From: BIOLOGICAL MAGNETIC RESONANCE, Vol. 8
Edited by Lawrence J. Berliner
and Jacques Reuben
(Plenum Publishing Corporation, 1989)

1

Calculating Slow Motional Magnetic Resonance Spectra

A User's Guide

David J. Schneider and Jack H. Freed

1. INTRODUCTION

In the first volume of *Spin Labeling: Theory and Applications* a chapter was written by one of us (J.H.F.) in which a detailed theory for the interpretation of ESR spectra of spin labels in the slow motional regime (Freed, 1976) was presented. The specific emphasis of that review was on the interpretation of nitroxide spin label spectra and contained many such examples. In the ensuing 13 years, there have been a number of important developments. First and foremost has been the development and implementation of powerful computational algorithms that have been specifically tailored for the solution of these types of problems (Moro and Freed, 1981; Vasavada *et al.*, 1987). The use of these algorithms often leads to more than an order-of-magnitude reduction in computer time for the calculation of any given spectrum as well as a dramatic reduction in computer memory requirements. Concomitant with these improvements in computational methodology has been the revolution in the power and availability of computer hardware. Taken together, these improvements in hardware and software have made it possible to quickly and conveniently perform spectral calculations on small laboratory computers which formerly required the resources of a large mainframe computer. The increase in the available computing power has also made it feasible to incorporate more sophisticated models of molecular structure and dynamics into the line-shape calculation programs.

In this chapter, we wish to make the results of the above-mentioned developments conveniently available to a wider audience. To this end, a diskette containing a set of programs for ESR line-shape programs has been included with this book (see inside back cover). These programs incorporate those features of the general theory that have been found to be most useful for the calculation of conventional cw ESR spectra of nitroxide spin labels in isotropic liquids, liquid crystals, and covalently labeled polymers. Our desire to supply a set of programs that are relatively compact and simple to use has necessarily involved some compromises in versatility and generality. Nevertheless, we feel that the compromises are not severe and that the programs supplied here will likely satisfy the needs of most of the readers. We will clearly state below the range of validity and limitations of the programs presented here. More general programs are, of course, available from the authors.

Unfortunately, time and space do not permit us to cover all of the applications of these programs that already exist in the literature, but a brief survey is provided in Section 5.2, along with appropriate references. The programs provided here should enable the reader to redo many of these calculations in order to gain familiarity with their use and to actually begin to compare the output with experimental spectra.

Once one begins to compare experimental and computed spectra in earnest, another basic problem emerges: the procedure of varying input parameters in a trial-and-error fashion to obtain a good fit to experimental data can be quite tedious and time-consuming for complicated systems, even with more efficient programs and faster computers. The manual fitting of spectra in this manner is as much an art as a science. However, since spectra can now be so quickly and reliably computed, it is no longer unreasonable to attempt the analysis of experimental spectra in a completely automated fashion using a general nonlinear least-squares fitting procedure to optimize the parameters characterizing the system (Crepeau *et al.*, 1987). Significant progress toward this goal has recently been made in our laboratory. Such a computation frequently requires the calculation of more than 100 spectra as the nonlinear least-squares procedure is iterated toward an optimal parameter set. Not many years ago this herculean task would have been unthinkable for complex slow motional spectra. In addition, much progress has recently been made toward the solution of the general problem of an automated determination of optimal basis sets (Vasavada *et al.*, 1987). It is important to realize that these developments have been made possible, in large part, by the more efficient algorithms on which the programs supplied with this book are based.

The set of programs described in this guide are designed for the efficient and accurate calculation of cw ESR spectra of doublet radicals ($S = \frac{1}{2}$), possibly coupled to one nuclear spin (e.g., typical nitroxide spin labels), in the limit of high static magnetic fields, slow molecular motions, and weak microwave fields. In its present form, it can also be used on a variety of NMR problems in the analogous case of an $I = \frac{1}{2}$ nucleus coupled to one other nuclear spin (Campbell *et al.*, 1979) and, for an $I = 1$ nucleus, by a simple modification (Meirovitch and Freed, 1979). However, these programs should not be used for the calculation of spectra in the low field or very fast motional limits, since the nonsecular terms that are omitted from the spin Hamiltonian in these programs give rise to significant effects in either of these cases. In particular, these programs should not be used for accurate calculations of X-band spectra of transition metal ions such as Cu^{2+} and VO^{2+}. Fast motional spectra can easily be calculated using a motional narrowing theory and a far simpler program, while low field spectra can be calculated by methods similar to those in these programs (Moro, 1980a; Meirovitch *et al.*, 1982; Gorcester, 1985).

These programs are an outgrowth of a set of programs written by G. Moro in 1980 (Moro, 1980a, 1980b; Moro and Freed, 1981; Meirovitch *et al.*, 1982). The older

David J. Schneider and Jack H. Freed • Baker Laboratory of Chemistry, Cornell University, Ithaca, New York 14853-1301. David J. Schneider's present address is IBM Corporation, Kingston, New York 12401.

programs were quite complicated as they were written in FORTRAN IV and highly optimized for use on a small computer with a very limited amount of physical memory. The present set of programs are written in a more modular, easily portable fashion in FORTRAN 77. As supplied, these programs should compile and run without error on IBM PCs[1] or compatibles under MS-DOS[2] (ver. 2.11 or higher) and Microsoft[2] Fortran (ver. 3.0 or higher). Because of the wide availability of this type of hardware and software, users of these programs can use the results of small calculations with the PC version as a reference point in porting these programs to a more powerful machine to be used for routine calculations. In addition, the faster PCs equipped with floating point coprocessors can themselves be used for many calculations.

Since this chapter is devoted primarily to a description of the programs and their implementation and use, it will be assumed that the reader is already familiar with the basic aspects of the theory as presented in the chapters by Nordio (1976) and Freed (1976) in the first volume of this series. References to equations in the article by Freed, denoted here as I, will be given whenever appropriate.

The body of this work is broken up into four main sections. First, in Section 2 a qualitative overview of the theory behind these programs, along with the necessary definitions and a summary of the types of terms which are included in the calculations, are presented. Next, Section 3 is devoted to a description of the algorithms used in the spectral calculations and related topics such as the continued fraction representation of the spectral line shape function. The implementation of the programs is discussed in Section 4, as well as suggestions on modifying them to run on other machines. Finally, in Section 5 a general strategy for fitting experimental data to these programs is presented as well as several model calculations and a survey of published results.

This organization was adopted so that the work could be utilized in two very different ways, depending on the interests and inclinations of the reader. One approach would be to simply find a PC, insert the diskette, and use Sections 4.3, 4.5, and 5 as a user's manual for the programs, referring to the definitions in Section 2 only when necessary. This pragmatic approach has the distinct advantage of quickly acquainting the user with the art of fitting theoretical models to experimental data. A second, more cautious, approach would be to begin by carefully studying the material in Section 2 before attempting to use thé programs. The reader who adopts this approach will probably make fewer serious errors in choosing appropriate parameters and interpreting the results. Regardless of the initial approach, there is no substitute for experience in using the programs and developing a feel for the effects of varying different input parameters on the calculated spectrum. Likewise, it is difficult to properly interpret the final results without being familiar with the theory behind the parameters and how they enter into the calculations. In addition, Section 3 should provide valuable insight into the workings, both in theory and in practice, of the algorithms on which the spectral calculations are based.

2. GENERAL THEORETICAL CONSIDERATIONS

The stochastic Liouville equation (SLE) is used here to describe the time evolution of the orientation-dependent density operator, and thereby the time evolution of the magnetization operator of the individual spins. Averaging this result over the ensemble of spins in the sample leads to an equation describing the time evolution of the macroscopic magnetization. The Fourier-Laplace transform of the time autocorrelation function of the x component of the macroscopic magnetization in the laboratory-fixed frame can be identified with the frequency-swept cw ESR spectrum in the limit of very weak microwave fields (i.e., in the linear response regime). However, in most experiments it is the field-swept rather than the frequency-swept cw ESR spectrum that is obtained. This restricts the direct application of the present method to the high-field limit where the results from the two types of experiments are essentially indistinguishable.

[1] IBM is a registered trademark of International Business Machines Corp.

[2] MS-DOS and Microsoft are registered trademarks of Microsoft Corporation.

For the present purposes, the stochastic Liouville equation amounts to treating the spin degrees of freedom of the system in a quantum-mechanical fashion, while the orientation of the molecule is determined by a classical stochastic process. The quantum-mechanical spin degrees of freedom are coupled to the classical orientational degrees of freedom through the anisotropic part of the orientation-dependent spin Hamiltonian, so that the tumbling of the molecule drives time-dependent fluctuations in the energy levels of the spin system. The choice of stochastic process used to model the reorientation of the molecules affects the time evolution of the spin system and therefore the resulting ESR spectrum. More detailed explanations and derivations of the ideas presented here are available in several places in the literature (Freed, 1976; Schneider and Freed, 1989).

The central result of the stochastic Liouville theory (Freed, 1976; Schneider and Freed, 1989) as applied to the magnetic resonance line-shape problem is that the unsaturated, high-field, frequency-swept cw ESR spectrum, $I(\omega - \omega_0)$, is given in compact "matrix element" notation (cf. equation I-7) as

$$I(\omega - \omega_0) = \left(\frac{1}{\pi}\right)\langle\!\langle v|[(\tilde{\Gamma} - i\mathscr{L}) + i(\omega - \omega_0)I]^{-1}|v\rangle\!\rangle \qquad (1)$$

In this equation, ω is the sweep frequency and $\omega_0 = g_0\beta_e B_0/\hbar$ where B_0 is the static magnetic field, $g_0 = \frac{1}{3}(g_{xx} + g_{yy} + g_{zz})$, β_e is the Bohr magneton, and $\hbar$ is Planck's constant divided by 2π. Also, $\mathscr{L}$ is the Liouville superoperator associated with the orientation-dependent spin Hamiltonian, $\tilde{\Gamma}$ is the "symmetrized" diffusion superoperator used to model the classical reorientational motion, and I is the identity operator. The starting vector, $|v\rangle\!\rangle$, includes both the spin operator for the allowed ESR transitions and the equilibrium probability distribution function for the orientation of the radicals. The quantity $(\tilde{\Gamma} - i\mathscr{L})$ is commonly referred to as the stochastic Liouville superoperator.

2.1. Terms Included in the Liouville and Diffusion Superoperators

The types of terms allowed in the Liouville and diffusion superoperators limit the types of systems which can be modeled. The Liouville superoperator used here is for doublet radicals ($S = \frac{1}{2}$) interacting with a single nucleus of arbitrary spin, such as a nitroxide. In particular, the secular and pseudosecular terms in the spin Hamiltonian arising from electron Zeeman and hyperfine interactions are included in the calculation. The nuclear quadrupole, nuclear Zeeman, and the nonsecular portions of the electron

Zeeman and hyperfine tensors are completely neglected. These terms can be included in a more complete treatment (Moro, 1980a; Meirovitch *et al.*, 1982; Gorcester, 1985), but are usually not necessary for nitroxide radicals in the slow motional and high-field limits.

The reorientational dynamics of the spin label (assumed to be rigid) is modeled by a symmetrized rotational diffusion superoperator with a restoring potential of the form (cf. equation I-55)

$$U(\Omega) = -k_B T\left\{\sum_{L=2,4} \lambda_0^L \mathcal{D}_{00}^L(\Omega) + \lambda_2^L[\mathcal{D}_{02}^L(\Omega) + \mathcal{D}_{0-2}^2(\Omega)]\right\} \tag{2}$$

where $\Omega = (\alpha, \beta, \gamma)$ represents a set of Euler angles describing the orientation of the radical relative to a laboratory-fixed frame (cf. Section 2.2). The functions $\mathcal{D}_{MK}^L(\Omega)$ are the generalized spherical harmonics or Wigner rotation matrix elements. We note that for $M = 0$,

$$\mathcal{D}_{0K}^L(\Omega) = \left(\frac{4\pi}{2L+1}\right)^{1/2} Y_{LK}(\beta, \gamma) \tag{3}$$

where the $Y_{LK}(\beta, \gamma)$ are the well-known spherical harmonics (Rose, 1957; Messiah, 1962; Biedenharn and Louck, 1981). A restoring potential of this type is appropriate for modeling the rotational dynamics of spin probes in uniaxial liquid-crystalline media. In addition to the diffusive type terms one may model jump processes between an arbitrary number of equivalent sites, and Heisenberg spin exchange interactions are included and can be utilized when appropriate. The overall "diffusion" operator is assumed to be the simple superposition of these terms. The user can choose between the three canonical models of Brownian, jump, and free diffusion (cf. equations I-34 to I-44) to describe the dynamics of radicals in isotropic media (Freed, 1976; Korst and Antsiferova, 1979).

2.2. Definitions of Coordinate Systems

Before discussing the types of terms which are included in the Liouville and diffusion superoperators in these calculations, several coordinate systems must be defined (cf. Figures 1 and 2 below). First, $(\hat{x}_L, \hat{y}_L, \hat{z}_L)$ is the laboratory-fixed reference frame, where the $\hat{z}_L$ axis is along the static magnetic field, the $\hat{y}_L$ axis is along the axis of polarization of the (linearly polarized) oscillating magnetic field of the microwave radiation, and the $\hat{x}_L$ axis is chosen to make a right-handed coordinate system.

The second frame of interest is the director frame, $(\hat{x}_d, \hat{y}_d, \hat{z}_d)$. The $\hat{z}_d$ axis of the director frame is taken to be along the symmetry axis of the restoring potential while the $\hat{x}_d$ and $\hat{y}_d$ axes are taken to be coincident with the $\hat{x}_L$ and $\hat{y}_L$ laboratory axes, respectively, when $\hat{z}_L \parallel \hat{z}_d$. With this choice of axes, the director tilt is defined by rotation through an angle ψ about the $\hat{y}_L$ axis which takes the $\hat{z}_L$ axis into the $\hat{z}_d$ axis. If there is no restoring potential present, the director frame is taken to be coincident with the laboratory frame. This generalization of distinct director and laboratory frames was not included in the programs given in I, but is included here, as it is important for the analysis of many experiments on model membranes and other liquid-crystalline samples.

The director and laboratory frames will be referred to as space-fixed frames, since they are defined to appear to the experimenter as being fixed in space and are related to one another by a simple (time-independent) coordinate transformation. The orientation of the laboratory and director frames for a simplified experiment involving spin probes dissolved in an oriented uniaxial liquid-crystal sample sandwiched between two glass plates is given in Figure 1. In this figure ψ would be positive, since this rotation would advance a right-handed screw along the positive $\hat{y}_L = \hat{y}_d$ axis. Note that in I the laboratory frame is denoted (x, y, z) and the director frame (x'', y'', z''). Also, both of the Euler angles used in I to relate laboratory and director frames are not needed for uniaxial liquid crystals.

The third coordinate system is defined by the principal axis system of the magnetic tensors, $(\hat{x}_m, \hat{y}_m, \hat{z}_m)$. Here it is assumed that the principal axis systems of the A and g tensors are coincident and rigidly fixed in the molecular framework. The final coordinate system is the principal axis system of the rotational diffusion tensor, $(\hat{x}_R, \hat{y}_R, \hat{z}_R)$. The diffusion tensor, R, is assumed to be axially symmetric, and the $\hat{z}_R$ axis is defined to be parallel to the unique axis of the diffusion tensor. Because of the assumed axial symmetry of the diffusion tensor, the $\hat{x}_R$ and $\hat{y}_R$ axes are arbitrary and can be chosen to simplify the calculation of the matrix elements. As in the case of the laboratory and diffusion frames, it is assumed that the diffusion frame can be rotated into the magnetic frame by a rotation through an angle ϕ about the $\hat{y}_R$ axis. The angle ϕ is called the diffusion tilt angle. In general, a second Euler angle is required to specify an arbitrary tilt between the diffusion and magnetic frames. This second angle, as well as a general tilt between the principal axis systems of the g and A tensors (Meirovitch *et al.*, 1982), have not been included in the present set of programs for simplicity. In practice, the quality of the rigid limit spectra for nitroxide spin labels rarely justifies the inclusion of these features in dynamical calculations.

The magnetic and diffusion frames are rigidly fixed with respect to the molecular framework and are therefore referred to as molecular or body-fixed frames. The relationship between the magnetic and diffusion frames for an idealized spin labeled

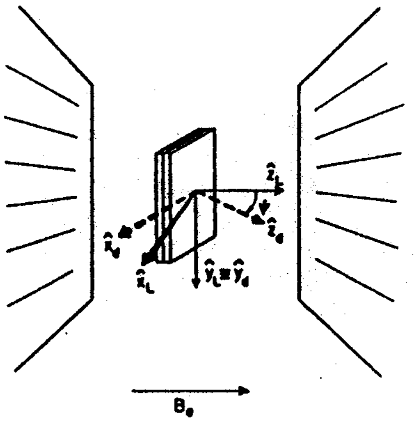

Figure 1. Diagram of laboratory and director frames for an experiment involving spin probes dissolved in an oriented liquid crystal sandwiched between two glass plates. Note that $\hat{z}_L \parallel B_0$ and that $\hat{x}_L$, $\hat{x}_d$, $\hat{z}_L$, and $\hat{z}_d$ all lie in the same plane. For most conventional experimental setups, where TE_{102} cavities are used, the $\hat{y}_L = \hat{y}_d$ axis is vertical when viewed by the experimenter in the laboratory while $\hat{z}_L$ is horizontal.

polymer is given in Figure 2. For simplicity, it is assumed here that the six-membered piperidine ring is planar and that the N—O bond lies in the symmetry plane defined by the ring. These simplistic assumptions are made only to simplify the diagram and the related discussion and are not required in general. In this idealized situation, symmetry restrictions constrain the principal axis systems of the g and A tensors to be coincident. In addition, symmetry arguments also require that one of the principal axes of the magnetic tensors lie along the N—O bond, one be perpendicular to the plane defined by the piperidine ring, and one be parallel to the line connecting C_2 and C_6. The established convention for labeling the magnetic axes is to take $\hat{z}_m$ as the axis which is perpendicular to the ring, the $\hat{x}_m$ axis along N—O bond, and the $\hat{y}_m$ axis parallel to the line connecting C_2 and C_6 (Freed, 1976; Lajzerowicz-Bonneteau, 1976) to give a right-handed coordinate system. A consequence of this convention is that A_{zz} is numerically the largest of the three principal components of the hyperfine interaction tensor for nitroxide spin labels based on the 2,2,6,6-tetramethylpiperidine-1-oxyl moiety (e.g., TEMPONE, TEMPOL, TEMPAMINE, etc.). It should be recognized that this choice is a matter of convention and that there are many situations when a different choice is very advantageous. It should be noted that in I the diffusion frame is denoted (x', y', z') and the magnetic frame (x''', y''', z''').

2.3. Basis Vectors and Scalar Product in Operator Space

The role of the SLE is to determine the time evolution of the orientation-dependent density operator. Hence the stochastic Liouville operator is a superoperator in the sense that it maps operators into operators rather that wave functions into wave functions. To use this formalism, a basis set of orientation-dependent spin operators must be introduced. The orientation-dependent density operator acts as both a classical probability distribution function for the orientation of the molecule relative to a space-fixed frame and a quantum-mechanical density operator for the spin system. As such, it can be expanded in the direct product space spanned by a complete orthonormal set of

spatial functions of the Euler angles specifying the orientation of the radical in space and a set of electron and nuclear spin projectors. The general properties of superoperators and bases in operator space have been treated by many authors (Fano, 1957; Jeener, 1982; Löwdin, 1982, 1985).

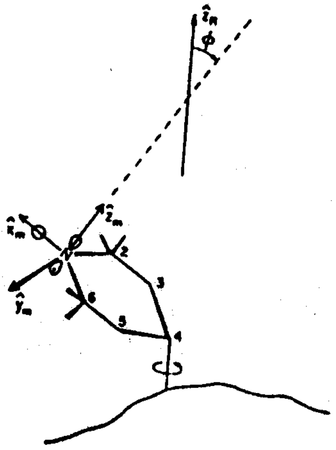

Figure 2. Diagram of the magnetic and diffusion frames for an experiment involving an idealized spin-labeled polymer. It is assumed that $\hat{z}_R$ is parallel to the covalent bond between C_4 and the polymer and that there is rapid rotation about this bond modulated by the slow overall tumbling of the polymer backbone. In accordance with the standard convention, $\hat{x}_m$ is taken along the N—O bond, $\hat{y}_m$ is parallel to the line connecting C_2 and C_6, and $\hat{z}_m$ is normal to the plane defined by the idealized flat piperidine ring.

A convenient and advantageous choice of spatial functions for rotational diffusion problems such as this are the normalized Wigner rotation matrix elements (Freed, 1976). Using Dirac notation, the basis functions are

$$\langle \Omega | L, M, K\rangle = \sqrt{\frac{2L+1}{8\pi^2}}\,\mathcal{D}^L_{MK}(\Omega) \tag{4}$$

where the indices L, M, and K are integral, $0 \le L$, and $|M|, |K| \le L$. These functions form an orthonormal set in the sense that

$$\begin{aligned}\langle L_1, M_1, K_1 | L_2, M_2, K_2\rangle &= \int d\Omega \langle L_1, M_1, K_1|\Omega\rangle\langle\Omega| L_2, M_2, K_2\rangle \\ &= N_L(L_1, L_2)\left(\frac{1}{8\pi^2}\right)\int d\Omega\, \mathcal{D}^{L_1 *}_{M_1 K_1}(\Omega)\mathcal{D}^{L_2}_{M_2 K_2}(\Omega) \\ &= \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\end{aligned} \tag{5}$$

where the normalization factor $N_L(L_1, L_2)$ is defined as

$$N_L(L_1, L_2) = \sqrt{(2L_1+1)(2L_2+1)} \tag{6}$$

This choice of basis enables one to easily apply group theoretical arguments in evaluating matrix elements.

The most convenient basis for the spin part of the basis vector for ESR problems is a direct product of projectors for the electronic spins (with quantum numbers S and m_S) and nuclear spins (with quantum numbers I and m_I):

$$|p^S, q^S; p^I, q^I\rangle\!\rangle = (|S, m_S\rangle\langle S, m_S'|) \otimes (|I, m_I\rangle\langle I, m_I'|) \tag{7}$$

where $p^S = m_S - m_S'$ and $q^S = m_S + m_S'$, and similarly for p^I and q^I. In the following discussions the principal spin quantum numbers S and I will be suppressed. These direct product projectors are complete and orthonormal with respect to the trace metric defined as

$$\langle\!\langle \rho|\sigma\rangle\!\rangle = \mathrm{tr}\{\rho^\dagger\sigma\} \tag{8}$$

where ρ and σ are arbitrary spin operators, the symbol † implies Hermitian conjunction, and the trace is taken over both the electronic and nuclear spin quantum numbers, i.e.,

$$\begin{aligned}\langle\!\langle p_1^S, q_1^S; p_1^I, q_1^I | p_2^S, q_2^S; p_2^I, q_2^I\rangle\!\rangle &= \langle m_S | m_S''\rangle\langle m_I | m_I''\rangle\, \mathrm{tr}_S\{|m_S'\rangle\langle m_S'''|\}\, \mathrm{tr}_I\{|m_I'\rangle\langle m_I'''|\} \\ &= \delta_{p_1^S,p_2^S}\delta_{q_1^S,q_2^S}\delta_{p_1^I,p_2^I}\delta_{q_1^I,q_2^I}\end{aligned} \tag{9}$$

This is the conventional scalar product of two operators.

