number of steps taken, N, must be large such that the section to be run in parallel (Step 5) will dominate the run-time (cf. Figure 3.3). However, instead of taking such a large number of steps, we could instead build $\hat{\rho}_{N\Delta t}$ in a much more efficient way serially. This more efficient method consists of first constructing $\hat{P}_{N\Delta t}$ using scaling and squaring (Section 1.5.1), followed by use of Equation (3.3) to generate $\hat{\rho}_{N\Delta t}$ from this propagator and $\hat{\rho}_0$. For this reason we do not discuss Algorithm 2 further.

A modification of Algorithm 1 that computes and stores all of the components required to build the density matrix at each step (e.g. for indirect detection periods of multi-dimensional pulse sequences; see Section 1.5) is also possible. The modified version of Algorithm 1 that achieves this is shown in Algorithm 3. Unfortunately, testing by Ilya Kuprov has found this algorithm to be unsuitable for efficient parallelization [3]. This is ostensibly because of the communication required to build all of the density matrices from their constituent vectors at the end of the parallel section of Algorithm 3 (Step 6); the costs involved in communicating $2N$ vectors are much larger than those involved in Step 6 of Algorithm 1.

Work is on-going to attempt to apply methodology similar to that used in the de-

Algorithm 3 Indirect detection algorithm.

1. Compute $\hat{P}_{\Delta t}$.
2. Compute decomposition of $\hat{\rho}_0$, Equation (3.7).
3. Distribute $|u_0^{(k)}\rangle = |u^{(k)}\rangle$ and $\langle v_0^{(k)}| = \langle v^{(k)}|$ to processors.
4. Replicate $\hat{P}_{\Delta t}$ on each processor.
5. In parallel over k:
   for n from 1 to N **do**
   $|u_{n\Delta t}^{(k)}\rangle = \hat{P}_{\Delta t}|u_{(n-1)\Delta t}^{(k)}\rangle$.
   $\langle v_{n\Delta t}^{(k)}| = \langle v_{(n-1)\Delta t}^{(k)}|\hat{P}_{\Delta t}^{\dagger}$.
   end for
6. Generate $\hat{\rho}_{n\Delta t}$ for n from 1 to N by recombining all $|u_{n\Delta t}^{(k)}\rangle$, $\langle v_{n\Delta t}^{(k)}|$, and p_k according to Equation (3.11).

velopment of Algorithm 1 to the detection of observables in Liouville space (i.e. parallelization of Equation (1.58) with detection using Equation (1.37)). These attempts have so far not yielded any efficient method of parallel computation, meaning that at present the subspace decomposition method discussed in Section 3.2 is the best method for parallel Liouville space computation, in the absence of embarrassingly parallel structure.

3.4 Conclusions

This chapter has revealed several instances of latent parallelism in spin dynamics simulations. While utilization of this parallelism will never overcome the exponential scaling problem (Section 1.2), it can accelerate simulations that are already possible.

Chapter 4

Liouville space elevation

As discussed in Section 1.5.2, the presence of time-dependence in the Liouvillian complicates magnetic resonance simulations. Section 1.5.2 also included descriptions of a number of ways of overcoming these complications that are commonly used in magnetic resonance; in this chapter we shall discuss a conceptually different method.

The method described in this chapter maps the Liouville space describing the spin dynamics of the system into a much larger space. This mapping is performed in such a way that the effective Liouvillian generating the dynamics in this space is time-independent. The methods of time-independent quantum mechanics (Section 1.5.1) can then be used to propagate the system. Following Pfeifer and Levine we shall refer to this mapping as 'elevation' of the state space [131]. In practice, elevation is effected by introducing additional coordinates into the simulation.

In general, elevation will result in an infinite dimensional state space. This problem is not insurmountable in the cases treated in this chapter: judicious choice of matrix representation allows for truncation to a finite dimensional space.

In the field of time-dependent quantum mechanics, several elevations are already known: (t, t') theory [131, 132], Floquet theory [133, 134], and stochastic Liouville equation (SLE) theory [135, 136, 137]. In Section 4.1 a method of elevating the state spaces of spin systems is derived that includes (t, t') theory and Floquet theory as special cases. This method has been published [4], and allows Liouvillians that depend upon classical coordinates to be mapped into an elevated space. Relations between

the new method and previously known elevation theories are discussed in Section 4.3.

Section 4.2 details the minutiæ involved in building a basis for the elevated state space. These details are required in order to implement the method on a computer.

In Section 4.4 the method of elevation developed in this chapter is used to derive a method for simulating MAS-powder spectra without requiring an explicit grid of points. Further, in Section 4.4.5, it is shown how the novel viewpoint provided by this method could give some insight into the experiences Butler, et al. have had with basis set truncation [80].

4.1 Equation of motion in elevated space

In this section we derive the equation of motion for a state vector in an elevated space.

We start by supposing that the time-dependence of the Liouvillian results from a parametric dependence on some set of N dynamic coordinates, which we express as a vector $\vec{x}$. We thus express the time-dependent Liouvillian as $\mathcal{L}_{\vec{x}}$. One of these coordinates could be 'time'; see Section 4.3.1. These coordinates must be independent of spin. They must also follow classical trajectories [91], i.e. the motion must not be stochastic [138].

Within the above assumptions, given a large enough set of coordinates, the coordinates at time t are a function of t and the coordinates at some arbitrary zero-time, $\vec{x}_0$. Allowing stochastic trajectories would mean that the trajectory could not be uniquely specified by the coordinates at a single time.

The state vector generated by evolution under a Liouvillian with one set of initial coordinates will likely differ from that generated by evolution under a Liouvillian with a different set of initial coordinates. We label these individual state vectors with the initial coordinates: $|\hat{\rho}_{t,\vec{x}_0}\rangle$. The distribution of initial coordinates is given by a probability density function [139], which we write $P(\vec{x}_0)$. The observed state vector $|\hat{\rho}_t\rangle$ is then generated by averaging over the state vectors of all trajectories [139], i.e.

$$|\hat{\rho}_t\rangle = \int_{V_{\vec{x}_0}} |\hat{\rho}_{t,\vec{x}_0}\rangle P(\vec{x}_0)\,\mathrm{d}\vec{x}_0\,. \tag{4.1}$$

There are two important special cases of probability density functions. In the 'single crystal' case $P(\vec{x}_0) = \delta(\vec{x}_0 - \vec{x}_0{}')$, a δ-'function' [73, 140], thus

$$|\hat{\rho}_t\rangle = |\hat{\rho}_{t,\vec{x}_0{}'}\rangle. \tag{4.2}$$

In the isotropic case $P(\vec{x}_0)$ is a constant. This is the situation when performing 'powder averaging' in the solid state; see Section 4.4.3. The density function is normalized, so $P(\vec{x}_0) = (\int_{V_{\vec{x}_0}} \mathrm{d}\vec{x}_0)^{-1}$. The observed state vector is then

$$|\hat{\rho}_t\rangle = \left(\int_{V_{\vec{x}_0}} \mathrm{d}\vec{x}_0\right)^{-1} \int_{V_{\vec{x}_0}} |\hat{\rho}_{t,\vec{x}_0}\rangle \, \mathrm{d}\vec{x}_0 \,. \tag{4.3}$$

With the above definitions, the Liouville–von Neumann equation for a single trajectory is

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}_{t,\vec{x}_0}\rangle = -\mathrm{i}\mathcal{L}_{\vec{x}}|\hat{\rho}_{t,\vec{x}_0}\rangle. \tag{4.4}$$

Unfortunately, even though the Liouvillian is not explicitly time-dependent, the implicit time-dependence still complicates solution of the Liouville–von Neumann equation in the manner discussed in Section 1.5.2. We can, however, construct a solution of Equation (4.4) without having to solve it directly using the elevation method derived below.

To begin, we define an auxiliary state vector $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ which is a function of N auxiliary time-independent coordinates, written as the vector $\vec{y}$. The nth coordinate of this vector is $\vec{y}_n$. We shall eventually select the trajectory of $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ where $\vec{y} = \vec{x}$. The auxiliary state vector is a solution of the Liouville–von Neumann-like equation

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = (-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}})|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle \tag{4.5}$$

with initial condition

$$|\hat{\sigma}_{0,\vec{x}_0,\vec{y}}\rangle = |\hat{\rho}_{0,\vec{x}_0}\rangle \tag{4.6}$$

for all $\vec{y}$. In Equation (4.5) the 'space operator' $\Gamma_{\vec{y}}$ is defined by

$$\Gamma_{\vec{y}} = -\left.\frac{\mathrm{d}\vec{x}}{\mathrm{d}t}\right|_{\vec{x}=\vec{y}} \bullet \vec{\nabla}_{\vec{y}}, \tag{4.7}$$

where $\mathrm{d}\vec{x}/\mathrm{d}t|_{\vec{x}=\vec{y}}$ is the coordinate velocity with any residual dependence on $\vec{x}$ replaced with a dependence on the static coordinates $\vec{y}$;

$$\vec{\nabla}_{\vec{y}} = \begin{pmatrix} \partial/\partial\vec{y}_1 \\ \partial/\partial\vec{y}_2 \\ \vdots \\ \partial/\partial\vec{y}_N \end{pmatrix}; \tag{4.8}$$

and '$\bullet$' is the dot product operation [19]. In general this space operator will be time-dependent. For simplicity we shall assume that it is independent of time; any time-dependence could be formally eliminated by further elevation of the state space (see Section 4.3.1). Equation (4.5) may then be solved to give the auxiliary state vector as

$$|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = \exp\{(-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}})t\}|\hat{\sigma}_{0,\vec{x}_0,\vec{y}}\rangle; \tag{4.9}$$

we note that no time-ordering considerations (Section 1.5.2) have been required in obtaining this solution.

We assert that with the above definitions the state vector $|\hat{\rho}_{t,\vec{x}_0}\rangle$ solving Equation (4.4) can be extracted from $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ via the relation

$$|\hat{\rho}_{t,\vec{x}_0}\rangle = \int \delta(\vec{y} - \vec{x})|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle \,\mathrm{d}\vec{y} \tag{4.10}$$

$$= |\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle, \tag{4.11}$$

where the latter identity arises from contraction using the δ-'function'. This assertion is true by definition at $t = 0$ by virtue of Equation (4.6). At later times, it may be seen that we are selecting the trajectory corresponding to $\vec{y} = \vec{x}$.

In order to prove the assertion found in Equation (4.10), we shall show that the right hand side of Equation (4.10) satisfies the Liouville–von Neumann equation, Equation (4.4), for all t. This manner of proof is similar to one which has been used to prove the validity of the equation of motion of (t, t') theory [132].

Using the chain rule [141, Section 9.6], differentiation of $|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle$ with respect to time may be written

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle = \frac{\partial}{\partial t}|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle + \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}}|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle \tag{4.12}$$

where $\vec{\nabla}_{\vec{x}}$ is defined analogously to $\vec{\nabla}_{\vec{y}}$ in Equation (4.8). Using the left hand side of Equation (4.11), we may write this as

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}_{t,\vec{x}_0}\rangle = \frac{\partial}{\partial t}\,|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle + \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}}|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle. \tag{4.13}$$

We must now evaluate the derivatives on the right-hand side.

Examination of Equations (4.9) and (4.10) allows us to write the partial derivative with respect to time on the right-hand side of Equation (4.13) as

$$\frac{\partial}{\partial t}\,|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle = \frac{\partial}{\partial t}\int \delta(\vec{y}-\vec{x})\exp\{(-\mathrm{i}\mathcal{L}_{\vec{y}}+\Gamma_{\vec{y}})t\}|\hat{\rho}_{0,\vec{x}_0}\rangle\,\mathrm{d}\vec{y}, \tag{4.14}$$

where we have also used the definition in Equation (4.6). The way we have written the $\vec{y}$ dependence of $\Gamma_{\vec{y}}$ in Equation (4.7) belies a difficulty: the effect of composition of $\Gamma_{\vec{y}}$ with the δ-'function' is not well defined. This is because the $\vec{y}$ dependence actually appears as the variable with respect to which partial derivatives are taken. When the space operator is composed with a function, however, we may write that

$$\int \delta(\vec{y}-\vec{x})\Gamma_{\vec{y}}f(\vec{y})\,\mathrm{d}\vec{y} = -\int\left(\delta(\vec{y}-\vec{x})\,\frac{\mathrm{d}\vec{x}}{\mathrm{d}t}\bigg|_{\vec{x}=\vec{y}} \bullet \vec{\nabla}_{\vec{y}}f(\vec{y})\right)\mathrm{d}\vec{y} \tag{4.15}$$

$$= -\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}}f(\vec{x}). \tag{4.16}$$

The definition of the matrix exponential as a Taylor series (Equation (1.10)) reveals that the action of the exponential of the space operator on the state vector can be reduced to iterated application of the space operator in this way, hence we may define the 'space operator'

$$\Gamma_{\vec{x}} = -\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}} \tag{4.17}$$

and write

$$\frac{\partial}{\partial t}\,|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle = \frac{\partial}{\partial t}\exp\{(-\mathrm{i}\mathcal{L}_{\vec{x}}+\Gamma_{\vec{x}})t\}|\hat{\rho}_{0,\vec{x}_0}\rangle. \tag{4.18}$$

With the partial derivative in this form, it can be evaluated to give

$$\frac{\partial}{\partial t}\,|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle = (-\mathrm{i}\mathcal{L}_{\vec{x}}+\Gamma_{\vec{x}})\exp\{(-\mathrm{i}\mathcal{L}_{\vec{x}}+\Gamma_{\vec{x}})t\}|\hat{\rho}_{0,\vec{x}_0}\rangle \tag{4.19}$$

$$= (-\mathrm{i}\mathcal{L}_{\vec{x}}+\Gamma_{\vec{x}})|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle. \tag{4.20}$$

Inserting Equation (4.20) into Equation (4.13) gives

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\rho}_{t,\vec{x}_0}\rangle = \left(-\mathrm{i}\mathcal{L}_{\vec{x}} + \Gamma_{\vec{x}} + \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}}\right)|\hat{\sigma}_{t,\vec{x}_0,\vec{x}}\rangle \tag{4.21}$$

$$= \left(-\mathrm{i}\mathcal{L}_{\vec{x}} + \Gamma_{\vec{x}} + \frac{\mathrm{d}\vec{x}}{\mathrm{d}t} \bullet \vec{\nabla}_{\vec{x}}\right)|\hat{\rho}_{t,\vec{x}_0}\rangle, \tag{4.22}$$

where the latter relation results from substitution of Equation (4.11). Noting the definition of the space operator, Equation (4.17), the present equation reduces to Equation (4.4), the Liouville–von Neumann equation.

We have thus proven that we can obtain the solution of Equation (4.4) in the following manner:

1. Define $|\hat{\sigma}_{0,\vec{x}_0,\vec{y}}\rangle$ by Equation (4.6).
2. Propagate this to the desired time using Equation (4.9).
3. Map $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ into $|\hat{\rho}_{t,\vec{x}_0}\rangle$ using Equation (4.10).

Performing these steps within a computer requires specification of a representation of $|\hat{\sigma}_{0,\vec{x}_0,\vec{y}}\rangle$. That is the topic of Section 4.2.

This procedure does not make calculating the ensemble average of $|\hat{\rho}_{t,\vec{x}_0}\rangle$ over $\vec{x}_0$ (Equation (4.1)) any easier. Section 4.2 includes discussion of how the specification of an explicit basis can make such averages more easily computable.

4.2 Expansion in basis functions

Equation (4.5) defines a time-independent effective Liouvillian, $(-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}})$. In general the coordinates $\vec{y}$ will be continuous: instead of just one continuous variable t, our Liouvillian is now a function of many. The form of the equation of motion, Equation (4.5), is identical to that of the SLE (see e.g. References [136, 137]); we may thus use the methods developed to solve the SLE to solve our equation. One method of SLE solution approximates the continuous variables by a discrete set of points [85, 142], another method expands the auxiliary state vector in a set of mutually-orthogonal basis functions of the coordinates [136, 137]. We opt for the latter because, as we shall

see in Section 4.4, the Liouvillians we wish to use are readily expressible in terms of well-tabulated basis functions.

The manner of expanding the auxiliary state vector in basis functions is detailed below. This results in a set of linear equations that can be represented in terms of matrices and vectors suitable for numerical calculation using a computer. We demonstrate the mapping into vectors and matrices explicitly. The derivation below bears many similarities to the derivation of Section 1.3.1, where an expansion in basis 'functions' was used to derive a Liouvillian mapping. The treatment in Reference [4] follows the same lines as below, but only examines normalized basis functions.

In this derivation, the mth element of the set of basis functions will be denoted $g_{\vec{y}}^{(m)}$. The result of differentiation of the basis functions with respect to elements of $\vec{y}$ is assumed to be expressible as a linear combination of the basis functions. We also assume that the product of any two basis functions can be expressed as a linear combination of the basis functions. The inner product [18] with respect to which the basis functions are mutually orthogonal [19] is denoted by $\langle \cdot\,, \cdot \rangle$. This inner product induces a norm [20], $\| \cdot \| = \sqrt{\langle \cdot\,, \cdot \rangle}$. While the summations will use a single index in this section, a general set of basis functions will have multiple indices (e.g. the Wigner D-functions used in Section 4.4.2). The single index used in this section should thus in general be taken to be an appropriate linear index over all of the indices of the basis functions.

Expansion of a Liouvillian $\mathcal{L}_{\vec{y}}$ in the set of basis functions is written

$$\mathcal{L}_{\vec{y}} = \sum_{m=0}^{\infty} g_{\vec{y}}^{(m)} \mathcal{L}_m, \tag{4.23}$$

where the 'coefficients' $\mathcal{L}_m$ are given by

$$\mathcal{L}_m = \left\| g_{\vec{y}}^{(m)} \right\|^{-2} \left\langle g_{\vec{y}}^{(m)}, \mathcal{L}_{\vec{y}} \right\rangle. \tag{4.24}$$

For the rest of this chapter it will be assumed that the number of terms required in the Liouvillian expansion is finite, i.e. $\mathcal{L}_m = 0$ for all $m > L$; judicious choice of basis functions makes this true for all of the examples given in this chapter. The Liouvillian

expansion will thus be written

$$\mathcal{L}_{\vec{y}} = \sum_{m=0}^{L} g_{\vec{y}}^{(m)} \mathcal{L}_m. \tag{4.25}$$

An expansion of the auxiliary vector $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ in the set of basis functions is assumed to exist; this expansion takes the form

$$|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = \sum_{n=0}^{\infty} g_{\vec{y}}^{(n)} [|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n, \tag{4.26}$$

with

$$[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n = \left\| g_{\vec{y}}^{(n)} \right\|^{-2} \left\langle g_{\vec{y}}^{(n)}, |\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle \right\rangle. \tag{4.27}$$

The 'proof' that this expansion is valid lies in the fact that it does not result in any contradictions during the derivation below. In general this expansion requires an infinite number of terms; whether this is a problem will depend on the specifics of a given simulation. This is examined in more detail in the cases presented in Sections 4.3.2 and 4.4.

In practice, we shall have to truncate the expansion in Equation (4.26) in order to represent it on a computer. This is discussed for two specific cases in Sections 4.3.2 and 4.4.2.

We can now write Equation (4.5) in terms of these basis functions. Upon inserting the expansion of Equation (4.26) into the left-hand side of Equation (4.5) we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = \sum_{n=0}^{\infty} g_{\vec{y}}^{(n)} \frac{\mathrm{d}}{\mathrm{d}t}[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n, \tag{4.28}$$

where we have made use of the time-independence of the basis functions. The right-hand side of Equation (4.5) is a little more complicated: $(-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}})$ can transfer population between the elements $[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n$. We thus write

$$(-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}})|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = \sum_{m,n=0}^{\infty} g_{\vec{y}}^{(m)} [\Lambda]_{mn} [|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n \tag{4.29}$$

where $[\Lambda]_{mn}$ are block-matrix elements effecting this transfer. These elements are defined by

$$[\Lambda]_{mn} = \left\| g_{\vec{y}}^{(m)} \right\|^{-2} \left\langle g_{\vec{y}}^{(m)}, (-\mathrm{i}\mathcal{L}_{\vec{y}} + \Gamma_{\vec{y}}) g_{\vec{y}}^{(n)} \right\rangle. \tag{4.30}$$

Specific examples are given in Sections 4.3.2 and 4.4. Equating the right-hand side of Equation (4.28) and the right-hand side of Equation (4.29) in line with Equation (4.5) yields the equation of motion

$$\sum_{m=0}^{\infty} g_{\vec{y}}^{(m)} \frac{\mathrm{d}}{\mathrm{d}t}[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_m = \sum_{m,n=0}^{\infty} g_{\vec{y}}^{(m)}[\Lambda]_{mn}[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n. \tag{4.31}$$

Equation (4.31) describes a set of linear equations. Taking the inner product of both sides of Equation (4.31) with respect to a given $g_{\vec{y}}^{(m)}$ (and then dividing by $\|g_{\vec{y}}^{(m)}\|^2$ gives the mth equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_m = [\Lambda]_{mn}[|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n. \tag{4.32}$$

A set of linear equations is most conveniently represented as a matrix equation [19] for numerical computation; we thus seek such a representation of Equation (4.31).

In order to express the set of equations implied by Equation (4.32) in terms of explicit vectors and matrices we define an explicit vector basis; we shall write the nth vector as $|n\rangle$, defined such that it has a 'bra–ket' product [140]

$$\langle n|\, m\rangle = \delta_{nm}, \tag{4.33}$$

where $\langle n| = (|n\rangle)^\dagger$.

In terms of this basis, the set of equations implied by Equation (4.32) become

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\sum_{m=0}^{\infty} |m\rangle \otimes [|\hat{\sigma}_{t,\vec{x}_0}\rangle]_m\right) = \left(\sum_{m,n=0}^{\infty} |m\rangle \otimes \langle n| \otimes [\Lambda]_{mn}\right)\left(\sum_{n=0}^{\infty} |n\rangle \otimes [|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n\right), \tag{4.34}$$

where the sums within parentheses describe vectors and matrices:

$$|\hat{\sigma}_{t,\vec{x}_0}\rangle = \sum_{n=0}^{\infty} |n\rangle \otimes [|\hat{\sigma}_{t,\vec{x}_0}\rangle]_n,\ \Lambda = \sum_{m,n=0}^{\infty} |m\rangle \otimes \langle n| \otimes [\Lambda]_{mn}. \tag{4.35}$$

These definitions contain Kronecker product operations as the 'coefficients' are not scalars, but rather themselves vectors/matrices.

