

***The Development and Application of New NMR
Methods for Measuring Diffusion in Biological
and Non-Biological Heterogeneous Systems***

Simon Andrew Clark BSc (London)

Thesis Submitted for the Degree of Doctor of Philosophy

Department of Physical Chemistry

University of Leicester

September 1997

UMI Number: U123422

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U123422

Published by ProQuest LLC 2013. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

The aim of this work has been to design and develop a series of new NMR tools and data analysis techniques to measure diffusion, and in particular, restricted diffusion within heterogeneous systems. This area of research has many different applications in medicine and the pharmaceutical, oil-recovery, food, and chemicals industries. The information about fluid filled structures, be they cells, rock pores, or emulsions, and how these may alter under different conditions, is of great interest.

The work covered in this thesis is grouped into three sections, where in all cases constant, static magnetic field gradients are used, either by choice or necessity, for encoding diffusion.

The first section covers the design and development of a technique to isolate the signal of restricted diffusing spins from that of freely diffusing spins. This results in a more accurate and robust technique for quantifying restriction parameters relating to the size and structure of the restricting barrier.

The second study approaches the problem of measuring diffusion in heterogeneous environments where differences in magnetic susceptibility of the constituent parts gives rise to strong internal magnetic field gradients. Traditional techniques try to overcome these gradients in a manner of different ways. Here, the intrinsic gradients are poacher turned game-keeper, and used themselves as the diffusion encoding gradients.

The final study again uses large constant magnetic field gradients. Here, the fringe field of a superconducting magnet is used. Different pulse sequences and techniques are modelled and used experimentally to demonstrate how liquid diffusion coefficients may be measured, and it is shown that simple pulse sequences which do not allow for relaxation lead to inaccuracies.

Acknowledgements

This work was carried out with the funding from the Engineering and Physical Sciences Research Council of Great Britain and from Zeneca Pharmaceuticals. The small-bore imaging and fringe-field work was performed using the University of London Intercollegiate Research Service equipment at Queen Mary and Westfield College.

The assistance and friendship of all the members of the Chemistry Department was greatly appreciated. In particular, I would like to thank David Hardy, Gerry Griffith, Mike Blandamer and Amirah Chaudry for their technical expertise together with Paul Cullis for his patience in waiting for a Leicester title. I would also like to thank Rebekah Jukes and Chris Nolan of the Centre for Mechanisms of Human Toxicity in Leicester for their contribution in providing NMR time and surgical skills respectively.

Outside of Leicester, my sincere gratitude goes to Steve Williams, who after recommending me to Leicester, shot off to the Institute of Psychiatry in London; Thanks for providing the imaging time, technical expertise, and enthusiasm that enticed me to NMR in the first place. I should also like to thank Paul Kinchesh, manager of the ULIRS at QMW for all of his help, especially the fringe-field study.

Thanks to the personnel of Zeneca Pharmaceuticals and in particular John Waterton whose broad understanding and realism gave an insight into commercial MRI. I would also like to acknowledge Bob Kleinberg of Schlumberger-Doll Research for providing the glass beads and susceptibility measurements used this thesis.

For my research time, I was blessed with two fantastic supervisors in Tim Norwood and Mark Horsfield, both to whom I am massively indebted. Both have been described to me at some time or other as having “brains the size of planets” and metaphorically they do, both have a massive amount of understanding and insight from which I was able to learn, and develop ideas of my own; a big thankyou to you both.

Finally, but most importantly, I must acknowledge Sarah Hussain, whose love, support and understanding has always been there for me and without whose strength none of this would have ever been possible; thanks Babe.

In memory of Tim,
friend, teacher and gentleman; I will miss you greatly.

CONTENTS.

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Measurement of Diffusion | 1 |
| 1.2 | Nuclear Magnetic Resonance. | 2 |
| 1.3 | Scope of Diffusion Studies by NMR | 3 |
| 1.4 | Scope of this thesis. | 4 |
| 2 | Theory | 5 |
| 2.1 | NMR Theory | 5 |
| 2.1.1 | Basic NMR | 5 |
| 2.1.2 | Spin Echoes. | 11 |
| 2.1.3 | Product Operators | 14 |
| 2.1.4 | Coherence Transfer and Phase Cycling | 16 |
| 2.2 | Diffusion. | 20 |
| 2.2.1 | Restricted Diffusion | 21 |
| 2.2.2 | Measuring Diffusion by NMR | 24 |
| 3 | A Robust NMR Technique for Measuring the Signal of Molecules whose Diffusion has been Restricted | 31 |
| 3.1 | Introduction and Aim | 31 |
| 3.2 | Pulse Sequence Design | 31 |
| 3.3 | Theoretical Analysis | 33 |
| 3.3.1 | Model 1 (Von Meerwall and Ferguson). | 34 |
| 3.3.2 | Model 2 (Balinov <i>et al.</i>). | 37 |
| 3.4 | Experimental 1 | 40 |
| 3.5 | Results and Analysis | 41 |
| 3.5.1 | Semi-quantitative Analysis | 43 |
| 3.6 | RD-CTPG Imaging Studies | 48 |
| 3.6.1 | NMR Imaging Theory. | 49 |

| | | |
|-------|----------------------------------|----|
| 3.6.2 | Experimental 2 | 51 |
| 3.6.3 | Results and Discussion | 52 |
| 3.7 | Summary | 54 |

4 NMR Diffusion Measurements in Heterogeneous Samples with

| | |
|--|--|
| Large Susceptibility Variations | 55 |
| 4.1 | Introduction 55 |
| 4.2 | Magnetic Susceptibility 56 |
| 4.3 | Theory 57 |
| 4.4 | Computer Simulations. 61 |
| 4.4.1 | Cubic Pack Field Model 62 |
| 4.4.2 | Hexagonal Pack Field Model. 64 |
| 4.4.3 | Line-shape Simulation 65 |
| 4.4.4 | Diffusion Experiment Simulation 71 |
| 4.5 | Experimental Methods 74 |
| 4.5.1 | Glass Bead Packs 74 |
| 4.5.2 | Lung 75 |
| 4.6 | Results 77 |
| 4.6.1 | Glass Bead Packs 77 |
| 4.6.2 | Lung 81 |
| 4.6.3 | Computer Simulations. 85 |
| 4.7 | Discussion 87 |
| 4.8 | Summary 90 |

5 Measuring Diffusion in the Fringe Fields of

| | |
|--|--|
| Superconducting Magnets | 92 |
| 5.1 | Introduction 92 |
| 5.2 | Theory 92 |
| 5.2.1 | Relaxation Considerations 95 |
| 5.2.2 | Pulse Width Considerations 98 |
| 4.4.4 | Diffusion Experiment Simulation 99 |
| 5.4 | Experimental Methods 101 |

| | | |
|----------|---|------------|
| 5.5 | Results | 104 |
| 5.6 | Discussion | 105 |
| 5.7 | Summary | 106 |
| 6 | Summary and Conclusions | 108 |
| 7 | Appendices | 109 |
| 1 | Balinov Simulation | I |
| 2 | Line-shape Simulation (Cubic Model) | V |
| 3 | Line-shape Simulation (Hexagonal Model) | XIII |
| 4 | Line-shape Simulation (Random Model) | XXV |
| 5 | Line-shape Simulation with Random B_0 (Cubic Model) | XXXIII |
| 6 | Line-shape Simulation with Random B_0 (Hexagonal Model) | XXXIX |
| 7 | Diffusion Experiment Simulation in Cubic Model. | L |
| 8 | Relaxation Errors in ADC (Spin Echo) | LXIV |
| 9 | Relaxation Errors in ADC (Stimulated Echo) (5.2.2). | LXVII |

Abbreviations:

Constants

γ Gyromagnetic ratio for proton ($26.7520 \times 10^7 \text{ s}^{-1}\text{T}^{-1}$).

k Boltzmann constant ($1.38066 \times 10^{-23} \text{ JK}^{-1}$).

h Plancks constant ($6.62618 \times 10^{-34} \text{ JHz}^{-1}$).

Variables

I Spin quantum number.

m_I Magnetic quantum number.

μ Magnetic moment.

H Applied magnetic field.

B Net magnetic field.

M Magnetisation.

$A_{(t)}/A_{(0)}$ Amount of signal observed at time t as a ratio of signal at time zero.

ω Precessional frequency.

σ Electronic shielding constant.

Φ Magnetic phase.

δ Duration of magnetic field gradient pulse.

Δ Diffusion time.

G Magnetic field gradient strength.

χ Volume magnetic susceptibility.

ρ Cavity/pore radius.

D Diffusion coefficient.

T Absolute temperature.

T_1 Longitudinal relaxation constant.

T_2 Transverse relaxation constant.

λ Wavelength.

Pulse Sequences

CPMG Carr Purcell Meiboom Gill.
PGSE Pulsed Gradient Spin Echo.
CTPG Constant Time Pulse Gradient.
OE-CTPG One Echo CTPG.
RD-CTPG Restricted Diffusion CTPG.

Terminology

FID Free Induction Decay.
RF Radio Frequency.
Acq. Acquisition.
CPU Computer Processing Units.
NMR Nuclear Magnetic Resonance.
MRI Magnetic Resonance Imaging.
CAT Computer Assisted Tomography.
STRAFI STRAy Field Imaging.
MFG Magnetic Field Gradient.
ADC Apparent Diffusion Coefficient.
SDC Self-Diffusion Coefficient.
RMS Root Mean Square.

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!', but "That's funny...".

Isaac Asimov

Chapter 1

Introduction.

1.1 Measurement of Diffusion.

Traditionally, diffusion coefficients were measured with radioactive tracer techniques. Tracer methods are the most accurate techniques for this purpose. Unfortunately, they are time consuming because they require isotopic labelling; often difficult synthetic work and measurements for a single component may require days or even weeks. These drawbacks can be overcome by NMR, resulting in quick and precise self-diffusion coefficients of individual components of multi-component systems without isotopic labelling.

Diffusion studies can provide information on molecular size, viscosity of the diffusing media and an insight into the diffusing environment. Self-diffusion rates also respond to structural changes in macromolecular systems in solutions. Additionally, self-diffusion coefficients are directly related to molecular displacement and may require no further interpretation. Restricted diffusion in biological tissues may be exploited to extract cell size and permeability values; in rocks it can provide pore size and their interconnectivity; and in imaging it can be used as a non-invasive method for better assessment of particular diseases and medical disorders. Since the involvement of NMR there has been a rapid growth in diffusion studies in solution and in the solid state.

It is possible to probe diffusion using NMR calculating a diffusion coefficient from the attenuation of an echo formed in the presence of a pulsed magnetic field gradient. Diffusion experiments are based on the conventional spin echo pulse sequence with magnetic field gradients applied during dephasing and rephasing. The degree of irreversible dephasing depends on the diffusion coefficient and is proportional to the strength of the gradient pulse. The most commonly used sequence is Pulsed Gradient Spin Echo (PGSE) which is discussed in detail along with its variants, and compared to new techniques for observing diffusion by NMR. The affects of restricting boundaries have on these measurements are also analysed.

1.2 Nuclear Magnetic Resonance.

Nuclear magnetic resonance (NMR) spectroscopy has been used as a tool in the analysis of chemical structure since the 1950s, where the frequency of peaks, together with their relative intensities and splittings give a valuable insight to the location of nuclei in a molecule. Proton (^1H) spectra were the first to be used and are still the most common, though nuclei such as carbon (^{13}C), fluorine (^{19}F) and phosphorous (^{31}P) are frequently used on modern spectrometers. Progress in the development of superconducting magnets, computers and experimental techniques have given rise to a broader range of applications such as high-resolution multidimensional NMR and magnetic resonance imaging (MRI).

High-resolution NMR spectroscopy is used in the analysis of simple chemical structure, but increasingly as a tool to probe the structure of larger organic molecules and biological macromolecules such as proteins. For example, this technology is widely used in the pharmaceutical industry to understand the structure of macromolecules such as proteins and locate their active sites. This assists organic chemists with the design of drugs to interact with specific sites within the tertiary structure of the protein molecule. These techniques are typified by the use of a high magnetic field (up to 15 Tesla) spectrometer to ensure maximum sensitivity, with a highly uniform magnetic field (better than 1 part in 100 million some cases) across a small sample.

The field of NMR imaging has seen enormous growth since its conception by Lauterbur in 1973. Its potential for medical diagnosis was soon recognised by medical research scientists seeking to comprehend some of the vast amount of physiological and biological information that can be encoded into the NMR signal. Because study of the human body requires a large-diameter magnet, medical imaging machines tend to have a somewhat poorer field homogeneity and for this and other reasons, a lower magnetic field strength (normally between 0.1 and 2.0 Tesla). MRI uses magnetic field gradients to encode spatial information into the NMR signal. For example, a simple two-dimensional NMR image encodes the information (for instance proton density) in two perpendicular directions to give a two-dimensional proton density map of the sample, such as a human brain. Since water is the major proton source and the local environment affects the physical properties of the water (including its self-diffusion coefficient), it is possible to distinguish between regions of different tissues. As such, this technique offers a non-invasive way to look inside a patient, or any other appropriate sample for non-medical applications.

The other main cross-sectional technique adopted for such non-invasive medical imaging is X-ray computer-aided tomography (CAT). With this technique, the mechanism of contrast generation is the variation of atomic mass in the patient. This allows for strong contrast

between bone and soft tissue, though very poor discrimination between different tissue types. There are also the dangers associated with X-rays; even at the low dosage given by modern scanners, the number of examinations that can be made safely is limited. At present, there is no known risk from the magnetic field of an MRI scanner; the only restriction on its use arises from the possibility of patient heating caused by RF power deposition.

The non-invasive nature of NMR lends itself to applications beyond the realms of medicine and into other research and industrial applications, some of which are now being performed. Studies observing the internal structure and physical properties of opaque objects have been carried out over the last two decades with great interest, and the ability to distinguish between chemical species has enabled the study of heterogeneous systems, an area discussed at length in this thesis.

The advantages of the NMR methodology have been introduced in this chapter. It is appropriate that its limitations are also introduced at this point. NMR is essentially an insensitive technique requiring sophisticated electronics to minimise corruption of the small amount of available signal. This can require long periods to acquire data either to improve data quality, or to extract the useful information from the large number of parameters available. This limitation is partially technology driven, but on the whole implicit to the phenomenon. In addition, a skilled operator with a clear understanding of NMR principles may also be required unless routine procedures alone are to be implemented. The capital and running costs of equipment can be a prohibitive factor, although recent and future developments will reduce both of these: for example, the introduction of improved superconductors may lead to more efficient magnets. An area that was previously a concern, the processing of vast amounts of data, is rapidly becoming a lesser problem, as computer power over the past decade has exploded and the associated costs fallen.

1.3 Scope of Diffusion Studies by NMR.

The measurement of self-diffusion coefficients by NMR techniques has been around since the 1960s. The accuracy of these measurements is very high and the time it takes to perform such experiments is much less than radioactive trace techniques, and needless to say safer. As with many scientific measurements, it is not the absolute measurement of these values that is of greatest interest despite their own scientific merits, it is their change that is the key to explaining the physical properties of the system. Diffusion is by definition a dynamic process and it is the restriction of diffusion that is exploited to give us information about the surrounding environment. Since magnetic field gradients can be applied in any direction and are necessary for the measurement of diffusion, it is possible to see how fluid may diffuse in

different directions and by different amounts thus giving information about the structure of the surrounding environment. It is this idea that has been adopted in analysing the structure of many heterogeneous structures such as brain physiology, porous rock structure and food composition, areas covered in this thesis.

1.4 Scope of this thesis.

The thesis is organised so that the reader is initially introduced to the theory and background of diffusion and NMR in Chapter 2. The work then leads into the main three research areas, each covered in a separate chapter. In Chapter 3, a new technique is devised to isolate the effects on signal attenuation from restricted spins and goes on to compare and contrast the technique with existing studies for its analytical use in obtaining information about the systems structure. The fourth chapter continues along the theme of obtaining information about the structure of a heterogeneous system, but novelly makes use of intrinsic magnetic susceptibility gradients as a means of encoding the diffusion. The final research chapter investigates the ways in which diffusion can be measured in very strong fringe-fields and analyses the relative affects of relaxation on the accuracy of the ADC measurements obtained. The work is summarised and concluded in the final chapter after which the appendices containing the source code for computer simulation performed in this thesis are found.

Chapter 2

Theory.

2.1 NMR Theory.

The vast majority of NMR studies of diffusion involve the observation of the change in the NMR signal of hydrogen nuclei. Here the NMR properties of the proton will be explained though the theory is equally applicable to other nuclei whose nuclear spin quantum number, I , is equal to $\frac{1}{2}$.

2.1.1 Basic NMR.

The hydrogen nucleus (or proton) can be considered as a positively charged sphere that spins and obeys the laws of quantum mechanics. The spinning charge gives rise to a magnetic moment, μ , about the proton. When a proton is placed in a static magnetic field it will align itself in the field. The angular momentum or spin of a nucleus is quantised. The angular momentum quantum number of the nucleus determines how many orientations it takes up in the magnetic field. In general, a nucleus with a spin quantum number I , can take $(2I+1)$ orientations, each of which is characterised by a magnetic quantum number m_I , where $m_I = I, (I-1), \dots, (I-n), -I$. In the case of the proton $I=\frac{1}{2}$, and consequently, for the proton there are two possible orientations, where m_I has a value of either $+\frac{1}{2}$ or $-\frac{1}{2}$. These states have different energies, and fig.[2.1.1] shows the orientations of a proton in these states vectorally. It is found that more spins lie in the lower energy state in the direction of the magnetic field than opposed to it. However, the excess of one state over the other is extremely small and for every million spins there are typically only three or four extra in the lower energy state. The two energy states are known by convention as α and β , where α has the lower energy of the two.

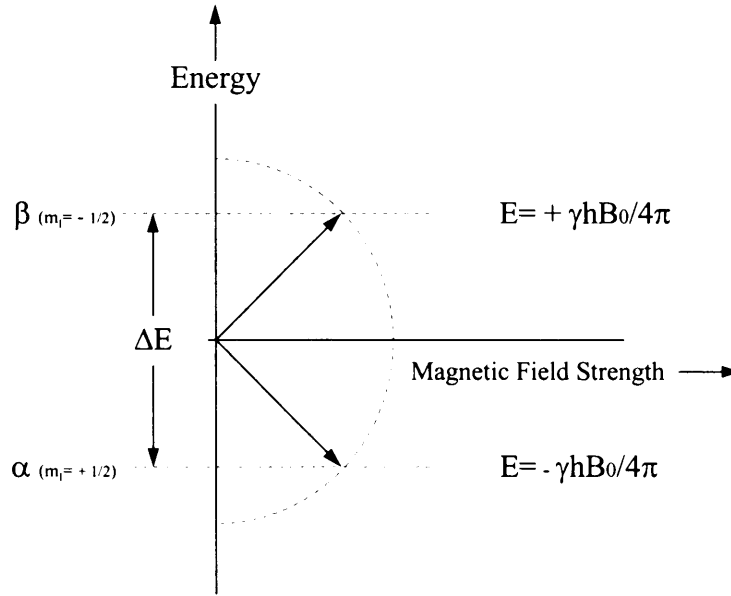


Figure 2.1. 1: In the presence of a magnetic field a proton can assume two orientations, the α and β states. As the magnetic field strength increases the difference between the energy level increases. This together with the gyromagnetic ratio governs the sensitivity of the NMR experiment. The radius of the semi-circle indicates the magnetic field strength and that the greater the strength, the greater the value of ΔE .

This distribution of spins between the two states is described by the Boltzmann equation:

$$\frac{N_{\alpha}}{N_{\beta}} = \exp\left(-\frac{\Delta E}{kT}\right) \quad [2.1.1]$$

$$N_{(\alpha-\beta)} = N_{(\alpha)} - N_{(\beta)} = N_{(\alpha+\beta)}\left(\frac{\Delta E}{2kT}\right) \quad [2.1.2]$$

where N_{α} is the number of parallel spins in the lower energy α state and N_{β} is the number of antiparallel spins in the higher energy β state. $N_{\alpha+\beta}$ and $N_{\alpha-\beta}$ are the sum and difference in population of the two states respectively. ΔE is the energy difference between the two states, k is the Boltzmann constant and T is the absolute temperature at equilibrium. The result is a small net magnetisation along the direction of the static magnetic field. The energies of the α and β states and therefore ΔE can be predicted using basic quantum mechanics. If eqn.[2.1.1] is rearranged to show the population difference (eqn.[2.1.2]) and incorporated with the information given in fig.[2.1.1], the net magnetisation, M_0 , can be calculated by multiplying the population difference by the magnetic moment, μ :

$$M_0 = N_{(\alpha-\beta)}\mu \quad [2.1.3]$$

Hence a higher magnetic field strength will give a larger net magnetisation which in turn increases the sensitivity of the NMR experiment.

When placed in a static magnetic field, B_0 , which is conventionally taken as being along the z axis, the nuclear spins precess about it, in analogy to a gyroscope precessing in a gravitational field. These spins are randomly distributed in a cone about the z-axis at a fixed angle θ from the z-axis caused by the interaction between B_0 and the magnetic moment of the spins. This angle is known as the *magic angle* and is calculated from $\cos\theta = m/[I(I+1)]^{1/2}$ and the frequency with which they precess about B_0 is known as the Larmor frequency (eqn.[2.1.4]).

$$\omega = (1 - \sigma)\gamma B_0 \quad [2.1.4]$$

σ is a shielding constant specific to the chemical and hence electronic environment of the spin concerned.

The individual spins can each be represented semi-classically by vectors as shown in fig.[2.1.2a]. The magnetic fields arising from these vectors, as previously mentioned, combine to give a net magnetisation which can also be represented by a vector. This net magnetisation vector is illustrated in fig.[2.1.2b] and is seen to be in the direction of B_0 (along the z axis), due to the small excess of spins in the α state. The vector has no net x or y component since the phase of spins in the precessional cone is random.

To observe an NMR signal, it is necessary to apply a radiofrequency (RF) pulse to the system. The energy provided by the pulse alters the distribution of spins between the energy states so that the net magnetisation vector is disturbed from its equilibrium. The phase of the RF pulse determines the axis about which the net magnetisation vector will be rotated. After the RF pulse is completed, the vector continues to precess about B_0 . It is the observation of the coherent spins precessing about B_0 at the Larmor frequency that forms the basis of most NMR experiments. The spin system reverts to the equilibrium state by means of relaxation processes which can be characterised by specific relaxation times for any given spin species in a system. There are two main modes of relaxation. T_1 is the characteristic time taken for the populations of the energy states to return towards a Boltzmann distribution. T_2 is the characteristic time taken for the coherence of magnetisation in the xy plane to return towards equilibrium (i.e. zero). These processes will be explained further at a later point.

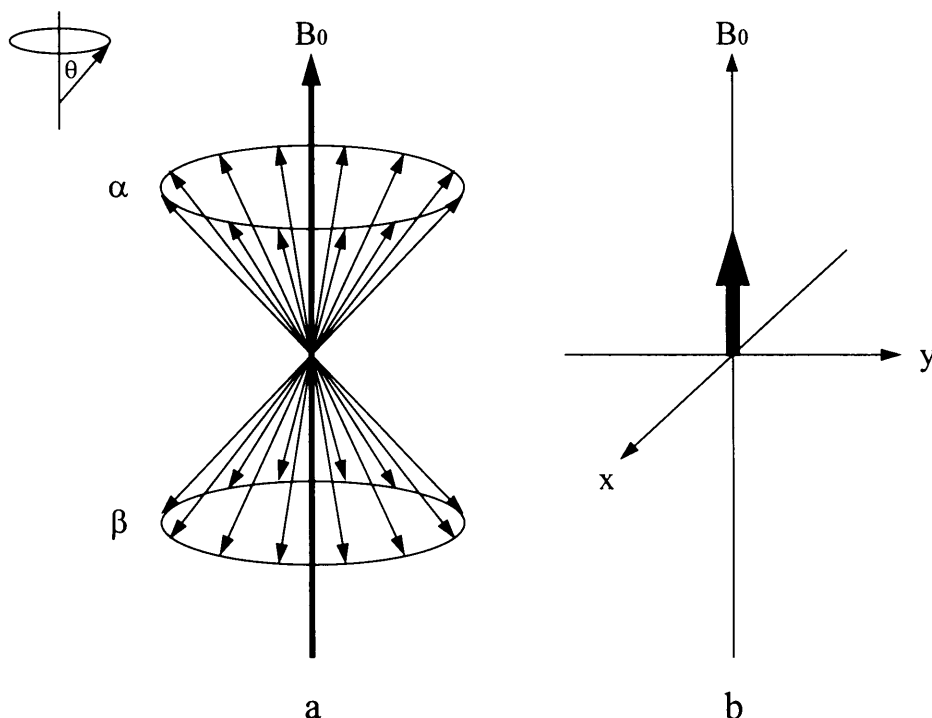


Figure 2.1. 2: Individual spins assume either the α or β orientation in a magnetic field (a). Each vector lies at the magic angle θ from the B_0 direction in a random spread. Due to the slight excess of spins in the lower energy α state the net magnetisation vector lies along the positive B_0 direction (b).

It has already been noted that for a static observer, a given spin system precesses about the B_0 axis at the Larmor frequency. This representation is known as the *laboratory frame of reference*. It is also possible to visualise the system as if the observer were also rotating about B_0 at the Larmor frequency, when the precessing spins will appear static. This is known as the *rotating frame of reference*. The advantage of the rotating frame model is that the RF field appears as a static field since this will also oscillate at the Larmor frequency assuming that it is on resonance. This will simplify the nutational motion to a simple rotation about a fixed axis. It is also useful when looking at NMR experiments, since small differences in precession frequency due to scalar coupling, relaxation and diffusion are easily represented without the complications of Larmor precession.

As previously mentioned, we can look at NMR using a vector model^{1,2}. At the initial point of an NMR experiment the magnetisation is at equilibrium, with the net magnetisation vector M_0 aligned along the direction of the static magnetic field (conventionally labelled the z axis). The RF field (B_1) interacts with M_0 , rotating the vector away from the z axis. Typically, the RF pulse is applied long enough to move the vector into the xy plane so that there is no component along the z axis. This is known as a 90° pulse and is the most common pulse used in NMR experiments, though any angle and phase may be chosen. The phase of the pulse can be interpreted as the

direction along which the B_1 field is applied. For example, if B_1 were applied along the x axis, the spins would precess about this axis for the duration of the pulse. Ideally, this would rotate the net magnetisation vector by 90° . On completion of the RF pulse the individual spins of the system will undergo free precession (this process is shown by figs.[2.1.3a and c], though it is easier to visualise using the net magnetisation vector model shown by figs[2.1.3b and d]) and the system begins to relax.

The force generated by the magnetic field B_0 on the magnetic moment is a torque given by $\mu \times B_0$ which will cause the individual spins and therefore the vector M to precess about B_0 at the Larmor frequency (eqn. 2.1.4). As the vector precesses about B_0 it induces a voltage in the receiver coil which is recorded. The voltage is measured with and without a 90° phase shift, to give real and imaginary components of the net magnetisation vector. As the net magnetisation vector precesses, the intensity of the signal observed over time along each axis will be a decaying sinusoid. These decaying time-dependent sinusoids are known as *free induction decays* (FID) and are converted from the time domain to the frequency domain spectrum by Fourier transformation³. With a single spin system (i.e., single chemical shift) such as water, the FID is relatively simple. However, in a more complicated system, such as organic molecules, there are many spin environments each with their own vector, which will precess at different frequencies due to differing amounts of electron shielding (σ) affecting the local magnetic field. Thus the FID will be extremely complicated. After Fourier transformation, the one-dimensional NMR spectrum will have peaks for each spin environment.

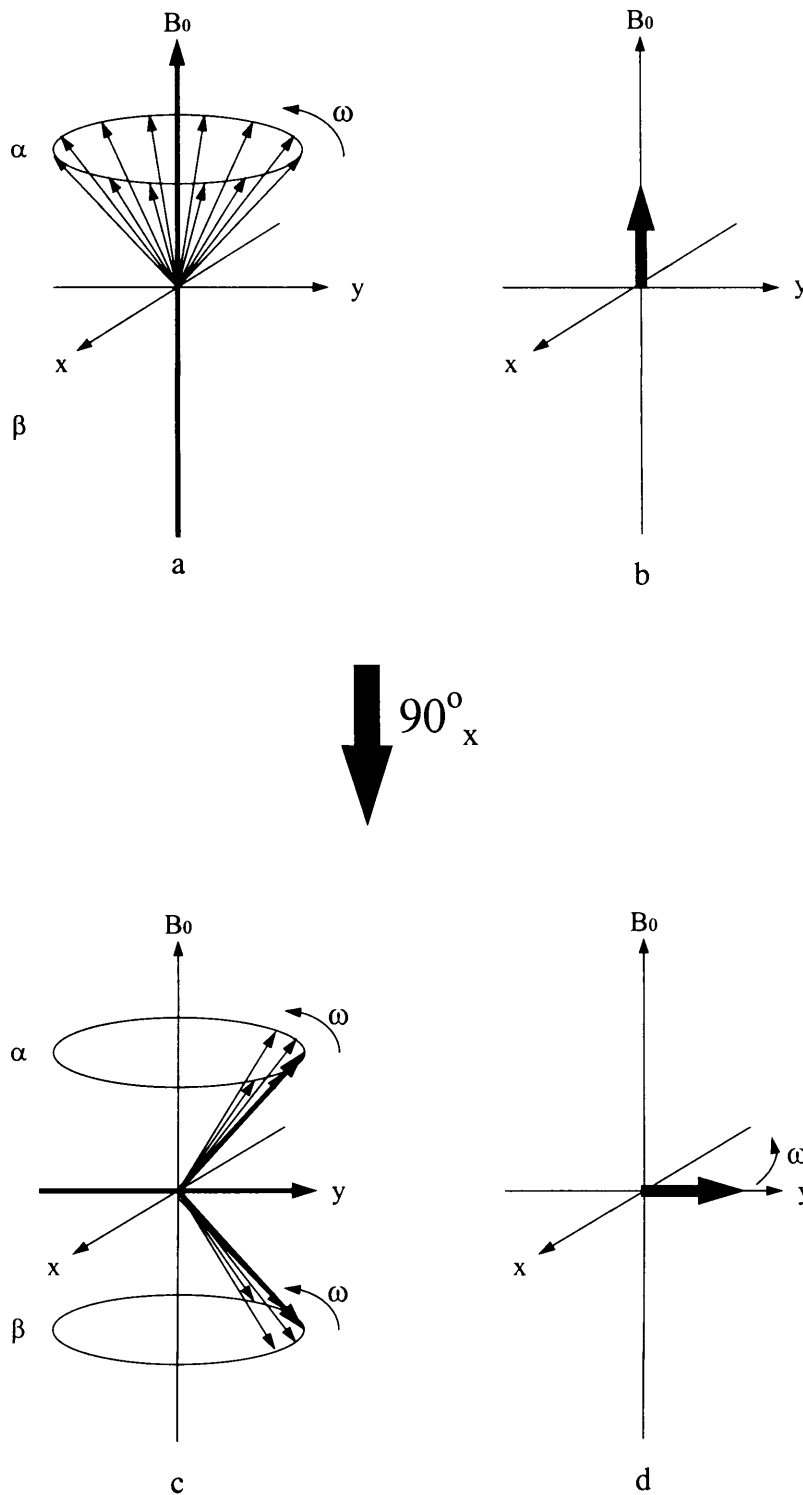


Figure 2.1. 3: The effect of the 90°_x RF pulse on the equilibrium magnetisation of the spin system is to equalise the population of spin in the α and β states and generates phase coherence in the xy plane (c). Hence, the net magnetisation vector lies in the xy plane (d). The RF radiation is pulsed with a phase along the x -axis rotating the vector about the x -axis so that the net magnetisation vector will lie along the y -axis after a 90° pulse.

During relaxation, the population of the two states will return to their Boltzmann distribution, and phase coherence is lost. As previously mentioned, the time taken after excitation of $\alpha\beta$ transitions of the system to reach equilibrium is related to the T_1 relaxation time. This process is essentially one of a change of enthalpy since spins move between energy states. As the equilibrium is approached, the z component of M increases in magnitude, finally reaching its maximum at equilibrium. The T_2 relaxation time is related to the loss of x and y components of M and is therefore affected by the same enthalpy process. This is however not the only process which affects T_2 relaxation. While magnetisation precesses about B_0 , the individual spins will dephase due to slight changes in the local magnetic field caused by dipolar interactions. This results in a loss of coherence, which in turn reduces the magnitude of the xy component of the net magnetisation vector. This entropy effect only affects T_2 relaxation and, because of this, T_2 is generally shorter than T_1 .

2.1.2 Spin Echoes.

In the previous section, the effect that an RF pulse has upon the nuclear magnetisation of a system was shown. If an RF pulse is applied for twice the length of time or at twice the power of the 90° pulse, the system experiences a 180° pulse. Here, magnetisation is rotated through 180° so that the net magnetisation vector moves from the positive direction to the negative direction of the z axis, assuming magnetisation starts from its equilibrium state along the z axis. The phases of the individual spins are distributed randomly about z in this case, unlike the ordered phase after a 90° pulse, i.e. there is no phase coherence in the xy plane.

An important NMR pulse sequence for diffusion studies, and one of the simplest, is the spin echo. The spin echo was first observed by Hahn⁴ in 1950. A pulse sequence which generates a spin echo is a 90° excitation pulse, a delay (τ), followed by a 180° refocusing pulse and a final delay, τ (fig.[2.1.4a]). A set of vector model diagrams using the rotating frame is given in fig.[2.1.5] to illustrate the process. The magnetisation dephases due to chemical shift and magnetic field inhomogeneities after the initial excitation pulse (fig[2.1.5c]). A 180° pulse is then applied. This reverses the phase of the dephased vectors by rotating the magnetisation about the y axis. The individual vectors then continue to precess at their individual frequencies but this now results in rephasing (fig[2.1.5d]). After a period 2τ , the signal refocuses, forming an echo. There is some signal loss during this experiment though this is purely because of T_2 relaxation.

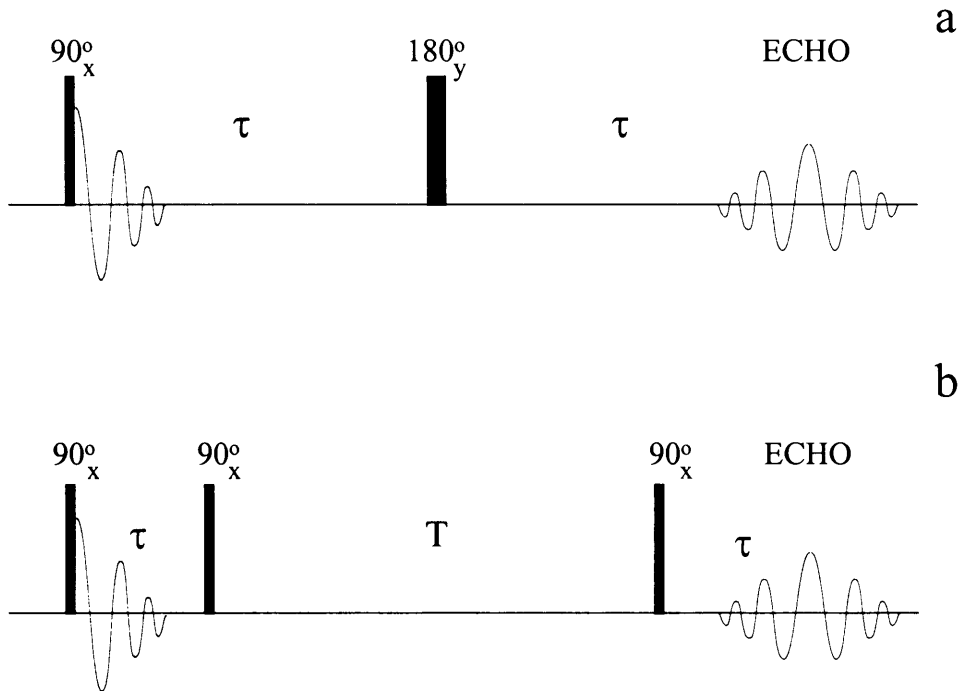


Figure 2.1. 4: (a) The spin echo experiment reverses the phase of dephasing spins by use of a 180° RF pulse to give an echo after a period 2τ . Signal is lost due to T_2 relaxation. (b) The stimulated echo experiment can be adopted when T_2 is much shorter than T_1 . Magnetisation is stored along the z axis during T . Signal is lost during this period due to T_1 relaxation but this is usually longer than T_2 . The two 90° pulses refocus the magnetisation to form the echo at a time $(2\tau+T)$ after excitation.

The spin echo experiment allows a time delay between excitation and acquisition. This is a key requirement for measuring diffusion by NMR and will be discussed further in later sections. In many diffusion studies it is preferable to use a long echo time. However, some spins have very short T_2 values, which restricts the maximum value of τ that can be used experimentally. This problem can be overcome by using *stimulated echo* pulse sequences⁴ (fig[2.1.4b]). Here, the refocusing 180° RF pulse is effectively broken down into two 90° pulses. The second 90° pulse flips the magnetisation from the xy plane back along the z axis. This is done because T_1 is usually greater than T_2 , and it is therefore possible to “store” the coherence along z for a relatively long time (T) with little signal loss due to relaxation. This therefore fulfils the need for relatively long periods between the excitation and the echo required in some diffusion studies. The final 90° pulse flips the magnetisation once more into the xy plane, where the signal refocuses and an echo is formed at a time $(2\tau+T)$ in a similar manner to the spin echo.

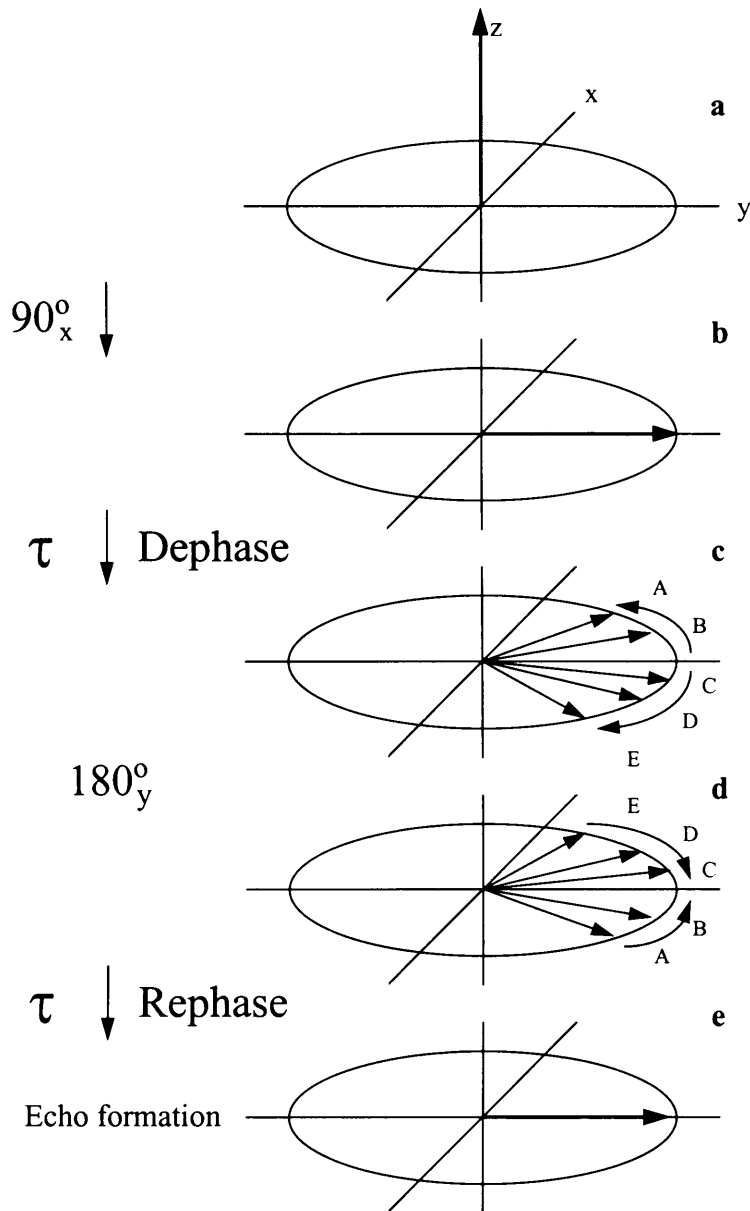


Figure 2.1. 5: A vector model diagram in the rotating frame following the course of a spin echo experiment. Signal dephases due to magnetic field inhomogeneities and chemical shift (c). The 180° pulse effectively reverses the order of the dephasing vectors (d) so that after a period twice that between the 90° and the 180° pulses, the signal refocuses (e). Only T_2 relaxation contributes to signal attenuation.

2.1.3 Product Operators.

In all Fourier Transform NMR experiments from a simple pulse-acquire sequence to a more complicated echo based experiment, the information is observed in the time domain. During these pulse sequences, magnetisation will dephase at a frequency close to the Larmor frequency. with multi-spin systems, there are a number of net magnetisation vectors. For example, ethanol ($\text{CH}_3\text{CH}_2\text{OH}$) which has protons in three different electronic environments will give rise to three vectors (assuming scalar coupling is ignored), whereas, water (H_2O) with its single proton environment only has one. This behaviour of spin systems is known as *evolution*. To explain the sometimes extremely complicated evolution of spin systems in NMR pulse sequences, especially in multidimensional and multiple-quantum NMR experiments, the vector model has limitations. The general method for overcoming these shortcomings is to use the density matrix. This, however, can become extremely involved mathematically, especially with systems of multiple spins and multiple pulses. Between these two approaches lies product operators⁵, which are simpler to use and easier to interpret than the density matrix, but without the limitations of the vector model. Product operators are based on the decomposition of the density operator into a linear combination of products of single spin angular momentum operators⁶. The fate of the spins under various operations can therefore be followed algebraically throughout a complex series of pulses and free precession periods, allowing a detailed appreciation of the state of a spin system during evolution.

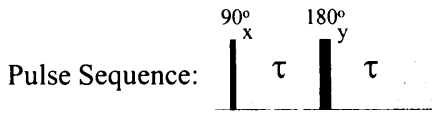
For the sake of simplicity and relevance to this thesis, only single spin operators are described below:

I_{kz} = longitudinal magnetisation of spin k

I_{kx} = in-phase x-magnetisation of spin k

I_{ky} = in-phase y-magnetisation of spin k .

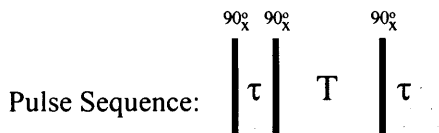
I_{kz} represents the magnetisation along the z axis of the rotating frame as would be found at thermal equilibrium. I_{kx} and I_{ky} correspond to the x and y components of the magnetisation of spin k in the rotating frame. As mentioned, product operators are used to get an appreciation of what is happening to spins in a system at any point in an experiment. For example, product operators may be used to show that no phase coherence is lost due to chemical shift evolution and magnetic field inhomogeneity (ignoring diffusion) during a spin echo.



$$I_{kz} \xrightarrow{90^\circ_x} -I_{ky} \xrightarrow{\tau} \begin{matrix} -I_{ky} \cos \omega \tau \\ +I_{kx} \sin \omega \tau \end{matrix} \xrightarrow{180^\circ_y} \begin{matrix} -I_{ky} \cos \omega \tau \\ -I_{kx} \sin \omega \tau \end{matrix}$$

$$\xrightarrow{\tau} \begin{matrix} -I_{ky} \cos^2 \omega \tau + I_{kx} \cos \omega \tau \sin \omega \tau \\ -I_{kx} \sin \omega \tau \cos \omega \tau - I_{ky} \sin^2 \omega \tau \end{matrix} \equiv -I_{ky}$$

All of the phase accumulation terms cancel out to leave I_{ky} at echo formation after a time 2τ . Another example using product operators is given below. Here, the stimulated echo pulse sequence is shown. From the operator analysis it is possible to see that half of the magnetisation is always lost with stimulated echo based NMR experiments.



$$I_{kz} \xrightarrow{90^\circ_x} -I_{ky} \xrightarrow{\tau} \begin{matrix} -I_{ky} \cos \omega \tau \\ +I_{kx} \sin \omega \tau \end{matrix} \xrightarrow{90^\circ_x} \begin{matrix} -I_{kz} \cos \omega \tau \\ (+I_{kx} \sin \omega \tau) \end{matrix} \xrightarrow{T} -I_{kz} \cos \omega \tau$$

This term will dephase and therefore disappear

$$\xrightarrow{90^\circ_x} I_{ky} \cos \omega \tau \xrightarrow{\tau} I_{ky} \cos^2 \omega \tau - I_{kx} \cos \omega \tau \sin \omega \tau \equiv \frac{1}{2} (I_{ky})$$

| | |
|---|--|
| $\cos^2 \omega + \sin^2 \omega = 1$ $\cos^2 \omega - \sin^2 \omega = \cos 2\omega$ $\cos \omega \cdot \sin \omega = 1/2 \sin 2\omega$ | $\cdot \cdot$ $\cos \omega \tau \sin \omega \tau = 1/2 \sin 2\omega \tau$ $\cos^2 \omega \tau = 1/2 (\cos 2\omega \tau + 1)$ |
|---|--|

The final product operator term contains two parts; a $\cos^2(\omega\tau)$ term and a $\cos(\omega\tau)\sin(\omega\tau)$ term. These should be rearranged in terms of 2τ , the time spent in the xy plane before echo formation. The value of the phase ($\omega\tau$) at echo formation is zero as coherence loss due to chemical shift and

field inhomogeneities is refocused. Thus the $\cos(\omega\tau)\sin(\omega\tau)$ term becomes zero and the $\cos^2(\omega\tau)$ term becomes $1/2(I_{ky})$ as described by the trigonometric rules shown in the box. Hence, a stimulated echo based experiments will lose half of the signal over its course.

2.1.4 Coherence Transfer and Phase Cycling.

We have seen how, in an NMR experiment, magnetisation can be manipulated by choosing the number, angle and phase of RF pulses in a pulse sequence. The examples given in previous sections make the assumption that the RF pulses rotate all of the spins by exactly the same, desired amount. In reality, this is not the case as the excitation of the sample by the RF radiation is always slightly inhomogeneous. Even with the most accurate RF coils exciting very small samples some of the spins will experience a rotation greater, and some a rotation smaller than that desired. Therefore, when an NMR experiment is performed, it is necessary that the pulses are calibrated to be as near to the required angle of rotation as possible. This calibration will minimise, though not avoid this range in the angle of rotation. For example, after a 90° excitation pulse M will lie in the xy plane with no z component to the vector, yet, many of the individual spins will have some small z component in their own individual vectors. This is analogous to some of the magnetisation experiencing a 180° or 0° RF pulse for over and under-rotation respectively. If this system were then subjected to further 90° pulse the spin system becomes more complicated as those spins with both xy and z components will undergo rotation into the z axis and xy plane for each component respectively. The result is that magnetisation will end up with different *phases* at the end of the pulse sequence, the majority of the magnetisation having followed the requisite pathway, but the remaining magnetisation will not; as a consequence of this may give rise to spurious data. These different routes that the magnetisation may follow are known as *coherence transfer pathways*^{7,8} and it is the implementation of a *Phase Cycle*⁹ that selects only the magnetisation from the required coherence transfer pathway whilst cancelling out the unwanted magnetisation from the other pathways. This is achieved by specific selection of the phase of the RF radiation for each pulse of the pulse sequence. To be able to do this it is necessary to understand how the phase of the magnetisation changes with the application of the RF pulse and its phase. This is described in the equation below,

$$\text{Change in the phase of magnetisation} = \Phi\Delta P \quad [2.1.5]$$

where, Φ is the change in pulse phase and ΔP is the change in coherence order. For a 90° excitation pulse $\Delta P = 1$ whilst for a 180° refocusing pulse $\Delta P = 2$. If no magnetisation is excited by the RF pulse, there will be no change in coherence order, i.e. $\Delta P = 0$.

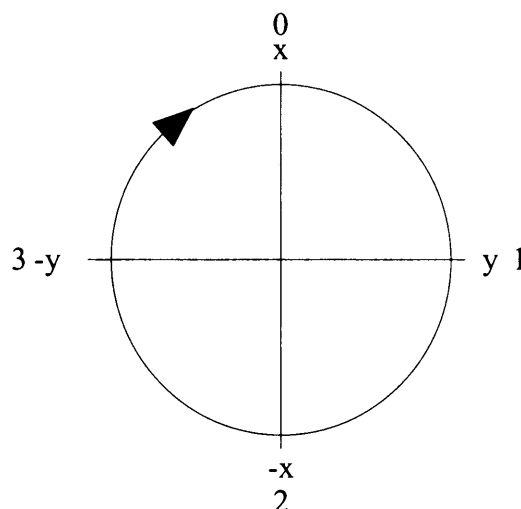


Figure 2.1. 6: A basic model describing conventional ways of labelling RF pulse phases. These are used when designing and implementing phase cycles. The phase of the RF pulse applied (x,y,-x,and -y) have a Φ value of 0,1,2,and 3 respectively.

Phase cycles work by altering the phases of the RF pulses. For instance, a 90°_x pulse will rotate the magnetisation about the x axis to lie along the y axis (though, as previously revealed, some magnetisation will also exist in the z axis). If, after a period long enough for thermal equilibrium to once more become established, the experiment is repeated this time with a 90°_{-x} pulse, the magnetisation will be rotated to lie along the -y axis though again there will be a z axis contribution. If the signal acquired from this experiment is then subtracted from that of the initial experiment any magnetisation in the z axis will cancel out whilst the desired magnetisation - that along the $\pm y$ axis is accumulated. This is the basic principle behind phase cycling, though as experiments become more complicated, with more pulses, so do the associated phase cycles. These are generally however composites of phase cycles for individual 90° and 180° pulses. The phase conventions used in phase cycles are shown in fig.[2.1.6]. The figure is labelled with two classifications for the phase of a pulse. As already mentioned, for any given NMR experiment there may be several coherence transfer pathways though we only usually wish to observe one. To choose the desired pathway and subsequently design our pulse sequence it is necessary to break down the experiment into each individual pulses. To phase cycle any pulse we need to know the change in coherence order that brings about the desired coherence transfer pathway. The coherence

order at thermal equilibrium is zero. For a 90° pulse the desired change in coherence order (ΔP) is ± 1 and for a 180° pulse $\Delta P = \pm 2$. A change in coherence order of zero, indicates that these spins are unaffected by the RF radiation. Knowing this it is now possible to govern the phases of the pulse and the receiver to observe only the desired coherence transfer. The following scheme in table[2.1.1] shows how a phase cycle is calculated and chosen for a 180° pulse (i.e. isolating $\Delta P = \pm 2$).

| Change in the phase of the pulse | Change in the phase of magnetisation ($\Delta P = \pm 2$) | Change in the phase of magnetisation ($\Delta P = 0$) ($\Delta P = \pm 1$) | |
|----------------------------------|--|--|-----------------|
| | Wanted Magnetisation | Unwanted Magnetisation | |
| x (0) | 0 \uparrow | 0 \uparrow | 0 \uparrow |
| y (1) | 2 \downarrow | 0 \uparrow | 1 \rightarrow |
| -x (2) | 0 \uparrow | 0 \uparrow | 2 \downarrow |
| -y (3) | 2 \downarrow | 0 \uparrow | 3 \leftarrow |

Table 2.1. 1: Changes in the phase of magnetisation for a 180° pulse. The arrows indicate the phase of the receiver (the phase of the magnetisation at acquisition) in the same manner described in fig[2.1.6].

The change in magnetisation phase is the product of the change in the phases of the pulse and the change in coherence order of the magnetisation as expressed in eqn.[2.1.5]. The phase cycle for the 180° pulse is simply 0,1,2,3, however, to remove the unwanted magnetisation the receiver pulse has a cycle of 0,2,0,2, thus negating any effect from unwanted magnetisation as this will result in a net sum of zero. The case is very similar for a 90° pulse, as shown below in table [2.1.2].

| Change in the phase of the pulse | Change in the phase of magnetisation ($\Delta P = \pm 1$) | Change in the phase of magnetisation ($\Delta P = 0$) ($\Delta P = \pm 2$) | |
|----------------------------------|--|--|---|
| | Wanted Magnetisation | Unwanted Magnetisation | |
| x (0) | 0 | 0 | 0 |
| y (1) | 1,3 | 0 | 2 |
| -x (2) | 2 | 0 | 0 |
| -y (3) | 3,1 | 0 | 2 |

Table 2.1. 2: Changes in the phase of magnetisation for a 90° pulse.