It is quite useful to give the following physical interpretation of the spin indices of the basis vectors. First, consider a g-tensor-only problem ($I = 0$). Since the p^S index labels the difference in electron spin projection numbers, a state with $p^S = 0$ corresponds to a diagonal spin density matrix element, i.e., it represents the population of the spin state with projection quantum number $m_S = q^S/2$. In the same manner, the states with $p^S = \pm 1$ represent off-diagonal density matrix elements which are connected to the diagonal density matrix elements by the microwave radiation field, i.e., they represent magnetic-dipole allowed electron spin transitions. The q_S index distinguishes between the (possibly degenerate) transitions with the same value of p^S. The interpretation is only slightly more complicated when the electron spin is coupled to a nuclear spin via the hyperfine interaction. Here, the $p^S = p^I = 0$ states correspond to diagonal density matrix elements, the $p^S = \pm 1$, $p^I = 0$ states correspond to allowed ESR transitions, and the states with $p^S = 0$ and $p^I = \pm 1$ represent allowed NMR transitions. The rest of the states represent forbidden transitions of various types. For example, the states with $p^S = \pm 1$ and $p^I = \pm 1$ represent singly forbidden ESR transitions, since both the electron and nuclear spin projection quantum numbers are different. Similarly, the states with $p^S = \pm 1$ and $p^I = \pm 2$ represent doubly forbidden ESR transitions. Analogous interpretations are possible for the remainder of the states in terms of forbidden ESR and/or NMR transitions. This interpretation is very useful to keep in mind when trying to understand the physical meaning of the sometimes overwhelming equations which arise in these problems.

The direct product of the spin and spatial parts gives the total basis vector

$$|L, M, K; p^S, q^S; p^I, q^I\rangle\rangle = |L, M, K) \otimes |p^S, q^S; p^I, q^I\rangle\rangle \tag{10}$$

which, by equations (5) and (9), must satisfy the following orthonormality conditions:

$$\langle\langle L_1, M_1, K_1; p_1^S, q_1^S; p_1^I, q_1^I | L_2, M_2, K_2; p_2^S, q_2^S; p_2^I, q_2^I\rangle\rangle$$
$$= \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\delta_{p_1^S,p_2^S}\delta_{q_1^S,q_2^S}\delta_{p_1^I,p_2^I}\delta_{q_1^I,q_2^I} \tag{11}$$

The basis set which consists of all basis vectors with $L \leq L_{\max}$ will be denoted $\{\mathscr{B}(L_{\max})\}$. This basis has been constructed in a manner which makes no use of any symmetries related to the stochastic Liouville superoperator or starting vector. However, it has the distinct advantage over the symmetry-adapted basis set constructed in Section 2.8 in that the matrix elements of the Liouville and diffusion superoperators are more convenient to calculate in this basis.

The large number of indices needed to completely specify the states makes the expressions for the matrix elements of the Liouville and diffusion superoperators rather unwieldy. To partially alleviate these difficulties, the shorthand notation

$$|\sigma_n\rangle\rangle = |L_n, M_n, K_n; p_n^S, q_n^S; p_n^I, q_n^I\rangle\rangle \tag{12}$$

for the basis vectors will often be used in the following.

A vector in the direct product space can be associated with an arbitrary orientation-dependent spin operator in an unique manner. This vector can be defined in terms of its components in the basis $\{\mathscr{B}(L_{\max})\}$,

$$\sigma \leftrightarrow |\sigma\rangle\rangle = \sum c_{L,M,K,p^S,q^S,p^I,q^I} |L, M, K, p^S, q^S, p^I, q^I\rangle\rangle \tag{13}$$

2.4. Construction of the Spin Hamiltonian

One of the most tedious aspects of calculating matrix elements of the Liouville and diffusion superoperators is keeping track of the coordinate frames in which the various quantities are defined. For instance, components of the magnetic tensors are time-independent in a body-fixed frame such as the magnetic or diffusion frames introduced above. In contrast, the static magnetic field, B_0, is only static in a reference frame which is fixed in space. The confusion is compounded by the fact that the spectrum is given in terms of the autocorrelation function of the component of the magnetization along the $\hat{x}_L$ axis, but the magnetic tensors which determine magnetization are most conveniently studied in a frame fixed in the molecule. To unravel these complications in a general manner it is extremely useful to introduce the concept of an irreducible spherical tensor operator (ISTO).

There are several alternative ways to define an ISTO (Biedenharn and Louck, 1981). Because of the present interest in the behavior of various quantities under rotation, it seems appropriate to use the definition that an operator T^J is an ISTO of rank J if it can be written as a sum of components which transform as

$$T_i^{(J,M)} \rightarrow T_f^{(J,M)} = \sum_{M'=-J}^{J} T_i^{(J,M')}\mathscr{D}_{M'M}^J(\Xi_{i\rightarrow f}) \tag{14}$$

where $\Xi_{i\rightarrow f}$ represent the set of Euler angles which take the initial frame into the final frame and the functions $\mathscr{D}_{M'M}^J(\Xi)$ are again the Wigner rotation matrix elements (Messiah, 1962; Biedenharn and Louck, 1981). These functions will play a central role in the following discussion. For this problem, only tensor operators of rank zero, one, and two are important. A list of the important tensorial quantities, their ranks, and the most "natural" type of frame of reference for defining them are given in Table 1.

We note here that the spin Hamiltonian is a rank-zero tensor. This must be so, since it is necessary that the energy of the spin system is independent of the coordinate system used to describe the orientation of the molecule in space and the quantization axes of the spin and magnetic field operators (Kivelson, 1972; Biedenharn and Louck, 1981). Since the Hamiltonian must be a scalar (i.e., a rank-zero tensor), it can be written as a contraction of tensors (cf. equation I-17),

$$H(\Omega) = \sum_{\mu,m,l} F_{\mu,\eta}^{(l,m)*}A_{\mu,\eta}^{(l,m)} = \sum_{\mu,m,l} (-1)^m F_{\mu,\eta}^{(l,-m)}A_{\mu,\eta}^{(l,m)} \tag{15}$$

TABLE 1
Important Tensorial Quantities

Tensor	Rank	Frame of definition
Spin Hamiltonian	0	
Electron spin	1	Space
Nuclear spin	1	Space
Magnetic field	1	Space
g tensor	2	Molecular
A tensor	2	Molecular

where the $F_{\mu,\eta}^{(l,m)}$ are proportional to the standard ISTO components of the magnetic tensor of type μ in the reference frame η. The quantities $A^{(l,m)}$ are the ISTO components of the tensors which arise from the coupling of spin and/or magnetic field operators. For reasons which will become clearer in Section 2.7, it is convenient to parameterize the orientation of the molecule by specifying the orientation of the diffusion frame (body-fixed) with respect to the director frame (space-fixed) by a set of Euler angles (Rose, 1957; Messiah, 1962; Biedenharn and Louck, 1981),

$$\Omega \equiv \Omega_{d \to R} = (\alpha, \beta, \gamma) \tag{16}$$

Since the simple product of two distinct ISTOs does not necessarily transform as an ISTO [cf. equation (14)], it is useful to introduce the decomposition of a product of two ISTOs into a new set of ISTOs. This coupling relies on well-known results, presented below, from the quantum theory of angular momentum (Messiah, 1962; Biedenharn and Louck, 1981). If $T^{j_1}(1)$ and $T^{j_2}(2)$ are two distinct ISTOs of rank j_1 and j_2, respectively, then the product $[T^{j_1}(1) \times T^{j_2}(2)]$ in general decomposes into a sum of ISTO operators, $T_m^j(1, 2)$.

$$T_m^j(1,2) \equiv [T^{j_1}(1) \times T^{j_2}(2)]_m^j = \sum_{m_1, m_2} \langle j_1, m_1, j_2, m_2 | j_1, j_2, j, m \rangle T_{m_1}^{j_1}(1) T_{m_2}^{j_2}(2) \tag{17}$$

where $\langle j_1, m_1, j_2, m_2 | j_1, j_2, j, m \rangle$ is a Clebsch–Gordon coefficient. This same scheme of coupling ISTOs can be used to form the $A_{\mu,\eta}^{(l,m)}$ in equation (15) and to construct a rotationally invariant quantity from the product of $F_{\mu,\eta}^{(l,m)*}$ and $A_{\mu,\eta}^{(l,m)}$. Actually, the spin Hamiltonian is usually defined as a sum of dot products of ISTOs rather than a contraction of ISTOs. These two definitions differ by a factor of $\sqrt{2l+1}$. The spin Hamiltonian as defined in equation (15) is consistent with the usual definition in terms of dot products.

To illustrate these concepts, consider the g-tensor contribution to the spin Hamiltonian, $H = \beta_e \mathbf{B}_0 \cdot \mathbf{g} \cdot \mathbf{S}$. The g tensor is a real symmetric Cartesian tensor which, by definition, is diagonal in the magnetic frame (cf. Section 2.2). As such, it must decompose into a sum of a rank-zero tensor (Nordio, 1976),

$$g^{(0,0)} = -\frac{1}{\sqrt{3}}(g_{xx} + g_{yy} + g_{zz}) \tag{18}$$

and the five components of a second-rank tensor, $g^{(2,m)}$,

$$g^{(2,\pm 2)} = \tfrac{1}{2}(g_{xx} - g_{yy}), \qquad g^{(2,\pm 1)} = 0, \qquad g^{(2,0)} = \sqrt{\tfrac{2}{3}}[g_{zz} - \tfrac{1}{2}(g_{zz} + g_{yy})] \tag{19}$$

The constant of proportionality between the $F_{g,m}^{(l,m)}$ and the $g^{(l,m)}$ is just the Bohr magneton, β_e, which would give the Hamiltonian the proper units of energy. In practice, however, it is desirable to convert the Hamiltonian into angular frequency units since a frequency-swept spectrum is desired.[3] Hence the $F_{g,m}^{(l,m)}$ are given by

$$F_{g,m}^{(0,0)} = -\frac{1}{\sqrt{3}}\left(\frac{\beta_e}{h}\right)(g_{xx} + g_{yy} + g_{zz}) \tag{20}$$

and

$$F_{g,m}^{(2,\pm 2)} = \frac{1}{2}\left(\frac{\beta_e}{h}\right)(g_{xx} - g_{yy}), \qquad F_{g,m}^{(2,\pm 1)} = 0, \qquad F_{g,m}^{(2,0)} = \sqrt{\frac{2}{3}}\left(\frac{\beta_e}{h}\right)[g_{zz} - \tfrac{1}{2}(g_{xx} + g_{yy})] \tag{21}$$

The next step is to construct the ISTO components of the tensor product of $\mathbf{B}_0$ and $\mathbf{S}$. This is most conveniently done in the laboratory frame, since $\mathbf{B}_0 \parallel \hat{z}_L$ and the autocorrelation function of the component of the magnetization along the $\hat{x}_L$ axis is (hopefully!) related to the observed experimental spectrum. In the laboratory frame, the coupling of the rank-one operators associated with $\mathbf{B}_0$ and $\mathbf{S}$ also gives rise to a zero- and a second-rank tensor,

$$A_{g,L}^{(0,0)} = -\left(\frac{1}{\sqrt{3}}\right) B_0 S_z \tag{22}$$

$$A_{g,L}^{(2,\pm 2)} = 0, \qquad A_{g,L}^{(2,\pm 1)} = \mp(\tfrac{1}{2}) B_0 S_\pm, \qquad A_{g,L}^{(2,0)} = (\sqrt{\tfrac{2}{3}}) B_0 S_z \tag{23}$$

The arguments for construction of the hyperfine portion of the spin Hamiltonian follow the same lines. The results are

$$A_{A,L}^{(0,0)} = -\left(\frac{1}{\sqrt{3}}\right)[S_z I_z + \tfrac{1}{2}(S_+ I_- + S_- I_+)] \tag{24}$$

$$A_{A,L}^{(2,\pm 2)} = \tfrac{1}{2} S_\pm I_\pm, \qquad A_{A,L}^{(2,\pm 1)} = \mp\tfrac{1}{2}(S_\pm I_z + S_z I_\pm), \qquad A_{A,L}^{(2,0)} = \sqrt{\tfrac{2}{3}}[S_z I_z - \tfrac{1}{4}(S_+ I_- + S_- I_+)] \tag{25}$$

and

$$F_{A,m}^{(0,0)} = -\frac{1}{\sqrt{3}}\left(\frac{g_e \beta_e}{h}\right)(A_{xx} + A_{yy} + A_{zz}) \tag{26}$$

$$F_{A,m}^{(2,\pm 2)} = \frac{1}{2}\left(\frac{g_e \beta_e}{h}\right)(A_{xx} - A_{yy}), \qquad F_{A,m}^{(2,\pm 1)} = 0,$$

$$F_{A,m}^{(2,0)} = \sqrt{\frac{2}{3}}\left(\frac{g_e \beta_e}{h}\right)[A_{zz} - \tfrac{1}{2}(A_{xx} + A_{yy})] \tag{27}$$

This completes the construction of the ISTO components of the $F_{\mu,m}$ and $A_{\mu,L}$ tensors.

Once the ISTO components $F_{\mu,m}^{(l,m)}$ and $A_{\mu,L}^{(l,m)}$ have been constructed, they must be transformed into a common coordinate frame, so that the spin Hamiltonian can be written in the form given in equation (15). To accomplish this, the transformation law used to define an ISTO can be used. Since the quantity of interest is the autocorrelation function of the projection of the magnetization along the $\hat{x}_L$ axis, it makes sense to leave the spin and magnetic field operators alone and transform the $F_{\mu,m}$ tensor into the laboratory frame. The complete transformation from the magnetic frame into the

[3] This is somewhat misleading in the sense that the Hamiltonian as used in the computation of the matrix elements of the Liouville superoperator is converted into units of gauss rather than angular frequency. This is done so that the resulting spectrum can easily be compared to experimental ESR spectra obtained in the field-swept mode.

laboratory frame can be broken up into three successive transformations which can be schematically represented as $m \to R \to d \to L$. This can be stated more precisely as

$$F^{(0,0)}_{\mu,L} = F^{(0,0)}_{\mu,m} \tag{28}$$

$$F^{(2,m)}_{\mu,L} = \sum_{m',m'',m'''=-2}^{2} F^{(2,m''')}_{\mu,m} \mathscr{D}^{2}_{m''m'''}(\Phi_{m\to R}) \mathscr{D}^{2}_{m'm''}(\Omega_{R\to d}) \mathscr{D}^{2}_{m'm}(\Psi_{d\to L}) \tag{29}$$

However, the rotation angles defined in Section 2.2 are for the opposite sense of rotation, $\Omega \equiv \Omega_{d\to R} \neq \Omega_{R\to d}$ [cf. equation (16)]. Put another way, in terms of the sense of rotations used to define the angles of rotation in Section 2.2, the components of the F_μ tensors have been constructed in the final (L) frame rather than the initial (d) frame (Nordio, 1976) given definitions of the coordinate systems and Euler angles. This confusion always arises in the problems involving both space- and body-fixed frames (Biedenharn and Louck, 1981).

One way to resolve this problem is to use the properties of the Wigner rotation matrix elements (Messiah, 1962; Biedenharn and Louck, 1981),

$$\mathscr{D}^{l}_{M'M}(\Xi_{f\to i}) = \mathscr{D}^{l*}_{MM'}(\Xi_{i\to f}) \tag{30}$$

which follows from the unitary nature of the rotation operator. Using equations (29) and (30), the components of the F_μ tensor in the laboratory frame can be written as

$$F^{(0,0)}_{\mu,L} = F^{(0,0)}_{\mu,m} \tag{31}$$

$$F^{(2,m)}_{\mu,L} = \sum_{m',m'',m'''=-2}^{2} \mathscr{D}^{2*}_{mm'}(\Psi_{L\to d}) \mathscr{D}^{2*}_{m'm''}(\Omega_{d\to R}) \mathscr{D}^{2*}_{m''m'''}(\Phi_{R\to m}) F^{(2,m''')}_{\mu,m} \tag{32}$$

where the senses of the rotations are now consistent with the definitions in Section 2.2, i.e., $\Psi_{L\to d} = (0, \psi, 0)$, $\Omega_{d\to R} = (\alpha, \beta, \gamma)$, and $\Phi_{R\to m} = (0, \phi, 0)$. Finally, it is actually the complex conjugate of the tensor F_μ which appears in equation (15), so

$$F^{(0,0)*}_{\mu,L} = F^{(0,0)*}_{\mu,m} \tag{33}$$

$$F^{(2,m)*}_{\mu,L} = \sum_{m',m'',m'''=-2}^{2} d^{2}_{mm'}(\psi) \mathscr{D}^{2}_{m'm''}(\Omega) d^{2}_{m''m'''}(\phi) F^{(2,m''')}_{\mu,m} \tag{34}$$

where use has been made of the fact that general Wigner rotation matrix elements take on a simpler form when the rotation depends only on the second of the three Euler angles (Messiah, 1962; Biedenharn and Louck, 1981) and that the elements of the $F_{\mu,m}$ tensors are real.

Now, putting the pieces together, the spin Hamiltonian is a sum of two parts, the sum of contractions of zero-rank tensors

$$H_0 = \sum_{\mu} F^{(0,0)}_{\mu,L} A^{(0,0)}_{\mu,L} \tag{35}$$

and a sum of contractions of second-rank tensors

$$H_1(\Omega) = \sum_{\mu,m',m'',m'''} d^{2}_{mm'}(\psi) \mathscr{D}^{2}_{m'm''}(\Omega) d^{2}_{m''m'''}(\phi) F^{(2,m''')}_{\mu,m} A^{(2,m)}_{\mu,L} \tag{36}$$

It is noteworthy that the only portion of the spin Hamiltonian which is dependent on the angles Ω arises from the contraction of second-rank tensors. The interested reader is encouraged to independently verify equation (36) by transforming the A_μ tensor from the laboratory frame into the magnetic frame. This construction is more straightforward, since the A_μ tensor transforms in the manner stated in equation (14) with the sense of rotations given in Section 2.2.

The construction given here is consistent with the results given by Nordio (1976) and Meirovitch *et al.* (1982). For more details on these matters, the reader is urged to consult the review chapters written by Kivelson (1972) and Nordio (1976) and the more comprehensive treatments given in standard texts (Rose, 1957; Messiah, 1962; Biedenharn and Louck, 1981). A different convention has been used in I [compare equation (14) and equation I-46], but the final results are equivalent.

2.5. Matrix Elements of the Liouville Superoperator

Using the definition of the scalar product in operator space [cf. equations (5), (9), and (11)] and the definition of the Liouville superoperator as the commutator with the spin Hamiltonian [cf. equation (15)],

$$\mathscr{L}|\sigma\rangle\!\rangle = |H\sigma - \sigma H\rangle\!\rangle \tag{37}$$

the matrix elements of the Liouville superoperator in the basis $\{\mathscr{B}(L_{\max})\}$ are

$$\begin{aligned}\langle\!\langle\sigma_1|\mathscr{L}|\sigma_2\rangle\!\rangle = &\sum_{\mu,l,m,m',m'',m'''} d^{l}_{mm'}(\psi) d^{l}_{m''m'''}(\phi) F^{(l,m''')}_{\mu,m} N_L(L_1, L_2)\\ &\times \left(\frac{1}{8\pi^2}\right) \int d\Omega\, \mathscr{D}^{L_1*}_{M_1K_1}(\Omega) \mathscr{D}^{l}_{m'm''}(\Omega) \mathscr{D}^{L_2}_{M_2K_2}(\Omega)\\ &\times \langle p^S_1, q^S_1; p^I_1, q^I_1|[A^{(l,m)}_{\mu,L}]^{\times}|p^S_2, q^S_2; p^I_2, q^I_2\rangle\end{aligned} \tag{38}$$

The integral over Ω can easily be expressed in terms of Wigner 3-J symbols as (Edmonds, 1957)

$$\left(\frac{1}{8\pi^2}\right) \int d\Omega\, \mathscr{D}^{L_1*}_{M_1K_1}(\Omega) \mathscr{D}^{l}_{m'm''}(\Omega) \mathscr{D}^{L_2}_{M_2K_2}(\Omega) = \begin{pmatrix} L_1 & l & L_2 \\ M_1 & m' & M_2 \end{pmatrix} \begin{pmatrix} L_1 & l & L_2 \\ K_1 & m'' & K_2 \end{pmatrix} \tag{39}$$

Since $l = 0$ or 2 in equations (38) and (39), the so-called "triangle" conditions on the 3-J symbols restrict the nonzero matrix elements of the Liouville superoperator to lie within a band about the diagonal determined by $|L_1 - L_2| \leq 2$.

To evaluate the spin part of the matrix elements, it is first necessary to examine the action of the spin commutator superoperators on the standard basis of spin projectors. We note that these basis vectors are eigenvectors of the $J^{\times}_z$ operator. This fact will be used later in the discussion of the symmetry-adapted basis set (cf. Section 2.8). The matrix elements of the $[A_{\mu,L}]^{\times}$ superoperators can also be evaluated in this manner using equations (22)–(25). The complete expressions for the matrix elements of the Liouville superoperator are quite tedious and will not be presented here (but see Meirovitch *et al.* 1982). The results presented so far are sufficient to give the reader a feeling for the symmetry arguments presented in Section 2.8.

2.6. Construction and Matrix Elements of the Diffusion Superoperator

The symmetrized diffusion operator, $\tilde{\Gamma}$, is assumed to be a sum of five independent terms corresponding to the processes:

1. $\tilde{\Gamma}_{\text{iso}}$: potential-independent part of rotational motion,
2. $\tilde{\Gamma}_{\text{ex}}$: Heisenberg spin exchange,
3. $\tilde{\Gamma}_U$: potential-dependent part of rotational motion,
4. $\tilde{\Gamma}_{\text{av}}$: effects of anisotropic viscosity,
5. $\tilde{\Gamma}_{\text{dj}}$: jumping between equivalent sites,

so that the total symmetrized diffusion superoperator can be written as

$$\tilde{\Gamma} = \tilde{\Gamma}_{\text{iso}} + \tilde{\Gamma}_{\text{ex}} + \tilde{\Gamma}_U + \tilde{\Gamma}_{\text{av}} + \tilde{\Gamma}_{\text{dj}} \tag{40}$$

The first two processes are independent of the existence of a restoring potential and are thus appropriate for modeling the motion of radicals in isotropic liquids. In contrast, the last three processes require the local environment of the radical to have lower than spherical symmetry. These contributions will be discussed separately below.