The set of equations implied by Equation (4.32) can thus be written in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0}\rangle = \Lambda\hat{\sigma}_{t,\vec{x}_0}. \tag{4.36}$$

This equation can be formally solved to give

$$|\hat{\sigma}_{t,\vec{x}_0}\rangle = \exp\{\Lambda t\}|\hat{\sigma}_{0,\vec{x}_0}\rangle. \tag{4.37}$$

Assuming that the number of basis functions has been truncated such that Λ is a finite matrix, the propagator in this equation may be built using the techniques of Section 1.5.1.

The state vector $|\hat{\sigma}_{t,\vec{x}_0}\rangle$ is mapped back into $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ by multiplication on the left by a 'mapping operator'

$$F(\vec{y}) = \sum_{m=0}^{\infty} g_{\vec{y}}^{(m)}\langle m| \otimes \mathbb{I}, \tag{4.38}$$

where $\mathbb{I}$ is the identity matrix the same size as $\mathcal{L}_{\vec{y}}$, i.e.

$$|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = F(\vec{y})\exp\{\Lambda t\}|\hat{\sigma}_{0,\vec{x}_0}\rangle. \tag{4.39}$$

This may be confirmed by comparing Equations (4.26) and (4.35), noting the definition of the bra–ket product, Equation (4.33), and the relation between Kronecker products and matrix products given by Equation (1.7).

Equation (4.10) implies that generation of the desired state vector $|\hat{\rho}_{t,\vec{x}_0}\rangle$ requires contraction of $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ with a δ-'function'. Performing the contraction over Equation (4.39) gives

$$|\hat{\rho}_{t,\vec{x}_0}\rangle = F(\vec{x})\exp\{\Lambda t\}|\hat{\sigma}_{0,\vec{x}_0}\rangle. \tag{4.40}$$

This implies that we can implicitly perform the contraction by using the mapping operator $F(\vec{x})$ instead of $F(\vec{y})$; the former is thus the operator that we will use as the 'mapping operator' for the rest of this chapter. It should be noted that the ability to perform the contraction in this implicit way is a result of the assumption made in our definition of $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$, Equation (4.26).

As mentioned in Section 4.1, we also generally need to average over the initial coordinates $\vec{x}_0$; having constructed an explicit basis, we are now in a position to

evaluate such an average. Performing averaging according to Equation (4.1) does not in general lead to any significant simplification:

$$|\hat{\rho}_t\rangle = \int_{V_{\vec{x}_0}} F(\vec{x}) \exp\{\Lambda t\} |\hat{\sigma}_{0,\vec{x}_0}\rangle P(\vec{x}_0)\, \mathrm{d}\vec{x}_0\,. \tag{4.41}$$

However, in the case where $|\hat{\sigma}_{0,\vec{x}_0}\rangle$ is independent of $\vec{x}_0$ the average becomes

$$|\hat{\rho}_t\rangle = \left[\int_{V_{\vec{x}_0}} F(\vec{x}) P(\vec{x}_0)\, \mathrm{d}\vec{x}_0\right] \exp\{\Lambda t\} |\hat{\sigma}_{0,\vec{x}_0}\rangle, \tag{4.42}$$

i.e. the averaging is performed over just the mapping operator.

For brevity in later sections we define an averaged mapping operator $\overline{F}$ by

$$\overline{F} = \int_{V_{\vec{x}_0}} F(\vec{x}) P(\vec{x}_0)\, \mathrm{d}\vec{x}_0, \tag{4.43}$$

such that in the case where $|\hat{\sigma}_{0,\vec{x}_0}\rangle$ is independent of $\vec{x}_0$, Equation (4.42) is written simply as

$$|\hat{\rho}_t\rangle = \overline{F} \exp\{\Lambda t\} |\hat{\sigma}_{0,\vec{x}_0}\rangle. \tag{4.44}$$

Implicit contraction and averaging can thus be performed in this case by using $\overline{F}$ as the mapping operator. Simplification arising from this is demonstrated in Sections 4.3.2 and 4.4.3.

4.3 Relation to other elevation methods

As mentioned at the beginning of this chapter, several elevations of the spaces involved in quantum mechanics are already known: (t, t') theory [131, 132], Floquet theory [133, 134], and SLE theory [135, 136, 137]. We shall discuss in this section how they relate to the theory presented in Sections 4.1 and 4.2.

4.3.1 (t, t') theory

In (t, t') theory parameterization is made with respect to an auxiliary time-coordinate, t' [131, 132]. We can relate our theory to (t, t') theory by choosing $\vec{y}$ to be this single auxiliary time-coordinate. The auxiliary state vector is then defined by the relation

$$|\hat{\rho}_{t,\vec{x}_0}\rangle = \int \delta(t - t') |\hat{\sigma}_{t,\vec{x}_0,t'}\rangle\, \mathrm{d}t'\,. \tag{4.45}$$

Examination of Equation (4.7) reveals that the space operator is given in this case by

$$\Gamma_{t'} = -\frac{\partial}{\partial t'}\,. \tag{4.46}$$

Inserting this space operator into Equation (4.5), we derive the equation of motion of (t, t') theory [132]:

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0,t'}\rangle = \left(-\mathrm{i}\mathcal{L}_{t'} - \frac{\partial}{\partial t'}\right)|\hat{\sigma}_{t,\vec{x}_0,t'}\rangle. \tag{4.47}$$

Proceeding in the manner described in Section 4.2 requires specification of a basis. As an example, in Section 4.3.2 we use the equation of motion here derived and the basis set of complex exponential functions to derive Floquet theory.

It should be noted that (t, t') theory is limited in that coordinates that are independent of time cannot be treated within it: an integration would still have to be taken numerically over any static coordinates. The more general theory presented in this chapter is not limited in this way, as is demonstrated in Section 4.4.

4.3.2 Floquet theory

A time-periodic Liouvillian can, in common with any other periodic function, be expanded as a Fourier series [56]. Fourier series expansions have complex exponential functions as basis functions. Floquet's theorem [143] implies that, in addition to an expansion of the Liouvillian, there exists an expansion of the propagator in complex exponential functions [144, 145]. The state vector can thus also be expanded in terms of complex exponential functions.

Using the two expansions described in the previous paragraph, a method of elevation can be constructed that uses complex exponential functions as basis functions [144, 145]. This method is known as 'Floquet theory' [133, 134], and is suitable when the time-dependence of the Liouvillian is periodic.

It is well-known that Floquet theory can be derived from (t, t') theory [132], the theory derived in Section 4.3.1. Below, we perform a derivation of Floquet theory in this way. This allows us to exemplify the techniques developed in Section 4.2.

We express the nth complex exponential basis function in the general form

$$\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)}, \tag{4.48}$$

where ω is the angular frequency of the periodic time dependence, ϕ_0 is an arbitrary phase factor representing the 'initial coordinates', and t' is the auxiliary time coordinate of Section 4.3.1. This set of basis functions has an infinite number of elements: the index n ranges from $-\infty$ to $+\infty$. The physical meaning of ϕ_0 will depend on the physical basis for the periodic time-dependence; see Section 4.4.4 for an example. The complex exponentials thus defined are orthogonal and normalized with respect to the inner product

$$\langle f(t'), g(t')\rangle = \frac{1}{T}\int_0^T f^*(t')g(t')\,\mathrm{d}t', \tag{4.49}$$

where the superscript '$*$' denotes complex conjugation. The upper limit of the integral is the period, $T = 2\pi/\omega$.

We wish to find the elements of the effective Liouvillian, as defined by Equation (4.30). The expansion of the Liouvillian in the basis of complex exponential functions is assumed to be finite and will be written (see Equation (4.25))

$$\mathcal{L}_{t'} = \sum_{m=-L}^{+L} \mathcal{L}_m \mathrm{e}^{\mathrm{i}m(\omega t'+\phi_0)}; \tag{4.50}$$

taking the product of this with $\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)}$ yields

$$\mathcal{L}_{t'}\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)} = \sum_{m=-L}^{+L} \mathcal{L}_m \mathrm{e}^{\mathrm{i}(m+n)(\omega t'+\phi_0)}. \tag{4.51}$$

The space operator, as given in Equation (4.46), applied to the generic complex exponential function defined in Equation (4.48) gives

$$\Gamma_{t'}\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)} = -\frac{\partial}{\partial t'}\,\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)} = -\mathrm{i}n\omega\mathrm{e}^{\mathrm{i}n(\omega t'+\phi_0)}, \tag{4.52}$$

Following the prescription of Equation (4.30), the matrix elements of the effective Liouvillian Λ are found by taking inner products with the linear combination of Equations (4.51) and (4.52); using the inner product defined by Equation (4.49), this results in

$$[\Lambda]_{mn} = -\mathrm{i}(\mathcal{L}_{m-n} + n\omega\delta_{mn}). \tag{4.53}$$

The effective Liouvillian built from these basis elements corresponds to that used in Floquet theory [133, 134].

We write the auxiliary state vector at time t as $|\hat{\sigma}_{t,\phi_0,t'}\rangle$, with 'initial coordinate' ϕ_0 and 'static coordinate' t'. The auxiliary state vector at time zero, as defined by Equation (4.6), is independent of t'. We write this initial state as $|\hat{\rho}_{0,\phi_0}\rangle$. Following Equation (4.27), taking the inner product of $|\hat{\rho}_{0,\phi_0}\rangle$ with each of the complex exponential basis functions in turn gives the elements of the auxiliary state vector at time zero:

$$\left[|\hat{\sigma}_{0,\phi_0}\rangle\right]_n = \delta_{n0}|\hat{\rho}_{0,\phi_0}\rangle. \tag{4.54}$$

This initial state matches that commonly used in Floquet theory [133, 134]. Propagation of this initial state using Equation (4.37) and the representation of the effective Liouvillian given above then gives the representation of the auxiliary state vector at later times; we shall write the elements of this representation as $\left[|\hat{\sigma}_{t,\phi_0}\rangle\right]_n$.

We note that the definition of the initial state representation given by Equation (4.54) implies that only a finite number of elements are non-zero at the beginning of the simulation. Since the expansion of the Liouvillian contains only a finite number of terms by assumption, and the complex exponential functions are eigenfunctions [18] of the space operator (as the representation is diagonal), states 'far away' from the initially populated state ($\left[|\hat{\sigma}_{t,\phi_0}\rangle\right]_n$ with $|n| \gg 0$) can become populated only slowly. This implies that truncation of the infinite state space is possible. By truncation at the nth 'Fourier rank' we shall mean that $\left[|\hat{\sigma}_{t,\phi_0}\rangle\right]_m$ is not included in the basis if $|m| > n$, resulting in a vector with only a finite number of elements. Similarly, the effective Liouvillian is defined such that $\left[\Lambda\right]_{mm'}$ is not part of the matrix representation if $|m| > n$ or $|m'| > n$, giving a matrix of finite size.

Finally, we define the mapping operator, $F(t)$, as described at the end of Section 4.2. In general, this mapping operator is given by

$$F(t) = \sum_{n=-\infty}^{+\infty} \mathrm{e}^{\mathrm{i}n(\omega t+\phi_0)}\langle n| \otimes \mathbb{I}. \tag{4.55}$$

Assuming that $|\hat{\rho}_{0,\phi_0}\rangle$ is independent of ϕ_0, Equation (4.44) implies that we can instead

use the mapping operator

$$F(t) = \sum_{n=-\infty}^{+\infty} \int_0^{2\pi} P(\phi_0) \mathrm{e}^{\mathrm{i}n(\omega t+\phi_0)} \, \mathrm{d}\phi_0 \, \langle n| \otimes \mathbb{I}, \tag{4.56}$$

where $P(\phi_0)$ is the probability density function of ϕ_0 (see Section 4.1). In the case that $P(\phi_0)$ is isotropic, this mapping operator reduces to

$$\overline{F} = \frac{1}{2\pi} \int_0^{2\pi} F(t) \, \mathrm{d}\phi_0 = \langle 0| \otimes \mathbb{I}, \tag{4.57}$$

which is independent of t. The simplification that this gives has been noted previously in the literature [146].

4.3.3 SLE theory

The stochastic Liouville equation (SLE) is an equation identical in form to Equation (4.5) [135, 136, 137]. Where the SLE differs from that equation is that the coordinates on which the original Liouvillian depends, and which are used in the elevation, are not deterministic variables. They are instead stochastic random variables [139, 147]. The stochastic nature of the coordinates complicates the derivation of the SLE (see e.g. Reference [135]) as compared to the derivation in Section 4.1.

It is known that a 'drift' term representing deterministic motion can be included in the SLE space operator, see e.g. Reference [148]. If such a term is included, then taking the limit such that the probability of any stochastic motion tends to zero, if this limit exists, will recover Equation (4.5). This relationship corroborates the validity of the theory presented in this chapter.

4.4 Application to MAS-NMR simulations

Broad spectral peaks caused by anisotropic terms in the Liouvillian (see Section 1.1.2) plague NMR experiments in the solid state [33]. Magic angle spinning (MAS) is conventionally used to eliminate a significant proportion of this broadening from such spectra [33]. MAS consists of spinning the NMR sample about an axis tilted at an angle of $\sim 54.7°$, the 'magic angle', from the direction of the magnetic field.

Spinning of the sample renders the Liouvillian time-dependent. If the spinning is fast enough, and all of the anisotropic terms are secular (see Section 1.1.2), then spinning fast enough to completely average the anisotropic interactions will effectively reduce the Liouvillian to just its isotropic terms [33]. However, if the spinning is slower than this, then sidebands appear in the spectrum [13]. Analysis of such sideband-containing spectra is aided by spectral simulation [51, 66]. As an example, simulations of the sidebands arising in a dipolar-coupled two-spin system at different spinning-rates are shown in Figure 4.1.

In this section we shall use the formalism derived in Sections 4.1 and 4.2 to derive a time-independent effective Liouvillian that we can use to simulate MAS-NMR experiments. This method has been detailed previously in Reference [4].

A further benefit arising from the use of this formalism is presented in Section 4.4.3. The formalism will be shown to allow for analytical averaging over all of the initial coordinates, removing any need for the discrete grids normally used to calculate such spectra [116]. This has been detailed previously in Reference [4]. It should be noted, though, that the simple parallelizability of discrete grids (see Section 3.1) will still make them more appropriate in some situations.

The fresh view that this effective Liouvillian can give on the basis set truncation experiences of Butler, et al. [80] will be discussed in Section 4.4.5.

4.4.1 Derivation of the space operator

The space operator that we require has been derived before in the literature [4, 149]. However, here we provide a different derivation which exemplifies a general scheme for deriving such operators.

The Liouvillian of the spin system of a molecule depends on the orientation of the molecule with respect to the laboratory frame (see Section 1.1.2). For simplicity, we treat a molecule undergoing MAS as a rigid body rotating about an axis passing through its centre of mass. Any molecule in the sample with the same initial orientation must undergo changes of orientation at the same rate, and the Liouvillian is

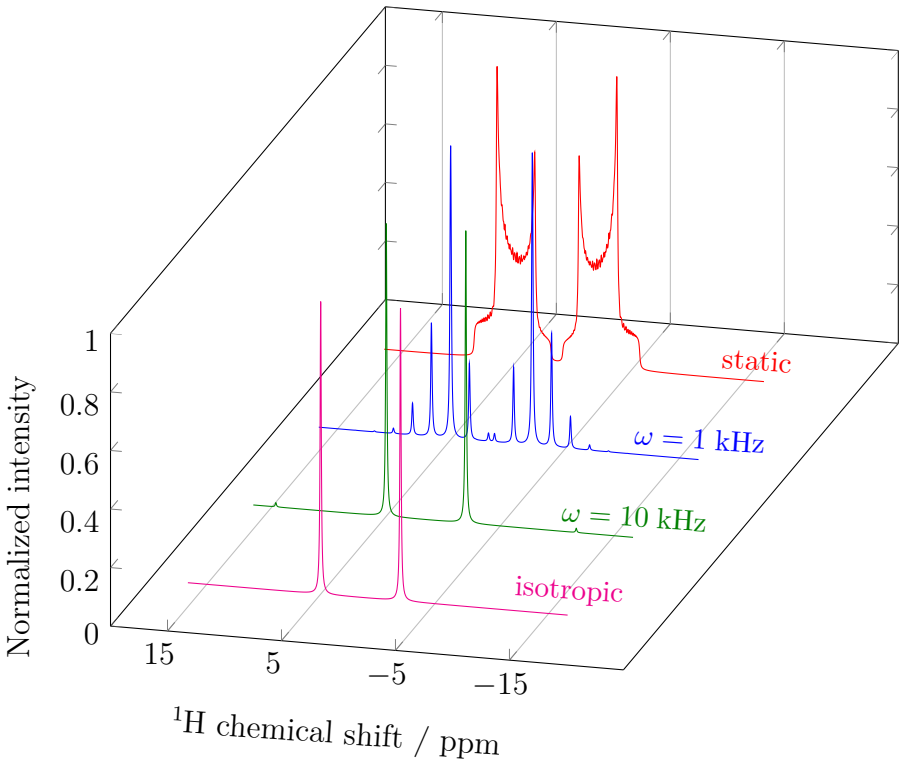


Figure 4.1: Demonstration of spinning sidebands in a dipolar-coupled spin system. Rotation at a rate ω about the magic angle results in the static spectrum being split into spinning sidebands. Rotating faster moves these sidebands outside of the observed window, such that the spectrum tends towards the isotropic spectrum. The spin system consists of a pair of ^{1}H spins, with respective isotropic chemical shifts of 5 ppm and -2 ppm, a distance 0.39 nm apart. The static ($\omega = 0$ Hz) simulation makes use of a rank-131 Gaussian spherical quadrature grid [116]. Powder-averaged MAS spectra (labelled by the magnitude of the angular velocity of rotation, ω) have been simulated using the formalism presented in Section 4.4. A Wigner rank (Section 4.4.2) of 9 was used for the faster rotation, a rank of 17 for the slower rotation. These Wigner ranks were chosen such that the errors due to truncation, quantified by comparison with a simulation with increased rank, are less than 10^{-3} over the whole of each spectrum. The strength of the applied magnetic field is 14.1 T.

assumed to be independent of absolute position in space; examining this special case is thus sufficient.

The orientation of a molecule with respect to the lab-frame is adequately specified by the set of position vectors of the atoms of the molecule; it is these vectors that we shall use as our dynamic coordinates, $\vec{x}$. We shall express the position vector of the jth atom as $\vec{q}^{(j)}$. We shall write the time-independent coordinates collectively as $\vec{y}$; the time-independent vector corresponding to the jth atom will be written $\vec{y}^{(j)}$. With such definitions, the general space operator of Equation (4.7) becomes

$$\Gamma_{\vec{y}} = \sum_k \Gamma_{\vec{y}^{(k)}}, \tag{4.58}$$

where we define

$$\Gamma_{\vec{y}^{(j)}} = \left.\frac{\mathrm{d}\vec{q}^{(j)}}{\mathrm{d}t}\right|_{\vec{x}=\vec{y}} \bullet \vec{\nabla}_{\vec{y}^{(j)}}. \tag{4.59}$$

In Equation (4.58) the summation is taken over all nuclei; this is true of all summations in this subsection. We shall begin by examining a single term of Equation (4.58); subsequently we shall examine the linear combination of these terms that make up the full space operator.

Inspecting Equation (4.59), we see that we need to determine the velocities of the position vectors. We use the Hamiltonian formulation of classical mechanics [91, Section 40] to determine these velocities.

Within the Hamiltonian formulation, velocities are given by the vector equation

$$\frac{\mathrm{d}\vec{q}^{(j)}}{\mathrm{d}t} = \vec{\nabla}_{\vec{p}^{(j)}} H, \tag{4.60}$$

where H is the classical Hamiltonian of the system and $\vec{p}^{(j)}$ is the conjugate momentum to $\vec{q}^{(j)}$ [91, Section 40]. In order to calculate these velocities, we need the classical Hamiltonian of the system.