Here, the receiver would read the same as the change in phase of magnetisation for the wanted magnetisation 0,1,2,3, and the unwanted magnetisation would cancel out. However, this is not the only phase cycle one might choose to apply to this pulse. Since the phases of the unwanted magnetisation are the same when $\Phi = 0$ and 2, and when $\Phi = 1$ and 3, whilst the phases of the

desired magnetisation are opposite, it is possible to use a shortened phase cycle here of 0,2 or 1,3. The result will accumulate the desired signal and cancel out that which is not desired yet halve the experimental time as only two transients are required. For this practical reason many of the phase cycle adopted in this thesis use this shortened cycle. In the examples given, the phase of the receiver is the same as that of the pulse since they are effectively only single pulse sequences. In more complicated multi-pulse sequences, the receiver is the logical combination of the phases of the RF pulses. For example, a 90_x° pulse followed by a 180_y° pulse will result in the desired magnetisation having a phase of x (or zero) upon acquisition and therefore the receiver should be set to x for this transient. It is not always necessary to phase cycle a pulse at all and in the majority of cases only a partial phase cycle is adopted for a sequence.

2.2 Diffusion.

At all temperatures above absolute zero, molecules possess various amounts of translational energy. In liquids or solutions these molecules move very rapidly relative to solids. Each molecule has its own momentum and direction and as they collide with one another, momenta are exchanged and direction is liable to be altered. Due to the large numbers of molecules in close proximity, the collisions are extremely frequent and the path of the molecules is purely random. This is the basic phenomenon of diffusion (analogous to that of *Brownian Motion*). There is a number of factors that affect this random motion and the rate at which the molecules move such as: molecular weight, temperature, hydrogen-bonding and Van der Waals forces. These rates are known as *Diffusion Coefficients* and will be discussed later. In the case of solutions, concentration is a further factor that should be taken into account. For work described in this thesis, it has been assumed that all solutes are evenly distributed and therefore, no concentration gradients exist.

It should be made clear at this point that when diffusion is referred to in this thesis it is actually describing the process of *self-diffusion*; the random motion of molecules within their bulk phase. In the majority of studies presented in this thesis it is the diffusion of water molecules within water.

In 1905, Einstein¹⁰ published a paper in which he rationally explained the phenomenon of Brownian motion. He described the motion in terms of a random walk process and derived an equation that relates the mean square distance a particle/molecule diffuses to the observation time. This equation is given below in its three dimensional form:

$$\langle r^2(t) \rangle = 6Dt \quad [2.2.1]$$

where D = self-diffusion coefficient, t = time, and r = displacement. The angular brackets indicate that the term is an ensemble average. It is often preferable to treat diffusion in each of the three dimensions separately. Eqn.[2.2.1] can be amended to show the one dimensional case (eqn.[2.2.2]) where only the diffusion distance along the x axis is considered despite there being two or three-dimensional diffusion:

$$\langle x^2(t) \rangle = 2Dt \quad [2.2.2]$$

The molecules are free to move in both the positive and negative directions. From this it is possible to calculate the probability of finding a molecule at a particular position at a given time. For a one dimensional random walk it is comparable to flipping a coin to see if each move is in the positive or negative direction and then recording the number of occasions the molecule is present at

any given point. For free diffusion, this will always result in a normal (or Gaussian) distribution about its starting position, which is given by the equation:

$$p(x,t) = \frac{1}{\sqrt{2\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right) \quad [2.2.3]$$

where x is the distance travelled in a time t . Both the time and the diffusion coefficient will affect the width of the distribution. For example, the effect of a short time with a large diffusion coefficient is the same as that of a small diffusion coefficient at a longer time. At short times, the probability of zero displacement is high and the distribution is narrow. Increasing the time allows greater spin diffusion and gives rise to an increase in the average magnitude of the distance travelled and consequently the width of the distribution. This concept is represented in fig.[2.2.1]. The normal distributions are symmetrical so that the average molecule displacement is zero although from Einstein's equation we have seen that the mean square displacement is non-zero.

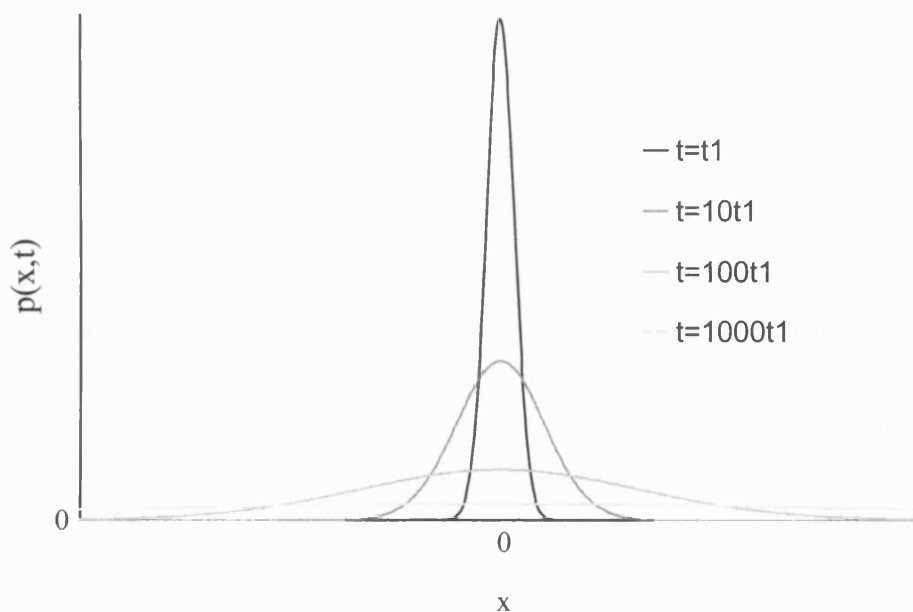


Figure 2.2. 1: Probability distribution for the location of a spin after various times. The shape of the distribution is normal or Gaussian, where the mean displacement is zero.

2.2.1 Restricted Diffusion

There are, however, situations where the diffusing molecules encounter barriers, be they cell walls or membranes, or pore walls in porous rock. These will affect the manner in which the molecules are able to diffuse. The two main types of interaction with the barriers are fully reflective restriction where the molecule can only rebound from the barrier, and restriction by

permeable barriers. In this latter case, the molecule may either be reflected from or pass through the permeable barrier. The ratio between these two possibilities depends on the permeability of the membrane.

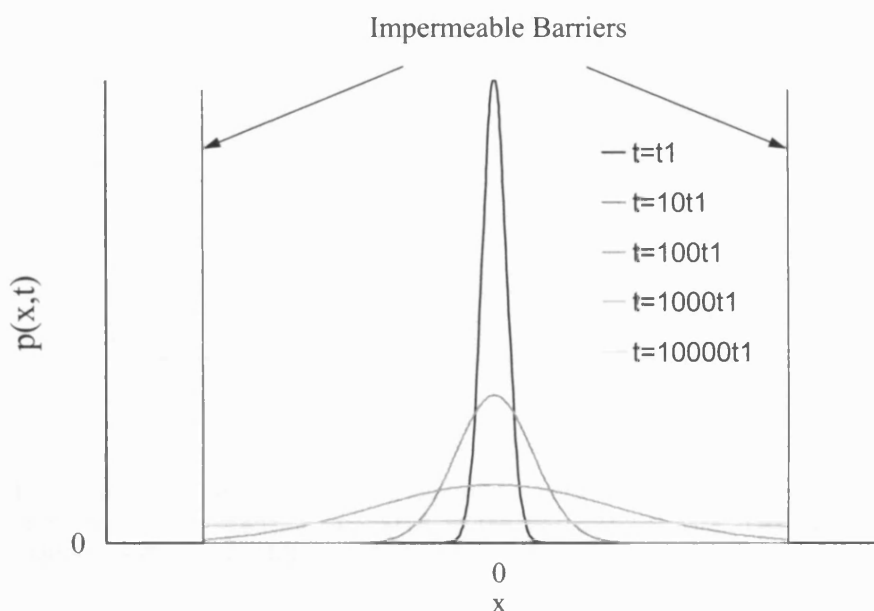


Figure 2.2. 2: The probability distribution for a fully restricted system. The probability of spin presence beyond the barrier is zero.

In the situation where a molecule is always reflected, the probability of zero displacement at long times is greater than that of the normal unrestricted case since the probability of a spin being beyond the barrier is zero, as shown in fig.[2.2.2]. This in turn affects the measured value of the self-diffusion coefficient (SDC) which deviates from its true value. In this situation the SDC is referred to as an *apparent diffusion coefficient* (ADC). This variation in the value of the coefficient is due to spins being reflected by the confining barriers which, therefore, have not moved as far away from their initial point as expected. It appears that the ADC has a lower value than anticipated. As time increases, more and more spins will become reflected resulting in lower ADCs until a point as illustrated in fig.[2.2.2] at $t=10000t_1$, where the probability of a spins presence is equal throughout the system and the ADC tends towards zero. It is the observation of these ADCs which form the basis of all restricted diffusion studies using NMR. In the case where the restricting barrier is permeable, the probability distributions are different again. Here, the molecules may pass through the barrier and therefore the width of distribution is infinite at infinite time as with the free diffusion. However, as fig.[2.2.3] shows, there is a step between the probabilities of molecule presence immediately on either side of the barrier. It is this step which is related to the permeability constant of the barrier (fig.[2.2.4]).

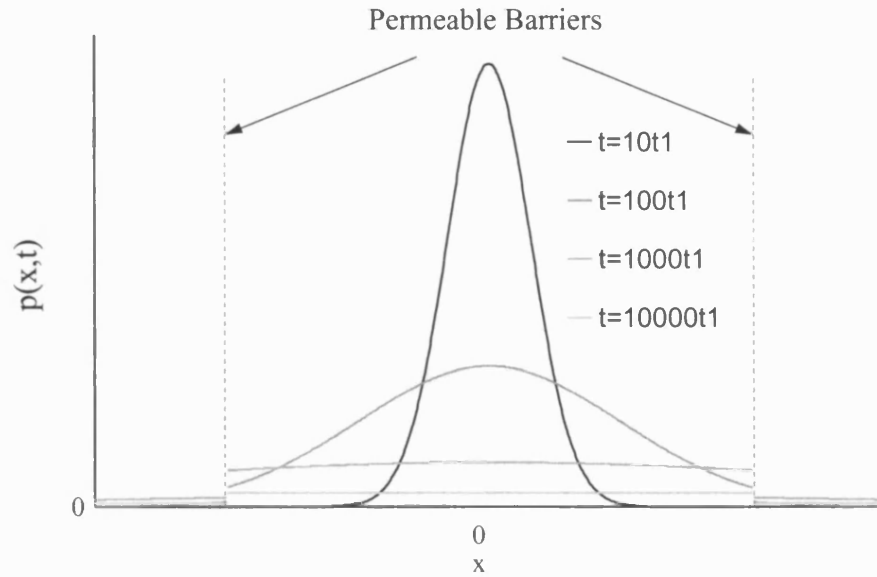


Figure 2.2. 3: The probability distribution for a partially restricted system. The probability of spin presence beyond the barrier is reduced compared to the unrestricted case. The magnitude of this reduction is dependent on the permeability of the barrier.

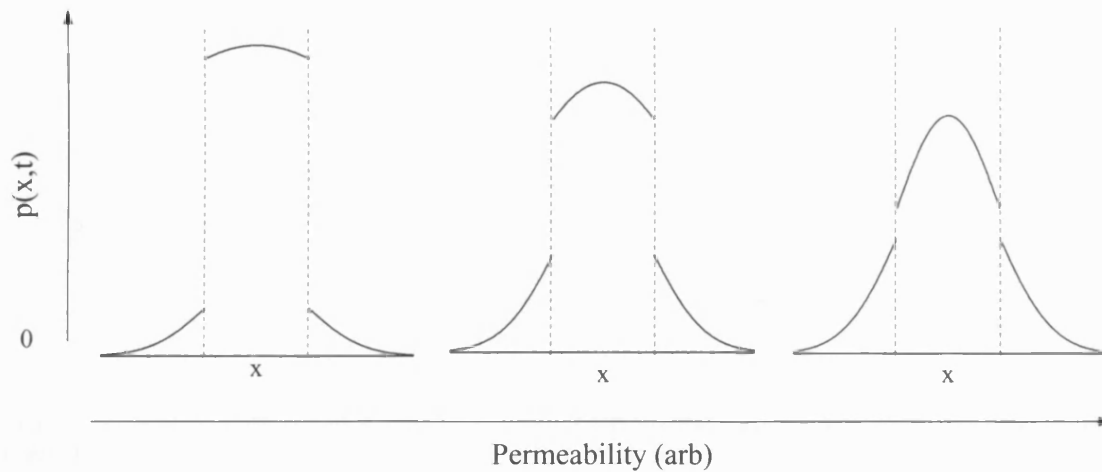


Figure 2.2. 4: Illustration of how the permeability affects the probability distributions which in turn will be reflected in the ADC values at different diffusion times.

2.2.2 Measuring Diffusion by NMR

The key to measuring diffusion by NMR techniques is the application of a magnetic field gradient (MFG) across the sample. It has already been stated that the frequency with which spins precess about B_0 in a uniform field is given by:

$$\omega = \gamma B_0. \quad [2.2.4]$$

However, as illustrated in fig.[2.2.5], when a magnetic field gradient is applied across the system the local magnetic field varies along the axis of the gradient according to the revised Larmor equation

$$\omega = \gamma(B_0 + xG_x). \quad [2.2.4]$$

The strength of the magnetic field gradient is given by G_x and the position along the axis of the gradient is given by x . Thus, identical spins at different locations along the gradient axis will experience different magnetic fields and therefore precess at different frequencies. Therefore, as the spins diffuse and change their position along the direction of the gradient and encounter a different magnetic field strength, their precessional frequency will change. It is this phenomenon which is usually exploited when studying diffusion by NMR^{11,12} or performing MRI^{13,14}.

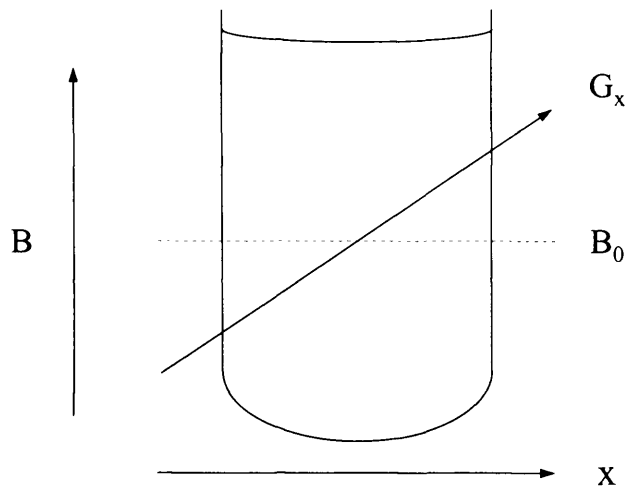


Figure 2.2. 5: The presence of a linear magnetic field gradient across a sample will affect the local magnetic field experienced as a function of position along the gradient axis.

The spin echo experiment (section 2.1.2) refocuses evolution due to field inhomogeneities, chemical shift and scalar coupling. After the 90° excitation pulse, the magnetisation begins to dephase, with each spin precessing at a specific frequency. After the 180° pulse, the orientation of the dephasing spins is reversed, though the frequency with which they precess remains the same. The result is that after second period τ , they return to the point from which they began and an echo forms. Signal is lost only due to T_2 relaxation. If, however, a magnetic field gradient is applied across the sample for the duration of the experiment, the frequency with which the spin vectors precess during the dephasing period will not always be the same as the frequency with which they precess during the rephasing period. Nor will it be constant during the dephasing and rephasing periods. This is due to random diffusion of the spins which will alter their position and therefore their precession frequency. Some of the rephasing spins will over or undershoot their starting point, resulting in incomplete refocusing on echo formation. By repeating the experiment several times with different gradient strengths and in some experiments, diffusion times, it is possible to calculate the SDC of the sample.

The most common sequence for measuring diffusion is the *pulsed gradient spin echo* (PGSE) experiment^{15, 16, 17, 18}. Here, magnetic field gradients are pulsed during the dephasing and rephasing periods causing signal loss due to diffusion. By varying the gradient strength G or the pulse duration δ , the attenuation due to diffusion in each experiment can be changed, whilst the time allowed for diffusion (and hence T_2 relaxation) is kept constant. The time between the beginning of dephasing to the beginning of rephasing in fig.[2.2.6] is labelled Δ . In the ideal case, where the gradient pulses are infinitely short, Δ is equal to the diffusion time. In reality, the gradient pulses have finite widths so there is in effect an ensemble of many different diffusion times. This experimental case addressed in the Stejskal-Tanner equation (eqn.[2.2.5]) where an “average” value for the diffusion time is defined as $\Delta\delta/3$. For calculations the Stejskal-Tanner definition will be adopted whilst in the text, diffusion time will take its true meaning. The relationship between signal intensity and gradient strength for the PGSE experiment is shown below in eqn.[2.2.5]:

$$\frac{M}{M_0} = \exp\left(-\frac{2\tau}{T_2} + \gamma^2 G^2 D \delta^2 \left(\Delta - \frac{\delta}{3}\right)\right). \quad [2.2.5]$$

A plot of $\ln(M/M_0)$ against G^2 gives rise to a linear slope equal to $-\gamma^2 \delta^2 (\Delta - \delta/3) D$ from which the SDC can be extracted. However, this technique has limitations when looking at restricted diffusion in a single experiment, as the diffusion time must be kept constant to avoid introducing T_2 attenuation.

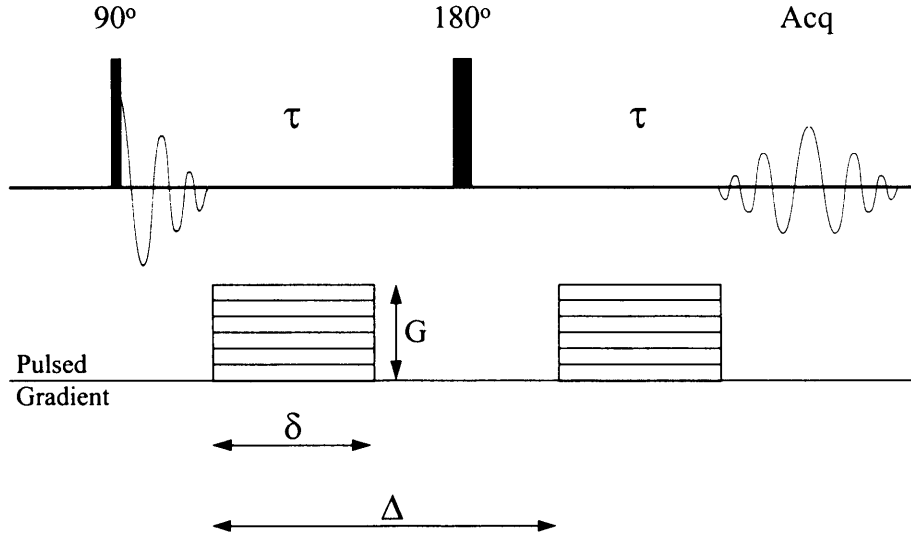


Figure 2.2. 6: The pulsed gradient spin echo (PGSE) experiment. Phase accumulated by the magnetisation during the first gradient pulse is not fully refocused by the second gradient pulse due to diffusion, resulting in incomplete refocusing of the magnetisation.

It was stated earlier that as the diffusion time increases in a restricted environment, the number of spins which encounter the restricting barrier also increases and this lowers the ADC. In a constant time experiment such as PGSE it is not possible to monitor the change in ADC in one experiment. The problem is approached by repeating several series of PGSE experiments with different diffusion times, then producing a plot of ADC against diffusion time from which it is possible to analyse restriction effects.

An alternative to measuring diffusion by varying the gradient strength is to vary the time parameters; this also allows more scope for studying restriction. The problem in doing this is that the attenuation due to relaxation will vary and lead to complications and inaccuracies in calculating the ADC. It is therefore necessary to design a pulse sequence so that the total times that the coherent magnetisation spends in the xy plane and along the z axis are constant. One such approach is the *constant time, pulse, gradient (CTPG)*¹⁹ type of experiments, where a constant gradient is applied in a pulse sequence designed so that by varying certain delays incrementally, an ADC measurement can be made. The basic CTPG sequence has two spin echo sequences back to back with a variable balance of delays. The time between excitation and detection for the sequence is constant and equal to $2(\tau_1 + \tau_2)$. However, on inspection of the appropriate attenuation equation

(eqn.[2.2.6]), it can be seen that the signal attenuation is dependant on the term $(\tau_1^3 + \tau_2^3)$ which will vary in magnitude depending on the ratio of τ_1 and τ_2 assuming that the static gradient is constant throughout.

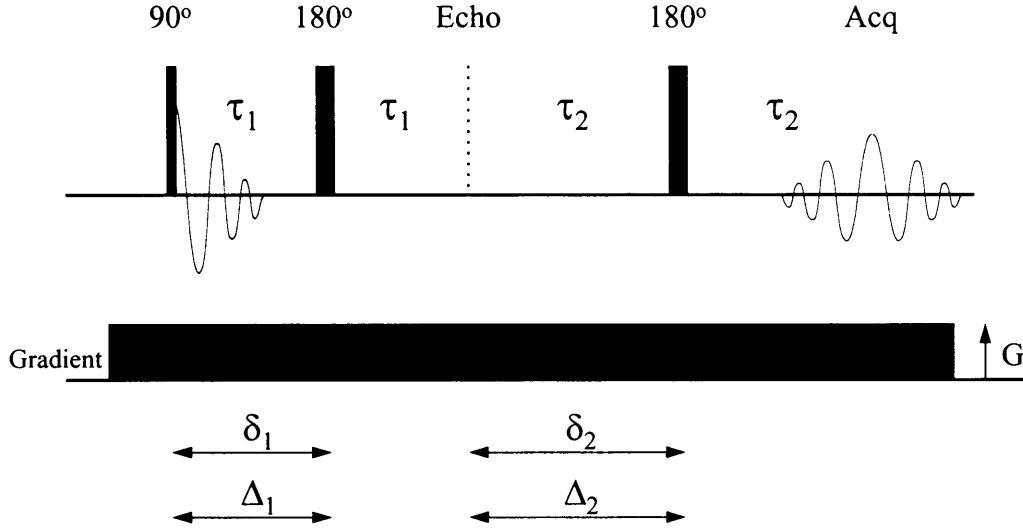


Figure 2.2. 7: The constant time and pulsed gradient (CTPG) experiment. Varying the ratio of τ_1 and τ_2 whilst keeping their sum constant, results in different amounts of diffusion attenuation but constant relaxation. From this the ADC may be calculated.

$$\frac{M}{M_0} = \exp\left(-\left(\frac{2(\tau_1 + \tau_2)}{T_2} + \frac{2}{3}\gamma^2 G^2 D(\tau_1^3 + \tau_2^3)\right)\right) \quad [2.2.6]$$

It is possible to see the two separate periods in which the diffusion attenuates the signal the CTPG experiment in fig.[2.2.7]. The applied magnetic field gradient is present throughout the experiment so phase is accumulated throughout. The two half echo times τ_1 and τ_2 are equivalent to the durations of the gradient pulses in the Stejskal Tanner which are in turn approximate to the two diffusion times of the sequence, Δ_1 and Δ_2 . This sequence is, however, limited when applied to studies of restricted systems since the sum of the two diffusion times is always constant. Also, for systems with short T_2 values, the overall time the magnetisation spends in the x-y plane must be short giving little scope for great variation in $(\tau_1^3 + \tau_2^3)$.

A notable adaptation to the CTPG sequence is the *one echo constant time and pulsed gradient* (OE-CTPG) experiment²⁰. Here, there is only a single diffusion period and one echo as the

name suggests. The sequence gives a stimulated echo, with an extra period attached to the end of the sequence (fig.[2.2.8]). Here the lengths of the effective gradient pulses τ are constant and, therefore, the time the net magnetisation vector spends in the x-y plane is constant, giving constant attenuation from transverse relaxation. Yet, the diffusion time may be varied by changing the value of the delay t . The time the magnetisation vector spends along the z axis is always fixed (T) and hence longitudinal relaxation is also constant. The attenuation equation for OE-CTPG is given below.

$$\frac{M}{M_0} = \frac{1}{2} \exp - \left(\frac{2\tau}{T_2} + \frac{T}{T_1} + \gamma^2 G^2 D \left(\tau^2 \left(t + \frac{2}{3} \tau \right) \right) \right) \quad [2.2.7]$$

The advantage OE-CTPG has over the CTPG experiment, and indeed PGSE, is that the total diffusion time can be varied for one experiment. It is therefore an ideal pulse sequence to use when studying restricted diffusion by NMR. When using a PGSE type of experiment for such studies, one experiment is required to calculate the ADC at each diffusion time. However, in the OE-CTPG experiment each measurement is made at a different diffusion time. A semi-log plot of OE-CTPG data will give rise to a straight line in an unrestricted environment, where the derivative of the plot is proportional to the ADC. More interestingly, in a restricted environment the slope will change with increasing diffusion time directly indicating the presence of restriction.

The major benefit constant gradient experiments offer over pulsed gradient diffusion experiments concerns eddy currents. These are fluctuations in the magnetic field induced when switching magnetic field gradients on and off. Any gradient pulse may give rise to eddy currents which can last for up to a few seconds and, therefore, are a concern in diffusion studies since they give rise to extra signal attenuation. This extra attenuation will vary between the different amplitudes of a PGSE experiment since the eddy currents are directly proportional to the gradient strength. This causes inaccurate estimates of the ADC. These problems may be reduced by using shielded gradients although the suppression of the eddy currents is not complete. A constant gradient experiment can avoid these problems by switching the gradient on well in advance of the initial excitation pulse so that the eddy currents have died away for the experiment. Others have used a pulse sequence similar to OE-CTPG with pulsed gradients²¹ in the hope that eddy currents subside whilst the magnetisation resides along the z axis.

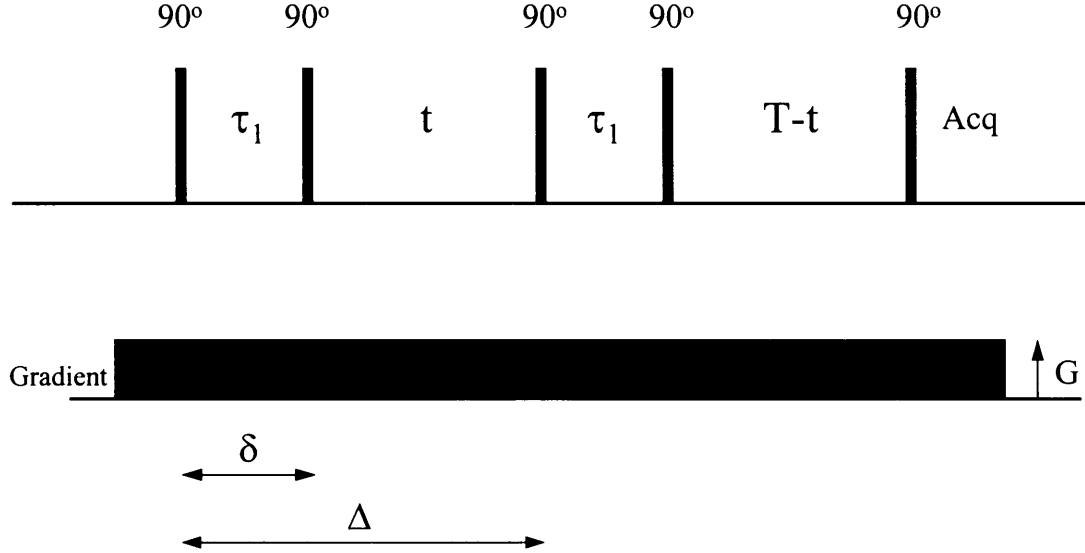


Figure 2.2. 8: The one-echo CTPG (OE-CTPG) pulse sequence. This modification allows the user to vary diffusion time whilst maintaining attenuation due to relaxation.

One point to be aware of when using constant gradient experiments is that when an RF pulse is applied in the presence of a magnetic field gradient, there is the potentially undesirable side-effect that the volume of the sample observed varies with the strength of the gradient and length of the RF pulse. This is a combination of two effects: the presence of the magnetic field gradient will cause the frequency of the spins to vary with their spatial location, and the effect of an RF pulse on nuclear spins varies with their relative frequencies. When the spins precess at the same frequency as the transmitter (i.e. *on resonance*) they will behave ideally, when they are not (i.e. *off resonance*), they behave non-ideally. The further the resonance frequency of the spins deviate from the transmitter frequency, the less ideally it behaves; for large offsets from resonance the RF pulse has little, or no effect. Consequently, if an RF pulse is applied in the presence of a magnetic field gradient, only a region of the sample in the direction of the gradient may be perturbed; as the strength of the gradient is increased so the volume of the sample affected will decrease. Whilst this effect is useful for *slice selection* in magnetic resonance imaging, in the present context it may cause unwanted variations in the signal intensity, which should only change due to diffusion. Consequently, variable strength gradients should only be applied between RF pulses. In the above examples and other constant gradient pulse sequences shown in this thesis, this effect is negated by

using the same number of identical RF pulses in each experiment of a set. Thus, the excited volume is constant throughout all comparative experiments.

The majority of the work presented in this thesis uses these constant gradient techniques. The main advantage of these sequences is that they can be used on any NMR spectrometer, irrespective of whether or not it has shielded gradients, since shims coils can be heavily mis-set to produce a static linear gradient across the sample.

-
- ¹ F. Bloch, W.W. Hansen and M. Packard. *Phys.Rev.* **70**, 474 (1946).
- ² E. M. Purcell, H. C. Torrey and R. V. Pound. *Phys. Rev.* **69**, 127 (1946).
- ³ R. R. Ernst and W. A. Anderson. *Phys. Sci. Instrum.* **37**, 93 (1966).
- ⁴ E. L. Hahn, *Phys. Rev.* **80**, 580 (1950).
- ⁵ O. W. Sørensen, G. W. Eich, M. H. Levitt, G. Bodenhausen and R. Ernst. *Progr. Nucl. Magn. Reson. Spectrosc.*, **16**, 163 (1983).
- ⁶ U. Fano. *Rev. Mod. Phys.*, **29**, 74 (1957).
- ⁷ R. R. Ernst, G. Bodenhausen and A. Wokaun. *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Oxford University Press (1987).
- ⁸ G. Bodenhausen, H. Kogler and R. R. Ernst. *J. Magn. Reson.* **58**, 370 (1984).
- ⁹ G. Bodenhausen, R. Freeman and D. L. Turner. *J. Magn. Reson.* **27**, 511 (1977).
- ¹⁰ A. Einstein. *Ann. Phys. Lpz.*, **17**, 549 (1905).
- ¹¹ H. Y. Carr and E. M. Purcell. *Phys. Rev.*, **94**, 630 (1954).
- ¹² D. W. McCall, D. C. Douglas and E. W. Anderson. *Ber. Bunsenges. Physik. Chem.*, **67**, 336 (1963).
- ¹³ P. C. Lauterbur. *Nature*, **242**, 190 (1973).
- ¹⁴ P. Mansfield and P. K. Grannell. *Phys. Rev. B*, **12**, 3618 (1975).
- ¹⁵ E. O Stejskal and J. E. Tanner. *J. Chem. Phys.*, **42**, 288 (1965).
- ¹⁶ E. O Stejskal. *J. Chem. Phys.*, **43**, 3597 (1965).
- ¹⁷ J. E. Tanner. *Rev. Sci. Instr.*, **36**, 1086 (1965).
- ¹⁸ J. E. Tanner. *PhD Thesis.*, University of Wisconsin (1966).
- ¹⁹ T. J. Norwood and R. A. Quilter. *J. Magn. Reson.*, **97**, 99 (1992).
- ²⁰ T. J. Norwood. *J. Magn. Reson. A*, **103**, 258 (1993).
- ²¹ S. J. Gibbs and C. S. Johnson Jr. *J. Magn. Reson.*, **93**, 395 (1991).

Chapter 3

A Robust NMR Technique for Measuring the Signal Of Molecules whose Diffusion has been Restricted.

3.1 Introduction and Aim.

It has been shown that it is possible to measure ADC using both static and pulsed magnetic field gradient NMR techniques¹, and that these sequences can be sensitive to restriction within the system. However, it is not possible to use any of these techniques to look purely at signal arising from differences in restriction since there is always a contribution to the signal loss from freely diffusing, unrestricted spins. In this chapter, a new pulse sequence is discussed that has been designed and developed to isolate the restricted contribution from all other factors adding to signal loss.

3.2 Pulse Sequence Design.

For this area of research, it was decided to initially develop a constant gradient technique to allow the experiment to be applied using any NMR machine both spectrometer and imaging. Since we are using a constant gradient and looking at restricted diffusion, it is reasonable to assume that a variable diffusion time would be a major feature of the sequence and that any attenuation due to relaxation should be kept constant. This can be achieved by fixing the overall period of time spent in the xy plane and along the z-axis. To this point the criteria would be fulfilled by the OE-CTPG² pulse sequence discussed in the previous chapter. However, the aim of this new pulse sequence is to isolate signal differences arising solely from differences in restricted diffusion. To do this, it is required to maintain constant attenuation from diffusion whilst having a variable diffusion time since this forms the basis of the ability to separate signal attenuation from free and restricted spin-paths. This separation is made possible when it is considered that a freely diffusing, unrestricted sample will give rise to identical signal intensity regardless of increasing diffusion time if the sequence is appropriately designed. However, in the restricted case, increasing diffusion time will give rise to decreasing ADC values resulting in a reduction in signal attenuation. Therefore, in a

series of experiments where free diffusion gives constant signal intensity and a restricted sample gives variable signal intensity, the difference between any two intensities is solely due to differences in restriction.

A pulse sequence that incorporates all these design criteria, the *restricted diffusion constant time pulsed gradient* (RD-CTPG) experiment can be seen in fig.[3.2.1]. The sequence consists of two diffusion encoding components, where the sums of the periods magnetisation spend in the xy plane and along the z-axis are constant for a range of experimental diffusion times, in the same manner as the OE-CTPG pulse sequence.

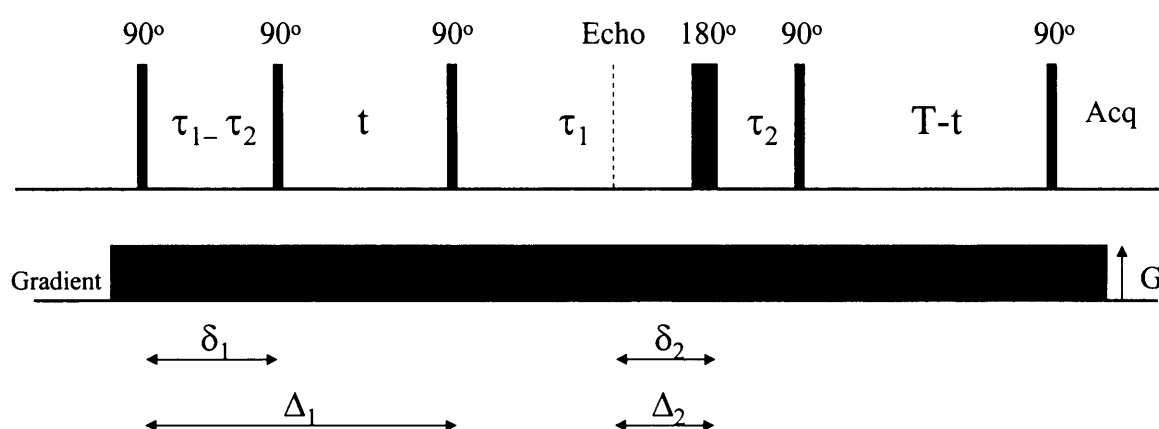


Figure 3.2.1: The RD-CTPG pulse sequence shown above consists of two diffusion-encoding modules. These are in essence a stimulated echo followed by a spin echo where the two may be balanced to produce identical attenuation from both diffusion and relaxation effects whilst the diffusion time is varied.

The first 90° excitation pulse flips the magnetisation into the xy plane where it begins to dephase. The second and third 90° R.F. pulses refocus the magnetisation, as does a 180° pulse, but stores the magnetisation along z between the pulses, in this case for a period t . As the vector is flipped back into the xy plane, the magnetisation begins to rephase, finally refocusing after a further period of $(\tau_1 - \tau_2)$ after which the signal dephases, for a period τ_2 , before being refocused by the 180° pulse. On the refocusing of the second echo the magnetisation is again flipped into the z-axis this time for a period $(T - t)$. As the MFG is present throughout the experiment, the flipping of the net magnetisation vector into the z-axis during t means that system effectively experiences the gradient for the duration of the time magnetisation is in the xy plane (i.e. $(\tau_1 - \tau_2)$, τ_1 and τ_2). By varying the period t , we can alter the diffusion time and hence the effect that restriction has on the final signal. However, it is not possible to acquire the signal yet as the time spent in the z-axis will vary, and T_1 relaxation effects will not be constant. With similar considerations in mind, for signal

attenuation from free diffusion to be constant whilst t varies, the value of τ_2 must change to compensate. Again, if this is changed, the amount of time spent in the xy plane is not constant and T_2 relaxation effects will vary. It is for this reason that there is a second diffusion-encoding module with a diffusion time and effective gradient pulse duration of τ_2 . On the formation of the second echo at a period τ_2 after the 180° RF pulse, the overall time that magnetisation spends along the z -axis is still variable, consequently it is possible to introduce another period equal to $(T-t)$ to compensate for this variation; this must always be positive, To acquire the final signal the vector is flipped into the xy plane by the last 90° pulse. The total length of the sequence is constant and equal to $(2\tau_1+T)$ where $2\tau_1$ is the fixed time that magnetisation spends in the xy plane and T is the time spent along the z -axis, thereby fulfilling the requirements for fixed contribution in attenuation due to relaxation. Since the MFG is constant, it is preferable to have a fixed number of RF pulses to maintain a constant amount of sample excited for all experiments assuming that the pulses remain of the same power and length.

As mentioned above, the pulse sequence consists of two separate diffusion encoding modules where the values of τ_2 and t can be varied thus changing the overall diffusion time whilst maintaining a constant amount of attenuation (assuming that the diffusion coefficient remains constant). This is possible as the relationship between τ_1 and t is non-linear as seen in eqn.[3.2.1].

$$\frac{A(t)}{A(0)} = \exp - \left[\frac{2\tau_1}{T_2} + \frac{T}{T_1} + (\gamma G)^2 D \left(t(\tau_1 - \tau_2)^2 + \frac{2}{3} \left((\tau_1 - \tau_2)^3 + \tau_2^3 \right) \right) \right] \quad [3.2.1]$$

The two modules can be broken down to a stimulated echo and a spin echo. For a given diffusion time the stimulated echo will give rise to a smaller loss of signal due to diffusion than the spin echo. This is due to the stimulated echo having a period where the magnetisation is stored along an orthogonal axis (the z -axis). In addition, the stimulated echo can be manipulated to give a desired amount of signal attenuation from diffusion by varying the length of the effective gradient pulses. Therefore, when a system is restricted the value of the ADC (denoted as D in the equation) will decrease with increasing diffusion time to a point where all the spins are reflected by the barriers and the probability of spin position is equal throughout the sample. Therefore, using eqn.[3.2.1] we can generate a series of experimental parameters to produce a succession of experiments in which the only difference in signal intensity is due to restriction.

3.3 Theoretical Analysis.

Since the theory predicts a decrease in signal attenuation as a result of an increase in the number of spins that are restricted and the degree to which they are restricted³, it is reasonable to

assume that to a certain extent, the greater the restriction, the less the signal attenuation. This suggests a relationship between the signal intensity variation, restriction size and diffusion time.

In this section two theoretical models from the literature are adapted for application to the RD-CTPG pulse sequence. The aim was to investigate how signal intensity varies with barrier spacing for a pair of diffusion times. If these distributions were different, it may be possible to select specific ranges of restriction sizes by the precise choice of the diffusion times used in the two parameter sets required for the difference experiment. To test this hypothesis the models were applied to the RD-CTPG pulse sequence as described below. Data is simulated to predict what changes may occur.

3.3.1 Model 1 (Von Meerwall and Ferguson).

The first model for predicting barrier separation from diffusion data is taken from a paper by von Meerwall and Fergusson⁴ in which a PGSE method was adopted to observe the variation in restriction over diffusion time. Below, the original equation, eqn.[3.3.1], is recast to make it applicable for applicability to the RD-CTPG pulse sequence.

$$R_{(t)} = \exp\left[-\frac{\theta^2 D \Delta}{a^2}(\sin^2 \alpha + A)\right] \frac{2}{\pi^2 d^2} \left\{ 1 - \cos \pi d + 2 \sum_{n=1}^{\infty} \left[\frac{1 - (-1)^n \cos \pi d}{\left(1 - n^2/d^2\right)^2} \exp\left(-\frac{n^2 \pi^2 D B \Delta}{a^2}\right) \right] \right\} \quad [3.3.1]$$

where $R_{(t)} = A(t)/A(0)$, $\theta = \gamma \delta G a$, $d = (\theta/\pi) |\cos \alpha|$. The variable a is the barrier spacing, α is the angle between the barrier and B_0 whilst A and B are related to barrier permeability by:

$$A = \cos^2 \alpha / \left(1 + \frac{1}{P}\right) \quad \text{and} \quad B = \frac{1}{1 + P}, \quad \text{where } P \text{ is reduced permeability.}$$

Eqn.[3.3.1] expands to:

$$\frac{A_{(t)}}{A_{(0)}} = \exp\left[-(\gamma G \delta)^2 D \Delta (\sin^2 \alpha + A)\right] \frac{2}{(\gamma G \delta a)^2 |\cos^2 \alpha|} \left\{ 1 - \cos((\gamma G \delta a) \cos \alpha) + 2 \sum_{n=1}^m \left[\frac{1 - (-1)^n \cos((\gamma G \delta a) \cos \alpha)}{\left(1 - n^2 \pi^2 / (\gamma G \delta a)^2 |\cos^2 \alpha|\right)^2} \exp\left(-\frac{n^2 \pi^2 D B \Delta}{a^2}\right) \right] \right\} \quad [3.3.2]$$

If we assume that the parallel planes lie at a tangent to the direction of B_0 , then $\alpha=0$, $\sin\alpha=0$, and $\cos\alpha=1$, giving:

$$\frac{A_{(t)}}{A_{(0)}} = \exp\left[-(\gamma G \delta)^2 D \Delta A\right] \frac{2}{(\gamma G \delta a)^2} \left\{ 1 - \cos(\gamma G \delta a) + 2 \sum_{n=1}^m \left[\frac{1 - (-1)^n \cos(\gamma G \delta a)}{\left(1 - n^2 \pi^2 / (\gamma G \delta a)^2\right)^2} \exp\left(-\frac{n^2 \pi^2 D B \Delta}{a^2}\right) \right] \right\} \quad [3.3.3]$$

If we also assume that there is no permeability, where $P=0$, then $A=0$, $B=1$ simplifying the model to:

$$\frac{A_{(t)}}{A_{(0)}} = \frac{2}{(\gamma G \delta a)^2} \left\{ 1 - \cos(\gamma G \delta a) + 2 \sum_{n=1}^m \left[\frac{1 - (-1)^n \cos(\gamma G \delta a)}{\left(1 - n^2 \pi^2 / (\gamma G \delta a)^2\right)^2} \exp\left(-\frac{n^2 \pi^2 D \Delta}{a^2}\right) \right] \right\} \quad [3.3.4]$$

Substituting in the relative terms for the RD-CTPG pulse sequence, where the pulse length δ is equal to τ and the variable diffusion time Δ is equal to $\tau + t$.

$$\frac{A_{(t)}}{A_{(0)}} = \frac{2}{(\gamma G \tau a)^2} \left\{ 1 - \cos(\gamma G \tau a) + 2 \sum_{n=1}^m \left[\frac{1 - (-1)^n \cos(\gamma G \tau a)}{\left(1 - n^2 \pi^2 / (\gamma G \tau a)^2\right)^2} \exp\left(-\frac{n^2 \pi^2 D (\tau + t)}{a^2}\right) \right] \right\} \quad [3.3.5]$$

Using eqn.[3.3.5], a computer program was written to simulate the effect on signal intensity for a series of barrier spacings, for different RD-CTPG parameter sets* where diffusion time is varied though expected to be constant if the sample is totally free and unrestricted.

This model is of course for a system of parallel planes, though the principle still holds true for any regular restricting environment. Initial simulations were performed for a pair of RD-CTPG experimental parameters within a set. An extensive range of barrier spacings was used for these simulations. The results for a typical pair of parameters are shown in fig.[3.3.1].

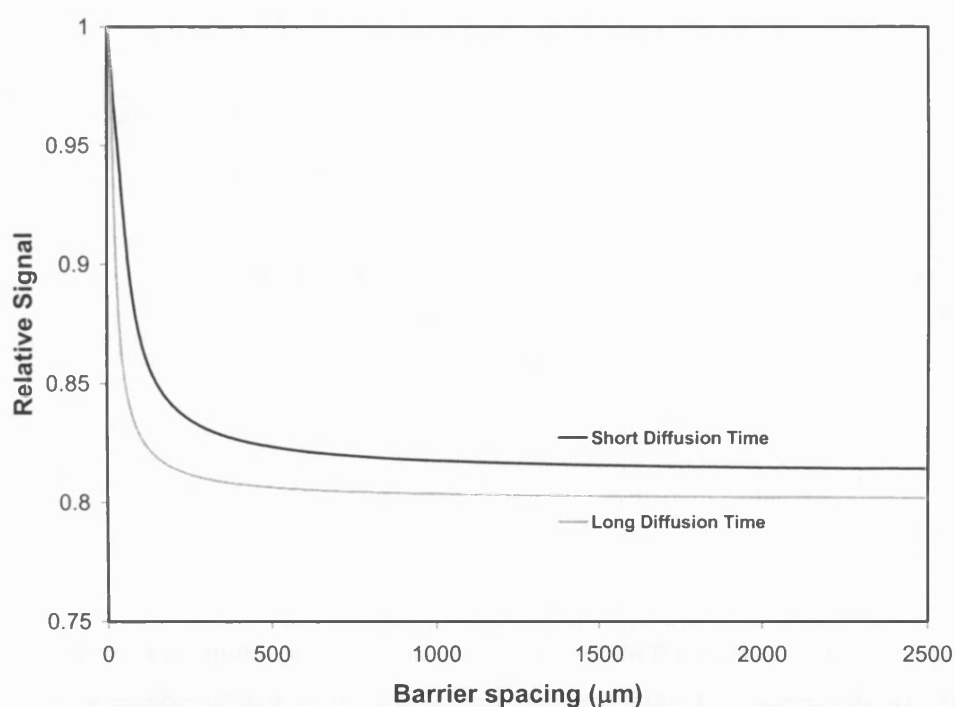


Figure 3.3.1: The graph showing the relationship between relative signal intensity and barrier spacing for two sets of experimental parameters for the RD-CTPG pulse sequence. The parameter sets have been calculated to give identical signal for an unrestricted system.

Since the signal intensities do not meet at a common value at infinity spacing, where there is no restriction and therefore should be no difference in signal intensity, the suitability of this model is questionable. For this reason, an alternative model taken from a paper published by Balinov *et al* was applied to the RD-CTPG pulse sequence under the same constraints.

* A parameter set for RD-CTPG is a series of experimental parameters, which in diffusion time only and in a non-restricting environment will give rise to identical signal attenuation as one another.

3.3.2 Model 2 (Balinov *et al*)

Again, this paper⁵ bases its application on the PGSE pulse sequence, though it is equally applicable to the RD-CTPG pulse sequence as we shall see in this section.

In the paper, the authors describe how signal attenuation varies with barrier spacing based on a modification to the theoretical framework derived from Neuman⁶ and Murday and Cotts⁷. The equation below, eqn.[3.3.6], is recast for the RD-CTPG pulse sequence.

$$\frac{A(t)}{A(0)} = \exp - \left\{ \frac{-8\gamma^2 g^2 a^4}{D\pi^6} \sum_{n=0}^{\infty} \frac{1}{(2n+1)^6} \left[\begin{aligned} &4 + \exp\left[-(2n+1)^2 \pi^2 D(\Delta_1 - \delta_1)/a^2\right] + \exp\left[-(2n+1)^2 \pi^2 D(\Delta_2 - \delta_2)/a^2\right] \\ &- 2 \exp\left[-(2n+1)^2 \pi^2 D(\delta_1)/a^2\right] - 2 \exp\left[-(2n+1)^2 \pi^2 D(\delta_2)/a^2\right] \\ &- 2 \exp\left[-(2n+1)^2 \pi^2 D(\Delta_1)/a^2\right] - 2 \exp\left[-(2n+1)^2 \pi^2 D(\Delta_2)/a^2\right] \\ &+ \exp\left[-(2n+1)^2 \pi^2 D(\Delta_1 + \delta_1)/a^2\right] + \exp\left[-(2n+1)^2 \pi^2 D(\Delta_2 + \delta_2)/a^2\right] \end{aligned} \right] \right\} \left[2(\delta_1 - \delta_2) - \frac{(2n+1)^2 \pi^2 D / a^2}{(2n+1)^2 \pi^2 D / a^2} \right] \quad [3.3.6]$$

Once more, a computer program was written to express the equation for a variety of barrier spacings using a number of diffusion times within the same RD-CTPG parameter set. The results for the simulation of this model are shown in fig.[3.3.2] and show a common end point for infinite barrier spacing.

The difference between the two, results from the different values of τ_2 and t since the experiment with the larger diffusion time will experience more restriction for a given barrier spacing than the shorter diffusion time. The two distributions are initially identical since the barrier spacing is so small that almost all of the molecules will experience near continual restriction. As the barrier spacing increases however, the number of molecules experiencing restriction will vary and the two distributions diverge. Eventually the two distributions will, once more, merge as in both cases very few molecules, if any, will experience restriction.

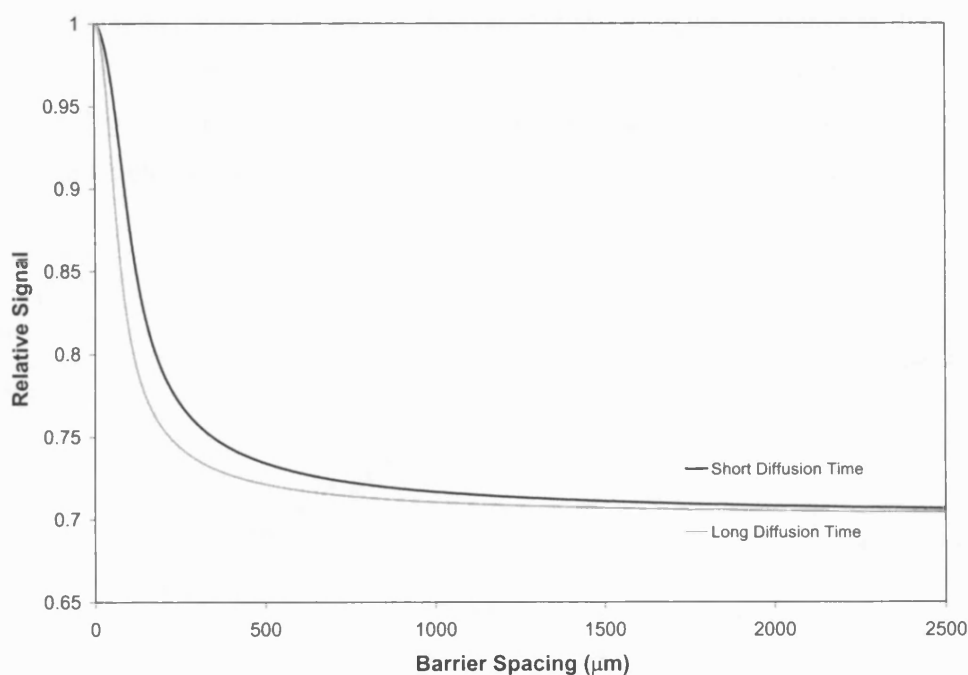


Figure 3.3. 2: The graph showing the relationship between relative signal intensity and barrier spacing for two sets of experimental parameters for the RD-CTPG pulse sequence based on the Balinov *et al.* model. The parameter sets have been calculated to give identical signal for an unrestricted system and therefore should merge towards infinite barrier spacing (no restriction).