The general forms of some of the contributions to the diffusion superoperator are somewhat awkward. In these cases, the matrix elements of the corresponding operator will be presented rather than the operator itself.

2.6.1. Contributions to the Diffusion Superoperator in Isotropic Media

The matrix elements of $\tilde{\Gamma}_{\text{iso}}$ in the basis $\{\mathscr{B}(L_{\max})\}$ are (Freed, 1976; Moro, 1980a; Moro and Freed, 1981; Meirovitch *et al.*, 1982)

$$\begin{aligned}\langle\!\langle \sigma_1|\tilde{\Gamma}_{\text{iso}}|\sigma_2\rangle\!\rangle = {} & \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\delta_{p_1^s,p_2^s}\delta_{q_1^s,q_2^s}\delta_{p_1^I,p_2^I}\delta_{q_1^I,q_2^I} \\ & \times\left\{\frac{R_\perp L_1(L_1+1)}{[1+\tau_\perp R_\perp L_1(L_1+1)]^{E_\perp}}+\frac{R_\parallel K_1^2}{[1+\tau_\parallel R_\parallel K_1^2)]^{E_\parallel}}\right. \\ & \left.-\frac{R_\perp K_1^2}{[1+\tau'_\perp R_\perp L_1(L_1+1)]^{E'_\perp}}\right\}\end{aligned} \tag{41}$$

Here, $R_\parallel$ and $R_\perp$ are the parallel and perpendicular components of the rotational diffusion tensor, **R**. That is, $R_\parallel$ is related to the correlation time for the motion of the spin probe about the symmetry axis, $\hat{z}_R$, of the diffusion tensor, while $R_\perp$ is related to motion perpendicular to this symmetry axis. The parameters characterizing the three canonical rotational diffusion models are

- Brownian: $\tau_j = 0$, $E_j = 0$,
- free: $\tau_j \neq 0$, $E_j = \frac{1}{2}$,
- jump: $\tau_j \neq 0$, $E_j = 1$,

where $j = \parallel, \perp$ and the nonzero values of the τ_j parameters for jump and free diffusion are to be interpreted as mean residence times at each site, between jumps (cf. equations I-34 to I-44). The matrix of $\tilde{\Gamma}_{\text{iso}}$ is seen to be both real and diagonal in this basis.

The use of different models for the parallel and perpendicular rotational motions (with subscripts $\parallel$ and $\perp$) is justifiable only if there exists a clear time-scale separation characterizing these two processes (Mason *et al.*, 1974; Freed, 1976; Meirovitch and Freed, 1979; Campbell *et al.*, 1979; Meirovitch *et al.*, 1984) (e.g., when $R_\parallel$ is dominated by a relatively more rapid *internal* motion, see Campbell *et al.*, 1979).

The form of the Heisenberg spin exchange contribution to the diffusion superoperator is based on two approximations:

- The lifetime of the radical-pair interaction is short compared both to the effective exchange time and the rotational correlation times.
- The exchange rate does not depend on the relative orientation of the two members of the radical-pair (see also Zientara and Freed, 1979).

The validity of these approximations for the system under study should be carefully scrutinized before including this term in a calculation. The matrix elements of $\tilde{\Gamma}_{\text{ex}}$ in these limits are

$$\begin{aligned}\langle\!\langle \sigma_1|\tilde{\Gamma}_{\text{ex}}|\sigma_2\rangle\!\rangle = {} & \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\delta_{p_1^s,p_2^s}\delta_{p_1^I,p_2^I} \\ & \times\omega_{\text{HE}}[\delta_{q_1^s,q_2^s}\delta_{q_1^I,q_2^I}-\tfrac{1}{2}\delta_{p_1^s,0}\delta_{q_1^I,q_2^I}-(2I+1)^{-1}\delta_{p_1^I,0}\delta_{q_1^s,q_2^s}]\end{aligned} \tag{42}$$

2.6.2. Contributions to the Diffusion Superoperator in Anisotropic Media

The Fokker-Planck equation for the stochastic motion of a particle in an external potential is conveniently soluble only for the Brownian model, so attention will be restricted to this special case. The symmetrized diffusion (Smoluchowski) superoperator in this case is

$$\tilde{\Gamma}_O = \tilde{\Gamma}_{\text{iso}} + \tilde{\Gamma}_U = [\hat{\mathcal{L}} - (\hat{\mathcal{L}}U)/2k_BT]\mathbf{R}[\hat{\mathcal{L}} + (\hat{\mathcal{L}}U)/2k_BT] \tag{43}$$

where $\hat{\mathcal{L}}$ is the generator of infinitesimal rotations of the diffusion frame relative to the director frame, **R** is the rotational diffusion tensor, and $U \equiv U(\Omega)$ is given in equation (2). A simple decomposition of $\tilde{\Gamma}_U$ exists due to the particular form of the restoring potential. This becomes more obvious if equation (43) is cast into the form

$$\begin{aligned}\tilde{\Gamma}_O = {} & R_\perp\hat{\mathcal{L}}^2 + (R_\parallel - R_\perp)\hat{\mathcal{L}}_z^2 + \left(\frac{1}{2k_BT}\right)[R_\perp(\hat{\mathcal{L}}U)^2 + (R_\parallel - R_\perp)(\hat{\mathcal{L}}_zU)^2] \\ & - \left(\frac{1}{2k_BT}\right)^2[R_\perp(\hat{\mathcal{L}}_+U)(\hat{\mathcal{L}}_-U) + R_\parallel(\hat{\mathcal{L}}_zU)^2]\end{aligned} \tag{44}$$

The matrix elements of the sum of the first two terms in equation (44) coincide with those of the Brownian diffusion superoperator in isotropic media (cf. Section 2.6.1). We note that in equation (44), the action of each of the operators $\hat{\mathcal{L}}^2$, $\hat{\mathcal{L}}_z^2$, and $\hat{\mathcal{L}}_\pm$ on members of the basis set $\{\mathscr{B}(L_{\max})\}$ is

$$\hat{\mathcal{L}}^2|L, M, K; p^S, q^S; p^I, q^I\rangle\!\rangle = L(L+1)|L, M, K; p^S, q^S; p^I, q^I\rangle\!\rangle \tag{45}$$

$$\hat{\mathcal{L}}_z|L, M, K; p^S, q^S; p^I, q^I\rangle\!\rangle = K|L, M, K; p^S, q^S; p^I, q^I\rangle\!\rangle \tag{46}$$

$$\hat{\mathcal{L}}_\pm|L, M, K; p^S, q^S; p^I, q^I\rangle\!\rangle = \sqrt{L(L+1) - K(K\pm 1)}\,|L, M, K\pm 1; p^S, q^S; p^I, q^I\rangle\!\rangle \tag{47}$$

If these three equations are used to further simplify the form of equation (44), we obtain

$$\tilde{\Gamma}_U = \sum_{L=2}^{8} \sum_{K=-L}^{L} X^L_K \mathcal{D}^L_{0K}(\Omega) \tag{48}$$

The X^L_K parameters in equation (48) for $L \neq 0$ are numerical coefficients which depend on the potential parameters (Meirovitch *et al.*, 1982).[4] These coefficients satisfy

$$X^L_K = X^L_{-K} \tag{49}$$

and $X^L_K = 0$ if either L or K is odd.

The matrix elements of $\tilde{\Gamma}_U$ can now be written as

$$\begin{aligned}\langle\!\langle \sigma_1|\tilde{\Gamma}_U|\sigma_2\rangle\!\rangle = {} & \delta_{M_1,M_2}\delta_{p_1^S,p_2^S}\delta_{q_1^S,q_2^S}\delta_{p_1^I,p_2^I}\delta_{q_1^I,q_2^I} \\ & \times \sum_{L=2}^{8} (-1)^{M_1-K_1} X^L_{K_1-K_2} N_L(L_1, L_2) \\ & \times \begin{pmatrix} L_1 & L & L_2 \\ M_1 & 0 & -M_1 \end{pmatrix} \begin{pmatrix} L_1 & L & L_2 \\ K_1 & K_2-K_1 & -K_2 \end{pmatrix}\end{aligned} \tag{50}$$

The matrix of $\tilde{\Gamma}_U$ is seen to be real and symmetric in this basis, but is not diagonal.

Before proceeding, it is useful to make a few remarks about the implications of the form of $\tilde{\Gamma}_U$. Since the restoring potential is a linear combination of the Wigner rotation matrix elements for which $M = 0$, it is clear that the restoring potential is independent of the Euler angle α. Therefore, the restoring potential is axially symmetric in the director frame, hence the term is uniaxial. Indeed, this requirement is used to define the director frame in uniaxial liquid crystals. The main point, however, is that this symmetry implies that all spin probe molecules whose orientations in space are related by a simple rotation about the $\hat{z}_d$ axis experience similar forces and thus will reorient in a similar manner. In contrast, the restoring potential can depend on the Euler angle γ if the $K = 2$ terms are included in the expansion. If the potential does depend on γ, then spin probes whose orientations in space are related by a nontrivial rotation about the $\hat{z}_R$ axis at a given time will experience different forces and thus will reorient dissimilarly, so long as they continue to experience different forces. However, it has been assumed for convenience that the diffusion tensor, which is related to the geometry of the spin probe molecules, is axially symmetric about $\hat{z}_R$. This symmetry of the diffusion tensor implies that the reorientational dynamics of spin probes related by rotations about $\hat{z}_R$ should be similar, not dissimilar as suggested by the preceding argument. The resolution of this apparent conflict is simply that the axial symmetry of the spin probe is usually only approximate. In essence, the effect of the diffusional terms which depend only on the deviation of the diffusion tensor from axial symmetry are usually quite small and have been neglected. However, even if the assumption of an axially symmetric diffusion tensor for a given radical is valid for studies involving isotropic media, the terms which involve the effect of the asymmetric portion of the restoring potential on the reorientational dynamics can still be quite large for the same radical in liquid crystalline media. Therefore, these terms may be included in the calculation if necessary. In summary, the effects of molecular asymmetry are usually more pronounced in asymmetric environments than in symmetric environments.

[4] A factor of 1/4 is missing in front of the double summation in equation A23 in this reference.

It is also possible to have discrete-jump-type motions between equivalent sites. This type of motion can be incorporated into the model for the reorientational dynamics of the probe molecules by the inclusion of a term of the form

$$\langle\!\langle \sigma_1|\tilde{\Gamma}_{dj}|\sigma_2\rangle\!\rangle = \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\delta_{p_1^S p_2^S}\delta_{q_1^S q_2^S}\delta_{p_1^I p_2^I}\delta_{q_1^I q_2^I}\tau_{dj}^{-1}[1 - \delta_{0,(K_1 \bmod n_s)}] \tag{51}$$

where n_s is the number of equivalent sites and τ_{dj}^{-1} is the discrete jump frequency.

In the presence of an orienting potential, the viscosity can be a tensorial quantity. This effect can also be included by adding a term $\tilde{\Gamma}_{av}$. The correction term to $\tilde{\Gamma}_{iso}$ can be written as

$$\begin{aligned}\langle\!\langle \sigma_1|\tilde{\Gamma}_{av}|\sigma_2\rangle\!\rangle_{iso} = {} & \delta_{L_1,L_2}\delta_{M_1,M_2}\delta_{K_1,K_2}\delta_{p_1^S,p_2^S}\delta_{q_1^S,q_2^S}\delta_{p_1^I,p_2^I}\delta_{q_1^I,q_2^I} \\ & \times [\hat{R}_\perp L_1(L_1+1) + (\hat{R}_\parallel - \hat{R}_\perp)M_1^2]\end{aligned} \tag{52}$$

where $\hat{R}_\parallel$ is the parallel diffusion coefficient in the director frame. The correction terms to $\tilde{\Gamma}_U$ can be found in Polnaszek and Freed (1975).

2.7. Components of the Starting Vector

Since a symmetrized diffusion operator is being used, and the spectral function associated with the $\hat{x}_L$ component of the electronic magnetization is desired, the appropriate starting vector is

$$\begin{aligned}|v\rangle\!\rangle & = (2I+1)^{-1/2}|S_x \otimes I_I \otimes P_0^{1/2}\rangle\!\rangle \\ & = (1/\sqrt{2})[|v_1\rangle\!\rangle + |v_{-1}\rangle\!\rangle]\end{aligned} \tag{53}$$

where $P_0(\Omega)$ is the equilibrium probability distribution for the spin probes and is given by

$$P_0(\Omega) = \frac{\exp[-U(\Omega)/k_B T]}{\int d\Omega \exp[-U(\Omega)/k_B T]} \tag{54}$$

In addition, I_I is the unit operator in the nuclear spin space, and

$$|v_{\pm1}\rangle\!\rangle = (2I+1)^{-1/2}|S_\pm \otimes I_I \otimes P_0^{1/2}\rangle\!\rangle \tag{55}$$

In the basis $\{\mathcal{B}(L_{max})\}$ [cf. equations (10) and (11)] the elements of the vectors $|v_{\pm1}\rangle\!\rangle$ making up the starting vector are

$$\begin{aligned}\langle L, M, K; p^S, q^S; p^I, q^I|v_{\pm1}\rangle = {} & \left[\frac{2L+1}{8\pi^2(2I+1)}\right]^{1/2}\left[\int d\Omega \exp[-U(\Omega)/k_B T]\right]^{-1/2} \\ & \times \langle p^S, q^S; p^I, q^I|S_\pm \otimes I_I\rangle\!\rangle \\ & \times \int d\Omega\, \mathcal{D}^{L*}_{MK}(\Omega) \exp[-U(\Omega)/2k_B T]\end{aligned} \tag{56}$$

Some simple symmetry arguments can greatly simplify this expression. First, the restoring potential has only $M = 0$ terms in its expansion [cf. equation (2)] and thus does not depend on the first Euler angle. This implies that only the basis vectors with $M = 0$ can have a nonzero projection on the starting vector. Second, all terms in the expansion of the restoring potential have even K quantum numbers, therefore only basis vectors with even K quantum numbers can have nonzero projections on the starting vector. Finally, since the Wigner rotation matrix elements with $M = 0$ are proportional to the usual spherical harmonics (Messiah, 1962; Biedenharn and Louck, 1981) and only the states with even values of K are important, one can further verify that only states with even L quantum numbers can have nonzero projections on the starting vector using a simple parity argument. Using these three observations and evaluating the spin part of equation (56) gives

$$\langle L, M, K; p^S, q^S; p^I, q^I \mid v_{\pm 1}\rangle = \delta_{p^S,\pm 1}\delta_{q^S,0}\delta_{p^I,0}\delta_{M,0}\delta_{(K \bmod 2),0} N_I \left[\frac{(2L+1)(L-K)!}{(L+K)!}\right]^{1/2}$$

$$\times \int_{-1}^{1} dz\, P_L^K(z) \exp[A(z)] \int_0^{2\pi} d\gamma \exp[iK\gamma + \cos(2\gamma) B(z)] \quad (57)$$

where $z = \cos\beta$, $P_L^K(z)$ is an associated Legendre function of the first kind, and

$$N_I = \left[8\pi^2(2I+1)\int d\Omega \exp[-U(\Omega)/k_B T]\right]^{-1/2} \quad (58)$$

$$A(z) = (\lambda_0^2/2)P_2^0(z) + (\lambda_0^4/2)P_4^2(z) \quad (59)$$

$$B(z) = (\lambda_2^2/2\sqrt{6})P_2^2(z) + (\lambda_2^4/6\sqrt{10})P_4^2(z) \quad (60)$$

The normalization factor N_I and similar factors which are independent of the basis-set quantum numbers will be ignored in the following. In the calculation of the starting vector, the unnormalized vector resulting from the neglect of trivial normalization factors is normalized numerically.

The integral over the angle γ can be expressed in terms of the modified Bessel functions of the first kind $I_n(y)$ (McLachlan, 1961) of strictly real argument as

$$\int_0^{2\pi} d\gamma \cos(K\gamma) \exp[\cos(2\gamma)B(z)] = 2\pi(2 - \delta_{K,0}) I_{K/2}(B(z)) \quad (61)$$

It is also possible to simply evaluate the double integral in equation (57) numerically, though this method becomes unstable for large L and/or K where the integrand is highly oscillatory.

2.8. The High-Field Approximation

Considerable simplifications in the above discussion are possible if one is able to take advantage of the approximations valid in the unsaturated, high-field, and slow motional limits. More precisely, if the follow statements are true:

- The anisotropic part of the spin Hamiltonian (the part arising from the contraction of second-rank tensors) is small compared to the isotropic part of the Zeeman interaction.
- The rotational motion is slow enough that $\omega_0\tau_R \gg 1$, where τ_R is the rotational correlation time on the order of unity.
- The microwave radiation field is sufficiently weak that the spin system is not being saturated (i.e., the absorption mode signal is directly proportional to the incident microwave field).

then one can neglect all terms in the spin Hamiltonian which do not commute with S_z (i.e., the nonsecular terms) and also make the rotating wave approximation. When these approximations are valid, one can neglect the coupling of the basis vectors with different values of the p^S quantum number. Therefore, the subspaces labeled by different values of the p^S index evolve independently, and one can calculate the spectrum by knowing only the time evolution of the states with $p^S = 1$, since these are the ones which couple to the microwave radiation field in the rotating wave approximation.

With these simplifications in mind, it is possible to reformulate the problem by neglecting the nonsecular terms in the spin Hamiltonian and developing a new basis set which takes advantage of the fact that only the $p^S = 1$, $q^S = 0$ states need be considered, since these are the only states connected to the starting vector by the simplified stochastic Liouville superoperator. A careful analysis of this stochastic Liouville superoperator shows that the matrix of the stochastic Liouville superoperator is reduced to a complex symmetric form in the subspace spanned by the basis vectors (Meirovitch *et al.*, 1982; Schneider, 1989),

$$|L, M, K'; 1, 0; p^I, q^I\rangle\rangle = [2(1 + \delta_{K,0})]^{-1/2}$$

$$\times (|L, M, K; 1, 0; p^I, q^I\rangle\rangle + (-1)^{L+K}|L, M, -K; 1, 0; p^I, q^I\rangle\rangle) \quad (62)$$

where the K' index is now constrained to be positive. Furthermore, the starting vector can have nonzero projections only on vectors within this subspace.

A further reduction in the size of the basis is possible by taking advantage of the symmetry of the matrix elements of the Liouville superoperator under the simultaneous reversal in sign of the M and p^I indices. The result of this analysis is that it is possible to further restrict one's attention to the subspace spanned by the basis vectors

$$|L, M', K'; 1, 0; p^I, q^I\rangle\rangle = [2(1 + \delta_{M,0}\delta_{p^I,0})]^{-1/2}$$

$$\times (|L, M, K'; 1, 0; p^I, q^I\rangle\rangle$$

$$+ (-1)^{L+M}|L, -M, K'; 1, 0; -p^I, q^I\rangle\rangle) \quad (63)$$

and the new M' index is also positive. This is true only if the nonsecular terms in the spin Hamiltonian are omitted. Although the expressions for the matrix elements of the approximate high-field stochastic Liouville superoperator are somewhat awkward in this basis, it does have the very significant advantage of reducing the dimension of the resulting matrix to a minimal size (Meirovitch *et al.*, 1982).

By symmetry arguments, it can also be shown that in special cases only small subsets of these basis vectors are needed (Meirovitch *et al.*, 1982). For instance, in the absence of director tilt, only the states with $M' = p'$ are needed. Similarly, if there is no diffusion tilt, then $F^{2,\pm 1}_{\mu,R} = 0$, so that only basis vectors with even values of K' are important. Finally, if the magnetic tensors are axially symmetric in the diffusion frame so that $F^{2,m}_{\mu,R} = \delta_{m,0}$, then only the states with even values of L and $K' = 0$ are needed.

The programs in this package use all of these approximations and take advantage of the resulting reduction in the size of the basis set which must be considered. In the following, this high-field basis set will be used exclusively, therefore the primes on the M and K subscripts will be dropped to simplify the notation.

3. MAGNETIC RESONANCE LINE SHAPES AND THE COMPLEX SYMMETRIC LANCZOS ALGORITHM

In 1950, the Hungarian physicist and mathematician Cornelius Lanczos developed a very powerful algorithm for tridiagonalizing arbitrary square matrices (Lanczos, 1950), which now carries his name. The algorithm attracted quite a bit of interest shortly after its development, but fell into disrepute when it was found that it was prone to certain instabilities in practical numerical applications (Wilkinson, 1965). Interest in the Lanczos algorithm was renewed when the source of these instabilities was uncovered by Paige (Paige, 1976, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985).

The basic scheme behind the Lanczos algorithm is quite simple. For simplicity, the attention will be restricted here to real, symmetric, positive definite (RSPD) matrices with nondegenerate eigenvalues. Given an $N \times N$ RSPD matrix $\mathbf{A}$ and an arbitrary nonzero, real N-vector $\mathbf{v}$, we form the sequence of vectors $\mathbf{k}_j$, $j = 1, 2, \ldots, N$ using the formula

$$\mathbf{k}_n = \mathbf{A}^{(n-1)}\mathbf{v} \tag{64}$$

The elements of the set of vectors $\{\mathbf{k}_n\}$ generated in this manner are referred to as Krylov vectors. The application of the Gram-Schmidt orthonormalization procedure to the sequence of Krylov vectors, in order of appearance, gives a new orthornomal basis, denoted $\{\mathbf{q}_j\}$. Let $\mathbf{Q}$ be the $N \times N$ matrix composed of the vectors $\mathbf{q}_j$. In exact arithmetic, the matrix $\mathbf{T} = \mathbf{Q}\mathbf{A}\mathbf{Q}^{\mathrm{tr}}$ is tridiagonal and has the same eigenvalues as the original matrix $\mathbf{A}$.

In practice, the Lanczos algorithm does not actually generate the vectors $\mathbf{k}_n$. Instead, it forms the vectors $\mathbf{q}_j$, which will be referred to as Lanczos vectors, in a direct manner using a stepwise orthogonalization procedure that will be discussed in detail in Section 3.1. This procedure has the tremendous advantage that it requires only two, not N, intermediate vectors of dimension N. The instabilities observed in the computer implementation of the Lanczos algorithm are due to the fact that in finite precision

arithmetic the overall, or global, orthogonality of the sequence of Lanczos vectors generated by the simple stepwise orthogonalization procedure cannot be maintained.