The classical Hamiltonian of a rotating rigid body in the absence of any external potentials contains just a kinetic energy term, which may be written in terms of the angular velocity $\vec{\omega}$ and the (moment of) inertia tensor $\mathbf{I}$ of the body [91, Section 32]:

$$H = \frac{1}{2}\vec{\omega} \bullet \mathbf{I} \bullet \vec{\omega}. \tag{4.61}$$

We express the angular velocity in terms of its components as

$$\vec{\omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{4.62}$$

in the following. The inertia tensor is independent of the momenta and symmetric [91, Section 32]. Therefore

$$\frac{\partial H}{\partial \vec{p}_i^{(j)}} = \vec{\omega} \bullet \frac{\partial}{\partial \vec{p}_i^{(j)}} (\mathbf{I} \bullet \vec{\omega}). \tag{4.63}$$

We need to express the Hamiltonian, and thus $\mathbf{I} \bullet \vec{\omega}$, in terms of position vectors and their conjugate momenta in order to perform the partial differentiation. Using the identity [91, Section 33]

$$\vec{M} = \mathbf{I} \bullet \vec{\omega}, \tag{4.64}$$

where $\vec{M}$ is the angular momentum vector of the body, and the definition of angular momentum in terms of position vectors and momenta [91, Section 9]

$$\vec{M} = \sum_k \vec{q}^{(k)} \times \vec{p}^{(k)} \tag{4.65}$$

this derivative may be written

$$\frac{\partial H}{\partial \vec{p}_i^{(j)}} = \vec{\omega} \bullet \frac{\partial}{\partial \vec{p}_i^{(j)}} \left(\sum_k \vec{q}^{(k)} \times \vec{p}^{(k)} \right). \tag{4.66}$$

In these equations '$\times$' is the cross product [19]. Carrying out the partial differentiation gives

$$\frac{\partial H}{\partial \vec{p}_i^{(j)}} = \vec{\omega} \bullet \vec{q}^{(j)} \times \vec{\mathrm{e}}_i, \tag{4.67}$$

where $\vec{\mathrm{e}}_i$ is the unit vector

$$\vec{\mathrm{e}}_i = \begin{pmatrix} \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \end{pmatrix}, \tag{4.68}$$

so the velocities are (Equation (4.60))

$$\frac{\mathrm{d}\vec{q}^{(j)}}{\mathrm{d}t} = \begin{pmatrix} \vec{\omega} \bullet \vec{q}^{(j)} \times \vec{\mathrm{e}}_1 \\ \vec{\omega} \bullet \vec{q}^{(j)} \times \vec{\mathrm{e}}_2 \\ \vec{\omega} \bullet \vec{q}^{(j)} \times \vec{\mathrm{e}}_3 \end{pmatrix}. \tag{4.69}$$

Upon examination of Equation (4.59), we see that we need to obtain the vector $\mathrm{d}\vec{q}^{(j)}/\mathrm{d}t\big|_{\vec{x}=\vec{y}}$ from Equation (4.69); performing the appropriate replacements gives

$$\frac{\mathrm{d}\vec{q}^{(j)}}{\mathrm{d}t}\bigg|_{\vec{x}=\vec{y}} = \begin{pmatrix} \vec{\omega} \bullet \vec{y}^{(j)} \times \vec{\mathrm{e}}_1 \\ \vec{\omega} \bullet \vec{y}^{(j)} \times \vec{\mathrm{e}}_2 \\ \vec{\omega} \bullet \vec{y}^{(j)} \times \vec{\mathrm{e}}_3 \end{pmatrix}. \tag{4.70}$$

Following Equation (4.59), taking the dot product of this vector with $\vec{\nabla}_{\vec{y}^{(j)}}$ gives the jth space operator:

$$\Gamma_{\vec{y}^{(j)}} = \vec{\omega} \bullet \vec{y}^{(j)} \times \vec{\nabla}_{\vec{y}^{(j)}}. \tag{4.71}$$

The vector formed by the cross product on the right hand side is proportional to the angular momentum operator (in the coordinate representation) used in quantum mechanics [17, Section 2.2],

$$\vec{S}^{(j)} = -\mathrm{i}\vec{y}^{(j)} \times \vec{\nabla}_{\vec{y}^{(j)}}, \tag{4.72}$$

allowing for the simplification

$$\Gamma_{\vec{y}^{(j)}} = \mathrm{i}\vec{\omega} \bullet \vec{S}^{(j)}. \tag{4.73}$$

Given the result of Equation (4.73), the space operator of Equation (4.58) becomes

$$\Gamma_{\vec{y}} = \sum_k \mathrm{i}\vec{\omega} \bullet \vec{S}^{(k)}. \tag{4.74}$$

We can define a total 'angular momentum operator' by [17, 22, 23]

$$\vec{S} = \sum_k \vec{S}^{(k)} = \begin{pmatrix} \hat{S}_x \\ \hat{S}_y \\ \hat{S}_z \end{pmatrix}. \tag{4.75}$$

Inserting this into Equation (4.74) gives

$$\Gamma_{\vec{y}} = \mathrm{i}\vec{\omega} \bullet \vec{S}. \tag{4.76}$$

This is the form of the space operator that we shall use.

4.4.2 Expansion in Wigner D-functions

With the space operator defined by Equation (4.76), the equation of motion of an auxiliary state vector $|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle$ is

$$\frac{\mathrm{d}}{\mathrm{d}t}|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle = -\mathrm{i}\Big(\mathcal{L}_{\vec{y}} - \vec{\omega} \bullet \vec{S}\Big)|\hat{\sigma}_{t,\vec{x}_0,\vec{y}}\rangle. \tag{4.77}$$

As discussed in Section 4.2, we wish to write this equation in terms of a set of basis functions in order to build a matrix representation. This will allow us to solve this equation of motion computationally.

In SLE simulations of rotational diffusion [136], it is usual to use as spatial basis functions the Wigner D-functions [17, Chapter 4]. We shall denote the (klm)th Wigner D-function by D^k_{lm}. The index k, referred to in the following as the 'Wigner rank' (or just 'rank' when this not ambiguous) of the Wigner D-function, runs from 0 to $+\infty$. For a given rank k the indices l and m both range from $-k$ to $+k$. We utilize these Wigner D-functions as basis functions for two reasons. Firstly, the action of the terms in the space operator of Equation (4.76) on these functions is well documented (see e.g. Reference [17, Chapter 4]). Secondly, the Liouvillian of a mechanically rotated spin system can typically be expanded in terms of such functions, as described below.

Unfortunately, manipulation of these basis functions is more complicated than manipulation of the complex exponential basis functions used in Section 4.3.2; the notation must likewise become more complicated. Typically, parameterization of the Wigner D-functions is made with respect to three Euler angles [17, Section 1.4], α, β, and γ; the inner product with respect to which these functions are orthogonal is then written [17, Section 4.10]

$$\begin{aligned}&\langle f(\alpha,\beta,\gamma), g(\alpha,\beta,\gamma)\rangle \\ &\quad = \int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi} f^*(\alpha,\beta,\gamma)g(\alpha,\beta,\gamma)\,\mathrm{d}\alpha\sin\beta\,\mathrm{d}\beta\,\mathrm{d}\gamma\,.\end{aligned} \tag{4.78}$$

These Euler angles are sufficient to describe rotations in 3D space [91]. The Euler angle-arguments of the Wigner D-functions will normally be suppressed for brevity.

As Euler angles provide a convenient complete parameterization of rotations, we utilize them instead of the space coordinates implied in Section 4.4.1; we can do this because application of our space operator to Wigner D functions expressed in terms of such angles is well defined (as mentioned above). We define the Euler angles $(\alpha_t, \beta_t, \gamma_t)$ to be the dynamic coordinates; this set of angles takes the special value $(\alpha_0, \beta_0, \gamma_0)$ at time zero. The time-independent coordinates will be written with no subscripts: (α, β, γ).

Using the above definitions, we now proceed to generate the various components detailed in Section 4.2 using these basis functions.

Effective Liouvillian

We split derivation of the effective Liouvillian, Λ, in this basis into two parts: the first deriving a representation of $\mathcal{L}$, the second a representation of Γ, where we omit the subscript coordinate dependences for clarity. In line with Equation (4.30), the elements of the effective Liouvillian are computed from these separate representations by

$$[\Lambda]_{klm,k'l'm'} = -\mathrm{i}[\mathcal{L}]_{klm,k'l'm'} + [\Gamma]_{klm,k'l'm'}. \tag{4.79}$$

A typical NMR Liouvillian can be written in spherical tensor form as [8, 33]

$$\mathcal{L} = D^0_{00}\mathcal{L}_{\mathrm{iso}} + \sum_{L,M=-2}^{+2} D^2_{LM}\mathcal{Q}_{LM}, \tag{4.80}$$

where $\mathcal{L}_{\mathrm{iso}}$ is the isotropic Liouvillian (consisting of terms not averaged to zero by isotropic tumbling in solution), and $\mathcal{Q}_{km}$ are various second-rank anisotropic terms. This corresponds to an expansion of the Liouvillian in Wigner D-functions. We shall deal exclusively with Liouvillians that can be expanded in the form of Equation (4.80). Rank one terms can exist, but their effects on spectra require special efforts to detect [33], and so they shall be ignored here for brevity. We assume in the following that time-dependence of the Liouvillian is exclusively due to MAS modulating the Euler angles that the Wigner D-functions take as arguments.

We need to compute the elements of $\mathcal{L}$ in the elevated representation. Examining Equation (4.30) reveals that this is effected by computing

$$[\mathcal{L}]_{klm,k'l'm'} = \left\|D^k_{lm}\right\|^{-2}\left\langle D^k_{lm}, \mathcal{L}D^{k'}_{l'm'}\right\rangle \tag{4.81}$$

where

$$\left\|D^k_{lm}\right\| = \sqrt{\left\langle D^k_{lm}, D^k_{lm}\right\rangle}. \tag{4.82}$$

Inserting Equation (4.80) into Equation (4.81) reveals that the inner product will involve products of Wigner D-functions.

The product of two Wigner D-functions is given by [17, Section 4.6]:

$$D^k_{lm}D^{k'}_{l'm'} = \sum_{K=|k-k'|}^{k+k'}\sum_{M,N=-K}^{+K} C^{KL}_{kl,k'l'}C^{KM}_{km,k'm'}D^K_{LM}, \tag{4.83}$$

where $C^{KL}_{kl,k'l'}$ are Clebsch–Gordan coefficients [17, Chapter 8]. Inserting this into Equation (4.81), along with the Liouvillian as given by Equation (4.80), and making use of orthogonality gives

$$[\mathcal{L}]_{klm,k'l'm'} = C^{kl}_{00,k'l'}C^{km}_{00,k'm'}\mathcal{L}_{\text{iso}} + \sum_{L,M=-2}^{+2} C^{kl}_{2L,k'l'}C^{km}_{2M,k'm'}\mathcal{Q}_{LM}, \tag{4.84}$$

the representation of the Liouvillian part of the effective Liouvillian.

We now move onto computation of the space operator part of the effective Liouvillian. Expanding the space operator Γ, derived in Section 4.4.1, into its components gives

$$\Gamma = \mathrm{i}\vec{\omega} \bullet \vec{S} \tag{4.85}$$

$$= \mathrm{i}\Big(\omega_1\hat{S}_x + \omega_2\hat{S}_y + \omega_3\hat{S}_z\Big). \tag{4.86}$$

The element $[\Gamma]_{klm,k'l'm'}$ may thus be expressed as

$$[\Gamma]_{klm,k'l'm'} = \mathrm{i}\left(\omega_1\left[\hat{S}_x\right]_{klm,k'l'm'} + \omega_2\left[\hat{S}_y\right]_{klm,k'l'm'} + \omega_3\left[\hat{S}_z\right]_{klm,k'l'm'}\right). \tag{4.87}$$

The three individual components may be conveniently built separately and then combined to construct the space operator. We will deal with each of the components in turn.

The x component can be found by decomposing $\hat{S}_x$ into a raising and a lowering operator: an operator that increases the value of l by unity, and an operator that decreases the value of l by unity, respectively. Explicitly [17, Section 4.2],

$$\hat{S}_x D^{k'}_{l'm'} = \frac{1}{2}(S_+ + S_-) D^{k'}_{l'm'} \tag{4.88}$$

$$= \frac{1}{2}\sqrt{\frac{k'(k'+1) - l'(l'-1)}{2}} D^{k'}_{l'-1,m'} \tag{4.89}$$

$$- \frac{1}{2}\sqrt{\frac{k'(k'+1) - l'(l'+1)}{2}} D^{k'}_{l'+1,m'}.$$

We may thus write that

$$\mathrm{i}\omega_1 \left[\hat{S}_x\right]_{klm,k'l'm'} = \frac{\mathrm{i}\omega_1}{2}\delta_{kk'}\delta_{mm'} \cdot \tag{4.90}$$

$$\cdot \left(\sqrt{\frac{k'(k'+1) - l'(l'-1)}{2}}\delta_{l,l'-1} - \sqrt{\frac{k'(k'+1) - l'(l'+1)}{2}}\delta_{l,l'+1} \right)$$

by inspection.

The operator $\hat{S}_y$ is similarly decomposed [17, Section 4.2]:

$$\hat{S}_y D^{k'}_{l'm'} = \frac{1}{2\mathrm{i}}(S_+ - S_-) D^{k'}_{l'm'} \tag{4.91}$$

$$= \frac{1}{2\mathrm{i}}\sqrt{\frac{k'(k'+1) - l'(l'-1)}{2}} D^{k'}_{l'-1,m'} \tag{4.92}$$

$$+ \frac{1}{2\mathrm{i}}\sqrt{\frac{k'(k'+1) - l'(l'+1)}{2}} D^{k'}_{l'+1,m'},$$

to give the y component

$$\mathrm{i}\omega_2 \left[\hat{S}_y\right]_{klm,k'l'm'} = \frac{\mathrm{i}\omega_2}{2\mathrm{i}}\delta_{kk'}\delta_{mm'} \cdot \tag{4.93}$$

$$\cdot \left(\sqrt{\frac{k'(k'+1) - l'(l'-1)}{2}}\delta_{l,l'-1} + \sqrt{\frac{k'(k'+1) - l'(l'+1)}{2}}\delta_{l,l'+1} \right),$$

again by inspection.

Wigner D-functions are eigenfunctions of $\hat{S}_z$ [17, Section 4.2]:

$$\hat{S}_z D^k_{lm} = -l D^k_{lm}. \tag{4.94}$$

Thus

$$\mathrm{i}\omega_3 \left[\hat{S}_z\right]_{klm,k'l'm'} = -\mathrm{i}\omega_3 l \delta_{kk'}\delta_{ll'}\delta_{mm'} \tag{4.95}$$

is the z component.

With the identification of this final component, we have completely described all of the individual parts of the effective Liouvillian. Summation of the appropriate terms according to Equations (4.86) and (4.79) then completes the construction.

It is often preferable to use normalized Wigner D-functions, which we shall here write $D^k_{lm}/\|D^k_{lm}\|$, as basis functions. With such basis functions the matrix representation of the effective Liouvillian is Hermitian (cf. Section 1.3.1). The representation of Λ^{norm}, the effective Liouvillian in the normalized basis, can be derived from the definition of the elements of Λ in the unnormalized basis. The definition of an element of the unnormalized effective Liouvillian, Equation (4.30), may be rewritten as

$$[\Lambda]_{klm,k'l'm'} = \left\|D^k_{lm}\right\|^{-2}\left\langle D^k_{lm}, (-\mathrm{i}\mathcal{L}+\Gamma_{\vec{y}})D^{k'}_{l'm'}\right\rangle \tag{4.96}$$

$$= \frac{\|D^{k'}_{l'm'}\|}{\|D^k_{lm}\|}\left\langle D^k_{lm}/\left\|D^k_{lm}\right\|, (-\mathrm{i}\mathcal{L}+\Gamma_{\vec{y}})\left(D^{k'}_{l'm'}/\left\|D^{k'}_{l'm'}\right\|\right)\right\rangle, \tag{4.97}$$

where we have substituted in the specific basis functions used in this section. The inner product on the right-hand side can be recognized as an element of Λ^{norm}:

$$[\Lambda^{\text{norm}}]_{klm,k'l'm'} = \left\langle D^k_{lm}/\left\|D^k_{lm}\right\|, (-\mathrm{i}\mathcal{L}+\Gamma_{\vec{y}})\left(D^{k'}_{l'm'}/\left\|D^{k'}_{l'm'}\right\|\right)\right\rangle. \tag{4.98}$$

Elements of Λ^{norm} are thus related to elements of Λ by

$$[\Lambda]^{\text{norm}}_{klm,k'l'm'} = \frac{\|D^k_{lm}\|}{\|D^{k'}_{l'm'}\|}[\Lambda]_{klm,k'l'm'}. \tag{4.99}$$

Auxiliary state vector

Following Equation (4.6), the auxiliary state vector at time zero, $|\hat{\sigma}_{0,(\alpha_0,\beta_0,\gamma_0),(\alpha,\beta,\gamma)}\rangle$, is defined to be independent of (α,β,γ). Examining the definition of the inner product, Equation (4.78), reveals that the result of taking the inner product of the initial state with each Wigner D-function is zero with every function except the (000)th element. Thus Equation (4.27) implies that the (klm)th element of the representation of our auxiliary state vector at time zero is

$$\left[\left|\hat{\sigma}_{0,(\alpha_0,\beta_0,\gamma_0)}\right\rangle\right]_{klm} = \delta_{k0}\delta_{l0}\delta_{m0}\left|\hat{\rho}_{0,(\alpha_0,\beta_0,\gamma_0)}\right\rangle. \tag{4.100}$$

This initial state can then be propagated, as per Equation (4.37), to obtain the representation of the auxiliary state vector at later times.

Mapping operator

The mapping operator at time t is given by

$$F(\alpha_t, \beta_t, \gamma_t) = \sum_{klm} D^k_{lm}(\alpha_t, \beta_t, \gamma_t)\langle klm| \otimes \mathbb{I}, \tag{4.101}$$

as may be seen from inspection of Equation (4.38). Averaging of this operator over the initial coordinates, as discussed at the end of Section 4.2, is examined in Section 4.4.3.

Truncation

As in the Floquet case in Section 4.3.2, the initial auxiliary state vector (Equation (4.100)) has only a finite number of non-zero elements. The space operator can only map between elements of the same Wigner rank (see above), and the Liouvillian is assumed to have a finite expansion in Wigner D-functions. This again implies that truncation is valid for finite time simulations. By truncation at the nth Wigner rank we shall mean that the (klm)th element of the auxiliary state vector, $\left[|\hat{\sigma}_{t,(\alpha_0,\beta_0,\gamma_0)}\rangle\right]_{klm}$, will be omitted from the basis if $k > n$, giving a vector with only a finite number of elements. Similarly, the effective Liouvillian will be truncated such that $[\Lambda]_{klm,k'l'm'}$ is absent from the basis if $k > n$ or $k' > n$, giving a square matrix of finite size.

4.4.3 Powder averaging

Many solid state spectra are of 'powders'. A powder sample is a solid sample consisting of isotropically distributed crystallites. It is usual to assume that the distribution of orientations of these crystallites is continuous: powder averaging is then equivalent to averaging over the initial coordinates with constant probability density function $P(\vec{x}_0) = (8\pi^2)^{-1}$ (Equation (4.3)), i.e. [116]

$$|\hat{\rho}_t\rangle = \frac{1}{8\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi} |\hat{\rho}_{t,\alpha_0,\beta_0,\gamma_0}\rangle \,\mathrm{d}\alpha_0 \sin\beta_0 \,\mathrm{d}\beta_0 \,\mathrm{d}\gamma_0 \,. \tag{4.102}$$

Integration over the initial coordinates for simulation purposes is conventionally performed numerically as so [116]: a discrete grid is constructed, the individual crystallite spectra at each point are simulated separately, and they are then summed with the appropriate weighting. This method of integration is trivially parallelizable, as discussed in Section 3.1.2. However, the number of points required to obtain convergence to within a certain tolerance can be very large. Carousel averaging can be used to perform integration over one angle analytically using the γ-COMPUTE [60] or Floquet [146] formalisms, but even with this simplification two angles still have to be treated numerically.

If the initial state is independent of the initial coordinates, then the present formalism may be used to perform powder averaging analytically. Following the discussion of Section 4.2, averaging over the initial coordinates is performed by averaging over just the mapping operator, as described by Equation (4.43). The mapping operator in this case, Equation (4.101), is a linear combination of the Wigner D-functions $D^k_{lm}(\alpha_t, \beta_t, \gamma_t)$. These functions can be thought of as the representation of two consecutive rotations: first a rotation from (0,0,0), the arbitrarily defined origin, to $(\alpha_0, \beta_0, \gamma_0)$, the initial set of coordinates; then a rotation from $(\alpha_0, \beta_0, \gamma_0)$ to $(\alpha_t, \beta_t, \gamma_t)$. The definition of Wigner D-functions resulting from successive rotations [17, Section 4.7] implies that at time t a given Wigner D-function, D^k_{lm}, will be a linear combination of the initial Wigner D-functions of the same rank, k. As [17, Section 4.11]

$$\frac{1}{8\pi^2}\int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi} D^k_{lm}(\alpha_0, \beta_0, \gamma_0)\,\mathrm{d}\alpha_0 \sin\beta_0\,\mathrm{d}\beta_0\,\mathrm{d}\gamma_0 = \delta_{k0}\delta_{l0}\delta_{m0}, \tag{4.103}$$

powder-averaging over all initial coordinates in the mapping operator is equivalent to using the averaged mapping operator

$$\overline{F} = \langle 000| \otimes \mathbb{I}. \tag{4.104}$$

In the powder-averaged case the mapping operator thus becomes as simple as the initial state, Equation (4.100), mirroring the simplification found in the Floquet case (Section 4.3.2).

We make the assumption for the rest of the chapter that initial state vectors are independent of the initial coordinates. This allows us to use the mapping operator defined by Equation (4.104) for powder averaging. As the present theory does not require the use of discrete grids, it will be referred to as the 'grid-free' formalism in the following. Example simulations making use of this grid-free formalism are shown in Figures 4.1 and 4.2.

4.4.4 Comparison with Floquet theory

The periodic time-dependence of the Liouvillian due to MAS may be accounted for in simulations using Floquet theory [13, 133, 134]. However, unlike with the grid-free formalism, powder averaging in the Floquet case will still require numerical averaging over a grid of points (see Section 4.4.3). It is pertinent to compare the grid-free and Floquet methods for simulating powder-averaged MAS spectra.

Presented in Table 4.1 are matrix statistics of the Liouvillians involved in simulations of the spectrum shown in Figure 4.2(a) using the Floquet and grid-free formalisms. The matrices are sparse (Section 1.4), and so we also give the number of non-zeroes in these matrices. In the Floquet case, powder averaging over the Euler angle γ may be performed analytically by identifying this angle with the initial phase ϕ_0 (see Section 4.3.2), meaning that only a two-angle grid is needed for powder averaging [13]. The statistics for the three-angle simulation (which will be required in general [116]) are also shown for contrast.

For the two-angle Floquet simulation the total state vector size (product of number of grid points and state vector length at each grid point) is less than that required for the grid-free approach; it is approximately half. Furthermore, the total number of non-zero elements involved in the two-angle Floquet simulation is only approximately one-tenth the number required in the grid-free approach. Comparison to the three-angle case is much more favourable to the grid-free method: both total state vector size and total number of non-zero elements are smaller in the grid-free case.

These comparisons of total size overlook some important facts, however. Firstly,

	Floquet (two-angle grid)	Floquet (three-angle grid)	Grid-free
Number of grid points	194	4,656	1
State vector length for each point	6,016	6,016	2,358,272
Number of nonzeroes at each point	$\sim 1.73 \times 10^5$	$\sim 1.73 \times 10^5$	$\sim 2.93 \times 10^8$

Table 4.1: Comparison of the size of auxiliary state vectors involved in the simulation of the ^{13}C proton-decoupled powder-averaged solid state MAS spectrum of alanine [150] shown in Figure 4.2(a). The spin system contains three ^{13}C and one ^{15}N, and the spin system parameters are taken from a DFT calculation using Gaussian 09 performed by Ilya Kuprov. The initial state is $\sum_k |\hat{I}_x^{(k)}\rangle$, with the sum running over all ^{13}C nuclei, representing approximately the result of a $\pi/2$ radiofrequency pulse applied to the ^{13}C nuclei at thermal equilibrium. The detection state is $\sum_k \langle \hat{I}_+^{(k)}|$, with the sum again running over all ^{13}C nuclei. The rate of rotation about the magic angle is 2 kHz, and the applied magnetic field strength is 14.1 T. Transverse states of the ^{15}N are not included in the basis as they cannot be populated by the secular Liouvillian (Section 1.5.2). The Floquet and grid-free function-bases are both truncated at rank 23 (Floquet and Wigner rank, respectively), the two-angle grid is a rank 23 Lebedev grid, and the three-angle grid is a rank 23 Gaussian spherical quadrature grid [116]. Other examples may be found in Reference [4].

the Floquet method does not require that all of the matrices be generated at the same time; this results in a large easing of the memory requirements not possible in the grid-free case. It should be noted, however, that the use of 'tensor trains' can overcome the large memory requirements of the grid-free method, at the possible expense of a longer simulation time [4]. Secondly, the many separate simulations required when using the Floquet method to generate powder averaged spectra can be performed independently, allowing for efficient parallelization (at the cost of extra memory requirements); see Section 3.1.2. Path tracing (Section 2.2.1) can be used to parallelize the grid-free simulation, but such a parallelization is likely to be poorly load balanced and thus inefficient (Section 3.2).