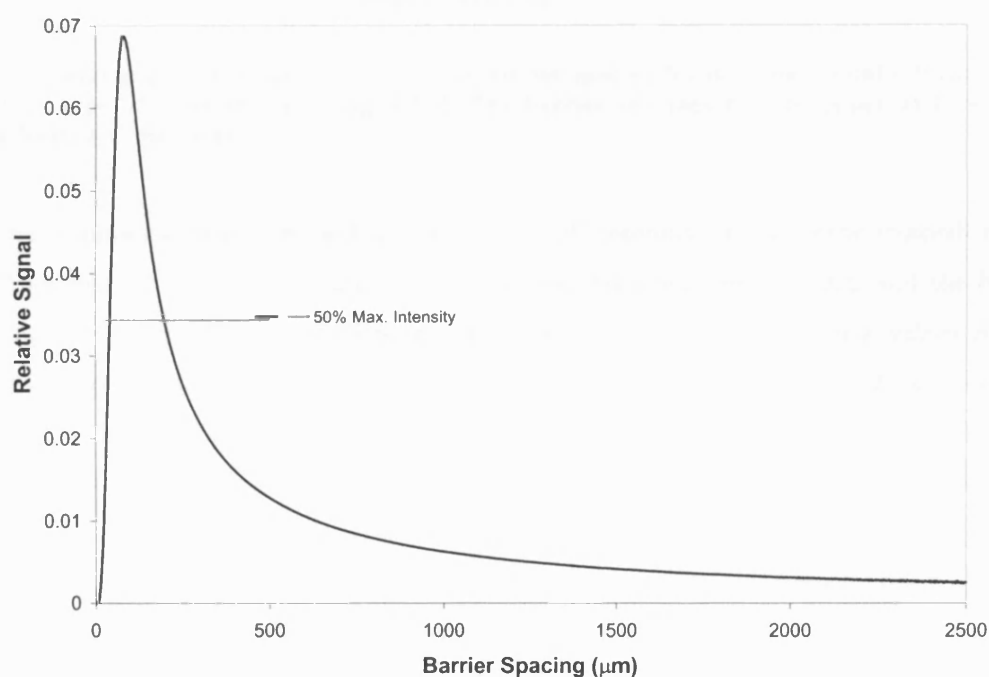


Figure 3.3. 3: The profile for the difference in signal intensity between the data profiles given in fig.[3.3.2]. This plot indicated the contribution of signal observed in an RD-CTPG difference spectrum/image from each barrier spacing. The upper 50% of signal is identified, implying that a finite range of barrier spacing can be identified by using this upper limit of signal.

The subtraction of one distribution from another results in the difference profile is given in fig.[3.3.3]. From this difference profile, it is clearly apparent that a limited range of barrier distances contribute to it and by disparate amounts where the peak of the profile corresponds to the barrier spacing which gives rise to the greatest signal change between experiments.

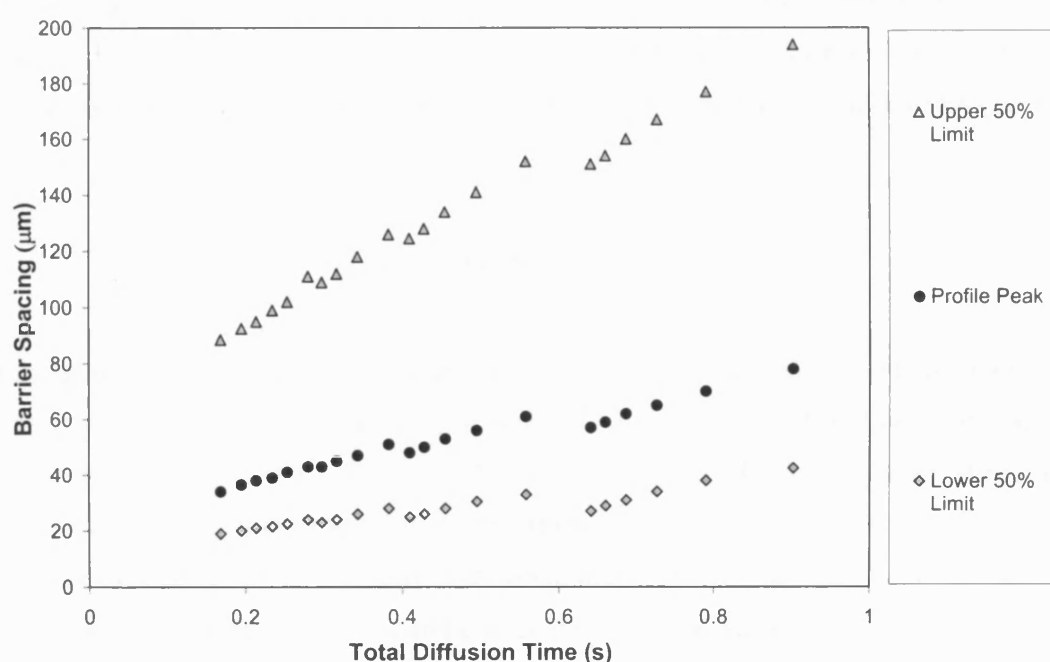


Figure 3.3. 4: This graph shows an increase in the barrier spacing for increasing total diffusion time based on the type of plots shown in fig.[3.3.3]. The barrier spacings for the upper and lower 50 percentile limits are also given.

The simulation was repeated for a variety of combinations of experimental delays calculated to give the same signal attenuation in a non-restricting environment, and the barrier spacing maxima for the difference profiles recorded. The relationship between these values and the sum of the diffusion times for the two parameter sets is shown in fig.[3.3.4]. In reality to observe only the peak of the distribution would be impracticable since the overall signal measured. For this reason, the 50% intensity limits of the distributions have also been plotted in fig.[3.3.4]. The resulting range of barrier spacings is still broad especially at the larger spacing end of the distribution. Although we are unable to select signal only for a pre-determined barrier space by careful selection of diffusion times, this simulated work does indicate applications for the pulse sequence. For instance, it is apparent that the pulse sequence can be used as a test for restriction within a system and implies that it may be used, under the right conditions, for barrier size measurements, something that is approached later in this chapter.

3.4 Experimental 1.

The majority of experiments were performed using a Bruker ARX 250 MHz spectrometer, though a Bruker DRX 400 MHz was used later in some emulsion measurements. A linear magnetic field gradient was created by mis-setting the shim coils. The strength of the gradient was calibrated by measuring the width of the peak (or profile) for a tube of pure water. The width relates to the strength of the gradient according to the modification of the Larmor equation shown in eqn.[3.4.1] where the internal diameter of the NMR tube is δx in metres and the total peak width (1D profile) is $\delta\omega$ in Hz.

$$\delta\omega = \gamma G_x \delta x / 2\pi \quad [3.4.1]$$

The inclusion of 2π is to accommodate the units of γ which include radians and therefore converts the result to Hz. The gradient strengths were found to be 7.2 and 9.6 mTm⁻¹ from profile widths of 1270 and 1690 Hz for the ARX250 and the DRX400 respectively. Once the gradient strengths have been calibrated it was possible to calculate the experimental parameters. These were determined by first fixing the desired amount of signal attenuation due to diffusion and a value for τ_1 . The values of τ_2 can then be incremented linearly and a value for t extracted according to eqn.[3.2.1].

The initial experiments were performed using a water/D₂O mixture as an unrestricted control where signal intensity is constant for all diffusion times used. The exact proportion of water to D₂O in the mixture is arbitrary as we are assuming a constant apparent diffusion coefficient and its absolute value is not important. The desired result for these unrestricted experiments is constant signal intensity for all measurements made within a parameter set. Once this had been set up, so that the experiment was repeatable, a suitable restricted environment was chosen. The systems chosen were water in oil emulsions, partially because of previous studies in the literature^{8,9,10,11,12}, but mainly because there are minimal internal field gradients generated as a consequence of susceptibility difference. This latter criterion is crucial for validating the technique, as it is assumed that the constantly applied MFG is linear throughout the sample. A 1D spectrum of a well-tuned, well-shimmed sample was recorded and the water peak width was found to be 15 Hz. From this is safe to assume that any internal field variation is small and can be ignored.

With oil and water emulsions, there are two possible situations: the formation of oil droplets in water, or the formation of water droplets in oil. In the oil in water condition, the water molecules are limited in their diffusive path by the oil droplets, though it is possible for them to spread out in a gaussian manner as there is no finite boundary. This is known as a tortuous environment. With a water in oil emulsion, the droplets of water are fully contained by the oil and

the environment is truly restricted. The size and distribution of the oil and water phases in emulsions is of great interest to a number of industries and has been the subject of several NMR studies. For this study we used an established emulsion preparation technique⁸, where, 0.1g of SPAN65 (sorbitan tristearate) was dissolved in 7.5 cm³ of warm paraffin oil. An equal volume of distilled water was then added whilst agitating the mixture. Once all of the water had been added, the mixture was shaken using a vibromixer for 10 minutes.

The experimental parameters used were calculated from eqn[3.1.1], where the values for τ , T , D and $A_{(t)}/A_{(0)}$ were fixed to 30ms, 1.5s, $2.3 \times 10^{-9} \text{m}^2 \text{s}^{-1}$ and 0.5 respectively. This allowed a variety of combinations of δ and t for the parameter set, which would give rise to the same signal intensity for unrestricted environments, despite the different diffusion times used.

To get maximum accuracy, the 32 step phase described below was adopted where the phases in brackets are repeated by the associated multiplying value and a relaxation delay of 7s was used to allow for full relaxation to occur.

RF pulse 1 phase = x

RF pulse 2 phase = x y -x -y

RF pulse 3 phase = x y -x -y -x -y x y

RF pulse 4 phase = {x}*8 {y}*8 {-x}*8 {-y}*8

RF pulse 5 phase = x

RF pulse 6 phase = x

Receiver phase = {x -x x -x -x x -x x -x x -x x x -x x -x}*2

3.5 Results and Analysis.

Data acquired using the RD-CTPG pulse sequence and the parameters previously described, is represented in the fig.[3.5.1], for both a restricted emulsion sample and unrestricted, free water control.

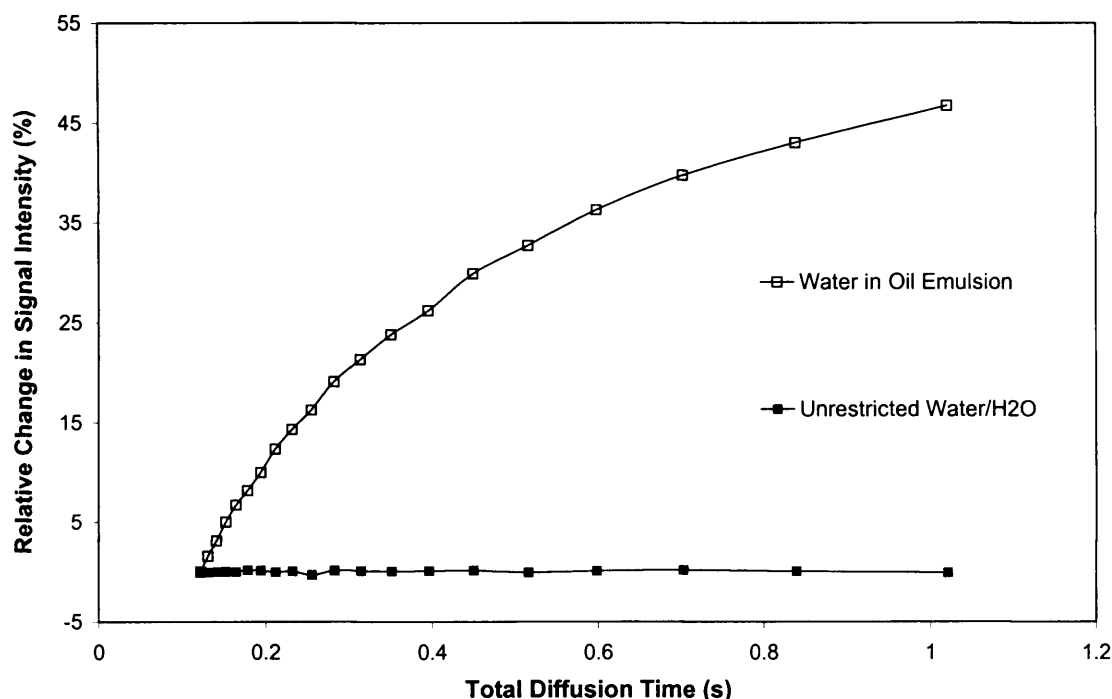


Figure 3.5.1: A graph plotting the relative signal intensities measured using the RD-CTPG pulse sequence on a restricted and an unrestricted system. The experimental parameters for these results were: $G = 9.6 \text{mTm}^{-1}$, $\tau_1 = 30 \text{ ms}$, $\delta = 2\text{--}22 \text{ ms}$, $T = 2 \text{ s}$, $t = 100\text{--}1000 \text{ ms}$. The experimental temperature was 298K.

From this data, it is clear that the pulse sequence is sensitive to the ADC as suggested earlier in section 3.2. The restricted emulsion system shows a rapid increase in signal intensity as the diffusion time increases, levelling off towards higher diffusion time. In contrast, the freely diffusing, unrestricted water shows no significant deviation in the signal intensity.

The reason for this change in signal intensity for the restricted sample has been discussed at length already in this thesis and is no different to that observed in any equivalent PGSE based experiment. The significant point about these data, is that the restriction is observed almost immediately. All that is required to say that a systems spins experience restricted diffusion are two acquisitions recorded at different diffusion times (though obviously for the same parameter set). This will give rise to two different signal intensities arising from different ADCs. The difference between the two diffusion times will of course affect the magnitude of the difference in signal intensity. It would be logical to use experimental parameters for a comparatively short diffusion time and one much longer one to maximise this difference. This would be particularly advisable when the barrier spacing are towards the edge of the experiments sensitivity as demonstrated in the difference distribution (fig.[3.3.5]), calculated from Balinovs model.

Further to this, as the diffusion time approaches it's maximum in the graph the rate at which the ADC is changing reduces. The experimental parameters were calculated using an

assumed loss of 50% signal, solely from diffusion since relaxation effects are constant. As the diffusion time increases, the number of spins experiencing restriction increases. At high diffusion times, the restriction is so prolific, that little signal is lost from diffusion and the graph levels off towards 50%. In contrast, the freely diffusing water shows little or no deviation in signal intensity since the system experiences a constant ADC throughout the experiment.

3.5.1 Semi-quantitative Analysis.

The use of restricted diffusion measurements by NMR as a means of probing the size of the restricting environment is well documented^{3,8-12,13}. The idea was initially suggested by Tanner, from whom theoretical models, such as the one discussed earlier in this chapter by Balinov, have been developed. In a later paper, Tanner and Stejskal³ suggested a model describing diffusion within different environments. Of these, the model for a spherical cavity is of great interest in the study of emulsion since the assumption that the globules in the system tend to a spherical form. The approximation suggested in the paper is given below.

$$\ln R_{\infty} \approx -\frac{(\gamma \delta g \rho)^2}{5} \quad [3.5.1]$$

Where R_{∞} is the ratio of signal at infinite diffusion time divide by signal at zero diffusion time and ρ is the radius of the spherical cavity. In this analysis, this approximation is used as a comparison to the RD-CTPG data. Of course, work has been performed that allows the fully quantitative analysis of pulsed field gradient NMR experimental data, notably that presented by Packer and Rees⁸; but this amount of detail is beyond the scope of the principle that this study aims to demonstrate.

As has been shown in this chapter, the RD-CTPG pulse sequence is sensitive to restricted diffusion and can provide data independent of the effects of free diffusion. Further to acting as a simple test for restriction, the RD-CTPG pulse sequence provides an insight into particle size analysis. Here, the RD-CTPG pulse sequence is compared with the OE-CTPG pulse sequence, which for arguments sake may be used in the same manner as the PGSE pulse sequence has been previously in the analysis of particle size with eqn.[3.5.1].

For this study a variety of emulsions was made using the method previously described. By using different surfactants: SPAN40 (sorbitan monopalmitate), SPAN65 (sorbitan tristearate) and SPAN (sorbitan trioleate) and different amounts of them, it was possible to create different sized emulsions. This is because these surfactants contain different lengths of hydrophobic carbon-hydrogen chains affecting the size of the vesicles produced. Although we were able to produce a range of emulsions with different particle size, we were unable to control the exact size of the emulsion by the preparatory method. Indeed, the particle size of an emulsion is also governed by

the amount of energy put into the system during the mixing. For example, an emulsion made using ultrasound would produce a much finer emulsion where the globule size was extremely small, whereas an emulsion made using the vibromixer would produce a coarser emulsion. By visual examination of the emulsions made during the progress of this study, this was indeed found to be the case. Unfortunately, the finer emulsions were not thermodynamically stable enough to remain in the emulsion state for a practical length of time.

Below (fig.[3.5.2]) are a series of results measured using the RD-CTPG pulse sequence with identical experimental parameters and the same 32-step phase cycle as defined above. Here, the values for τ , T , D and $A_{(t)}/A_{(0)}$ were fixed to 30 ms, 5 s, $2.3 \times 10^{-9} \text{ m}^2 \text{ s}^{-1}$ and 0.5 respectively. The only variation between experiment sets is the emulsion samples themselves. The data have been normalised using the difference between each point and the first point, divided by the difference between the first and last points to achieve a range of values between 0 and 1. This also visually maximises the difference between the results for each emulsion, and also emphasises the characteristic size of the emulsion globules. This is because signal will increase for a given set of parameters when the ADC is reduced by decreasing globule size.

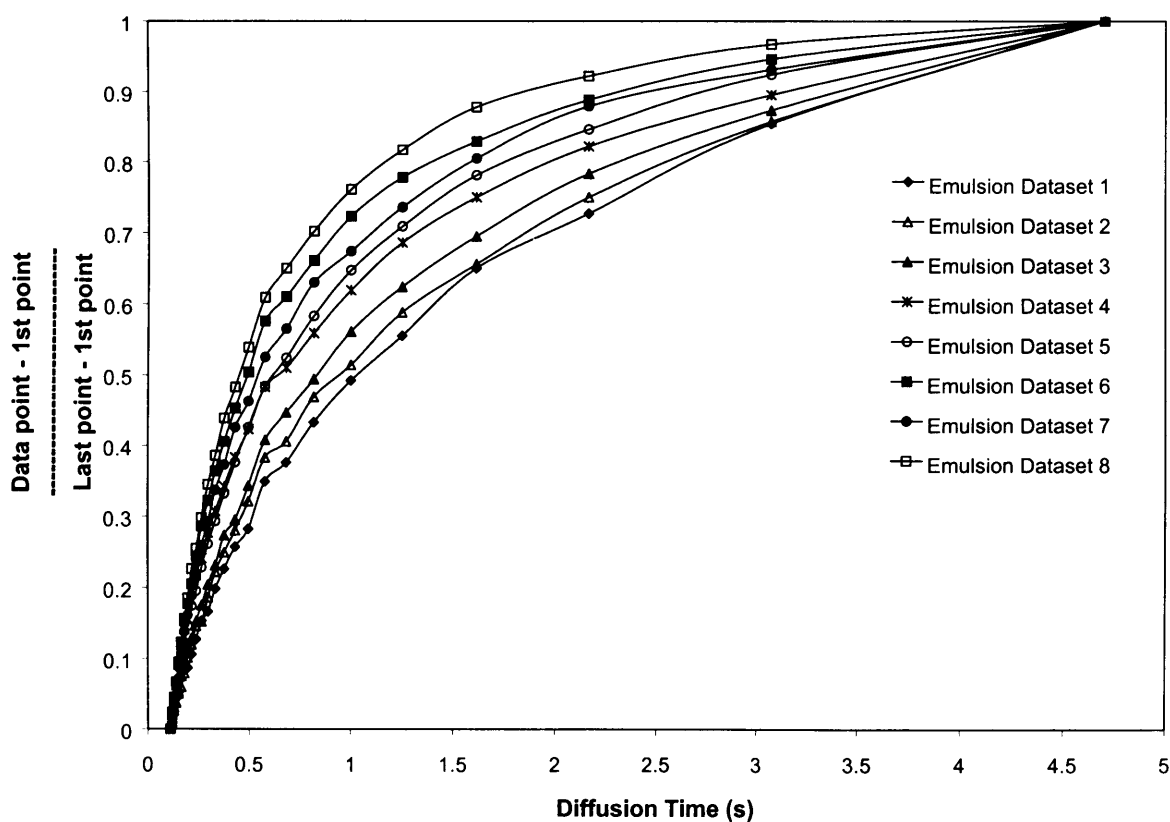


Figure 3.5.2: A plot of standardised NMR signal intensities measured from various emulsion samples under identical experimental conditions. The difference in the shape of each emulsion profile is dependent on the relative size of the water globules within the emulsion

To test the idea that these profiles from the RD-CTPG measurements were dependant on the emulsion structure, the same emulsion samples were analysed using the OE-CTPG pulse sequence. Each emulsion gave a different decay rate and ratio between the first point (at near zero diffusion time) and the final point, measured when the decay has levelled off. From this, the approximate radius of the globule can be determined using the approximation given in eqn.[3.5.1]. An example of such a measurement is given below in fig.[3.5.3]. The values for τ and T were 0.01 s and 7 s respectively.

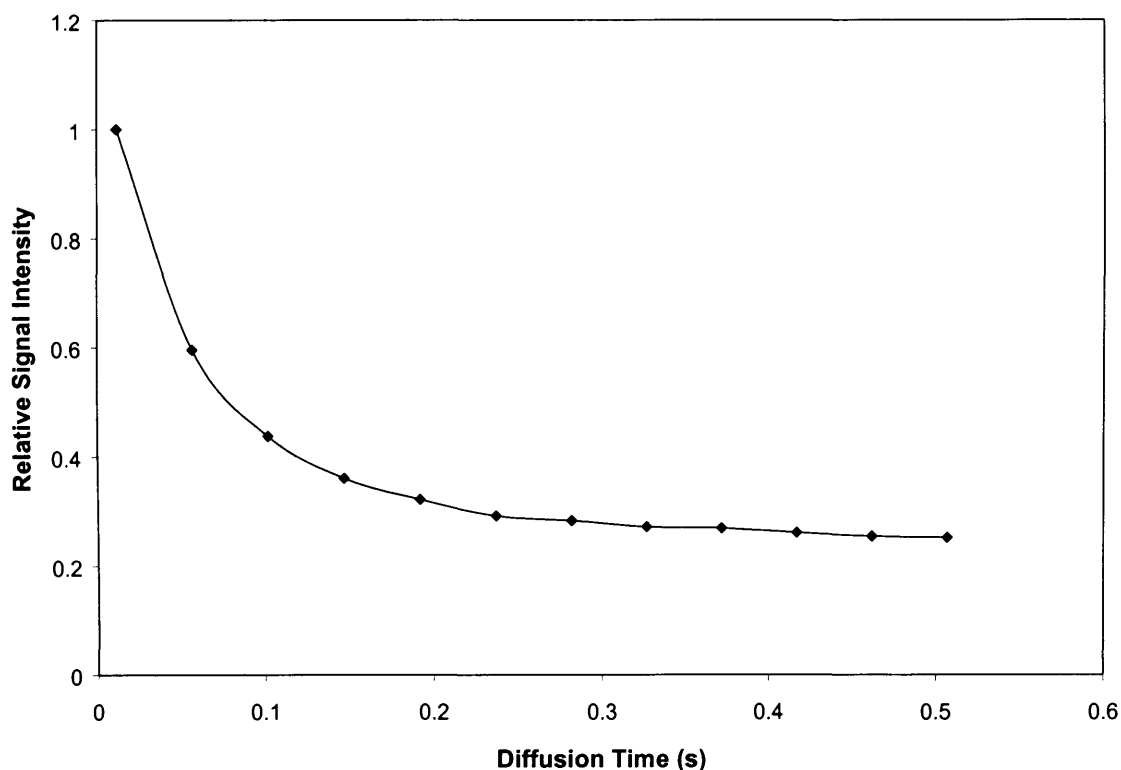


Figure 3.5.3: A plot demonstrating the decay in signal intensity with increasing diffusion time using the OE-CTPG pulse sequence with a water in oil emulsion sample. The decay levels out as expected with a restricted sample.

From these, experiments the $\frac{A_{\infty}}{A_0}$ ratios were measured giving a value proportional to the average globule radius (ρ) according to eqn.[3.5.1]. This ratio was the plotted against the ratio of signal intensity calculated using $\frac{A_{(t)} - A_{(0)}}{A_{(\infty)} - A_{(0)}}$ from the RD-CTPG experiments, where t is the same for all emulsions (i.e. the signal measured using the same time parameters for different emulsions). From this, the relationship between pore size and the magnitude of the signal when normalised, is apparent; demonstrating that RD-CTPG may also be used in pore size related applications.

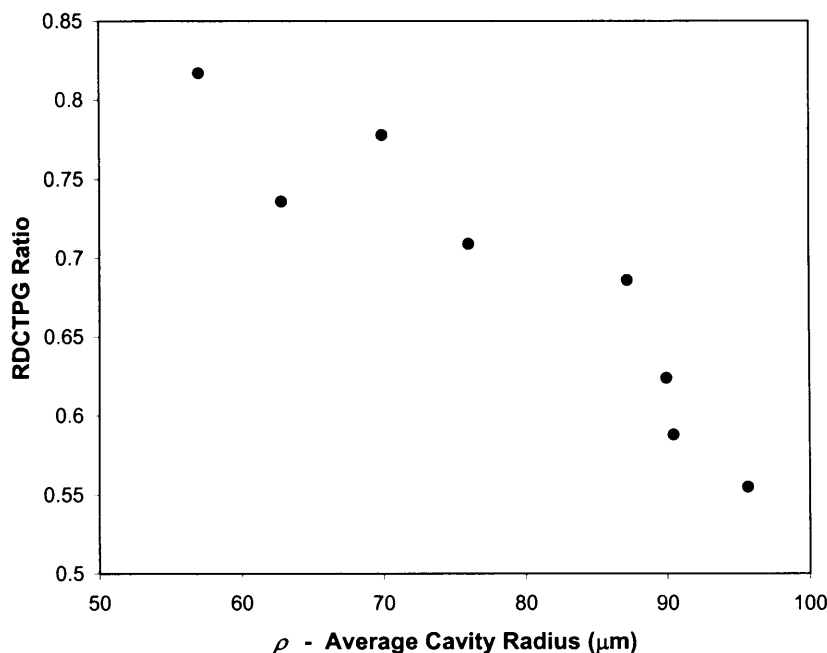


Figure 3.5. 4: The correlation between the cavity radius ρ and the RD-CTPG ratio $\frac{A_{(t)} - A_{(0)}}{A_{(\infty)} - A_{(0)}}$, is given above and measured at -0.931. This indicates the sensitivity of the RD-CTPG pulse sequence to pore size in the same manner as OE-CTPG and PGSE.

Further to this analysis, it is possible to demonstrate the advantage that the RD-CTPG pulse sequence has over the traditional methods; that being its insensitivity to free diffusion. The problem that other techniques are sensitive to free diffusion will occur when the system of interest changes with time. For example, an emulsion may begin to separate over time, this will leave the overall sample with a localised region of the emulsion, but also regions of the oil and water phases. The signal contribution from the free oil is easily removed, however, the freely diffusing water will contribute, leading to inaccurate values for the ADC and pore size. This set of circumstances is not an issue when the RD-CTPG pulse sequence is adopted since the contribution to signal loss from freely diffusing spins is constant throughout. To demonstrate this effect, two measurements were made using both the RD-CTPG and the OE-CTPG pulse sequences. In the first experiments an emulsion sample was used and the results recorded. The second experiment used the same sample in the same NMR tube, but with an addition of water. It should be noted that the sample was not agitated in any form after this addition, leaving the sample consisting of two phases, the water and the emulsion. The results for the OE-CTPG pulse sequence are shown below in fig.[3.5.5].

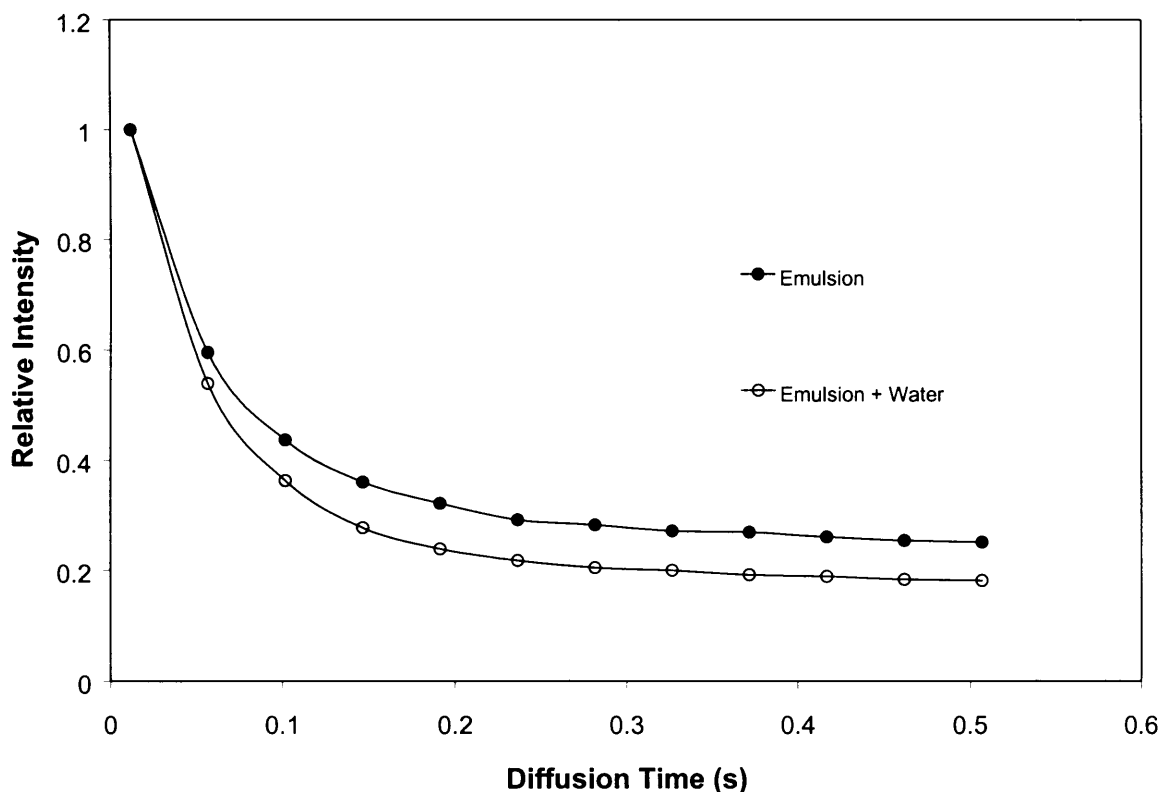


Figure 3.5. 5: This plot shows the same decay as shown in the previous figure (3.5.3) together with the signal decay measured using the exact same experimental parameters. The only difference is the addition of a drop of water to the same, previously used emulsion sample. It is clear from this plot that the OE-CTPG pulse sequence and other similar pulse sequences such as PGSE are sensitive to the presence of free water when making diffusion measurements.

From these results it is clear that the addition of free water to the emulsion system perturbs the accuracy of the experiment measurement since the two profiles level out at different asymptotic values. Indeed if the ADC were measured for both sets of results using the first two points of the decay only, there is a 20% increase in the sample where the water was added. Using the approximation given in eqn.[3.5.1] to determine the approximate average globule radius (ρ), the radius appears to be 11% larger in the sample with the pool of free water.

In contrast to these results, those measured using the RD-CTPG pulse sequence appear to be unaffected by the addition of the water as shown below in fig.[3.5.6].

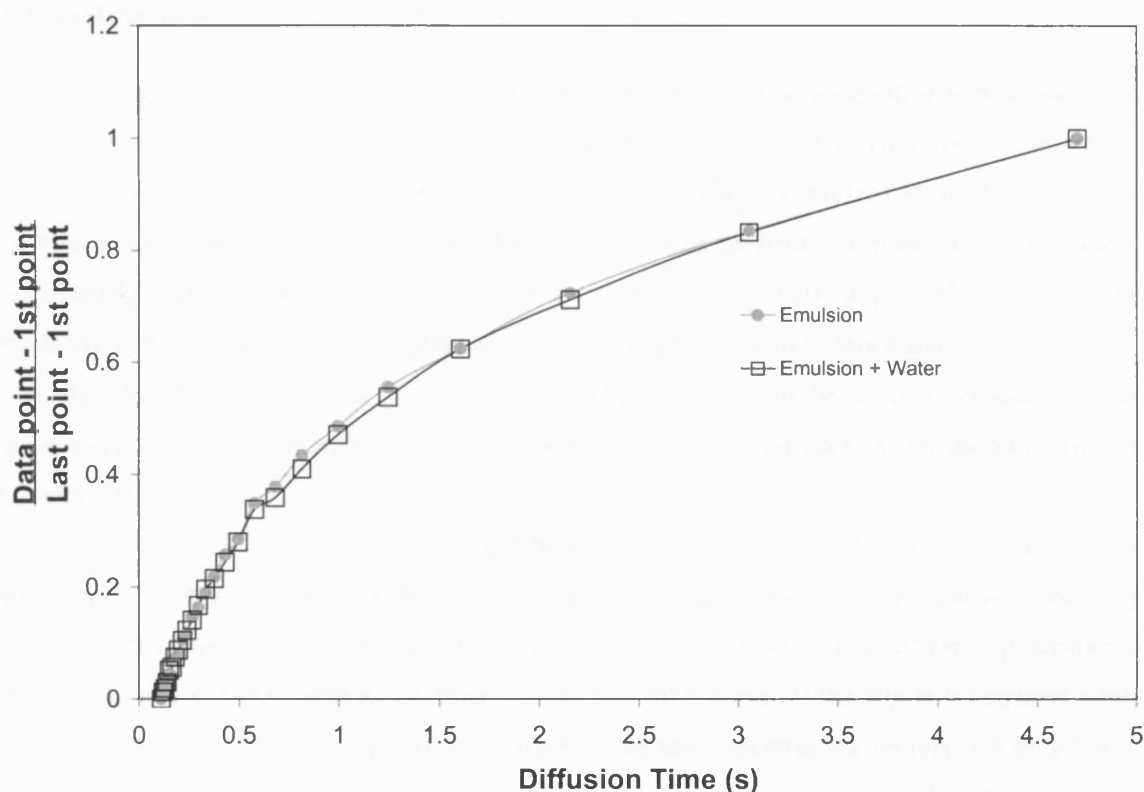


Figure 3.5. 6: A graph showing the RD-CTPG equivalent to that shown in fig.[3.5.5]. NMR signal was measured using the RD-CTPG pulse sequence for both the pure emulsion sample and that with the added water droplet. In this plot it is clear that the results do not deviate from one another by any significant amount, demonstrating the insensitivity of the RD-CTPG pulse sequence to freely diffusing, unrestricted spins.

The two sets of data are almost identical, further demonstrating the insensitivity towards free diffusion of the RD-CTPG pulse sequence. This demonstration indicates the necessity for having a consistent heterogeneous sample when making such measurements with conventional techniques since the presence of other material, in this case free water, renders the results questionable.

3.6 RD-CTPG Imaging Studies.

In the previous sections it has been shown how the RD-CTPG pulse sequence may be used to determine the amount of signal attenuation arising exclusively from restricted spins. It would, therefore, seem feasible to convert the pulse sequence into an imaging sequence, where, by taking the difference between two RD-CTPG images of different diffusion times, one would expect to see an image where only those spins that have been restricted will appear. This section will discuss and demonstrate the implementation of this technique for visualising restriction of diffusing spins.

3.6.1 NMR Imaging Theory.

In chapter 2 the effect of placing a linear magnetic field across a sample was discussed, and how a slice of the sample inside the magnet may be selectively excited when a gradient is present when R.F. energy is emitted. In NMR imaging, this phenomenon is referred to as slice selection. The selective excitation of a plane occurs when a single linear gradient is applied during excitation. This selectivity can also be made specific to rows and individual voxels (single volume unit) of the sample by applying two and three perpendicular linear gradients simultaneously during the R.F. pulse. The building up of lines or voxels are methods which can be used to produce a two-dimensional slice image, though it is the excitation of a slice which is adopted for the vast majority of imaging techniques.

In section 2.2.2, the effect of a magnetic field gradient presence during acquisition on the spectral line-width was considered. It was suggested that the gradient adjusts the Larmor frequency of spins so that spins in different positions along the gradient will have different precessional frequencies, thus encoding spatial information into the NMR signal. If the signal is acquired while the gradient is present, the range in the frequencies of the net magnetisation vectors will give rise to a resulting "spectrum" upon Fourier Transformation. This spectrum is a broad continuous peak constituting contribution from all of the spins at different frequencies. This type of spectrum is known as a 1D image or profile of a sample. If a second, perpendicular magnetic field gradient were applied simultaneously with this frequency-encoding (or *read*) gradient, a variety of profiles may be acquired depending on the relative strengths of the two gradients, where the angle of the overall read gradient changes between measurements. From these profiles, a two-dimensional slice-selected image of the sample can be calculated. This original method of obtaining a two-dimensional image is known as back projection and was first described by Laturbur in 1973¹⁴. This method for measuring images can give good results, though the reconstruction method is extremely slow to perform. Most modern NMR imaging techniques are based upon the spin-warp method first introduced by Edelstein and his co-workers in 1980^{15,16}. The sequence itself is based on Hahn's spin echo experiment¹⁷ and is shown in fig.[3.6.1].

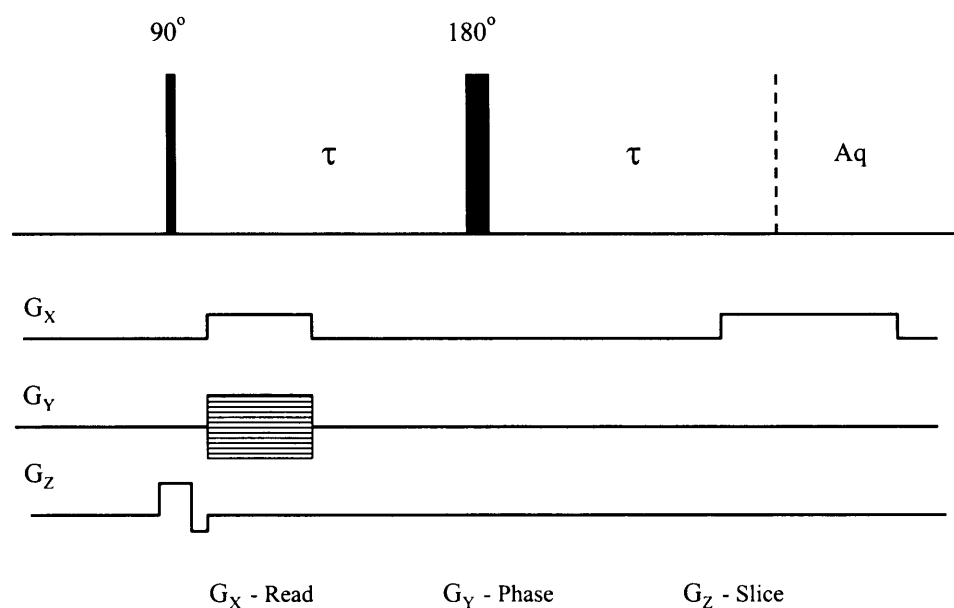


Figure 3.6.1: Graphical representation of the Spin-Warp NMR imaging pulse sequence

Here, the x-gradient is the *read* gradient and the z-gradient is the *slice-selective* gradient. The third gradient, the y gradient in this example, is an incremental gradient that is known as the *phase-encoding* gradient. This gradient encodes the second dimension of the two-dimensional image by changing the phase of the magnetisation in the second dimension. This is done by applying the y gradient for a period during the pulse sequence. This will cause a phase shift to the spins depending on their spatial location with regard to the gradient. For example, those spins at the point where there is no deviation from B_0 (i.e. no gradient experienced) will observe no change in phase, whereas those that experience a field variation during the gradient pulse, will encounter a phase shift proportional to the field strength variation and duration of this gradient pulse. Since the amount by which B_0 deviates depends on spatial location along the gradient axis, the magnitude of the change in phase is spatially dependent. The total phase shift for the entire image must be equal to 360° , therefore if an image is to contain a resolution of n the difference in phase shift between each line must be equal to $360^\circ/n$. This is usually achieved by repeating the pulse sequence by as many times as the line resolution specifies, each time linearly incrementing the gradient strength from a value of $-G_{\text{phase}}$ through zero gradient to $+G_{\text{phase}}$. Since the read gradient is reproduced for each repetition of the pulse sequence, spatial information is always recorded in the read direction. The resolution in the read direction is dependent on the number of points sampled during the acquisition and therefore does not affect the total experiment duration unlike the resolution encoded by phase variation. Each line of information is then constructed to form a two-dimensional raw data set, which essentially looks like a circularly

symmetrical echo with the greatest signal observed towards the centre in both the x and y directions. This raw data may then be Fourier transformed to generate the two-dimensional image.

All of the general principles for the imaging version of the RD-CTPG sequence are the same as for spectroscopy, though as the previous section has disclosed this could be seen as 1D imaging. The main difference is that a constant pulsed gradient is used as an alternative to mis-setting the shim coils for diffusion encoding (this is also an option should a spectrometer be implemented with gradients). It should be made clear that the gradient should be switched on well in advance of the initial excitation pulse to allow for any eddy currents generated to have curtailed. The full imaging sequence is shown below is an amalgamation of the basic RD-CTPG pulse sequence with a spin-warp imaging sequence added to the end. By using the diffusion-encoding gradient in the slice direction, the final 90° pulse of the RD-CTPG section can be made shaped and therefore selective.

3.6.2 Experimental 2.

The samples that were used in the development of this study included celery, lime, tomato and banana, of which celery was found to give the clearest results. As an unrestricted control, a tube of water was placed alongside the sample in the same direction as the read gradient so that its intensity would be seen as a separate peak in the profile work. To minimise any potential susceptibility effects when used as the restricting sample, the celery was soaked in water for 4 hours prior to experimentation. The efficacy of this treatment has been shown to work in a study by Dixon and his co-workers¹⁸. All of the imaging experiments were performed using a SISCO Varian imaging system operating at a proton resonance of 200MHz. The experimental protocol consisted of a series of profile measurements using different diffusion times as defined by the RD-CTPG attenuation equation (eqn.[3.2.1]) from which two sets of delay parameters, one of a short diffusion time and the other of a longer diffusion time, were selected and used for the imaging measurement. A 16 step phase cycle was applied to the sequence. The RD-CTPG imaging pulse sequence used for the acquisition of these images is shown below in fig.[3.6.2]. The sequence is designed so that the final 90° RF pulse of the RD-CTPG section is also the initial 90° excitation pulse of the imaging component of the pulse sequence. The imaging component is a simple spin warp sequence which will contribute to overall signal loss, but will be a constant factor between images acquired using a RD-CTPG parameter set and will therefore not affect the accuracy of the experiment^{19,20}. The constant gradient used for all RD-CTPG experiments is created using a long pulsed gradient that is switched on well in advance of the pulse sequence starting (2 s) to minimise any possible eddy currents. The gradient is then switched off just after the final 90° RF pulse, where it acts as a

slice selective gradient pulse (note that there is a refocusing lobe to minimise signal contribution from spins outside of the selected region).

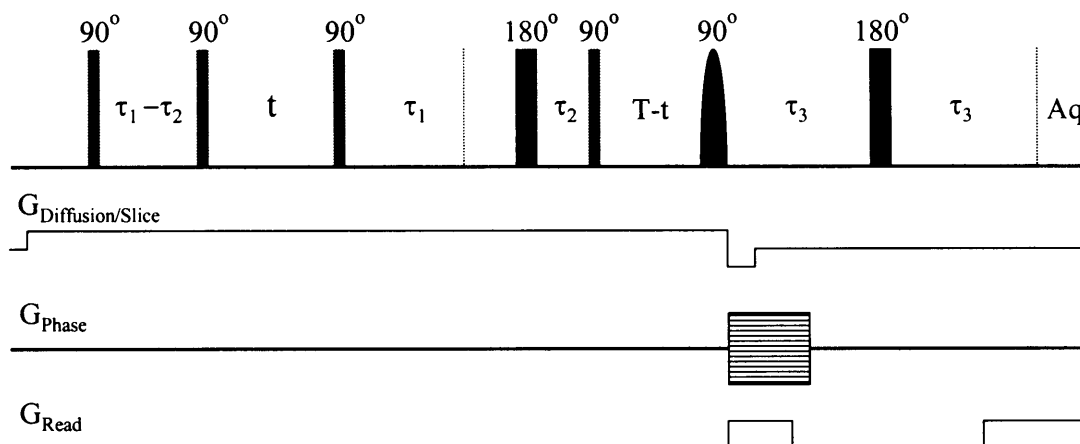


Figure 3.6.2: The imaging pulse sequence for the RD-CTPG imaging experiments. The sequence consists of the basic RD-CTPG pulse sequence together with a spin warp imaging sequence. Making the final 90° R.F. pulse of RD-CTPG and the initial 90° R.F. pulse of spin warp common joins the two components.

3.6.3 Results and Discussion.

The images produced from the resulting measurements are shown in fig.[3.6.3]. Two images are shown where there is a difference in diffusion time, which should generate a difference in the magnitude of signal acquired for the restricted regions of the sample. The unrestricted regions should show no change in signal intensity. This is shown to be the case in the third image; the difference image where the long diffusion time image (a) has been subtracted from the short diffusion time image (b). Here, the image of the two pieces of celery is distinct, whereas the part of the image where the unrestricted tube of water is in the normal images is not there. This further shows the insensitivity of the RD-CTPG pulse sequence to free diffusion. Indeed, it is possible to say that the difference image produced contains signal that is purely related to a change in signal resulting from restricted diffusion.

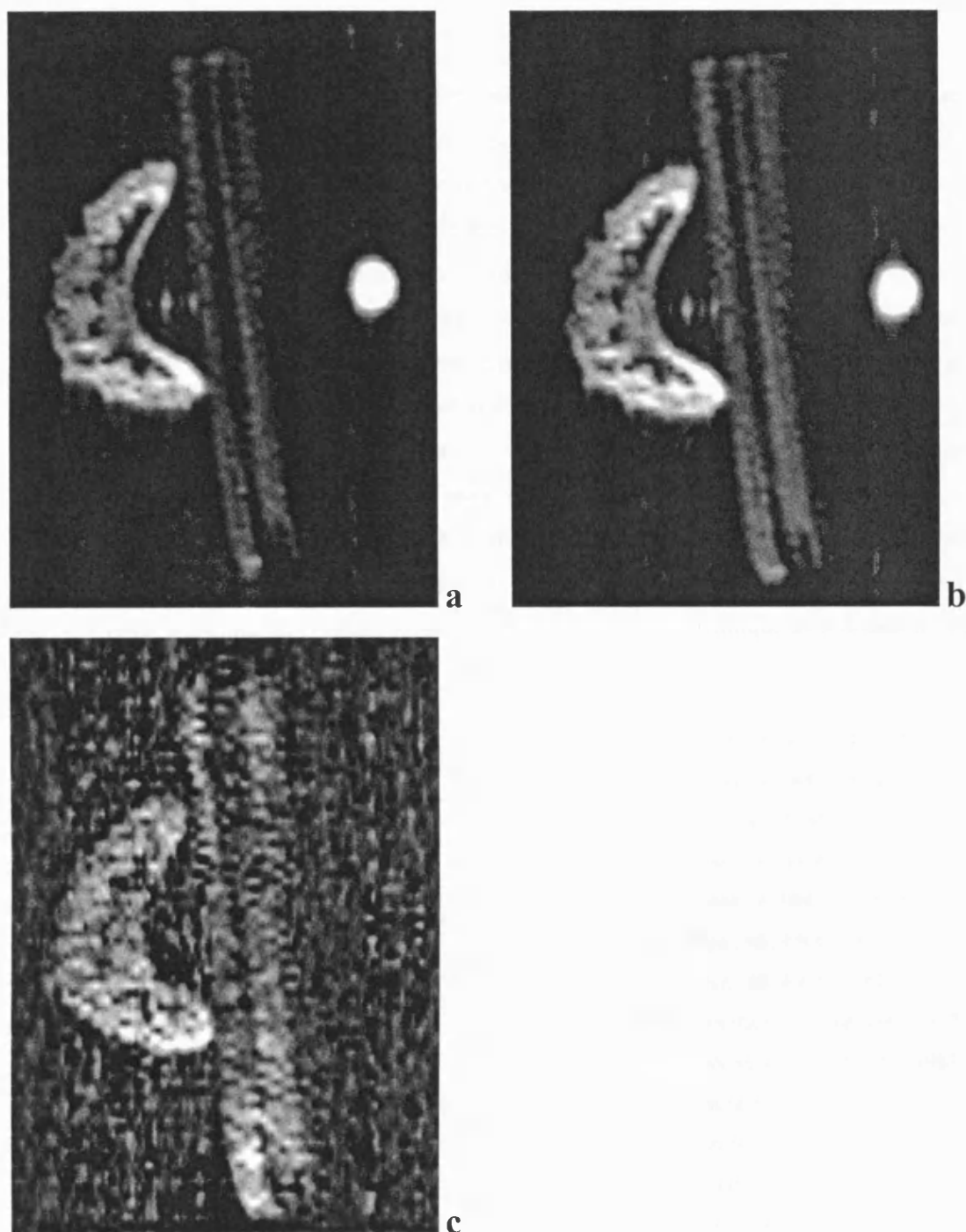


Figure 3.6.3: In both images a and b it is possible to see two pieces placed tangentially to one another together with a tube of water. The images were made using the RD-CTPG pulse sequence bolted onto a spin echo imaging sequence as seen in fig.[3.6.2]. The experimental parameters for the two images are as with other RD-CTPG experiments calculated to be insensitive to free diffusion and relaxation. The only difference between the two images are their diffusion times; image a has a longer diffusion time than b. The subtraction of one image from the other results in image c. Here, it is still possible to see the image of the celery, the restricted sample. However, the free, unrestricted sample is not visible since the contribution to signal attenuation is identical in images a and b for free diffusion and therefore cancel out.

3.7 Summary.

It has been shown that the RD-CTPG pulse sequence is insensitive to free diffusion, which makes it a valuable tool for use in diffusion studies of heterogeneous systems. This enables the sequence to be used as a quick test for restriction, simply by making two measurements and comparing their relative intensities. Should the system be unrestricted, then no difference will be apparent; if the system is restricted then the difference should be evident. There are of course limitations to the technique. The main restraint was demonstrated by the simulation made using the model in section 3.3.2, where it was apparent that the pulse sequence was sensitive only to a broad, yet finite range of barrier spacings depending on the experimental parameters used. This in itself leads to further applications, which are shown in the imaging section. In a complicated heterogeneous sample, there is a range of pore, cell or particle sizes. Insensitivity to some of these may be used as a means of contrast in a difference image. For example, apart from liquid experiencing free diffusion not appearing in the image, it may well be possible to suppress larger and or smaller sized particle region within a sample and use the technique as a means of mapping cell size for an intact sample.

-
- ¹ P. Stilbs. *Prog. NMR Spectrosc.*, **19**, 1 (1987).
- ² T. J. Norwood. *J. Magn. Reson. A*, **103**, 258 (1993).
- ³ J. E. Tanner and K. O. Stejskal. *J. Chem. Phys.*, **49**, 1768 (1968).
- ⁴ E. von Meerwall and R. D. Ferguson. *J. Chem. Phys.*, **74**, 6956 (1981).
- ⁵ B. Balinov, B. Jönsson, P. Linse, and O. Söderman. *J. Magn. Reson. A*, **104**, 17 (1993).
- ⁶ C. H. Neuman. *J. Chem. Phys.*, **60**, 4508 (1974).
- ⁷ J. S. Murday and R. M. Cotts. *J. Chem. Phys.*, **48**, 4938 (1968).
- ⁸ K. J. Packer and C. Rees. *J. Colloid Interface Sci.*, **40**, 206 (1972).
- ⁹ P. T. Callaghan and K. W. Jolley. *J. Colloid Interface Sci.*, **93**, 521 (1983).
- ¹⁰ J. C. Van Den Enden, D. Waddington, H. Van Aalst, C. G. Van Kralingen and K. J. Packer. *J. Colloid Interface Sci.*, **140**, 105 (1990).
- ¹¹ I. Lönnqvist, A. Kahn, O. Söderman. *J. Colloid Interface Sci.*, **144**, 401 (1990).
- ¹² X. Li, J. C. Cox, R. W. Flumerfelt. *AIChE Journal*, **38**, 1671 (1992).
- ¹³ I. Fourel, J. P. Guilleminet and D. Le Botlan. *J. Colloid Interface Sci.*, **164**, 48 (1994).
- ¹⁴ P. C. Laturbur. *Nature*, **242**, 190(1973).
- ¹⁵ W. A. Edelstein, J. M. S. Hutchinson, G. Johnson and T. Redpath. *Phys. Med. Biol.*, **25**, 751 (1980).
- ¹⁶ M. A. Foster and J. M. S. Hutchinson. *J. Biomed. Eng.*, **7**, 171 (1985).
- ¹⁷ E. L. Hahn. *Phys. Rev.*, **80**, 580 (1950).
- ¹⁸ J. D. Trudeau, W. T Dixon and J. Hawkins. *J. Magn. Reson. B*, **108**, 22 (1995).
- ¹⁹ T. J. Norwood, S. L. Duce and L. D. Hall. *J. Magn. Reson. A*, **102**, 370 (1993).
- ²⁰ T. J. Norwood and S. C. R. Williams. *Magn. Reson. Imag.*, **11**, 367 (1993).