Several methods, all essentially based on the results of Paige's error analysis, have been proposed to circumvent this difficulty in order to enable one to reliably determine the entire set of eigenvalues of large, sparse matrices (Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985). One set of methods is based on trying to maintain global orthogonality in an efficient manner (Parlett, 1980; Golub and Van Loan, 1983). The computer implementation of these methods has the drawback of requiring relatively large amounts of memory to hold intermediate vectors and the associated CPU time to perform the reorthogonalization steps. Alternatively, one can simply allow the instabilities to arise and run their course, giving rise to spurious or "ghost" eigenvalues. These spurious eigenvalues can be identified after the fact and discarded (Cullum and Willoughby, 1985). This method also requires the storage or recalculation of large numbers of intermediate results. It also may require the execution of far more than N Lanczos steps on an $N \times N$ matrix.

The loss of global orthogonality is a severe problem if one tries to use the Lanczos algorithm to calculate the entire eigenvalue spectrum of large matrices. Fortunately, it poses little or no problem if one uses the Lanczos algorithm to calculate spectral functions in the manner discussed below. These assertions will be justified in the following sections.

Not long after Lanczos made his contribution, Hestenes and Stiefel (1952) developed a related algorithm to solve linear systems of algebraic equations known as the conjugate gradients algorithm. The relationship between these two algorithms is very close, but not necessarily obvious at first glance. Given an $N \times N$ RSPD matrix $\mathbf{A}$ and a real N-vector $\mathbf{v}$, the conjugate gradients algorithm attempts to solve the system of linear algebraic equation $\mathbf{A}\mathbf{x} = \mathbf{v}$ for the unknown vector $\mathbf{x}$. The conjugate gradients algorithm can be thought of as a Lanczos algorithm where the tridiagonal matrix $\mathbf{T}$ is constructed in a factored form (Golub and Van Loan, 1983; Cullum and Willoughby, 1985). This factored form of $\mathbf{T}$ can then be used to generate a sequence of approximate solution vectors $\mathbf{x}_j$ where $\mathbf{x}_0 = \mathbf{v}$. In exact arithmetic, this sequence terminates after at most N steps giving the exact solution vector.

As with the Lanczos algorithm, the conjugate gradients method is prone to the loss of orthogonality of the sequence of Lanczos vectors. This is manifested in the fact that the computer implementations of the conjugate gradients algorithm do not converge to the exact solution vector in at most N steps.

Again, this loss of orthogonality does not pose serious difficulties in the use of the conjugate gradients algorithm for the calculation of spectral functions. In addition, it can be used as a very effective means of determining a minimal truncation scheme (MTS) for a given problem (cf. Section 3.4).

3.1. The Real Symmetric Lanczos Algorithm

A short derivation of the Lanczos algorithm for real symmetric matrices will be given here as a reference for readers who are unfamiliar with the algorithm. The derivation given here is based on simple linear algebra and is slanted toward the generalizations needed for complex symmetric matrices. First, the Gram-Schmidt orthonormalization procedure will be used to derive the general three-term recursion formula for the Lanczos vectors. These results will then be collected into a single matrix equation for the Lanczos tridiagonal matrix. The majority of this section is devoted to the study of the algorithm in infinite precision arithmetic. The impact of finite precision computer arithmetic on the algorithm is briefly summarized at the end.

As stated previously, the Lanczos algorithm amounts to the application of the Gram-Schmidt orthogonalization procedure to a sequence of vectors defined by the matrix in question and a starting vector. This starting vector is determined by the form of the desired spectral function.

To start the process of generating the set of Lanczos vectors, assume A is an $N \times N$ real symmetric matrix and that v is a given real N-vector. The vector v will serve as the starting vector. The first member in the orthonormal set of Lanczos vectors, $\mathbf{q}_1$, is taken to be parallel to the starting vector v, i.e.,

$$\beta_0 \mathbf{q}_1 = \mathbf{k}_1 = \mathbf{v} \tag{65}$$

The requirement that the new set of vectors be normalized ($\mathbf{q}_i^{tr}\mathbf{q}_i = 1$) implies $\beta_0 = \|\mathbf{v}\|$.

In the following steps, the vector $\mathbf{q}_{j+1}$ being added to the existing set of Lanczos vectors is written as a linear combination of the previous vectors and the vector $\mathbf{A}\mathbf{q}_j$, i.e.,

$$\mathbf{q}_{j+1} = c_{j+1}^{(j+1)}\mathbf{A}\mathbf{q}_j + \sum_{k=1}^{j} c_k^{(j+1)}\mathbf{q}_k \tag{66}$$

The expansion coefficients, $c_k^{(j+1)}$, are determined by requiring that $\mathbf{q}_{j+1}$ be normalized and orthogonal to all previous Lanczos vectors. The relationship between the sets of Krylov and Lanczos vectors becomes evident if the relations defining the previous Lanczos vectors are inserted into equation (66) and terms of the form $\mathbf{A}^k\mathbf{v}$ are collected. The advantage of using the Lanczos vectors rather than the Krylov vectors is demonstrated below.

Now, one can proceed with the construction of the second Lanczos vector using equation (66) in the form

$$\beta_1 \mathbf{q}_2 = (\mathbf{A} - \alpha_1 \mathbf{I})\mathbf{q}_1 \tag{67}$$

To determine the coefficient α_1, equation (67) is multiplied on the left by $\mathbf{q}_1^{tr}$. If $\mathbf{q}_2$ is to be orthogonal to $\mathbf{q}_1$, it is clear that

$$\alpha_1 = \mathbf{q}_1^{tr}\mathbf{A}\mathbf{q}_1 \tag{68}$$

In contrast, the coefficient β_1 is chosen to normalize the second Lanczos vector, $\mathbf{q}_2$, giving

$$\beta_1 = \|(\mathbf{A} - \alpha_1 \mathbf{I})\mathbf{q}_1\| \tag{69}$$

In a similar fashion, the third vector is

$$\beta_2 \mathbf{q}_3 = (\mathbf{A} - \alpha_2 \mathbf{I})\mathbf{q}_2 - \gamma_1 \mathbf{q}_1 \tag{70}$$

Again, γ_1 and α_2 are determined from the orthogonality requirements,

$$\gamma_1 = \mathbf{q}_1^{tr}\mathbf{A}\mathbf{q}_2 = \beta_1 \tag{71}$$

$$\alpha_2 = \mathbf{q}_2^{tr}\mathbf{A}\mathbf{q}_2 \tag{72}$$

These coefficients can now be used in equation (70) to give the general form of the three-term recursion relation on which the Lanczos algorithm is based,

$$\beta_2 \mathbf{q}_3 = (\mathbf{A} - \alpha_2 \mathbf{I})\mathbf{q}_2 - \beta_1 \mathbf{q}_1 \tag{73}$$

Again, β_2 is chosen such that $\mathbf{q}_3$ is normalized, as done previously.

The explicit calculation of the fourth vector demonstrates the validity of the general form of the Lanczos recursion relation given in equation (73). In general, $\mathbf{q}_4$ must be expressible as a linear combination of $\mathbf{A}\mathbf{q}_3$, $\mathbf{q}_3$, $\mathbf{q}_2$, and $\mathbf{q}_1$,

$$\beta_3 \mathbf{q}_4 = (\mathbf{A} - \alpha_3 \mathbf{I})\mathbf{q}_3 - \gamma_2 \mathbf{q}_2 - \delta_1 \mathbf{q}_1 \tag{74}$$

To verify that equation (73) is indeed a prototype of a three-term recursion relation, it must be shown that $\delta_1 = 0$ and $\gamma_2 = \beta_2$. It is easy to see that $\delta_1 = 0$ by premultiplying equation (74) by $\mathbf{q}_1^{tr}$,

$$\delta_1 = \mathbf{q}_1^{tr}\mathbf{A}\mathbf{q}_3 \tag{75}$$

Using equation (67) this can be rewritten as

$$\delta_1 = (\beta_1 \mathbf{q}_2 + \alpha_1 \mathbf{q}_1)^{tr}\mathbf{q}_3 \tag{76}$$

which vanishes by the orthogonality of the Lanczos vectors. In addition, premultiplying equations (73) and (74) by $\mathbf{q}_3^{tr}$ gives $\beta_2 = \gamma_2$.

The same behavior is observed for all further vectors. Therefore, the general Lanczos recurrence relation

$$\beta_m \mathbf{q}_{m+1} = (\mathbf{A} - \alpha_m \mathbf{I})\mathbf{q}_m - \beta_{m-1}\mathbf{q}_{m-1} \tag{77}$$

can be used to generate all successive vectors q so long as the quantities β are nonzero. If $\beta_m = 0$ for $m < N$ then the procedure terminates. This occurrence is a manifestation of the fact that the starting vector does not have a projection on all eigenvectors of A. In this case, the eigenvalues of the matrix $\mathbf{T}_m$ correspond to the subset of the eigenvalues of A whose associated eigenvectors have a nonzero projection on the starting vector (Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985).

We note that, from equation (77), it is clear that only the last two Lanczos vectors are required for the construction of the next member of the set. This surprising fact is one of the major reasons why the Lanczos algorithm is such an efficient means of computing spectral functions. The other major reason is that the matrix A is not modified, so full advantage can be taken of any special structure of the matrix elements of A to simplify the computation (e.g., sparsity or bandedness).

It is informative to summarize these results in a matrix form. The transformation matrix, $\mathbf{Q}_N$, whose jth row is given by the elements of $\mathbf{q}_j$ for $1 \le j \le N$, transforms the original $N \times N$ matrix A into a $N \times N$ real symmetric tridiagonal matrix, $\mathbf{T}_N$,

$$\mathbf{T}_N = \mathbf{Q}_N \mathbf{A} \mathbf{Q}_N^{tr} \tag{78}$$

Using the Lanczos recursion relation derived above [see equation (77)], it is straightforward to verify that the matrix $\mathbf{T}_N$ is indeed symmetric and tridiagonal. The diagonal and off-diagonal matrix elements of $\mathbf{T}_N$ are just the coefficients appearing in equation (77),

$$(T_N)_{i,i} = \alpha_i \tag{79}$$

$$(T_N)_{i,i+1} = \beta_i \tag{80}$$

It can be shown that the matrix $\mathbf{T}_N$ can have only distinct eigenvalues, and that if $\beta_{m+1} = 0$ then the eigenvalues of $\mathbf{T}_m$ are identical to those eigenvalues of A whose corresponding eigenvectors have a nonzero projection on the starting vector [30, 31, 32] (Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985).

In addition, it is easy to verify that for each $m \leq N$,

$$\mathbf{Q}_m \mathbf{A} \mathbf{Q}_m^{tr} = \mathbf{T}_m \tag{81}$$

where $\mathbf{Q}_m$ is the $m \times N$ matrix constructed from the first m Lanczos vectors, and $\mathbf{T}_m$ is identical to the $m \times m$ submatrix in the upper left-hand corner of $\mathbf{T}_N$. Furthermore, from the orthonormality of the Lanczos vectors, it follows that

$$\mathbf{Q}\mathbf{Q}^{tr} = \mathbf{I}_m \tag{82}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix.

The effect of the finite precision arithmetic on the Lanczos algorithm is the subject of Paige's analysis (Paige, 1976, 1980; Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985). The basic result of this analysis is quite simple, though the analysis itself is rather sophisticated and will not be presented here. The finite precision error analysis is based on the fact that as the matrices $\mathbf{T}_m$ in equation (81) increase in size as m is increased, the eigenvalues and eigenvectors of $\mathbf{T}_m$ become better and better approximations to the eigenvalues and eigenvectors of A, but the convergence rate is not the same for all eigenvalue-eigenvector pairs. Assume for a moment that at step $m = M$ one of the eigenvalues of $\mathbf{T}_M$ is identical to one of the eigenvalues of the original matrix A to the working precision of the computer. The basic result of the finite precision error analysis is that all further Lanczos vectors $\mathbf{q}_m$ for $m > M$ tend to have a spurious projection along the eigenvector associated with the numerically converged eigenvalue. This phenomenon leads to a lack of orthogonality of these Lanczos vectors and the appearance of spurious eigenvalues.

However, this loss of orthogonality does not adversely affect spectral calculations performed in the manner advocated here. This subject is treated in Section 3.5.

3.2. The Complex Symmetric Lanczos Algorithm

The formal extension of the real symmetric Lanczos algorithm to handle non-Hermitian, complex symmetric matrices of the type which occur in magnetic resonance line-shape problems is relatively simple. In his original paper Lanczos (1950) showed

how certain classes of complex nonsymmetric matrices could be reduced to a complex tridiagonal form using a pair of closely related three-term recurrence relations. A starting vector is also needed for each recurrence relation. These relations are used to generate a biorthogonal set of vectors (Householder, 1964; Wilkinson, 1965; Golub and Van Loan, 1983; Cullum and Willoughby, 1985). The matrices constructed from these sets of biorthogonal vectors then define a general similarity transformation which reduces the matrix to tridiagonal form. These very general Lanczos recursion relations are sometimes used in the study of physical problems (Wassam, 1985a, 1985b).

However, if one is interested in studying complex symmetric matrices, a more definitive statement can be made. A classical result in linear algebra states that if two complex symmetric matrices are similar, then they are related to one another by a complex orthogonal transformation (Gantmacher, 1959; Horn and Johnson, 1985). Thus, if an $N \times N$ complex symmetric matrix A can be reduced to a complex symmetric tridiagonal matrix $\mathbf{T}_N$ by the Lanczos algorithm, the transformation matrix must be complex and orthogonal. A reexamination of the biorthogonal Lanczos relations for nonsymmetric matrices in this light shows that the pair of recursion relations degenerates to a single three-term recursion relation identical in form to equation (77) if the two starting vectors are chosen properly (Moro and Freed, 1986; Vasavada *et al.*, 1987). In the original Moro and Freed work (1981) the complex-symmetric form was introduced at the outset.

Though the form of the Lanczos recursion relation is identical for real symmetric and complex symmetric matrices, the elements of the resulting tridiagonal matrix and Lanczos vectors for the complex symmetric case require more explanation. In particular, the fact that the Lanczos vectors form a complex orthogonal matrix in the original basis means that they are not orthonormal in the usual sense. The canonical scalar product of two vectors x and y in an N-dimensional complex vector space is

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\dagger}\mathbf{y} = \sum_{i=1}^{N} \xi_i^* \eta_i \tag{83}$$

A complete orthonormal set of basis vectors $\{\hat{e}_j, j = 1, 2, \ldots, N\}$, where $\langle \hat{e}_i, \hat{e}_j \rangle \langle \hat{e}'_i\, \hat{e}_j \rangle = \delta_{ij}$, has been introduced to obtain the final form (Gantmacher, 1959; Horn and Johnson, 1985), so that $\xi_i = \langle \hat{e}_i, \mathbf{x} \rangle$ and $\eta_i = \langle \hat{e}_i, \mathbf{y} \rangle$. The canonical vector norm in the same space is simply the positive square root of the canonical scalar product of a vector with itself,

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^{\dagger}\mathbf{x}} = \left(\sum_{i=1}^{N} \|x_i\|^2 \right)^{1/2} \tag{84}$$

It should be noted that this norm is positive for all nonzero vectors, as it is a sum of nonnegative terms.

A complex vector space equipped with such a norm and scalar product is called a unitary space. This norm and scalar product endows the complex vector space with a geometrical structure very similar to a real Euclidean space (Schneider and Freed, in press).

The Lanczos vectors generated by the complex symmetric Lanczos algorithm, however, are not orthonormal with respect to this canonical scalar product. Instead, to form a complex orthogonal matrix the Lanczos vectors must satisfy (Gantmacher, 1959; Horn and Johnson, 1985)

$$(\mathbf{q}_i, \mathbf{q}_j) = \mathbf{q}_i^{\text{tr}}\mathbf{q}_j = \sum_{k=1}^{N} q_i^{(k)} q_j^{(k)} = \delta_{ij} \tag{85}$$

where $q_m^{(k)} = (\hat{e}_k, \mathbf{q}_m)$. A set of vectors which satisfies equation (85) is called rectanormal (Choudhury and Horn, 1986; Schneider, 1989; Schneider and Freed, 1989) rather than orthonormal. We note that summands in equation (85) are not necessarily nonnegative, therefore it cannot be used to define a vector norm in the strict sense. However, one is free to define a "pseudonorm" or rectanorm of a vector as

$$\|\mathbf{x}\|_{\text{pn}} = \left(\sum_{i=1}^{N} \xi_i^2\right)^{1/2} \tag{86}$$

Complex vector spaces equipped with a generalized norm and scalar product of this type [cf. equations (85) and (86)] are called complex orthogonal spaces. Since the rectanorm of a vector can be negative or even imaginary, the geometry of these spaces is much different from the more familiar unitary and Euclidean spaces (Schneider and Freed, 1989).

The calculation of the spectral function from the Lanczos tridiagonal matrix is discussed in Section 3.5.

3.3. The Real Symmetric Conjugate Gradients Algorithm

The conjugate gradients method of Hestenes and Stiefel (1952) amounts to a clever adaptation of the Lanczos algorithm for solving systems of linear algebraic equations. An exposition of the conjugate gradients algorithm as applied to RSPD matrices and its connection with the Lanczos algorithm (cf. Section 3.1) is presented here.

In the conjugate gradients algorithm, a sequence of approximate solution vectors, $\mathbf{x}_j$, for the problem

$$\mathbf{A}\mathbf{x} = \mathbf{v} \tag{87}$$

is desired. Associated with each approximate solution vector is a residual vector,

$$\mathbf{r}_j = \mathbf{v} - \mathbf{A}\mathbf{x}_j \tag{88}$$

Clearly, the norm of these residual vectors can be used as a measure of the deviation of the approximate solution vectors from the true solution vector, $\mathbf{x}$.

This algorithm can be used in several ways in magnetic resonance problems. First, the spectral function at a particular field position can be calculated directly, since equation (1) is equivalent to

$$I(\omega - \omega_0) = \left(\frac{1}{\pi}\right)\langle\langle x(\omega - \omega_0) | v\rangle\rangle \tag{89}$$

where $|x(\omega - \omega_0)\rangle\rangle$ is the solution to

$$[\tilde{\Gamma} - i(\omega - \omega_0)\mathbf{1} - i\mathscr{L}]|x(\omega - \omega_0)\rangle\rangle = |v\rangle\rangle \tag{90}$$

Second, it can be used as an alternative means of generating the Lanczos tridiagonal matrix [cf. equation (106) below] in the calculation of the entire spectral function. In fact, the conjugate gradients algorithm often has a significant advantage over the usual Lanczos tridiagonalization procedure as described in Section 3.2. The third major application of the conjugate gradients algorithm is in the determination of a minimal truncation scheme for a given problem. Here, equation (90) is solved for a set of field positions; during this "field sweep," the maximum value of the modulus of each of the components of the solution vector divided by the amplitude of the spectral function [cf. equation (89)] is monitored. This quantity,

$$s_j = \max_{\omega - \omega_0} \frac{\|\hat{e}_j^{\text{tr}}\mathbf{x}(\omega - \omega_0)\|}{\|\mathbf{v}^{\text{tr}}\mathbf{x}(\omega - \omega_0)\|} \tag{91}$$

where $\hat{e}_j$ is the jth member of the original basis set, can then be used to assess the significance of the contribution of each basis vector in the overall solution of the problem. This application is discussed in more detail in Section 3.4.

In the conjugate gradients algorithm, the residual vectors are taken to be colinear with the Lanczos vectors,

$$\mathbf{q}_{j+1} = \pm\frac{1}{\rho_j}\mathbf{r}_j \tag{92}$$

This choice ensures that the sequence of the residual vectors, and therefore the sequence of approximate solution vectors, terminates after at most N steps (in exact arithmetic).

If this scheme of defining a residual vector and determining the approximate solution vector is to be successful, then one must be able to solve equation (88), or its equivalent, for the associated approximate solution vectors without actually inverting the matrix. This difficulty is most easily overcome by reformulating the conjugate gradients algorithm as a minimization problem of a special type. This discussion follows Golub and Van Loan (1983) quite closely. We consider the functional

$$F[\mathbf{y}] = (\tfrac{1}{2})\mathbf{y}^{\text{tr}}\mathbf{A}\mathbf{y} - \mathbf{y}^{\text{tr}}\mathbf{v} \tag{93}$$

and note that, if $\mathbf{y} = \mathbf{x}$ where $\mathbf{x}$ is the solution to equation (87), then

$$\begin{aligned} F[\mathbf{x}] &= (\tfrac{1}{2})\mathbf{x}^{\text{tr}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\text{tr}}\mathbf{v} \\ &= (\tfrac{1}{2})\mathbf{x}^{\text{tr}}\mathbf{A}(\mathbf{x} - \mathbf{v}) - (\tfrac{1}{2})\mathbf{x}^{\text{tr}}\mathbf{v} \\ &= (\tfrac{1}{2})\mathbf{v}^{\text{tr}}\mathbf{A}^{-1}\mathbf{v} \end{aligned} \tag{94}$$

It is easy to see that this is actually a unique minimum point by examining $F[\mathbf{x} + \mathbf{y}]$ where $\mathbf{y}$ is an arbitrary vector,

$$\begin{aligned} F[\mathbf{x} + \mathbf{y}] &= (\tfrac{1}{2})(\mathbf{x} + \mathbf{y})^{\text{tr}}\mathbf{A}(\mathbf{x} + \mathbf{y}) - (\mathbf{x} + \mathbf{y})_j^{\text{tr}}\mathbf{v} \\ &= F[\mathbf{x}] + (\tfrac{1}{2})\mathbf{y}^{\text{tr}}\mathbf{A}\mathbf{y} \end{aligned} \tag{95}$$

Since A has been assumed to be RSPD, the solution vector x must be unique and

$$F[\mathbf{x}+\mathbf{y}] - F[\mathbf{x}] > 0 \tag{96}$$

for all vectors $\mathbf{y} \neq 0$. Thus, the problem of solving the system of linear algebraic equations in equation (87) can be reformulated as a minimization problem involving the functional $F[\mathbf{y}]$ defined by equation (93).