As such, for the simulation of MAS, Floquet theory will usually be preferred. In principle, the grid-free method could be preferred for simulations requiring very large Wigner ranks, for which no accurate grids exist [116]. Unfortunately, the larger number of Wigner ranks that would be required would result in astronomically large matrices, likely making the simulation incomputable.

The generality of the formalism presented in this chapter, however, allows it to be applied to time dependent Liouvillians for which Floquet theory cannot be used. This is the subject of future work. Further, the new perspective that the formalism can cast on the dynamics of systems under time dependent Liouvillians may allow for novel analyses of problems, as alluded to in Section 4.4.5.

4.4.5 Relation to basis set truncation

The large magnitude and number of couplings in the solid state [33] mean that states with large spin-order (Section 1.1.1) can be populated very rapidly [2]. In the absence of rapid relaxation of the system to equilibrium [2], basis set truncation (Section 2.1) can only be expected to be valid for very short times [81], or when the magnitudes of these couplings are effectively reduced by very fast MAS [82]. This would seem to imply that the computational requirements for solid state simulations scale exponentially with the number of spins in the system (Section 1.2), severely limiting the size

of spin system that can be simulated.

However, simulations by Butler, et al. show empirically that basis set truncation can be applicable to simulations of spin diffusion experiments including MAS and powder-averaging [80]. Figure 4.2 implies that basis set truncation can be similarly applied to powder-averaged MAS pulse–acquire experiments. These simulations extend outside 'short' times, there are no explicit relaxation processes, and the spinning is not necessarily fast enough to reduce the coupling magnitude to the extent described in Reference [82]. We shall show below that the combination of powder averaging and MAS could provide a justification for the observed validity of basis set truncation in these systems.

In order to investigate the validity of basis set truncation in the simulation presented in Figure 4.2, we can decompose the detected zeroth-Wigner rank element of the auxiliary state vector into the contributions from individual spin orders. The rationalization for only analysing the zeroth-Wigner rank is that only this block is detected in a powder-averaged simulation (see Section 4.4.3).

Figure 4.3(b) shows the decomposition of the auxiliary state vector with the four spin-order basis (full basis) up to the time propagated to in order to generate the spectra of Figure 4.2. The negligibility of contributions from the three- and four-spin-order states can be seen. This is to be contrasted with the single crystal MAS case, shown in Figure 4.3(a). All the multi-spin orders become rapidly populated, and are thus not negligible over the extent of the simulation. In this case contributions from all Wigner ranks of the auxiliary state vector are included, weighted by normalized Wigner D-functions using the mapping operator of Equation (4.101). As the powder average spectrum is constructed as a combination of such single crystal spectra, it is not immediately clear why the population of the multi-spin orders would be slower in the powder-averaged case.

In order to try and rationalize the slower rate of population, Figure 4.4 presents decompositions of the state vectors required to simulate spectra in two distinct states: the isotropic liquid state in (a), and the anisotropic solid state in (b).

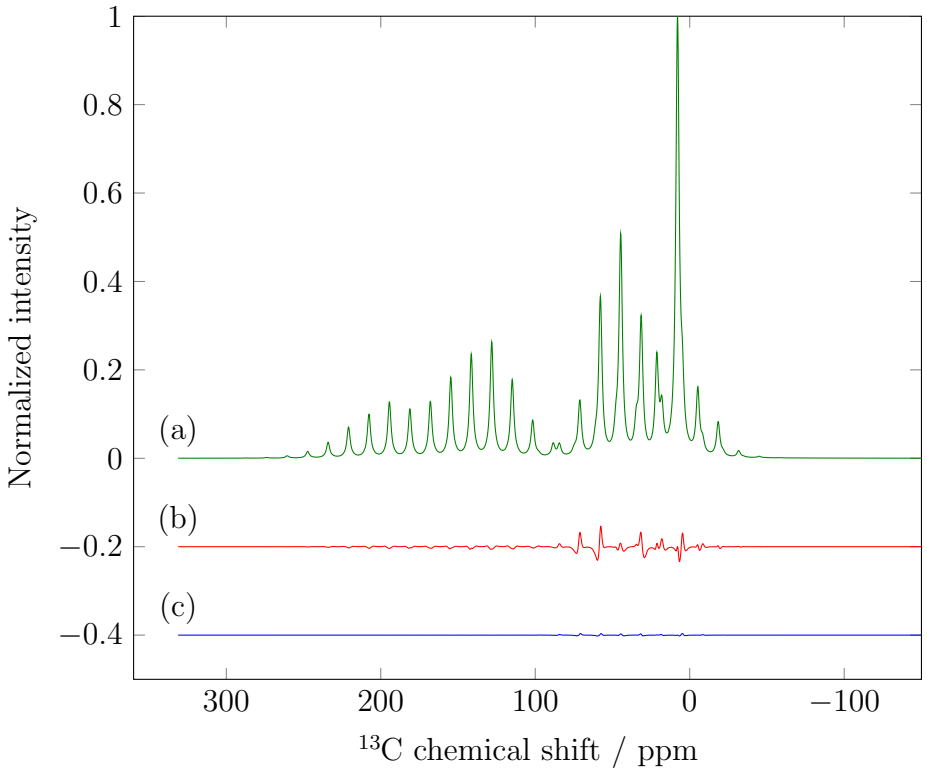


Figure 4.2: Demonstration of the validity of basis set truncation in the simulation of a solid state powder-averaged MAS spectrum. (a) Simulation of the ^{13}C proton-decoupled solid state MAS spectrum of alanine [150] using the full spin state space. (b) Plot of the difference between the spectrum generated using the full spin state space and the spectrum generated using only states with spin-order ≤ 2. The maximal absolute difference is 3.4×10^{-2}. (c) Plot of the difference between the spectrum generated using the full basis and the spectrum generated using only states with spin-order ≤ 3. The maximal absolute difference is 2.0×10^{-3}. Plots (b) and (c) are offset from (a) by -0.2 and -0.4, respectively, in order to allow a clearer comparison. Spin system parameters are identical to those found in the caption of Table 4.1. The Wigner rank is 23; with this choice the errors due to truncation, determined by comparison to a simulation with higher Wigner rank, are less than 10^{-3} over the entire spectrum.

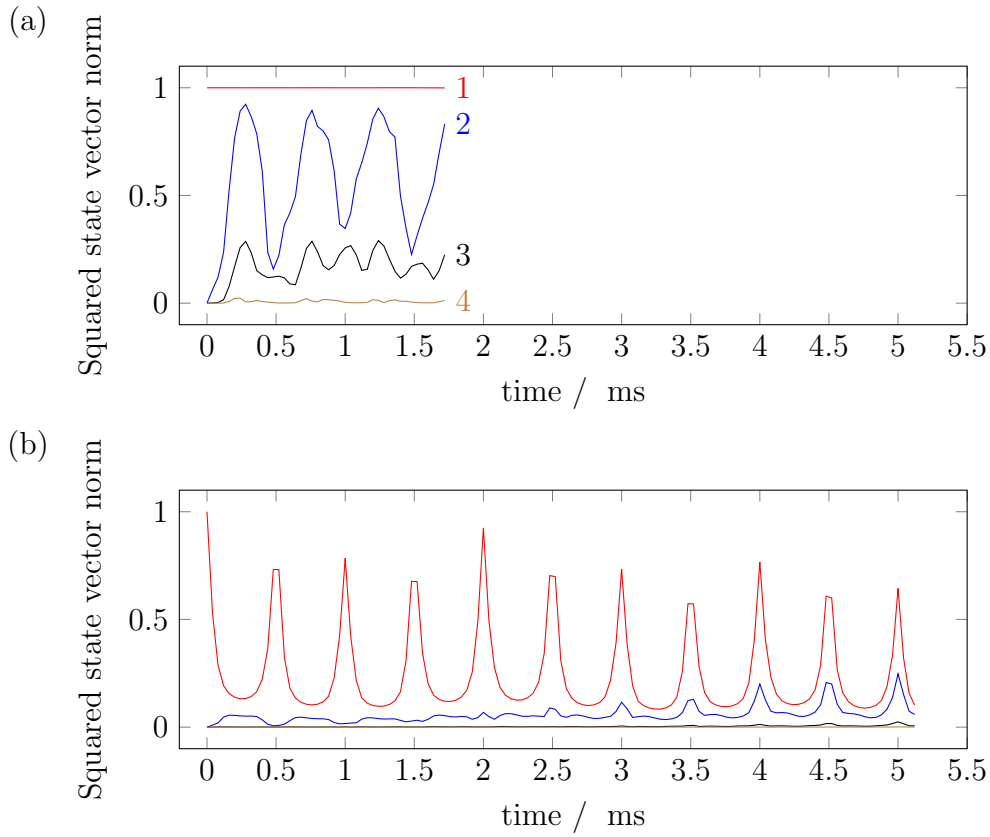


Figure 4.3: Spin order decomposition of the squared state vector norm of the state vectors that would be used to simulate 2 kHz MAS ^{13}C-NMR spectra of alanine [150] as (a) a single crystal and (b) a powder. Spin system parameters are as of Figure 4.2. Both state vectors have been generated using the elevation method described in Section 4.4. Spin orders are labelled in (a); the same colouring applies to both plots. Only the subspace with +1 coherence with respect to ^{13}C nuclei is shown as only this subspace is detected in Figure 4.2; Liouvillian secularity restricts detectable population to this subspace (Section 1.5.2). In plot (a), contributions from all Wigner ranks up to a maximum of 35 are included, weighted by (normalized) Wigner D-functions. Only the portion of the trajectory over which the total squared norm differs from unity by less than 10^{-3} is shown. Increasing the maximum Wigner rank increases the amount of time over which the squared norm is conserved, with concomitant increase in simulation time. The portion shown is long enough to demonstrate the much faster increase in high spin order population as compared to (b). Plot (b) does not have a constant squared norm because only the component of the state vector corresponding to the detected zeroth-Wigner rank is shown. This is the decomposition of the state vector used to generate Figure 4.2(a); the maximum Wigner rank is the same here as there. Low population of larger spin orders explains the low maximal error due to basis set truncation demonstrated in Figures 4.2(b) and (c).

Examination of Figure 4.4(a), the isotropic case, reveals that for the times propagated to in this case the three and four spin orders could be dropped on the basis of the early time approximation [81]. It should be noted that in this case no Wigner ranks other than the zeroth can become populated. The longer timescale of the dynamics in this case is due to the lower magnitude of the couplings relative to the anisotropic case.

We may argue by analogy to the isotropic liquid state case as to why the population of higher spin orders occurs much more slowly in Figure 4.2(b) than in Figure 4.2(a). As the anisotropic interactions are secular in this system, the effect of magic angle spinning of the powder is to average the anisotropic interactions [33]: the magnitudes of these interactions tend to their isotropic magnitudes as the spinning rate increases. Population of the higher multi-spin orders can thus be qualitatively argued to be slowed, extending the region of applicability of the initial rate approximation regime when simulating the system of Figure 4.2(a). Incomplete cancellation of the various terms occurs in the single crystal case [151], resulting in the faster population of multi-spin orders observed in the single crystal case, Figure 4.4(b).

The effect of powder averaging provides further qualitative justification for the validity of basis set truncation in this system. Examining the decomposition of the static powder state vector, Figure 4.3(b), we can see that the total square norm decreases with time, giving the appearance of relaxation towards zero population for all spin orders. This apparent relaxation derives from loss of population to higher Wigner-rank states which are not detected. The short timescale over which dynamics can actually be detected suggests that basis set truncation may be possible in general solid state spectra: the states of large spin-order will never contribute to spectra if they are damped before being populated to any significant extent.

It is important to note, however, that while larger spin-orders may not be populated, relatively large Wigner-ranks definitely *do* get populated in the simulation. The simulation of Figure 4.4(b) used a rank 131 Gaussian spherical quadrature grid [116], with smaller grids showing less damping. The decrease in matrix size due to basis

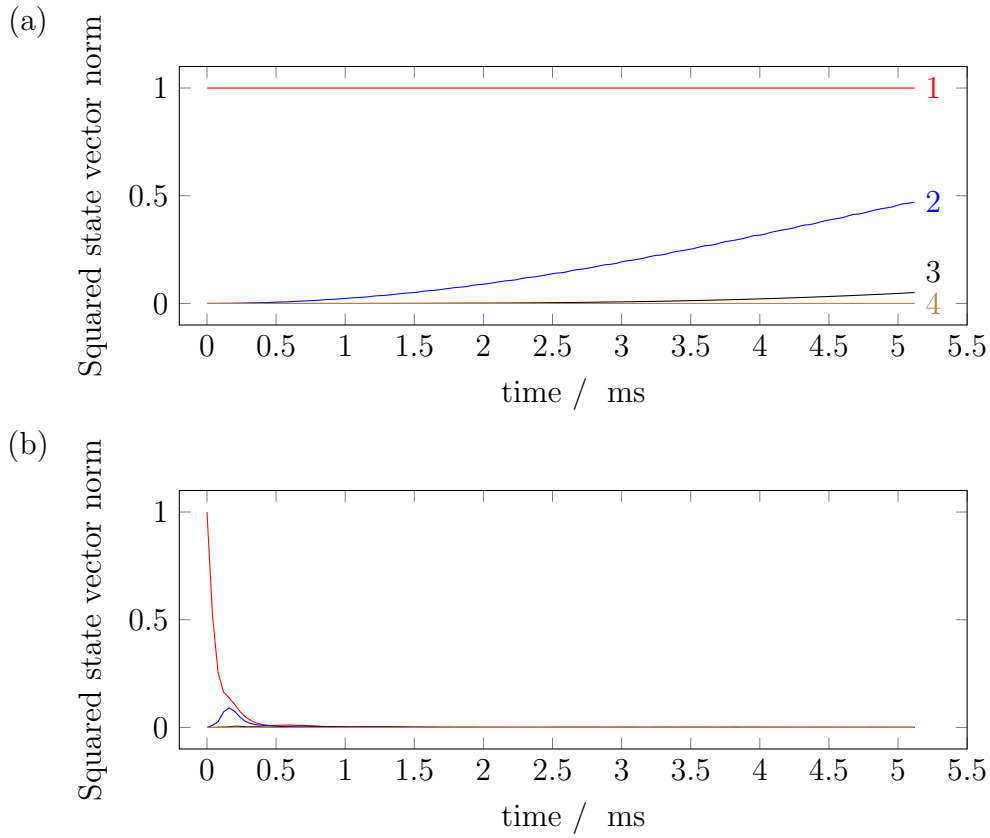


Figure 4.4: Spin order decomposition of the squared state vector norms of the state vectors that would be used to simulate the NMR spectrum of alanine [150] in the (a) isotropic liquid and (b) anisotropic static powder states. The spin orders are labelled in (a), and the same colour corresponds to the same spin order in each plot. Only the subspace with +1 coherence with respect to ^{13}C nuclei is shown as only this subspace would be detected; Liouvillian secularity restricts detectable population to this subspace (Section 1.5.2). As discussed in the text, the low population of the higher spin orders implies the validity of state space restriction for these two simulations. Further, the MAS powder averaged simulation in Figure 4.3(b) may be qualitatively argued to inherit the factors allowing for state space restriction in these systems, potentially explaining the observed validity of basis set truncation in that system. All spin system parameters are as in Figure 4.2. Only the component of the state vector corresponding to the population of the zeroth Wigner rank, computed using a rank 131 Gaussian spherical quadrature grid [116], is analysed in the solid case, as only that component is detected.

set truncation is thus overshadowed in the solid state by a much larger increase in effective matrix size, as compared to liquid state simulations, due to the need for large grids. This also implies that grid size, rather than spin system size, may turn out to be the limiting factor. It should be noted that the large Wigner rank required meant that the grid-free formalism could not be used in the static case; the matrices were too large.

The damping of the squared-norm in the MAS case, Figure 4.2(b) is much less pronounced than in Figure 4.3(b). This is likely because MAS partially averages the anisotropic interactions [33], reducing their magnitude and thus slowing the rate of population of higher Wigner ranks. This damping could prevent the population of high multi-spin orders at longer timescales, however.

Despite best efforts, a quantitative description of the qualitative effects described above remains elusive. We can, however, suggest some possible lines of investigation utilizing the formalism of Section 4.4.

Use of the analogy to the isotropic liquid state allows an upper bound on the population of multi-spin orders to be obtained by inserting the highest eigenvalue of the effective Liouvillian into the formulæ presented in Reference [2]. This will, in general, result in a very large overestimate of the number of spin-orders required. Calculation of the 'averaged' coupling parameters (taking into account the extent to which Wigner-ranks are populated) would be needed in order to obtain a more useful criterion. Construction of average Liouvillians using the formalism of this chapter could allow for a better estimate of these couplings, and thus a better upper bound on the number of spin orders required.

Analogy to the powder case could be made quantitative by application of projection operator methods [152] to the grid-free formalism. Unfortunately, the development of valid approximations that would allow computation of the integrals involved has turned out to be non-trivial.

Investigation of these possibilities is the subject of further work in the Kuprov group.

4.5 Summary

In this chapter we have developed a general 'elevation' method for computing the dynamics of a state vector under a time-dependent Liouvillian. It is assumed that the time-dependence arises from a set of underlying time-varying spin-independent coordinates; one of these coordinates could actually be time itself. While the matrices involved in this method are formally infinite in size, truncation of these matrices, where valid, can allow computation of the dynamics using the methods of Section 1.5.1. We have also shown how this method relates to previously known methods of elevation, and asserted how it could be used to help explain the validity of state space restriction in solid-state magic-angle spinning NMR simulations.

Chapter 5

Pulse sequences

While the spin dynamics simulation toolbox *Spinach* (Section 1.7) began its life as a proof-of-concept for state space restriction (SSR) [24], it has become much more. For examples of its growth, one need look no farther than the the previous chapters of this thesis; all of the ideas and concepts therein have been incorporated into the *Spinach* code base. This growth has been facilitated by the non-specificity of the underlying algorithms; they are designed where possible to apply generally to spin dynamics. Looking to the future, it may be hoped that this generality and growth will lead to *Spinach* becoming an essential part of every magnetic resonance spectroscopist's toolbox.

In order to achieve this goal, the many different experimental protocols of magnetic resonance must be implemented in *Spinach*; this has constituted a significant part of my research. Miscellaneous fruits of this labour are presented in this chapter. The unifying theme is utilization of 'quirks' of the experimental protocols and simulation methodology to improve efficiency. The novel 'tips and tricks' given here for taking advantage of these quirks will hopefully be of use to the future maintainers of *Spinach*.

5.1 ESEEM

Two-pulse Electron Spin Echo Envelope Modulation (ESEEM) is a simple Electron Spin Resonance (ESR) pulse sequence (Section 1.5) for detecting coherence between electrons and nuclei [28]. The pulse sequence is shown in Figure 5.1. Originally imple-

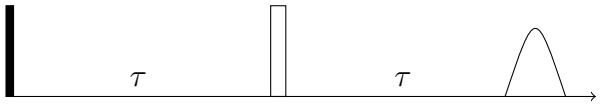


Figure 5.1: The two-pulse ESEEM pulse sequence [28]. Rectangles represent microwave-frequency pulses applied to the electrons: the thin filled rectangle is a $\pi/2$ pulse, the wider empty rectangle a π pulse (Section 1.5.1). These pulses have phase x (Section 1.1.2). The curve represents the spin echo [28] that is detected at time 2τ. The intensity of the echo peak is measured as a function of τ.

mented into *Spinach* by Ilya Kuprov, this pulse sequence is of interest to us here as it provides a simple test bed for destination state screening (DSS); see Section 2.3 for a detailed explanation of DSS. This application of DSS to ESEEM has been published previously [1].

The pulse sequence of Figure 5.1 is expressed mathematically as

$$\left\langle \hat{s}^{\dagger} \right\rangle_{\tau} = \langle \hat{s}| \exp\{-\mathrm{i}\mathcal{L}\tau\} \exp\{-\mathrm{i}\mathcal{I}_x\pi\} \exp\{-\mathrm{i}\mathcal{L}\tau\} \exp\{-\mathrm{i}\mathcal{I}_x\pi/2\}|\hat{\rho}_0\rangle \tag{5.1}$$

where $\exp\{-\mathrm{i}\mathcal{I}_x\theta\}$ represents an x pulse with flip-angle θ acting on the electrons (see Section 1.5.1), $\langle\hat{s}|$ is the detection state, and detection is performed at the echo maximum. It is possible to apply DSS to the latter τ-precession period in the manner described in Section 2.3. We shall now show that in simulations of ESEEM experiments we can also use DSS in the first τ-precession period.

The state

$$\langle \hat{s}| \exp\{-\mathrm{i}\mathcal{I}_x\pi\} \tag{5.2}$$

represents a linear combination of the states that will be mapped into the detection state by the $\pi/2$ pulse. This state can thus be used as the destination state for the first precession period. It is possible that using DSS in this way during the first τ-precession period can result in a greater reduction than when using the the 'source state' $\exp\{-\mathrm{i}\mathcal{I}_x\pi/2\}|\hat{\rho}_0\rangle$ for trajectory-level SSR; we provide an example below.