Chapter 4

NMR Diffusion Measurements in Heterogeneous Samples with Large Susceptibility Variations.

4.1 Introduction.

The main techniques for studying restricted diffusion have been mentioned in section 2.2 and chapter 3. In these sections it was demonstrated how the ADC varies with diffusion time in a restricted environment, and how it is possible to obtain quantitative information about the restricting barrier from this variation.

By their nature, studies of restricted diffusion are performed on materials with some degree of heterogeneity. Because of this, the samples frequently exhibit variations in magnetic susceptibility. When placed in the static magnetic field used to polarise the nuclei, these susceptibility variations give rise to usually non-uniform magnetic field gradients within the sample. These gradients, while effective over only small distances, can be much stronger than those used to encode diffusion. The gradients interfere with diffusion measurements and, while methods have been proposed to tackle this problem, they require the use of very strong applied gradients or very careful experimental techniques^{1,2}.

Most conventional pulse sequences used to measure diffusion employ a magnetic field gradient whose amplitude is increased in consecutive experiments to encode diffusion^{3,4}. However, it has recently been demonstrated that diffusion can be measured as a function of a single diffusion time using a static magnetic field gradient of constant amplitude with the OE-CTPG pulse sequence⁵ (fig.[4.1.1]). In this chapter, the OE-CTPG pulse sequence is used in conjunction with the *internal* magnetic field gradients present within a series of glass beads-packs, soaked in a variety of fluids to study the form of the magnetic field gradients and the effects of restriction on diffusion within these samples.

Some of the experimentation in this chapter uses lung as the porous media. Several studies have been performed using lung adopting either the PGSE⁶ or CPMG⁷ technique or even a hybrid of the two⁸. In all cases it was found that the large internal magnetic field gradients arising from difference in volume magnetic susceptibility affect the measurement of the ADC⁹.

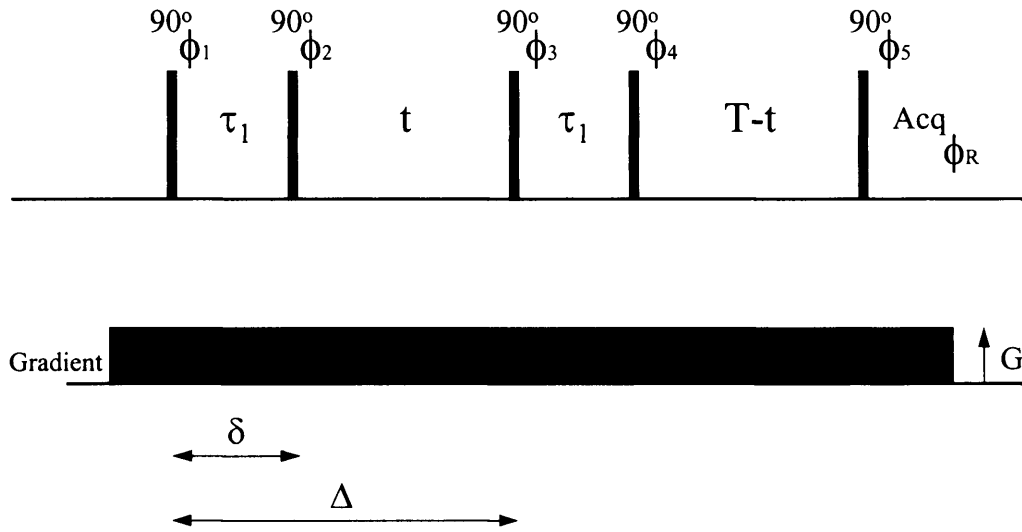


Figure 4.1. 1: Schematic timing diagram showing the OE-CTPG pulse sequence for measuring diffusion in the presence of a constant magnetic field gradient. As the period t is varied, the diffusion attenuation also changes. Magnetisation is dephased during the first δ interval and rephased during the second; the coherent magnetisation is stored along the z-axis during the t and $(T-t)$ intervals. The phase cycle used in the experiments is: $\Phi 1 = (x, -x)$; $\Phi 2 = 2(x, y, -x, -y)$; $\Phi 3 = 2(x, y, -x, -y) + 8(x, -x)$; $\Phi 4 = 16(x, -x)$; $\Phi 5 = 32(x, -x)$; $\Phi R = (x, -x) + 2(x, -x) + 8(x, -x) + 16(x, -x) + 32(x, -x)$. The number before each cycle in parentheses indicates the number of consecutive transients that are acquired with each step. Where a phase expression is a linear combination of cycles in parentheses, the phases calculated from each are added to obtain the phase used.

4.2 Magnetic Susceptibility.

When a material is exposed to a magnetic field, it will become polarised to a characteristic amount of magnetisation, M . The measure of this magnetisation is equal to the product of the field strength of the applied field, H , and the volume magnetic susceptibility, χ , of the material.

$$M = \chi H \quad [4.1.1]$$

The volume magnetic susceptibility of a material is particular to that material and is a dimensionless constant. The total magnetic field experienced by the material (B) comprises of two parts; the applied field H and M in accordance with eqn.[4.1.2].

$$B = \mu_0(H + M) = \mu_0(1 + \chi)H \quad [4.1.2]$$

where μ_0 is the vacuum permeability constant. Therefore, in a heterogeneous system, where χ will vary between constituent parts, the magnetic field of the system will also vary. This variation in the magnetic field across a sample is equivalent to having localised magnetic field gradients within the

sample. Although the magnitude of the variation in B will not be very large, these variations occur over a very small distance and therefore the magnitude of the magnetic field gradients may be large. In samples such as porous media, the gradients generated by susceptibility variations are generally non-linear, of unknown strength, and vary from point to point. In normal spin echo¹⁰ or CPMG¹¹ pulse sequences these gradients can considerably enhance transverse relaxation due to spin diffusion which will give rise to a variation in the Larmor frequency. When working with many porous media, the effects of diffusion can only be eliminated at very low applied magnetic field strength (where the amplitude of the B_0 variation is low) or by using very short echo spacing in a CPMG sequence¹².

4.3 Theory.

For the purposes of analysing diffusion effects, OE-CTPG can be considered the same as a PGSE pulse sequence. The two periods, δ , in the x-y plane can be likened to applied gradient pulses since any period the magnetisation spends in the x-y plane it will experience the constant gradient. Between these two periods, the magnetisation is rotated into the z-axis for a period of $(\Delta - \delta)$, where, the magnetisation will not experience the field gradient. The diffusion time is then equal to $(\Delta - \delta/3)$. Below the signal attenuation for the OE-CTPG pulse sequence is analysed for situations when the diffusing spins experience magnetic field gradients typical of those found inside porous media. In this analysis, both longitudinal and transverse relaxation of the bulk fluid are ignored, since they affect the magnetisation uniformly and are constant for the OE-CTPG pulse sequence. Transverse relaxation caused by surface interactions with the spins is also ignored, since the time magnetisation spends in the transverse plane (2δ) is extremely short.

For a packet of spins, the value of the squared phase dispersion due to diffusion at time of the refocused echo is:

$$\langle \varphi^2 \rangle = \gamma^2 \left\langle \left(\left\{ \int_0^\delta \Delta B_0(t) dt \right\} - \left\{ \int_\Delta^{\Delta+\delta} \Delta B_0(t) dt \right\} \right)^2 \right\rangle, \quad [4.1.3]$$

where ΔB_0 represents the deviation from the mean magnetic field across the entire sample, and the phase deviations are measured in the rotating frame. It is possible to write this expression in terms of correlation functions for the magnetic field strength as the spins diffuse throughout the inhomogeneous field. This is given below in eqn.[4.1.4].

$$\langle \varphi^2 \rangle(\delta, \Delta) = \gamma^2 \left\{ \int_0^\delta \int_0^\delta \langle \Delta B_0(t) \Delta B_0(s) \rangle dt ds - 2 \int_0^\delta \int_\Delta^{\Delta+\delta} \langle \Delta B_0(t) \Delta B_0(s) \rangle dt ds + \int_\Delta^{\Delta+\delta} \int_\Delta^{\Delta+\delta} \langle \Delta B_0(t) \Delta B_0(s) \rangle dt ds \right\}. \quad [4.1.4]$$

To simplify the analysis of this model, the dephasing period, δ , is assumed to be much shorter than the time required for a significant number of spins within the packet to reach regions of different magnetic fields strength, ie $\delta \ll \lambda_{\min}^2/D$, where λ_{\min} is a minimum length scale for field variation. Then:

$$\langle \varphi^2 \rangle(\Delta) = \gamma^2 \delta^2 \left\{ \Delta B_{01}^2 - 2 \langle \Delta B_{01} \Delta B_{02} \rangle + \langle \Delta B_{02}^2 \rangle \right\}, \quad [4.1.5]$$

where ΔB_{01} is the deviation in magnetic field during the dephasing period of the OE-CTPG pulse sequence and ΔB_{02} is that experience during the rephasing period. Given that this describes just a packet of spins, the phase dispersion for the entire spin system may be described as

$$\langle \Phi^2 \rangle(\Delta) = \gamma^2 \delta^2 \int_V \left\{ \Delta B_{01}^2(\mathbf{r}) - 2 \langle \Delta B_{01} \Delta B_{02} \rangle(\mathbf{r}) + \langle \Delta B_{02}^2 \rangle(\mathbf{r}) \right\} d\mathbf{r}. \quad [4.1.6]$$

Here, all terms within the integral are spatially-dependant functions and the integral itself is performed over the entire volume. Hence, the volume-averaged magnetic field deviation evolved during the rephasing period, $\int_V \langle \Delta B_{02}^2 \rangle(\mathbf{r}) d\mathbf{r}$, will be equal to that during the dephasing period, reducing eqn.[4.1.6] to

$$\langle \Phi^2 \rangle(\Delta) = 2\gamma^2 \delta^2 \left\{ \int_V \Delta B_{01}^2(\mathbf{r}) d\mathbf{r} - \int_V \langle \Delta B_{01} \Delta B_{02} \rangle(\mathbf{r}) d\mathbf{r} \right\}. \quad [4.1.7]$$

Given this simplified model, it now remains to characterise the cross-correlation function, $\langle \Delta B_{01} \Delta B_{02} \rangle$, as a function of the experimental diffusion time, Δ . In the case of a uniform applied magnetic field gradient across certain types of restrictive geometry, such as restriction between planes, within cylinders and spheres¹³, as well as for free diffusion, the exact forms of the correlation function (and echo attenuation) have been found. Echo attenuation for a restricted diffusion in a system with uniform magnetic field gradient has similar characteristics as those observed for free diffusion in a sample where the magnetic field oscillates with a regular wavelength, λ . Here, $\lambda/2$ would be equivalent to a , where a is the characteristic restriction size of the system showing similar echo attenuation behaviour¹⁴. To illustrate, a spin that diffuses from one restricting barrier to the other in a simple parallel plane system will experience the maximum

change in magnetic field. In a free system with an oscillating magnetic field gradient, the maximum change in magnetic field is experience when the spin diffuses a distance $\lambda/2$ along the gradient axis. Should this spin carry on and diffuse a further $\lambda/2$ in the same direction, the overall change in magnetic field would be zero in the same manner as if the restricted spin were reflected to its original position (see fig.[4.1.2]). A common feature in the known solutions for restricted geometry is that the cross-correlation function, $\langle \Delta B_{01} \Delta B_{02} \rangle$, is in the form of a sum of exponentials, where the decay rates for each exponential component are given as ratios that are fixed for any given geometry. For example, the relative decay rates for planar barriers are 1 : 9 : 25 etc. whilst for a spherical geometry, they are 1 : 8.13 : 19.6 etc. Since ΔB_0 has a mean of zero, for long diffusion times the value of $\langle \Delta B_{01} \Delta B_{02} \rangle$ must decay to zero. Therefore, we can write, without loss of generality, the correlation function as an infinite sum of exponential terms:

$$\langle \Delta B_{01} \Delta B_{02} \rangle(T) = \langle \Delta B_0^2 \rangle \sum_{i=1}^{\infty} b_i \exp\left(-\frac{a_i \Delta}{\tau_c}\right), \quad [4.1.8]$$

where a_i gives the modes for loss of magnetisation; b_i are the amplitudes of these modes; and τ_c is a correlation time, or residence time, of molecules in a particular magnetic field strength. Note that $\sum_{i=1}^{\infty} b_i = 1$. For a system where the magnetic field variations are generated by internal differences in susceptibility, it is expected that the wavelengths correspond to the variations in the solid/fluid matrix. In the case of bead packs, such as are examined below, this would be equal to the diameter of the beads. In the case of lung tissue, as is examined later in this chapter, the structure is more complicated due to additional restriction from cell walls, but should approximate to the mean distance between the alveoli (air pockets).

| Geometry | a_1 | a_2 | a_3 |
|-------------|-------|-------|-------|
| Planar | 9.869 | 88.8 | 246.0 |
| Cylindrical | 13.58 | 113.7 | 291.5 |
| Spherical | 17.32 | 141.1 | 338.9 |

Table 4.3. 1: First three decay rate constants for the geometries that have been solved exactly from Neuman's paper.

For the case where a uniform magnetic field gradient is placed across a restricted system, the first three values of a_i are given in table 4.3.1 for planar, cylindrical and spherical geometries^{9,15}. As seen in fig.[4.1.2], for these values of a_i the wavelength of a field variation

corresponds to twice the value of the barrier separation in a planar system. Clearly, these values of a_i are somewhat inappropriate for the analysis of our studies with fluid filled bead packs and lung since neither possess the above geometries, or have uniform magnetic field gradients present. Further to this, the magnetic field gradients vary in all three dimensions, whereas field variation is only along one dimension in the case of applied magnetic field gradients. Despite these inconsistencies, the general form of the correlation function described by eqn.[4.1.8] is adopted in the analysis.

Assuming that the distribution of phases in the NMR echo are Gaussian, then the magnitude of the echo may be given as:

$$\frac{M}{M_0} = \exp\left(-\frac{\langle\Phi^2\rangle}{2}\right). \quad [4.1.9]$$

It is not clear, in the context of the OE-CTPG experiment performed on porous media, that the phase distributions are indeed Gaussian. It is argued in Brown and Fantazzini's paper¹⁶ that for narrow distributions, the exact form of the distribution does not greatly affect the validity of eqn.[4.1.9]. Though, for wide distributions, where the diffusion time is longer, the phase distribution reflects the frequency distribution present over distances covered by diffusing spins during the diffusion time, which is unlikely to be Gaussian.

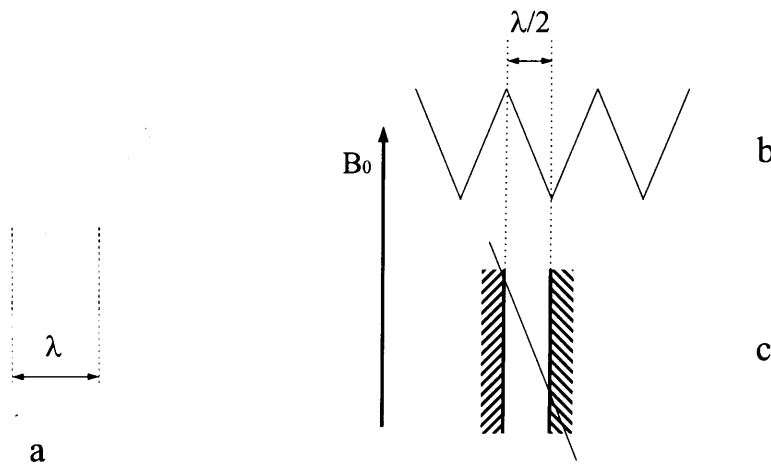


Figure 4.1. 2: Schematic illustration of a magnetic field having a saw-tooth one-dimensional periodic variation in B_0 field strength, with characteristic repeat length λ (a). For diffusing molecules, this situation is exactly equivalent to the case for a uniform field gradient with planar barriers hindering motion, and barrier separation $\lambda/2$ (b).

Despite this possibility, a Gaussian phase distribution is assumed for all diffusion times. Using this premise, and by rearranging eqns. [4.1.7], [4.1.8] and [4.1.9], together with writing the

correlation time in terms of a “wavelength” of changes in field, λ , and the diffusion coefficient, D , such that $\tau_c = (\lambda/2)^2 / D$, it can be seen that

$$\frac{M}{M_0} = \exp \left(-\gamma^2 \delta^2 \left\{ \int_V \Delta B_0^2(\mathbf{r}) d\mathbf{r} - \int_V \Delta B_0^2(\mathbf{r}) \sum_{i=1}^{\infty} b_i \exp \left(-\frac{a_i D \Delta}{(\lambda/2)^2} \right) d\mathbf{r} \right\} \right). \quad [4.1.10]$$

From this equation, it is expected that by performing the OE-CTPG experiment with a range of diffusion times, the observed magnetisation should decay with a dependence on Δ , towards some asymptotic value with multi-component behaviour. The component decay rates and amplitudes reflect the modes of decay, and from the asymptote, the strength of local magnetic-field variations likely to be experienced by diffusing molecules can be found.

For this study and for many real porous media, the dispersion of pore size is not unique resulting in irregular packing/spacing. It is therefore necessary to take this situation into account within the analysis of the data. However, it is not possible to differentiate between the modes of the exponential decay and the distribution of decay rates caused by the irregular packing. The way in which it was decided to allow for this variation was by incorporating a stretching exponential in similar manner to Kenyon *et al*¹⁷ into the model where the stretching exponent will allow for a range of decay rates to be represented in a concise form

$$\frac{M}{M_0} = \exp \left(-\gamma^2 \delta^2 \int_V \Delta B_0^2(\mathbf{r}) d\mathbf{r} \left\{ 1 - \exp \left(-\frac{a D \Delta}{(\lambda/2)^2} \right)^\alpha \right\} \right) \quad [4.1.11]$$

where α is the stretching exponent. From the fits to experimental data, the values of M_0 , $\int_V \Delta B_0^2(\mathbf{r}) d\mathbf{r}$, λ^2/a and α were estimated.

4.4 Computer Simulations.

In this section, the design and construction of computer programmes to calculate the magnetic field distribution in a bead/fluid pack and to mimic the diffusion of spins within the pack allowing the simulation of NMR experiments, are discussed. The programmes were written in FORTRAN 77 and run on a Sun Sparcstation5.

The effect that a solid sphere of one volume magnetic susceptibility has upon the local magnetic field of the surrounding fluid of different volume magnetic susceptibility when placed in a uniform magnetic field, was considered in a publication by Glasel and Lee in 1974¹⁸. They suggested that the analytical form for the resultant magnetic field variation could be described as:

$$B = H_0 \left(1 - \left(\frac{(\chi_s - \chi_f)r_s^3}{(\chi_s - 2\chi_f + 3)r_p^3} \right) (1 - 3\cos^2 \theta) \right), \quad [4.1.12]$$

where, χ_s and χ_f are the volume magnetic susceptibilities for the sphere and the fluid respectively, r_s is the sphere radius, r_p is the distance between the point of interest and the centre of the sphere, and θ is the angle between H_0 and the line between the sphere centre and the point of interest. Majumdar and Gore¹⁹ took this model further by stating in their paper that in a system of numerous spheres in a fluid, the deviation from B_0 induced by the presence of each sphere may be summed up as expressed in eqn.[4.1.13]. This adaptation has been adopted on the simulations described in this chapter.

$$B_{Total} = H_0 \left(1 - \sum_{i=1}^n \left(\frac{(\chi_s - \chi_f)r_{si}^3}{(\chi_s - 2\chi_f + 3)r_{pi}^3} \right) (1 - 3\cos^2 \theta_i) \right) \quad [4.1.13]$$

In addition to knowing how to calculate the magnetic field, assumptions must be made regarding the structure and packing of the bead/fluid matrix. Three packing regimes have been used in the process of the modelling: a random close packing, a hexagonal close packing and a cubic packed environment. Although the true packing of the experimental system may be more accurately represented by the random pack or indeed the hexagonal close pack, the reasoning behind the use of the cubic structure included the fact that the programme would be simpler and use less CPU time.

4.4.1 Cubic packed field model.

The principle on which all three models are based is the creation of a cubic three-dimensional array consisting of 100*100*100 grid points. In the cubic pack model, a sphere of constant radius is placed on each of the eight corners of this *unit cell*, where the maximum sphere radius is half that of the physical dimensions of the unit cell so that the spheres are touching (fig.[4.4.1]).

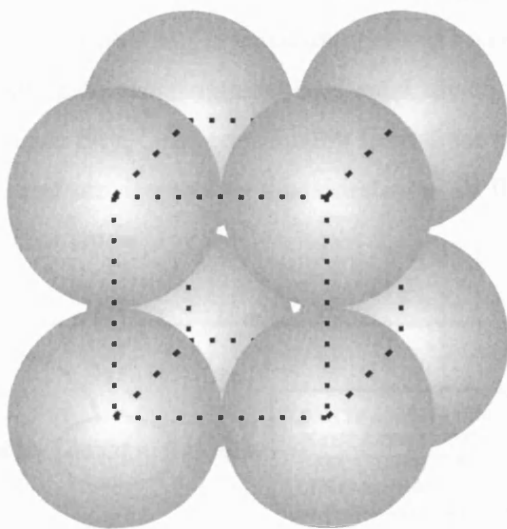


Figure 4.4. 1: The unit-cell used in the cubic close packed model.

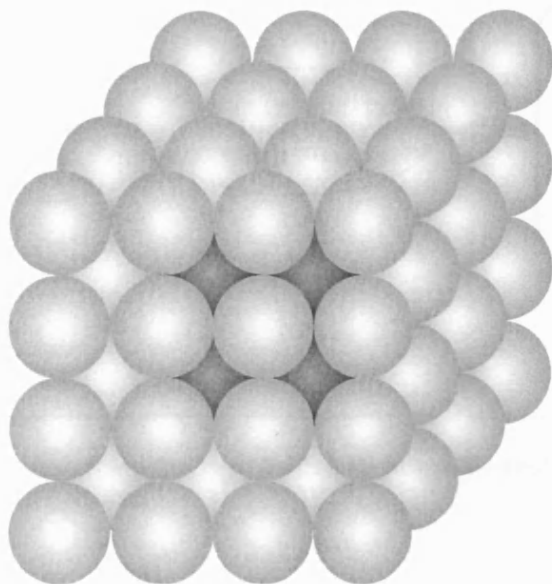


Figure 4.4. 2: In addition to the 8 spheres in the central unit-cell, spheres from the surrounding unit cells will also contribute to the deviation in the magnetic field of the central unit-cell. For simulations presented in this thesis, these are limited to the spheres in the unit-cells directly adjacent to the central unit-cell.

For each of the grid points inside the unit cell, but outside of the spheres themselves, the contribution to variation in magnetic field is calculated and summed for each of the spheres in the unit cell, and for spheres in a surrounding layer of unit cells as represented in fig.[4.4.2], so that a total of 64 spheres contribute to the magnetic field map. Since the relationship described in eqn.[4.12] shows a dependence on r_p^{-3} , the contribution from those spheres beyond this outer layer of unit cells is negligible and are ignored in this simulation. A flow chart describing the field simulation programme is given below.

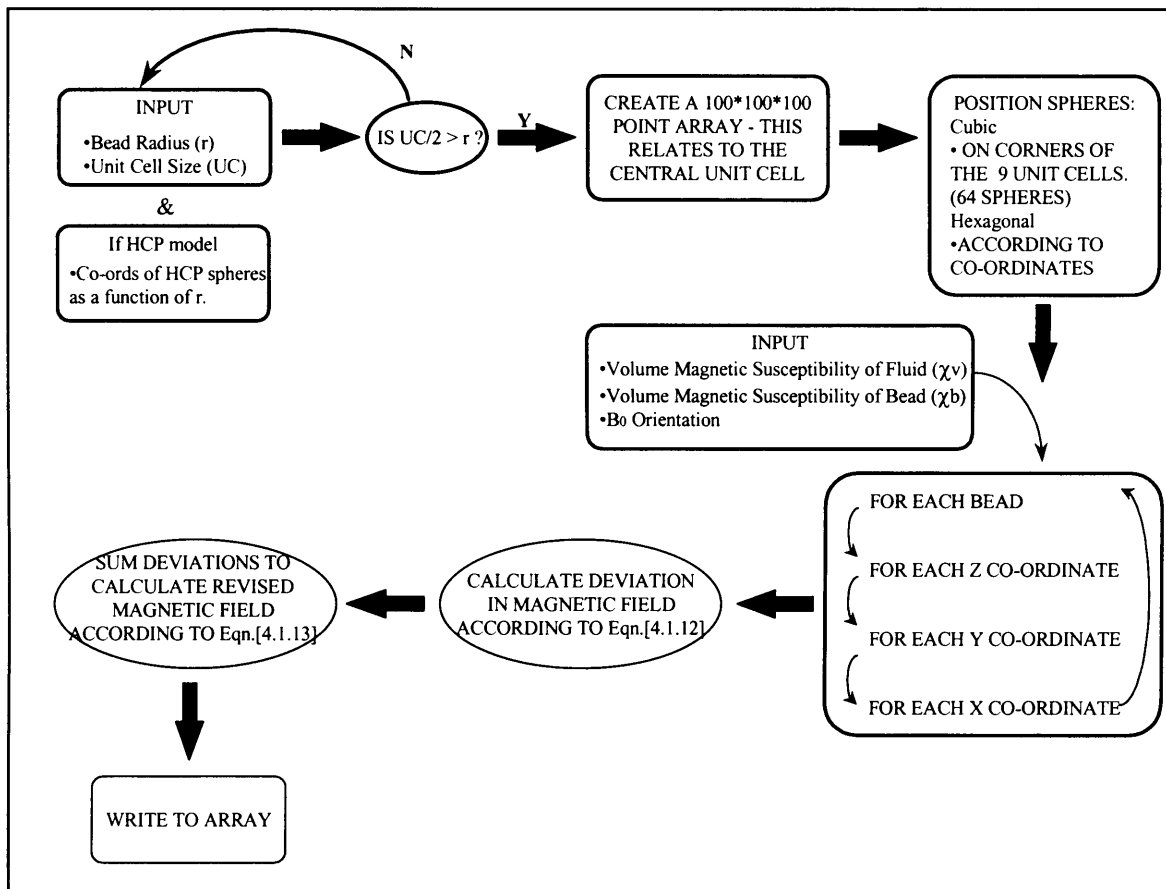


Figure 4.4. 3: Flow chart describing the logic behind the magnetic field calculation for cubic packed and hexagonal close packed systems.

4.4.2 Hexagonal pack field model.

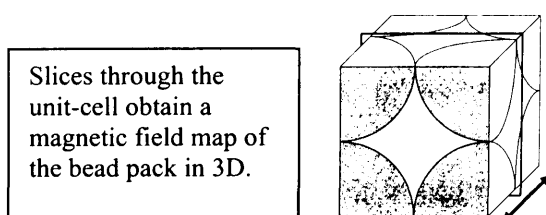
Since the co-ordinate array on which the programme is based is cubic, it was difficult to produce a logical stepping order to create a hexagonal close pack about this cube. Instead, the known Cartesian co-ordinates for the location of each sphere were typed into the code by hand as a function of the cell size. This still allowed for a variety of cell sizes in the same manner as the cubic close pack model. Only spheres that were within the 27 unit-cell region, explained in the

cubic close packed section, were included. Once the programme had a reference as to the location of the spheres, the programme reverted back to the standard magnetic field deviation calculations, though it should be noted that the reading of these references increased the CPU time significantly. The flow diagram for the hexagonal close-packed magnetic fields simulation is given below.

4.4.3 Line-shape Simulation.

It is well known that spins precessing at different frequencies (usually due to chemical shift variation), will give rise to different peaks in an NMR spectrum where the peak shapes are Lorentzian. In a similar manner it has been discussed in this thesis how water molecules of common chemical shift, will precess at different frequencies when they reside in a heterogeneous magnetic field, usually as a result of magnetic field gradient application. Here, since the range of frequencies is continuous, the spectrum will be a single broad peak (assuming that the spins form a spatially continuous pool) where the intensity of the peak (or profile in imaging terms) at any given frequency is proportional to the spin density for that given magnetic field. It is therefore feasible to suggest that the NMR line-shape produced for water in a porous media is indicative of the range of magnetic field variations and the relative quantities of spins located for all field values. The line-shape can therefore be said to be indicative of the type of packing or structure of such porous media.

The line shape programme adds up all of the individual line-shapes which will arise from the cubic unit cell by creating a linear array relating to the proton frequency which, in turn, depends on the magnetic field variation within the unit cell. Each array point of the unit cell is checked to see if it relates to the fluid or the bead, if it is the latter the programme will move on to the next array point; if it is the former the programme calls the next subroutine. This subroutine calculates the Lorentzian nature of the line for 5Hz either side of the main resonant frequency, the relative intensities for each of the frequencies is then added to the cumulative linear array, which will result in the final line-shape. On completion of this step the programme will move on to the next array point. Slices through a cubic magnetic field map and the line shape resulting from the whole unit-cell are shown in fig.[4.4.4]. The source for the main programme can be found in the appendices.



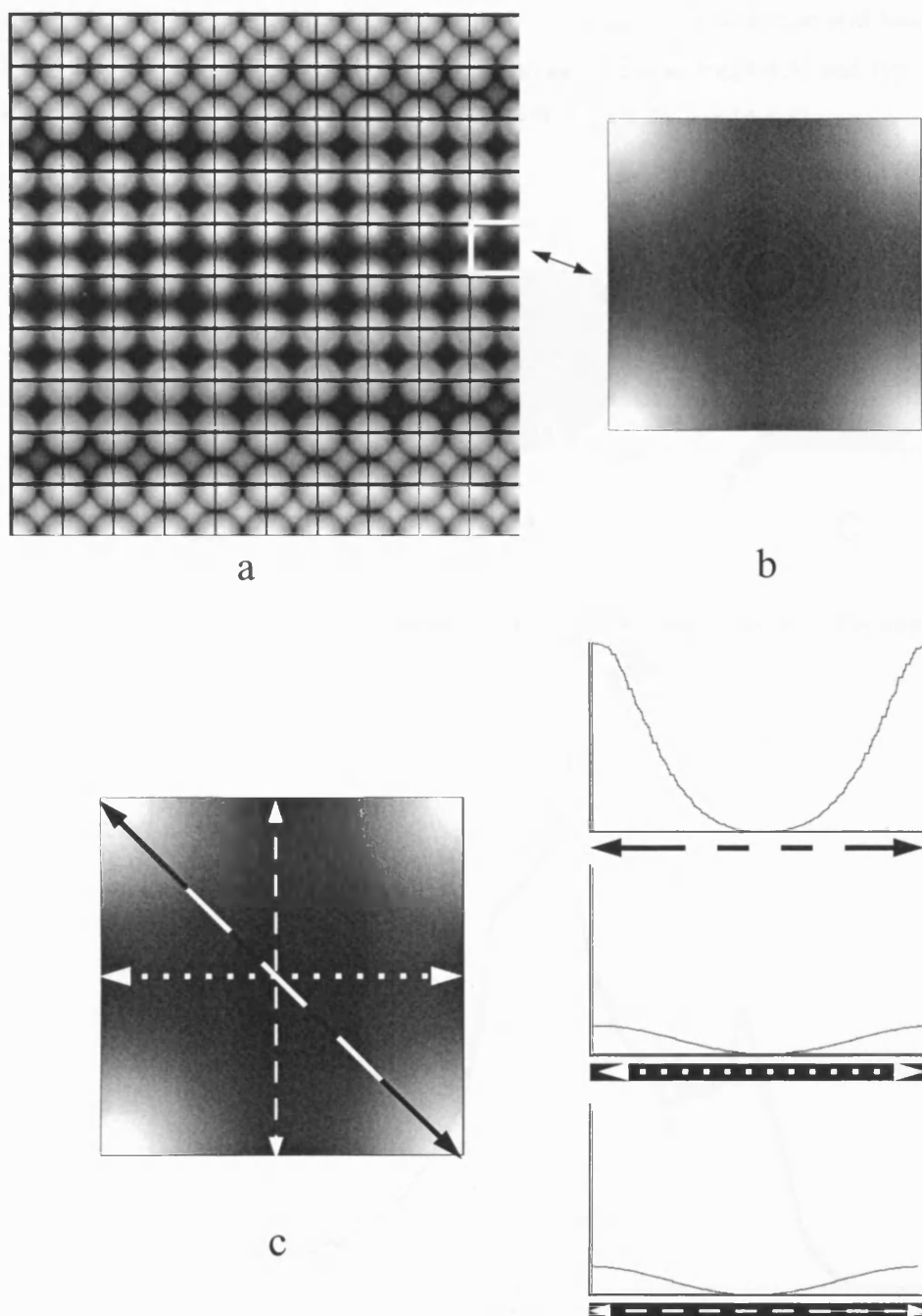


Figure 4.4. 4: (a) The magnetic field map has been sliced through the xz plane of the unit cell to show how the field alters across it. The slice shown in (b) is towards the centre of the unit cell and will therefore be mostly fluid. The profiles of the variation in magnetic field are shown in (c). Although the magnetic field gradients are clearly non-linear, they appear to oscillate in a regular manner.

It is known from eqn.[4.12] that the local magnetic field variation is dependant on the angle between H_0 and the line between the sphere centre and the location of interest. It was therefore thought that the orientation of the unit cell with respect to H_0 may give rise to different line widths which, if it were the case, may help to suggest the best unit cell structure to use for the more CPU demanding diffusion simulations. The line-shape programme was therefore re-written

for the case where H_0 is orientated along xy and xyz , with the default case still being where H_0 is orientated in the z -axis. These orientations are given below in fig.[4.4.5] and typical line-shapes resulting from the simulations are shown in figs.[4.4.6], [4.4.7], and [4.4.8].

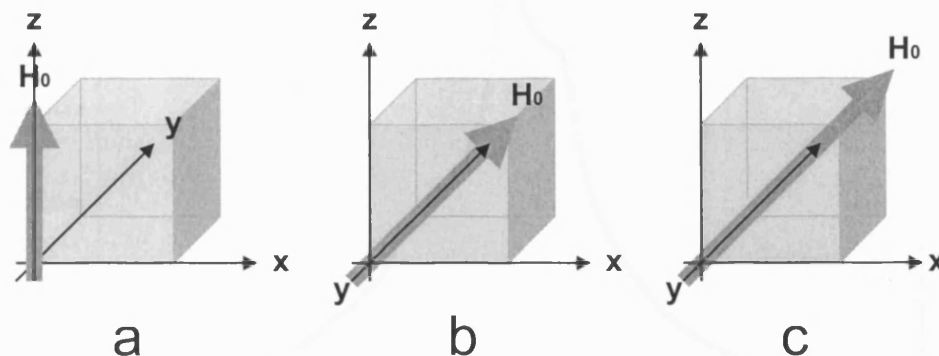


Figure 4.4. 5: The three alignments of the magnetic field with respect to the cubic unit-cell: along the z -axis (a), along $x=z$ (b), and along $x=y=z$ (c).

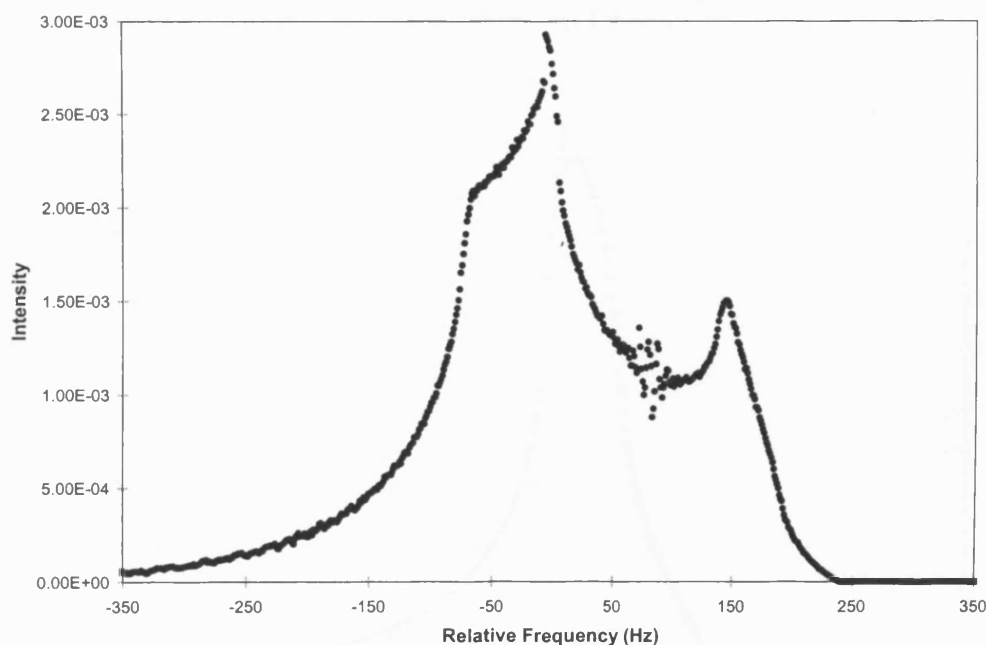


Figure 4.4. 6: Line-shape generated from the cubic unit-cell, where the magnetic field is orientated along the z -axis. The spheres in this simulation had zero susceptibility whilst the fluid had a volume magnetic susceptibility of -9.1×10^{-6} , the magnetic field was 1 Tesla.

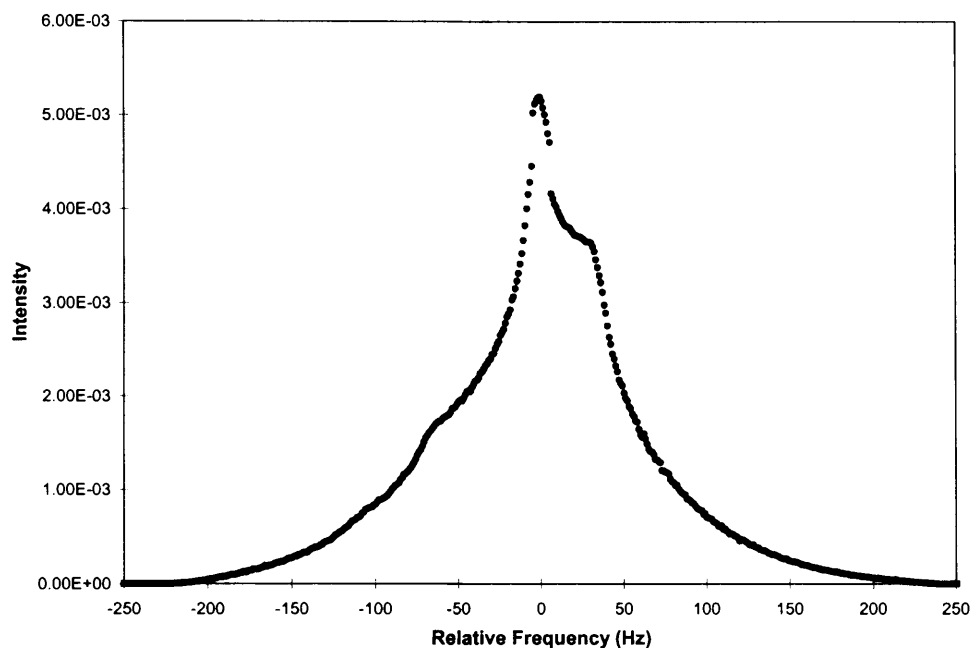


Figure 4.4. 7: Line-shape generated from the cubic unit-cell, where the magnetic field is orientated along $x=z$. The spheres in this simulation had zero susceptibility whilst the fluid had a volume magnetic susceptibility of -9.1×10^{-6} , the magnetic field was 1 Tesla.

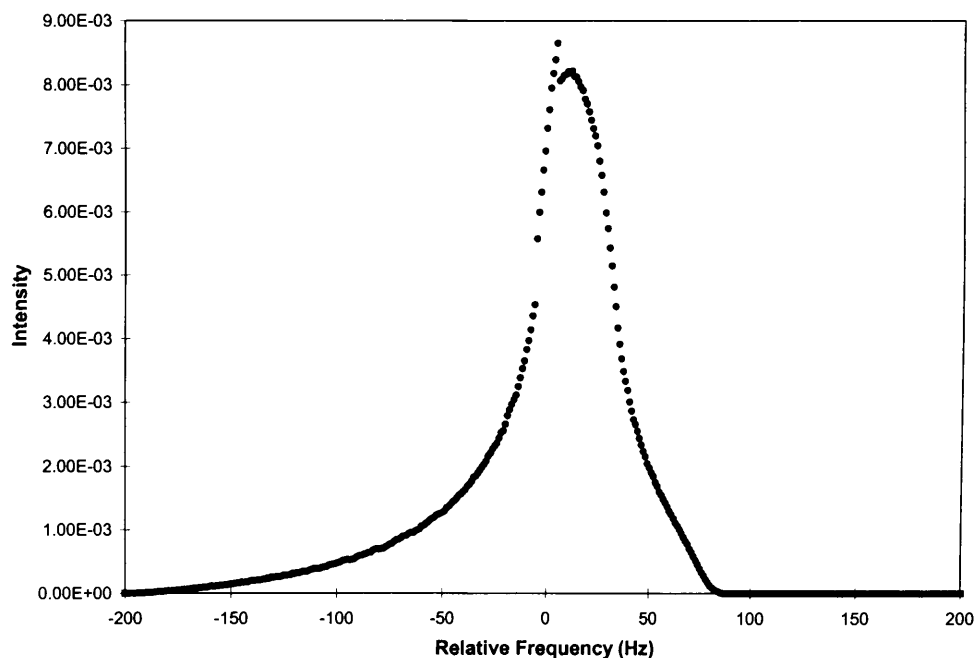


Figure 4.4. 8: Line-shape generated from the cubic unit-cell, where the magnetic field is orientated along $x=y=z$. The spheres in this simulation had zero susceptibility whilst the fluid had a volume magnetic susceptibility of -9.1×10^{-6} , the magnetic field was 1 Tesla.

As expected, the orientation of the unit cell with respect to the applied magnetic field does affect the line shapes. Since in the real random bead/fluid packs an average of all orientations would be expected, a scheme by which various random orientations of the applied magnetic field

yielding several line-shapes that could be averaged was designed. In addition to these techniques aimed at giving a true average between all orientations of the applied magnetic field expected in real bead/fluid packs, a hexagonal close-packing unit cell was also used for comparison with the cubic close pack to see if this made any significant difference to the shape and indeed width of the resulting spectrum. The first and most obvious scheme used, was an even weighting between the three orientations previously described in this chapter, though more elaborate techniques were also adopted. Of these, the two main techniques pursued in obtaining a true average of all possible H_0 orientations, were, a polar co-ordinate method and a random walking method about the surface of a sphere.

The polar co-ordinate method depended on two random angles, one about the z-axis in the xy plane and the other an angle to the z-axis. These together with a fixed radius were expected to generate random H_0 orientations, and when the sum of many line-shapes were added an accurate simulation was expected. However, it was found that this technique is extremely biased towards the z-axis resulting in a line-shape dominated by the “face” type line-shape. This can most easily be compared to a globe where lines of longitude and latitude represent α and θ , and the points of intersection represent the possible orientations about the origin that H_0 can take. The key observation to make, is that these intersections are positioned far more densely about the poles than at the “equator” region of the globe resulting in z axial domination.

To overcome this problem and to obtain an accurate range of H_0 orientations a random walk method on the surface of a sphere was adopted. This was achieved by randomly selecting a starting point on the surface then randomly selecting a rotation of 5° about either the x or y axes and in either a positive or negative direction. The field and line-shape would then be calculated for this orientation of the applied magnetic field and for each orientation of the applied magnetic field generated by the random walk. It was found that a minimum of around 500 steps was necessary to achieve orientations covering the complete sphere. This was checked by calculating the cosines of the angle between the field and the axis (for the x, y and z axes), where the sum should tend to zero for each axis.

This technique was adopted for the cubic packed system, shown above, and a hexagonal close-packed system, the optimal close pack for spheres. The resulting line-shapes are given below.

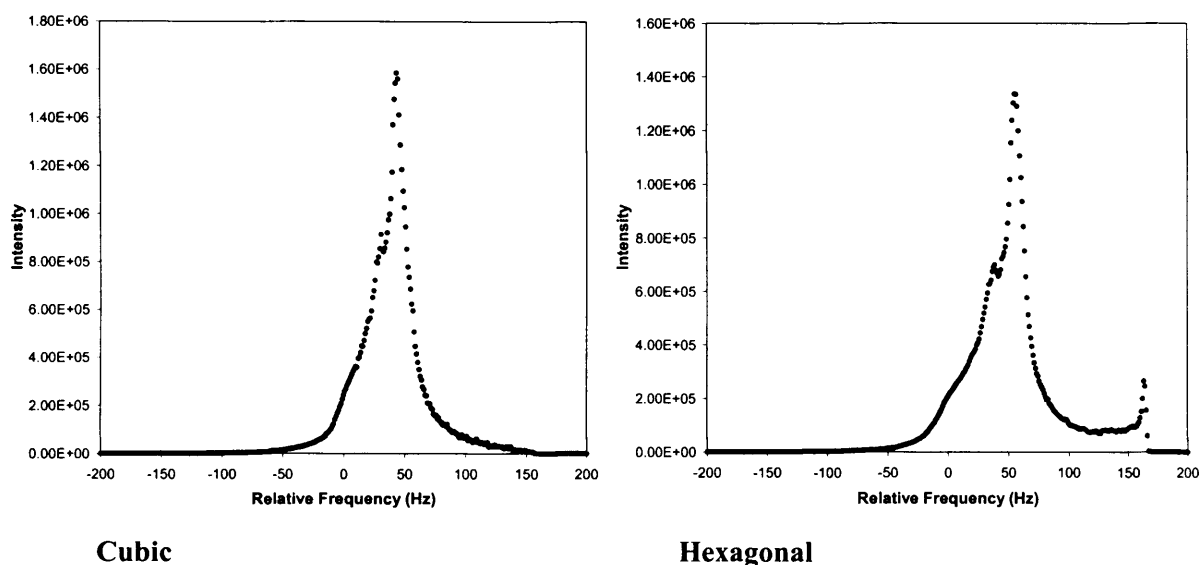


Figure 4.4. 9: Line-shapes generated using the step technique to ‘randomise’ the orientation of the magnetic field for cubic pack and hexagonal close-pack models.

A final simulation was performed using a random packing structure that allowed for a range of bead radii. Since the packing was random, this model should be independent of the direction of the magnetic field H_0 . The programme to do this is given in the appendices, though the technique was not adopted since the CPU demand was too high and time consuming for the hardware available.

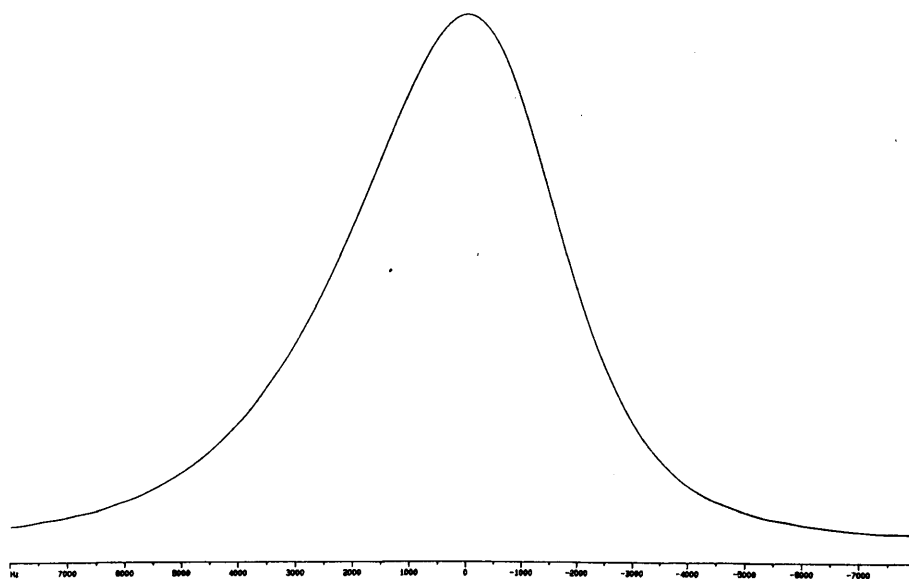


Figure 4.4. 10: Line-shape measured on a Bruker ARX NMR spectrometer operating at a proton resonance frequency of 250 MHz. The sample is water in a close pack of spherical beads whose radius ranged between 1 and 37 μm .

When the line-shapes generated by these simulations are compared to a real line-shape (fig.[4.4.10]) measured from one of the bead packs, those generated by the stepping technique are similar though the cubic close-pack where H_0 is orientated in $x=y=z$ is the closest and is also the simplest structure to programme. As such this was the chosen pack and orientation used for the diffusion experiments.

4.4.5 Diffusion Experiment Simulation.

Once the three-dimensional magnetic field maps were created it was possible to simulate the self diffusion of a spin through the system and calculate the signal attenuation due to self-diffusion for a given NMR pulse sequence with specified parameters. In this section a three dimensional random walk of spins through the bead/fluid matrix is discussed with relevance to the OE-CTPG pulse sequence adopted for all experimental work for reasons described earlier in this chapter.

The first objective in designing the simulation was to obtain a reliable random number generator since the whole basis of the simulation would depend on this. Most UNIX workstations have at least one of these function built into their architecture. One distinct advantage this built in function has over a coded routine is the speed that it takes to execute: this is a vital point since the function would be called millions of times during a simulation. The internal function in Solaris (the operating system of the Sparcstations used for all of the simulations) called *rand*, and for integrity, this was compared to a routine for a random number generator taken from *Numerical Recipes*²⁰. The programme, which incorporates both of these functions performs a simple one-dimensional random walk, where a molecule starts at the origin. The random number generator is then called, where, a number between 0 and 1 is produced. If this number is less than 0.5, the molecule moves one step in the negative direction, if the number is greater than or equal to 0.5 it will move in the positive direction. This process is repeated a specified number of times, and when completed, the final co-ordinate of the molecule is recorded. This process is then repeated for a specified number of molecules. The resulting distribution should be a normal, where the width of the distribution is dependant on the number of steps, and the smoothness dependant on the number of molecules the process is repeated for. A typical pair of the distributions generated from the two random number generators are shown in fig.[4.4.6]. From these it is clear that the "built in" function *rand* produces a more typical normal distribution than the function taken from *Numerical Recipes*, which appears to plateau in a most unsatisfactory manner. Using these tests, the internal function was selected as the random number generator for the diffusion simulations. This had an added benefit since this function also required less CPU time.

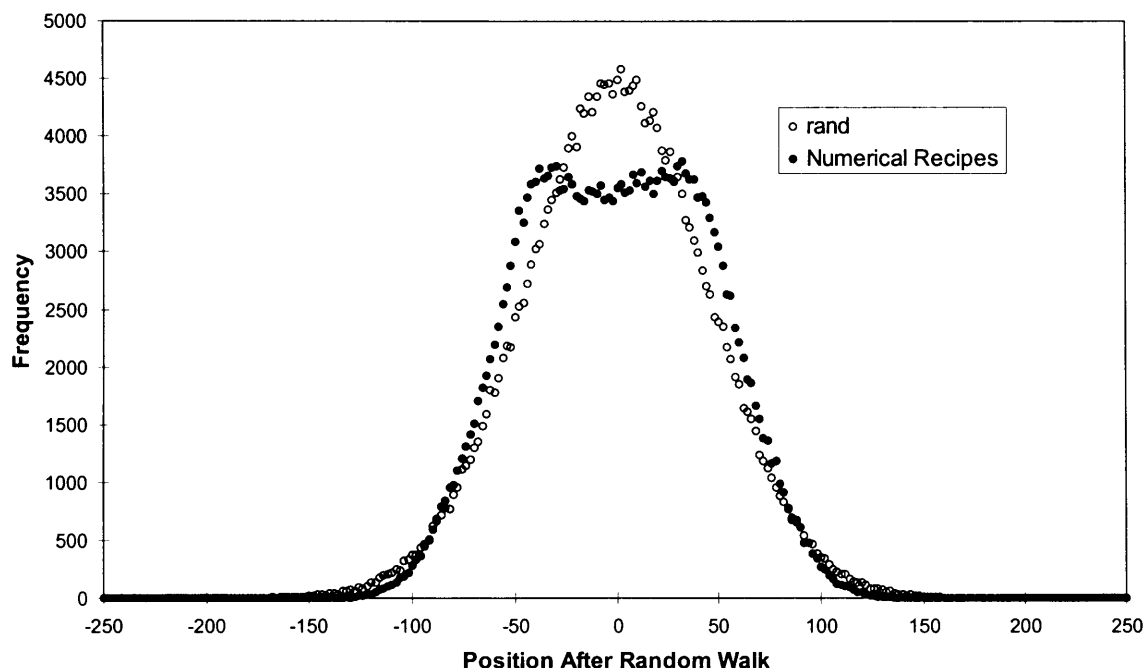


Figure 4.4. 11: Random walk distributions using two random number generators. The random walk model makes 2,000 steps for each of the 250,000 spins. Since the distribution should be normal, the “rand” function is shown to be fair, while the function taken from Numerical Recipes is shown to be unreliable.