One is now left with the problem of developing a systematic scheme for the minimization of $F[\mathbf{y}]$ subject to the constraint that the residual vectors are collinear with the Lanczos vectors [cf. equation (92)]. To begin, it is clear from equation (88) that the difference between two successive residual vectors can be written as

$$\begin{aligned} \mathbf{r}_j - \mathbf{r}_{j-1} &= -\mathbf{A}(\mathbf{x}_j - \mathbf{x}_{j-1}) \\ &= -a_j \mathbf{A}\mathbf{p}_j \end{aligned} \tag{97}$$

This defines a new sequence of vectors, $\mathbf{p}_j$, which are known as conjugate direction vectors. The relationships between successive members of the sets of residual and approximate solution vectors can be reexpressed using the conjugate direction vectors:

$$\mathbf{r}_j = \mathbf{r}_{j-1} - a_j \mathbf{A}\mathbf{p}_j \quad \text{and} \quad \mathbf{x}_j = \mathbf{x}_{j-1} + a_j \mathbf{p}_j \tag{98}$$

Now, $F[\mathbf{x}_j]$ is just

$$\begin{aligned} F[\mathbf{x}_j] &= F[\mathbf{x}_{j-1} + a_j \mathbf{p}_j] \\ &= F[\mathbf{x}_{j-1}] + \left(\frac{a_j^2}{2}\right)\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j - a_j \mathbf{p}_j^{\mathrm{tr}}\mathbf{v} \end{aligned} \tag{99}$$

By setting the derivative of equation (99) with respect to a_j equal to zero, it is easy to show that the minimum value

$$F[\mathbf{x}_j] = F[\mathbf{x}_{j-1}] - \left(\frac{1}{2}\right)\frac{(\mathbf{p}_j^{\mathrm{tr}}\mathbf{r}_{j-1})^2}{\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j} \tag{100}$$

is attained when

$$a_j = \mathbf{p}_j^{\mathrm{tr}}\mathbf{r}_{j-1}/\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j \tag{101}$$

It follows from this result and equation (98) that $\mathbf{p}_j$ and $\mathbf{r}_j$ are orthogonal,

$$\begin{aligned} \mathbf{A}\mathbf{p}_j &= -a_j^{-1}(\mathbf{r}_j - \mathbf{r}_{j-1}) \\ -a_j \mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j &= \mathbf{p}_j^{\mathrm{tr}}(\mathbf{r}_j - \mathbf{r}_{j-1}) \\ 0 &= \mathbf{p}_j^{\mathrm{tr}}\mathbf{r}_j \end{aligned} \tag{102}$$

A more detailed treatment (Golub and Van Loan, 1983; Cullum and Willoughby, 1985) shows that the conjugate direction vectors can also be obtained by

$$\mathbf{p}_j = \mathbf{r}_{j-1} + b_j \mathbf{p}_{j-1} \tag{103}$$

where

$$b_j = \rho_{j-1}^2/\rho_{j-2}^2 \tag{104}$$

Now, equations (101)-(103) can be used to derive a more symmetrical formula for a_j,

$$a_j = \frac{(\mathbf{r}_{j-1} + \beta_j \mathbf{p}_{j-1})^{\mathrm{tr}}\mathbf{r}_{j-1}}{\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j} = \frac{\rho_{j-1}^2}{\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j} \tag{105}$$

This completes the exposition of the basic equations used in the conjugate gradients algorithm [cf. equations (98), (103)-(105)]. Using these equations, the problem of minimizing the functional $F[\mathbf{y}]$ over the entire N-dimensional space, and therefore the solution of the original set of linear algebraic equations, has been reduced to a sequence of simple one-dimensional minimizations.

The connection between the residual and Lanczos vectors in equation (92) can be exploited to derive the following equation for the Lanczos tridiagonal matrix from the quantities a_j and ρ_j as calculated by the conjugate gradients algorithm (Golub and Van Loan, 1983; Cullum and Willoughby, 1985):

$$\mathbf{T}_N = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{tr}} \tag{106}$$

where D is a diagonal matrix with elements

$$D_{ij} = a_j^{-1}\delta_{ij} \tag{107}$$

and L is a lower bidiagonal matrix with elements

$$L_{ij} = \delta_{ij} - \left(\frac{\rho_{j-1}}{\rho_{j-2}}\right)\delta_{i-1,j} \tag{108}$$

The sign ambiguity noted in equation (92) is a result of the fact that it is only ρ_j^2 that is actually calculated within the conjugate gradients algorithm. This same sign ambiguity can also lead to sign differences in the off-diagonal matrix elements of the tridiagonal matrices generated by the Lanczos and conjugate gradients algorithms. These sign differences do not affect the calculated spectrum, since only the squares of the off-diagonal matrix elements occur in the continued-fraction expansion of the spectral function [cf. equation (124)].

3.4. The Complex Symmetric Conjugate Gradients Algorithm

The generalization of the conjugate gradients algorithm to handle complex symmetric matrices is analogous to the generalization of the Lanczos algorithm treated in Section 3.2. In particular, the rectanorm [cf. equation (86)] and associated scalar product [cf. equation (85)] should be used everywhere instead of the usual unitary space norm and scalar product [cf. equations (83) and (84)].

However, there is one significant difficulty which can arise. Since the matrix A is now complex and symmetric, it cannot be assured that $\mathbf{p}_j^{\mathrm{tr}}\mathbf{A}\mathbf{p}_j$ does not vanish for some index j, thereby disrupting the algorithm. In fact, the way that the matrix elements of the stochastic Liouville superoperator are calculated, this problem always arises on the very first step for calculations involving isotropic liquids. Fortunately, there is a very simple solution to this problem in that one can always choose an offset δ such that $\mathbf{p}_j^{\mathrm{tr}}(\mathbf{A} + \delta\mathbf{I})\mathbf{p}_j \neq 0$ for all j. In practice, δ is usually chosen to be a small positive number on the order of the expected inhomogeneous linewidth of the experimental spectrum under consideration. Alternatively, one could simply use the Lanczos algorithm for the calculation. The presence of such an offset does not affect the results of either the Lanczos or conjugate gradients algorithms.

Using the connection between the Lanczos and conjugate gradients algorithms [as defined in equation (106)], one can calculate the entire spectral function while taking advantage of the modulus of the rectanorm of the residual vector to determine when to terminate the algorithm. The use of the error estimate provided by the residual vector can lead to substantial savings in computer time, despite the fact that each conjugate gradients step involves a little more computation than the corresponding Lanczos step. The offset δ which is required to avoid spurious divisions by zero is simply subtracted from the diagonal matrix elements of the tridiagonal matrix to give, within sign changes in the off-diagonal matrix elements, the same tridiagonal matrix as generated by the complex symmetric Lanczos algorithm in the absence of the offset.

Finally, the utility of the complex symmetric conjugate gradients algorithm in the determination of the MTS warrants further discussion. In this application, one is interested only in obtaining the quantities s_j, as defined in equation (91), which are used to assess the overall significance of the contribution of the jth basis vector to the spectral function. Therefore, it is not necessary to obtain the Lanczos tridiagonal matrix, but rapid convergence becomes very important since equation (90) must be solved at 10 to 100 different values of the sweep variable. Various methods of accelerating the convergence of the real symmetric conjugate gradients algorithm are known (Golub and Van Loan, 1983). Of these methods, the simplest one is known as the diagonal preconditioning scheme, which works well in cases where the diagonal matrix elements are much larger in amplitude than the off-diagonal elements. This is precisely the case for the matrix of the stochastic Liouville superoperator where the diagonal elements of the diffusion superoperator tend to increase in amplitude like L^2 while the magnitude of the elements of the Liouville superoperator tend to decrease as L increases. The diagonal preconditioning scheme amounts to solving the modified set of equations,

$$\mathbf{M}^{-1}(\mathbf{A} + \delta\mathbf{I})\mathbf{x} = \mathbf{M}^{-1}\mathbf{v} \tag{109}$$

where M is the diagonal matrix whose elements are just the positive square roots of the real parts of the diagonal matrix elements of $\mathbf{A} + \delta\mathbf{I}$. Thus, the real parts of the diagonal matrix elements of the scaled matrix

$$\tilde{\mathbf{A}} = \mathbf{M}^{-1}(\mathbf{A} + \delta\mathbf{I})\mathbf{M}^{-1} \tag{110}$$

are unity. This scaling improves both the numerical stability and rate of convergence of the algorithm. It is important to observe that the transformation defined in equation (110) is not a similarity transformation. Using equation (110), it is straightforward to rewrite equation (109) in the form

$$\tilde{\mathbf{A}}\mathbf{y} = \mathbf{w} \tag{111}$$

where $\mathbf{w} = \mathbf{M}^{-1}\mathbf{v}$, and the solution to the original set of equations can be obtained from the relation

$$\mathbf{x} = \mathbf{M}\mathbf{y} \tag{112}$$

Thus, it is clear that a minor variant of the standard conjugate gradients algorithm may be used to solve the set of diagonally preconditioned equations and obtain the desired quantities. However, since this scheme involves a transformation which is not a similarity transformation [cf. equation (110)], one cannot reconstruct the Lanczos tridiagonal matrix in any simple manner.

3.5. The Continued-Fraction Representation of the Spectral Function

The continued-fraction representation of the spectral function holds a central position in both the theoretical and practical aspects of the methodology presented here. From a theoretical point of view, it allows one to view the Lanczos and conjugate gradients algorithms as a natural translation of the very powerful but abstract projection operator techniques of nonequilibrium statistical mechanics into a concrete computational framework. From a practical point of view, it represents an efficient means of obtaining an entire sequence of approximations to the spectral functions directly from the tridiagonal matrix. The emphasis here will be on the practical side. The interested reader is encouraged to consult the literature for surveys of the theoretical aspects (Moro and Freed, 1981; Wassam, 1985a; Dammers, 1985; Moro and Freed, 1986; Schneider and Freed, 1989).

The sequence of continued-fraction approximants to the spectral function is defined in terms of the elements of the tridiagonal matrix. The derivation of these approximants is quite straightforward in that it relies only on some basic linear algebra. To begin, the spectral function can be written in the form

$$J^{(N)}(z) = \mathbf{v}^{\dagger}[z\mathbf{I}_N + \mathbf{A}]^{-1}\mathbf{v} \tag{113}$$

The superscript and subscript N refer to the dimension of the matrix A. The sequence of approximants to be constructed is actually a sequence of approximations to $J^{(N)}(z)$. The extent to which $J^{(N)}(z)$ is a good approximation to the experimental spectrum depends on the values chosen for the parameters on which the matrix A depends as well as on the basis set (cf. Section 2).

The transformation matrix $\mathbf{Q}_N$ defined by the Lanczos vectors allows one to rewrite equation (113) in terms of the tridiagonal matrix $\mathbf{T}_N$,

$$J^{(N)}(z) = \mathbf{v}^{\mathrm{tr}}\mathbf{Q}_N^{\mathrm{tr}}\mathbf{Q}_N[z\mathbf{I}_N + \mathbf{A}]^{-1}\mathbf{Q}_N^{\mathrm{tr}}\mathbf{Q}_N\mathbf{v} \tag{114}$$

$$= (\mathbf{Q}_N\mathbf{v})^{\mathrm{tr}}[z\mathbf{I}_n + \mathbf{T}_n]^{-1}(\mathbf{Q}_N\mathbf{v}) \tag{115}$$

Since the matrix $\mathbf{Q}_N$ is orthogonal, it must satisfy $\mathbf{Q}_N^{\text{tr}}\mathbf{Q}_N = \mathbf{I}_N$. In addition, the matrix elements in the first row of $\mathbf{Q}_N$ are just the components of $\mathbf{v}$. Using these two facts to simplify the right-hand side of equation (115) yields the particularly simple result that $J^{(N)}(z)$ is just the (1, 1) matrix element of the inverse of $[z\mathbf{I}_N + \mathbf{T}_N]$, i.e.,

$$J^{(N)}(z) = [z\mathbf{I}_N + \mathbf{T}_N]^{-1}_{1,1} \tag{116}$$

The entire sequence of continued-fraction approximants is obtained directly from equation (116). This is accomplished by using the determinant-cofactor formulas for the elements of the inverse matrix and by using Laplace's method to expand the determinants and cofactors. Using this approach, it is convenient to introduce the determinants $D^N_{l,m}$ of the diagonal blocks of $[z\mathbf{I}_N + \mathbf{T}_N]$,

$$D^N_{l,m}(z) = \det\begin{pmatrix} z+\alpha_l & \beta_l & & & \\ \beta_l & z+\alpha_{l+1} & \beta_{l+1} & & \\ & \beta_{l+1} & z+\alpha_{l+2} & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & z+\alpha_m \end{pmatrix} \tag{117}$$

where it is assumed that $N \geq m \geq l$. The spectral function can then be written as the ratio of two of these determinants,

$$J^{(N)}(z) = D^N_{2,N}(z)/D^N_{1,N}(z) \tag{118}$$

It is noteworthy that the determinants appearing in the numerator and denominator of equation (118) are simply polynomials in z of order $N-1$ and N, respectively. The continued-fraction representation of $J^{(N)}(z)$ follows from equation (118) by using Laplace's method to derive a recurrence relation for the determinants. The expansion of $D^N_{1,N}(z)$ about its first row or column gives the result

$$D^N_{1,N}(z) = (z+\alpha_1)D^N_{2,N}(z) - \beta_1^2 D^N_{3,N}(z) \tag{119}$$

This expansion can then be inserted into the denominator of equation (118). Dividing both numerator and denominator by $D^N_{2,N}(z)$ gives

$$J^{(N)}(z) = \frac{D^N_{2,N}(z)}{(z+\alpha_1)D^N_{2,N} - \beta_1^2 D^N_{3,N}(z)} \tag{120}$$

$$= \frac{1}{z+\alpha_1 - \beta_1^2 D^N_{3,N}(z)/D^N_{2,N}(z)} \tag{121}$$

The reapplication of this technique for the ratio of determinants occurring in the denominator of equation (121) gives

$$J^{(N)}(z) = \cfrac{1}{z+\alpha_1 - \cfrac{\beta_1^2}{z+\alpha_2 - \beta_2^2 D^n_{4,N}(z)/D^N_{3,N}(z)}} \tag{122}$$

The recursive nature of the expansion into a continued fraction is now obvious. Using the standard notation for continued fractions, the complete expansion of $J^{(N)}(z)$ can be written as

$$J^{(N)}(z) = \frac{1}{z+\alpha_1-}\,\frac{\beta_1^2}{z+\alpha_2-}\,\frac{\beta_2^2}{z+\alpha_3-}\cdots\frac{\beta_{N-2}^2}{z+\alpha_{N-1}-}\,\frac{\beta_{N-1}^2}{z+\alpha_N} \tag{123}$$

The sequence of continued-fraction approximants, $J^{(N)}_m(z)$, is obtained simply by truncating the continued-fraction expansion appearing in equation (123),

$$J^{(N)}_m(z) = \frac{1}{z+\alpha_1-}\,\frac{\beta_1^2}{z+\alpha_2-}\,\frac{\beta_2^2}{z+\alpha_3-}\cdots\frac{\beta_{m-2}^2}{z+\alpha_{m-1}-}\,\frac{\beta_{m-1}^2}{z+\alpha_m} \tag{124}$$

This amounts to assuming that $\beta_m = 0$ for $m = 2, 3, \ldots, N$.

The line-shape function, $S^{(N)}_m(\Delta\omega)$, associated with a particular continued-fraction approximant is (in angular frequency units)

$$S^{(N)}_m(\Delta\omega) = \frac{1}{\pi}\text{Re}\{J^{(N)}_m(1/T_2^0 + i\Delta\omega)\} \tag{125}$$

where $1/T_2^0$ is a real constant used to model line-broadening effects which are not explicitly included in the computation. However, following established practice, all calculated line-shape functions presented here will be plotted as a function of applied magnetic field rather than angular frequency. As stated earlier in Section 2, this conversion is only valid in the high-field approximation. In addition, the usual ESR signal obtained using field modulation and a lock-in amplifier is related to the derivative of $J^{(N)}_n(z)$,

$$S'^{(N)}_m(\omega-\omega_0) = \frac{1}{\pi}\text{Re}\left\{\frac{dJ^{(N)}_m(1/T_2^0 + i(\omega-\omega_0))}{d\omega}\right\} \tag{126}$$

It is very important to note that since the previous elements of the Lanczos tridiagonal matrix are not modified by the execution of further steps, the continued-fraction approximant $J^{(N)}_m(z)$ can actually be evaluated after only m Lanczos steps. In other words, each member of the set of continued-fraction approximants to $J^{(N)}(z)$ can be evaluated after the corresponding number of Lanczos steps has been completed. The behavior of the family of continued-fraction approximants for a model problem is demonstrated in Section 3.6.

Since the determinant in the denominator of equation (118) is just the characteristic polynomial of $\mathbf{T}_N$, it is clear that there must be a close relationship between the eigenvalues and weight factors of $\mathbf{T}_m$ and the associated continued-fraction approximant. It is useful to explore this point in some detail, as these ideas are used frequently in the discussion of the convergence of the sequence of continued-fraction approximants in Section 3.6.

First, we note that the mth approximate spectral function $J_m^{(N)}(z)$, just like the true spectral function $J^{(N)}(z)$, may be written as the ratio of two determinants, where the order of the determinant in the denominator is one greater than the order of the determinant in the numerator,

$$J_m^{(N)}(z) = D_{2,m}^N(z)/D_{1,m}^N(z) \tag{127}$$

The denominator of equation (127) is just the characteristic polynomial of the truncated tridiagonal matrix $\mathbf{T}_m$. Assuming that the polynomial in the denominator of equation (127) can always be factored into the product of m distinct linear factors

$$D_{1,m}^N(z) = \prod_{j=1}^{m} (z + \Lambda_j) \tag{128}$$

one may decompose the ratio of polynomials in equation (127) into a sum of m distinct partial fractions (Gantmacher, 1959),

$$J_m^{(N)}(z) = \sum_{j=1}^{m} \frac{w_j^2}{z + \Lambda_j} \tag{129}$$

The coefficients w_j^2 are called weight factors. This procedure of expanding ratios of polynomials into partial fractions is commonly used in the indefinite integration of rational functions.

The connection between the eigenvalue-weight factor and continued-fraction representations can also be derived by writing equation (127) as the (1, 1) element of the inverse of the tridiagonal matrix $[z\mathbf{I}_m + \mathbf{T}_m]^{-1}$,

$$J_m^{(N)}(z) = \hat{e}_1^{tr}[z\mathbf{I}_m + \mathbf{T}_m]^{-1}\hat{e}_1 \tag{130}$$

where $\hat{e}_1^{tr} = (1, 0, 0, \ldots, 0)$. Assuming that the eigenvalues of $\mathbf{T}_m$ are distinct (Schneider, 1988), $\mathbf{T}_m$ can always be diagonalized by a complex orthogonal transformation (Gantmacher, 1959; Horn and Johnson, 1985),

$$\mathbf{O}\mathbf{T}_m\mathbf{O}^{tr} = \mathbf{\Lambda}^{(m)} \tag{131}$$

This result can now be used to simplify equation (130),

$$\begin{aligned} J_m^{(N)}(z) &= \hat{e}_1^{tr}\mathbf{O}^{tr}\mathbf{O}[z\mathbf{I}_m + \mathbf{T}_m]^{-1}\mathbf{O}^{tr}\mathbf{O}\hat{e}_1 \\ &= (\mathbf{O}\hat{e}_1)^{tr}[z\mathbf{I}_m + \mathbf{\Lambda}_m]^{-1}(\mathbf{O}\hat{e}_1) \\ &= \sum_{j=1}^{m} \frac{O_{j1}^2}{z + \Lambda_j^{(m)}} \end{aligned} \tag{132}$$

A comparison of equations (129) and (132) shows that the weight factors w_j^2 are just the squares of the elements of the first row of the transformation matrix $\mathbf{O}$ which diagonalizes $\mathbf{T}_m$. In contrast to the elements of $\mathbf{T}_m$, which are not altered by the execution of more Lanczos steps, the eigenvalues and weight factors of $\mathbf{T}_m$ and $\mathbf{T}_{m+1}$ can be dramatically different. This point is explored in Section 3.6.

3.6. Convergence of the Sequence of Approximate Spectral Functions

There are two different types of convergence which must be considered in the application of the Lanczos algorithm to calculating magnetic resonance spectra. First, there is the question of how well $J^{(N)}(z)$ approximates the "true" spectral function defined by equation (1). The difference between equations (1) and (123) is due to the truncation of the basis set to a finite dimension. This is just the MTS problem discussed in Section 3.4. The second type of convergence is the central topic in this section, namely, the convergence of the sequence of approximants to the spectral function. For simplicity, this discussion will be presented in terms of the eigenvalue-weight factor representation of the approximate spectral functions rather than the continued-fraction representation and will be heuristic in nature with little attention paid to the mathematical details. The discussion of the convergence of the spectral functions in terms of the continued-fraction representation is more pleasing from a theoretical point of view, but it also involves more complicated and unfamiliar mathematical techniques (Schneider, 1989; Schneider and Freed, 1989).

To orient the reader, the values of the line-shape function $S_m^{(N)}(\omega - \omega_0)$ associated with a sequence of continued-fraction approximants for a particular calculation are presented in Figure 3 as a function of both the number of Lanczos steps and the applied magnetic field. The final line-shape function in the absorption [cf. equation (125)] and derivative modes is presented in Figures 4 and 5, respectively. This calculation was performed using TEMPONE-like magnetic parameters and moderately slow isotropic Brownian diffusion ($R_\parallel = R_\perp = 10^6\,\mathrm{s}^{-1}$). The MTS used here was taken from the appropriate entry in Table 2. These values were determined (Vasavada *et al.*, 1987) using the field-sweep conjugate gradients method (cf. Section 3.4). These plots all include a residual linewidth of 0.3 G, which is fairly typical for TEMPONE radicals in this motional regime.

The general features of the derivative spectrum given in Figure 5 are probably quite familiar to most readers. The large overall spectral width (60 G $\sim 2A_{zz}$), the narrowness of features of the high- and low-field extrema, and the splitting observed in the central region are all indications of slow rotational motion. This spectrum was calculated using 62 Lanczos steps giving a conjugate gradients residual of $\|r\|_m \sim 10^{-2}$ (cf. Section 3.4) indicating the convergence of the cw ESR spectrum. The corresponding absorption spectrum given in Figure 4 is probably not so familiar. It is presented here to help the reader in understanding the two-dimensional plot in Figure 3.

This particular calculation was chosen because it demonstrates the most common convergence behavior of the approximants as a function of the number of Lanczos steps. In Figure 3 the absorption-mode line-shape function is plotted as a function of the number of Lanczos steps and the difference between the applied magnetic field and the field at the center of the spectrum (3200 G). According to convention, each line-shape function is normalized to unit area.