The detected observable is usually that represented by the detection state $\langle\hat{I}_+|$. This allows for collection of the complex data required for Fourier transformation of the signal to obtain an ESEEM spectrum [28]. Taking the inner product of the

destination state $\langle \hat{I}_+| \exp\{-\mathrm{i}\mathcal{I}_x\pi\}$ and an arbitrary state $|\hat{\sigma}\rangle$ gives $\langle \hat{I}_+| \exp\{-\mathrm{i}\mathcal{I}_x\pi\}|\hat{\sigma}\rangle$. Noting the definition of Equation (1.105), we write this inner product as

$$\left\langle \hat{I}_+ \middle| \mathrm{e}^{-\mathrm{i}\mathcal{I}_x\pi} \middle| \hat{\sigma} \right\rangle = \left\langle \hat{I}_+ \middle| \mathrm{e}^{-\mathrm{i}\hat{I}_x\pi} \hat{\sigma} \mathrm{e}^{+\mathrm{i}\hat{I}_x\pi} \right\rangle. \tag{5.3}$$

Expression of the inner product in terms of the trace (Equation (1.36)), and the recollection that $\hat{I}_+^\dagger = \hat{I}_-$ (Section 1.1.1) permit us to write this as

$$\left\langle \hat{I}_+ \middle| \mathrm{e}^{-\mathrm{i}\mathcal{I}_x\pi} \middle| \hat{\sigma} \right\rangle = \mathrm{tr}\left\{ \hat{I}_+^\dagger \mathrm{e}^{-\mathrm{i}\hat{I}_x\pi} \hat{\sigma} \mathrm{e}^{+\mathrm{i}\hat{I}_x\pi} \right\} \tag{5.4}$$

$$= \mathrm{tr}\left\{ \hat{I}_- \mathrm{e}^{-\mathrm{i}\hat{I}_x\pi} \hat{\sigma} \mathrm{e}^{+\mathrm{i}\hat{I}_x\pi} \right\}. \tag{5.5}$$

Cyclic permutation under the trace [19] results in

$$\left\langle \hat{I}_+ \middle| \mathrm{e}^{-\mathrm{i}\mathcal{I}_x\pi} \middle| \hat{\sigma} \right\rangle = \mathrm{tr}\left\{ \left(\mathrm{e}^{+\mathrm{i}\hat{I}_x\pi} \hat{I}_- \mathrm{e}^{-\mathrm{i}\hat{I}_x\pi} \right) \hat{\sigma} \right\}. \tag{5.6}$$

The operator in parentheses can be shown via product operator methods to be given by [8, Equation 2.1.109]

$$\mathrm{e}^{+\mathrm{i}\hat{I}_x\pi} \hat{I}_- \mathrm{e}^{-\mathrm{i}\hat{I}_x\pi} = \hat{I}_+, \tag{5.7}$$

thus

$$\left\langle \hat{I}_+ \middle| \mathrm{e}^{-\mathrm{i}\mathcal{I}_x\pi} \middle| \hat{\sigma} \right\rangle = \mathrm{tr}\left\{ \hat{I}_+ \hat{\sigma} \right\} \tag{5.8}$$

$$= \left\langle \hat{I}_- \middle| \hat{\sigma} \right\rangle. \tag{5.9}$$

Since this is true for all $|\hat{\sigma}\rangle$, our required destination state for the first τ-precession period is

$$\left\langle \hat{I}_+ \right| \exp\{-\mathrm{i}\mathcal{I}_x\pi\} = \left\langle \hat{I}_- \right|. \tag{5.10}$$

Examples of the extra reduction in state space size obtained by utilizing destination state screening in ESEEM are presented in Table 5.1. With DSS, the size of the state space required in order to propagate the state vector through the two τ-precession periods is seen to be a quarter of the size of the full state space. This can be rationalized by noting that the high-field Liouvillian is secular with respect to the electron (Section 1.1.2). It thus cannot mix states of different coherence order with

Radical	State vector length		
	Full state space	Source state screening	Destination state screening
(^{15}N-TEMPOL)$^{\bullet}$	16	16	4
(Methyl)$^{\bullet}$	256	147	64
(Phenyl)$^{\bullet}$	4096	2078	1024

Table 5.1: State vector size for the Liouville-space simulation of high-field single crystal ESEEM spectra of several small radicals. The data presented here differ from those found in Reference [1] because trajectory-level SSR was performed using an implementation improved by Ilya Kuprov in the time since publication. For simplicity, no basis level SSR was used. Quoted numbers apply to both τ-precession periods. Crystal orientation is $[\pi/3, \pi/4, \pi/5]$. The Hamiltonian coupling parameters (Section 1.1.2) were obtained from vacuum DFT calculations performed by Ilya Kuprov using Gaussian 03 [101]. The chemical structure of TEMPOL may be found in Reference [153].

respect to the electron; the +1 coherence with respect to the electron corresponds to one quarter of the state space, as does the −1 coherence. The result of an ideal $\pi/2$ pulse applied to a spin-1/2 particle like the electron will be population of both +1 and −1 coherence orders [28]; source state screening will thus result in a space at least double the size of that found by DSS. Table 5.1 reveals that source state screening does not even result in this level of reduction.

5.2 CLIP-HSQC

Heteronuclear Single Quantum Coherence (HSQC) is a two-dimensional NMR experiment that can be used to measure spin–spin couplings accurately [10, 12]. Spin–spin couplings determined using HSQC are often used to help characterize molecules [10]. Spin–spin couplings can also contain structural information; analysis of such information aids in the deduction of three-dimensional molecular structures [11]. There is thus interest in using HSQC to accurately measure such structural information-containing spin–spin couplings.

When performed on a liquid sample, HSQC experiments can give information on only the isotropic J-couplings (Section 1.1.2), due to motional averaging of dipolar

interactions [8]. Some structural information can be obtained from J-couplings [154], but dipolar couplings are much richer in the structural information that they contain [8]. In the liquid state, dipolar couplings give rise to a mechanism of relaxation (Section 1.1.2), the effects of which can be observed through appropriate experiments, but quantification of these effects requires the use of approximations that will not be valid in general [11]. Measurement in the solid state retains dipolar information as splittings of the spectral peaks, but leads to multiple overlapping broad peaks that are difficult to analyse [83].

The 'ideal' situation would thus be to measure spectra in a state somewhere in between 'solid' and 'liquid': the dipolar couplings would lead to observable splittings, but not result in broad difficult-to-interpret spectra. To this end, measurements in liquid crystals [155] and gels [156] are performed; these solvents partially restrict motion, meaning that the sample is no longer entirely isotropic and thus partially-averaged dipolar couplings may be observed. Such a partially-averaged dipolar coupling is known as a 'residual dipolar coupling' (RDC); collectively, these couplings can be used to determine the structures of molecules [157].

In simple cases the extent of this restricted motion can be described by a single parameter, an 'order parameter' [155]. This parameter ranges from zero (completely isotropic molecular tumbling) to unity (completely aligned molecules). It is this order parameter that is used to define the extent of anisotropy in the simulations presented in this section, but it should be noted that, due to the efforts of Ilya Kuprov, *Spinach* can also incorporate anisotropies described by the more general Saupe matrix [155].

The ability to extract accurate couplings using HSQC may lead to the hope that measurement of HSQC spectra in liquid crystals or gels would allow for accurate determination of these RDCs. Unfortunately, a number of problems arise when attempting to measure RDCs using the HSQC pulse sequence, as discussed in Reference [158]. The most significant is that in order to obtain high resolution spectra, the HSQC experiment requires the time interval Δ in the pulse sequence (Figure 5.2) to be matched to the ^{13}C–^{1}H coupling strength [10]. RDCs broaden the range of couplings; this means

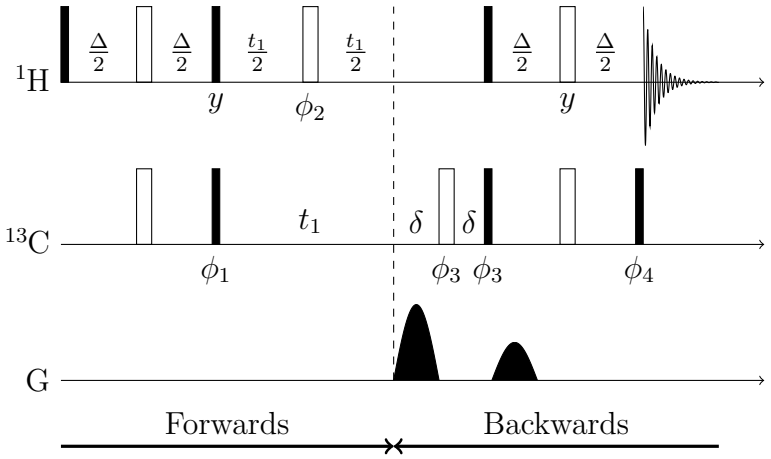


Figure 5.2: The CLIP-HSQC pulse sequence, adapted from Reference [158]. The labels ^{1}H and ^{13}C at the beginning of the lines denote which nuclei the radiofrequency pulses, represented by rectangles, are to be applied to. Thin filled rectangles are $\pi/2$ pulses, wider empty rectangles are π pulses. Labels under pulses denote the phase (Section 1.1.2); labels ϕ_k relate to the phase cycles shown in Table 5.2. The decaying sine curve at the end of the pulse sequence represents the detected signal. The indirect detection period is labelled with the period length, t_1. Unlabelled pulses have phase x. Pulsed field gradients [12] are shown on the line marked G (see Section 5.2.1); for experimental gradient parameters, see Reference [158, Figure 2(c)]. Part of the simulation is run backwards in order to increase efficiency (see Section 5.2.4); arrows below the pulse sequence show the separation into forward and backward parts.

that matching Δ to 'the' coupling strength in the presence of non-negligible RDCs is infeasible. The effect of this on lineshapes can be seen in simulations presented in Figure 5.3. Accurate couplings can still be determined from such spectra if one manually phases each set of multiplets, but this is tedious, and automations are prone to error [158].

To allow for the efficient determination of residual dipolar couplings, Andreas Enthart, et al. suggested a modified HSQC pulse sequence, Clean In-Phase-HSQC (CLIP-HSQC), that aims to compensate for the infeasibility of matching Δ to the coupling strength [158]. This CLIP-HSQC pulse sequence is shown in Figure 5.2 and has been implemented into *Spinach* as part of a collaboration with the group of Burkhard Luy. Such simulations could be used to determine accurate residual dipolar couplings, and thus accurate molecular structures.

Several important *Spinach* implementation points are described below.

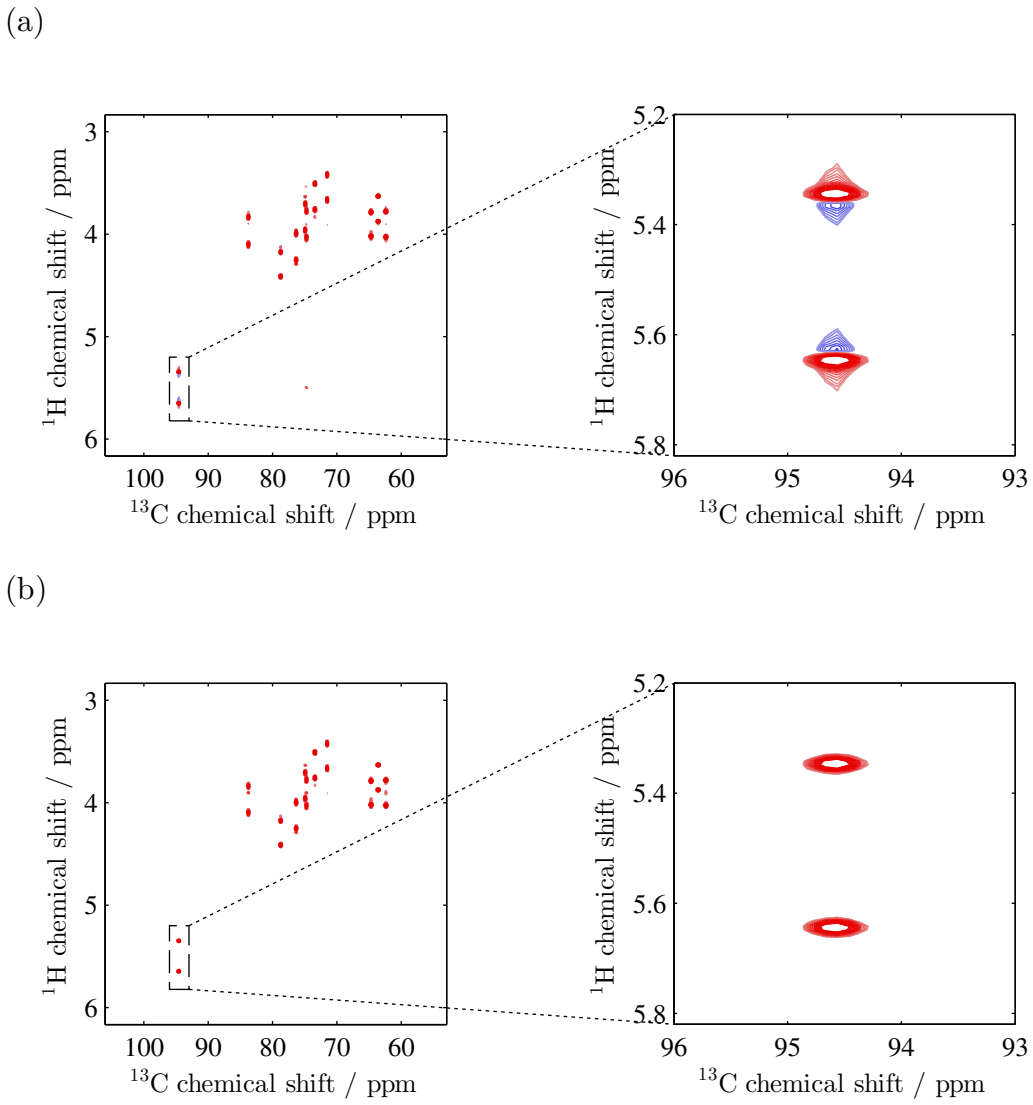


Figure 5.3: Comparison of (a) HSQC [10, 12] and (b) CLIP-HSQC [158] simulated spectra of sucrose. The zoomed-in portions allow for comparison of lineshapes observed using the two pulse sequences. Positive signals are red, negative signals are blue. The simulation parameters are the same in both cases: magnetic field of 14.1 T; order parameter [155] 10^{-3}; $\Delta = 1/(2J)$, $J = 140$ Hz; nuclear positions and isotropic couplings taken from a vacuum DFT simulation performed by Ilya Kuprov using Gaussian 03 [101]; isotropic chemical shifts taken from Reference [102]; restricted basis IK–2(3) (Section 2.1). The lineshape in the zoomed-in portion of the HSQC spectrum is not well phased; in the CLIP-HSQC spectrum a pure absorption signal may be observed, allowing for accurate determination of RDCs [158]. The method detailed in Section 5.2.1 has been used in both simulations.

Step	ϕ_1	ϕ_2	ϕ_3	ϕ_4	Receiver
1	$+x$	$+x$	$+x$	$+x$	$-$
2	$-x$	$+x$	$+x$	$-x$	$+$
3	$+x$	$+x$	$-x$	$+x$	$+$
4	$-x$	$+x$	$-x$	$-x$	$-$
5	$+x$	$-x$	$+x$	$+x$	$-$
6	$-x$	$-x$	$+x$	$-x$	$+$
7	$+x$	$-x$	$-x$	$+x$	$+$
8	$-x$	$-x$	$-x$	$-x$	$-$

Table 5.2: Phase cycling of the CLIP-HSQC pulse sequence from Reference [158]. The ϕ_k elements reflect the phase of the radiofrequency pulses (Section 1.1.2) in the pulse sequence shown in Figure 5.2, and the Receiver elements reflect whether the signal resulting from the step is to be added or subtracted from the total. Implementation of these phase cycles in *Spinach* is detailed in Section 5.2.2.

5.2.1 Gradient implementation

The CLIP-HSQC pulse sequence, as shown in Figure 5.2, contains pulsed field gradients (PFGs) [12]. The combination of these gradients and the pulses ensure that all state population except that which has undergone a specific transfer between coherence orders (Section 1.1.2) is dephased such that it becomes unobservable [12]. Specifically, the only state population that is not dephased is in states with $+1$ coherence on the indirectly detected spins (and zero coherence order on all other spins) at the end of the first detection period, and in states with -1 coherence on the directly detected spins (and zero coherence order on all other spins) just after the last pulse marked ϕ_3 [158]. While efficient explicit simulation of PFGs is possible (see e.g. References [159, 160, 161, 162]), we may take advantage of the knowledge of which coherences are to be selected to speed up the computation. Selecting these coherences at the appropriate times, which may be carried out efficiently in *Spinach* because it uses a irreducible spherical tensor operator basis set (Section 1.1.2), emulates the experimental coherence selection. It should be noted that as the gradients are not being simulated explicitly in the simulations presented here, the delay δ (see Figure 5.2) is set to zero.

The gradients also have another purpose. In high-resolution NMR experiments

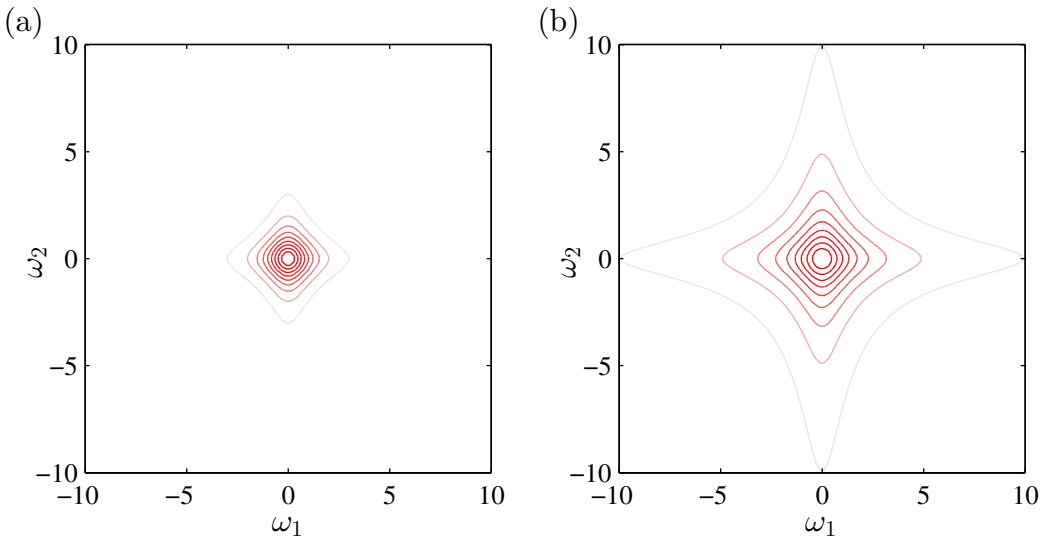


Figure 5.4: Contour plots of lineshapes discussed in the text. (a) Contour plot of $A_1(\omega_1)A_2(\omega_2)$ showing a Lorentzian lineshape in both dimensions. (b) Contour plot of $|[A_1(\omega_1)+\mathrm{i}D_1(\omega_1)][A_2(\omega_2)+\mathrm{i}D_2(\omega_2)]|$. The parameters are $\Omega_1 = \Omega_2 = 0$, $R_1 = R_2 = 1$. It may be seen that the lineshape in (a) is much narrower.

there are two important requirements for the spectra [12]:

1. Each signal should correspond to one frequency, i.e. the signal should not be reflected in any of the axes.

2. Each signal should be phase discriminated, i.e. have a Lorentzian lineshape in all dimensions (Figure 5.4(a)).

The gradients in the CLIP-HSQC pulse sequence can be used to obtain spectra that achieve both of these requirements.

The first point is achieved in the directly detected dimension by taking the detection state to be $\sum_k \langle \hat{s}_+^{(k)}|$, where k runs over the directly detected spins [12]. In the indirectly detected dimension this can be achieved by selecting states of only either $+1$ (or -1) coherence with respect to the indirectly detected spins during the indirect detection period [12]. This selection is achieved by the pulse sequence element containing the pair of gradients, as discussed above.

The detection scheme that achieves the first spectral requirement is not sufficient for the second. We can see this by examining a typical 2D NMR signal. Following the

notation of Keeler [12], such a signal can be written in the form

$$S_{\pm}(t_1, t_2) = \begin{cases} S_0 \mathrm{e}^{(\pm \mathrm{i}\Omega_1 - R_1)t_1} \mathrm{e}^{(+\mathrm{i}\Omega_2 - R_2)t_2} & \text{if } t_1, t_2 \geq 0, \\ 0 & \text{if } t_1, t_2 < 0. \end{cases} \tag{5.11}$$

This corresponds to calculation in the diagonal frame of the Liouvillian, with Ω_k an effective angular frequency, R_k an effective relaxation rate, and t_k the time at which detection is made within the kth detection period. Prior to each of the detection periods (i.e. when either t_1 or t_2 is less than zero), the signal must be zero. The full spectrum consists of a linear combination of such signals. The signal at $t_1 = t_2 = 0$, S_0, is assumed to be real, following the analysis of CLIP-HSQC in Reference [158]. The signal $S_+(t_1, t_2)$ corresponds to detection of the $+1$ coherence with respect to the indirectly detected spins during the indirect detection period, while $S_-(t_1, t_2)$ corresponds to detection of the -1 coherence during this period [12].

Denoting the Fourier transformation [56] which maps functions of the variable t_k into functions of the conjugate variable ω_k by $\xrightarrow{\mathrm{FT}_{t_k}}$, we may express the result of Fourier transforming this signal sequentially with respect to t_2 and t_1 as [12]

$$S_{\pm}(t_1, t_2) \xrightarrow{\mathrm{FT}_{t_2}} \xrightarrow{\mathrm{FT}_{t_1}} S_0[A_1(\omega_1) \pm \mathrm{i}D_1(\omega_1)][A_2(\omega_2) + \mathrm{i}D_2(\omega_2)], \tag{5.12}$$

where

$$A_k(\omega) = \frac{R_k}{{R_k}^2 + (\omega - \Omega_k)^2}, \quad D_k(\omega) = \frac{\Omega_k - \omega}{{R_k}^2 + (\omega - \Omega_k)^2}. \tag{5.13}$$

It should be noted that $A_k(\omega)$ describes a Lorentzian lineshape in one dimension [12].