The next stage in the development of the simulation was to model the random walking motion of the spins throughout the bead/fluid matrix. The first step was to select a random starting position for the spin. This was achieved by multiplying three random numbers by 100 (the dimension of each side of the three-dimensional array containing the unit cell) for the x, y and z co-ordinates respectively. Once these provisional starting co-ordinates were selected, they were examined to see if they resided in the bead or fluid component of the matrix. The examination process was done by comparing the distance between the provisional starting point and each sphere centre, with the radius of the sphere. If the co-ordinates were inside the beads, the position was rejected, since the fluid molecules did not penetrate or diffuse in these regions, and new co-ordinates were selected and checked by the same processes until a position was found to be present in the fluid. The spin then underwent three one-dimensional random walks in the three main axes in the same manner as that used in testing the random number generators. The magnitude of each step is determined by a variable which is defined at the initial input of the executable. The smaller this step, the more often calculations will have to be made, so a balance between CPU time and accuracy must be made. These small steps are made in real distances as opposed to co-ordinates of the three-dimensional array. This gives rise to a far more flexible and far more accurate model. Once this provisional step has been performed, it was examined using the subroutine that

differentiates between regions inside and out of the spheres as previously mentioned. Should this new position reside inside one of the beads of the unit cell, then the programme calls a subroutine which will repeat the random step though in only two of the three directions. There are of course three combinations of these axes and should the first fail, a second combination is tried and so on. If it is found that all of these two axes motions are not accepted, three single axis steps are attempted in turn. The final option should all seven attempts fail it to leave the spin to rest for this step and move on to the next. Another situation that may arise is when the step takes the spin outside of the unit cell. For the cubic close packed model, this motion is akin to moving into an identical unit cell and this is the action taken by the programme in such a situation.

The final stage of the simulation design involves calculating signal attenuation resulting from self-diffusion in the bead/fluid matrix. In the case of all simulation performed in this thesis, the programme was optimised for the OE-CTPG pulse sequence, though only very simple modifications need be done to apply the simulation to other pulse sequences. The pulse sequence consists of four main periods. These can be split into two categories, the first is where the net magnetisation vector is aligned along the z-axis which is the case during the periods t and $(T-t)$, the second is where the vector is in the xy plane. During the periods spent along z, the spins are assumed to move freely without any change in their phase since no precession occurs. However, in the periods τ , the spins do precess because of the field variations and it is therefore necessary to calculate their change in phase for each step. To calculate the change in phase accurately, it is necessary to know the precise field experienced by the spin during each step. Since the exact location of the spin will lie between the array co-ordinates (for which the magnetic field has previously been calculated) this must be determined by use of an algorithm. The one adopted for use in this simulation was a tri-linear interpolation algorithm adapted from a multi-linear interpolation routine, again from *Numerical Recipes*²¹. Once the field strength has been determined, the phase accumulated for this step may be calculated using eqn.[4.14]

$$\phi = \gamma t B_{local}. \quad [4.14]$$

The total phase shift accumulated in the first period τ will be the sum of the phase shifts for each step. Following this time interval the magnetisation is rotated into the z-axis, where no phase is accumulated though the spin will still diffuse as previously discussed. At the end of this period the net magnetisation vector undergoes a further 90° R.F. pulse and is rotated back in to the xy plane. Since this is the second 90° pulse, the direction of the amplitude of the phase accumulated is reversed to mimic the effects of a 180° R.F. pulse.

4.5 Experimental Methods.

4.5.1 Glass Bead Packs.

All of the NMR experiments were performed using a Bruker ARX spectrometer operating at a proton resonance frequency of 250MHz. The sample was maintained at 25°C throughout all of the experiments.

The systems that were measured were packs of glass beads with different ranges of bead diameter which were immersed in a variety of solvents. The ranges of bead diameter used in the experiments were: 1-37 μm , 44-53 μm , 74-88 μm , 140-170 μm , 210-250 μm , and 297-354 μm . The ranges of bead size were produced by sieving, though the relative distribution of bead size within the range was not known. The solid material of the beads also varied with bead size so that volume magnetic susceptibility also altered throughout the range of beads. The mean mass magnetic susceptibility for each range was measured using a commercial magnetic susceptometer and their density measured volumetrically. From these, the volume magnetic susceptibility was calculated; the values of which are given in table[4.5.1].

| Size range / microns | 1-37 | 44-53 | 74-88 | 140-170 | 210-250 | 297-354 |
|--|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Volume magnetic susceptibility (SI units) | 3.8×10^{-5} | 9.7×10^{-6} | 1.2×10^{-6} | 9.7×10^{-7} | 1.7×10^{-6} | 1.1×10^{-6} |

Table 4.5. 1 Volume magnetic susceptibilities of the glass beads for each size range.

Three different solvents were used as pore fluids for each of the bead size ranges: pure water, cyclohexane and acetone. The volume magnetic susceptibilities for each of the solvents were extracted from the literature (CRC Handbook) while the self-diffusion coefficients were measured using the OE-CTPG pulse sequence, as described earlier and shown in fig[2.2.8]. All of these values are shown in table[4.5.2]. The samples were made by placing the glass beads into a dry 5mm NMR tube, and then adding the solvent so that all of the beads were submerged to minimise any capillarity effects. The amount of sample made was large enough to ensure that the bead/solvent pack extended well beyond the sensitive region of the transmit/receive coil of the NMR spectrometer. The pack was then sonicated for several hours to maximise the tightness of the packing, and to eliminate any air bubble from the system. A separate bead pack was used for each of the three solvents used as pore fluids.

CHAPTER 4 NMR DIFFUSION MEASUREMENTS IN HETEROGENEOUS SAMPLES WITH LARGE SUSCEPTIBILITY VARIATIONS

| Pore fluid | Water | Acetone | Cyclohexane |
|--|-----------------------|-----------------------|-----------------------|
| Volume magnetic susceptibility / (SI units) | -9.1×10^{-6} | -5.8×10^{-6} | -7.9×10^{-6} |
| Molecular self-diffusion coefficient / m^2s^{-1} at 25°C | 2.30×10^{-9} | 4.52×10^{-9} | 1.61×10^{-9} |

Table 4.5. 2 Volume magnetic susceptibilities and molecular self-diffusion coefficients of the pore fluids.

The OE-CTPG pulse sequence (fig.[4.5.1]) was used with δ equal to $250\mu\text{s}$, and Δ ranging from $253\mu\text{s}$ to 1.45s . The extent of the range of Δ used varied depending on the SDC of the solvent, though the number of data points sampled was constant for each. The 90° pulse length was calibrated for each of the samples, and found to be approximately $8.5\mu\text{s}$ in each case. The spectrometer was shimmed using a tube of $\text{H}_2\text{O}/\text{D}_2\text{O}$ mix of similar volume to the bead packs to minimise any constant gradient present, though in reality this would be much smaller than the internal magnetic field gradients arising from the susceptibility variation. A 32 step phase cycle, as described in fig.[4.5.1], was used in acquiring the signal for each of the diffusion times.

The magnitude of the centre of the echo was measured and plotted against diffusion time for each bead set. These graphs are shown in fig[4.6.1] for water, acetone and cyclohexane respectively. Fig.[4.6.1] also shows the echo magnitude of pure water (i.e. without any beads), where the amplitude is constant throughout since there are no magnetic field variations (inherent or applied) to give rise to signal attenuation from diffusion, to within experimental error. The time domain data was also Fourier transformed to give the line shapes of the packs. The intensities of these lines were also measured both phased and in magnitude mode and found to give values similar to those measured directly from the echo as one might expect. The widths of the spectral lines were also measured to give an indication to the range of the magnetic field resulting from the susceptibility variations in the bead/fluid systems.

4.5.2 Lung.

All of the NMR experiments were performed using a Bruker DRX spectrometer operating at a proton resonance of frequency 400MHz . The instrument was maintained at 25°C throughout all of the experiments.

The lungs were excised from a rat and in the initial experiment inflated and tied off by the trachea. The lungs were then suspended in a 10mm NMR tube with a small pool of $\text{H}_2\text{O}/\text{D}_2\text{O}$ in the bottom to maintain a moist atmosphere, thus reducing the potential for the lungs to dry during the

CHAPTER 4 NMR DIFFUSION MEASUREMENTS IN HETEROGENEOUS SAMPLES WITH LARGE SUSCEPTIBILITY VARIATIONS

experiment and reduce error. The OE-CTPG pulse sequence was employed to make the measurements using the same 32 step phase cycle adopted for the glass bead experiments. The experimental timings where $\tau = 1\text{ms}$, $t_3 = \mu\text{s}$ to 800ms in steps of 5ms and $T = 1\text{s}$, corresponding to a range in diffusion times of 1ms to 801ms . The dephasing period in the xy plane is longer in this experiment than with the glass beads since the internal gradients appear not to be so strong despite the increased field strength.

The initial experiment looks at the lung with different volumes of air in the alveoli to see if the experiment is sensitive to this changing environment. After the first measurement on the inflated lung, the lung was punctured and the measurement repeated. The experiment then repeated the measurement on the lung after it had been deflated by vacuum and again after a period of 10 hours.

The second experiment used exactly the same techniques adopted in the initial experiment, though the lung was not initially inflated and the experiment repeated many times at regular intervals. This experiment was intended to observe the change in the lung's physical structure over time by means of change in magnetic susceptibility gradients.

4.6 Results.

4.6.1 Glass Bead Packs.

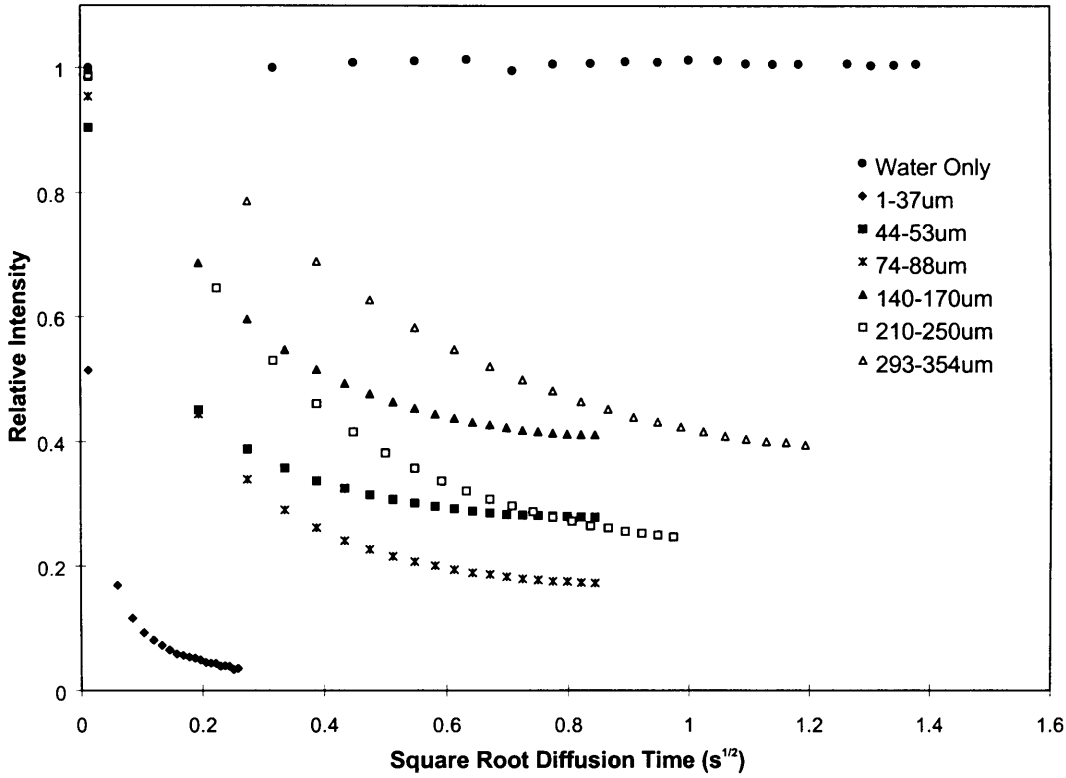


Figure 4.6. 1: Signal intensity as a function of square root of diffusion time (Δ) for glass bead packs with beads within the size range shown. The pore-space fluid was water.

In fig.[4.6.1], the signal intensity from a sample of pure water does not vary with increasing diffusion time with the OE-CTPG pulse sequence. This demonstrates that the pulse sequence eliminates the effects of both longitudinal and transverse relaxation, and that any background magnetic field is of such small magnitude as to have a negligible effect on the attenuation of the signal from spin diffusion when δ is short. However, when the bead/fluid packs are measured, the internal magnetic gradients within the samples are strong enough to induce diffusion related attenuation even for the short value of τ used in this study. As the experimental diffusion time is increased, the amount of attenuation also increases to an asymptotic level, as expected.

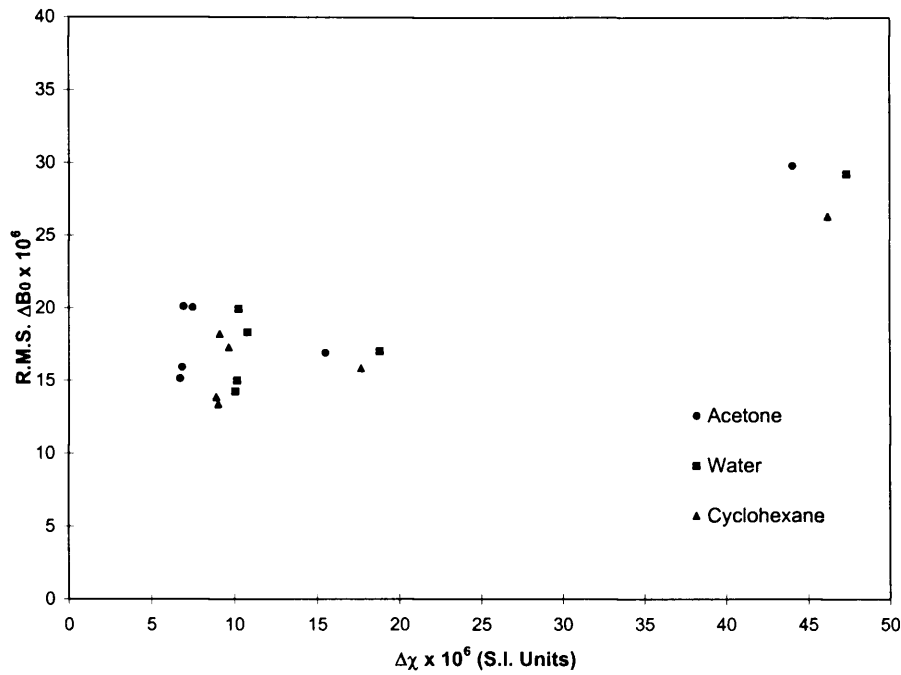


Figure 4.6. 2: Root mean square (RMS) field deviation estimated using eqn.[4.1.11]. The RMS field distribution is related to the asymptotic signal intensity in decay curves such as those seen in Fig. 3. There appears to be little relationship between the estimated RMS field deviation and the magnetic susceptibility contrast between the pore fluid and the solid material ($\Delta\chi$).

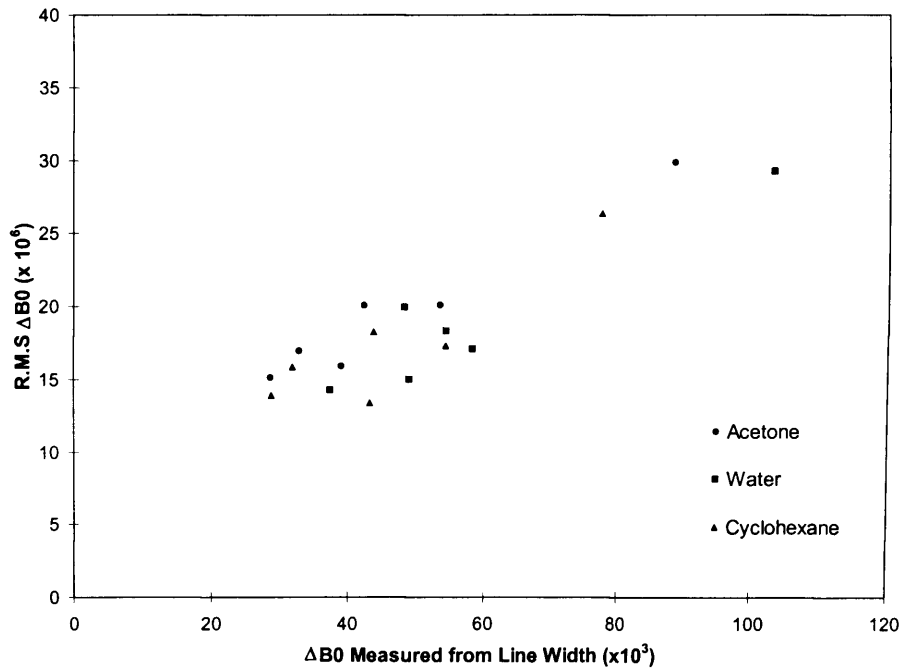


Figure 4.6. 3: Magnetic field deviation estimated using eqn.[4.1.11] compared to that measured directly from the sample half-height line-width.

CHAPTER 4 NMR DIFFUSION MEASUREMENTS IN HETEROGENEOUS SAMPLES WITH LARGE SUSCEPTIBILITY VARIATIONS

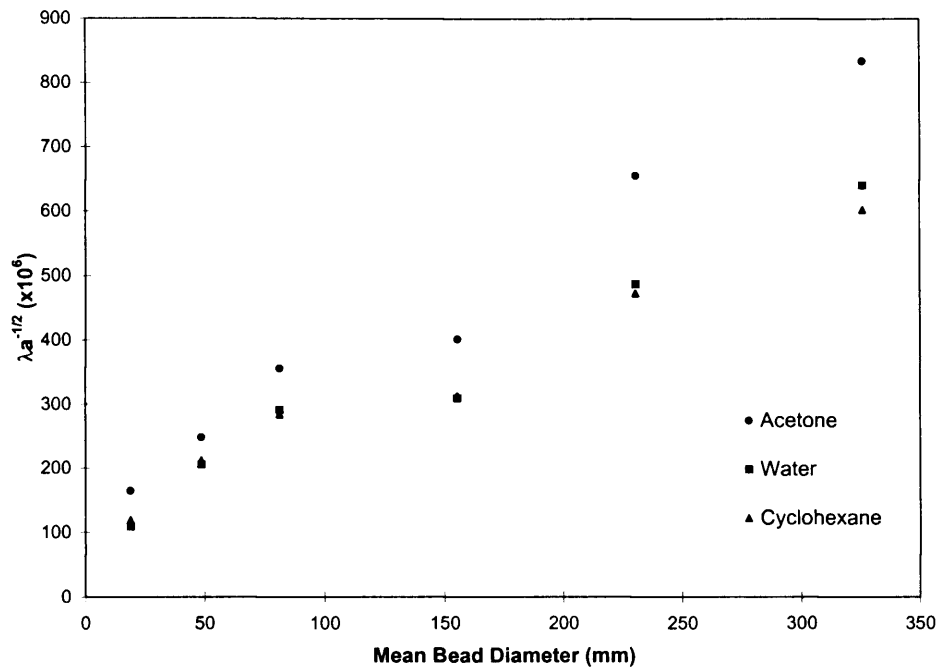


Figure 4.6. 4: Size parameter for variations in the magnetic field strength ($\lambda a^{1/2}$) estimated from eqn.[4.1.11] for all bead packs and pore fluids. If λ is assumed to be the bead diameter, the linear regression to all these data gives a value for a of approximately 22.

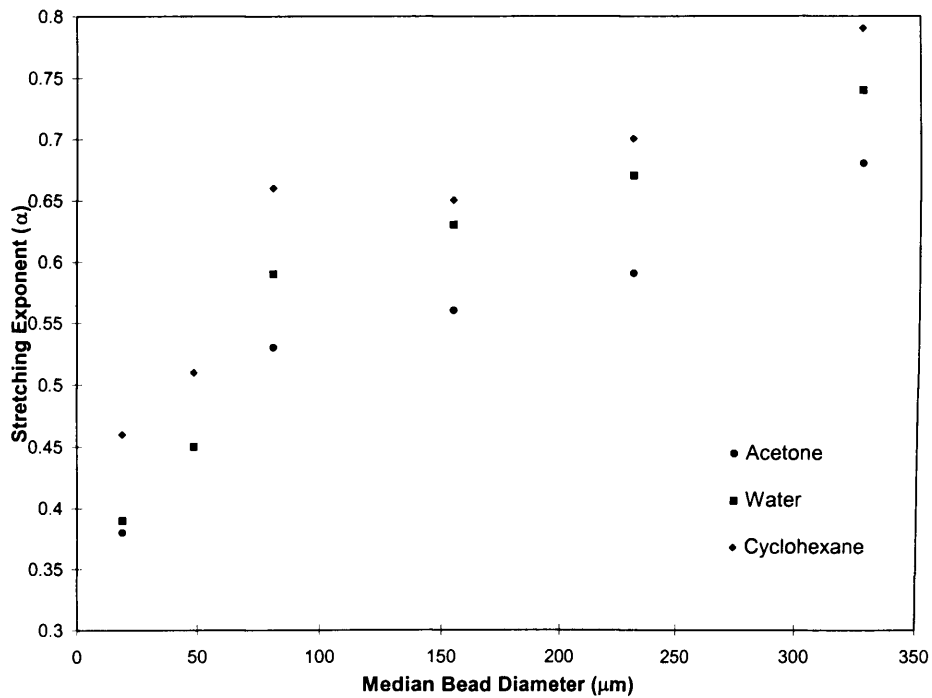


Figure 4.6. 5: Stretching exponent (α) in the stretched exponential form for all bead sizes and pore fluids.

The magnitudes of deviations in the magnetic field due to variations in susceptibility in a particular geometry do not vary with changes in size of the elements making up that geometry²². Thus, for monodispersed packings of glass beads, the square root of the field variation parameter, $\int_V \Delta B_0^2(\mathbf{r}) d\mathbf{r}$, should be independent of bead size but should be proportional to the magnetic susceptibility contrast between the bead and the pore fluid ($\Delta\chi$). The root mean square (RMS) field deviation is estimated from one of the fit parameters in eqn.[4.1.10]; it is related to the asymptotic echo attenuation for very long diffusion times, when the diffusing pore fluid has sampled a representative volume of the whole pore space. Figure [4.6.2] shows this parameter plotted against the measured $\Delta\chi$ for all beads and pore fluids used. Whilst there is some evidence for increasing field deviation at higher susceptibility contrast, the relationship is not very strict. However, the use of different pore fluids can be seen to have little influence. For the largest beads, estimation of the asymptote may be hampered by the fact that the diffusing molecules are unable to sample a representative volume of the pore space because the diffusion path length at the maximum diffusion time used is only on the order of 100 μm , considerably shorter than the large bead diameters.

When the field deviation present in these samples was measured directly from the line width of the NMR signal, as shown in fig.[4.6.3], the correlation with the field variation parameter is much stronger. For a Gaussian distribution of field strength, the RMS field deviation is 0.27 times the half-height line-width. From fig.[4.6.3], the RMS field deviation for our samples was estimated by linear regression to be 0.34 times the half-height line-width.

The size parameter, $\lambda/a^{1/2}$, which is estimated from the rate of decay towards the asymptote, should be independent of both the pore fluid and the solid matrix magnetic susceptibility. In fig.[4.6.4], $\lambda/a^{1/2}$ is shown plotted against the median bead diameter for all bead sizes and pore fluids. The pore fluid diffusion coefficient varies by a factor of almost three, but this has little influence on the measured size parameter. Whilst the estimated size parameter increases monotonically with bead diameter, it does not increase in proportion to it - there is a non-zero intercept at very small bead sizes. Linear least-squares regression of a straight line to all these data gives a value for a of 22.1, which is between the first and second decay modes for all the geometries in table[4.3.1].

The stretching exponent, α , is shown in fig.[4.6.5] plotted against the median size of the beads, for the three pore fluids. The stretching exponent decreases with decreasing bead size, indicating a wider distribution of decay modes for the smaller beads. Note also that the stretching exponent is lowest for the pore fluid with the highest diffusion coefficient.

4.6.2 Lung.

We have seen that for a well shimmed system with no magnetic field gradient, the OE-CTPG pulse sequence gives rise to no deviation in signal intensity since no contribution to signal attenuation may appear from diffusion effects. In the graph below, figs.[4.6.6] and [4.6.7], the signal very distinctively decays with increasing diffusion time with the same shape as observed in the packed glass bead experiments. There is a clear difference in the asymptotic value for the different measurements that should be reflected in the values extracted from the fitting of the data to the model described by eqn.[4.1.11].

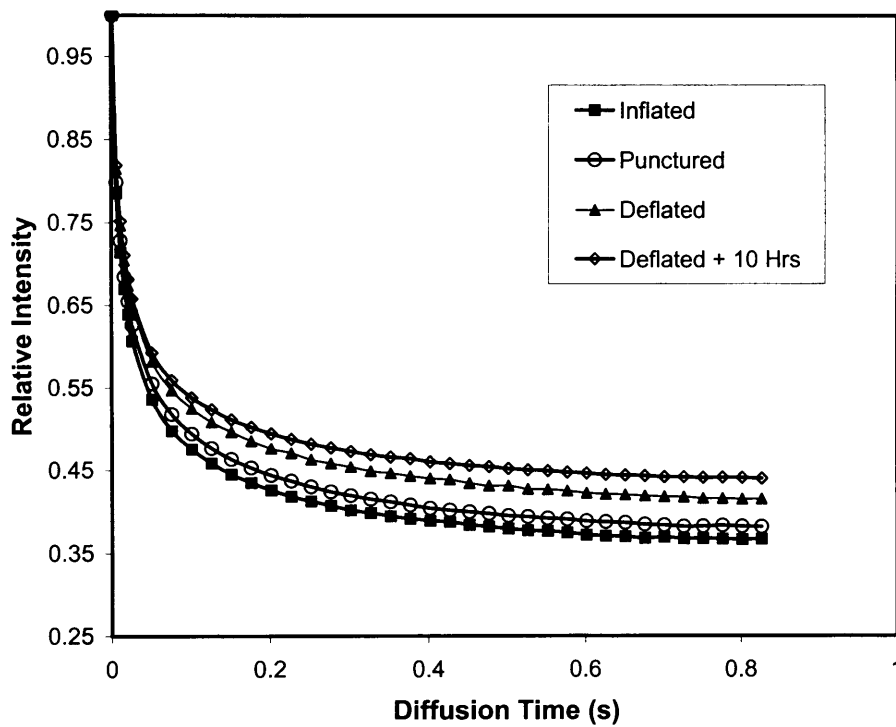


Figure 4.6. 6: Signal intensity as a function of diffusion time (Δ) for rat lung. The sample was originally inflated with air before being punctured for the second measurement and deflated by vacuum for the final two measurements.

The shape of some of the decay curves in the second experiment (fig.[4.6.7]) is a little uneven. These irregularities appear in the earlier measurements; the most noticeable being the very first. This irregularity will affect the accuracy of the parameters extracted from our model.

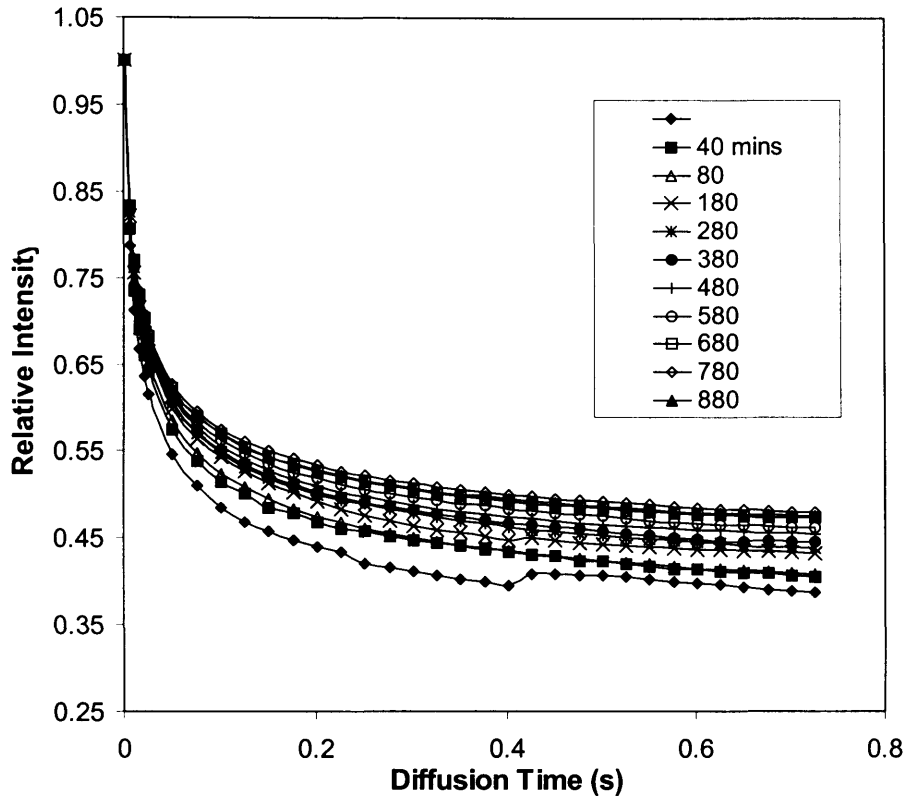


Figure 4.6. 7: Signal intensity as a function of diffusion time (Δ) for rat lung. The sample was not inflated or deflated by vacuum. Repeated measurements were made at regular intervals.

The results of fitting the data to our model are given in the following graphs, where the results for the first experiment, that of the initially inflated lung, are given on the left hand side, and the second experiment, performed over an increased time, are on the right.

For the root mean square (RMS) field deviation, there appears to be a decrease in the magnitude with increasing time for both experiments. The field deviation for the first measurement in the second experiment (shown by fig.[4.6.7b]) is noticeably lower than one might expect according to the trend. However, since the field deviation is related to the asymptotic value for signal intensity observed in fig.[4.6.7], it is clear the asymptote for this decay has the lowest value, which should give rise to the largest RMS value. This inaccuracy in the fit is likely to be an effect of the uneven decay curve seen for this measurement.

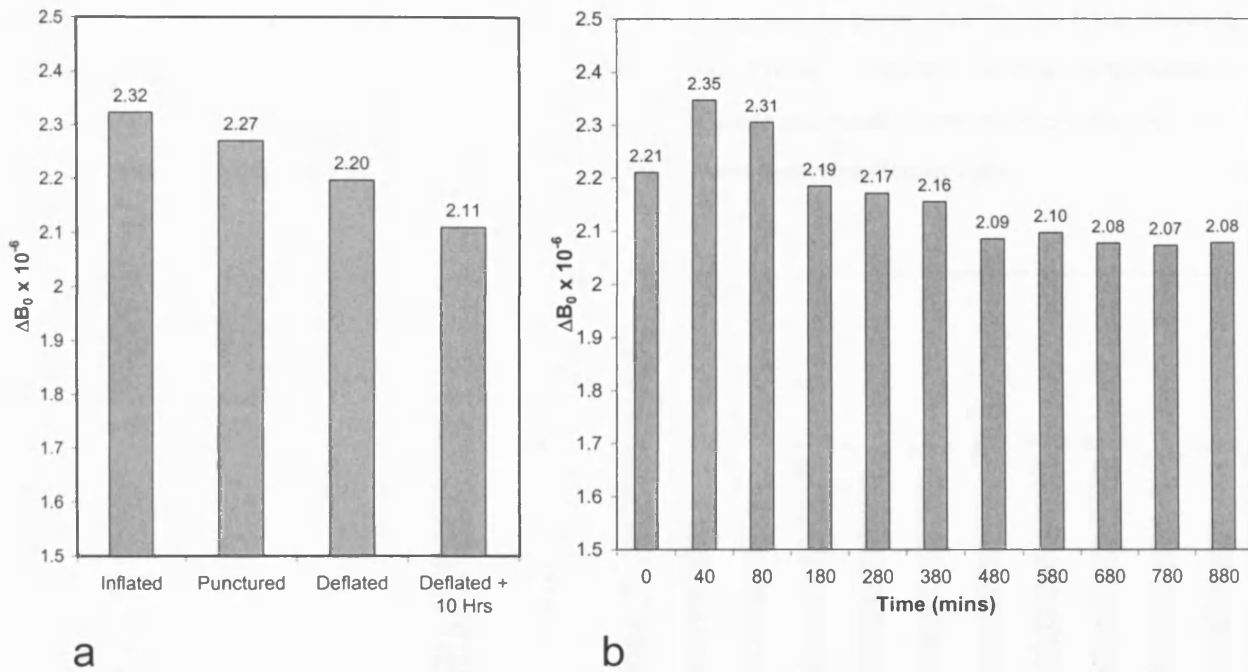


Figure 4.6. 8: Root mean square (RMS) field deviation estimated using eqn.[4.1.11]. The value of the field deviation is related to the asymptote of the signal intensity decays observed in figs.[4.6.6] and [4.6.7]. The trend appears to be a decrease in field deviation over time and with reduced air pressure.

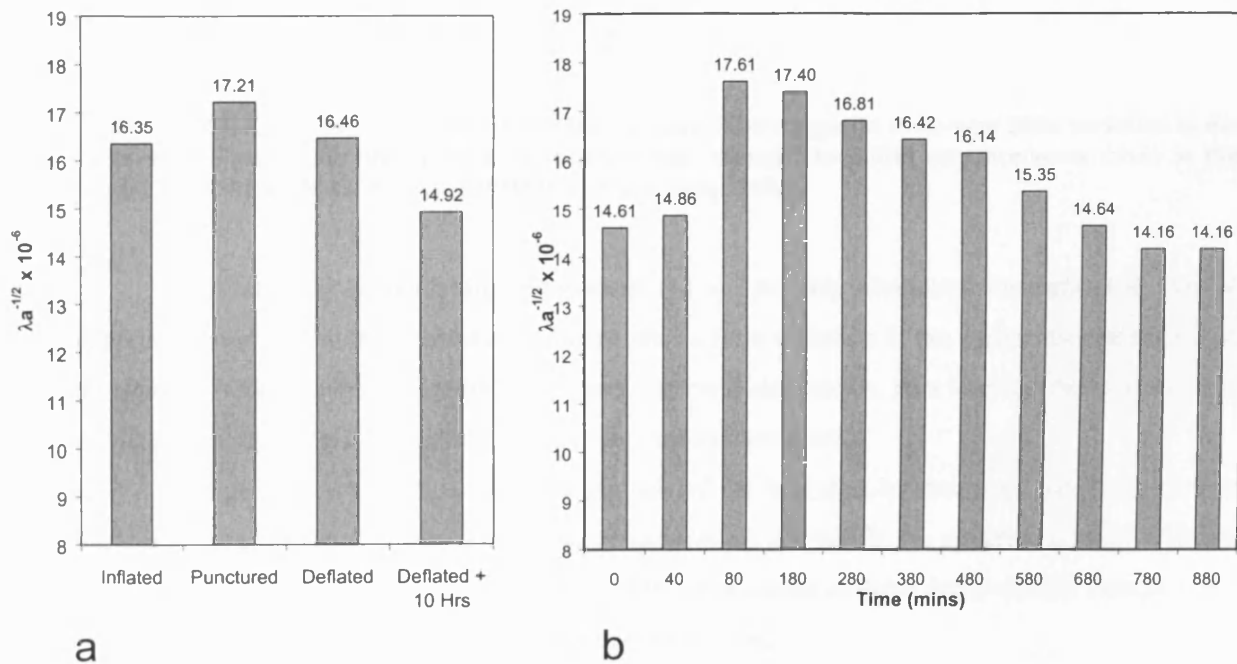


Figure 4.6. 9: Size parameter for variations in the magnetic field strength ($\lambda a^{1/2}$) estimated from eqn.[4.1.11] for rat lung. Unlike the glass bead experiments, we are unable to calculate an approximate value for a since the alveoli are not assumed to be of the same size over the experimental period. The general trend shows a decrease in the value of the parameter though early measurements do not appear to follow this trend.

For the characteristic wavelength / packing constant complex ($\lambda/a^{1/2}$), the trend shows a decrease in the value for both experiments. Since it is difficult to separate the two components of the complex (as could be done with the glass beads since the bead diameters were known), it is impossible to say with authority which of the two dictates this reduction in value.

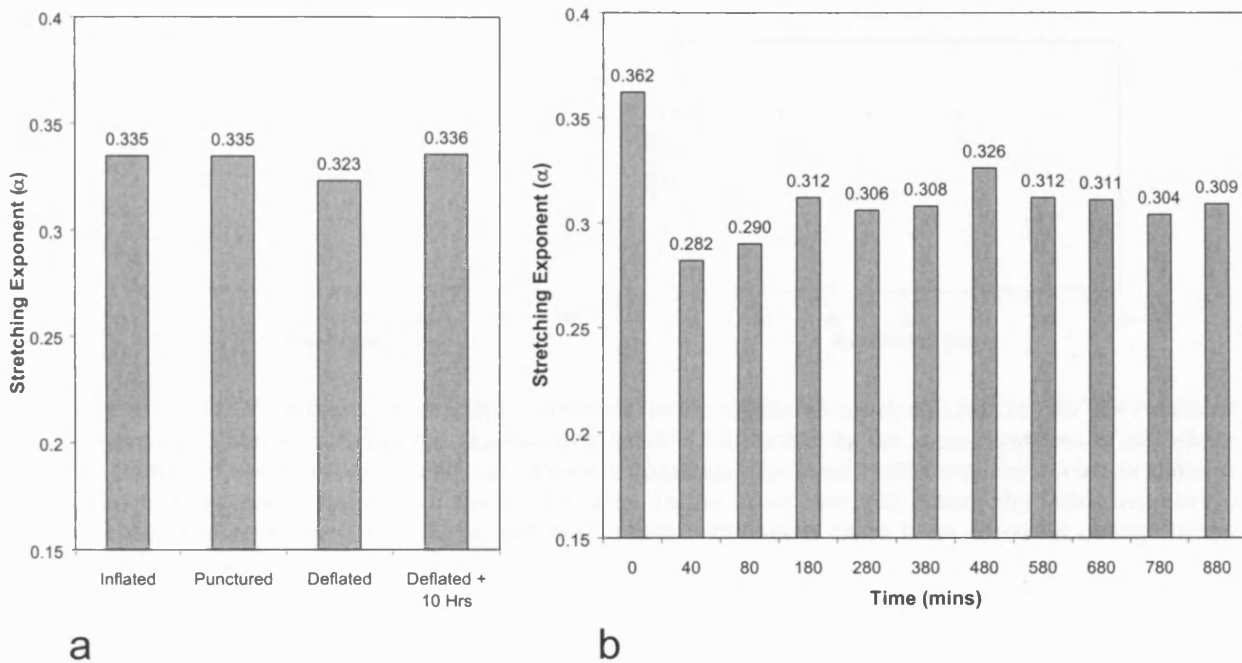


Figure 4.6. 10: Stretching exponent (α) for the rat lung. There appears to be very little variation in the value of this parameter throughout the experiments, though the initial measurements made in the second experiment (b) do deviate a little from the average value.

The value of the stretching component (α) is relatively consistent throughout the initial experiment and also in the second experiment after a little variation in the earlier measurements of the study. Unlike the bead experiments where the bead size varies, thus varying the value of the decay modes, the lung is thought to consist of a uniform structure.

The general observation from the analysis of the lung data by means of our model is that the initial measurements made in the second experiment do not fit the apparent trends. This, as previously discussed, is likely to be an effect of the unevenness of these decay curves, though other possibilities are covered in the following discussion section.

4.6.3 Computer Simulations.

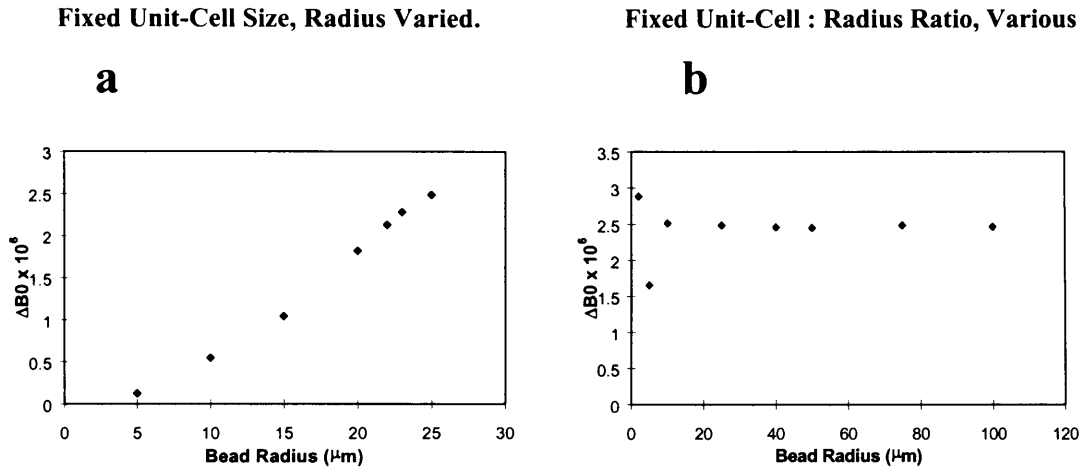


Figure 4.6. 11: Root mean square (RMS) field deviation estimated using eqn.[4.1.11] for the results of the computer simulations plotted against their relative bead radii. In the scenario shown in (a), where the centres of the beads are a constant distance apart and the bead radii vary, the deviation shows a clear proportional increase with the bead radius. In the other case, (b), where the beads are always touching and both the unit cell size and bead radius are varied, there is no apparent change in the deviation value.

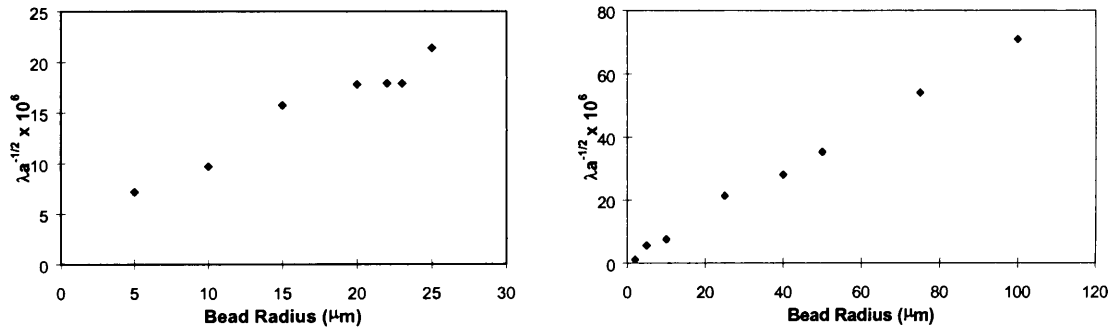


Figure 4.6. 12: The parameter that defines the *wavelength* for the variation in the magnetic field estimated from eqn.[4.1.11] plotted against the bead radius. In both scenarios, the parameter increases proportionately with the bead radius. If it is assumed that λ is equal to the bead diameter when the beads touch as in the fixed ratio example (b), the linear regression to all these data give values for a of approximately 8.2. However, it would be inaccurate to make the same assumption in the case where all beads do not touch for a constant unit-cell size (a) since the packing will vary as the bead surfaces do not touch. There will therefore be a dependence on the squared ratio between bead spacing and bead diameter if we assume λ to be constant. This ratio when touching is equal to 1, and when the radius is 10 times smaller than λ , then a will be inflated by a factor of 25. This would explain the reason why there is an unexpected increase in the y-axial expression with increasing bead side to the maximum, where the beads touch and the other scenario (b) is observed.

CHAPTER 4 NMR DIFFUSION MEASUREMENTS IN HETEROGENEOUS SAMPLES WITH LARGE SUSCEPTIBILITY VARIATIONS

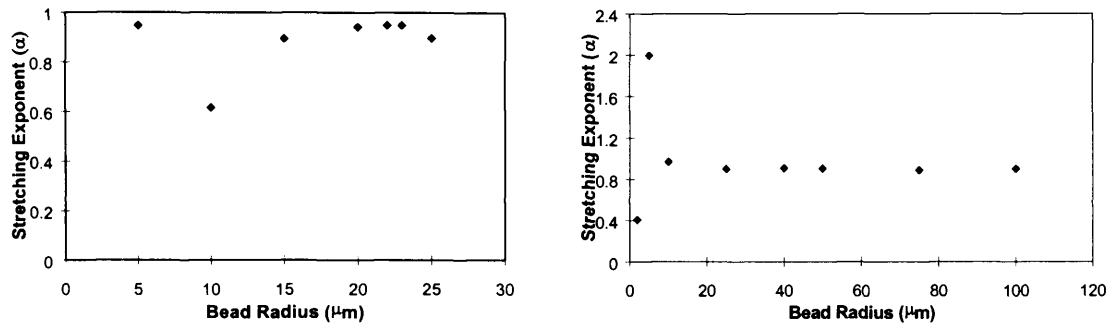


Figure 4.6. 13: The stretching exponent, α , for the two scenarios are shown against their respective bead radii. The value is approximately the same for all values.

Since the simulations offer an ideal system, where the beads are always of an identical size, it is possible to use the model described by eqn.[4.1.10], where the individual values for the decay rates constants, a_i may be extracted in preference to the stretching exponent α . The first two decay rate constants for each of the simulations are given below together with the related ratio.

| Fixed Ratio | | | | |
|------------------------------------|---------------------------------|-------|-------|-------|
| Unit-Cell Length (μm) | Sphere Radius (μm) | a_1 | a_2 | Ratio |
| 200 | 100 | 7.74 | 53.04 | 6.85 |
| 150 | 75 | 7.02 | 39.38 | 5.61 |
| 100 | 50 | 7.39 | 41.96 | 5.68 |
| 80 | 40 | 7.52 | 48.49 | 6.48 |
| 50 | 25 | 4.89 | 27.72 | 5.67 |
| Variable Ratio | | | | |
| Unit-Cell Length (μm) | Sphere Radius (μm) | a_1 | a_2 | Ratio |
| 50 | 25 | 4.89 | 27.72 | 5.67 |
| 50 | 23 | 7.12 | 30.98 | 4.35 |
| 50 | 22 | 7.45 | 52.99 | 7.12 |
| 50 | 20 | 7.33 | 39.40 | 5.37 |
| 50 | 15 | 8.75 | 46.20 | 5.28 |
| 50 | 10 | 10.03 | 46.74 | 4.66 |

Table 4.6. 1: The initial two decay rate constants and their ratio estimated using eqn.[4.1.10] for the aforementioned computer simulations.

The values for the modes do vary across the simulations and are slightly lower than those observed in table[4.3.1], on average, though the values for the ratio are relatively constant throughout. The fixed and variable unit cell simulations give values of 6.06 and 5.41 for the ratio, respectively. These compare to values of 8.99, 8.37 and 8.14 for planar, cylindrical and spherical geometries respectively (taken from table[4.3.1]).

4.7 Discussion.

From the analysis of the simulation data using our model, it is possible to observe trends in the values of our parameters for the root mean square deviation in the magnitude of the magnetic field, the size parameter λ/\sqrt{a} and the decay constants a_i (or the stretching exponent α).

From this computer simulation study we can suggest that magnetic field deviations do vary with the packing of the beads. It appears primarily that the tightness of the pack rather than the absolute size of the beads within it that contribute to this deviation. For instance, in fig.[4.6.11a] we saw an increase in the RMS field deviation proportional to the bead radius when the centres of the beads were fixed. This is because the model we adopted in eqn.[4.1.1] considers the change in B_0 across the entire sample volume. Therefore in the scenario where the volume of the sphere relative to the entire unit-cell (and therefore the entire sample since the sample consists of repeated unit cells) is increased, the RMS in the field deviation will also increase. There is of course a maximum point when the beads are touching and packed as closely as possible in a hexagonal close pack. At this point one might expect the maximum magnetic field deviation over volume since the changes in field will occur over the smallest distance relative to the bead radius. In the other set of simulations, this theme was addressed using bead packs where the beads all touched in a cubic structure. While this structure is not as tight a pack as the hexagonal close pack, it serves the purpose as demonstrated by the line shape simulations earlier in this chapter. In these simulations it can be seen in fig.[4.6.11b], that the RMS of the magnetic field deviation is relatively constant throughout, despite the wide range of bead sizes used. The only consistency between the simulations are the geometry and packing of the structure together with the magnetic susceptibilities of the two phases. This further enhances the notion that since the model measures the magnetic field deviations over volume, it is the tightness and probably the structure of the pack that dictates the magnitude of the deviations in magnetic field.

Like the magnetic field deviations, the size parameter, $(\lambda/a^{1/2})$, is shown to increase with the bead radius as one might expect. We have extracted a value for a of 8.2 for the tightly packed models where beads always touch their neighbour. This approximates to the first decay modes given in table[4.3.1]. As mentioned in the results section, we cannot make the same extraction of

the decay constant a_i for the fixed unit cell with variable bead radii data since λ is constant for all calculations. However, because λ is constant and the plot shows that the size term ($\lambda/a^{1/2}$) increases, it is safe to assume that the packing will be different in each case. In fact we can assume that as the space between the spheres surfaces increases, so does the value of the packing term a . This idea is supported by the results from the data analysis using eqn.[4.1.10] that are shown in table[4.6.1]. Here we see that for a common structure, where only the size of the solid spheres varies, the value for a_1 is comparatively constant and approximates to the value of a extracted using a linear regression from the size term ($\lambda/a^{1/2}$) (assuming that λ was equal to the sphere radius and the only variable). In comparison, the value of a_1 for the structure where λ is common and the closeness of the packing varies, increases inversely with the differing sphere radius. This is in accordance with the results observed in fig.[4.6.12a], where the term ($\lambda/a^{1/2}$) shows a proportional increase with the bead radius. Since λ is assumed to be equal to the unit-cell dimension, it is therefore constant, implying that the value of a decreases.

Despite these values for a_1 , the ratio of a_2/a_1 appear to be reasonably constant over the full range of sphere sizes for both types of sphere pack used in this study; a result that is reflected by the consistency of the stretching exponent measured from all data. The stretching exponent is an indication of the range of sphere/bead/pore radii present in the system. For hypothetical systems where the radius is constant throughout, such as those created for the computer simulation, the exponent value, α tends to a value of one. In systems such as the glass bead packs and lung, a range in the radius is present. This affects the standard deviation of the radius and hence the value of α . For lung, this approximated from 0.3 to 0.35 depending on the experimental lung and in the bead packs gave a range of values from around 0.4 to 0.8 depending on the range of bead sizes used.

In the glass bead packs, the value of α increases with the median bead radius for the packs used as seen in fig.[4.6.5]. This is probably because the smaller beads have the largest range of radii as a percentage of the median value (assuming a uniform distribution of bead radii).

With the lung we see very little deviation across the experimental time despite the morphological changes in the lung from deflation and decay. This suggests that despite changes that do occur in the sample, this does not affect the standard deviation of the size of the lung's chambers. As the lung branches out from the broncae, the capillaries become smaller until we reach the alveoli. This branching will contribute to the wide distribution of air pockets in the lung, giving rise to the value of the stretching exponent. The results suggest several explanations: the alveoli only contribute little to the signal decay, or the range in chamber size is initially so large that a deviation caused by inflation or decay is unlikely to affect the standard deviation of this, else the lung decays very quickly and in this state, the alveoli are unaffected by inflation.

Despite the relative insensitivity of the stretching exponent to the changes in lung, we see from figs.[4.6.8] and [4.6.9] changes both in the RMS magnetic field deviation and size parameter ($\lambda/a^{1/2}$).

The decrease in RMS field deviation occurs both for the deflating and decaying lung. The deflating lung may itself be affected by decay, but in either case it is assumed that the alveoli reduce in volume and therefore disturb the field less. This is equivalent to the computer simulation result given in fig.[4.6.11a], where smaller sphere size for a fixed unit cell reduces the RMS field deviation. It is of course also true that an inflated lung will have a greater distance between alveoli centres than a deflated lung, so the scenario probably lies between the two models studied in the computer simulations. Indeed, these assumptions are mirrored in the size parameter, ($\lambda/a^{1/2}$). In fig.[4.6.9] we see a trend to decreasing $\lambda/a^{1/2}$ over time, where the lung is either deflated and/or decaying. This decrease is probably a combination of affects observed in the computer simulations. As the lung deflates, the alveoli radius probably decreases as well, as previously discussed, with the distance between the alveoli centres, being equivalent to λ . This decrease in λ will decrease the value of ($\lambda/a^{1/2}$), but so will any increase in a . In fig.[4.6.12a], we see that as a bead/pore decreases in size, the value of a increases. It is these two affects that contribute to the change in ($\lambda/a^{1/2}$) that we see.