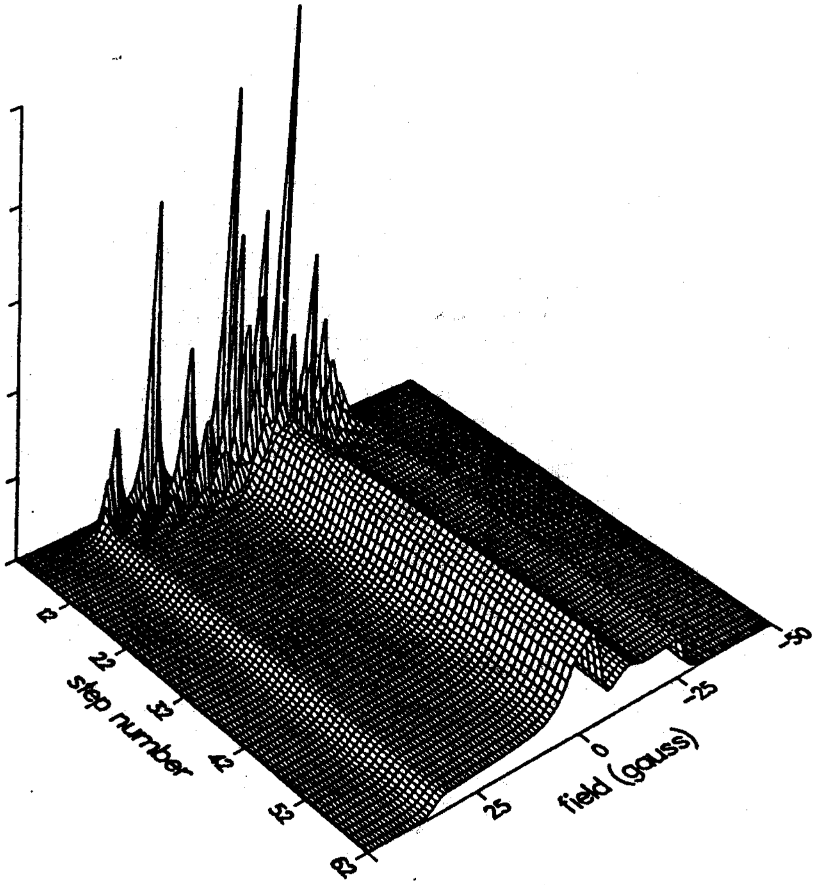

Figure 3. Two-dimensional plot of the convergence of the absorption-mode spectral function [cf. equation (125)] as a function of magnetic field and the number of Lanczos steps for TEMPONE spin probe undergoing slow isotropic Brownian diffusion. The diffusion and basis-set parameters were taken from line 2 of Table 2 and the magnetic parameters are given at the bottom of that table. The spectral functions for Lanczos steps 2 to 62 are plotted here. Note that the magnetic field increases to the left.

The final trace along the field axis in the immediate foregound of Figure 3 represents the fully converged absorption mode cw ESR spectrum determined by these input parameters. This trace is identical to the trace in Figure 4, except that Figure 3 is plotted so that the field decreases from left to right. The initial trace in the extreme background in Figure 3 corresponds to only two Lanczos steps.

The most striking aspect of Figure 3 is how quickly, as a function of the number of Lanczos steps, the approximate line-shape function begins to look like a reasonable cw ESR spectrum. An examination of the traces in the background of Figure 3 shows that the gross features of the final spectrum are well represented after only 10 to 15

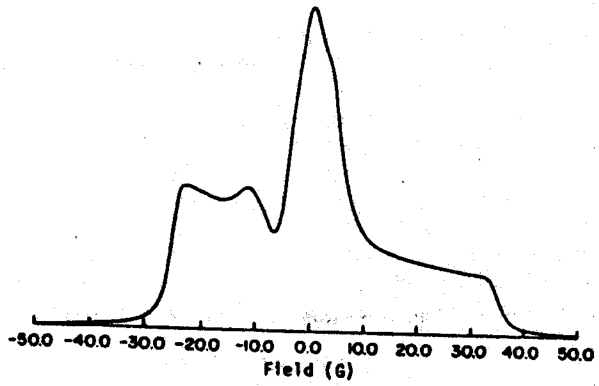

Figure 4. Fully converged (62 Lanczos steps) cw ESR absorption spectrum for the same magnetic, diffusion, and basis-set parameters as for Figure 3. This plot corresponds to the trace in the immediate foreground of Figure 3. Note that the magnetic field increases toward the right.

Lanczos steps. It is most convenient to discuss this phenomenon in terms of an eigenvalue-weight factor decomposition of the line-shape functions [cf. equations (129) and (132)] rather than in terms of continued fractions. These representations are, however, equivalent as was shown in Section 3.5. From this point of view, the spikes in the line-shape functions corresponding to the individual eigenvalues of the smaller tridiagonal matrices appear to decrease in amplitude and move out toward the wings to the spectrum as the dimension of the tridiagonal matrix is increased (cf. Figure 3). The regions between the spikes are gradually filled in by the new eigenvalues which must appear as the dimension of the tridiagonal matrix is increased. This extremely rapid convergence of the gross features of the spectrum is observed in essentially all calculations.

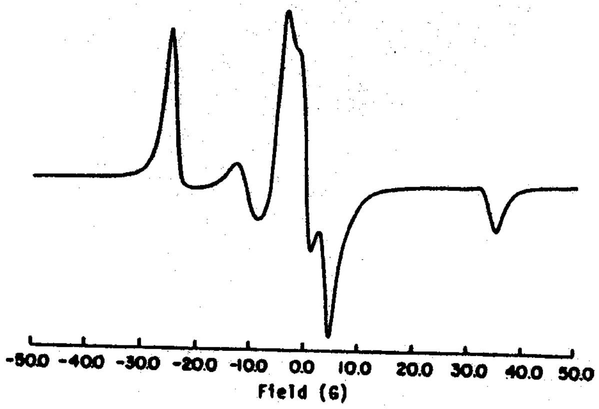

Figure 5. Fully converged (62 Lanczos steps) cw ESR derivative spectrum [cf. equation (126)] for the same magnetic, diffusion, and basis-set parameters as for Figures 3 and 4. Note that the magnetic field increases toward the right.

TABLE 2
Truncation Parameters and MTS for cw ESR Spectra

No.	Spin probe[a]	R[b]	λ[c]	$L^e_{\max}$[d]	$L^o_{\max}$[d]	$K_{\max}$[d]	$M_{\max}$[d]	N[e]	$N_{\min}$[f]
1	TEMPONE	10^7	0	6	3	2	2	42	33
2	TEMPONE	10^6	0	14	7	6	2	171	100
3	TEMPONE	10^5	0	30	13	10	2	543	256
4	TEMPONE	10^4	0	54	15	10	2	990	447
5	TEMPONE	10^7	10	10	None	2	2	63	26
6	TEMPONE	10^6	5	12	3	2	2	78	42
7	TEMPONE	10^6	10	10	None	0	2	33	29
8	TEMPONE (90° tilt)	10^7	1	6	3	2	6	288	74
9	TEMPONE (90° tilt)	10^7	10	10	9	4	4	822	69
10	TEMPONE (90° tilt)	10^6	10	12	11	6	6	1779	245
11	CSL	10^6	0	14	7	14	2	231	162
12	CSL	10^5	0	30	13	30	2	762	474

[a] Values of g and A tensors for TEMPONE: $g_{xx} = 2.0088$, $g_{yy} = 2.0061$, $g_{zz} = 2.0027$, $A_{xx} = 5.8$ G, $A_{yy} = 5.8$ G, $A_{zz} = 30.8$ G. Values of g and A tensors for CSL: $g_{xx} = 2.0021$, $g_{yy} = 2.0089$, $g_{zz} = 2.0058$, $A_{xx} = 33.44$ G, $A_{yy} = 5.27$ G, $A_{zz} = 5.27$ G. Static magnetic field $B_0 = 3300$ G, $(\gamma_e T_2)^{-1} = 1.0$ G.

[b] Rotational diffusion constant (units of s^{-1}).

[c] Coefficient of first term in the expansion of the scaled restoring potential, $-U(\Omega)/k_B T$.

[d] $L^e_{\max}$ and $L^o_{\max}$ are the largest even value of L and odd value of L, respectively, for which there exist basis vectors with $s_j > 0.03$. Similarly, $K_{\max}$ and $M_{\max}$ are the largest values of K and M for which this occurs.

[e] N is the dimension of the matrix if all the basis vectors whose indices are less than or equal to $L^e_{\max}$, $L^o_{\max}$, $K_{\max}$, and $M_{\max}$ are included.

[f] $N_{\min}$ is the dimension of the MTS (the number of basis vectors for which $s_j > 0.03$).

The convergence of the fine details of the line-shape function is emphasized in the difference plot presented in Figure 6. This plot was generated by subtracting the final converged line-shape trace from all the previous traces in Figure 3. There are three major points illustrated in Figure 6. The most obvious point is again the rapid convergence of the gross spectral features. The second point is that the wings of the spectrum converge far more rapidly than the central region. This phenomenon can be rationalized in terms of the close spacing of the eigenvalues in this area. It is well known in the real symmetric case that eigenvalues of the tridiagonal matrix corresponding to eigenvalues of the original matrix which are widely separated from their neighbors converge most quickly (Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985). This general trend also seems to hold true for complex symmetric matrices, where it has previously been noted that the line-shape function converges to within experimental accuracy well before closely spaced eigenvalues of the original matrix are resolved (Moro and Freed, 1981). Finally, it should be noted that it can happen that there is little change in the approximate line shape for a reasonably large number (10–15) of Lanczos steps followed by a sudden change where significant features can appear or disappear. Again, this phenomenon is well known in the real symmetric case where it is referred to as misconvergence (Parlett and Nour-Omid, 1985).

Until now, this discussion has been focused on interpreting the convergence of the approximate line shape in terms of the eigenvalues and weights of the sequence of tridiagonal matrices generated by the Lanczos algorithm. However, to understand the effects of finite precision computer arithmetic on the approximate line shapes, the

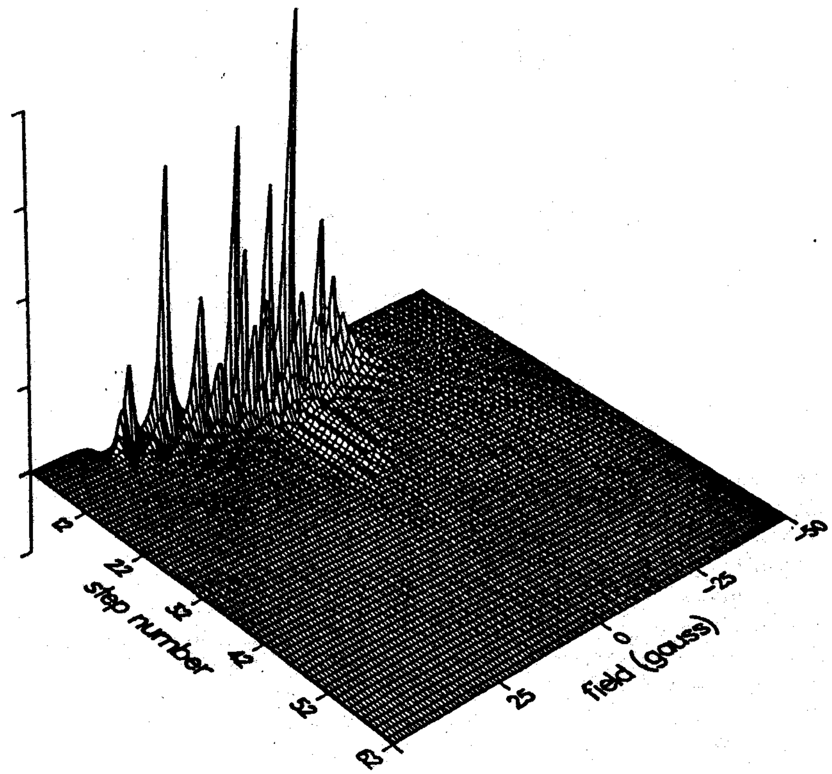

Figure 6. Two-dimensional difference plot of the convergence of the absorption-mode spectral function [cf. equation (12$)] emphasizing the rapid convergence of the wings of the spectrum but slower convergence of the central region. This plot was generated by subtracting the fully converged absorption spectrum (shown in Figure 4) from all of the traces in Figure 3. Note that the magnetic field increases to the left as in Figure 3.

behavior of the eigenvalues and weights themselves must be examined more closely. Unfortunately, since both the eigenvalues and weights are complex numbers, it is nearly impossible to plot all of these quantities, as a function of the dimension of the associated matrices, in a meaningful way. However, it is possible to plot the imaginary parts of the eigenvalues of the tridiagonal matrices (i.e., the magnetic field positions of the individual complex Lorentzians) used to generate Figure 3. This is done in Figure 7 to further illustrate the spreading of the eigenvalues toward the wings of the spectrum as the dimension of the tridiagonal matrix increases. This is, however, just a small part of the story.

An examination of the lists of eigenvalues and weights reveals that, for the matrices of dimension 2 to 16, the real parts of the eigenvalues (i.e., the widths of the complex Lorentzians) are relatively small ($\mathrm{Re}\{\Lambda_j\} \leq 7.1$ G) and all eigenvalues have relatively large weights ($\|w_j^2\| \geq 10^{-3}$). In addition, the real parts and weights for the eigenvalues

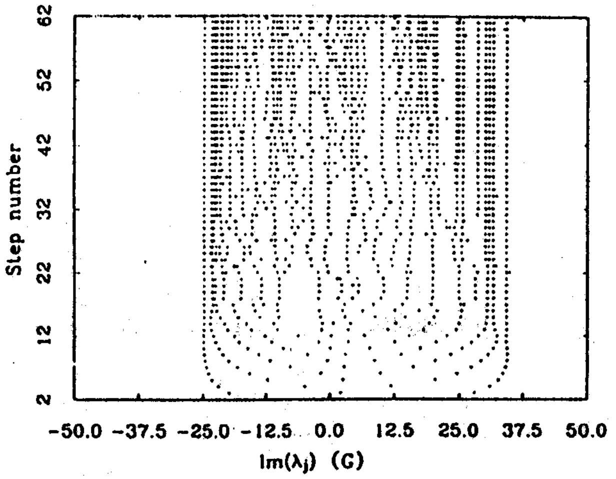

Figure 7. The imaginary parts of the eigenvalues of the sequence of Lanczos tridiagonal matrices used to generate Figure 3 as a function of the number of Lanczos steps. Note the rapid convergence of the eigenvalues near the high- and low-field extrema. In contrast, the central region shows considerable variation in the positions of the eigenvalues as the number of Lanczos steps increases.

in any given region in the spectrum vary substantially from step to step. Note that the gross features of the full converged spectrum have essentially been defined by step 15 (cf. Figure 3).

Now, after 17 steps a qualitatively different type of eigenvalue appears—one with a large real part (13.7 G) and a small weight ($\|w^2\| = 1.75 \times 10^{-6}$). The imaginary part of this eigenvalue is −13.0 G. This eigenvalue "disappears" at step 18. The eigenvalues and weights display bizarre behavior for the next several steps. In particular, some eigenvalues develop negative real parts corresponding to negative linewidths! Though these eigenvalues usually have negligible weight factors, occasionally they do not. However, if the eigenvalues with negative real parts have nonnegligible weight factors, the real part of these weight factors is also negative. In addition, these eigenvalues appear as part of a "pair." The imaginary parts of eigenvalues of the two members of the pair are nearly the same, while the real parts of the eigenvalues are nearly equal in magnitude, but opposite in sign. The weight factor associated with the member whose eigenvalue has a positive real part is usually quite small. The net contribution of this pair to the spectrum is quite small. As the dimension of the tridiagonal matrix increases still further, the frequency and relative importance of these physically unrealistic eigenvalues diminish rapidly which is indicative of their "spurious" nature.

In addition, near the end of this calculation, duplicate or "ghost" eigenvalues also begin to appear. However, the weight factor for all but one of the numerically multiple eigenvalues is on the order of the machine precision used in the calculation. In general, this remains valid until one performs far more than N Lanczos/conjugate gradients steps on an $N \times N$ matrix. Therefore, these unwanted duplicates of converged eigenvalues cannot corrupt the calculated spectrum using the computational strategy advocated here.

These observations suggest that it is not necessarily sufficient to calculate the line-shape function after every Lanczos step and terminate the procedure when no significant change is observed. This was the original suggestion by Moro and Freed (1981) on how to monitor convergence in practical problems. Indeed, this procedure works well on simple problems but sometimes can be misleading in problems involving potentials or very anisotropic reorientation.

Extensive experience (Vasavada *et al.*, 1987; Schneider, 1989; Crepeau, private communication) now suggests that the rectanorm of the residual occurring in the complex symmetric conjugate gradients algorithm can be used as a more reliable means of monitoring the convergence of the line-shape function (cf. Section 3.4).

More information on the qualitative and quantitative aspects of the convergence of the complex symmetric Lanczos algorithm for the calculation of ESR line-shape functions can be found in the references (Moro and Freed, 1981; Dammers, 1985; Vasavada *et al.*, 1987; Schneider, 1989; Schneider and Freed, 1989).

4. COMPUTATIONAL CONSIDERATIONS

Until about 1980, almost all slow motional magnetic resonance spectral calculations were either based on the solution of a system of linear equations by the *LU* algorithm (Freed *et al.*, 1971a, 1971b) or the complete diagonalization of the stochastic Liouville matrix (Gordon and Messenger, 1972; Goldman *et al.*, 1972; Freed, 1976). At that time, Moro and Freed (1981) introduced a modification of the Lanczos algorithm for real symmetric matrices (Lanczos, 1950; Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985) which is capable of handling complex symmetric matrices and which leads to at least an order of magnitude reduction in computer time over the traditional methods. The reader is referred to the literature for more information on the application of the complex symmetric Lanczos algorithm (Moro, 1980b; Moro and Freed, 1981; Cullum and Willoughby, 1985; Dammers, 1985; Moro and Freed, 1986; Schneider, 1988; Schneider and Freed, in press) and the related conjugate gradients algorithm (Vasavada *et al.*, 1987; Schneider, 1989; Schneider and Freed, 1989) to spectral calculations of various types.

In this section, the names of variables used in the computer programs will be printed in lower case teletype-style typeface (e.g., `ndim`, `zmat`) while the names of the programs, subroutines, and disk files will be printed in the corresponding upper case teletype-style typeface (e.g., `LBLL`, `EPRLL`).

The general organization of the program is as follows:

`LBLL` This program will prompt the user for the magnetic tensor parameters, diffusional model and associated parameters, basis-set truncation values, etc. This must be the first program of the set run, since all the other programs require the parameter file generated by `LBLL` as input. The parameters required by `LBLL` are defined in Section 2. This program also determines whether or not the dimension of the matrix is too large for the associated arrays in the main calculation programs, `EPRLL`, `EPRCGL`, and `EPRBL`.

`EPRLL` This program constructs the stochastic Liouville matrix, then tridiagonalizes it using the complex symmetric Lanczos algorithm.

EPRCGL This program is quite similar to EPRLL, but uses the complex symmetric conjugate gradients algorithm to tridiagonalize the stochastic Liouville matrix. The ouput of this program is compatible with the output of EPRLL.

EPRBL This program constructs the stochastic Liouville matrix in the same manner as done in EPRLL and EPRCGL. Rather than constructing the Lanczos tridiagonal matrix, the main calculation performed in EPRBL is the solution of the system of linear algebraic equations defining the spectral function by means of a preconditioned complex symmetric conjugate gradients procedure as outlined in Section 3.4. This calculation is performed at a sequence of magnetic fields to determine the contribution of each basis vector to the overall spectral function. This information can then be used to truncate the basis set for future calculations involving similar input parameters.

TDLL This program reads in the parameter file written by LBLL and the Lanczos tridiagonal matrix generated and stored by EPRLL or EPRCGL. The user is presented with several options:

1. List the elements of the tridiagonal matrix on the screen.
2. Write the elements of the tridiagonal matrix into a formatted file.
3. Diagonalize the tridiagonal matrix using the complex symmetric *QL* algorithm to obtain the eigenvalues and associated weighting factors.
4. Calculate the approximate cw ESR spectrum by evaluating the continued fraction representation of the spectral function.

The first three options are employed by more experienced users to analyze the behavior of the Lanczos algorithm as explained in Section 3.6. The last option is the preferred method of obtaining the approximate cw ESR spectrum for comparison with experimental data.

TNLL This is an auxiliary program which is used to process and print the output file generated by EPRBL.

MATLST This is an auxiliary program included here to aid users in porting these programs to new systems. It reads an optional matrix file generated by EPRLL or EPRCGL and prints it out in an intelligible manner.

VECLST This program plays a role analogous to MATLST for the starting vector.

D200 This program will calculate the order parameter for a radical subjected to a certain restoring potential. The order parameter is defined to be the canonically weighted average value of the $L = 2$ Legendre polynomial,

$$\langle \mathscr{D}^2_{00}(\Omega)\rangle = \langle P_2(\cos\beta)\rangle = \frac{\displaystyle\int d\Omega\, P_2(\cos\beta)\exp[-U(\Omega)/k_B T]}{\displaystyle\int d\Omega \exp[-U(\Omega)/k_B T]} \qquad (133)$$

This program treats only the special case where $-U(\Omega)/k_B T = \lambda_0^2 P_2(\cos\beta)$, since in this case the integral above can easily be evaluated. This program can be used in conjunction with LBLL to find the proper potential coefficient to give a desired order parameter.

4.1. Naming Conventions for Files

To simplify the problem of keeping track of the FORTRAN source files for these programs, as well as the output files generated by them, certain naming conventions have been adopted.

First, all of the FORTRAN source files contain only one program or subroutine. The file names for these source files are constructed from the name of the program or subroutine followed by the suffix ".F." For example, the source code for the program EPRCGL can be found in the file EPRCGL.F.

The common block and parameter definitions that are needed in many programs and subroutines are defined in so-called include files. These include files have been given names which reflect the usage of their contents and carry the suffix ".INC." For example, the value of the unit roundoff error is defined in the include file RNDOFF.INC and the arrays to hold the matrix elements of the stochastic Liouville superoperator are defined in EPRMAT.INC.

A consistent naming convention has also been used for the output files generated by these programs. This convention is similar to the one used by Moro (1980a). In particular, all files associated with a given set of input parameters are referred to using a two-character file identifier. This two-character file identifier is specified by the user when LBLL is executed to create the input parameter file for the calculation. The actual file names are constructed from a two-character prefix which indicates the nature of the contents of file, the two-character file identifier which was assigned in LBLL, and a three-character suffix which specifies whether the file is formatted or unformatted. The names of all formatted output files carry the suffix ".FMT," while the file names of unformatted files use the ".DAT" suffix.

To illustrate this convention, suppose that a given input parameter set was assigned the identifier "XY" by the user when running LBLL. The output file written by LBLL which contains the desired input parameters is given the name LBXY.DAT. The prefix "LB" indicates that this particular file was created by LBLL, while the suffix reflects the fact that this file was written in an unformatted manner. The names of the output files generated by the various programs will be outlined below when the programs are discussed in detail. In those discussions, the two-character file identifiers will be referred to simply as "??" to indicate any two characters that can be used to construct a valid file name.