Spin–spin couplings give rise to splittings in each of the dimensions of the spectrum; an example can be seen in the zoomed in portion of Figure 5.3(b). In order to extract the coupling information from spectra, then, the lineshapes need to be well resolved from one another. Figure 5.4(a) shows the spectral shape that would be obtained if we had a double-Lorentzian lineshape, $A_1(\omega_1)A_2(\omega_2)$. Figure 5.4(b), in contrast, shows the absolute magnitude of the signal generated using the above procedure. It should be noted that we must use the absolute magnitude of this signal as both the real and imaginary parts alone will have both positive and negative components (as may be seen by inserting Equation (5.13) into the right hand side

of Equation (5.12)), rendering them inappropriate for the purpose of determining couplings. The narrow lineshapes arising from $A_1(\omega_1)A_2(\omega_2)$ will allow much more accurate determination of couplings, as the splittings will be much better resolved. For this reason, the more convoluted procedure described below, which results in extraction of $A_1(\omega_1)A_2(\omega_2)$, is used to generate the final spectrum. This procedure is well known in high-resolution NMR spectroscopy [12], but we show it explicitly here for completeness.

This procedure requires that we repeat the experiment so as to obtain both $S_+(t_1,t_2)$ and $S_-(t_1,t_2)$ [12]. Fourier transformation of these signals in the directly detected dimension results in

$$S_\pm(t_1,t_2) \xrightarrow{\mathrm{FT}_{t_2}} T_\pm(t_1,\omega_2) = \begin{cases} S_0 \mathrm{e}^{(\pm \mathrm{i}\Omega_1 - R_1)t_1}[A_2(\omega_2) + \mathrm{i}D_2(\omega_2)], & \text{if } t_1 \geq 0, \\ 0 & \text{if } t_1 < 0. \end{cases} \tag{5.14}$$

Combining the two transformed signals in the manner

$$U(t_1,\omega_2) = \frac{1}{2}[T_+(t_1,\omega_2) + (T_-(t_1,\omega_2))^*], \tag{5.15}$$

where $*$ denotes complex conjugation, results in

$$U(t_1,\omega_2) = \begin{cases} S_0 \mathrm{e}^{(+\mathrm{i}\Omega_1 - R_1)t_1} A_2(\omega_2), & \text{if } t_1 \geq 0, \\ 0 & \text{if } t_1 < 0, \end{cases} \tag{5.16}$$

i.e. we are left with just the $A_2(\omega_2)$ component in the directly detected dimension. Fourier transformation of $U(t_1,\omega_2)$ with respect to t_1 gives the result

$$U(t_1,\omega_2) \xrightarrow{\mathrm{FT}_{t_1}} S_0[A_1(\omega_1) + \mathrm{i}D_1(\omega_1)]A_2(\omega_2). \tag{5.17}$$

The real component of this signal consists of $S_0 A_1(\omega_1)A_2(\omega_2)$, allowing us to extract just that component.

The above procedure is that used, both experimentally [158] and in the simulation, to obtain phase discriminated lineshapes where each signal corresponds to only one frequency.

5.2.2 Phase cycling

The CLIP-HSQC pulse sequence requires phase cycling to be performed. Phase cycling consists of repeating an experiment with different phases for specific pulses, followed

by linear combination of the results. Such a procedure is usually performed in order to eliminate unwanted coherence order states [12]. The phase cycles involved in CLIP-HSQC are given in Table 5.2. Reference [158] should be consulted for the justification of these phase cycles.

Simulation of the CLIP-HSQC pulse sequence must necessarily represent these phase cycles as faithfully as possible. Simplifications are, however, made possible by taking full advantage both of the approximations inherent in the simulation, and of the flexibility that simulation can offer over experiment. The simplifications that have been made use of in the implementation of CLIP-HSQC are described below.

Ignore phase cycles of π-pulses

Within the phase cycle of CLIP-HSQC, ϕ_2 is an independent cycle of a π-pulse using the phases $\pm x$. For spin-1/2 particles the effect of a perfect π-pulse about $+x$ will have the same effect as a perfect π-pulse about $-x$. As we make the assumption of perfect pulses, we ignore this portion of the phase cycle. This halves the number of phase cycles that have to be performed.

Avoid redundant computation of non-unique portions of cycles

When performing a simulation there is flexibility as to when the individual cycles are begun. It is usually not the case that a full simulation has to be run for every cycle.

In the specific case of CLIP-HSQC (where mentions of 'pulse labels' relate to those given in Figure 5.2):

- Up until the pulse labelled ϕ_1 all the phase cycles are identical; this portion of the simulation thus only need be run once.

- Ignoring the phase cycle on ϕ_2, as discussed above, simulation of the portion of the simulation between the pulse labelled ϕ_1 and the first pulse labelled ϕ_3 must be carried out twice: once for each value of ϕ_1.

It is only after application of the first pulse labelled ϕ_3 that four unique pathways

exist. In the following we use the notation $|\hat{\rho}_{\phi_1,\phi_3}\rangle$, where $\phi_1, \phi_3 = \pm x$, to discriminate between state vectors generated from these four pathways.

It should be noted that the concept discussed here also applies to the emulation of gradient selection described in Section 5.2.1. It is not necessary to repeat the calculations before the coherence selection step for each individual coherence.

Combine phase cycles at the earliest possible point

There is also flexibility as to when the individual phase cycles are recombined. Unlike when running the experiment, this does not have to be after the whole simulation has been run.

In the case of CLIP-HSQC it is clear that the four individual cycles can be combined before the direct detection period, after the pulse labelled ϕ_4 in Figure 5.2. This will result in a significant reduction in the computational requirements of the simulation.

A greater reduction can be achieved, however. Just after the second pulse labelled ϕ_3 there are four unique states, but the phase cycle ϕ_4 is identical to that of ϕ_1 (see Table 5.2). We can thus immediately define two linear combinations of states

$$|\hat{\rho}_{+x}\rangle = -|\hat{\rho}_{+x,+x}\rangle + |\hat{\rho}_{+x,-x}\rangle, \tag{5.18}$$

$$|\hat{\rho}_{-x}\rangle = +|\hat{\rho}_{-x,+x}\rangle - |\hat{\rho}_{-x,-x}\rangle, \tag{5.19}$$

and propagate them instead of the four separate states, halving the number of cycles that have to be run of this portion of the simulation. At the pulse labelled ϕ_4 a pulse of phase $+x$ is applied to $|\hat{\rho}_{+x}\rangle$, and a pulse of phase $-x$ is applied to $|\hat{\rho}_{-x}\rangle$. Summation of the results of the two cycles is possible after this pulse; the single linear combination may then be propagated through the directly detected portion of the pulse sequence.

5.2.3 Dilute nuclei

HSQC is normally performed with ^{13}C and ^{1}H as the heteronuclei. As discussed in Section 3.1.1, the low natural abundance of ^{13}C means that simulations of the

spectra of molecules with a 'small' number of carbon atoms can commonly be split into simulations each with only a single ^{13}C nucleus.

The spectrum simulated in Figure 5.3 is of sucrose, with ^{13}C detected indirectly, and ^{1}H detected directly. Sucrose contains 12 carbon atoms. Using Equation (3.2) and the natural abundance of ^{13}C, 1.07 % [32], the probability that two ^{13}C nuclei are present in any given molecule is found to be 6.8×10^{-3}. In contrast, the probability that any given molecule contains a single ^{13}C nucleus is found to be 0.11. We thus see that this molecule is small enough that, for natural abundance spectra, omitting the contribution of molecules with more than a single ^{13}C nucleus to the spectrum is a reasonable approximation. As previously discussed in Section 3.1.1, this allows for efficient parallelization of the simulation, as used to accelerate both simulations shown in Figure 5.3.

5.2.4 State space restriction

As the residual dipolar couplings are of small magnitude, the use of basis-level SSR methodologies developed for liquid state simulations (Chapter 2) is still appropriate. This leads to large reductions in the size of the state space and makes these simulations possible.

Presented in Figure 5.3 is a simulation of the CLIP-HSQC spectrum of sucrose at 14.1 T with order parameter 10^{-3}. Ignoring ^{1}H nuclei that rapidly exchange with H_2O because they do not contribute to the experimental spectrum, each of the contributing simulations contains 15 spin-1/2 nuclei (14 ^{1}H and one ^{13}C). The full state vector for this system would contain 4^{15} elements. Using the IK–2(3) approximation (Section 2.1), this number of elements is reduced to less than 4×10^3; the exact number depends on which of the 12 ^{13}C nuclei is being included (Section 5.2.3).

Residual dipolar couplings, despite their weakness, reduce the effectiveness of trajectory-level SSR techniques relative to liquid state simulations. During the first few precession periods trajectory-level methods reduce the state space to about a quarter of the size of the IK–2(3) basis; application of these methods to later preces-

sion periods results in reductions to approximately one third of the size of the IK–2(3) basis.

During conversations with Martin Koos, the idea was formed to improve the situation in the final stages of the pulse sequence by propagating the detection state 'backwards' rather than the state vector 'forwards'. This evokes the methodology of DSS (Section 2.3), but instead of just propagating the detection state backwards for as long as is required for the trajectory-level SSR methods (Section 2.2.2), the detection state is propagated backwards for the entire precession period. The separation of the CLIP-HSQC pulse sequence into forward and backward calculation is shown by labelled arrows below the pulse sequence diagram, Figure 5.2.

The explicit splitting of the CLIP-HSQC pulse sequence into forward and backward calculations is required for implementation, and is given here. The expectation value of the observable operator $\hat{s}^\dagger$ will be a function of the delays t_1 and t_2 and may be schematically represented by

$$\left\langle \hat{s}^\dagger \right\rangle (t_1, t_2) = \langle \hat{s} | \mathrm{e}^{-\mathrm{i}\mathcal{L}t_2} \mathcal{P}_2 \mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2} \mathcal{M} \mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2} \mathcal{P}_1 | \hat{\rho}_0 \rangle. \tag{5.20}$$

The superoperators $\mathcal{P}_n$ represent the preparation pulses, delays, and coherence selection steps in the pulse sequence, and $\mathcal{M}$ is the π-pulse in the centre of the first detection period. Separation may be made into a partially-propagated state vector $|\hat{\rho}(t_1)\rangle$, and a partially-propagated detection state $\langle \hat{s}(t_2)|$:

$$|\hat{\rho}(t_1)\rangle = \mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2} \mathcal{M} \mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2} \mathcal{P}_1 | \hat{\rho}_0 \rangle \tag{5.21}$$

$$\langle \hat{s}(t_2)| = \langle \hat{s} | \mathrm{e}^{-\mathrm{i}\mathcal{L}t_2} \mathcal{P}_2. \tag{5.22}$$

The required expectation value is then computed by multiplying the two together:

$$\left\langle \hat{s}^\dagger \right\rangle (t_1, t_2) = \langle \hat{s}(t_2) | \, \hat{\rho}(t_1) \rangle. \tag{5.23}$$

The arguments used to explain the results of DSS in 2D NMR spectra (Section 2.3) also apply here. Application of trajectory-level SSR methods to the detection state at the beginning of each backwards propagated section reveal that only approximately

8 % of the IK–2(3) state space is required during the portion of the simulation run backwards. While not as significant as the orders of magnitude decrease in basis size afforded by the use of the IK–2(3) basis set relative to the full basis, this decrease is still very beneficial and ensures that simulations run in a reasonable amount of time.

5.3 HN(CO)C_α

NMR has become an essential technique in protein structure determination [163]. The large number of NMR-active nuclei present in proteins ($> 10^3$) make simple NMR spectra very crowded, and therefore hard to interpret. Multidimensional techniques are thus vital: spreading the spectra out into multiple dimensions allows for analysis of these complex spectra [11, 163].

A common 3D-NMR pulse sequence applied to proteins is HN(CO)C_α [103], shown in Figure 5.5. The rather strange looking name, 'HN(CO)C_α', derives from the coherence transfer pathway employed. This pathway is intended to span the peptide bond linking the amino acid residues of which proteins are comprised [150], drawn schematically in Figure 5.6. The spins which make up the pathway are coloured and have isotope labels in this schematic diagram, and the peptide bond is shown in bold. The ^{1}H is initially excited by a radiofrequency pulse; the excited population is then transferred from ^{1}H to ^{15}N to the carbonyl ^{13}C ('CO') to $^{13}C_\alpha$, giving rise to the name of this sequence, and then back again for direct detection of ^{1}H. Detection of just those spins combining to form the moiety of Figure 5.6 allows for mapping of the amino-acid backbone of the protein, aiding protein structure determination efforts.

In this section we detail the implementation of the HN(CO)C_α pulse sequence into *Spinach*. As an example simulation, the HN(CO)C_α spectrum of the protein ubiquitin [11] is presented in Section 5.3.6. Use of SSR is essential in this simulation: ubiquitin contains over 10^3 spins.

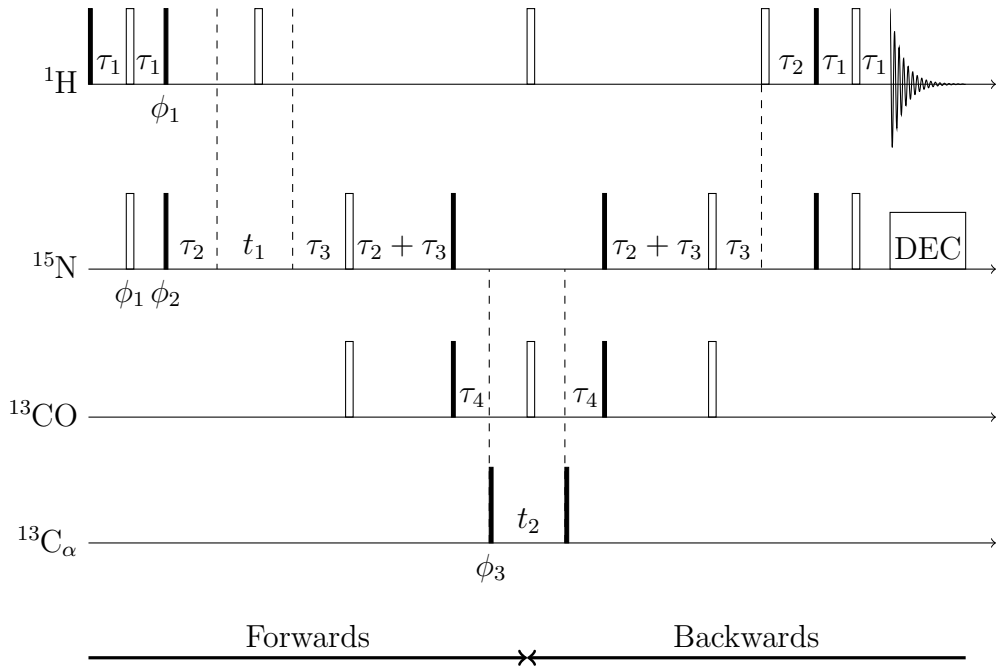


Figure 5.5: The HN(CO)C_α pulse sequence, adapted from Reference [103]. Symbols are as of Figure 5.2. As discussed in section 5.3.3, the phase cycles specified in that reference are omitted in favour of explicit coherence selection, hence $\phi_1 = y$, $\phi_2 = \phi_3 = x$. The block labelled 'DEC' denotes decoupling, performed analytically in the simulation (see Section 5.3.4). Arrows beneath the pulse sequence denote the portion of the simulation run backwards, leading to the increases in efficiency examined in Section 5.3.5. Indirect detection periods are labelled by t_n; ^{15}N states are detected during the first period, $^{13}C_\alpha$ states during the second [103].

5.3.1 Approximate equilibrium state

In the absence of any other information, the system must assumed to be in thermal equilibrium at the beginning of the HN(CO)C_α pulse sequence. For the large spin systems we wish to simulate, building the thermal equilibrium state vector in the manner described in Section 1.3.3 would be very computationally expensive, as the left Hamiltonian superoperator would have to be built and then used to propagate the unit state.

We avoid this computational expense by instead generating an approximate thermal equilibrium state. Firstly, we make use of the high temperature approximation (Section 1.1.3) so we do not have to compute a matrix exponential. Secondly, we note

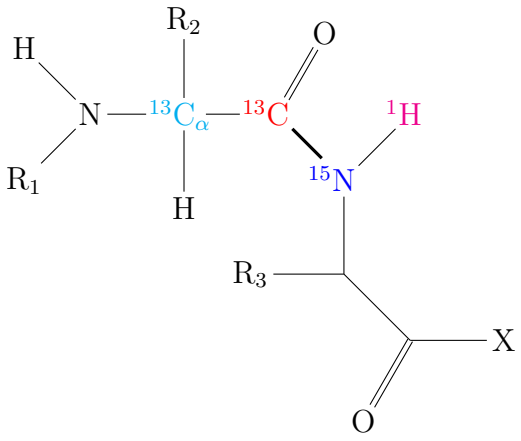


Figure 5.6: Schematic protein fragment showing the nuclei involved in the HN(CO)C_α coherence transfer path [103]. The nuclei involved in the transfer path are coloured and labelled by their mass number. During the HN(CO)C_α sequence the ^{1}H nuclei are excited, giving excited state population that is then transferred to the ^{15}N, then to the carbonyl ^{13}C, then finally to $^{13}C_\alpha$. This transfer is then reversed to allow for the final direct detection of the ^{1}H nuclei transverse state population.

that Zeeman terms will dominate at the high fields we are interested in (Section 1.1.2); we thus ignore couplings in the definition of this state. Thirdly, during the course of the pulse sequence, it is population of the ^{1}H states that is transferred through the course of the pulse sequence [103]; the initial populations of ^{13}C and ^{15}N states can thus be set to zero. Lastly, the differences in energy level splittings between the different ^{1}H nuclear states, due to differences in chemical shift, will be much smaller than the Larmor frequency at high field [8]. It is thus reasonable to approximate the populations of the ^{1}H nuclei to all be equal. We therefore arrive at the result that

$$|\hat{\rho}_{\text{eq}}\rangle \approx \sum_k \left|\hat{I}_z^{(k)}\right\rangle, \tag{5.24}$$

where the summation is over all ^{1}H nuclei.

5.3.2 Selective excitation of carbon moieties

An important requirement of the HN(CO)C_α sequence is that ^{13}CO and $^{13}C_\alpha$ are able to be excited independently. This is possible experimentally because the chemical shifts of the two moieties are very different [164]. In a simulation, exciting the two

sets of spins separately is simply a case of defining two separate operators that each act on one of the two separate sets of spins.

5.3.3 Phase cycling

Having to repeat individual steps of the simulation many times in order to faithfully represent phase cycles (Section 5.2.2) will extend the time required for already long and time-consuming simulations. The purpose of phase cycling in this experiment is to ensure that only populations of states of the desired coherence order are measured during the indirect detection periods [12]. As was the case for the simulation of CLIP-HSQC (see Section 5.2.1) that 'desired coherence' is +1 on the detected spin, such that each resonance corresponds to a single signal [12]. In order to attain this we analytically select just the +1 coherence on the spin being detected during each of the indirect detection periods, skipping the phase cycling steps entirely.

It is not possible to phase discriminate the signal (Section 5.2.1) in this experiment [103], so the absolute value spectra are plotted. Modifications of the pulse sequence that allow for phase discrimination do exist [165], however, we have opted to implement the original HN(CO)C_α pulse sequence for simplicity.

5.3.4 Decoupling during direct detection

The paper introducing HN(CO)C_α uses a WALTZ decoupling sequence [10] in order to remove explicit couplings of ^{15}N to ^{1}H during the direct detection period [103]. This is represented in Figure 5.5 by the block labelled 'DEC'. The intended result, a reduction in the number of splittings observed in the ^{1}H dimension, simplifies the spectrum considerably.

In order to emulate this efficiently using *Spinach*, we decouple the ^{1}H spins from the ^{15}N spins analytically in the Liouvillian used during the direct detection period. We call this decoupled-Liouvillian $\mathcal{L}_{\text{dec}}$ in the following.

The implementation of the decoupling, due to Ilya Kuprov, operates by zeroing the elements of $\mathcal{L}$ arising from couplings between ^{1}H and ^{15}N. This is equivalent to subtracting a 'coupling' part from $\mathcal{L}$.

5.3.5 Backward propagation

Below the HN(CO)C$_\alpha$ pulse sequence diagram shown in Figure 5.5 are arrows showing which portion of the pulse sequence is run backwards in the *Spinach* implementation. As was the case for the implementation of CLIP-HSQC (Section 5.2.4) this makes trajectory-based SSR more efficient. We describe decomposition into forward and backward calculations explicitly below.

The state vector of a spin system at the end of a 3D NMR experiment such as HN(CO)C$_\alpha$ is a function of the three delays t_1, t_2, and t_3 [166], and will here be written $|\hat{\rho}(t_1,t_2,t_3)\rangle$. The propagator taking the initial state $|\hat{\rho}_0\rangle$ to the final state vector may be abstractly expressed as

$$|\hat{\rho}(t_1,t_2,t_3)\rangle = \mathrm{e}^{-\mathrm{i}\mathcal{L}_{\mathrm{dec}}t_3}\mathcal{P}_3\mathrm{e}^{-\mathrm{i}\mathcal{L}t_2/2}\cdot \\ \cdot\,\mathcal{M}_2\mathrm{e}^{-\mathrm{i}\mathcal{L}t_2/2}\mathcal{P}_2\mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2}\mathcal{M}_1\mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2}\mathcal{P}_1|\hat{\rho}_0\rangle \tag{5.25}$$

where $\mathcal{L}_{\mathrm{dec}}$ is the analytically decoupled Liouvillian discussed in Section 5.3.4; $\mathcal{P}_n$ consists of preparation pulses, delays, and coherence selection steps; and $\mathcal{M}_n$ are pulses in the middle of the indirect detection periods. This propagator is general: it can represent any 3D NMR experiment consisting of perfect pulses, coherence selection steps, and delays. The expectation value of the observable $\hat{s}^\dagger$ is calculated from the state vector $|\hat{\rho}(t_1,t_2,t_3)\rangle$ using

$$\left\langle \hat{s}^\dagger \right\rangle (t_1,t_2,t_3) = \langle \hat{s}|\,\hat{\rho}(t_1,t_2,t_3)\rangle. \tag{5.26}$$

Inserting Equation (5.25) into Equation (5.26), we see that the expectation value may be rewritten

$$\left\langle \hat{s}^\dagger \right\rangle (t_1,t_2,t_3) = \langle \hat{s}(t_3,t_2/2)|\,\hat{\rho}(t_1,t_2/2)\rangle \tag{5.27}$$

where

$$|\hat{\rho}(t_1,t_2/2)\rangle = \mathrm{e}^{-\mathrm{i}\mathcal{L}t_2/2}\mathcal{P}_2\mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2}\mathcal{M}_1\mathrm{e}^{-\mathrm{i}\mathcal{L}t_1/2}\mathcal{P}_1|\hat{\rho}_0\rangle, \tag{5.28}$$

$$\langle \hat{s}(t_3,t_2/2)| = \langle \hat{s}|\mathrm{e}^{-\mathrm{i}\mathcal{L}t_3}\mathcal{P}_3\mathrm{e}^{-\mathrm{i}\mathcal{L}t_2/2}\mathcal{M}_2. \tag{5.29}$$

We have thus split the 3D-simulation into two 2D-simulations, with a 'dot product' to stitch them together.