In the glass bead experiments we see the other type of model investigated in the computer simulations, where the spheres are assumed to be packed tightly in the same manner so that a constant value of a may be assumed. In fig.[4.6.4], we see that the value of the complex increases with median bead diameter. Again, as with the computer simulations, λ is assumed to be equal to this median diameter and a value of 22.1 is extracted for a . This value lies somewhere between the first and second decay modes given in table[4.3.1]. The deviations from linearity seen in fig.[4.6.4] may be a consequence of our use of the stretched exponential form to describe the data. For the cases described earlier with linear magnetic field gradients applied to well-characterised geometries, the decay modes are fixed, but the amplitudes of the individual components vary with the length scale of restriction and with the diffusion coefficient of the pore fluid. At shorter length scales or higher diffusion coefficients, the amplitudes of the more rapid decay modes increase at the expense of the slowest modes. The effect of this can be seen in fig.[4.6.5], where, for acetone, which has the highest diffusion coefficient, the stretching exponent is reduced, reflecting the more even distribution of the amplitudes of the decay modes, which may have resulted in the shift of the length scale parameter towards larger bead sizes seen in fig.[4.6.4]. It can also be seen in fig.[4.6.5] that the stretching exponent is lowest at the low bead sizes. This again is suggestive of the increased importance of the more rapid decay modes for small bead sizes.

The relationship between magnetic susceptibility contrast and the RMS field deviation seen in these samples is not fully understood and can not be compared directly to the computer simulations since spheres of only one magnetic susceptibility were assumed. An independent measure of the field deviation (the sample line width, see fig.[4.6.3]) correlated more strongly with the RMS field deviation than did the susceptibility contrast (fig.[4.6.2]). The expected linear relationship between susceptibility contrast and the magnitude of the field deviations did not emerge; this may be related to the different packing geometries of the different samples with their different ranges of bead sizes and susceptibilities.

4.8 Summary.

In this chapter I have presented both computer simulated and experimental work based on the molecular self-diffusion of a fluid in porous media that, for a fixed pore geometry, allows the determination of the relative sizes of packings of solid matrix materials²³. The computer simulations have been used to explain the experimental findings. To optimise these simulations in terms of accuracy and with reasonable computer CPU times, lineshapes were generated from the hypothetical matrices. The true randomness of the diffusion paths were tested using different random number generators and only when one that produced a Normal distribution was found, was it adopted.

We use only the diffusion behaviour in the internal susceptibility-induced magnetic field variations to determine the length scale of these variations, and hence we can deduce the length scale over which the solid matrix and fluid-filled pore space change. Clearly, a disadvantage of this method is that there must be a significant difference in magnetic susceptibility between the solid matrix and the pore fluid. In addition, the molecular self-diffusion coefficient of the pore fluid must be known, in the case of lung it was assumed that the fluid was water.

In our analysis, we have not distinguished between the effects of repetitions in the pattern of the magnetic field, and restriction of diffusion near the solid grain surfaces. These two factors will have a different effect on the detailed phase distribution of the nuclear spins, particularly at the shortest diffusion times. For example, as a spin diffuses to an adjacent pore space, and moves through a cusp in the magnetic field, the field strength is likely to vary smoothly. For a spin being reflected at a grain/pore surface, the magnetic field experienced is more likely to change abruptly. We have not taken details such as this into account in our analysis, but have merely assumed a Gaussian phase distribution throughout; this may be a major deficiency in the current theory.

The simple stretched exponential form was used to ensure robust fitting to the experimental data. This may not be most appropriate model to use; another form which takes into account the

CHAPTER 4 NMR DIFFUSION MEASUREMENTS IN HETEROGENEOUS SAMPLES WITH LARGE SUSCEPTIBILITY VARIATIONS

systematic shift in the distribution of decay with λ and D may be more appropriate. This can be addressed in future work.

- ¹ R. F. Karlicek and I. J. Lowe. *J. Magn. Reson.*, **37**, 75 (1980).
- ² X. Hong and W. T. Dixon. *J. Magn. Reson.*, **99**, 561 (1992).
- ³ D. W. McCall, D. C. Douglass, and E. W. Anderson. *Ber. Bunsenges. Phys. Chem.*, **67**, 366(1963).
- ⁴ E. O. Stejskal and J. E. Tanner. *J. Phys. Chem.*, **42**, 288 (1965).
- ⁵ T. J. Norwood. *J. Magn. Reson. A*, **103**, 258 (1992).
- ⁶ M. Kveder, G. Lahajnar, R. Blinc and I. Zupancic. *Magn. Reson. Med.*, **6**, 194 (1988).
- ⁷ H. Kolem, C. Goodrich, K. Ganesan, D. C. Ailion, A. G. Cutillo, S. Chen, A. H. Morris, S. Shioya and S. Watanabe. *Abs. Soc. Magn. Reson. Med.*, 8th Annual Meeting, 783 (1989).
- ⁸ G. Laicher, D. C. Ailion and A. G. Cutillo. *J. Magn. Reson. B*, **111**, 243 (1996).
- ⁹ T. A. Case, C. H. Durney, D. C. Ailion, A. G. Cutillo and A. H. Morris. *J. Magn. Reson.*, **73**, 304 (1987).
- ¹⁰ E. L. Hahn. *Phys. Rev.* **80**, 580 (1950).
- ¹¹ S. Meiboom and D. Gill. *Rev. Sci. Instr.*, **29**, 688 (1958).
- ¹² R.L. Kleinberg and M.A. Horsfield. *J. Magn. Reson.*, **88**, 9 (1990).
- ¹³ C.H. Neuman. *J. Phys. Chem.*, **60**, 4508 (1974).
- ¹⁴ J. C. Tarczozon and W. P. Halperin. *Phys. Rev. B.*, **32**, 2798 (1985).
- ¹⁵ B. Balinov, B. Jönsson, Per Linse and O. Söderman. *J. Magn. Reson Ser. A*, **104**, 17 (1993).
- ¹⁶ R. J. S. Brown and P. Fantazzini. *Phys. Rev. B*, **47**, p. 823 (1993).
- ¹⁷ W. E. Kenyon, P. I. Day, C. Straley, J. F. Willemsen. *SPE Formation Evaluation*, p. 622, (September 1988).
- ¹⁸ J. A. Glasel and K. H. Lee. *J. Am. Chem. Soc.*, **96**, 970, (1974).
- ¹⁹ S. Majumdar and J. C. Gore. *J. Magn. Reson.*, **78**, 41 (1988).
- ²⁰ S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *"Numerical Recipes in C"*, p.362 W.H. Press, Cambridge University press, (1994).
- ²¹ S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *"Numerical Recipes in C"*, p.123, W.H. Press, Cambridge University press, (1994).
- ²² B.I. Bleaney and B. Bleaney, *"Electricity and Magnetism"*, 2nd Edition, p. 141, Clarendon Press, Oxford, UK (1965).
- ²³ M. A. Horsfield, S. A. Clark and T. J. Norwood. *J. Magn. Reson. A*, **122**, 222 (1996).

Chapter 5

Measuring Diffusion in the Fringe Fields of Superconducting Magnets.

5.1 Introduction.

The volume inside a superconducting magnet where conventional NMR experiments are performed, is usually in the very centre of the magnet. It is in this region that the magnetic field is usually homogeneous and of a magnitude equal to that specified as the working field. As one moves out of this region, the homogeneity of the magnetic field deteriorates considerably; the field strength is reduced further away from the magnet centre. A plot of the magnetic field strength against distance from the centre of the bore of a magnet is given in fig.[5.2.1]. The measurement of the field for any given point is assumed to be exactly on the z axis since deviation from the centre along either the x or y axes will also contribute to field deviation. From fig.[5.2.1], it is possible to see that outside the homogeneous region of the magnetic field, the field strength changes continuously. This constant change gives a very strong magnetic field gradients (in the approximate range of 10 Tm^{-1} for imaging systems to 150 Tm^{-1} for narrow bore spectroscopic magnets). Magnetic field gradients of this strength are of great potential use both for NMR diffusion measurements^{1,2,3}, and more recently for NMR imaging, where the gradients have enabled imaging of samples with extremely short T_2 s, such as solids⁴. There have lately also been fringe field imaging studies of liquids penetrating solids^{5,6,7}, and the application of spin echoes to Stray-Field imaging⁸. In this chapter, relaxation-compensated pulse sequences, such as CTPG⁹, and non relaxation-compensated pulse sequences, such as Spin Echo¹⁰ are investigated and compared for their potential applications for measuring diffusion in very high, constant magnetic gradient fields.

5.2 Theory.

One requirement of great importance in standard NMR diffusion experiments is the uniformity of the gradient, especially when measuring diffusion in an imaging context³, where large volumes are often being measured. Should the gradient strength vary across the sample, it

will appear in the final analysis, that the ADC varies accordingly. It is therefore important to locate a region in the stray field where the magnetic field gradient is most linear. To do this, a plot of the field strength against distance from the centre of the magnet as shown in fig.[5.2.1], is required.

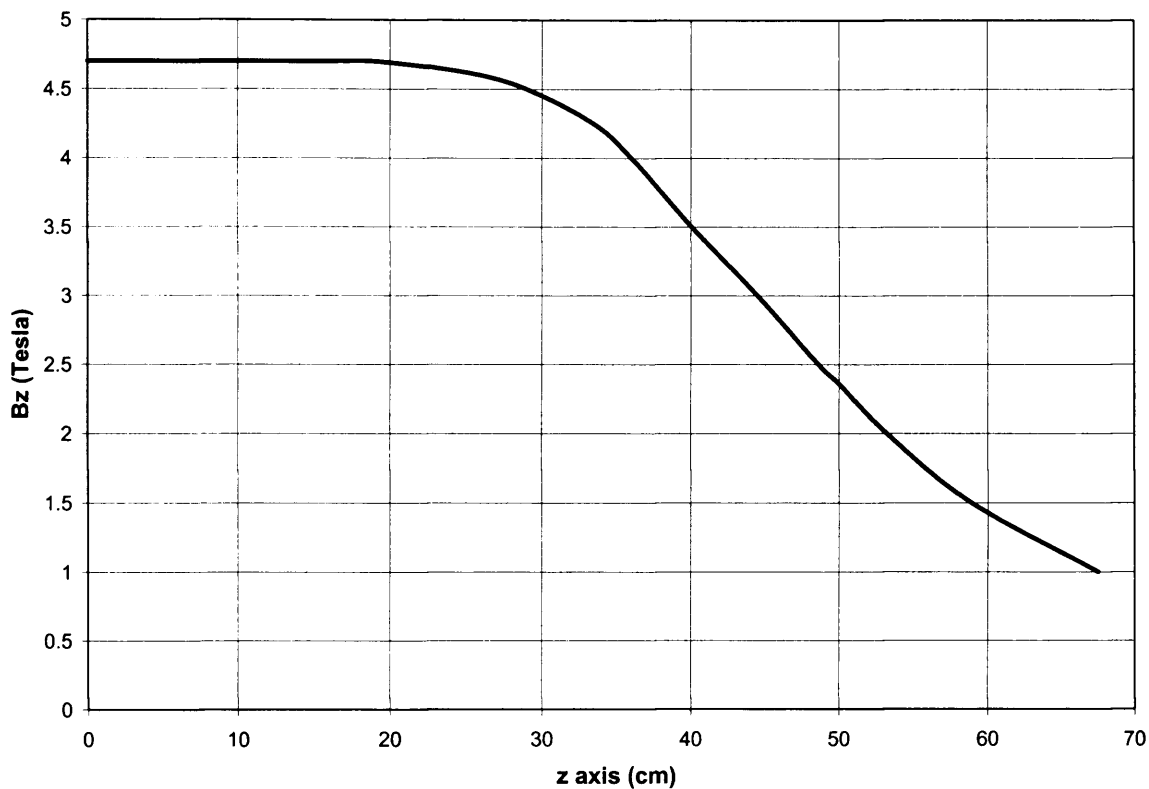


Figure 5.2. 1: Graph showing how the strength of magnetic field varies with distance from the homogeneous region in the centre of the magnet (0 cm), along the z axis. (Source: Oxford Instruments).

The magnetic field gradient is the rate with which the field changes with distance, and a plot showing this rate of change is obtained by taking the differential of the field plot. Such a plot is shown in fig.[5.2.2]. The region where the rate of change of magnetic gradient is zero is where the gradient is uniform, these are the minima of the graph shown in fig.[5.2.2]. From these graphs, the strength of the magnetic field at this point maybe found, from which the resonant frequency of proton for this magnetic field may be calculated. By placing the sample in this region of constant uniform magnetic field gradient with a transmit/receive coil of appropriate frequency, the ADC of a fluid may be measured by use of the appropriate pulse sequence.

In chapter 2, the relationship between signal attenuation resulting from self-diffusion, magnetic field gradient strength and the duration the net magnetisation vector spends in the xy plane, was discussed. Equations [2.2.5] and [2.2.7] show this relationship for the PGSE¹¹ and OE-CTPG pulse sequences respectively. From these equations, it is apparent that for strong field

gradients, such as those experienced in the fringe field, either the time component, or the SDC must be small if any signal is to be observed on acquisition. This is because relatively small movements will yield a large frequency change resulting in extensive signal attenuation. These situations offer probes for measuring very short root-mean square spin displacements, the ADCs of molecules with short T_1 s and T_2 s, and very small diffusion coefficients such as that for a fluid diffusing into a solid^{5,6,7} since the gradient resulting from the stray field of a superconducting magnet is clearly constant.

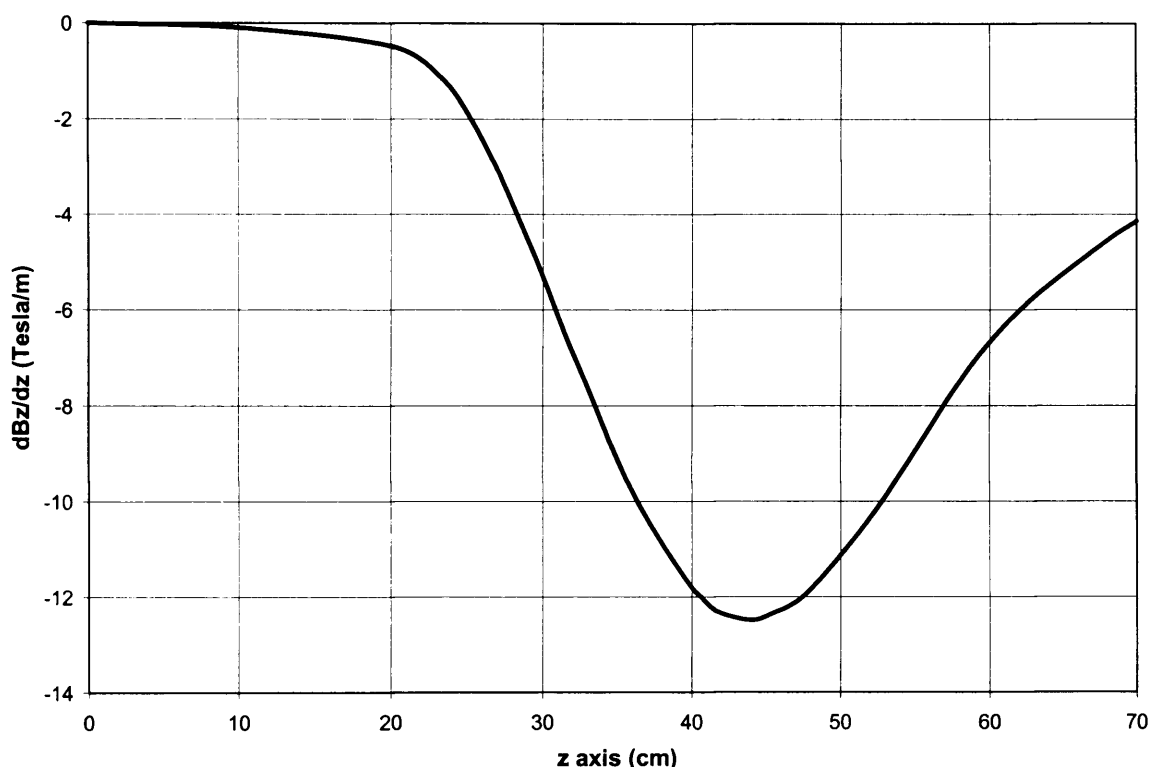


Figure 5.2. 2: Graph showing how the strength of magnetic field gradient varies with distance from the homogeneous region in the centre of the magnet along the z axis. The slope of the curve indicates the uniformity of the magnetic field gradient, where the uniform regions are at 0 and 43.8 cm from the centre of the magnet, at which points the slopes are zero. However, the gradient strength at the centre of the magnet is of course zero. The magnet to which this plot refers, is, an Oxford 40cm wide bore 4.7 Tesla superconducting magnet.

Studies using the fringe field to measure the ADC, have been performed by Kimmich *et al*², where a stimulated echo technique was adopted. In this study, the values for the T_1 and T_2 of the fluid have been substituted into the attenuation equation in order to calculate a value for the ADC of their experimental fluid. In section 2.2.3, other methods for measuring diffusion coefficients in a constant magnetic field, where relaxation effects are considered, are discussed. A similar, though more precarious, pulse sequence has also been used in this context for measuring

the ADC of polymers⁴. This pulse sequence, like CTPG pulse sequences, compensates for relaxation effects to obtain an accurate ADC measurement in a constant gradient.

5.2.1 Relaxation considerations

When making diffusion measurements in such strong magnetic gradients there are other considerations in addition to the relative magnitude of the diffusion coefficient. The first is that signal will dephase extremely quickly for most liquids (i.e. well within a few milliseconds for a spin echo based experiment). Since this is the case, the period where magnetisation is in the xy plane will be far shorter than the value for the T_2 . Therefore, the amount of signal attenuation resulting from relaxation processes may be insignificant compared to signal attenuation resulting from diffusion and it may be possible to totally disregard this in the calculation of the diffusion coefficient from the measured data. This idea can be pursued quite simply by determining the percentage error in ADC when it is calculated from simulated data using the attenuation equations with and without the relaxation terms, for a variety of gradient strengths, T_1 s, T_2 s, and experimental times. Three computer programmes were written to perform these calculations, one for a spin echo experiment, and two for a stimulated echo experiment. The spin echo simulation is the simplest programme and also the basis on which the stimulated echo simulation programmes were written.

For all of the simulations, a diffusion coefficient and maximum signal attenuation are chosen as input. From these the programme calculates the half echo times for a range of gradients strengths which will give the desired attenuation. For each of these gradients strengths, the programme then calculates the signal attenuation for half echo times ranging from zero to that which yields the maximum attenuation previously specified, according to eqn.[2.2.5]. The process is then repeated ignoring the relaxation term. From these two data sets, the ADC is calculated using regression analysis and compared to the ADC specified, to give a percentage error. The source code for this programme can be found in the appendices. Typical plots showing these errors for a range of T_2 values over a range of magnetic gradient strengths, are shown in fig.[5.2.3].

In addition to the ADC and maximum attenuation values, the stimulated echo versions require the maximum diffusion time to be chosen since unlike the spin echo, the diffusion time does not approximate to the half echo time (see fig.[2.1.4b]). Having calculated the value of τ and the periods magnetisation spends in the xy plane for all gradient strengths from the given input, the signal intensities are then calculated for a range of diffusion times. In one version of the stimulated echo simulation, τ , the period magnetisation spends in the xy plane, is varied and T , the period the net magnetisation vector is along the z axis after the second 90° RF pulse, is kept as a constant. In the second case the situation is reversed and T is varied whilst τ is kept constant. The first case is very similar to the spin echo simulation, where a range of T_2 values are also used. The T_1 relaxation

term is constant throughout in this case, and is ignored. A typical output from this simulation is given in fig.[5.2.4]. The second stimulated echo simulation keeps the value of τ constant and varies T_1 , so that the T_1 relaxation term will vary for subsequent diffusion times, while the T_2 term remain constant. For this reason, a range of T_1 values is employed in this simulation. Fig.[5.2.5] show data generated from this simulation. The source code for these programmes can be found in appendices [8] and [9] respectively.

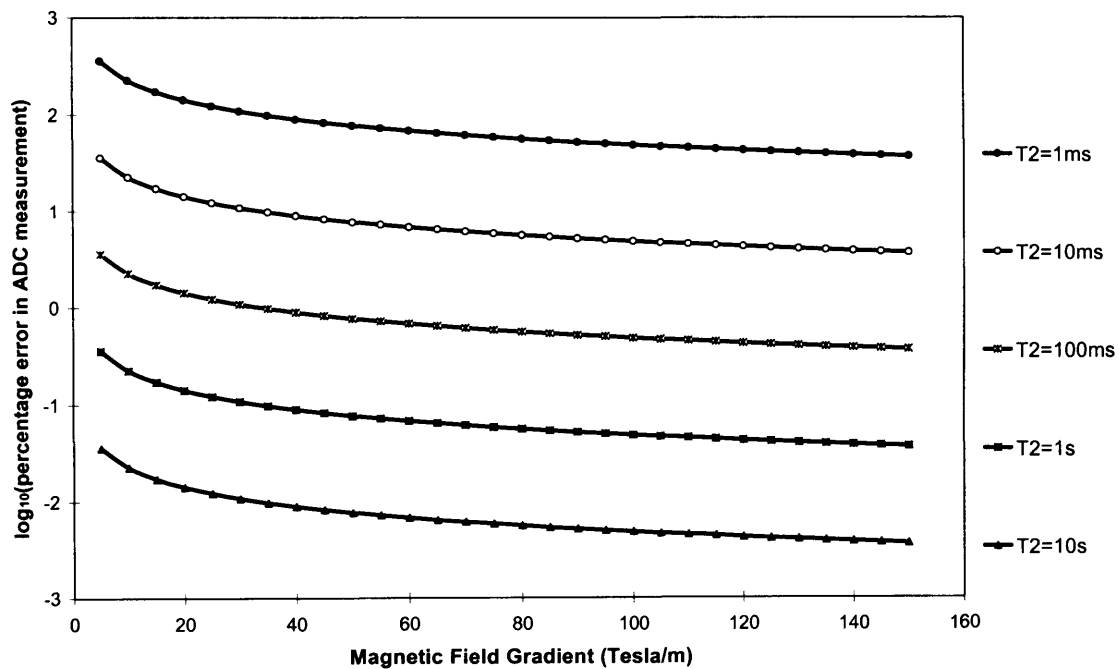


Figure 5.2. 3: A plot showing the error in ADC calculation when the relaxation term is ignored for a spin echo experiment in a constant magnetic field gradient for various T_2 values over a range of magnetic field gradient strengths. For these simulations, maximum attenuation was set to 50% and the diffusion coefficient used was that of pure water at room temperature.

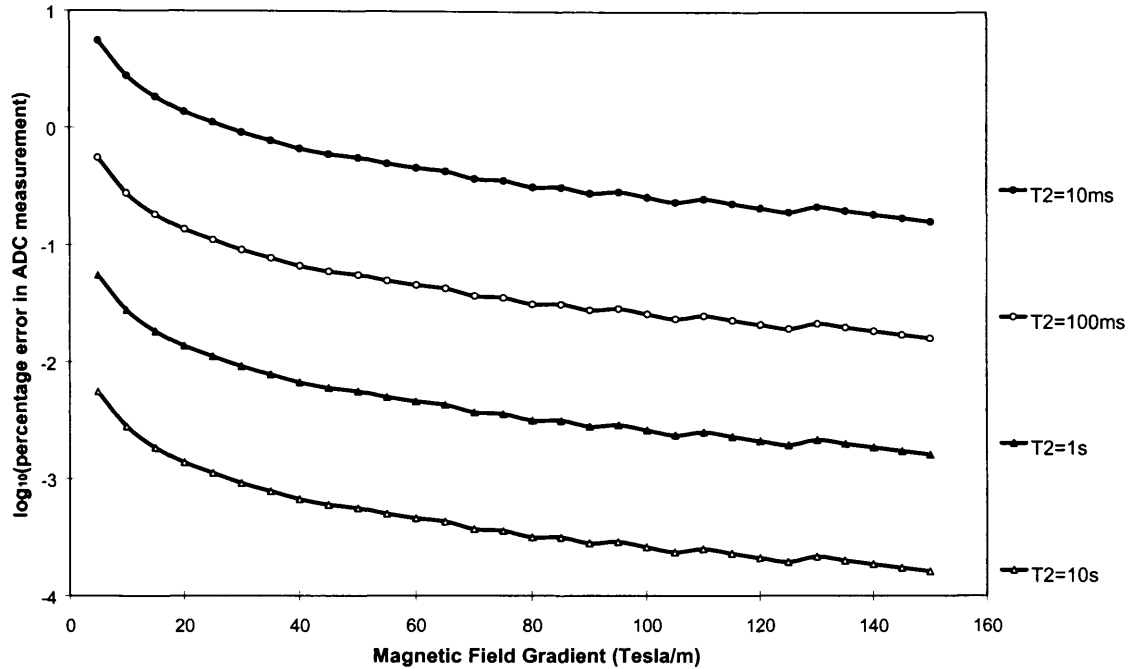


Figure 5.2. 4: A plot showing the error in ADC calculation when the relaxation terms are ignored for a stimulated echo experiment in a constant magnetic field gradient for various T_2 values over a range of magnetic field gradient strengths. The simulations were performed by fixing the period T and varying τ . For these calculations, the maximum attenuation was set to 50%, the maximum diffusion time was 10ms and the diffusion coefficient was that of pure water at room temperature.

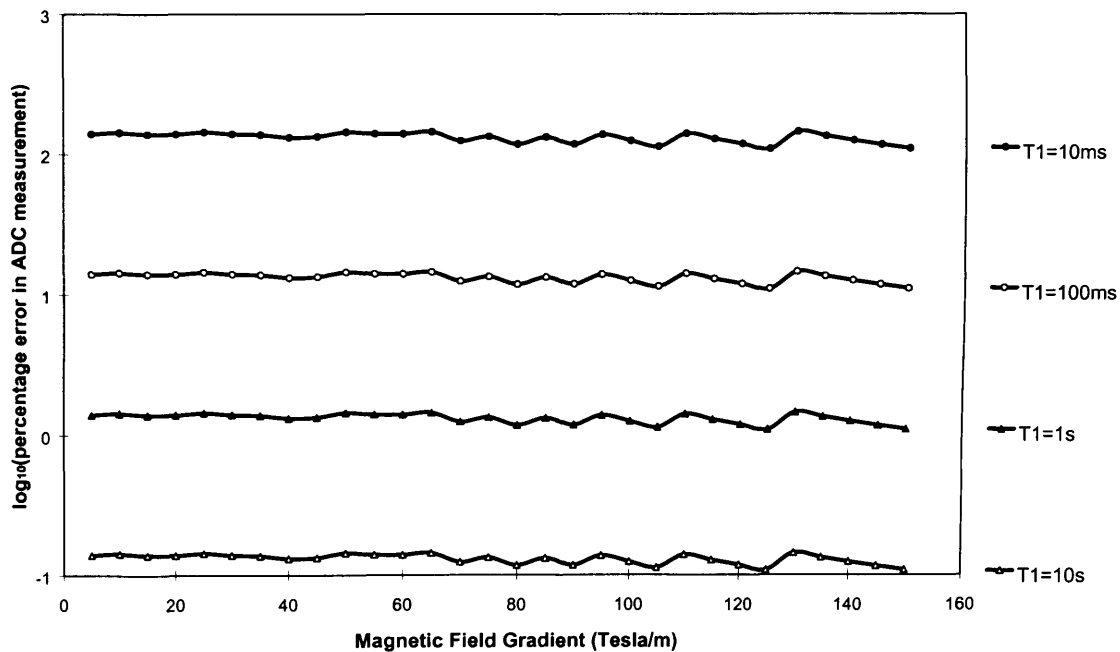


Figure 5.2. 5: A plot showing the error in ADC calculation when the relaxation terms are ignored for a stimulated echo experiment in a constant magnetic field gradient for various T_1 values over a range of magnetic field gradient strengths. The simulations were performed by fixing the periods τ and varying the length of T . For these calculations, the maximum attenuation was set to 50%, the maximum diffusion time was 10ms and the diffusion coefficient was that of pure water at room temperature.

5.2.2 Pulse width considerations

The second major consideration is the selective excitation/slice selection resulting from RF excitation in the presence of a magnetic field gradient, as discussed in section 2.2.2. Since the magnetic field gradients are relatively strong, the thickness of the slice, δz , will be extremely thin according to:

$$\delta z \propto \frac{1}{\gamma G_z t_w}, \quad [2.2.1]$$

where t_w is the length of the RF pulse applied and G_z is the magnetic gradient strength along the z axis, leading to very low signal intensities with which to work. In addition to this, diffusing spins may move out of the region initially excited, resulting in further signal loss when further RF pulses are applied¹. This will of course have implications on the accuracy of the ADC calculated, and therefore, the number and width of the RF pulses, together with the gradient strength, should remain constant for a series of acquisitions. This effect is further highlighted in the situation where the initial 90° excitation pulse is followed by longer refocusing pulses, such as the 180° pulses in spin echo and CTPG since the volume refocused will be approximately half that of the volume excited resulting in the immediate loss of half of the signal.

This may be avoided by halving the power and doubling the length of the excitation pulse, or by using twice the power and the same length for the refocusing pulse. However, this will increase the overall experiment time, which at very high magnetic gradient strength, or for very fast SDC, may limit the range of diffusion times available. One approach to balance out these options is to use two identical RF pulses for both excitation and refocusing. By using raising and lowering operators¹², it is possible to calculate the optimum angle for maximum signal at acquisition for any combination of RF pulses. Graphs showing these calculations for the CTPG and spin echo pulse sequences are given in fig.[5.1.6] for both the regular and proposed pulse widths.

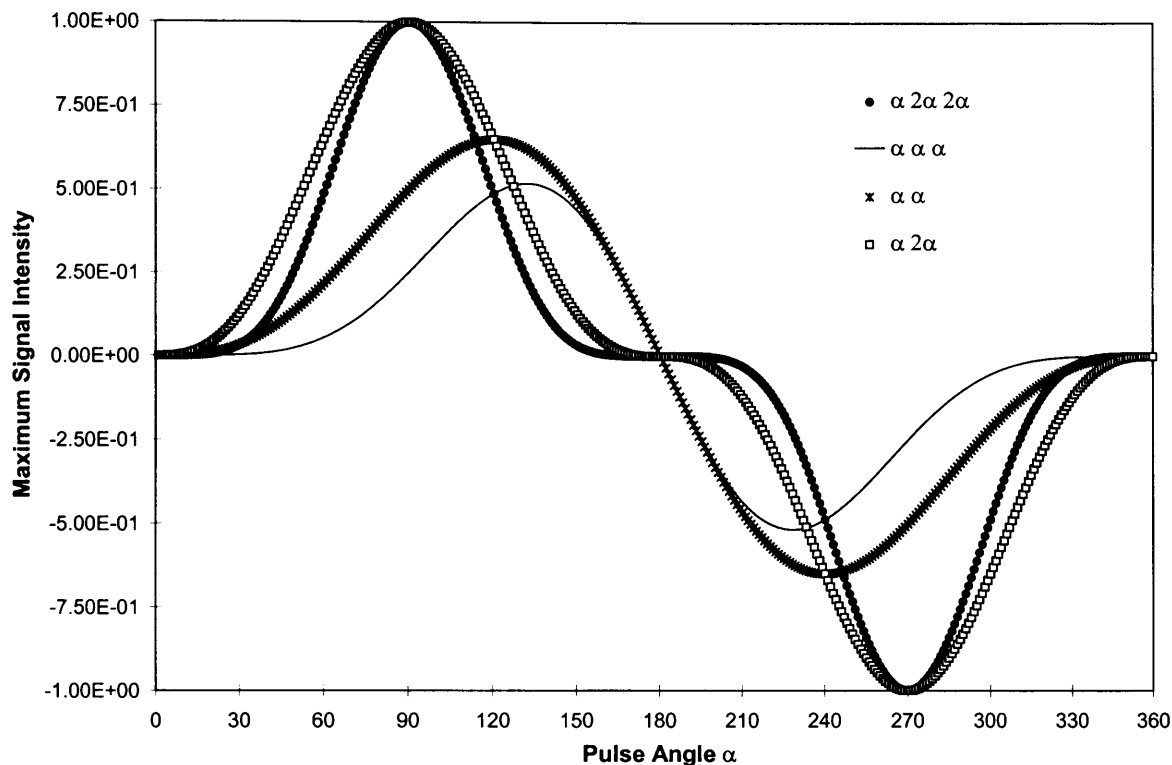


Figure 5.2. 6: Plot showing the optimal pulse angles for CTPG (3 pulse) and spin echo (2 pulse) experiments for common and mixed RF pulse schemes.

5.3 Discussion of Simulations.

There appear to be two conflicting factors that should be addressed when making diffusion measurements in very strong, constant magnetic field gradients: whether relaxation terms can be ignored, since their exclusion may distort the value of the diffusion coefficient obtained; and what is the simplest pulse sequence available to provide the greatest signal as the number of RF pulses in the pulse sequence will determine the sensitivity of the experiment. The simulations shown in figs.[5.2.3],[5.2.4] and [5.2.5], show clearly what percentage error can be expected in the ADC measurement provided that the magnetic gradient strength and the approximate relaxation parameters for the sample are known. For T_2 sensitive pulse sequences such as spin echo, the error is less than 5% for all gradients over $5Tm^{-1}$ and T_2 values greater than 100ms. Therefore diffusion measurements of liquids in fringe fields need not use more complicated, relaxation-compensating pulse sequences, though for slower diffusion coefficients such as gels where T_2 is a great deal shorter, the error will increase fairly rapidly. Under normal circumstances, stimulated echo pulse sequences would be adopted for these ADC measurements. However, as with all stimulated echoes

half of the signal is lost¹³, so it is preferable to use a spin echo wherever possible. The problem that arises from this is the requirement to minimise the experimental time so that signal loss from relaxation effects is minimised (attenuation from diffusion should be adequate for ADC measurement since the magnetic field gradients are so strong). To minimise this time, the RF pulses need to be as short as possible, yet for a spin echo, the refocusing pulse is long. In addition to this, the pulse will only refocus half of the original slice volume. As mentioned earlier in this chapter, this may be avoided by using a half power pulse for twice the duration for excitation, though the benefits of this are debatable since a narrower slice is being excited to start with resulting in a reduction in signal intensity. A compromise is to use two α pulses. Though the maximum signal intensity this pulse sequence generates is less than a full spin echo, there is no variation in the thickness of the slices excited by the first pulse and refocused by the second. From fig.[5.2.6], it can be seen that the maximum signal arises for this pulse sequence when α is equal to 120° .

When T_2 is short compared to T_1 in conventional NMR diffusion studies, a stimulated echo pulse sequence is usually adopted. The same will be applicable in the fringe field for species with very short T_2 s. Usually the period T , where the net magnetisation vector lies along the z axis, would be the interval which is incremented in sequential experiments using a constant gradient technique, though the condition where this period is maintained and τ is varied, has also been modelled.

This latter model mirrors the trends seen with the spin echo simulation, which is expected, though the overall error is less since the τ period of a stimulated echo is less than that in a corresponding spin echo because the diffusion time includes the T period. In fig.[5.2.5], where T , the period the net magnetisation vector is along the z axis after the second RF pulse, is varied for the stimulated echo pulse sequence, it appears that the error remains fairly constant for all gradient strengths. This is probably due to the programme first calculating the maximum value of τ in the same manner as the other stimulated echo simulation and then proceeding to calculate the value of T to fulfil the input criteria. The result of this is that only a small range of T values was used in the simulation. The range of T values does increase as the gradient strength increases as one would expect, though the related relaxation terms are much smaller at these high gradient strengths, as shown by the spin echo simulation, resulting in very little change in error throughout the range of gradient strengths. The errors for both methodologies are lower in comparison to those used in the straight spin echo because the τ period is kept fairly short minimising the relaxation term, $\exp(-2\tau/T_2)$. The disadvantages of using this pulse sequence in experiments are: when a stimulated echo is generated half of the signal is immediately lost; and the sequence will also give rise to other spurious echoes which may affect the accuracy of the measurement unless removed by means of phase cycling¹⁴.

From this initial discussion, it would appear that for many liquid samples there is no requirement for relaxation-compensated pulse sequences for measuring diffusion in fringe fields. However, the graphs indicate that for short T_1 s and T_2 s, and for gradients up to about 10 Tm^{-1} , the errors can be significant ($>5\%$). Therefore, it only seems appropriate to adopt a relaxation-compensated pulse sequence when studying samples with very short relaxation constants in the fringe field, or any sample if the gradient strength is relatively weak. The theoretical error in calculating ADCs is zero since these pulse sequences compensate for relaxation, and therefore for all the loss of signal from longer experiment times and extra pulses, such pulse sequences are useful. It is therefore essential to minimise the experimental time to maximise the amount of signal on acquisition. Hardware limitations, as previously described, require the pulse sequence to contain as few pulses as possible since the pulse lengths will often be greater than the duration of the delays between them. The simplest relaxation-compensating pulse sequence is the CTPG pulse sequence shown in fig.[2.2.7], consisting of three pulses. In its standard format, the sequence consists of an initial 90° excitation pulse followed by two 180° refocusing pulses. To minimise the experimental time, and excite as thick a slice as possible the CTPG pulse sequence may also use common α pulses. The optimal angle for this is shown in fig.[5.2.6]. One point to be aware of when using this sequence in this format, is that when τ_1 and τ_2 approach equality (remember that whilst the parameters do vary, their sum must remain constant), the echo that passes through the desired coherence transfer pathway will coincide with the stimulated echo generated from magnetisation which spends the central $(\tau_1 + \tau_2)$ period along the z axis.

5.4 Experimental Methods.

All of the experiments were performed using a Spectroscopy Imaging System Corporation NMR imaging spectrometer equipped with a 4.7T, 30 cm horizontal bore magnet operating at a proton resonance frequency of 200 MHz. This machine has been used for work involving NMR imaging in the fringe field (commonly known as *STRAY Field Imaging* or STRAFI^{15,16}), and therefore initial experiments were performed using this “local knowledge” in the same region as these previous fringe field studies. This set-up uses a ^{31}P surface probe resonating at 81 MHz⁶, which may be used for

proton at the appropriate fringe field strength of 1.9 Tesla, 54.7cm from the centre of the magnet, where the magnetic field gradient strength is, according to the manufacturer's data, 9.1 Tm^{-1} . *

A series of experiments using this hardware arrangement was performed for both spin echo and CTPG pulse sequences. Variations in pulse length and power were also used in experiments to see if the refocusing of narrower slices than were originally excited, affected the accuracy of the ADC measurement in any way. In addition to these experiments, where α and 2α RF pulses are used, the common pulse versions, discussed in the last section, were also performed.

The sample liquid was placed in a cylindrical vial of 15 mm diameter which was held inside the loop of the surface transmit/receive coil positioned at the designated point along z, from the centre of the magnet. For the spin echo experiments, the 90° pulse was calibrated, and found to be $16 \mu\text{s}$, at near maximum power for the RF amplifier, though for the variation of the experiment, where, the pulse width of the 90° pulse is the same as that for the 180° pulse, the power was halved and the pulse width doubled. The half echo time, τ , approximates to the diffusion time, which was varied between 10 and $40 \mu\text{s}$. At each diffusion time, 4 transients were acquired with a *Cyclops*¹⁷ phase cycle for the spin-echo/ α - α / α - 2α pulse sequences.

The amplitude of the NMR echo was measured as the peak echo amplitude in the time domain. Figure [5.4.1] shows the typical echo amplitudes as a function of diffusion time for water. However, to obtain the ADC, the resulting gradient from a plot of $\ln(\text{Intensity})$ against $\delta^2(\Delta-\delta/3)$ must be made; the linearity of this line will also give a rough indication into the linearity of the field gradient (fig.[5.4.2]).

Further experiments, using the CTPG pulse sequence, and α - α / α - 2α - 2α pulse sequences using the CTPG delay timings, were performed using the same physical set-up as the two pulse experiments. Again, the 90° pulse was calibrated, and found to be $16 \mu\text{s}$. The half echo times, τ_1 and τ_2 , approximate to the two diffusion times of the pulse sequence. The sum of the two periods was kept to a constant $500 \mu\text{s}$, where the two diffusion times started at $10 \mu\text{s}$ and $490 \mu\text{s}$, and gradually equalised with each successive measurement. At each diffusion time, 8 transients were acquired with an eight step phase cycle. Combinations of pulse length and power to generate specific angled pulses to vary the width of the excited region, were also used here to investigate signal loss by spins

* It should be noted that this position is not the region where the field gradient is linear; this was found to be at a position 43.8 cm from the centre of the magnet at a field of 3.1 Tesla, where the proton resonance frequency is 132MHz. Unfortunately, the RF amplifiers for the spectrometer did not produce this frequency. The only way 133 MHz could be produced was by using the broadband decoupling amplifier. This amplifier was not very powerful, resulting in pulses longer than the diffusion times required to observe signal and for this reason, studies at this position could not be pursued.

diffusing out of the excited region of the sample, though for all experiments, the second and third pulses were identical.

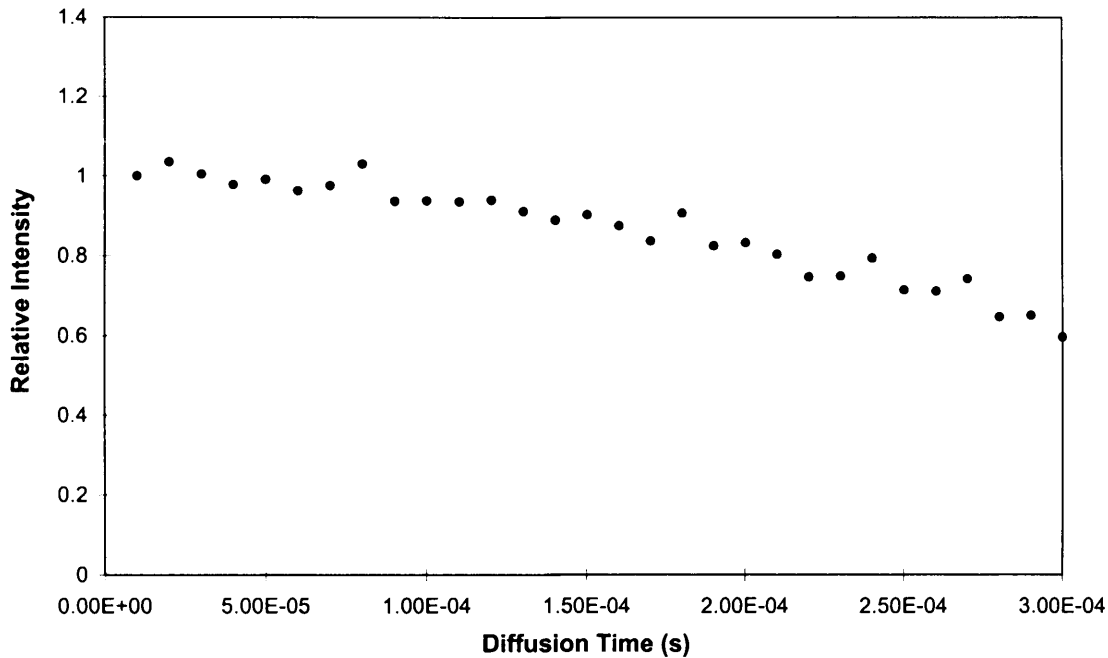


Figure 5.4. 1: Signal intensity as a function of square root of diffusion time (δ) for water using an $\alpha\alpha$ pulse sequence.

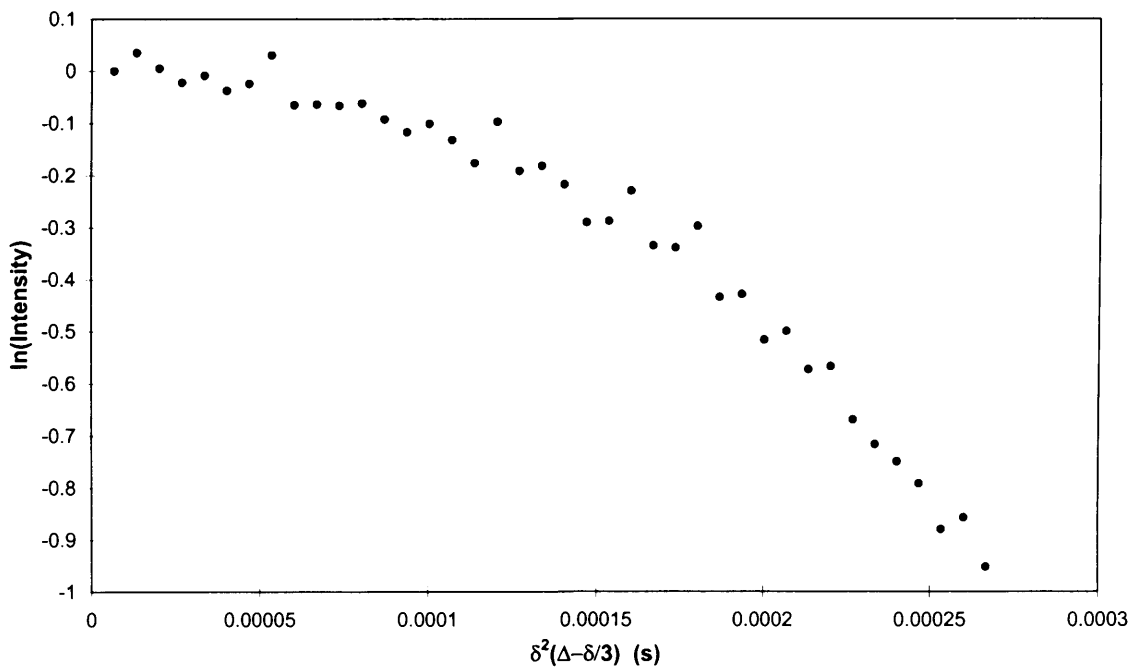


Figure 5.4. 2: Semi-log plot taken from the Stejskal-Tanner equation, where the ADC may be extracted from the gradient of the linear regression fit. Taken from the data shown in fig.[5.4.1].

5.5 Results.

From visual inspection of fig.[5.4.2], it appears obvious that the gradient of the slope is not constant. This is an effect of the two features discussed earlier in this chapter; the linearity of the magnetic field gradient and the contributions to diffusion times from the lengths of the RF pulses. It has been stated that the magnetic field gradient is not linear in this region*, and this problem would have been avoided under more favourable conditions. Despite this, the rate of change in magnetic gradient strength across this sample is only about $0.4 \text{ Tm}^{-1}\text{cm}^{-1}$, which approximates to a $\pm 3 \%$ change in gradient strength across the sample which cannot account for the deviation in the linearity of the plot in fig.[5.4.2].

Since the pulse lengths are of the same order of magnitude as the shorter diffusion times, at these times any inaccuracy in the assumed diffusion time will be greatest, as can be seen in the graph, where it is the earlier points that deviate from linearity. To account for this type of contribution to the overall diffusion time, half of the excitation pulse length and one quarter of the duration of the refocusing pulse are included in δ for the spin echo. For Δ , three-quarters the duration of the refocusing pulse was included. The same data as shown in fig.[5.4.2], is shown in fig.[5.5.1], though with these amendments to the diffusion periods. Clearly, this plot is far closer to being linear, though ADC calculations have been made from both data types.

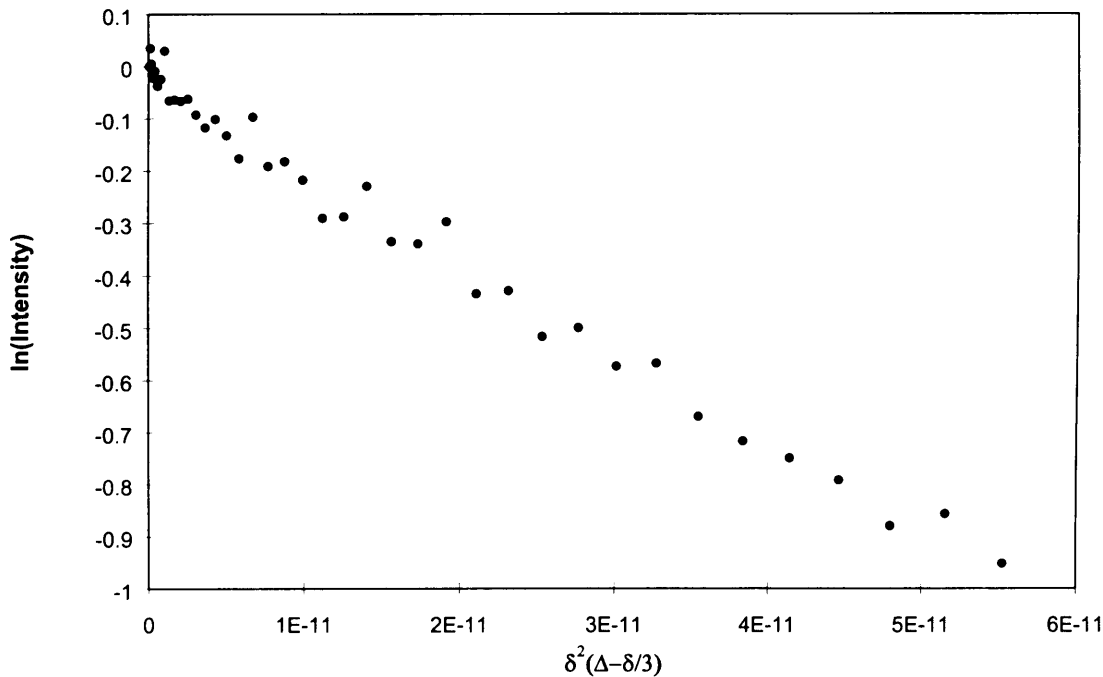


Figure 5.5. 1: Semi-log plot to calculate ADC for data which includes pulse length contributions to δ and Δ . Taken from the data shown in fig.[5.4.1].

A similar routine is also introduced into the CTPG data, though this will have less of an obvious effect since the overall time is constant. In addition, both delays will have similar pulse contributions, which, when added into the $(\tau_1^3 + \tau_2^3)$ term, will affect the slope uniformly for all combinations of τ_1 and τ_2 .

| Solvent | Expt | Pulse 1 (μ s) | Pulse 2 (μ s) P2 \equiv P3 | ADC (m^2s^{-1}) (With pulse contributions) | ADC (m^2s^{-1}) (No pulse contribution) | Apparent G (Tm^{-1}) (With pulse contributions) | Apparent G (Tm^{-1}) (No pulse contribution) |
|---------|--------------------------|-----------------------|---|--|---|---|--|
| Water | $\alpha \alpha$ | 32 | 16 | $(2.97 \pm 0.08)e^{-9}$ | $(3.49 \pm 0.10)e^{-9}$ | 10.35 ± 0.05 | 11.20 ± 0.06 |
| Water | $\alpha \alpha$ | 32 | 32 | $(2.94 \pm 0.06)e^{-9}$ | $(3.80 \pm 0.09)e^{-9}$ | 10.28 ± 0.05 | 11.70 ± 0.06 |
| Water | $\alpha 2\alpha$ | 32 | 32 | $(2.94 \pm 0.06)e^{-9}$ | $(3.80 \pm 0.06)e^{-9}$ | 10.28 ± 0.05 | 11.70 ± 0.05 |
| Water | $\alpha 2\alpha$ | 16 | 32 | $(3.23 \pm 0.13)e^{-9}$ | $(4.20 \pm 0.19)e^{-9}$ | 10.79 ± 0.05 | 12.30 ± 0.08 |
| Water | $\alpha \alpha \alpha$ | 32 | 16 | $(1.19 \pm 0.09)e^{-9}$ | $(1.19 \pm 0.08)e^{-9}$ | 6.46 ± 0.06 | 6.53 ± 0.05 |
| Water | $\alpha \alpha \alpha$ | 32 | 32 | $(8.51 \pm 0.13)e^{-10}$ | $(9.05 \pm 0.15)e^{-10}$ | 5.54 ± 0.07 | 5.71 ± 0.07 |
| Water | $\alpha 2\alpha 2\alpha$ | 32 | 32 | $(1.17 \pm 0.08)e^{-9}$ | $(1.31 \pm 0.09)e^{-9}$ | 6.48 ± 0.05 | 6.88 ± 0.05 |
| Water | $\alpha 2\alpha 2\alpha$ | 16 | 32 | $(9.72 \pm 0.12)e^{-10}$ | $(1.07 \pm 0.13)e^{-9}$ | 5.92 ± 0.07 | 6.20 ± 0.06 |
| Acetone | $\alpha \alpha$ | 32 | 16 | $(4.00 \pm 0.22)e^{-9}$ | $(4.81 \pm 0.28)e^{-9}$ | 8.57 ± 0.09 | 9.38 ± 0.10 |
| Acetone | $\alpha \alpha$ | 32 | 32 | $(3.83 \pm 0.20)e^{-9}$ | $(4.35 \pm 0.23)e^{-9}$ | 8.37 ± 0.08 | 8.93 ± 0.09 |

Table 5.5. 1: Summary of the results from fringe field experiments. The errors quoted are calculated from the variance of the linear regression. The true magnetic field gradient strength for these experiments was 9.1 Tm^{-1} and the SDCs for water and acetone are $2.30e^{-9}$ and $4.54e^{-9} \text{ m}^2\text{s}^{-1}$ respectively.