4.2. Array Dimensions and Common Blocks

The declarations of parameters used to define the dimensions of important arrays that are used by several programs or subroutines are given in the include files STDDIM.INC and MAXL.INC. The user can easily change the dimensions of all the arrays whose dimensions are determined by the parameters in these two files in all the programs in a consistent manner by simply changing the value assigned to these parameters and recompiling all of the programs.

The file MAXL.INC defines an integer parameter (mxlval) which is used in LBLL to determine whether or not the maximum L value requested by the user is too large for a particular array (array nrb) in the subroutine MATRLL. This array is used to hold the starting/ending column number for each L value. This information is used to skip over the calculation of matrix elements which must be zero on symmetry grounds.

The file STDDIM.INC declares the integer parameters defining the maximum allowable number of Lanczos or conjugate gradients steps (mxstep) which may be requested in LBLL. It also declares the maximum allowable values for the dimension of the matrix of the stochastic Liouville operator (mxdim), as well as the number of matrix elements (mxel). Finally, it defines the maximum number of input parameter sets (mxcalc). All of these parameters are used to define dimensions of arrays in many programs and subroutines. They are also used to make sure that array subscripts remain within bounds.

The common block used to hold the data read in from the input parameter files generated by LBLL is defined in the file EPRDAT.INC. This file is included in all main programs except D200. The data stored in this common block is read in by the subroutine RDDAT and written by the subroutine WRDAT. The contents of this common block are defined in Section 4.3.

The arrays to hold the matrix elements of the stochastic Liouville superoperator (arrays zmat and zdiag) and associated column pointer array (array izmat) are defined in EPRMAT.INC. The dimension of these arrays is determined by the values stored in STDDIM.INC. The function of these arrays is discussed in Section 4.4.2.

The logical unit numbers for reading from the user's keyboard (luttyi), writing to the user's display device (luttyo), and reading and writing disk files (ludisk) are defined in the include file STDIO.INC. All I/O is done to one or another of these logical units. Only one logical file unit is required for reading and writing to the disk, since these programs were all written in a manner which requires at most only one disk file open at a time.

Finally, the files PIDEF.INC and RNDOFF.INC define the values of π (pi) and the unit roundoff error (rndoff), respectively. A short program to determine the unit roundoff error for any given computer is also given in the header of the file RNDOFF.INC.

4.3. The Parameter Input Program: LBLL

This program is the starting point for all spectral and basis-set calculations. The definitions of the various input parameters have been presented in Section 2.

At the beginning of the program the user is prompted for a two-character file identifier. This file identifier must consist of two nonnull ASCII characters and is used to construct a file name for the input parameter file. These characters are read in and an inquiry is made as to whether or not an old parameter file with the same name already exists on the disk. If such a file exists, it is read in, and its contents are displayed on the screen. Otherwise, a new file will be created at the end of the program. If an old file exists, its contents will be overwritten with the new input parameters.

Several sample input parameter files are included in the program distribution. The contents of these files are displayed in the Appendix. The actual input parameter files are stored in unformatted files on the disk. This scheme was adopted to discourage the direct manipulation of the contents of the parameter files. This ensures that the input parameter file is internally consistent.

The process of entering or altering parameters is quite straightforward. For each different type of parameter, the old or default values are displayed on the screen, and the user is prompted to decide whether he/she wants to change these values. The prompt string which is used is

```
Do you want to change these values? [Y/N]:
```

to which the user can respond affirmatively by entering a Y, y, or 1 followed by a carriage return. Alternatively, negative responses can be indicated by entering N, n, or 0 followed by a carriage return. All other responses are ignored and the prompt string is redisplayed.

The first group of parameters requested by the program comprises the rigid limit magnetic parameters used in the definition of the Liouville superoperator. These parameters are the principal components of the g tensor (gxx, gyy, and gzz), two times the nuclear spin (in2), the principal components of the A tensor (axx, ayy, and azz), and the static magnetic field corresponding to the center of the spectrum (b0). The program assumes that the values of the principal components of the A tensor and static field are given in units of gauss. The properly scaled $F_{\mu,m}$ tensors are constructed from this information (arrays fgm and fam).

The members of the second group of parameters are all related to the definition of the diffusion superoperator. The first parameter in this second group is the diffusion parameter index (ipdf) which selects the rotational diffusion model used in the calculation. The options for ipdf are

- ipdf = 0: Brownian diffusion (isotropic or anisotropic media),
- ipdf = 1: Free or jump diffusion (isotropic media),
- ipdf = 2: Brownian diffusion with anisotropic viscosity (anistropic media).

We note that not all options are valid if a restoring potential is present. When starting a series of calculations, it is usually recommended to start with the Brownian diffusion model unless other experimental data clearly indicate that the jump or free diffusion models are more appropriate. In general, this choice leads to the smallest basis sets and therefore the fastest computations. In addition, the Brownian model often leads to the best agreement with experiment—at least for nitroxide radicals in viscous liquids.

The next set of parameters requested by the program are the perpendicular (dxy) and parallel (dzz) principal components of the rotational diffusion tensor, $R_\perp$ and $R_\parallel$, respectively, in units of s^{-1}. If a nonzero value of ipdf has been selected, then the user is prompted to input the residence times $\tau_\perp$, $\tau'_\perp$, and $\tau_\parallel$ from equation (41) in units of s (tl, tkxy, and tkzz) and the corresponding model exponents (mpl, mpkxy, and mpkzz). Values for these parameters are not requested if Brownian motion has been selected (ipdf = 0). Note that it is possible to use different models for the perpendicular and parallel rotational motions. As stated previously, choosing a mixed diffusion model can be justified only when the diffusion tensor is very anisotropic. It is important to bear in mind the fact that the components of the rotational diffusion tensor are closely related to the molecular structure of the spin-bearing molecule (Freed, 1976).

Next, the user is prompted for the parameters related to the contribution to the diffusion superoperator due to jump-type motions of the radical between equivalent sites as defined in equation (51). The relevant parameters are the number of sites (ist) and jump frequency (djf). Also, a value for the Heisenberg spin exchange rate (oss) is requested. These contributions are rarely used in calculations involving nitroxide radicals in viscous media. The units of the discrete jump and Heisenberg spin exchange frequencies are s^{-1}.

The third major group of parameters is related to the restoring potential. First, the user is prompted to input the number of nonzero values of λ^L_K in the expansion of the resoring potential (ipt) in equation (2). For isotropic liquids, ipt = 0. If a nonzero value of ipt is entered, then the user is requested to enter the associated values of L and K (array ixp) and λ^L_K (array cxp). These terms can be input in any order, since they are immediately rearranged into a standard order (array cpot). It is often helpful to consult Figure 9 (see Section 5.1) or run the program D200 (cf. Section 4.7) to determine a reasonable starting value for λ^2_0.

If a nonzero value of ipt has been requested, the user is also prompted for the angle, in degrees, between the static magnetic field and the unique symmetry axis of the restoring potential (psi). In isotropic liquids this angle is chosen to be zero. This completes the specification of the parameters used in the definition of the diffusion superoperator.

The specification of the Liouville superoperator is then completed by defining the diffusion tilt angle (bed) in units of degrees. Using this parameter, the scaled components of the $F_{\mu,R}$ tensors are calculated and stored (arrays fgd and fad). This completes the specification of the parameters defining the Liouville superoperator.

The last major group of parameters is related to defining the basis set and calculation type. First, the user must define the largest even value of L (lemx), odd value of L (lomx), and the maximum allowable values of the K, M, and p^I indices (kmx, mmx, and ipnmx) allowed in the basis. Suggested basis sets for a variety of problems can be found in Tables 2 and 3. We note that in these tables it has been assumed that the maximum value of the p^I index is two [cf. equation (7)]. The basis set defined in this manner is then checked for internal consistency and the appropriate symmetry restrictions are applied to make sure that no unnecessary basis vectors are included in the calculation. All basis vectors which satisfy the limits set by the user and can couple to the starting vector are included in the calculation. Second, the user is asked to set the maximum number of Lanczos/conjugate gradients steps allowed (nstep). Third, the user is asked to select the type of computation to be done (itype). The options are

- itype = 0: Lanczos spectral calculation,
- itype = 1: conjugate gradients spectral calculation,
- itype = 2: field swept conjugate gradients calculation for the determination of the MTS.

If a Lanczos calculation is selected, then no further questions are asked. Otherwise, the user must specify the maximum allowable modulus of the rectanorm of the conjugate gradients residual vector (cgtol) which is used as a termination criterion in the conjugate gradients iterations as well as the real and imaginary parts of the origin shift (shiftr and shifti). This origin shift is required to avoid spurious divisions by zero during the conjugate gradients iterations. Finally, if a field-sweep conjugate-gradients calculation has been requested, the user must supply the initial and final fields (fieldi and fieldf) as well as the number of field positions at which the conjugate-gradients iteration is to be performed (npt). This completes the specification of the basis set and calculation options.

At this point, the new parameters are displayed on the screen for review. Since most internal limits and symmetry restrictions are silently enforced, the final values of the parameters displayed may not agree with the parameters specified by the user. This phenomenon may occur if the maximum values of the indices defining the basis set are not consistent with the appropriate symmetry restrictions. The program uses a very conservative strategy in modifying parameters given by the user. The user is strongly encouraged to determine the source of any discrepancies between the final parameter values and those that he or she gave in response to the program prompts before proceeding with a spectral calculation.

The last question presented to the user is whether or not a complete listing of the basis set should be displayed on the screen. If this is desired, the indices for the entire basis set will be listed on the screen in order of appearance in the stochastic Liouville superoperator matrix element calculation subroutine. Whether or not this final display option is selected, the program checks to make sure that the basis set is not too large for the arrays currently dimensioned in the main calculation programs. A warning message is issued if the matrix is too large. The program will then write out the parameter file, overwriting the old file if present, and exit. This is the only output file generated by LBLL and is assigned the name LB??.DAT. This parameter file is required to run all programs except D200.

4.4. Spectral Calculations: EPRLL and EPRCGL

The main spectral calculation programs are EPRLL and EPRCGL. These two programs are very similar and will be treated together whenever possible. The programs proceed in two stages.

In the first stage, the user is prompted to input a sequence of two-character file identifiers. This sequence is terminated by simply entering a carriage return at the prompt for another file identifier.

The second stage is the main body of the program involving the sequential execution of the following steps for each input file:

- construct a set of file names using the file identifier supplied by the user,
- read in the input parameter file created by LBLL and verify that the calculation type is correct,
- calculate the matrix elements of the stochastic Liouville operator,
- calculate the elements of the starting vector,
- tridiagonalize the matrix of the stochastic Liouville operator by the desired method (complex symmetric Lanczos or conjugate-gradients algorithms),
- update the input parameter file and save it,

- create or overwrite the output file containing the matrix elements of the Lanczos tridiagonal matrix.

If problems arise for a given input file, the remainder of that calculation is bypassed, and the next file is read in and processed.

Typically, there is only one new output file created by these programs. This file, TD??.DAT, contains the elements of the Lanczos tridiagonal matrix. This file is processed by TDLL (cf. Section 4.5). The input parameter file LB??.DAT, originally created by LBLL, is also modified. In addition, there are several sections of code that will produce files containing the elements of the stochastic Liouville matrix and starting vector, MT??.DAT and VT??.DAT, respectively (cf. Section 4.7). These files are needed only for debugging purposes, thus the sections of code that generate them are commented out in the programs supplied on the diskette.

The parameter file LB??.DAT is read into the common block defined in the include file EPRDAT.INC by the subroutine RDDAT. This common block is then used to pass these parameters to the matrix element calculation subroutine MATRLL and the starting vector calculation subroutine STVECT.

The dimensions of the various arrays are defined in the include file STDDIM.INC. The vector arrays are, for the most part, defined in dimension statements and passed to subroutines as arguments rather than through common blocks.

In contrast to the vector arrays, the matrix element and matrix element index arrays are defined in a common block in the include file EPRMAT.INC. The dimensions of these arrays are also controlled by parameters defined in the include file STDDIM.INC. The matrix elements and associated indices are calculated in MATRLL and directly stored in this common block. The matrix element and index arrays are passed to the tridiagonalization subroutines via this common block.

4.4.1. The Starting Vector Calculation Subroutine: STVECT

The structure of the subroutine STVECT, which calculates the elements of the starting vector, is quite simple. It consists of five nested loops over the L, K, M, $p^{\prime}$, and $q^{\prime}$ indices (lr, kr, mr, ipnr, and iqnr). The selection rules described in Section 2.7 are used to reduce the number of numerical integrations of equation (57) to a minimum. The calculation of the modified Bessel functions and associated Legendre functions in the integrand of equation (57) is discussed below. Since this integral depends only on the L and K indices, it is evaluated within the loop over K, but before the loop over M. Within the innermost loop, the unnormalized elements of the starting vector are stored (array v) and the norm of the unnormalized starting vector is evaluated. Following the completion of this set of five loops, the starting vector is normalized, the number of nonnegligible elements of the starting vector is determined (nelv), and the imaginary part of the starting vector is zeroed out.

The numerical evaluation of the elements of the starting vector becomes increasingly more difficult as the maximum L and K values in the basis set become larger. This is due to the highly oscillatory nature of the integrand in the definition of the starting vector elements [cf. equation (57)].

In part, these difficulties can be overcome by doing the integral over the Euler angle γ in equation (57) analytically in terms of the modified Bessel functions [cf. equation (61)]. The modified Bessel functions can then be evaluated in a stable fashion. However, this evaluation is not trivial. The method employed in the subroutine BESSI is based on the following recurrence relation,

$$f_{n-1}(x) = (2n/x)f_n(x) + f_{n+1}(x) \tag{134}$$

This equation is satisfied by the modified Bessel functions themselves (McLachlan, 1961; Abramowitz and Stegun, 1964), but is used in the actual calculation to evaluate the ratio of two of these functions. It is known (Sookne, 1973a, 1973b; Cody, 1983; Press *et al.*, 1986) that this recurrence relation is numerically unstable in the direction of increasing n, but stable for decreasing n. Let us assume that one wishes to compute $I_m(x)$. In this scheme one chooses a sufficiently large M, where $M \gg m$, at which to begin the computation. The recurrence relation is then seeded with the values

$$f_{M+1}(x) = I_{M+1}(x)/I_M(x) \approx 0 \tag{135}$$

$$f_M(x) = I_M(x)/I_M(x) = 1 \tag{136}$$

and equation (134) is evaluated for decreasing values of n all the way down to $n = 1$. While equation (136) is exact by definition, equation (135) is only approximately valid. However, for the same reason that the forward recurrence is unstable, the error introduced by this approximation is inconsequential if M is chosen properly (Press *et al.*, 1986). Finally, the desired result is given by

$$I_m(x) = f_m(x)I_0(x)/f_0(x) \tag{137}$$

The value of $I_0(x)$ is computed by using the Taylor series expansion for small values of x, or by an asymptotic approximation for large values by the subroutine BESSI0. The function $I_1(x)$ is also evaluated in this manner in the subroutine BESSI1. All three of these subroutines are modifications of routines with similar functionality in the book by Press *et al.* (1986).

The values of the associated Legendre functions are also calculated using a well-known recurrence relation,

$$(L - M)P_L^M(x) = (2L - 1)xP_{L-1}^M(x) - (L + m - 1)P_{L-2}^M(x) \tag{138}$$

However, this relation is evaluated in the direction of increasing values of L starting with the (exact) initial conditions

$$P_M^M(x) = (-1)^M(2M - 1)!!(1 - x^2)^{M/2} \qquad \text{and} \qquad P_{M+1}^M(x) = (2M + 1)xP_M^M(x)$$

Again, the subroutine PLGNDR used to evaluate the associated Legendre functions is a slight modification of a routine given by Press *et al.* (1986).

The subroutines BESSI and PLGNDR are called by the subroutine FZ to evaluate the integrand of equation (57). The numerical integration itself is performed using a Romberg integration scheme with cautious extrapolation (Press *et al.*, 1986) using the subroutine CCRINT taken from Bruno's thesis (1973).

4.4.2. The Matrix Element Calculation Subroutine: MATRLL

The matrix elements of both the upper and lower halves of the matrix are calculated and sorted in MATRLL, even though the matrix is complex and symmetric. This is done for several reasons. Unfortunately, the most important reason for adopting this approach is not obvious, since it has to do with how most multitasking computer operating systems manage users' programs and data.

Most multiuser operating systems limit the amount of physical memory that can be allocated to any one user. If the matrix is too large to fit in the physical memory allocated to the user by the operating system, the remainder of the data must reside on a disk and be swapped in and out of the limited amount of physical memory allocated to the program. The entire data set is divided up into smaller sections, called pages or segments. Typically, each page contains several thousand floating point numbers.

Now, consider the matrix-vector multiplication operation occurring in both the Lanczos and conjugate gradients algorithms. In most instances, the majority of the computer time used in a spectral calculation is spent doing matrix-vector multiplications. As the computation proceeds, if a matrix element or index which is needed does not currently reside in the physical memory, the page containing that element or index is read-off the disk and overwrites one of the least recently accessed pages allocated to the user. This operation is called paging. Unfortunately, paging is a very expensive operation—typically hundreds or thousands of floating point operations can be executed in the time it takes to read a page in from the disk. Therefore, it is prudent to store the matrix elements in the order that they are accessed. This enables one to minimize these costly paging operations.

When these programs were written, the existence of a scheme for storing only half of the matrix elements, but wording excessive paging overhead, was not known to the authors; therefore, the full matrix store scheme was adopted. Since that time, it was pointed out to us by Dr. Vijay Sonnad that there is a scheme that is widely used in mechanical engineering calculations which does enable one to take advantage of the symmetry of the matrix, but not incur any paging overhead. Unfortunately, there was not enough time to incorporate and fully test this scheme in the present set of programs.

The matrix elements are calculated inside a group of ten nested loops. The outer set of five loops run over the L, K, M, p^{I}, and q^{I} indices for the row (`lr`, `kr`, `mr`, `ipnr`, and `iqnr`), while the inner set of five run over the related indices for the column (`lc`, `kc`, `mc`, `ipnc`, and `iqnc`). The number of the current row (`nrow`) and column (`ncol`) is monitored for use as indices for the matrix elements. The matrix elements of the stochastic Liouville superoperator are assembled inside the innermost loop. In addition, the row index corresponding to the beginning of each block of matrix elements with a given value of L is stored (array `nrb`). These indices are used later to determine in which column to start looking for nonzero matrix elements.

In all cases, the matrix elements of the diffusion superoperator are purely real, while the matrix elements of the Liouville superoperator are purely imaginary. Since the vast majority of off-diagonal matrix elements is purely imaginary, a special storage strategy for the matrix elements has been adopted. This strategy involves storing the real and imaginary parts of the nonzero matrix elements separately, and then only if they are nonzero. In particular, if a nonzero Liouville superoperator matrix element is found, the matrix element counter (`neltot`) is incremented, the matrix element is stored in the matrix element array (`zmat`), and the associated column number is stored in the matrix element index array (`izmat`). Next, if a nonzero diffusion superoperator matrix element for the same row and column is found, the matrix element counter is incremented, the matrix element is stored in the matrix element array, and -1 times the column number is stored in the index array. In addition, the diagonal matrix elements are also stored in a separate array (`zdiag`). This array is used only in the field-sweep conjugate-gradients calculation. The end of a row is signaled by incrementing the matrix element counter and inserting a zero value into both the matrix element and index arrays. In this way, the sign of the index element is used as a flag for whether the associated matrix element is real or imaginary, the absolute value of a nonzero index refers to the column number, and a zero index tags the end of a row. In addition, the end of the matrix is flagged by inserting another zero value into the matrix element and index arrays. This index information can be very efficiently decoded in the matrix-vector multiplication step coded in subroutine SCMVM.

If the number of matrix elements ever exceeds the dimensions of the arrays, the return error code (`ierr`) is set to one and control is passed back to the calling program. If this does not happen, the return code is set to zero indicating successful execution.

One of the major difficulties encountered in the calculation of matrix elements is the efficient and accurate evaluation of the Wigner 3-J symbols. Very slow motional calculations often require L truncation values of 80 or more. However, due to overflow problems, the function subroutine used by Moro is not usable for $L > 48$. This problem has been circumvented for the 3-J symbols of the type

$$\begin{pmatrix} L_1 & L_2 & L_3 \\ M_1 & M_2 & M_3 \end{pmatrix}$$

where $|L_2| \le 2$, by using the algebraic formulas for these quantities. These formulas are given in an appendix in Edmonds's book (1957). These are the only type of 3-J symbols required in the calculation of matrix elements of the stochastic Liouville operator in the absence of a restoring potential. These formulas can safely by used for $L \gg 100$ without overflow. One is still restricted to $L \le 48$ in the presence of a potential. This difficulty could be overcome by using an asymptotic approximation for the 3-J symbols for $L > 48$ (Edmonds, 1957).

4.4.3. The Lanczos and Conjugate-Gradients Subroutines: CSLNZS and CSCG

The complex symmetric Lanczos and conjugate-gradients algorithms used here are very straightforward. The more theoretical aspects of these algorithms have already been treated in Section 3.

The Lanczos subroutine (CSLNZS) is a simple translation of a single vector Lanczos tridiagonalization algorithm for real symmetric matrices given in algorithm 9.1-1 in the book by Golub and Van Loan (1983). The required numerical linear algebra operations are performed by calls to appropriate subroutines. The diagonal (array alpha) and off-diagonal (array beta) elements of the Lanczos tridiagonal matrix are stored and returned to the calling program.

Many of these subroutines are also called by the conjugate gradients subroutine (CSCG). This subroutine is also a simple translation of a single vector algorithm for real matrices given in algorithm 10.2-4 in the book by Golub and Van Loan (1983). The quantities generated by the conjugate-gradients algorithm that are needed for the construction of the Lanczos tridiagonal matrix (a1 and b1) are stored. The conjugate-gradients iteration is terminated when the modulus of the rectanorm of the residual vector becomes less than the tolerance specified by the user or if the maximum number of steps has been exceeded. When the conjugate-gradients iteration has finished, the required arrays are passed to the subroutine which actually constructs the Lanczos tridiagonal matrix (CGLTRI). This subroutine constructs the tridiagonal matrix using the scheme outlined by Golub and Van Loan (1983) and overwrites the input arrays with the elements of the tridiagonal matrix. If the calculation converges within the allotted number of steps, the number of conjugate-gradients steps actually executed (nstep) is returned to the calling program EPRCGL. If the calculation does not converge, −1 times the number of steps is returned as an error signal.

As stated previously, the most time-consuming operation in most spectral calculations is the matrix-vector multiplication step required in both the Lanczos and conjugate-gradients algorithms. Since this is such an important part of the calculation, and the implementation used here requires the decoding of the index array "on the fly," it seems worthwhile to discuss this procedure in detail.