Conceptually, a 3D-experiment is the contraction of two 2D-experiments; it is likely to be the rationale behind the design of any given 3D-experiment [166]. In general, however, one cannot take advantage of this in an experimental setting [167]; only simulations of these pulse sequences can take advantage of this splitting.

It should be noted that the number of multiplications and additions of the 'dot product' operation scales as the product of the number of steps taken for each detection period, as well as linearly with the size of the state vector. The apparent simplicity of this operation thus hides a lot of its computational complexity.

5.3.6 Protein simulation

Ultimately, the aim of implementing protein-NMR pulse sequences is to use them to simulate the spectra of proteins [5]. To this end, the spin system of ubiquitin [11], a protein with over 10^3 spins, was integrated into the *Spinach* codebase through the work of Zenawi Welderufael and Ilya Kuprov; for details see Reference [5]. The spin system is isotopically enriched: all the carbon nuclei are taken to be ^{13}C, and all the nitrogen nuclei are taken to be ^{15}N. A simulation of the HN(CO)C_α spectrum of this spin system is presented in Figure 5.7.

The ubiquitin spin system contains 573 ^{1}H, 374 ^{13}C, and 85 ^{15}N, giving a total of 1,032 spin-1/2 nuclei. The full state vector would thus have 4^{1032} elements; this is an infeasibly large vector to represent on a computer. We can use the power of basis set truncation (Section 2.1) to render the dynamics of this spin system computable. The coherence transfer pathway in HN(CO)C_α involves four spins; the minimal restricted basis required is thus IK–1(4, 1) [5, 24]. We utilize a Redfield relaxation superoperator in our simulation; this requires us to include product operator states corresponding to spins that are proximate in space in order to obtain accurate dipolar couplings. It has been found by Ilya Kuprov via comparison with experimental spectra that the minimal basis set required to obtain an accurate liquid state relaxation superoperator

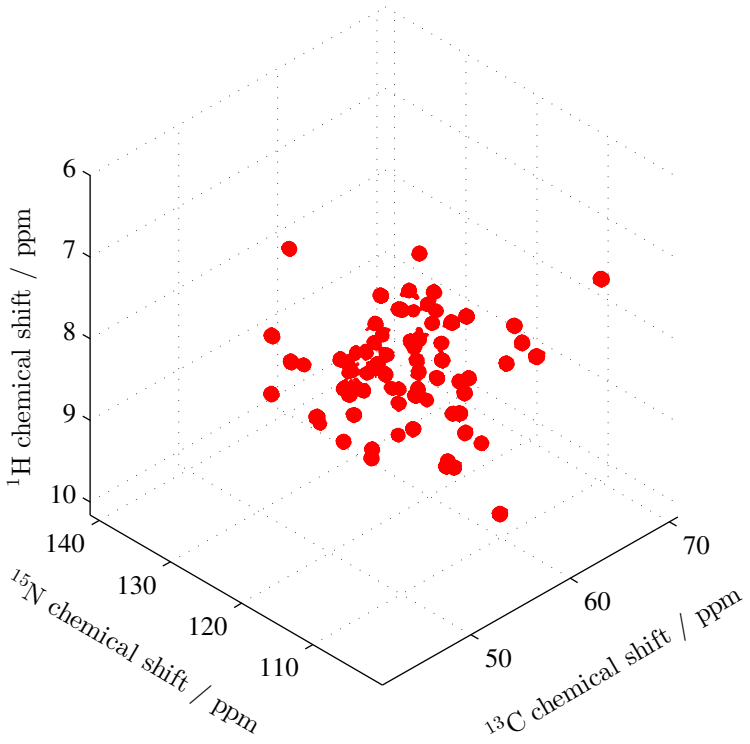


Figure 5.7: Simulation of the HN(CO)C_α [103] spectrum of the protein ubiquitin [11] using spin system parameters input into *Spinach* by Zenawi Welderufael and Ilya Kuprov [5]. The IK–1$(4, 3)$ restricted basis is used (Section 2.1), and the applied magnetic field strength is 14.1 T. Relaxation is included using a Bloch–Wangsness–Redfield superoperator (Section 1.3.2) with isotropic rotational correlation time 5 ns. The delays τ_n (see Figure 5.5) are chosen to be those suggested in Reference [103].

is the IK–1(4, 3) basis set [5]; this is the basis set that we use.

With the IK–1(4, 3) basis, the state vector has 2,080,828 elements. Trajectory level methods (Section 2.2) reduce this number significantly. At the beginning of the simulation, trajectory level methods reduce the state space to $\sim 10^5$, a small proportion of the basis level restricted vector. As the pulse sequence transfers state population along the coherence transfer path, significantly more of the IK–1(4, 3) state vector is required, as is to be expected. The increases are punctuated by massive decreases due to coherence selection steps (reducing the vector size to $\sim 5^4$). Utilization of backward propagation results in a similar pattern from the end of the simulation, where the detection period itself functions as a coherence selection step.

The state vectors after SSR are still large, so 'Krylov' propagation (Section 2.2.3) was utilized in order to reduce the amount of computer memory required for the simulation. Parallelization over the subspaces obtained by path tracing (Section 3.2) was used to accelerate the simulation. With all of the above restrictions, approximations, and optimizations, and 8 Intel Xeon processors, the simulation shown in Figure 5.7 took just under a week to run and required $\sim$ 160 GB of RAM.

The simulation of Figure 5.7 is unfortunately limited in an important way. Relaxation mechanisms arising from tumbling in solution are included using Bloch–Wangsness–Redfield theory (Section 1.3.2), but it is assumed that the whole molecule rotates as a solid body [83]. While this is often a good approximation for small molecules, proteins are large and consist of many sub-domains which could rotate with respect to one another. Any attempt to include relaxation accurately would have to take this into account, e.g. by using the Lipari–Szabo model [168].

As a further caveat, it should be noted that for a protein, ubiquitin is quite small [11]; proteins will usually have many more spins. However, this simulation, as well as others presented in Reference [5], marks a decisive beginning; now such simulations have been proved possible, they can only become faster and more efficient.

Bibliography

[1] M. Krzystyniak, Luke J. Edwards, and Ilya Kuprov.
"Destination state screening of active spaces in spin dynamics simulations".
In: *Journal of Magnetic Resonance* 210 (2011), pp. 228–232.
DOI: 10.1016/j.jmr.2011.03.010

[2] Alexander Karabanov, Ilya Kuprov, G. T. P. Charnock,
Anniek van der Drift, Luke J. Edwards, and Walter Köckenberger.
"On the accuracy of the state space restriction approximation for spin dynamics simulations".
In: *The Journal of Chemical Physics* 135.8 (2011), p. 084106.
DOI: 10.1063/1.3624564

[3] Luke J. Edwards and Ilya Kuprov.
"Parallel density matrix propagation in spin dynamics simulations".
In: *The Journal of Chemical Physics* 136.4 (2012), p. 044108.
DOI: 10.1063/1.3679656

[4] Luke J. Edwards, D. V. Savostyanov, A. A. Nevzorov, M. Concistrè,
G. Pileio, and Ilya Kuprov. "Grid-free powder averages: On the applications of the Fokker–Planck equation to solid state NMR".
In: *Journal of Magnetic Resonance* 235 (2013), pp. 121–129.
DOI: 10.1016/j.jmr.2013.07.011

[5] Luke J. Edwards, D. V. Savostyanov, Z. T. Welderufael, Donghan Lee, and Ilya Kuprov. "Quantum mechanical NMR simulation algorithm for protein-size spin systems".
In: *Journal of Magnetic Resonance* 243 (2014), pp. 107–113.
DOI: 10.1016/j.jmr.2014.04.002

[6] L. D. Landau and E. M. Lifshitz.
Quantum Mechanics: Non-Relativistic Theory. Third edition. Vol. 3.
Course of theoretical physics. Elsevier Science, 2003.

[7] A. Carrington and A. D. MacLachlan. *Introduction to Magnetic Resonance: With Applications to Chemistry and Chemical Physics*.
Chapman and Hall, 1979.

[8] R. R. Ernst, G. Bodenhausen, and A. Wokaun.
Principles of Nuclear Magnetic Resonance in One and Two Dimensions.
International Series of Monographs on Chemistry. Clarendon Press, 1990.

[9] M. A. Bernstein, K. F. King, and X. J. Zhou.
Handbook of MRI Pulse Sequences. Elsevier Science, 2004.

[10] T. D. W. Claridge. *High-Resolution NMR Techniques in Organic Chemistry*.
Tetrahedron Organic Chemistry. Elsevier Science, 2008.

[11] J. Cavanagh, W. J. Fairbrother, A. G. Palmer, III, N. J. Skelton, and
M. Rance. *Protein NMR Spectroscopy: Principles and Practice*.
Elsevier Science, 2010.

[12] J. Keeler. *Understanding NMR Spectroscopy*. Second edition. Wiley, 2013.

[13] Alex D. Bain and R. S. Dumont.
"Introduction to Floquet theory: The Calculation of Spinning Sideband
Intensities in Magic-Angle Spinning NMR".
In: *Concepts in Magnetic Resonance* 13 (2001), pp. 159–170.
DOI: 10.1002/cmr.1006

[14] P. de Fouquieres, S. G. Schirmer, S. J. Glaser, and Ilya Kuprov.
"Second order gradient ascent pulse engineering".
In: *Journal of Magnetic Resonance* 212 (2011), pp. 412–417.
DOI: 10.1016/j.jmr.2011.07.023

[15] Zdeněk Tošner, Thomas Vosegaard, Cindie Kehlet, Navin Khaneja,
Steffen J. Glaser, and Niels Chr. Nielsen. "Optimal control in NMR
spectroscopy: Numerical implementation in SIMPSON".
In: *Journal of Magnetic Resonance* 197 (2009), pp. 120–134.
DOI: 10.1016/j.jmr.2008.11.020

[16] Jasper D. Memory. *Quantum theory of magnetic resonance parameters*.
McGraw-Hill series in advanced chemistry. McGraw-Hill, 1968.

[17] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii.
*Quantum Theory of Angular Momentum: Irreducible Tensors, Spherical
Harmonics, Vector Coupling Coefficients, 3nj Symbols*.
World Scientific Publishing Company, Incorporated, 1989.

[18] N. Young. *An Introduction to Hilbert Space*.
Cambridge mathematical textbooks. Cambridge University Press, 1988.

[19] Richard Kaye and Robert Wilson. *Linear Algebra*.
Oxford University Press, UK, 1998.

[20] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Second edition.
Cambridge University Press, 2012.

[21] J. Jeener. "Superoperators in magnetic resonance". In:
ed. by John S. Waugh. Vol. 10. Advances in Magnetic Resonance.
Academic Press, 1982. DOI: 10.1016/B978-0-12-025510-8.50006-1

[22] A. R. Edmonds. *Angular Momentum in Quantum Mechanics*.
Second edition, with corrections. Investigations in Physics Series.
Princeton University Press, 1996.

[23] M. E. Rose. *Elementary Theory of Angular Momentum.*
Dover books on physics and chemistry. Dover, 1995.

[24] H. J. Hogben, M. Krzystyniak, G. T. P. Charnock, P. J. Hore, and Ilya Kuprov. "Spinach – A software library for simulation of spin dynamics in large spin systems".
In: *Journal of Magnetic Resonance* 208 (2011), pp. 179–194.
DOI: 10.1016/j.jmr.2010.11.008

[25] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.*
Third edition. Johns Hopkins Studies in the Mathematical Sciences.
Johns Hopkins University Press, 1996.

[26] Charles F. Van Loan. "The ubiquitous Kronecker product". In: *Journal of Computational and Applied Mathematics* 123 (2000), pp. 85–100.
DOI: 10.1016/S0377-0427(00)00393-9

[27] Robert Piziak and P. L. Odell.
Matrix Theory. From Generalized Inverses to Jordan Form.
Chapman & Hall/CRC Pure and Applied Mathematics.
Taylor & Francis, 2007.

[28] A. Schweiger and G. Jeschke.
Principles of Pulse Electron Paramagnetic Resonance.
Oxford University Press, 2001.

[29] W. Rossmann. *Lie Groups: An Introduction Through Linear Groups.*
Oxford Mathematics. Oxford University Press, 2006.

[30] Malcolm H. Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance.*
Wiley, 2001.

[31] Michael Willer, Jörg Forrer, Jürg Keller, Sabine Van Doorslaer, Arthur Schweiger, Rolf Schuhmann, and Thomas Weiland.
"S-band (2–4 GHz) pulse electron paramagnetic resonance spectrometer: Construction, probe head design, and performance".
In: *Review of Scientific Instruments* 71.7 (2000), pp. 2807–2817.
DOI: 10.1063/1.1150697

[32] Robin K. Harris, Edwin D. Becker, Sonia M. Cabral de Menezes, Robin Goodfellow, and Pierre Granger. "NMR nomenclature. Nuclear spin properties and conventions for chemical shifts".
In: *Pure and Applied Chemistry* 73.11 (2001), pp. 1795–1818.
DOI: 10.1351/pac200173111795

[33] U. Haeberlen. *High Resolution NMR in Solids: Selective Averaging.* Vol. 1.
Advances in magnetic resonance: Supplement. Academic Press, 1976.

[34] Jozef Kowalewski and Lena Mäler.
Nuclear Spin Relaxation in Liquids: Theory, Experiments, and Applications.
Series in Chemical Physics. Taylor & Francis, 2006.

[35] Richard C. Tolman. *The Principles of Statistical Mechanics*.
Dover books on physics and chemistry. Dover Publications, 1938.

[36] U. Fano. "Description of States in Quantum Mechanics by Density Matrix and Operator Techniques".
In: *Reviews of Modern Physics* 29.1 (1957), pp. 74–93.
DOI: 10.1103/RevModPhys.29.74

[37] David Goldberg. "What Every Computer Scientist Should Know About Floating-point Arithmetic".
In: *ACM Computing Surveys* 23.1 (1991), pp. 5–48.
DOI: 10.1145/103162.103163

[38] W. Kohn. "Nobel Lecture: Electronic structure of matter—wave functions and density functionals".
In: *Reviews of Modern Physics* 71.5 (1999), pp. 1253–1266.
DOI: 10.1103/RevModPhys.71.1253

[39] D. E. Manolopoulos and P. J. Hore.
"An improved semiclassical theory of radical pair recombination reactions".
In: *The Journal of Chemical Physics* 139.12 (2013), p. 124106.
DOI: 10.1063/1.4821817

[40] H. J. Hogben, P. J. Hore, and Ilya Kuprov.
"Strategies for state space restriction in densely coupled spin systems with applications to spin chemistry".
In: *The Journal of Chemical Physics* 132 (2010), p. 174101.
DOI: 10.1063/1.3398146

[41] G. T. P. Charnock.
"Computational Spin Dynamics and Visualisation of Large Spin Systems".
DPhil thesis. University of Oxford, 2012.

[42] H. J. Hogben.
"Coherent Spin Dynamics of Radical Pairs in Weak Magnetic Fields".
DPhil thesis. University of Oxford, 2011. URL: http://ora.ox.ac.uk/objects/uuid:61c4ab7e-406f-4193-949a-b5a70f43e3e1

[43] F. Bloch. "Nuclear Induction".
In: *Physical Review* 70 (7-8 1946), pp. 460–474.
DOI: 10.1103/PhysRev.70.460

[44] R. K. Wangsness and F. Bloch.
"The Dynamical Theory of Nuclear Induction".
In: *Physical Review* 89.4 (1953), pp. 728–739.
DOI: 10.1103/PhysRev.89.728

[45] A. G. Redfield. "The Theory of Relaxation Processes". In:
ed. by John S. Waugh. Vol. 1. Advances in Magnetic Resonance.
Academic Press, 1965, pp. 1–32.
DOI: 10.1016/B978-1-4832-3114-3.50007-6

[46] Maurice Goldman. "Formal Theory of Spin–Lattice Relaxation". In: *Journal of Magnetic Resonance* 149 (2001), pp. 160–187. DOI: 10.1006/jmre.2000.2239

[47] Ilya Kuprov. "Diagonalization-free implementation of spin relaxation theory for large spin systems". In: *Journal of Magnetic Resonance* 209 (2011), pp. 31–38. DOI: 10.1016/j.jmr.2010.12.004

[48] Malcolm H. Levitt and Lorenzo Di Bari. "Steady State in Magnetic Resonance Pulse Experiments". In: *Physical Review Letters* 69.21 (1992), pp. 3124–3127. DOI: 10.1103/PhysRevLett.69.3124

[49] T. O. Levante and R. R. Ernst. "Homogeneous versus inhomogeneous quantum-mechanical master equations". In: *Chemical Physics Letters* 241 (1995), pp. 73–78. DOI: 10.1016/0009-2614(95)00578-R

[50] Randall S. Dumont, Shashi Jain, and Alex Bain. "Simulation of many-spin system dynamics via sparse matrix methodology". In: *The Journal of Chemical Physics* 106.14 (1997), pp. 5928–5936. DOI: 10.1063/1.473258

[51] Mikhail Veshtort and Robert G. Griffin. "SPINEVOLUTION: A powerful tool for the simulation of solid and liquid state NMR experiments". In: *Journal of Magnetic Resonance* 178 (2006), pp. 248–282. DOI: 10.1016/j.jmr.2005.07.018

[52] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Second edition. Society for Industrial and Applied Mathematics, 2003. URL: http://www-users.cs.umn.edu/~saad/books.html

[53] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics, 1994.

[54] John R. Gilbert, Cleve Moler, and Robert Schreiber. "Sparse Matrices in MATLAB: Design and Implementation". In: *SIAM Journal on Matrix Analysis and Applications* 13.1 (1992), pp. 333–356. DOI: 10.1137/0613024

[55] *Sparse Matrices*. MATLAB R2014a documentation. MathWorks. URL: http://www.mathworks.co.uk/help/matlab/sparse-matrices.html Retrieved 16/06/2014.

[56] M. L. Boas. *Mathematical Methods in the Physical Sciences*. Third edition. John Wiley & Sons, 2005.

[57] Stefan Stoll and Arthur Schweiger. "EasySpin, a comprehensive software package for spectral simulation and analysis in EPR".
In: *Journal of Magnetic Resonance* 178 (2006), pp. 42–55.
DOI: 10.1016/j.jmr.2005.08.013

[58] Giorgio Moro and Jack H. Freed. "Calculation of ESR spectra and related Fokker–Planck forms by the use of the Lanczos algorithm".
In: *The Journal of Chemical Physics* 74.7 (1981), pp. 3757–3773.
DOI: 10.1063/1.441604

[59] Mattias Edén, Young K. Lee, and Malcolm H. Levitt.
"Efficient Simulation of Periodic Problems in NMR. Application to Decoupling and Rotational Resonance".
In: *Journal of Magnetic Resonance, Series A* 120 (1996), pp. 56–71.
DOI: 10.1006/jmra.1996.0099

[60] M. Hohwy, H. Bildsøe, H. J. Jakobsen, and N. C. Nielsen. "Efficient Spectral Simulations in NMR of Rotating Solids. The γ-COMPUTE Algorithm".
In: *Journal of Magnetic Resonance* 136 (1999), pp. 6–14.
DOI: 10.1006/jmre.1998.1593

[61] *SpinDynamica. Magnetic Resonance in Mathematica.*
URL: www.SpinDynamica.soton.ac.uk Retrieved 16/06/2014.

[62] *EasySpin.* URL: http://www.easyspin.org/ Retrieved 16/06/2014.

[63] Stefan Stoll and R. David Britt.
"General and efficient simulation of pulse EPR spectra".
In: *Phys. Chem. Chem. Phys.* 11 (31 2009), pp. 6614–6625.
DOI: 10.1039/B907277B

[64] S. A. Smith, T. O. Levante, B. H. Meier, and R. R. Ernst.
"Computer Simulations in Magnetic Resonance. An Object-Oriented Programming Approach".
In: *Journal of Magnetic Resonance, Series A* 106 (1994), pp. 75–105.
DOI: 10.1006/jmra.1994.1008

[65] *GAMMA with PyGAMMA.*
URL: http://scion.duhs.duke.edu/vespa/gamma Retrieved 16/06/2014.

[66] Mads Bak, Jimmy T. Rasmussen, and Niels Chr. Nielsen. "SIMPSON: A General Simulation Program for Solid-State NMR Spectroscopy".
In: *Journal of Magnetic Resonance* 147 (2000), pp. 296–330.
DOI: 10.1006/jmre.2000.2179

[67] *SIMPSON.* URL: http://nmr.au.dk/software/simpson/
Retrieved 16/06/2014.

[68] *Spinach library.* Version 1.4.2114.
URL: http://spindynamics.org/Spinach.php Retrieved 16/06/2014.

[69] *SpinEvolution.* URL: http://spinevolution.com/ Retrieved 16/06/2014.

[70] Cleve Moler and Charles Van Loan. "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later". In: *SIAM Review* 45.1 (2003), pp. 3–49. DOI: 10.1137/S00361445024180

[71] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008, p. 425.

[72] L. T. Muus. "Superoperators, time-ordering, and density operators". In: *Electron spin relaxation in liquids*. Ed. by L. T. Muus and P. W. Atkins. Plenum Press, 1972, pp. 1–24.

[73] Albert Messiah. *Quantum Mechanics*. Dover books on physics. Two volumes bound as one. Dover Publications, 1999.