5.6 Discussion.

The first and most obvious trend to notice, is the greater accuracy in the spin echo data compared to the CTPG. The reason for this does not appear obvious; the variation in ADC between the analysis with and without pulse length contribution is not exceptionally large. One reason may be that the phase cycle, though fairly robust still does not exclude all of the extra echoes generated by the sequence. The SDC values calculated from the spin echo data, are high for water, yet low for acetone. Since the experimental protocol was identical in both circumstances, this is difficult to account for, both in terms of gradient strength accuracy and experimental timings, though the values are of the right order of magnitude.

The experiments cover three situations with regard to slice thickness. The first scenario (case 1) is where a “thick” slice is excited and the subsequent refocusing is experienced in a “narrower” slice, the second (case 2) is where slice thickness is constant throughout and the final case (case 3) is where a narrow slice is excited and the refocusing occurs in a thicker slice. In general it appears that the best results (i.e. those nearest to the literature SDC) appear from experiments using case 3, where a narrow slice is excited but a wider slice is perturbed by all of the refocusing RF; the poorest results occur when the opposite is true (case 1). This is mainly due to spin diffusion. Even though the diffusion times are exceedingly short and only very short distances are covered, spins will diffuse out of the excited region. Therefore, when the refocusing pulse is applied, not all of the spins initially excited will be perturbed and part of the magnetisation will be lost. This loss in magnetisation is exacerbated when narrower slices are perturbed by the refocusing pulse since spins at the outer limits of the slice will only be partially refocused. Therefore, their loss due to diffusion will only affect the signal attenuation slightly. Indeed, the least accurate measurement of water ADC by a two pulse method is, ironically, the traditional α - 2α spin echo where it is the RF pulse length which is larger in the second pulse, not the power. An additional point, is that it does not appear at all detrimental to use a common pulse α - α , or α - α - α type sequence in preference to mixed α - 2α , or α - 2α - 2α pulse sequence; the results show very little difference between the two. The common pulse sequences are however inherently simpler to optimise and use, in addition to exciting a common slice thickness throughout the experiment. This will allow the RF amplifier to be used near maximum output, maximising the thickness of the slice perturbed and therefore the signal intensity, compensating for the lower maximum signal obtainable, as shown in fig.[5.2.6].

5.7 Summary and Conclusion.

This chapter has shown by simulation and experimentation how diffusion measurements in the fringe field are affected by relaxation effects, slice-selective excitation and pulse length. The results have shown that for very strong gradients, relaxation effects can be ignored for liquids, though as the relaxation times (especially T_2) diminish beneath the hundreds of milliseconds, they may require compensation. However, a stimulated echo pulse sequence where the longitudinal period is varied may possibly still yield accurate enough results. It has also been shown that accurate mixed pulse lengths are not required to obtain accurate measurements for either a two or three pulse diffusion experiment. In fact, it may prove beneficial to use common pulses since these excite a common slice throughout the experiment so signal is not lost as when the longer refocusing pulses of a conventional spin echo refocus a narrower slice than that initially excited.

Though the experiments have agreed with much that has been predicted both by speculation and simulation, the accuracy of the measurements is still not convincing. Performing the same experiments on a system where the fringe field gradient is more uniform with a suitable RF amplifier should resolve some of these issues.

-
- ¹ P. T. Callghan. *"Principles of Nuclear Magnetic Resonance Microscopy."*,
Oxford University Press, (1991).
- ² R. Kimmich, W. Unrath, G. Schnur and E. Rommel. *J. Magn. Reson.*, **91**, 136 (1991).
- ³ D. E. Demco, A. Johansson and J. Tegenfeldt. *J. Magn. Reson. A*, **110**, 183 (1994).
- ⁴ A. A. Samoilenko, D. Yu. Artemov and L. A. Sibeldina. *JETP Lett.*, **47**, 417 (1988).
- ⁵ K. L. Perry, P. J. McDonald, E. W. Randall and K. Zick. *Polymer.*, **35**, 2744 (1994).
- ⁶ P. Kinchesh, E. W. Randall and K. Zick. *J. Magn. Reson.*, **100**, 411 (1992).
- ⁷ E. W. Randall, A. A. Samoilenko and T. Nunes. *J. Magn. Reson. A*, **116**, 259 (1995).
- ⁸ T. B. Benson and P. J. McDonald. *J. Magn. Reson. B*, **109**, 314 (1995).
- ⁹ T. J. Norwood and R. A. Quilter. *J. Magn. Reson.*, **97**, 99 (1992).
- ¹⁰ E. L. Hahn. *Phys. Rev.*, **80**, 580 (1950)
- ¹¹ E. O. Stejskal and J. E. Tanner. *J. Phys. Chem.*, **42**, 288 (1965).
- ¹² R. R. Ernst, G. Bodenhausen, and A. Wokaun. *"Principles of Nuclear Magnetic Resonance in One and Two Dimensions."*,
Oxford University Press (1987).
- ¹³ D. E. Woessner. *J. Chem. Phys.*, **34**, 2057 (1961).
- ¹⁴ D. I. Hoult and R. E. Richards. *Proc. Roy. Soc. (Lond)*, **A344**, 311 (1975).
- ¹⁵ A. A. Samoilenko, D. Yu. Artemov and L. A. Sibeldina. *Bruker Rep.*, **2**, 30 (1987).
- ¹⁶ A. A. Samoilenko and K. Zick. *Bruker Rep.*, **1**, 40 (1990).
- ¹⁷ G. Bodenhausen, R. Freeman, D. L. Turner. *J. Magn. Reson.*, **27**, 511 (1977)..

Chapter 6

Summary and Conclusion.

In this final chapter, the work presented in this thesis will be summarised and its potential contribution to future developments will be discussed in conclusion. Like most final chapters in theses, it is a personal account and though I aim to be objective, I am inevitably biased to my own particular research interests and findings.

The work presented in this thesis has a common theme related to measuring diffusion by NMR, namely the type of magnetic field gradients used for encoding the information. In all of the studies covered, a constant magnetic field gradient is used, whether it is applied in the stray field or generated internally as a result of magnetic susceptibility differences. In the majority of work covered in the field of NMR and diffusion, the application of pulsed magnetic field gradients is used for this purpose. In certain situations such as the use of NMR logging tools in the field of oil recovery, an applied MFG might not be the easiest or most effective approach. Very little work using static gradients appears in the literature today, though applications such as logging are still being investigated, demonstrating the potential of measurements made using static gradients. The main reason for this is that modern applied gradients offer far more experimental flexibility both in terms of magnetic manipulation and power.

In my first research chapter a new NMR pulse sequence is developed and its applications discussed. The RD-CTPG pulse sequence is shown to be used by means of difference spectra/images to isolate NMR information that is solely resulting from diffusing spins that have been restricted. All signal contribution from freely diffusing, unrestricted spins is cancelled out. This effect is most clearly seen in the context of an imaging experiment. The images measured using the RD-CTPG pulse sequence with a spin echo imaging module show celery (restricted system) and a tube of water (unrestricted system). The only difference between the images is the experimental timings used in the RD-CTPG part of the pulse sequence, as explained in the chapter. However, when one image is subtracted from the other, only the restricted system, the celery, is visible.

Further advantages of the method over existing techniques are shown in some semi-quantitative analysis that demonstrates RD-CTPG's insensitivity to freely diffusing spins that interfere in the accuracy of the measurements made using conventional procedures to extract structural information of heterogeneous systems such as emulsions. At the same time, the pulse sequence is also shown to be comparable to these existing techniques in the analysis of environmental structure; a graph correlating analysis made using RD-CTPG and an existing technique is shown.

Modern techniques using strong applied MFGs to minimise the effects of internal MFGs generated by differences in magnetic susceptibility have been developed. Using an existing technique that adopts constant static gradients for encoding diffusion, I have made measurements on different heterogeneous samples. In addition to this, the data measured has been analysed to obtain information about the internal structure of the samples used. In order to understand these results, computer models were developed and used to simulate different scenarios. The analysis of the simulated data is shown in chapter 4 and is used to explain the results of the analysis of the data from lung and other porous media. The results show that parameters relating to the packing structures of the system can be extracted using this technique and that internal gradients can themselves be used as a means of encoding diffusion. The analysis itself does make some assumptions and may not be the perfect answer; this is an area that may be taken further at later date.

The final research chapter of this thesis addresses the possibilities of using the linear part of the MFGs generated in the stray field of a superconducting magnet as the diffusion encoding gradient. Stray field NMR itself has developed as a method of making NMR measurements in solids where relaxation parameters are notably very short, and more recently in measuring the diffusion of fluids into solids, where the diffusion coefficient is very slow. This is achieved by using the very strong MFG found in the stray field. This will encode a large phase change for a fluid despite such a small amount of movement. In the previous chapters, the pulse sequences that were adopted and used a constant MFG to encode diffusion, required the insertion of time periods to maintain a constant amount of signal attenuation from relaxation effects. In chapter 5, the potential for using very short, simple echo based pulse sequences to measure diffusion coefficients in fluids was investigated. Since these measurements are made over such an extremely short period, the possibility of ignoring relaxation effects was also considered. This was because the amount of time that the magnetisation was to spend in the xy plane or along the z axis was negligible compared to T_2 or T_1 . The results themselves, were not wholly conclusive, but did indicate that if superior RF amplifiers were available to generate shorter, stronger excitation and refocusing pulses, the technique was viable.

In conclusion, this thesis demonstrates the use constant gradients as a means of encoding diffusion. Some of the techniques used are subtle, both in concept and application. The data that is extracted from these experiments holds a great deal useful information about the restricting environment, and the concepts behind the extraction of this data are also given in this thesis. The potential use of the measurements made using only internal field gradients in the oil industry has already been mentioned, but in terms of machinery, while these measurements can be made using any NMR machine, higher fields will obviously give higher susceptibility contrasts. Further to this, since the gradients are inherent, the gradients move with the sample, assuming that the sample itself lies within the magnet. This allows greater freedom for tools such as logging tools in the sense that there need not be concerned that the sample is moving too much with respect to an applied gradient. This does of course assume that the homogeneity of the magnet covers a greater volume than that of the sample.

The RD-CTPG pulse sequence itself may be adapted to create a pulsed gradient version, since it is the concept of isolating the restricted information and not the use of constant gradients that is important here. This would allow greater adaptability for the concept so that smaller ranges of pore size might be isolated. However, it is its use as a quick, reliable test for restriction ignoring unrestricted spins that is its most likely application.

Appendices

Appendix 1: Balinov Simulation (3.3.2)

Appendix 2: Line-shape Simulation (Cubic Model) (4.4.1)

Appendix 3: Line-shape Simulation (Hexagonal Model) (4.4.2)

Appendix 4: Line-shape Simulation (Random Model) (4.4.3)

Appendix 5: Line-shape Simulation with Random B_0 (Cubic Model) (4.4.4)

Appendix 6: Line-shape Simulation with Random B_0 (Hexagonal Model) (4.4.4)

Appendix 7: Diffusion Experiment Simulation in Cubic Model (4.4.5)

Appendix 8: Relaxation Errors in ADC (Spin Echo) (5.2.1)

Appendix 9: Relaxation Errors in ADC (Stimulated Echo) (5.2.2)

Appendix 1:

Balinov Simulation (3.3.2)

```

c      Comparison to A.Smeeton's summation.f
c      Written by SAC 3.1.95
c      Difference is to use an alternative model
c
c      Original model -> von Meerwall & Ferguson
c      New model -> Balinov, Jonsson, Linse & Soderman
c

```

```

implicit none

```

```

c      *****DECLARATIONS*****

```

```

double precision gamma, g, pi, sdc
double precision delta_one, delta_two, diff_timeone, diff_tmetwo
double precision lobar_space, hibar_space, space_incr
double precision att_prod, signal
double precision consts_a, sum1, sum2, summation1, summation2
double precision bar_space

```

```

integer n, i, i1
integer incr, lobar, hibar

```

```

character*132 filename

```

```

c      *****DEFINITIONS*****

```

```

pi=3.1415926535897932384
gamma=26.7519e+7

```

```

write(6,'(A,$)'), 'Please enter the file name for results: '
read(5,*) filename

```

```

open(unit=2, file=filename, status= 'new')

```

```

write (6,'(A,$)'), 'Enter self-diffusion coefficient(m2/s): '
read (5,*) sdc

```

```

write (6,'(A,$)'), 'Enter gradient pulse magnitude (T/m): '
read (5,*) g

```

```

write (6,'(A,$)'), 'Enter duration of first magnetic pulse (ms): '
read (5,*) delta_one
delta_one=delta_one*1e-3

```

```

write (6,'(A,$)'), 'Enter duration of second magnetic pulse (ms):'
read (5,*) delta_two
delta_two=delta_two*1e-3

```

```

write (6,'(A,$)'), 'Enter first diffusion time (s): '
read (5,*) diff_timeone

```

```
write (6,'(A,$)'), 'Enter second diffusion time (s): '
read (5,*) diff_timetwo
```

```
write (6,'(A,$)'), 'Enter smallest barrier spacing (um): '
read (5,*) lobar_space
lobar_space=lobar_space*1.0e-6
```

```
write (6,'(A,$)'), 'Enter largest barrier spacing (um): '
read (5,*) hibar_space
hibar_space=hibar_space*1.0e-6
```

```
write (6,'(A,$)'), 'Enter barrier spacing increment (um): '
read (5,*) space_incr
space_incr=space_incr*1.0e-6
```

```
write (6,'(A,$)'), 'Enter summation value for infinity (n): '
read (5,*) n
```

c *****MANIPULATIONS*****

```
lobar=idint(lobar_space/1.0e-6)
hibar=idint(hibar_space/1.0e-6)
incr=idint(space_incr/1.0e-6)
consts_a=(8.d0*-1.d0)*(gamma**2)*(g**2)/(sdc*(pi**6))
```

c *****PROGRAM*****

```
do 20 i1=lobar, hibar, incr
```

```
bar_space=dfloat(i1*1.0e-6)
summation1=0.d0
summation2=0.d0
att_prod=0.d0
signal=0.d0
```

```
do 10 i=0, n
```

```
sum1=0.d0
sum2=0.d0
```

```
call echo_one(gamma, g, pi, sdc, delta_one,
&             diff_timeone, i, bar_space, sum1)
```

```
summation1=summation1+sum1
```

```
call echo_two(gamma, g, pi, sdc, delta_two,
&             diff_timetwo, i, bar_space, sum2)
```

```
summation2=summation2+sum2
```

```

10    continue

      call prod_echo(summation1, summation2, att_prod)

      signal=consts_a*att_prod*(bar_space**4)
      signal=dexp(signal)

c      write(2,*)'barrier space (um)='i1,
c      &          '      E(delta,gamma,g)='signal

      write(2,*)i1, signal

20    continue

c      write(6,*)summation1,summation2,att_prod

      end

c      *****SUBROUTINES*****

      subroutine echo_one (gamma, g, pi, sdc, delta_one,
&          diff_timeone, i, bar_space, sum1)

      double precision gamma, g, pi, sdc, bar_space
      double precision delta_one, diff_timeone
      double precision var_a, var_b, var_c, var_d, var_e
      double precision var_f, complex, sum1

      integer i

      var_a=1.d0/(2.d0*i+1.d0)**6
      var_b=((2.d0*i+1.d0)**2)*(pi**2)*sdc/bar_space**2
      var_c=dexp(-1.d0*var_b*(diff_timeone-delta_one))
      var_d=dexp(-1.d0*var_b*(delta_one))
      var_e=dexp(-1.d0*var_b*(diff_timeone))
      var_f=dexp(-1.d0*var_b*(diff_timeone+delta_one))

      complex=2.d0*delta_one-((2.d0+var_c-2.d0*
&          var_d-2.d0*var_e+var_f)/var_b)
      sum1=var_a*complex

      end

      subroutine echo_two (gamma, g, pi, sdc, delta_two,
&          diff_timetwo, i, bar_space, sum2)

      double precision gamma, g, pi, sdc, bar_space
      double precision delta_two, diff_timetwo
      double precision var_a, var_b, var_c, var_d, var_e
      double precision var_f, complex, sum2

```

```

integer i

var_a=1.d0/(2.d0*i+1.d0)**6
var_b=((2.d0*i+1.d0)**2)*(pi**2)*sdc/bar_space**2
var_c=dexp(-1.d0*var_b*(diff_timetwo-delta_two))
var_d=dexp(-1.d0*var_b*(delta_two))
var_e=dexp(-1.d0*var_b*(diff_timetwo))
var_f=dexp(-1.d0*var_b*(diff_timetwo+delta_two))

complex=2.d0*delta_two-((2.d0+var_c-2.d0*
&          var_d-2.d0*var_e+var_f)/var_b)
sum2=var_a*complex

end

subroutine prod_echo (summation1, summation2, att_prod)
c   Because the terms are 'log', then  $AB=C \rightarrow \ln A + \ln B = \ln C$ 

double precision summation1, summation2, att_prod

att_prod=summation1+summation2

end

```

Appendix 2:

Line-shape Simulation (Cubic Model) (4.4.1)

C A f77 program to model susceptibility
 C induced Bfield in a regular environment
 C This adaptation is to predict the line shape produced
 C Sweeps through 3d to vary orientation in bfield
 C

```
implicit none
real cell_size, pi
real radius, radmax
real x_point, y_point, z_point
real h0, t2, gamma
real sus_sphere, sus_fluid
real bx, by, bz
real bxold, byold, bzold
real unit_cell(50,50,50)
real rndm1, rndm2, rand
real alpha, beta, sign
parameter (h0 = 1.0)
character*132 filename
integer x_count, y_count, z_count, i1
integer j,orient
integer k, steps
common//unit_cell(50,50,50)
real coord(1500)
common//coord(1500)
```

```
write(6,'(A,$)'), 'Please enter the filename: '
read(5,*) filename
```

```
write(6,'(A,$)'), 'Please enter unit cell size(um): '
read(5,*) cell_size
```

```
radmax = cell_size / 2.e0
```

C radmax in um!

```
10 write(6,'(A,$)'), 'Please enter sphere radius (um): '
read(5,*) radius
```

```
if (radius.gt.radmax) then
  write(6,'(A, f10.2, A)'),
&    'radius greater than', radmax, ',not allowed.'
  goto 10
endif
```

```
write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
read(5,*) sus_sphere
```

```
write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
read(5,*) sus_fluid
```

```
write(6,'(A,$)'), 'Please enter T2 value (s): '
read(5,*) t2
```

```
write(6,'(A,$)'), 'Please enter number of small steps: '
read(5,*) steps
```

```
write(6,'(A,$)'), 'Please enter number of orientations: '
read(5,*) orient
```

```

radius = radius*1.0e-6
cell_size = cell_size*1.0e-6
rndm1 = rand(1)
rndm2 = rand(1)
x_point = 0.e0
y_point = 0.e0
z_point = 0.e0
gamma = 26.7519e+7
pi=3.141592653

```

C CREATES A UNIT CELL OF 50*50*50 VALUES, THEN MAKES EQUAL TO
C ZERO OR PREVIOUSLY CALCULATED BFIELD VALUE

```

open(unit=2, file=filename, status= 'new')

do 37 j=0,orient
  do 7 x_count=1,50
    do 8 y_count=1,50
      do 9 z_count=1,50
        unit_cell(x_count, y_count, z_count) = 0
9      continue
8      continue
7      continue

write(6,*), 'unit_cell formed'

```

C CREATES THE X-COORDINATE USING A RANDOM NUMBER GENERATOR
C AND ROTATIONS ABOUT THE X AND Y AXES IN MANY SMALL STEPS

```

bxold = 0
byold = 0
bzold = 1

do 38 k=1,steps

  rndm1 = rand(0)

  if (rndm1.lt.0.5) then

    rndm2 = rand(0)
    if (rndm2.lt.0.5) then
      sign = (-1.e0)
    elseif (rndm2.gt.0.5) then
      sign = (1.e0)
    endif

    alpha = (sign*pi)/36.e0

    bx = bxold

```

```

    by = (byold*cos(alpha)) + (bzold*sin(alpha))
    bz = (bzold*cos(alpha)) - (byold*sin(alpha))

    byold = by
    bzold = bz

    elseif (rndm1.gt.0.5) then

        rndm2 = rand(0)
        if (rndm2.lt.0.5) then
            sign = (-1.e0)

            elseif (rndm2.gt.0.5) then
                sign = (1.e0)

        endif

        beta = (sign*pi)/36.e0

        bx = (bxold*cos(beta)) - (bzold*sin(beta))
        by = byold
        bz = (bzold*cos(beta)) + (bxold*sin(beta))

        bxold = bx
        bzold = bz

    endif

38    continue

    call sphere_position(radius, h0, cell_size,
&                        x_point, y_point, z_point,
&                        sus_sphere, sus_fluid, bx, by, bz)

    call calc_lineshape(radius, x_count, y_count, z_count,
&                        cell_size, gamma, t2, pi)

37    continue

    do 1 i1=1,1500

        write (2,*) (i1-750), coord(i1)

1    continue

    close(2)

    end

    subroutine sphere_position(radius, h0, cell_size,
&                            x_point, y_point, z_point,
&                            sus_sphere, sus_fluid, bx, by, bz)

C PLACES SPHERES IN CUBIC ARRAY POSITION FOR UNIT CELL AND
C SURROUNDING UNIT CELLS

```

C COMMENT OUT IF BFIELD BEING READ IN!!!!

```
real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real radius, h0, cell_size, x_point, y_point, z_point
real sus_sphere, sus_fluid, bx, by, bz
integer x_count, y_count, z_count
real sphere_x, sphere_y, sphere_z
```

```
do 11 sphere_x = -1.e0*cell_size,2.e0*cell_size,cell_size
```

```
  write(6,*)sphere_x
```

```
  do 12 sphere_y=-1.e0*cell_size,2.e0*cell_size,cell_size
```

```
    do 13 sphere_z=-1.e0*cell_size,2.e0*cell_size,cell_size
```

```
      call add_bfield(x_point, y_point, z_point, sphere_x,
&        sphere_y, sphere_z, radius, h0,
&        sus_sphere, sus_fluid, cell_size,
&        x_count, y_count, z_count, bx, by, bz)
```

```
13      continue
```

```
12      continue
```

```
11      continue
```

```
end
```

```
subroutine add_bfield(x_point, y_point, z_point, sphere_x,
&    sphere_y, sphere_z, radius, h0,
&    sus_sphere, sus_fluid, cell_size,
&    x_count, y_count, z_count, bx, by, bz)
```

C ADDS TOGETHER EACH CONTRIBUTION TO ALTER THE BFIELD FOR
C EACH OF THE SURROUNDING SPHERES

```
real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real b, bx, by, bz
real x_point, y_point, z_point, sphere_x, sphere_y, sphere_z
real radius, h0, sus_sphere, sus_fluid, cell_size
integer x_count, y_count, z_count
```

```
do 1 x_count=1,50
  x_point = float((x_count - 1.e0)* cell_size / 49.e0)
```

```
do 2 y_count=1,50
  y_point = float((y_count - 1.e0)* cell_size / 49.e0)
```

```
do 3 z_count=1,50
  z_point = float((z_count - 1.e0)* cell_size / 49.e0)
```

```
  b=0.e0
```

```

      call bfield(x_point, y_point, z_point,
&               sphere_x, sphere_y,
&               sphere_z, radius, h0, sus_sphere
&               , sus_fluid, b, bx, by, bz)

      unit_cell(x_count, y_count, z_count)
&      = unit_cell(x_count, y_count, z_count) + b

3      continue
2      continue
1      continue

end

subroutine bfield(x_point, y_point, z_point, sphere_x,
&               sphere_y, sphere_z, radius, h0,
&               sus_sphere, sus_fluid, b, bx, by, bz)

C  CALCULATES THE CHANGE IN BFIELD DUE TO THE
C  SUSC. DIFFERENCE FOR A PARTICULAR SPHERE

real const_num, const_den, constants
real del_x, del_y, del_z, distance, cos_theta
real x_point, y_point, z_point, sphere_x
real sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b, bx, by, bz

del_x = (x_point - sphere_x)
del_y = (y_point - sphere_y)
del_z = (z_point - sphere_z)

distance = sqrt(del_x**2 + del_y**2 + del_z**2)

if (distance.le.radius) then
  distance = radius
endif

cos_theta = (del_x*bx) + (del_y*by) + (del_z*bz)
&          /(sqrt(bx**2+by**2+bz**2)*distance)

const_num = (sus_sphere - sus_fluid)
&          *(radius*radius*radius)

const_den = (sus_sphere + 2.e0*sus_fluid + 3.e0)
&          *(distance*distance*distance)

constants = const_num / const_den

C  bfield = h0 * (1.e0 - constants*(1 - 3*cos_theta**2))
b = h0 * ( - constants*(1 - 3*cos_theta**2))

return
end

```

```

subroutine calc_lineshape(radius, x_count, y_count,
&                        z_count, cell_size, gamma,
&                        t2, pi)

```

C ADDS TOGETHER THE INDIVIDUAL LINESHAPES FOR ALL ARRRAAY POINTS

```

real radius, cell_size, start
integer x_count, y_count, z_count
real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real pi, t2, v, b
real x_point, y_point, z_point
real gamma, pi
real coord(1500)
common//coord(1500)

do 31 x_count=1,50
  x_point = (x_count - 1.e0)* cell_size / 49.e0

  do 32 y_count=1,50
    y_point = (y_count - 1.e0)* cell_size / 49.e0

    do 33 z_count=1,50
      z_point = (z_count - 1.e0)* cell_size / 49.e0

      start=0.e0

      call add_in_out(x_point, y_point, z_point,
&                    radius, start, cell_size)

      if (start.eq.0.e0) then

        b=unit_cell(x_count,y_count,z_count)

        v=gamma*b/(2*pi)

        call linefreq (v, x_count, y_count, z_count,
&                      t2, pi)

      endif

33      continue
32      continue
31      continue

end

```

```

subroutine add_in_out(x_point, y_point, z_point, radius,
&                    start, cell_size)

```

C CHECKS TO SEE IF MOLECULE IS IN OR OUT OF ANY C OF THE SPHERES FOR A SPECIFIC MOVE

```

real x_point, y_point, z_point, start, radius

```

```

real sphere_x, sphere_y, sphere_z
real in_out_value, cell_size
real in_out_sphere

do 15 sphere_x=0.e0*cell_size,1.e0*cell_size,cell_size

  do 16 sphere_y=0.e0*cell_size,1.e0*cell_size,cell_size

    do 17 sphere_z=0.e0*cell_size,1.e0*cell_size,cell_size

      in_out_value = in_out_sphere(radius, x_point,
&      y_point, z_point, sphere_x,
&      sphere_y, sphere_z)

      start = start + in_out_value

17    continue

16  continue

15  continue

end

real function in_out_sphere(radius, x_point, y_point,
&      z_point, sphere_x, sphere_y, sphere_z)

C CHECKS TO SEE IF MOLECULE IS IN OR OUTSIDE OF A SPECIFIC
C SPHERE FOR A SPECIFIC MOVE

real distance, radius
real x_point, y_point, z_point
real del_x, del_y, del_z
real sphere_x, sphere_y, sphere_z

del_x = (x_point - sphere_x)
del_y = (y_point - sphere_y)
del_z = (z_point - sphere_z)

distance = sqrt(del_x**2 + del_y**2 + del_z**2)
if (distance.le.radius) then
in_out_sphere=1.e0
else
in_out_sphere=0.e0
endif

end

subroutine linefreq (v, x_count, y_count, z_count, t2,
&      pi)

C CALCULATES LORENTZIAN LINE SHAPE FOR FREQUENCY+-5 Hz

real v, v1, pi, t2
real unit_cell(50,50,50)
real func_v, func_denom
common//unit_cell(50,50,50)
integer x_count, y_count, z_count, i
real coord(1500), integer_v

```

```
common//coord(1500)

v = v + 750
integer_v = int (v)

if (integer_v.ge.1500.or.integer_v.le.0) goto 36

do 35 i= (integer_v - 5), (integer_v + 5),1
    v1 = float(i)

    func_denom=(1.e0+4.e0*(pi**2)*(t2**2)*((v-v1)**2))

    func_v = 2.e0*t2/func_denom

    coord(i)=coord(i) + func_v
35 continue
36 continue

end
```


Appendix 3:

Line-shape Simulation (Hexagonal Model) (4.4.2)

C A f77 program to model susceptibility
 C induced Bfield in a regular environment
 C This adaptation is to predict the line shape produced
 C
 C

```
implicit none
real cell_size, pi
real radius, radmax
real x_point, y_point, z_point
real h0, t2, gamma
real sus_sphere, sus_fluid
real bx, by, bz
real bxold, byold, bzold
real unit_cell(50,50,50)
real rndm1, rndm2, rand
real alpha, beta, sign
parameter (h0 = 1.0)
character*132 filename
integer x_count, y_count, z_count
integer i1, k, steps
common//unit_cell(50,50,50)
real coord(1500)
common//coord(1500)
integer orient, j
```

```
write(6,'(A,$)'), 'Please enter the filename: '
read(5,*) filename
```

```
write(6,'(A,$)'), 'Please enter unit cell size(um): '
read(5,*) cell_size
```

```
radmax = cell_size / 2.e0
```

C radmax in um!

```
10 write(6,'(A,$)'), 'Please enter sphere radius (um): '
read(5,*) radius
```

```
if (radius.gt.radmax) then
  write(6,'(A, f10.2, A)'),
&    'radius greater than', radmax, ',not allowed.'
  goto 10
endif
```

```
write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
read(5,*) sus_sphere
```

```
write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
read(5,*) sus_fluid
```

```
write(6,'(A,$)'), 'Please enter T2 value (s): '
read(5,*) t2
```

```
write(6,'(A,$)'), 'Please enter number of small steps: '
read(5,*) steps
```

```
write(6,'(A,$)'), 'Please enter number of orientations: '
read(5,*) orient
```

```

radius = radius*1.0e-6
cell_size = cell_size*1.0e-6
rndm1 = rand(1)
rndm2 = rand(1)
x_point = 0.e0
y_point = 0.e0
z_point = 0.e0
gamma = 26.7519e+7
pi=3.141592653

```

C CREATES A UNIT CELL OF 50*50*50 VALUES, THEN MAKES EQUAL TO
C ZERO OR PREVIOUSLY CALCULATED BFIELD VALUE

```

open(unit=2, file=filename, status= 'new')

do 37 j=0,orient
do 7 x_count=1,50
do 8 y_count=1,50
do 9 z_count=1,50
unit_cell(x_count, y_count, z_count) = 0
9      continue
8      continue
7      continue

write(6,*), 'unit_cell formed'

```

C CREATES THE X-COORDINATE USING A RANDOM NUMBER GENERATOR
C AND ROTATIONS ABOUT THE X AND Y AXES IN MANY SMALL STEPS

```

bxold = 0
byold = 0
bzold = 1

do 38 k=1,steps

rndm1 = rand(0)

if (rndm1.lt.0.5) then

rndm2 = rand(0)
if (rndm2.lt.0.5) then
sign = (-1.e0)

elseif (rndm2.gt.0.5) then
sign = (1.e0)

endif

alpha = (sign*pi)/36.e0

bx = bxold
by = (byold*cos(alpha)) + (bzold*sin(alpha))
bz = (bzold*cos(alpha)) - (byold*sin(alpha))

```

```

    byold = by
    bzold = bz

    elseif (rndm1.gt.0.5) then

        rndm2 = rand(0)
        if (rndm2.lt.0.5) then
            sign = (-1.e0)

            elseif (rndm2.gt.0.5) then
                sign = (1.e0)

        endif

        beta = (sign*pi)/36.e0

        bx = (bxold*cos(beta)) - (bzold*sin(beta))
        by = byold
        bz = (bzold*cos(beta)) + (bxold*sin(beta))

        bxold = bx
        bzold = bz

    endif

38    continue

    call sphere_position(radius, h0, cell_size,
&                        x_point, y_point, z_point,
&                        sus_sphere, sus_fluid, bx, by, bz)

    call calc_lineshape(radius, x_count, y_count, z_count,
&                        cell_size, gamma, t2, pi)

37    continue

    do 1 i1=1,1500

        write (2,*) (i1-750), coord(i1)

1    continue

    close(2)

    end

    subroutine sphere_position(radius, h0, cell_size,
&                            x_point, y_point, z_point,
&                            sus_sphere, sus_fluid, bx, by, bz)

C PLACES SPHERES IN HEXAGONAL CLOSE PACKED
C ARRAY POSITION FOR UNIT CELL AND SURROUNDING UNIT CELLS
C COMMENT OUT IF BFIELD BEING READ IN!!!!

```

C NOTE - CELL_SIZE = L

```

real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real radius, h0, cell_size, x_point, y_point, z_point
real sus_sphere, sus_fluid, bx, by, bz
integer x_count, y_count, z_count
real sphere_x, sphere_y, sphere_z
integer sphere

do 11 sphere = 1,59

if (sphere.eq.1) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.2) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.3) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.4) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.5) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.6) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.7) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (3.e0*(cell_size))/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.8) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.9) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.10) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

```

```

elseif (sphere.eq.11) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.12) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.13) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.14) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.15) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.16) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.17) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.18) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.19) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.20) then
  sphere_x = 0.e0
  sphere_y = -1.e0*(cell_size)
  sphere_z = 0.e0

elseif (sphere.eq.21) then
  sphere_x = 0.e0
  sphere_y = 0.e0
  sphere_z = 0.e0

elseif (sphere.eq.22) then
  sphere_x = 0.e0
  sphere_y = cell_size
  sphere_z = 0.e0

```

```

elseif (sphere.eq.23) then
  sphere_x = 0.e0
  sphere_y = 2.e0*cell_size
  sphere_z = 0.e0

elseif (sphere.eq.24) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.25) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.26) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.27) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = -1.e0*(cell_size)
  sphere_z = 0.e0

elseif (sphere.eq.28) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 0.e0
  sphere_z = 0.e0

elseif (sphere.eq.29) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = cell_size
  sphere_z = 0.e0

elseif (sphere.eq.30) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 2.e0*cell_size
  sphere_z = 0.e0

elseif (sphere.eq.31) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.32) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.33) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.34) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.35) then

```

```

sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = 0.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.36) then
sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = cell_size
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.37) then
sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = 2.e0*cell_size
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.38) then
sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = -1.e0*(cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.39) then
sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = (cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.40) then
sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
sphere_y = (3.e0*(cell_size))/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.41) then
sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = -1.e0*(cell_size)
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.42) then
sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = 0.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.43) then
sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = cell_size
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.44) then
sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = 2.e0*cell_size
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.45) then
sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = -1.e0*(cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.46) then
sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
sphere_y = (cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.47) then
sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)

```



```

sphere_y = 3.e0*(cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.48) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.49) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.50) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.51) then
  sphere_x = 0.e0
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.52) then
  sphere_x = 0.e0
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.53) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.54) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.55) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.56) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = -1.e0*(cell_size)
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.57) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.58) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.59) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 2.e0*cell_size

```

```

    sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

endif

    call add_bfield(x_point, y_point, z_point, sphere_x,
&    sphere_y, sphere_z, radius, h0,
&    sus_sphere, sus_fluid, cell_size,
&    x_count, y_count, z_count, bx, by, bz)
11  continue

end

subroutine add_bfield(x_point, y_point, z_point, sphere_x,
&    sphere_y, sphere_z, radius, h0,
&    sus_sphere, sus_fluid, cell_size,
&    x_count, y_count, z_count, bx, by, bz)
C  ADDS TOGETHER EACH CONTRIBUTION TO ALTER THE BFIELD FOR
C  EACH OF THE SURROUNDING SPHERES

    real unit_cell(50,50,50)
    common//unit_cell(50,50,50)
    real b, bx, by, bz
    real x_point, y_point, z_point, sphere_x, sphere_y, sphere_z
    real radius, h0, sus_sphere, sus_fluid, cell_size
    integer x_count, y_count, z_count

    do 1 x_count=1,50
        x_point = float((x_count - 1.e0)* (sqrt(3.e0))
&    *cell_size / 49.e0)

        do 2 y_count=1,50
            y_point = float((y_count - 1.e0)* cell_size / 49.e0)

            do 3 z_count=1,50
                z_point = float((z_count - 1.e0)
&    *(sqrt(13.e0) / 4.e0)*cell_size / 49.e0)

                b=0.e0

                call bfield(x_point, y_point, z_point,
&    sphere_x, sphere_y,
&    sphere_z, radius, h0, sus_sphere
&    , sus_fluid, b, bx, by, bz)

                unit_cell(x_count, y_count, z_count)
&    = unit_cell(x_count, y_count, z_count) + b

3          continue
2        continue
1      continue

```

```

end

subroutine bfield(x_point, y_point, z_point, sphere_x,
&               sphere_y, sphere_z, radius, h0,
&               sus_sphere, sus_fluid, b, bx, by, bz)

C  CALCULATES THE CHANGE IN BFIELD DUE TO THE
C  SUSC. DIFFERENCE FOR A PARTICULAR SPHERE

real const_num, const_den, constants
real del_x, del_y, del_z, distance, cos_theta
real x_point, y_point, z_point, sphere_x
real sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b, bx, by, bz

del_x = (x_point - sphere_x)
del_y = (y_point - sphere_y)
del_z = (z_point - sphere_z)

distance = sqrt(del_x**2 + del_y**2 + del_z**2)

if (distance.le.radius) then
  distance = radius
endif

cos_theta = (del_x*bx) + (del_y*by) + (del_z*bz)
&          /(sqrt(bx**2+by**2+bz**2)*distance)

const_num = (sus_sphere - sus_fluid)
&          *(radius*radius*radius)

const_den = (sus_sphere + 2.e0*sus_fluid + 3.e0)
&          *(distance*distance*distance)

constants = const_num / const_den

C  bfield = h0 * (1.e0 - constants*(1 - 3*cos_theta**2))
b = h0 * (- constants*(1 - 3*cos_theta**2))

return
end

subroutine calc_lineshape(radius, x_count, y_count,
&               z_count, cell_size, gamma,
&               t2, pi)

C  ADDS TOGETHER THE INDIVIDUAL LINESHAPES FOR ALL ARRAY POINTS

real radius, cell_size, start
integer x_count, y_count, z_count
real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real pi, t2, v, b
real x_point, y_point, z_point
real gamma, pi
real coord(1500)
common//coord(1500)

```

```

do 31 x_count=1,50
  x_point = (x_count - 1.e0)* (sqrt(3.e0))
  &      *cell_size / 49.e0

  do 32 y_count=1,50
    y_point = (y_count - 1.e0)* cell_size / 49.e0

    do 33 z_count=1,50
      z_point = (z_count - 1.e0)
      &      *(sqrt(13.e0) / 4.e0)* cell_size / 49.e0

      start=0.e0

      call add_in_out(x_point, y_point, z_point,
      &      radius, start, cell_size)

      if (start.eq.0.e0) then

        b=unit_cell(x_count,y_count,z_count)

        v=gamma*b/(2*pi)

        call linefreq (v, x_count, y_count, z_count,
      &      t2, pi)

      endif

33      continue
32      continue
31      continue

end

subroutine add_in_out(x_point, y_point, z_point, radius,
&      start, cell_size)
C CHECKS TO SEE IF MOLECULE IS IN OR OUT OF ANY
C OF THE SPHERES FOR A SPECIFIC MOVE

real x_point, y_point, z_point, start, radius
real sphere_x, sphere_y, sphere_z
real in_out_value, cell_size
real in_out_sphere

do 15 sphere_x=0.e0*cell_size,1.e0*cell_size,cell_size

  do 16 sphere_y=0.e0*cell_size,1.e0*cell_size,cell_size

    do 17 sphere_z=0.e0*cell_size,1.e0*cell_size,cell_size

      in_out_value = in_out_sphere(radius, x_point,
      &      y_point, z_point, sphere_x,
      &      sphere_y, sphere_z)

```

```

        start = start + in_out_value
17      continue
16      continue
15      continue

      end

      real function in_out_sphere(radius, x_point, y_point,
&          z_point, sphere_x, sphere_y, sphere_z)

C  CHECKS TO SEE IF MOLECULE IS IN OR OUTSIDE OF A SPECIFIC
C  SPHERE FOR A SPECIFIC MOVE

      real distance, radius
      real x_point, y_point, z_point
      real del_x, del_y, del_z
      real sphere_x, sphere_y, sphere_z

      del_x = (x_point - sphere_x)
      del_y = (y_point - sphere_y)
      del_z = (z_point - sphere_z)

      distance = sqrt(del_x**2 + del_y**2 + del_z**2)
      if (distance.le.radius) then
        in_out_sphere=1.e0
      else
        in_out_sphere=0.e0
      endif

      end

      subroutine linefreq (v, x_count, y_count, z_count, t2,
&          pi)

C  CALCULATES LORENTZIAN LINE SHAPE FOR FREQUENCY+-5 Hz

      real v, v1, pi, t2
      real unit_cell(50,50,50)
      real func_v, func_denom
      common//unit_cell(50,50,50)
      integer x_count, y_count, z_count, i
      real coord(1500), integer_v
      common//coord(1500)

      v = v + 750.e0
      integer_v = int (v)

      if (integer_v.ge.1500.or.integer_v.le.0) goto 36

      do 35 i= (integer_v - 5), (integer_v + 5),1

        v1 = float(i)

        func_denom=(1.e0+4.e0*(pi**2)*(t2**2)*((v-v1)**2))

        func_v = 2.e0*t2/func_denom

```

```
        coord(i)=coord(i) + func_v  
35  continue  
36  continue  
end
```

Appendix 4:

Line-shape Simulation (Random Model) (4.4.3)

```

C A f77 program to model susceptibility
C induced Bfield in a regular environment
C This adaptation is to create random packed Bfields
C
C
  implicit none
  real cell_size, pi
  real radius, radmax
  real x_point, y_point, z_point
  real h0, t2, gamma, pc
  real sus_sphere, sus_fluid
  real unit_cell(200,200,200)
  real random, rand
  parameter (h0 = 1.0)
  character*132 filename
  integer x_count, y_count, z_count, i1
  common/blocke/unit_cell(200,200,200)
  real coord(1000)
  common/blockd/coord(1000)

      real coord_x(10000), coord_y(10000), coord_z(10000),
&      radi(10000)
      common/blockb/coord_x(10000), coord_y(10000),
&      coord_z(10000), radi(10000)
      integer numc, numt
      real radmin
      common/blocka/ radmin
  random = rand(1)

c   write(6,'(A,$)'), 'Please enter the filename: '
c   read(5,*) filename

  write(6,'(A,$)'), 'Please enter unit cell size(um): '
  read(5,*) cell_size

  radmax = cell_size / 2.e0

C   radmax in um!

10  write(6,'(A,$)'), 'Please enter MAXIMUM sphere radius (um): '
    read(5,*) radius

    if (radius.gt.radmax) then
      write(6,'(A, f10.2, A)'),
&      'radius greater than', radmax, ',not allowed.'
      goto 10
    endif

100 write(6,'(A,$)'), 'Please enter MINIMUM sphere radius (um): '
    read(5,*) radmin

    if (radmin.gt.radius) then
      write(6,'(A, f10.2, A)'),
&      'MINIMUM radius greater than MAXIMUM not allowed!'
      goto 100
    endif

```



```

write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
read(5,*) sus_sphere

write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
read(5,*) sus_fluid

write(6,'(A,$)'), 'Please enter T2 value (s): '
read(5,*) t2

radius = radius*1.0e-6
radmin = radmin*1.0e-6
cell_size = cell_size*1.0e-6
x_point = 0.e0
y_point = 0.e0
z_point = 0.e0
gamma = 26.7519e+7
pi=3.141592653

C CREATES A UNIT CELL OF 200*200*200 VALUES, THEN MAKES EQUAL TO
C ZERO OR PREVIOUSLY CALCULATED BFIELD VALUE

open(unit=2, file='bfield', status= 'new')
open(unit=3, file='coord', status= 'new')

pc = 0.e0

do 7 z_count=1,200

    pc=pc+0.5

    do 8 y_count=1,200

        do 9 x_count=1,200

            unit_cell(x_count, y_count, z_count) = 0.e0

9            continue

8            continue

            write(6,*) pc,'% UC'

7        continue

        write(6,*), 'unit_cell formed'

c        generates sphere coordinates

        call sphere_position(radius, h0, cell_size,
&                             x_point, y_point, z_point,
&                             sus_sphere, sus_fluid, numt)

c        numt last ok here
write(6,*), 'sphere position finished!!'

call write_output (x_count, y_count, z_count,
&                  x_point, y_point, z_point, cell_size)

close(2)

```

```

close(3)

end

subroutine sphere_position(radius, h0, cell_size,
&      x_point, y_point, z_point,
&      sus_sphere, sus_fluid, numt)

C PLACES SPHERES IN CUBIC ARRAY POSITION FOR UNIT CELL AND
C SURROUNDING UNIT CELLS
C COMMENT OUT IF BFIELD BEING READ IN!!!!

real unit_cell(200,200,200)
common/blockc/unit_cell(200,200,200)
real radius, h0, cell_size, x_point, y_point, z_point
real sus_sphere, sus_fluid
integer x_count, y_count, z_count, success
real sphere_x, sphere_y, sphere_z

real coord_x(10000), coord_y(10000), coord_z(10000),
&      radi(10000)
common/blockb/coord_x(10000), coord_y(10000),
&      coord_z(10000), radi(10000)
real radmin
common/blocka/radmin
integer numc, numt, i2
double precision bx, by, bz, brad, bfit, fit
integer junk, yesno, i1, i2, i3, i4, i5, i6, i7, i10
real temp1,temp2, temp3

real random, rand

numc=0
write(6,*)'Sphere number'
110 numc=numc + 1
numt = numc -1
success=0
150 continue

bfit=0.e0
c start parameter loops here

c do 700 i3 =1, 10
radi(numc) = radmin +(radius - radmin)*rand(0)

do 701 i4 =1, 50
coord_x(numc) =(3.e0*rand(0) -1.e0)*cell_size

do 702 i5= 1, 50
coord_y(numc) =(3.e0*rand(0) -1.e0)*cell_size

do 703 i6 =1, 50

```

```

coord_z(numc) =(3.e0*rand(0) -1.e0)*cell_size

if(numc .eq. 1)goto 110
c    now test for overlap
yesno = 0

do 120 i1 =1, (numc-1)

    temp1 = (coord_x(i1) - coord_x(numc))**2
    temp1 = temp1 + (coord_y(i1) - coord_y(numc))**2
    temp1 = temp1 + (coord_z(i1) - coord_z(numc))**2

    temp1 =temp1**0.5

    temp1= temp1 -(radi(i1)+radi(numc))
    if(temp1 .lt. 0.e0)yesno=1
120    continue

if(yesno .eq. 0)then

c    calculate closeness factor to other spheres
c    and compare to previous most close position
c    and keep the best one
c    bx, by, bz, bfit, fit
c    success=1

    fit =0.e0
    do 704 i7 = 1, numt
        temp1 = (coord_x(i7) - coord_x(numc))**2
        temp1 = temp1 + (coord_y(i7) - coord_y(numc))**2
        temp1 = temp1 + (coord_z(i7) - coord_z(numc))**2
        temp1 =temp1**0.5

        temp2=radi(i7) + radi(numc)
        temp3=temp1 - temp2
        if(temp3 .gt. radi(numc))goto 704

c        NOW CALC A SCORE, fit
c        fit = fit + radi(numc)/temp3
        fit = fit + 100.e0*(1.e0- temp3/radi(numc))
704    continue

    if(fit .gt. bfit)then
        bx =coord_x(numc)
        by =coord_y(numc)
        bz =coord_z(numc)
        brad=radi(numc)
        bfit = fit
    endif

endif

```

c perhaps the following two ifs should become loops

```
703      continue
702      continue
701      continue
c       700      continue
```

c at this point, keep the best sphere parameters
c or if no found, goto 160

```
if(success .eq. 0)goto 160
```

```
write(6,*)numt, bfit
      coord_x(numc) = bx
      coord_y(numc) = by
      coord_z(numc) = bz
      radi(numc) = brad
```

```
160      goto 110
      continue
```

```
write(6,*)'sphere packing complete'
```

```
call write_coords(numt,cell_size)
```

c generate b field for each bead

```
call add_bfield(x_point, y_point, z_point, sphere_x,
&              sphere_y, sphere_z, radius, h0,
&              sus_sphere, sus_fluid, cell_size,
&              x_count, y_count, z_count, numt)
```

```
write(6,*)'after add_bfield numt'
write(6,*)numt
```

```
end
```

```
subroutine add_bfield(x_point, y_point, z_point, sphere_x,
&                    sphere_y, sphere_z, radius, h0,
&                    sus_sphere, sus_fluid, cell_size,
&                    x_count, y_count, z_count, numt)
```

C ADDS TOGETHER EACH CONTRIBUTION TO ALTER THE BFIELD FOR
C EACH OF THE SURROUNDING SPHERES

```
real unit_cell(200,200,200)
```

```

common/blockc/unit_cell(200,200,200)
real b
real x_point, y_point, z_point, sphere_x, sphere_y, sphere_z
real radius, h0, sus_sphere, sus_fluid, cell_size
integer x_count, y_count, z_count

      real coord_x(10000), coord_y(10000), coord_z(10000),
&      radi(10000)
      common/blockb/coord_x(10000), coord_y(10000),
&      coord_z(10000), radi(10000)
      integer numc, numt

do 1 z_count=1,200
  z_point = float((z_count - 1.e0)* cell_size / 199.e0)

  do 2 y_count=1,200
    y_point = float((y_count - 1.e0)* cell_size / 199.e0)

    do 3 x_count=1,200
      x_point = float((x_count - 1.e0)* cell_size / 199.e0)

      b=0.e0

      call bfield(x_point, y_point, z_point,
&                sphere_x, sphere_y,
&                sphere_z, radius, h0, sus_sphere
&                , sus_fluid,b, numt)

      unit_cell(x_count, y_count, z_count) = b
c    &      = unit_cell(x_count, y_count, z_count) + b

3      continue
2      continue
1      continue

      write(6,*)'magnetic field calculation complete'

      write(6,*)'numt'
      write(6,*)numt

end

subroutine bfield(x_point, y_point, z_point, sphere_x,
&                sphere_y, sphere_z, radius, h0,
&                sus_sphere, sus_fluid, b, numt)
C  CALCULATES THE CHANGE IN BFIELD DUE TO THE
C  SUSC. DIFFERENCE FOR A PARTICULAR SPHERE

real const_num, const_den, constants
real del_x, del_y, del_z, distance, cos_theta
real x_point, y_point, z_point, sphere_x

```

```

real sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b, junk

      real coord_x(10000), coord_y(10000), coord_z(10000),
&      radi(10000)
      common/blockb/coord_x(10000), coord_y(10000),
&      coord_z(10000), radi(10000)
      integer numc, numt, i1
      real pi

      pi=3.141592654

      b=0.e0
c      go through all sphere_x's etc and radii
c      loop from here
      junk=radius
      do 170 i1=1, numt
      radius = radi(i1)
      sphere_x = coord_x(i1)
      sphere_y = coord_y(i1)
      sphere_z = coord_z(i1)

      del_x = (x_point - sphere_x)
      del_y = (y_point - sphere_y)
      del_z = (z_point - sphere_z)

      distance = sqrt(del_x**2 + del_y**2 + del_z**2)

      if (distance.le.radius) then
      distance = radius
      endif

      cos_theta = del_z/distance

      const_num = (sus_sphere - sus_fluid)
&      *(radius*radius*radius)*4.e0*pi

      const_den = ((sus_sphere + 2.e0*sus_fluid)*4.e0*pi + 3.e0)
&      *(distance*distance*distance)

      constants = const_num / const_den

C      bfield = h0 * (1.e0 - constants*(1 - 3*cos_theta**2))
c      as summing over all spheres, ADD contributions to b

      b = b+ h0 * ( - constants*(1 - 3*cos_theta**2))

c      end loop here
170      continue
c      return
c      radius=junk
      end

c      sub to write formatted

```

```

subroutine write_output (x_count, y_count, z_count,
&      x_point, y_point, z_point, cell_size)

real x_point, y_point, z_point, cell_size
integer x_count, y_count, z_count
real unit_cell(200,200,200)
integer unit_cell_i(200,200,200)
common/blocke/unit_cell(200,200,200)

do 3 z_count=1,200
  do 2 y_count=1,200
    do 1 x_count=1,200

      unit_cell_i(x_count, y_count, z_count) =
&    int(1.e+8*unit_cell(x_count, y_count, z_count))
      write (2,4) (unit_cell_i(x_count, y_count, z_count))
4    format (i6)

1    continue
2    continue
3    continue

end

c  subroutine to write sphere  number, coords, and radii

subroutine write_coords (numt,cell_size)

integer numt, i10
real cell_size
real coord_x(10000), coord_y(10000), coord_z(10000),
&    radi(10000)
common/blockb/coord_x(10000), coord_y(10000),
&    coord_z(10000), radi(10000)

write (3,*) numt, cell_size

do 20 i10=1,numt
  write(3,*)coord_x(i10), coord_y(i10), coord_z(i10), radi(i10)
20 continue

end

```

Appendix 5:

Line-shape Simulation with Random B_0 (Cubic Model) (4.4.4)

C A f77 program to model susceptibility
 C induced Bfield in a regular environment
 C This adaptation is to predict the line shape produced
 C
 C

```
implicit none
real cell_size, pi
real radius, radmax
real x_point, y_point, z_point
real h0, t2, gamma
real sus_sphere, sus_fluid
real unit_cell(100,100,100)
real random, rand
parameter (h0 = 1.0)
character*132 filename
integer x_count, y_count, z_count, i1
common//unit_cell(100,100,100)
real coord(1000)
common//coord(1000)
```

```
write(6,'(A,$)'), 'Please enter the filename: '
read(5,*) filename
```

```
write(6,'(A,$)'), 'Please enter unit cell size(um): '
read(5,*) cell_size
```

```
radmax = cell_size / 2.e0
```

C radmax in um!