[74] Wilhelm Magnus. "On the Exponential Solution of Differential Equations for a Linear Operator". In: *Communications on Pure and Applied Mathematics* 7 (1954), pp. 649–673. DOI: 10.1002/cpa.3160070404

[75] N. N. Bogoliubov and I. U. A. Mitropolsky. *Asymptotic methods in the theory of non-linear oscillations*. Russian monographs and texts on advanced mathematics and physics. Hindustan Publishing Corporation (India), 1961.

[76] L. L. Buishvili, E. B. Volzhan, and M. G. Menabde. "Higher approximations in the theory of the average Hamiltonian". In: *Theoretical and Mathematical Physics* 46.2 (1981), pp. 166–173. DOI: 10.1007/BF01030852

[77] Alexander Karabanov, Anniek van der Drift, Luke J. Edwards, Ilya Kuprov, and Walter Köckenberger. "Quantum mechanical simulation of solid effect dynamic nuclear polarisation using Krylov–Bogolyubov time averaging and a restricted state-space". In: *Phys. Chem. Chem. Phys.* 14.8 (2012), pp. 2658–2668. DOI: 10.1039/c2cp23233b

[78] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[79] Ilya Kuprov, Nicola Wagner-Rundell, and P. J. Hore. "Polynomially scaling spin dynamics simulation algorithm based on adaptive state-space restriction". In: *Journal of Magnetic Resonance* 189 (2007), pp. 241–250. DOI: 10.1016/j.jmr.2007.09.014

[80] Mark C. Butler, Jean–Nicolas Dumez, and Lyndon Emsley. "Dynamics of large nuclear-spin systems from low-order correlations in Liouville space". In: *Chemical Physics Letters* 477 (2009), pp. 377–381. DOI: 10.1016/j.cplett.2009.07.017

[81] R. Brüschweiler and R. R. Ernst. "A Cog-Wheel Model for Nuclear-Spin Propagation in Solids". In: *Journal of Magnetic Resonance* 124 (1997), pp. 122–126. DOI: 10.1006/jmre.1996.1002

[82] Claudiu Filip, Siegfried Hafner, Ingo Schnell, Dan E. Demco, and Hans W. Spiess. "Solid-state nuclear magnetic resonance spectra of dipolar-coupled multi-spin systems under fast magic angle spinning". In: *The Journal of Chemical Physics* 110.1 (1999), pp. 423–440. DOI: 10.1063/1.478135

[83] A. Abragam. *The Principles of Nuclear Magnetism*. International Series of Monographs on Physics. Clarendon Press, 1961.

[84] P. J. Hore and R. W. Broadhurst. "Photo-CIDNP of biopolymers". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 25 (1993), pp. 345–402. DOI: 10.1016/0079-6565(93)80002-B

[85] Jack H. Freed and J. Boiden Pedersen. "The Theory of Chemically Induced Dynamic Spin Polarization". In: ed. by John S. Waugh. Vol. 8. Advances in Magnetic Resonance. Academic Press, 1976, pp. 1–84. DOI: 10.1016/B978-0-12-025508-5.50006-2

[86] Johannes Natterer and Joachim Bargon. "Parahydrogen induced polarization". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 31 (1997), pp. 293–315. DOI: 10.1016/S0079-6565(97)00007-1

[87] S. Berger and S. Braun. *200 and more NMR experiments: a practical course*. Wiley-VCH, 2004.

[88] Craig P. Butts, Catharine R. Jones, and Jeremy N. Harvey. "High precision NOEs as a probe for low level conformers—a second conformation of strychnine". In: *Chem. Commun.* 47 (2011), pp. 1193–1195. DOI: 10.1039/C0CC04114A

[89] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction To Algorithms*. Second edition. MIT Press, 2001.

[90] Robert Tarjan. "Depth-First Search and Linear Graph Algorithms". In: *SIAM Journal on Computing* 1.2 (1972), pp. 146–160. DOI: 10.1137/0201010

[91] L. D. Landau and E. M. Lifshitz. *Mechanics*. Third edition. Vol. 3. Course of theoretical physics. Elsevier Science, 1976.

[92] Aydın Buluç and John R. Gilbert. "Challenges and Advances in Parallel Sparse Matrix-Matrix Multiplication". In: *37th International Conference on Parallel Processing*. 2008, pp. 503–510. DOI: 10.1109/ICPP.2008.45

[93] Fred G. Gustavson. "Two Fast Algorithms for Sparse Matrices: Multiplication and Permuted Transposition". In: *ACM Transactions on Mathematical Software* 4.3 (1978), pp. 250–269. DOI: 10.1145/355791.355796

[94] Ilya Kuprov. "Polynomially scaling spin dynamics II: Further state-space compression using Krylov subspace techniques and zero track elimination". In: *Journal of Magnetic Resonance* 195 (2008), pp. 45–51. DOI: 10.1016/j.jmr.2008.08.008

[95] Giacomo Mazzi and Benedict J. Leimkuhler. "Dimensional Reductions for the Computation of Time-Dependent Quantum Expectations". In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 2024–2038. DOI: 10.1137/100788148

[96] Henk A. van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2003.

[97] Giorgio Moro and Jack H. Freed. "Efficient Computation of Magnetic Resonance Spectra and Related Correlation Functions from Stochastic Liouville Equations". In: *The Journal of Physical Chemistry* 84.22 (1980), pp. 2837–2840. DOI: 10.1021/j100459a001

[98] Roger B. Sidje. "Expokit: A Software Package for Computing Matrix Exponentials". In: *ACM Transactions on Mathematical Software* 24.1 (1998), pp. 130–156. DOI: 10.1145/285861.285868

[99] *Expokit*. URL: http://www.maths.uq.edu.au/expokit Retrieved 16/06/2014.

[100] Ronnie Kosloff. "Propagation Methods for Quantum Molecular Dynamics". In: *Annual Review of Physical Chemistry* 45.1 (1994), pp. 145–178. DOI: 10.1146/annurev.pc.45.100194.001045

[101] M. J. Frisch et al. *Gaussian 03*. Gaussian Inc. Wallingford CT 2003.

[102] Jennifer M. Duker and Anthony S. Serianni. "(^{13}C)-Substituted sucrose: ^{13}C–^{1}H and ^{13}C–^{13}C spin coupling constants to assess furanose ring and glycosidic bond conformations in aqueous solution". In: *Carbohydrate Research* 249 (1993), pp. 281–303. DOI: 10.1016/0008-6215(93)84096-O

[103] Ad Bax and Mitsuhiko Ikura. "An efficient 3D NMR technique for correlating the proton and ^{15}N backbone amide resonances with the α-carbon of the preceding residue in uniformly ^{15}N/^{13}C enriched proteins". In: *Journal of Biomolecular NMR* 1 (1991), pp. 99–104. DOI: 10.1007/BF01874573

[104] D. V. Savostyanov, S. V. Dolgov, J. M. Werner, and Ilya Kuprov. "Exact NMR simulation of protein-size spin systems using tensor train formalism". In: *Physical Review B* 90 (2014), p. 085139. DOI: 10.1103/PhysRevB.90.085139

[105] Georg Hager and Gerhard Wellein. "Architecture and Performance Characteristics of Modern High Performance Computers".
In: *Computational Many-Particle Physics*.
Ed. by H. Fehske, R. Schneider, and A. Weiße. Vol. 739.
Lecture Notes in Physics. Springer Berlin Heidelberg, 2008, pp. 681–730.
DOI: 10.1007/978-3-540-74686-7_26

[106] I. Foster. *Designing and Building Parallel Programs. Concepts and Tools for Parallel Software Engineering*. Addison-Wesley, 1995.
URL: http://www.mcs.anl.gov/~itf/dbpp/

[107] Ulrich Drepper. *What Every Programmer Should Know About Memory*. 2007.
URL: http://www.akkadia.org/drepper/cpumemory.pdf

[108] Gene M. Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities".
In: *Proceedings of the Spring Joint Computer Conference*.
Atlantic City, New Jersey: ACM, 1967, pp. 483–485.
DOI: 10.1145/1465482.1465560

[109] *Parallel Computing Toolbox*. MATLAB R2014a documentation. MathWorks.
URL: http://www.mathworks.co.uk/help/distcomp/index.html
Retrieved 16/06/2014.

[110] Cleve Moler. *The Intel Hypercube, part 2, reposted*. Cleve's Corner. MathWorks. Nov. 12, 2013.
URL: http://blogs.mathworks.com/cleve/2013/11/12/the-intel-hypercube-part-2-reposted/

[111] *Getting Started with parfor*. MATLAB R2014a documentation. MathWorks.
URL: http://www.mathworks.co.uk/help/distcomp/getting-started-with-parfor.html Retrieved 16/06/2014.

[112] John R. de Laeter, J. K. Böhlke, P. de Bièvre, H. Hidaka, H. S. Peiser, K. J. R. Rosman, and P. D. P. Taylor.
"Atomic weights of the elements: Review 2000 (IUPAC Technical Report)".
In: *Pure and Applied Chemistry* 75.6 (2003), pp. 683–800.
DOI: 10.1351/pac200375060683

[113] P. J. Hore. *Nuclear Magnetic Resonance*. Oxford Chemistry Primers. OUP, 2006.

[114] Ken A. Dill and Sarina Bromberg. *Molecular Driving Forces. Statistical Thermodynamics in Chemistry and Biology*. Garland Science, 2003.

[115] Ad Bax, Ray Freeman, and Stewart P. Kempsell. "Natural abundance ^{13}C–^{13}C coupling observed via double-quantum coherence".
In: *Journal of the American Chemical Society* 102.14 (1980), pp. 4849–4851.
DOI: 10.1021/ja00534a056

[116] Mattias Edén and Malcolm H. Levitt. "Computation of Orientational Averages in Solid-State NMR by Gaussian Spherical Quadrature". In: *Journal of Magnetic Resonance* 132 (1998), pp. 220–239. DOI: 10.1006/jmre.1998.1427

[117] Jørgen Holm Kristensen and Ian Farnan. "Efficient solid state NMR powder simulations using SMP and MPP parallel computation". In: *Journal of Magnetic Resonance* 161 (2003), pp. 183–190. DOI: 10.1016/S1090-7807(03)00007-7

[118] Vladislav Yu. Orekhov and Victor A. Jaravine. "Analysis of non-uniformly sampled spectra with multi-dimensional decomposition". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 59.3 (2011), pp. 271–292. DOI: 10.1016/j.pnmrs.2011.02.002

[119] Thomas E. Skinner and Steffen J. Glaser. "Representation of a quantum ensemble as a minimal set of pure states". In: *Physical Review A* 66 (3 2002), p. 032112. DOI: 10.1103/PhysRevA.66.032112

[120] T. Gradl, A. Spörl, T. Huckle, S. J. Glaser, and T. Schulte-Herbrüggen. "Parallelising Matrix Operations on Clusters for an Optimal Control-Based Quantum Compiler". In: *Euro-Par 2006 Parallel Processing*. Ed. by Wolfgang Nagel, Wolfgang Walter, and Wolfgang Lehner. Vol. 4128. Lecture Notes in Computer Science. Springer-Verlag Berlin / Heidelberg, 2006, pp. 751–762. DOI: 10.1007/11823285_78

[121] T. Auckenthaler, M. Bader, T. Huckle, A. Spörl, and K. Waldherr. "Matrix exponentials and parallel prefix computation in a quantum control problem". In: *Parallel Computing* 36 (2010), pp. 359–369. DOI: 10.1016/j.parco.2010.01.006

[122] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Second edition. Classics in Applied Mathematics. SIAM, 2011. URL: http://www-users.cs.umn.edu/~saad/books.html

[123] Ernest R. Davidson. "The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices". In: *Journal of Computational Physics* 17 (1975), pp. 87–94. DOI: 10.1016/0021-9991(75)90065-0

[124] *eigs*. MATLAB R2014a documentation. MathWorks. URL: http://www.mathworks.co.uk/help/matlab/ref/eigs.html Retrieved 16/06/2014.

[125] W. H. Press. *Numerical Recipes in C++: The Art of Scientific Computing*. Second edition. Cambridge University Press, 2002.

[126] M. W. Berry. "Large-Scale Sparse Singular Value Computations". In: *The International Journal of Supercomputer Applications* 6.1 (1992), pp. 13–49.

[127] Plamen N. Penchev, Neyko M. Stoyanov, and Marin N. Marinov. "A complete ^{1}H and ^{13}C NMR data assignment for 3-phenylmethylene-1*H*,3*H*-naphtho-[1,8-c,d]-pyran-1-one". In: *Spectrochimica Acta Part A* 78 (2011), pp. 559–565. DOI: 10.1016/j.saa.2010.11.015

[128] *wall time*. URL: http://www.catb.org/jargon/html/W/wall-time.html Retrieved 16/06/2014.

[129] *Amazon Elastic Compute Cloud Documentation*. Amazon Web Services, Inc. URL: http://aws.amazon.com/documentation/ec2/

[130] Guy E. Blelloch. *Prefix Sums and Their Applications*. Tech. rep. CMU-CS-90-190. School of Computer Science, Carnegie Mellon University, 1990.

[131] P. Pfeifer and R. D. Levine. "A stationary formulation of time-dependent problems in quantum mechanics". In: *The Journal of Chemical Physics* 79.11 (1983), pp. 5512–5519. DOI: 10.1063/1.445669

[132] Uri Peskin and Nimrod Moiseyev. "The solution of the time-dependent Schrödinger equation by the (t, t') method: Theory, computational algorithm and applications". In: *The Journal of Chemical Physics* 99.6 (1993), pp. 4590–4596. DOI: 10.1063/1.466058

[133] Michal Leskes, P. K. Madhu, and Shimon Vega. "Floquet theory in solid-state nuclear magnetic resonance". In: *Progress in Nuclear Magnetic Resonance Spectroscopy* 57 (2010), pp. 345–380. DOI: 10.1016/j.pnmrs.2010.06.002

[134] T. O. Levante, M. Baldus, B. H. Meier, and R. R. Ernst. "Formalized quantum mechanical Floquet theory and its application to sample spinning in nuclear magnetic resonance". In: *Molecular Physics* 86.5 (1995), pp. 1195–1212. DOI: 10.1080/00268979500102671

[135] Alexander A. Nevzorov and Jack H. Freed. "Spin relaxation by dipolar coupling: From motional narrowing to the rigid limit". In: *The Journal of Chemical Physics* 112.3 (2000), pp. 1413–1424. DOI: 10.1063/1.480695

[136] Jack H. Freed. "Theory of slow tumbling ESR spectra for nitroxides". In: *Spin labeling: theory and applications*. Ed. by Lawrence J. Berliner. Academic Press, inc., London, 1976, pp. 53–132.

[137] Jack H. Freed. "ESR line shapes and saturation in the slow motional region—the stochastic Liouville approach". In: *Electron spin relaxation in liquids*. Ed. by L. T. Muus and P. W. Atkins. Plenum Press, 1972, pp. 387–410.

[138] S. Chandrasekhar. "Stochastic Problems in Physics and Astronomy". In: *Reviews of Modern Physics* 15.1 (1943), pp. 1–89. DOI: 10.1103/RevModPhys.15.1

[139] J. Boiden Pedersen. "Stochastic processes". In: *Electron spin relaxation in liquids*. Ed. by L. T. Muus and P. W. Atkins. Plenum Press, 1972, pp. 25–70.

[140] P. A. M. Dirac. *The principles of quantum mechanics*. Fourth edition. Vol. 27. The international series of monographs on physics. Clarendon press, 2007.

[141] G. Stephenson. *Mathematical methods for science students*. Second edition. Longman text. Longman Publishing Group, 1973.

[142] Roy G. Gordon and Thomas Messenger. "Magnetic resonance lineshapes in slowly tumbling molecules". In: *Electron spin relaxation in liquids*. Ed. by L. T. Muus and P. W. Atkins. Plenum Press, 1972, pp. 341–382.

[143] A. Coddington and N. Levinson. *Theory of Ordinary Differential Equations*. International series in pure and applied mathematics. McGraw-Hill, 1955.

[144] Jon H. Shirley. "Solution of the Schrödinger Equation with a Hamiltonian Periodic in Time". In: *Physical Review* 138.4B (1965), B979–B987. DOI: 10.1103/PhysRev.138.B979

[145] Jon H. Shirley. "Interaction of a quantum system with a strong oscillating field". PhD thesis. California Institute of Technology, 1963. URL: http://resolver.caltech.edu/CaltechETD:etd-05142008-103758

[146] P. Hazendonk, Alex D. Bain, H. Grondey, P. H. M. Harrison, and R. S. Dumont. "Simulations of Chemical Exchange Lineshapes in CP/MAS Spectra Using Floquet Theory and Sparse Matrix Methods". In: *Journal of Magnetic Resonance* 146 (2000), pp. 33–42. DOI: 10.1006/jmre.2000.2111

[147] D. Williams. *Probability with Martingales*. Cambridge mathematical textbooks. Cambridge University Press, 1991.

[148] G. V. Bruno and J. H. Freed. "Analysis of Inertial Effects on Electron Spin Resonance Spectra in the Slow Tumbling Region". In: *The Journal of Physical Chemistry* 78.9 (1974), pp. 935–940. DOI: 10.1021/j100602a015

[149] M. Fixman and K. Rider. "Angular Relaxation of the Symmetrical Top". In: *The Journal of Chemical Physics* 51.6 (1969), pp. 2425–2438. DOI: 10.1063/1.1672362

[150] John Jones. *Amino Acids and Peptide Synthesis*. Second edition. Oxford Chemistry Primers. Oxford University Press, 2002.

[151] Malcolm H. Levitt. "Why Do Spinning Sidebands Have the Same Phase?" In: *Journal of Magnetic Resonance* 82 (1989), pp. 427–433. DOI: 10.1016/0022-2364(89)90050-4

[152] Robert Zwanzig. "Ensemble Method in the Theory of Irreversibility". In: *The Journal of Chemical Physics* 33.5 (1960), pp. 1338–1341. DOI: 10.1063/1.1731409

[153] Lawrence J. Berliner. "Refinement and location of the hydrogen atoms in the nitroxide 2,2,6,6-tetramethyl-4-piperidinol-1-oxyl". In: *Acta Crystallographica Section B* 26 (1970), pp. 1198–1202. DOI: 10.1107/S0567740870003874

[154] Michael J. Minch. "Orientational Dependence of Vicinal Proton–Proton NMR Coupling Constants: The Karplus Relationship". In: *Concepts in Magnetic Resonance* 6 (1994), pp. 41–56. DOI: 10.1002/cmr.1820060104

[155] P. G. de Gennes and J. Prost. *The Physics of Liquid Crystals*. International Series of Monographs on Physics. Clarendon Press, 1995.

[156] J. Christoph Freudenberger, Peter Spiteller, Reinhard Bauer, Horst Kessler, and Burkhard Luy. "Stretched poly(dimethylsiloxane) Gels as NMR Alignment Media for Apolar and Weakly Polar Organic Solvents: An Ideal Tool for Measuring RDCs at Low Molecular Concentrations". In: *Journal of the American Chemical Society* 126.45 (2004), pp. 14690–14691. DOI: 10.1021/ja046155e

[157] Ad Bax and Alexander Grishaev. "Weak alignment NMR: a hawk-eyed view of biomolecular structure". In: *Current Opinion in Structural Biology* 15 (2005), pp. 563–570. DOI: 10.1016/j.sbi.2005.08.006

[158] Andreas Enthart, J. Christoph Freudenberger, Julien Furrer, Horst Kessler, and Burkhard Luy. "The CLIP/CLAP-HSQC: Pure absorptive spectra for the measurement of one-bond couplings". In: *Journal of Magnetic Resonance* 192 (2008), pp. 314–322. DOI: 10.1016/j.jmr.2008.03.009

[159] Ghirmai H. Meresi, Miroslava Cuperlovic, William E. Palke, and J. T. Gerig. "Pulsed Field Gradients in Simulations of One- and Two-Dimensional NMR Spectra". In: *Journal of Magnetic Resonance* 137 (1999), pp. 186–195. DOI: 10.1006/jmre.1998.1665

[160] Peter Nicholas, David Fushman, Vladislav Ruchinsky, and David Cowburn. "The Virtual NMR Spectrometer: A Computer Program for Efficient Simulation of NMR Experiments Involving Pulsed Field Gradients". In: *Journal of Magnetic Resonance* 145.2 (2000), pp. 262–275. DOI: 10.1006/jmre.2000.2108

[161] Alexej Jerschow and Norbert Müller.
"Efficient Simulation of Coherence Transfer Pathway Selection by Phase Cycling and Pulsed Field Gradients in NMR".
In: *Journal of Magnetic Resonance* 134 (1998), pp. 17–29.
DOI: 10.1006/jmre.1998.1491

[162] Luke J. Edwards. "Simulation of coherence selection by pulsed field gradients in liquid-state NMR using an auxiliary matrix formalism".
In: *Journal of Magnetic Resonance* 240 (2014), pp. 95–101.
DOI: 10.1016/j.jmr.2014.01.011

[163] Kurt Wüthrich. "NMR studies of Structure and Function of Biological Macromolecules (Nobel Lecture)".
In: *Angewandte Chemie International Edition* 42 (2003), pp. 3340–3363.
DOI: 10.1002/anie.200300595

[164] H. Günther. *NMR Spectroscopy: Basic Principles, Concepts and Applications in Chemistry*. Third edition. Wiley, 2013.

[165] Ad Bax and Susan S. Pochapsky. "Optimized Recording of Heteronuclear Multidimensional NMR Spectra Using Pulsed Field Gradients".
In: *Journal of Magnetic Resonance* 99 (1992), pp. 638–643.
DOI: 10.1016/0022-2364(92)90221-R

[166] C. Griesinger, O. W. Sørensen, and R. R. Ernst. "Three-Dimensional Fourier Spectroscopy. Application to High-Resolution NMR".
In: *Journal of Magnetic Resonance* 84 (1989), pp. 14–63.
DOI: 10.1016/0022-2364(89)90004-8

[167] Ole Sørensen.
"Aspects and Prospects of Multidimensional Time-Domain Spectroscopy".
In: *Journal of Magnetic Resonance* 89 (1990), pp. 210–216.
DOI: 10.1016/0022-2364(90)90178-C

[168] Giovanni Lipari and Attila Szabo.
"Model-Free Approach to the Interpretation of Nuclear Magnetic Resonance Relaxation in Macromolecules. 1. Theory and Range of Validity".
In: *Journal of the American Chemical Society* 104 (1982), pp. 4546–4559.
DOI: 10.1021/ja00381a009