10 write(6,'(A,\$)'), 'Please enter sphere radius (um): '
 read(5,*) radius

```
if (radius.gt.radmax) then
  write(6,'(A, f10.2, A)'),
&      'radius greater than', radmax, ',not allowed.'
  goto 10
endif
```

```
write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
read(5,*) sus_sphere
```

```
write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
read(5,*) sus_fluid
```

```
write(6,'(A,$)'), 'Please enter T2 value (s): '
read(5,*) t2
```

```
radius = radius*1.0e-6
cell_size = cell_size*1.0e-6
random = rand(1)
x_point = 0.e0
y_point = 0.e0
z_point = 0.e0
gamma = 26.7519e+7
pi=3.141592653
```

C CREATES A UNIT CELL OF 100*100*100 VALUES, THEN MAKES EQUAL TO
 C ZERO OR PREVIOUSLY CALCULATED BFIELD VALUE

```

open(unit=2, file=filename, status= 'new')

do 7 x_count=1,100
  do 8 y_count=1,100
    do 9 z_count=1,100
      unit_cell(x_count, y_count, z_count) = 0
9      continue
8      continue
7      continue

write(6,*), 'unit_cell formed'

call sphere_position(radius, h0, cell_size,
&                  x_point, y_point, z_point,
&                  sus_sphere, sus_fluid)

call calc_lineshape(radius, x_count, y_count, z_count,
&                  cell_size, gamma, t2, pi)

do 1 i1=1,1000
  write (2,*) (i1-500), coord(i1)
1  continue

close(2)

end

subroutine sphere_position(radius, h0, cell_size,
&                  x_point, y_point, z_point,
&                  sus_sphere, sus_fluid)
C PLACES SPHERES IN CUBIC ARRAY POSITION FOR UNIT CELL AND
C SURROUNDING UNIT CELLS
C COMMENT OUT IF BFIELD BEING READ IN!!!!

real unit_cell(100,100,100)
common//unit_cell(100,100,100)
real radius, h0, cell_size, x_point, y_point, z_point
real sus_sphere, sus_fluid
integer x_count, y_count, z_count
real sphere_x, sphere_y, sphere_z

do 11 sphere_x = -1.e0*cell_size,2.e0*cell_size,cell_size

write(6,*)sphere_x

```

```

do 12 sphere_y=-1.e0*cell_size,2.e0*cell_size,cell_size

do 13 sphere_z=-1.e0*cell_size,2.e0*cell_size,cell_size

call add_bfield(x_point, y_point, z_point, sphere_x,
& sphere_y, sphere_z, radius, h0,
& sus_sphere, sus_fluid, cell_size,
& x_count, y_count, z_count)

13 continue

12 continue

11 continue

end

subroutine add_bfield(x_point, y_point, z_point, sphere_x,
& sphere_y, sphere_z, radius, h0,
& sus_sphere, sus_fluid, cell_size,
& x_count, y_count, z_count)

C ADDS TOGETHER EACH CONTRIBUTION TO ALTER THE BFIELD FOR
C EACH OF THE SURROUNDING SPHERES

real unit_cell(100,100,100)
common//unit_cell(100,100,100)
real b
real x_point, y_point, z_point, sphere_x, sphere_y, sphere_z
real radius, h0, sus_sphere, sus_fluid, cell_size
integer x_count, y_count, z_count

do 1 x_count=1,100
x_point = float((x_count - 1.e0)* cell_size / 99.e0)

do 2 y_count=1,100
y_point = float((y_count - 1.e0)* cell_size / 99.e0)

do 3 z_count=1,100
z_point = float((z_count - 1.e0)* cell_size / 99.e0)

b=0.e0

call bfield(x_point, y_point, z_point,
& sphere_x, sphere_y,
& sphere_z, radius, h0, sus_sphere
& , sus_fluid,b)

unit_cell(x_count, y_count, z_count)
& = unit_cell(x_count, y_count, z_count) + b

3 continue

2 continue

```

1 continue

end

```
subroutine bfield(x_point, y_point, z_point, sphere_x,
&               sphere_y, sphere_z, radius, h0,
&               sus_sphere, sus_fluid, b)
```

C CALCULATES THE CHANGE IN BFIELD DUE TO THE
C SUSC. DIFFERENCE FOR A PARTICULAR SPHERE

```
real const_num, const_den, constants
real del_x, del_y, del_z, distance, cos_theta
real x_point, y_point, z_point, sphere_x
real sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b
```

```
del_x = (x_point - sphere_x)
del_y = (y_point - sphere_y)
del_z = (z_point - sphere_z)
```

```
distance = sqrt(del_x**2 + del_y**2 + del_z**2)
```

```
if (distance.le.radius) then
  distance = radius
endif
```

```
cos_theta = (del_x + del_y + del_z) / ((sqrt(3.e0))*distance)
```

```
const_num = (sus_sphere - sus_fluid)
&          *(radius*radius*radius)
```

```
const_den = (sus_sphere + 2.e0*sus_fluid + 3.e0)
&          *(distance*distance*distance)
```

```
constants = const_num / const_den
```

C bfield = $h_0 * (1.e_0 - constants * (1 - 3 * \cos_{\theta}^2))$
b = $h_0 * (- constants * (1 - 3 * \cos_{\theta}^2))$

```
return
end
```

```
subroutine calc_lineshape(radius, x_count, y_count,
&                         z_count, cell_size, gamma,
&                         t2, pi)
```

C ADDS TOGETHER THE INDIVIDUAL LINESHAPES FOR ALL ARRAY POINTS

```
real radius, cell_size, start
integer x_count, y_count, z_count
real unit_cell(100,100,100)
common//unit_cell(100,100,100)
real pi, t2, v, b
real x_point, y_point, z_point
real gamma, pi
real coord(1000)
common//coord(1000)
```

```

do 31 x_count=1,100
  x_point = (x_count - 1.e0)* cell_size / 99.e0

do 32 y_count=1,100
  y_point = (y_count - 1.e0)* cell_size / 99.e0

do 33 z_count=1,100
  z_point = (z_count - 1.e0)* cell_size / 99.e0

  start=0.e0

  call add_in_out(x_point, y_point, z_point,
&               radius, start, cell_size)

  if (start.eq.0.e0) then

    b=unit_cell(x_count,y_count,z_count)

    v=gamma*b/(2*pi)

    call linefreq (v, x_count, y_count, z_count,
&               t2, pi)

  endif

33  continue

32  continue

31  continue

end

subroutine add_in_out(x_point, y_point, z_point, radius,
&               start, cell_size)

C CHECKS TO SEE IF MOLECULE IS IN OR OUT OF ANY
C OF THE SPHERES FOR A SPECIFIC MOVE

real x_point, y_point, z_point, start, radius
real sphere_x, sphere_y, sphere_z
real in_out_value, cell_size
real in_out_sphere

do 15 sphere_x=0.e0*cell_size,1.e0*cell_size,cell_size

do 16 sphere_y=0.e0*cell_size,1.e0*cell_size,cell_size

do 17 sphere_z=0.e0*cell_size,1.e0*cell_size,cell_size

  in_out_value = in_out_sphere(radius, x_point,
&               y_point, z_point, sphere_x,
&               sphere_y, sphere_z)

```

```

        start = start + in_out_value
17      continue
16    continue
15  continue

  end

  real function in_out_sphere(radius, x_point, y_point,
&      z_point, sphere_x, sphere_y, sphere_z)

C CHECKS TO SEE IF MOLECULE IS IN OR OUTSIDE OF A SPECIFIC
C SPHERE FOR A SPECIFIC MOVE

  real distance, radius
  real x_point, y_point, z_point
  real del_x, del_y, del_z
  real sphere_x, sphere_y, sphere_z

  del_x = (x_point - sphere_x)
  del_y = (y_point - sphere_y)
  del_z = (z_point - sphere_z)

  distance = sqrt(del_x**2 + del_y**2 + del_z**2)
  if (distance.le.radius) then
    in_out_sphere=1.e0
  else
    in_out_sphere=0.e0
  endif

  end

  subroutine linefreq (v, x_count, y_count, z_count, t2,
&      pi)

C CALCULATES LORENTZIAN LINE SHAPE FOR FREQUENCY+-5 Hz

  real v, v1, pi, t2
  real unit_cell(100,100,100)
  real func_v, func_denom
  common//unit_cell(100,100,100)
  integer x_count, y_count, z_count, i
  real coord(1000), integer_v
  common//coord(1000)

  integer_v = int (v) + 500

  if (integer_v.ge.1000.or.integer_v.le.0) goto 36

  do 35 i= (integer_v - 5), (integer_v + 5),1

    v1 = float(i)

    func_denom=(1+4*(pi**2)*(t2**2)*((v-v1)**2))

    func_v = 2*t2/func_denom

```

```
        coord(i)=coord(i) + func_v  
35  continue  
36  continue  
end
```

Appendix 6:

Line-shape Simulation with Random B_0 (Hexagonal Model) (4.4.4)

C A f77 program to model susceptibility
 C induced Bfield in a regular environment
 C This adaptation is to predict the line shape produced

C
 C

```
implicit none
real cell_size, pi
real radius, radmax
real x_point, y_point, z_point
real h0, t2, gamma
real sus_sphere, sus_fluid
real bx, by, bz
real unit_cell(50,50,50)
real random, rand
real phi, alpha, beta
parameter (h0 = 1.0)
character*132 filename
integer x_count, y_count, z_count, i1
common//unit_cell(50,50,50)
real coord(1500)
common//coord(1500)
integer orient, j
```

```
write(6,'(A,$)'), 'Please enter the filename: '
read(5,*) filename
```

```
write(6,'(A,$)'), 'Please enter unit cell size(um): '
read(5,*) cell_size
```

```
radmax = cell_size / 2.e0
```

C radmax in um!

10 write(6,'(A,\$)'), 'Please enter sphere radius (um): '
 read(5,*) radius

```
if (radius.gt.radmax) then
  write(6,'(A, f10.2, A)'),
& 'radius greater than', radmax, ',not allowed.'
  goto 10
endif
```

```
write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
read(5,*) sus_sphere
```

```
write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
read(5,*) sus_fluid
```

```
write(6,'(A,$)'), 'Please enter T2 value (s): '
read(5,*) t2
```

```
write(6,'(A,$)'), 'Please enter number of orientations: '
read(5,*) orient
```

```
radius = radius*1.0e-6
cell_size = cell_size*1.0e-6
random = rand(1)
x_point = 0.e0
y_point = 0.e0
```

```

z_point = 0.e0
gamma = 26.7519e+7
pi=3.141592653

```

C CREATES A UNIT CELL OF 50*50*50 VALUES, THEN MAKES EQUAL TO
C ZERO OR PREVIOUSLY CALCULATED BFIELD VALUE

```

open(unit=2, file=filename, status= 'new')

do 37 j=0,orient
do 7 x_count=1,50
do 8 y_count=1,50
do 9 z_count=1,50
unit_cell(x_count, y_count, z_count) = 0
9    continue
8    continue
7    continue

write(6,*), 'unit_cell formed'

random = rand(0)
phi = 2.e0*pi*random

random = rand(0)
alpha = 2.e0*pi*random

random = rand(0)
beta = 2.e0*pi*random

bx = (sin(phi))*(cos(alpha))
by = ((sin(phi))*(sin(alpha))*(cos(beta))) + (cos(phi)*sin(beta))
bz = (cos(phi))*(cos(beta)) - (sin(phi)*sin(alpha)*sin(beta))

call sphere_position(radius, h0, cell_size,
&                    x_point, y_point, z_point,
&                    sus_sphere, sus_fluid, bx, by, bz)

call calc_lineshape(radius, x_count, y_count, z_count,
&                    cell_size, gamma, t2, pi)

37  continue

do 1 i1=1,1500
write (2,*) (i1-750), coord(i1)
1  continue

close(2)

end

```

```

subroutine sphere_position(radius, h0, cell_size,
&                          x_point, y_point, z_point,
&                          sus_sphere, sus_fluid, bx, by, bz)

C PLACES SPHERES IN HEXAGONAL CLOSE PACKED
C ARRAY POSITION FOR UNIT CELL AND SURROUNDING UNIT CELLS
C COMMENT OUT IF BFIELD BEING READ IN!!!!

C NOTE - CELL_SIZE = L

real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real radius, h0, cell_size, x_point, y_point, z_point
real sus_sphere, sus_fluid, bx, by, bz
integer x_count, y_count, z_count
real sphere_x, sphere_y, sphere_z
integer sphere

do 11 sphere = 1,59

if (sphere.eq.1) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.2) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.3) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.4) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.5) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.6) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.7) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (3.e0*(cell_size))/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.8) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

```

```

elseif (sphere.eq.9) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.10) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.11) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.12) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.13) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.14) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.15) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.16) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = -1.e0*((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.17) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.18) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.19) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.20) then
  sphere_x = 0.e0
  sphere_y = -1.e0*(cell_size)
  sphere_z = 0.e0

```

```

elseif (sphere.eq.21) then
  sphere_x = 0.e0
  sphere_y = 0.e0
  sphere_z = 0.e0

elseif (sphere.eq.22) then
  sphere_x = 0.e0
  sphere_y = cell_size
  sphere_z = 0.e0

elseif (sphere.eq.23) then
  sphere_x = 0.e0
  sphere_y = 2.e0*cell_size
  sphere_z = 0.e0

elseif (sphere.eq.24) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.25) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.26) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.27) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = -1.e0*(cell_size)
  sphere_z = 0.e0

elseif (sphere.eq.28) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 0.e0
  sphere_z = 0.e0

elseif (sphere.eq.29) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = cell_size
  sphere_z = 0.e0

elseif (sphere.eq.30) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 2.e0*cell_size
  sphere_z = 0.e0

elseif (sphere.eq.31) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size))/2.e0
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.32) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size))/2.e0
  sphere_y = (cell_size)/2.e0
  sphere_z = 0.e0

elseif (sphere.eq.33) then

```

```

sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/2.e0)
sphere_y = 3.e0*(cell_size)/2.e0
sphere_z = 0.e0

elseif (sphere.eq.34) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.35) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.36) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.37) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.38) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.39) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.40) then
  sphere_x = ((sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (3.e0*(cell_size))/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.41) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = -1.e0*(cell_size)
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.42) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.43) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.44) then
  sphere_x = (3.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 2.e0*cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.45) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)

```

```

sphere_y = -1.e0*(cell_size)/2.e0
sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.46) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.47) then
  sphere_x = (5.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.48) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.49) then
  sphere_x = (7.e0*(sqrt(3.e0)*cell_size)/4.e0)
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/4.e0)

elseif (sphere.eq.50) then
  sphere_x = -1.e0*((sqrt(3.e0)*cell_size)/2.e0)
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.51) then
  sphere_x = 0.e0
  sphere_y = 0.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.52) then
  sphere_x = 0.e0
  sphere_y = cell_size
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.53) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = -1.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.54) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = (cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.55) then
  sphere_x = (sqrt(3.e0)*cell_size)/2.e0
  sphere_y = 3.e0*(cell_size)/2.e0
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.56) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = -1.e0*(cell_size)
  sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.57) then
  sphere_x = (sqrt(3.e0)*cell_size)
  sphere_y = 0.e0

```

```

    sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.58) then
    sphere_x = (sqrt(3.e0)*cell_size)
    sphere_y = cell_size
    sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

elseif (sphere.eq.59) then
    sphere_x = (sqrt(3.e0)*cell_size)
    sphere_y = 2.e0*cell_size
    sphere_z = ((sqrt(13.e0)*cell_size)/2.e0)

endif

    call add_bfield(x_point, y_point, z_point, sphere_x,
&    sphere_y, sphere_z, radius, h0,
&    sus_sphere, sus_fluid, cell_size,
&    x_count, y_count, z_count, bx, by, bz)
11 continue

end

subroutine add_bfield(x_point, y_point, z_point, sphere_x,
&    sphere_y, sphere_z, radius, h0,
&    sus_sphere, sus_fluid, cell_size,
&    x_count, y_count, z_count, bx, by, bz)
C  ADDS TOGETHER EACH CONTRIBUTION TO ALTER THE BFIELD FOR
C  EACH OF THE SURROUNDING SPHERES

    real unit_cell(50,50,50)
    common//unit_cell(50,50,50)
    real b, bx, by, bz
    real x_point, y_point, z_point, sphere_x, sphere_y, sphere_z
    real radius, h0, sus_sphere, sus_fluid, cell_size
    integer x_count, y_count, z_count

    do 1 x_count=1,50
        x_point = float((x_count - 1.e0)* (sqrt(3.e0))
&    *cell_size / 49.e0)

        do 2 y_count=1,50
            y_point = float((y_count - 1.e0)* cell_size / 49.e0)

            do 3 z_count=1,50
                z_point = float((z_count - 1.e0)
&    *(sqrt(13.e0) / 4.e0)*cell_size / 49.e0)

                b=0.e0

                call bfield(x_point, y_point, z_point,
&    sphere_x, sphere_y,
&    sphere_z, radius, h0, sus_sphere
&    , sus_fluid, b, bx, by, bz)

                unit_cell(x_count, y_count, z_count)

```



```

&          = unit_cell(x_count, y_count, z_count) + b

3          continue
2          continue
1          continue

end

subroutine bfield(x_point, y_point, z_point, sphere_x,
&              sphere_y, sphere_z, radius, h0,
&              sus_sphere, sus_fluid, b, bx, by, bz)

C  CALCULATES THE CHANGE IN BFIELD DUE TO THE
C  SUSC. DIFFERENCE FOR A PARTICULAR SPHERE

real const_num, const_den, constants
real del_x, del_y, del_z, distance, cos_theta
real x_point, y_point, z_point, sphere_x
real sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b, bx, by, bz

del_x = (x_point - sphere_x)
del_y = (y_point - sphere_y)
del_z = (z_point - sphere_z)

distance = sqrt(del_x**2 + del_y**2 + del_z**2)

if (distance.le.radius) then
    distance = radius
endif

cos_theta = (del_x*bx) + (del_y*by) + (del_z*bz)
&          /(sqrt(bx**2+by**2+bz**2)*distance)

const_num = (sus_sphere - sus_fluid)
&          *(radius*radius*radius)

const_den = (sus_sphere + 2.e0*sus_fluid + 3.e0)
&          *(distance*distance*distance)

constants = const_num / const_den

C  bfield = h0 * (1.e0 - constants*(1 - 3*cos_theta**2))
C  b = h0 * ( - constants*(1 - 3*cos_theta**2))

return
end

subroutine calc_lineshape(radius, x_count, y_count,
&              z_count, cell_size, gamma,
&              t2, pi)

C  ADDS TOGETHER THE INDIVIDUAL LINESHAPES FOR ALL ARRAY POINTS

real radius, cell_size, start

```

```

integer x_count, y_count, z_count
real unit_cell(50,50,50)
common//unit_cell(50,50,50)
real pi, t2, v, b
real x_point, y_point, z_point
real gamma, pi
real coord(1500)
common//coord(1500)

do 31 x_count=1,50
  x_point = (x_count - 1.e0)* (sqrt(3.e0))
&      *cell_size / 49.e0

  do 32 y_count=1,50
    y_point = (y_count - 1.e0)* cell_size / 49.e0

    do 33 z_count=1,50
      z_point = (z_count - 1.e0)
&      *(sqrt(13.e0) / 4.e0)* cell_size / 49.e0

      start=0.e0

      call add_in_out(x_point, y_point, z_point,
&      radius, start, cell_size)

      if (start.eq.0.e0) then

        b=unit_cell(x_count,y_count,z_count)

        v=gamma*b/(2*pi)

        call linefreq (v, x_count, y_count, z_count,
&      t2, pi)

      endif

33    continue
32  continue
31  continue

end

subroutine add_in_out(x_point, y_point, z_point, radius,
&      start, cell_size)

C CHECKS TO SEE IF MOLECULE IS IN OR OUT OF ANY
C OF THE SPHERES FOR A SPECIFIC MOVE

real x_point, y_point, z_point, start, radius
real sphere_x, sphere_y, sphere_z
real in_out_value, cell_size
real in_out_sphere

do 15 sphere_x=0.e0*cell_size,1.e0*cell_size,cell_size

```

```

do 16 sphere_y=0.e0*cell_size,1.e0*cell_size,cell_size

  do 17 sphere_z=0.e0*cell_size,1.e0*cell_size,cell_size

    in_out_value = in_out_sphere(radius, x_point,
&      y_point, z_point, sphere_x,
&      sphere_y, sphere_z)

    start = start + in_out_value

17    continue
16    continue
15  continue

  end

  real function in_out_sphere(radius, x_point, y_point,
&      z_point, sphere_x, sphere_y, sphere_z)

C  CHECKS TO SEE IF MOLECULE IS IN OR OUTSIDE OF A SPECIFIC
C  SPHERE FOR A SPECIFIC MOVE

  real distance, radius
  real x_point, y_point, z_point
  real del_x, del_y, del_z
  real sphere_x, sphere_y, sphere_z

  del_x = (x_point - sphere_x)
  del_y = (y_point - sphere_y)
  del_z = (z_point - sphere_z)

  distance = sqrt(del_x**2 + del_y**2 + del_z**2)
  if (distance.le.radius) then
    in_out_sphere=1.e0
  else
    in_out_sphere=0.e0
  endif

  end

  subroutine linefreq (v, x_count, y_count, z_count, t2,
&      pi)

C  CALCULATES LORENZIAN LINE SHAPE FOR FREQUENCY+-5 Hz

  real v, v1, pi, t2
  real unit_cell(50,50,50)
  real func_v, func_denom
  common//unit_cell(50,50,50)
  integer x_count, y_count, z_count, i
  real coord(1500), integer_v
  common//coord(1500)

  integer_v = int (v) + 750

```

```
if (integer_v.ge.1500.or.integer_v.le.0) goto 36
do 35 i= (integer_v - 5), (integer_v + 5),1
  v1 = float(i)
  func_denom=(1+4*(pi**2)*(t2**2)*((v-v1)**2))
  func_v = 2*t2/func_denom
  coord(i)=coord(i) + func_v
35 continue
36 continue
end
```

Appendix 7:

Diffusion Experiment Simulation in Cubic Model

(4.4.5)

```

C A f77 program to model susceptibility
C and diffusion in a regular environment
C
C
  implicit none
  real cell_size
  real radius, radmax
  double precision x_point, y_point, z_point
  real h0
  real sus_sphere, sus_fluid
  real pi, diff_const
  real unit_cell(100,100,100)
  real delta, CDELTA, diff_z_axis
  real time_step, step_size
  real random, rand
  real phase, angle
  real start
  real gamma
  parameter (h0 = 1.0)
c   character*132 filename1
  character*132 filename2
  integer x_count, y_count, z_count
  integer molecules, i, n, expts, i1
  common//unit_cell(100,100,100)
  real cos_phase, total_phase, total_signal, mean_signal

  total_phase = 0.0
  total_signal = 0

c   write(6,'(A,$)'), 'Please enter the reference bfield filename: '
c   read(5,*) filename1

  write(6,'(A,$)'), 'Please enter the output filename: '
  read(5,*) filename2

  write(6,'(A,$)'), 'Please enter unit cell size(um): '
  read(5,*) cell_size

  radmax = cell_size / 2.e0

C   radmax in um!

10  write(6,'(A,$)'), 'Please enter sphere radius (um): '
    read(5,*) radius

    if (radius.gt.radmax) then
      write(6,'(A, f10.2, A)'),
&      'radius greater than', radmax, ',not allowed.'
      goto 10
    endif

  write(6,'(A,$)'), 'Please enter the number of experiments: '
  read(5,*) expts

  write(6,'(A,$)'), 'Please enter sphere volume susceptibility: '
  read(5,*) sus_sphere

  write(6,'(A,$)'), 'Please enter fluid volume susceptibility: '
  read(5,*) sus_fluid

  write(6,'(A,$)'), 'Please enter number of molecules: '

```

```

read(5,*) molecules

write(6,'(A,$)'), 'Please enter duration of delta(us): '
read(5,*) delta

write(6,'(A,$)'), 'Please enter time step of spins(us): '
read(5,*) time_step

radius = radius*1.0e-6
cell_size = cell_size*1.0e-6
delta = delta*1.0e-6
time_step = time_step*1.0e-6

c  open(unit=2, file=filename1, status= 'old')
  open(unit=3, file=filename2, status= 'new')

do 7 z_count=1,100

  do 8 y_count=1,100

    do 9 x_count=1,100

c      read (2,*)unit_cell(z_count, y_count, x_count)

      unit_cell(z_count, y_count, x_count)=0.e0

9      continue
8      continue
7      continue

      write(6,*), 'unit_cell formed'

c    close(2)

      call sphere_position(radius, h0, cell_size,
&                          x_point, y_point, z_point,
&                          sus_sphere, sus_fluid)

do 22 i1=1,expts
11  write(6,'(A,$)'), 'Please enter duration of CDELTAS(us): '
    read(5,*) CDELTA

    CDELTA = CDELTA*1.0e-6

    if (CDELTA.le.delta) then
      write(6,'(A)'),'Error, CDELTA < delta !'
      goto 11
    endif

    pi=3.141592653
    diff_const = 2.299e-9
    random = rand(1)
    x_point = 0.e0

```

```

y_point = 0.e0
z_point = 0.e0
phase = 0.e0
angle = 0.e0
cos_phase=0.e0
total_signal=0.e0
total_phase=0.e0
mean_signal=0.e0
gamma = 26.7519e+7
diff_z_axis = CDELTA - delta
n=0.e0

step_size = sqrt(2.e0*diff_const*time_step)

c   write(6,*), 'Step size (1d) is ', step_size, 'm'

do 50 i=1, molecules

c   if (i.eq.28871) then
c     call wakeup
c   endif

phase = 0.e0
14  start = 0.e0

random=rand(0)
x_point=random*cell_size

random=rand(0)
y_point=random*cell_size

random=rand(0)
z_point=random*cell_size

call add_in_out(x_point, y_point, z_point,
&              radius, start, cell_size)

if (start.ge.1.e0) then
goto 14
endif

c   write(6,*)i

c   write(3,*)'DEPHASING'

call change_phase(x_point, y_point, z_point, phase, radius,
&                step_size, delta, time_step, gamma, i,
&                x_count, y_count, z_count, cell_size)

c   write(3,*)'ALONG Z'

call change_position(x_point, y_point, z_point,
&                  radius, diff_z_axis, time_step,
&                  cell_size, step_size)

phase = phase*(-1.e0)
c   write(3,*)'REPHASING'

```



```

call change_phase(x_point, y_point, z_point, phase, radius,
&                step_size, delta, time_step, gamma, i,
&                x_count, y_count, z_count, cell_size)

cos_phase = cos(phase)

total_signal= total_signal + cos_phase

total_phase = total_phase + phase

c   write (3,*) phase

n=n+1

50  continue

mean_signal = total_signal/molecules
total_phase = total_phase/molecules

write (3,*) 'Total Phase Change =', total_phase,
& ' ->0 if random generator is Normal'
write (3,*) 'Signal =', mean_signal
write (3,*) 'number molecules =', n

write(6,*)i1

22  continue

close (3)

end

subroutine sphere_position(radius, h0, cell_size,
&                        x_point, y_point, z_point,
&                        sus_sphere, sus_fluid)

real unit_cell(100,100,100)
common//unit_cell(100,100,100)
real radius, h0, cell_size
double precision x_point, y_point, z_point
real sus_sphere, sus_fluid
integer x_count, y_count, z_count
real sphere_x, sphere_y, sphere_z

do 11 sphere_z = -1.e0*cell_size,2.e0*cell_size,cell_size

write(6,*)sphere_z

do 12 sphere_y=-1.e0*cell_size,2.e0*cell_size,cell_size

do 13 sphere_x=-1.e0*cell_size,2.e0*cell_size,cell_size

call add_bfield(x_point, y_point, z_point, sphere_x,
&              sphere_y, sphere_z, radius, h0,
&              sus_sphere, sus_fluid, cell_size,

```

```

&          x_count, y_count, z_count)

13      continue

12      continue

11      continue

end

subroutine add_bfield(x_point, y_point, z_point, sphere_x,
&          sphere_y, sphere_z, radius, h0,
&          sus_sphere, sus_fluid, cell_size,
&          x_count, y_count, z_count)

real unit_cell(100,100,100)
common//unit_cell(100,100,100)
real b
double precision x_point, y_point, z_point
real sphere_x, sphere_y, sphere_z
real radius, h0, sus_sphere, sus_fluid, cell_size
integer x_count, y_count, z_count

do 1 z_count=1,100
    z_point = dble(z_count - 1.d0)* cell_size / 99.d0

    do 2 y_count=1,100
        y_point = dble(y_count - 1.d0)* cell_size / 99.d0

        do 3 x_count=1,100
            x_point = dble(x_count - 1.d0)* cell_size / 99.d0

            b=0.e0

            call bfield(x_point, y_point, z_point,
&                sphere_x, sphere_y,
&                sphere_z, radius, h0, sus_sphere
&                , sus_fluid,b)

            unit_cell(z_count, y_count, x_count)
&            = unit_cell(z_count, y_count, x_count) + b

3          continue

2          continue

1          continue

end

subroutine bfield(x_point, y_point, z_point, sphere_x,
&          sphere_y, sphere_z, radius, h0,
&          sus_sphere, sus_fluid, b)

real const_num, const_den, constants

```

```

double precision del_x, del_y, del_z, distance, cos_theta
double precision x_point, y_point, z_point
real sphere_x, sphere_y, sphere_z, radius, h0
real sus_sphere, sus_fluid, b
real pi

pi=3.141592654

del_x = dble(x_point - sphere_x)
del_y = dble(y_point - sphere_y)
del_z = dble(z_point - sphere_z)

distance = sqrt(del_x**2 + del_y**2 + del_z**2)

if (distance.le.radius) then
  distance = radius
endif

cos_theta = del_z/distance

const_num = (sus_sphere - sus_fluid)
&          *(radius*radius*radius)*4.e0*pi

const_den = ((sus_sphere + 2.e0*sus_fluid)*4.e0*pi + 3.e0)
&          *(distance*distance*distance)

constants = const_num / const_den
C  bfield = h0 * (1.e0 - constants*(1 - 3*cos_theta**2))
  b = h0 * ( - constants*(1 - 3*cos_theta**2))

return
end

subroutine add_in_out(x_point, y_point, z_point, radius,
&                  start, cell_size)

double precision x_point, y_point, z_point
real start, radius
real sphere_x, sphere_y, sphere_z
real in_out_value, cell_size
real in_out_sphere

do 15 sphere_z=0.e0*cell_size,1.e0*cell_size,cell_size

  do 16 sphere_y=0.e0*cell_size,1.e0*cell_size,cell_size

    do 17 sphere_x=0.e0*cell_size,1.e0*cell_size,cell_size

      in_out_value = in_out_sphere(radius, x_point,
&                                y_point, z_point, sphere_x,
&                                sphere_y, sphere_z)

      start = start + in_out_value
17    continue

```

```

16    continue
15    continue

    end

    real function in_out_sphere(radius, x_point, y_point,
&        z_point, sphere_x, sphere_y, sphere_z)

    real radius
    double precision x_point, y_point, z_point
    double precision del_x, del_y, del_z, distance
    real sphere_x, sphere_y, sphere_z

    del_x = dble(x_point - sphere_x)
    del_y = dble(y_point - sphere_y)
    del_z = dble(z_point - sphere_z)

    distance = sqrt(del_x**2 + del_y**2 + del_z**2)
    if (distance.le.radius) then
    in_out_sphere=1.e0
    else
    in_out_sphere=0.e0
    endif

    end

    subroutine change_phase(x_point, y_point, z_point, phase, radius,
&        step_size, delta, time_step, gamma, i,
&        x_count, y_count, z_count, cell_size)

    integer x_count, y_count, z_count,i
    real start, step, lbf
    double precision x_point, y_point, z_point
    double precision x_point_old, y_point_old, z_point_old
    real phase,radius, step_size
    real delta, time_step, gamma
    real random
    real rand, cell_size
    real unit_cell(100,100,100)
    common//unit_cell(100,100,100)

    start=0.e0
    lbf=0.e0

    do 19 step=1.e0*time_step,delta,time_step
c    if (i.eq.28871) then
c    write(3,*) 'timestep =',step
c    endif

c    if ((i.eq.1371773).and.(step.ge.2.0e-3)) then
c        call wakeupagain
c    endif

18    random=rand(0)

        if (random.lt.0.5)then
            random=(-1)

```

```

else
    random=1
endif

x_point_old=x_point
x_point=dble(x_point+random*step_size)

if (x_point.gt.cell_size) then
    x_point=dble(step_size)
elseif(x_point.lt.0.e0) then
    x_point=dble(cell_size-step_size)
endif

random=rand(0)

if (random.lt.0.5)then
    random=(-1)
else
    random=1
endif

y_point_old=y_point
y_point=dble(y_point+random*step_size)

if (y_point.gt.cell_size) then
    y_point=dble(step_size)
elseif(y_point.lt.0.e0) then
    y_point=dble(cell_size-step_size)
endif

random=rand(0)

if (random.lt.0.5)then
    random=(-1)
else
    random=1
endif

z_point_old=z_point
z_point=dble(z_point+random*step_size)

if (z_point.gt.cell_size) then
    z_point=dble(step_size)
elseif(z_point.lt.0.e0) then
    z_point=dble(cell_size-step_size)
endif

call add_in_out(x_point, y_point, z_point, radius,
&              start, cell_size)

if (start.ge.1.e0) then

    call local_motion (x_point_old, step_size, y_point_old,
&                    z_point_old, radius, start, cell_size,
&                    x_point, y_point, z_point)

endif

call local_bfield(x_point, y_point,z_point,
&                x_count, y_count, z_count,

```

```

&          lbf, cell_size)

    phase = phase+(gamma*time_step*lbf)
c   if (i.eq.28871) then
c     write(3,*) x_point, y_point, z_point
c   endif
19  continue

    end

    subroutine change_position(x_point, y_point, z_point,
&          radius, diff_z_axis, time_step,
&          cell_size, step_size)

    real start, step, time_step
    double precision x_point, y_point, z_point
    real radius, step_size, diff_z_axis
    real random, rand
    real cell_size
    double precision x_point_old, y_point_old, z_point_old

    start=0.e0

    do 21 step=1.e0*time_step,diff_z_axis,time_step
c   write(3,*) 'timestep =',step
20  random=rand(0)

        if (random.lt.0.5)then
            random=(-1)
        else
            random=1
        endif

        x_point_old=x_point
        x_point=dble(x_point+random*step_size)

        if (x_point.gt.cell_size) then
            x_point=dble(step_size)
        elseif(x_point.lt.0.e0) then
            x_point=dble(cell_size-step_size)
        endif

        random=rand(0)

        if (random.lt.0.5)then
            random=(-1)
        else
            random=1
        endif

        y_point_old=y_point
        y_point=dble(y_point+random*step_size)

        if (y_point.gt.cell_size) then
            y_point=dble(step_size)

```

```

elseif(y_point.lt.0.e0) then
  y_point=dbl(ceil_size-step_size)
endif

random=rand(0)

if (random.lt.0.5)then
  random=(-1)
else
  random=1
endif

z_point_old=z_point
z_point=dbl(z_point+random*step_size)

if (z_point.gt.ceil_size) then
  z_point=dbl(step_size)
elseif(z_point.lt.0.e0) then
  z_point=dbl(ceil_size-step_size)
endif

call add_in_out(x_point, y_point, z_point, radius,
&               start, ceil_size)

if (start.ge.1.e0) then

  call local_motion (x_point_old, step_size, y_point_old,
&                   z_point_old, radius, start, ceil_size,
&                   x_point, y_point, z_point)

endif

c  write(3,*) x_point, y_point, z_point

21 continue

end

subroutine local_bfield(x_point, y_point,z_point,
&                      x_count, y_count, z_count,
&                      lbf, ceil_size)

integer x_count, y_count, z_count
integer x0, y0, z0
integer x1, y1, z1
real x0y0z0, x1y0z0, x0y1z0, x1y1z0
real x0y0z1, x1y0z1, x0y1z1, x1y1z1
real q1, q2, q3, q4
real f1, f2
double precision x_point, y_point,z_point
real lbf, ceil_size
real frac_x, frac_y, frac_z
double precision x_point_gap, y_point_gap, z_point_gap
real unit_cell(100,100,100)
common//unit_cell(100,100,100)
double precision x0_real, x1_real, y0_real
double precision y1_real, z0_real, z1_real

```

```
x_point_gap=dble(cell_size/99.d0)
y_point_gap=dble(cell_size/99.d0)
z_point_gap=dble(cell_size/99.d0)
```

```
x0 = int (x_point/x_point_gap)
y0 = int (y_point/y_point_gap)
z0 = int (z_point/z_point_gap)
```

```
x1 = x0 +1.e0
y1 = y0 +1.e0
z1 = z0 +1.e0
```

```
x0_real=dble(x0*x_point_gap)
x1_real=dble(x1*x_point_gap)
y0_real=dble(y0*x_point_gap)
y1_real=dble(y1*x_point_gap)
z0_real=dble(z0*x_point_gap)
z1_real=dble(z1*x_point_gap)
```

c CONVERT THE CO-ORDS TO ARRAY DIMENSIONS

```
x0 = x0 +1.e0
y0 = y0 +1.e0
z0 = z0 +1.e0
```

```

x1 = x1 +1.e0
c  if (x1.gt.100) then
c    x1=2.e0
c  endif
y1 = y1 +1.e0
c  if (y1.gt.100) then
c    y1=2.e0
c  endif
z1 = z1 +1.e0
c  if (z1.gt.100) then
c    z1=2.e0
c  endif
```

```
x0y0z0=unit_cell(x0,y0,z0)
x1y0z0=unit_cell(x1,y0,z0)
x0y1z0=unit_cell(x0,y1,z0)
x0y0z1=unit_cell(x0,y0,z1)
x1y1z0=unit_cell(x1,y1,z0)
x0y1z1=unit_cell(x0,y1,z1)
x1y0z1=unit_cell(x1,y0,z1)
x1y1z1=unit_cell(x1,y1,z1)
```

```
frac_x = float(x_point - x0_real)
frac_y = float(y_point - y0_real)
frac_z = float(z_point - z0_real)
```

```
q1=x0y0z0*frac_x + x1y0z0*(1-frac_x)
q2=x0y1z0*frac_x + x1y1z0*(1-frac_x)
q3=x0y0z1*frac_x + x1y0z1*(1-frac_x)
q4=x0y1z1*frac_x + x1y1z1*(1-frac_x)
```

```
f1=q1*frac_z + q3*(1-frac_z)
f2=q2*frac_z + q4*(1-frac_z)
```



```

lbf=f1*frac_y + f2*(1-frac_y)

end

subroutine local_motion (x_point_old, step_size, y_point_old,
&                        z_point_old, radius, start, cell_size,
&                        x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start
real step_size
double precision x_point, y_point, z_point

start=0.e0

call xy_only (x_point_old, step_size, y_point_old,
&             z_point_old, radius, start, cell_size,
&             x_point, y_point, z_point)

end

subroutine xy_only (x_point_old, step_size, y_point_old,
&                  z_point_old, radius, start, cell_size,
&                  x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

x_point=x_point
y_point=y_point

z_point=z_point_old

call add_in_out(x_point, y_point, z_point,
&              radius, start, cell_size)

if (start.ge.1.e0) then

call xz_only (x_point_old, step_size, y_point_old,
&             z_point_old, radius, start, cell_size,
&             x_point, y_point, z_point)

endif

end

subroutine xz_only (x_point_old, step_size, y_point_old,
&                  z_point_old, radius, start, cell_size,
&                  x_point, y_point, z_point)

```

```

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

x_point=x_point
z_point=z_point

y_point=y_point_old

      call add_in_out(x_point, y_point, z_point,
&                    radius, start, cell_size)

      if (start.ge.1.e0) then

        call yz_only (x_point_old, step_size, y_point_old,
&                    z_point_old, radius, start, cell_size,
&                    x_point, y_point, z_point)

      endif

end

subroutine yz_only (x_point_old, step_size, y_point_old,
&                  z_point_old, radius, start, cell_size,
&                  x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

y_point=y_point
z_point=z_point

x_point=x_point_old

      call add_in_out(x_point, y_point, z_point,
&                    radius, start, cell_size)

      if (start.ge.1.e0) then

        call x_only (x_point_old, step_size, y_point_old,
&                    z_point_old, radius, start, cell_size,
&                    x_point, y_point, z_point)

      endif

end

subroutine x_only (x_point_old, step_size, y_point_old,

```

```

&          z_point_old, radius, start, cell_size,
&          x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

x_point=x_point

y_point=y_point_old
z_point=z_point_old

      call add_in_out(x_point, y_point, z_point,
&          radius, start, cell_size)

      if (start.ge.1.e0) then

        call y_only (x_point_old, step_size, y_point_old,
&          z_point_old, radius, start, cell_size,
&          x_point, y_point, z_point)

      endif

end

subroutine y_only (x_point_old, step_size, y_point_old,
&          z_point_old, radius, start, cell_size,
&          x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

y_point=y_point

x_point=x_point_old
z_point=z_point_old

      call add_in_out(x_point, y_point, z_point,
&          radius, start, cell_size)

      if (start.ge.1.e0) then

        call z_only (x_point_old, step_size, y_point_old,
&          z_point_old, radius, start, cell_size,
&          x_point, y_point, z_point)

      endif

end

```

```
subroutine z_only (x_point_old, step_size, y_point_old,
&                z_point_old, radius, start, cell_size,
&                x_point, y_point, z_point)

double precision x_point_old, y_point_old, z_point_old
real radius, cell_size, start, step_size
double precision x_point, y_point, z_point

start=0.e0

z_point=z_point

y_point=y_point_old
x_point=x_point_old

      call add_in_out(x_point, y_point, z_point,
&                    radius, start, cell_size)

      if (start.ge.1.e0) then

        write(6,*) 'SPIN STUCK!!'

      endif

z_point=z_point_old
end

subroutine wakeup

write(6,*) 'Oi wake up'

end

subroutine wakeupagain

write(6,*) 'Oi wake up again'

end
```

Appendix 8:

Relaxation Errors in ADC (Spin Echo) (5.2.1)

```
c      F77 program to distinguish whether or not T2 relaxation
c      has any significant effect when calculating D from a
c      spin echo in a large constant gradient
```

```
implicit none
real D, gamma,increment, mfg
real T2, D1,real_t
real attn,error(10)
real sig2(100), comp1(100)
common//sig2(100)
common//comp1(100)
integer t,grad,tau_f
character*132 filename2
parameter (gamma = 267522128)
```

```
write(6,'(A,$)'), 'Please enter the output filename: '
read(5,'(A)') filename2
```

```
write(6,'(A,$)'), 'Please enter D(m2/s): '
read(5,*) D
```

```
c      D= 2.2e-9
```

```
write(6,'(A,$)'), 'Please enter % attenuation: '
read(5,*) attn
```

```
open(unit=4, file=filename2, status= 'new')
```

```
do 6 grad=5,150,5
```

```
  mfg=float(grad)
```

```
  call calc_tau(mfg,gamma,tau_f,D,increment,attn)
```

```
  do 5 t= -3,1,1
```

```
    real_t=float(t)
```

```
    T2=(10.e0)**(real_t)
```

```
    call calc_pts(increment,tau_f,gamma,mfg,D,T2)
```

```
    call calc_D (D1)
```

```
    error(t+4)=((D1/D)*100.e0)-100.e0
```

```
5      continue
      write(4,*) grad,error(1),error(2),error(3),error(4),error(5)
```

```
6      continue
```

```
close(4)
```

```
end
```

C *****SUBROUTINES*****

```
subroutine calc_pts (increment,tau_f,gamma,mfg,D,T2)
```

```
real gamma,D,T2,ln_signal_t2, complex1,mfg
integer increment, tau, tau_f, i1
real sig2(100), comp1(100), tau1
common//sig2(100),comp1(100)
```

```
i1=0
do 1 tau = increment,tau_f,increment
  i1 = i1 + 1
```

```
tau1 = float(tau)*1.e-9
```

```
ln_signal_t2 =-1.e0*((2.e0*tau1/T2)+(gamma**2)*
& (mfg**2)*(tau1**3)*(2.e0/3.e0)*D)
```

```
complex1 = -1.e0*(gamma**2)*(mfg**2)*(tau1**3)*(2.e0/3.e0)
```

```
sig2(i1) = ln_signal_t2
comp1(i1) = complex1
```

```
1 continue
```

```
end
```

```
subroutine calc_tau (mfg,gamma,tau_f,D,increment,attn)
```

```
real signal,D,ln_sig
real gamma,attn,mfg,t,t3
integer increment,tau_f
```

```
signal = 1.e0 - (attn/100.e0)
ln_sig=log(signal)
t3=-1.e0*ln_sig/((gamma**2)*(mfg**2)*(2.e0/3.e0)*D)
t=t3**(1.e0/3.e0)
t=t*1.e9
tau_f=int(t)
increment=tau_f/50
```

```
end
```

```
subroutine calc_D (D1)
```

```
real sum_x,sum_y,num,num1,den,den1
real mean_x,mean_y,D1
integer i
real sig2(100),comp1(100)
common//sig2(100),comp1(100)
```

```
sum_y=0.e0
sum_x=0.e0
num=0.e0
den=0.e0

do 3 i=1,50

    sum_y = sum_y + sig2(i)

    sum_x = sum_x + comp1(i)
3  continue

mean_x=sum_x/50.e0
mean_y=sum_y/50.e0

do 4 i=1,50

    num1=sig2(i)*comp1(i)-50*(sum_y*sum_x)
    num=num+num1

    den1=comp1(i)**2 - (sum_x**2)*50.e0
    den=den+den1
4  continue

D1=num/den

end
```


Appendix 9:

Relaxation Errors in ADC (Stimulated Echo) (5.2.2)

```

c      F77 program to distinguish whether or not T2 relaxation
c      has any significant effect when calculating D from a
c      stimulated echo in a large constant gradient
c      vary T maintain tau

```

```

implicit none
double precision real_t,attn1
double precision T2, D1,T1
double precision error(10)
double precision sig2(100), comp1(100)
double precision tau3, D, gamma, mfg, attn, diffmax
common//sig2(100)
common//comp1(100)
integer t,grad
character*132 filename2
parameter (gamma = 267522128)

```

```
tau3=0.d0
```

```

write(6,'(A,$)'), 'Please enter the output filename: '
read(5,'(A)') filename2

```

```

write(6,'(A,$)'), 'Please enter max diffusion time(us): '
read(5,*) diffmax
diffmax=diffmax*1.d-6

```

```

write(6,'(A,$)'), 'Please enter D(m2/s): '
read(5,*) D

```

```
c      D= 2.2d-9
```

```

write(6,'(A,$)'), 'Please enter % attenuation: '
read(5,*) attn

```

```
attn1=1.d0-(9.d-1*(attn/100.d0))
```

```
open(unit=4, file=filename2, status= 'new')
```

```
do 6 grad=5,150,5
```

```
mfg=dble(grad)
```

```
call calc_tau(mfg,gamma,D,attn,diffmax, tau3)
```

```
do 5 t= -2,1,1
```

```
real_t=dble(t)
```

```
T1=(10.d0)**(real_t)
```

```
call calc_pts(tau3,gamma,mfg,D,T2,diffmax,T1,attn1)
```

```
call calc_D (D1)
```

```
error(t+3)=((D1/D)*100.d0)-100.d0
```

```
5      continue
```

```
write(4,*) grad,error(1),error(2),error(3),error(4)
```

```
6      continue
```

```
close(4)
```

```
end
```

```
C *****SUBROUTINES*****
```

```
subroutine calc_tau (mfg,gamma,D,attn,diffmax, tau3)
```

```
double precision signal,D,ln_sig
double precision gamma,attn,mfg,diffmax
double precision sig_int, sig_old, M, C
double precision tau, tau3, tau_old
```

```
signal = 1.d0 - (attn/100.d0)
```

```
c ln_sig=log(signal)
```

```
c T=diffmax - tau - 2.d0*p1
```

```
tau=0.d0
```

```
sig_int=0.d0
```

```
10 tau_old = tau
```

```
tau=tau+1.d-6
```

```
sig_old=sig_int
```

```
sig_int=dexp(-1.d0*((gamma**2.d0)*(mfg**2.d0)*D*
& ((tau**2.d0)*(diffmax-(tau/3.d0))))))
```

```
if (sig_int.gt.signal) then
```

```
goto 10
```

```
endif
```

```
tau=tau*1.d6
```

```
tau_old=tau_old*1.d6
```

```
M=(tau-tau_old)/(sig_int-sig_old)
```

```
C=sig_int - M*tau
```

```
tau3=(signal - C)/M
```

```
tau3=tau3/1.d6
```

```
end
```

```
subroutine calc_pts (tau3,gamma,mfg,D,T2,diffmax,T1,attn1)
```

```
double precision T2,ln_signal_t1, complex1,T1,zmax,attn1
```

```
integer z_inc, i1,z
```

```
double precision sig2(100), comp1(100),delta,tdelta
```

```
double precision diffmax,tau3,gamma,D,mfg
```

```
common//sig2(100),comp1(100)
```

```
zmax=diffmax-tau3
```

```
zmax=zmax*1.0e9
```

```
z_inc=int(zmax/50.d0)
```

```
i1=0
```

```

do 1 z = z_inc,zmax,z_inc
  i1 = i1 + 1

  tdelta = z*1.d-9
  delta = tau3

  ln_signal_t1 = -1.d0*((tdelta/T1)+(gamma**2)
&      *(mfg**2)*(delta**2)*(tdelta+delta*(2.d0/3.d0))*D)

  if (i1.eq.1.and.ln_signal_t1.lt.attn1) then
    goto 11
  endif

  complex1 = -1.d0*(gamma**2)*(mfg**2)*((delta**2)*
&      (tdelta+delta*(2.d0/3.d0)))

  sig2(i1) = ln_signal_t1
  comp1(i1) = complex1
c   write (6,*) sig2(i1),comp1(i1)
1   continue
11  end

```

```

subroutine calc_D (D1)

double precision sum_x,sum_y,num,num1,den,den1
double precision mean_x,mean_y,D1
integer i
double precision sig2(100),comp1(100)
common//sig2(100),comp1(100)

sum_y=0.d0
sum_x=0.d0
num=0.d0
den=0.d0

do 3 i=1,50

  sum_y = sum_y + sig2(i)

  sum_x = sum_x + comp1(i)
3   continue

mean_x=sum_x/50.d0
mean_y=sum_y/50.d0

do 4 i=1,50

  num1=sig2(i)*comp1(i)-50*(sum_y*sum_x)
  num=num+num1

  den1=comp1(i)**2 - (sum_x**2)*50.d0
  den=den+den1

```

```
4  continue
    D1=num/den
end
```