The matrix-vector multiplication subroutine is SCMVM. The actual matrix-vector multiplication is carried out as a sequence of sparse vector dot products in which the rows of the matrix are successively dotted with the input vector (x) to give the elements of the output vector (y). Both the input and output vectors are indexed as two-dimensional arrays of double precision floating point numbers and are passed to this subroutine through the argument list.

The sparsity of the matrix is taken advantage of by doing only those matrix element–vector element multiplications which can give rise to nonzero results. Initially, the accumulators for the real and imaginary parts of the output vector elements (accr and acci) as well as the matrix element (iel) and row counters (ir) are set to zero. Now, the list of matrix elements is processed sequentially by incrementing the matrix element counter and examining the contents of the matrix element index array. If the value of that particular index element is nonzero, the absolute value of the index, which represents the column number of the associated matrix element, is used to pick out the correct element of the input vector. The matrix element is then multiplied by the input vector element in the manner indicated by the sign of the index element, and the accumulators are updated. This process is then repeated with the next matrix element and index until a zero index value is found. Since a zero index value indicates the end of a row, the row counter is incremented, the contents of the accumulators are stored in the positions of the output vector array specified by the updated row counter, and the accumulators are then zeroed out. In this manner, the matrix is processed row by row. When the processing of the last row of the matrix is completed, the algorithm encounters the final zero index element labeling the end of the matrix. When this happens, the row counter is incremented to a value one greater than the dimension of the matrix. This event is used to detect the end of the matrix element and index lists and the completion of the matrix-vector multiplication step.

4.5. Calculation of the Spectral Function: TDLL

The program TDLL processes the Lanczos tridiagonal matrix generated by EPRLL or EPRCGL. Like the rest of the programs described here, the action that the program undertakes is to prompt the user to supply the two-character file identifier for the file to be processed. After accepting the two-character file identifier and constructing the file names, the program reads in the parameter file LB??.DAT written by LBLL and the Lanczos tridiagonal matrix file TD??.DAT which was generated by EPRLL or EPRCGL. The user is then presented with several options:

1. List the elements of the tridiagonal matrix on the screen.
2. Write the elements of the tridiagonal matrix into a formatted file.
3. Diagonalize the tridiagonal matrix.
4. Calculate the cw ESR spectrum.

The first option is self-explanatory. If the second option is selected, a file TD??.FMT is created. This file is just a list of the row number and the real and imaginary parts of the diagonal and superdiagonal elements of the Lanczos tridiagonal matrix.

The third option is to diagonalize the Lanczos tridiagonal matrix to obtain the eigenvalues and associated weighting factors. Though one can calculate the cw ESR spectrum from this information, this is not done here. If the cw ESR spectrum is desired, it can be calculated directly from the tridiagonal matrix by selecting the fourth option. The third option is most often used in the calculation of 2D-ESE spectra (Millhauser and Freed, 1984) and in the study of the convergence of the Lanczos and/or conjugate-gradients algorithms. The diagonalization algorithm used here is a simple variant of the complex symmetric *QL* algorithm of Cullum and Willoughby (1985). The major difference here is that the projections of the eigenvectors of the tridiagonal matrix on the starting vector are needed. The squares of these overlaps are the weighting factors [cf. equation (132)]. If this option is selected, a file EG??.FMT is written which contains the eigenvalue number, and the real and imaginary parts of the eigenvalues and associated weight factors. The eigenvalues are sorted in order of increasing imaginary parts (field positions) and are numbered accordingly. The *QL* and closely related *QR* algorithms are thoroughly treated in many reference works (Parlett, 1980; Golub and Van Loan, 1983; Cullum and Willoughby, 1985).

The most commonly used of the several options is the last one—the calculation of the approximate cw ESR spectrum. This is done by evaluating the continued-fraction representation of the spectral function. The elements of this continued fraction are defined by the matrix elements of the Lanczos tridiagonal matrix. If this option is selected, several more parameters must be supplied by the user. First, the magnetic field corresponding to the center of the spectrum is displayed, then the user is prompted for the beginning and ending fields (bi and bf), in units of gauss, relative to the center of the spectrum. The default values for these parameters are −50 G and 50 G, respectively. Next, the user must supply two parameters defining the intrinsic linewidth for the spectrum (`wint0` and `wint2`). The total intrinsic linewidth (`wint`), which is effectively a constant real positive term added to each diagonal element of the Lanczos tridiagonal matrix during the evaluation of the continued fraction [cf. equation (125)], is defined by

$$w^{\text{int}}(\psi) = w_0^{\text{int}} + w_2^{\text{int}} \cos^2 \psi \tag{139}$$

It is left to the user to verify that this quantity is positive. These parameters can be used to mimic the effects of dynamical processes which have not explicitly been included in the calculation and/or inhomogeneous broadening. The continued fraction itself is evaluated "from the bottom up" at a number of equally spaced points (`mxpt`) between the desired beginning and ending field values, inclusive. The selection of this option generates an output file SP??.FMT, which contains the field values and associated real and imaginary parts of the spectral function. To obtain the usual derivative mode spectrum, the real part of the spectral function must be numerically differentiated. Since many data plotting and analysis packages supply this function, it is not duplicated here.

4.6. "Field Sweep" Conjugate-Gradients Calculations: EPRBL and TNLL

These programs are used only for the determination of the MTS (cf. Section 3.4). The program EPRBL has a structure similar to EPRLL and EPRCGL, but uses the diagonally preconditioned complex symmetric conjugate-gradients algorithm to solve equations (111) and (112) at a set of equally spaced field positions. The range and number of points in the field sweep are determined by parameters entered into LBLL as described in Section 4.3.

For every desired set of input parameters, the matrix and starting vector are generated exactly as in EPRLL and EPRCGL. Within the loop over the input parameter sets, the field sweep is accomplished by varying the imaginary part of the origin shift and using the diagonally preconditioned complex symmetric conjugate-gradients algorithm to solve equations (111) and (112) at each point in the subroutine CSPCCG. This subroutine is based on the diagonally preconditioned version of the general preconditioned conjugate-gradients algorithm for real symmetric matrices given in algorithm 10.3-3 in the book by Golub and Van Loan (1983). However, we recall that only the real parts of the diagonal matrix elements are used as the preconditioning matrix (cf. Section 3.4). The subroutine CSPCCG is supplied with the starting vector (array b), dimension of the matrix (`ndim`), maximum number of conjugate-gradients steps allowed (`cftol`), and origin shift (`temp`). When it completes the calculation it returns the approximate solution vector (array x), the number of conjugate-gradient steps actually executed (`ndone`), and the estimated error based on the modulus of the rectanorm of the residual vector. Immediately after control is returned to EPRBL from CSPCCG, the exact residual is calculated and its unitary space norm is computed to give an improved error estimate (`terror`). The number of conjugate-gradients steps actually executed is also used as a flag to indicate whether or not the calculation in CSPCCG converged. After a short message is printed, the significance factors as defined in equation (91) (array `basis`) and spectral function for that field position as given by equation (89) (array `spect`) are calculated. The field positions at which the current maximum values of the individual significance factors have been detected are also monitored (array `basis`). When the field sweep has been completed, the significance factors along with their associated field-values and the spectral function are written to the disk file BS??.DAT.

The basis-set truncation information contained in the output file BS??.DAT generated by EPRBL is further processed by the program TNLL. In this program, the significance factors and field values are read in and the significance factors are normalized so that the largest factor is unity. This step is necessitated by the fact that the largest significance factor, as defined by equation (91), can vary widely in amplitude. After this scaling has been completed, the s_j are printed along with their associated field values and basis-set quantum numbers. This list can then be scrutinized to determine the MTS (Vasavada *et al.*, 1987).

4.7. Auxiliary Programs: D200, STVT, MATLST, and VECLST

The program D200 can be quite useful in deciding upon a reasonable choice of λ_0^2 if the order parameter is known. The form of the equation for the order parameter actually used in this program can be derived by starting from equation (133) and letting $z = \cos\beta$ to get

$$\langle P_2(z)\rangle = \frac{3}{2}\left\{\frac{\int_0^1 dz\, z^2 \exp[3\lambda_0^2 z^2/2]}{\int_0^1 dz \exp[3\lambda_0^2 z^2/2]}\right\} - \frac{1}{2} \tag{140}$$

Now, the changes of variable $\varepsilon = \sqrt{3\lambda_0^2/2}$ and $y = \varepsilon z$ followed by integration by parts gives

$$\langle P_2(y/s)\rangle = \frac{3}{4\varepsilon^2}\left\{\frac{\varepsilon}{\exp[-\varepsilon^2]\int_0^\varepsilon dy \exp[y^2]} - 1\right\} - \frac{1}{2} \tag{141}$$

The remaining integral in the denominator of equation (141) is known as Dawson's integral (Abramowitz and Stegun, 1964). This integral is evaluated numerically by evaluating a continued-fraction expansion (Dijkstra, 1977). For more information about the numerical and analytical properties of Dawson's integral, see the literature (Fried and Conte, 1961; Abramowitz and Stegun, 1964; Dijkstra, 1977; Bender and Orszag, 1978). This program does not generate any output files.

The program STVECT is also useful during the initial stages of calculations involving restoring potentials. This program is little more than a stand-alone version of the subroutine STVECT. It is used to calculate the elements of the starting vector without doing the entire spectral calculation. In this manner it is possible to determine if a sufficiently large value of $L_{\max}^e$ has been chosen and examine the overall structure of the starting vector. The determination of $L_{\max}^e$ is simply based on making sure that the average magnitude of the elements of the starting vector with $L = L^{e}_{\max}$ is much less than the magnitude of the $L = 0$ terms. A reasonable value for the ratio of the $L = L^{e}_{\max}$ to the $L = 0$ terms is 10^{-2} to 10^{-4}. This program does not produce any output files.

The last two programs are most useful in the process of porting these programs to other machines. The program MATLST is used to list out the matrix elements calculated by the subroutine MATRLL which is called by EPRBL, EPRCGL, and EPRBL. There exist some lines in the programs EPRLL and EPRCGL just after the calls to MATRLL which, if uncommented, will store the matrix elements of the stochastic Liouville operator in a file on the disk. This program will read in those files and generate an intelligible listing of the matrix elements on the screen. This is useful in verifying that the matrix elements generated on the target machine are correct. The program VECLST performs the analogous function for the starting vector. Again, there are commented lines in EPRLL and EPRCGL right after the call to the subroutine STVECT which, if uncommented, will store the elements of the starting vector in a file on the disk in the form that can be read in by VECLST. Neither of these programs generates output files.

4.8. Porting Programs to Other Machines

These programs are based on a set of programs written by G. Moro in 1980 (Moro, 1980a, 1980b; Moro and Freed, 1981; Meirovitch *et al.*, 1982). The older programs were very general, quite complicated, and written in FORTRAN IV for use on a computer with a very limited amount of physical memory. The present set of programs are written in a modular, more easily portable fashion in FORTRAN 77 and make use of the more advanced hardware that is available today. As supplied, these programs should compile and run without error on IBM PCs or compatibles under MS-DOS (ver. 2.11 or higher) and Microsoft Fortran (ver. 3.0 or higher). The PC version of these programs, as supplied here, is intended mainly as a reference point in porting these programs to a larger machine to be used for routine calculations. However, some users may find it sufficient to increase the matrix and vector dimensioning parameters in STDDIM.INC to the limits allowed by their compiler and use a PC for most calculations. The main limitation on this practice is that the segmented addressing used in these machines is not well suited to handling large arrays. Indeed, many FORTRAN 77 compilers for PCs limit the maximum size of single arrays or common blocks to 64 kbytes.

A serious attempt has been made to use only standard FORTRAN 77 features. The major exceptions to this rule are discussed below. This section is quite technical in nature, where it is assumed that the reader is a fluent FORTRAN programmer and has some knowledge of computer hardware.

The modular way in which these programs have been written makes it easier for the user to substitute more accurate or faster library subroutines for many of the function subroutines supplied here. In fact, most commonly available scientific library packages such as IMSL[5] and SLATEC[6] have many routines for calculating special functions and doing basic linear algebraic operations which are superior to those supplied here. Users are encouraged to use these library routines whenever possible.

[5] IMSL is a trademark of IMSL Corporation.

[6] Available from National Energy Software Center, Argonne National Laboratory.

4.8.1. Data Types and Explicit Declarations of Variables

In these programs, the following statements are true:

- All variables and parameters are explicitly declared. This has been done to make it easier to detect spelling errors with compilers which have the capability of flagging undeclared quantities.
- All integer variables and parameters are declared to be of type `integer`, not `integer * 2` or `integer * 4`.
- All floating point numbers are declared of type `double precision`, not `real * 4` or `real * 8`.
- Common blocks and parameters defining the dimensions of important arrays used in several programs or subroutines are defined in include files.

Unfortunately, include statements are not part of the FORTRAN 77 standard and their implementation varies widely.

It has been assumed throughout that long (4 byte) integers are the default integer length. If this is not the case, the use of long integers should be enabled or care taken to ensure that integer overflows do not occur at execution time.

The use of double precision (8 byte) floating point numbers throughout is somewhat excessive. However, most machines with enough computing power to handle serious calculations of this type also have enough physical memory where space is not a major issue. The use of double precision numbers throughout is actually required on some machines with small mantissas and large exponents.

The logical variables used here can be any length, but one cannot go wrong in sticking to the FORTRAN 77 default size of 4 bytes.

Extensive use has been made, especially in EPRLL, of the fact that most compilers store complex numbers as two consecutive real quantities with the real part being first. When this is true, an N-dimensional `complex * 16` array is compatible with an array dimensioned $2 \times N$ (not $N \times 2$!) of type `real * 8`. If, for some reason, complex numbers are not stored in this manner, or if complex arithmetic and data types are not supported at all, the easiest solution would be to change all arrays and variables of type `complex * 16` to $2 \times N$ dimensioned arrays of type `real * 8` and do all complex arithmetic explicitly. This would also require function subroutines of type `complex * 16` to be changed into conventional subroutines which return the real and imaginary part of the result through the argument list.

4.8.2. I/O Issues

One of the major sources of difficulty in porting FORTRAN 77 programs is the lack of uniformity of I/O features among compilers. As a rule, the required I/O functions in these programs are confined to specific subroutines and functions to make the programs easier to port to other machines.

A few of the more important sources of difficulty associated with doing I/O are listed below. Unfortunately, this list is not exhaustive.

- The logical units for writing to and reading from the screen and for opening disk files are defined in the include file STDIO.INC. All programs have been written such that at most one disk file is open at any given time.

- The status keyword "UNKNOWN" has been used in disk file open statements. Some compilers do not recognize this keyword. If this is the case, the keywords "OLD" or "NEW" could be substituted in appropriate places or the defaults status could be assumed and the status definition removed entirely.
- Not all compilers supply an inquire function as used in LBLL. If such a function is not supplied, it is relatively straightforward to write one by trying to open the file using the "NEW" and "OLD" status keywords with the I/O error handling mechanism provided by the ERR keyword.
- The $ edit descriptor to suppress the carriage return after writing to the screen is used when prompting the user for input. Some compilers do not recognize this edit descriptor. It can easily be omitted if necessary.
- A logical record length of 1024 bytes is used for the unformatted direct access files containing the input parameters which are written by the subroutine WRDAT and read in by RDDAT.

4.8.3. Vector and Parallel Computation

The modularity of these programs and the exclusion of screen and disk I/O from the subroutines where the majority of the heavy numerical work is done should make it easier to adapt these programs for use on array processors. We note that the word adapt rather than port has been used here. The work involved in getting programs running on array processors is often considerably more difficult than on conventional computers, because the unique features of the hardware often influence the extent to which the machine is programmable in standard FORTRAN. The added difficulty in adapting a program for use on an array processor must always be weighed against the expected decrease in execution time. In practice, only the serious user who is confronted with a large body of very slow motional spectra to analyze should consider this undertaking.

With this caveat in mind, the only programs which are serious candidates for adaptation to array processors are EPRLL, EPRCGL, and EPRBL. Even here, it may pay to adapt the code only for the Lanczos and/or conjugate-gradients algorithms, while doing the matrix element calculation on the host computer.

This story might be somewhat different on machines that have several array processors. Because of the structure of the arrays used in EPRLL to hold the entries of the stochastic Liouville matrix and the associated integer indices, it should not be too difficult to have different array processors generate the matrix elements for different sections of the matrix once a matrix element calculation subroutine is working at all. In addition, these same structures would allow one to break up the matrix-vector multiplication step of the Lanczos algorithm into several pieces, each of which could be independently executed by a separate array processor. This would not be so easy if only the upper or lower half of the matrix is stored, as was previously done (Moro,1980b; Gorcester, 1985), since each processor would need access to the entire matrix.

5. EXAMPLE CALCULATIONS

To aid the reader in using these programs, several model calculations are discussed in detail in Section 5.1. In addition, a short survey of actual applications of these programs to the analysis of experimental data is given in Section 5.2. The presentation of these examples is intended to demonstrate how to choose reasonable input parameters as well as the wide variety of spectra that can be quickly and easily generated.

5.1. Model Calculations and General Strategy

The purpose of this section is to discuss in some detail a set of example calculations that can be run on a PC. In addition, some hints are given for doing careful, accurate calculations on a new system to extract out dynamical information. The calculations actually presented here are relatively simple but they do model the dynamics of "real" radicals.

The first issue that must be dealt with is the proper choice of g- and A-tensor values. These quantities should be determined for the particular system under study by a separate experiment to obtain the rigid-limit values of the magnetic tensors (Freed, 1976). The validity of the dynamical and structural information obtained from the slow-motional spectral calculation programs depends crucially on the use of accurate magnetic tensors. A rigid-limit spectral calculation program is not included in this set of programs, but several reliable programs can be easily obtained. In the following it will be assumed that the proper rigid-limit magnetic tensors are known. Finally, if the field sweep required to obtain the rigid-limit absorption spectrum is not much smaller than the field at the center of the sweep range, the programs discussed here should not be used to analyze the dynamics of the system since the nonsecular terms in the spin Hamiltonian have been omitted. Systems of this type should be analyzed with a program which includes the nonsecular terms (Moro, 1980a; Meirovitch *et al.*, 1982; Gorcester, 1985).

The next thing to consider is the structure of the radical and the orientation of the principal axes of the magnetic tensors in relation to the principal axes of the diffusion tensor. In many cases, constructing a molecular model is very helpful in determining a proper choice of axes and the associated diffusion tilt angle, ϕ (cf. Figure 2). In many cases, such as for the TEMPONE class of nitroxide spin probes, one can assume a spherically symmetric diffusion tensor. In this case, it is useful to assume that $\hat{z}_R$ coincides with that principal axis of the magnetic frame whose principal value differs the most from the other two. For X-band experiments on nitroxide radicals, the magnetic axis which is most different is the one with the largest A-tensor component. The labels of the magnetic frame axes can also be permuted to make this the $\hat{z}_m$ axis. This choice implies $\phi = 0$ and leads to the stochastic Liouville matrix of smallest dimension. The use of nonzero diffusion tilt angles in conjunction with a spherically symmetric diffusion tensor can lead to much larger matrices, but must finally give the same spectrum as the $\phi = 0$ case. For axially symmetric diffusion tensors, the calculated spectra can depend greatly on the diffusion tilt angle, especially if the diffusion tensor is very anisotropic (Mason *et al.*, 1974; Meirovitch and Freed, 1979; Campbell *et al.*, 1979; Meirovitch *et al.*, 1984). Typically, spin probes bound to polymers and large, rigid radicals such as cholestane require the use of axially symmetric diffusion tensors.

The structure of the thermodynamic phase of the solution must also be known. If the solution under investigation is isotropic (i.e., does not form a liquid crystalline phase at the particular temperature and pressure at which the experiment was conducted), then no restoring potential should be included in the calculation ($\lambda^L_K = 0$ for all L and K), since one cannot exist in isotropic fluids. For radicals in liquid crystalline phases however, the restoring potential models the interaction of the radical with the surrounding solvent molecules and must be included in the calculation. If the sample is ordered on both the macroscopic and microscopic levels, then the director tilt angle, ψ, is determined by the angle between the static magnetic field and the preferred axis of alignment of the solvent molecules. A diagram showing this director tilt angle is given in Figure 1. On the other hand, if the sample is microscopically ordered but macroscopically disordered (MOMD), then it is possible to do a sequence of calculations at different director tilt angles and model the spectrum of the composite as the appropriately weighted integral of the spectra calculated as a function of the director tilt angle (Meirovitch *et al.*, 1984). These calculations are quite time-consuming, and it is more difficult to extract reliable information from such spectra, because they involve the superposition of many overlapping spectra. They should be avoided if possible by using macroscopically ordered samples. It also should be kept in mind that the diffusion tensor, at least to a first approximation, is independent of the value chosen for the coefficients of the restoring potential. It is strongly suggested that one start with a λ_0^2 term alone in the restoring potential and introduce the higher-order terms only if necessary. Often it is simpler to think in terms of the order parameter $\langle \mathcal{D}_{00}^2(\Omega)\rangle = \langle P_2(\cos\beta)\rangle$, rather than in terms of the potential expansion parameters, λ_K^L. To simplify this transitition, Figure 8 and the program D200 (cf. Section 4.7) have been provided.

Except when intentionally studying radical–radical interactions, it is very important to work with radical concentrations low enough to avoid Heisenberg spin exchange effects (as well as effects of dipole–dipole interactions) which broaden the spectra and obscure information about rotational reorientation. The radical concentration at which Heisenberg spin exchange begins to affect the spectrum varies widely and should be determined by experiment. Care should also be taken to purge the solution under investigation of any oxygen for the same reason. These considerations are especially important for relatively fast motional spectra where the intrinsic broadening of the spectral features is largely due to the rotational motion.

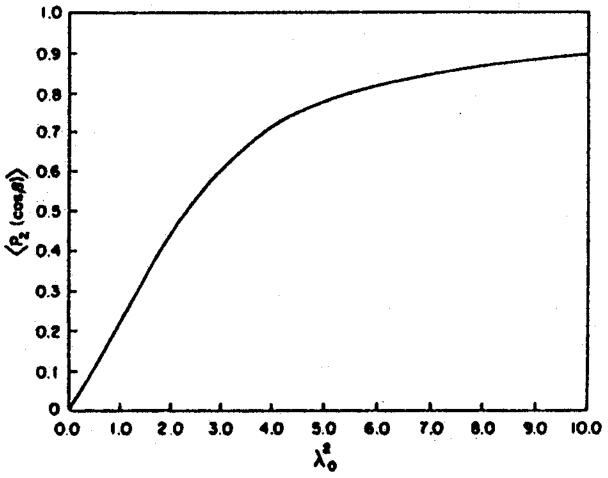

Figure 8. Plot of the order parameter as a function of λ_0^2 for $0 \le \lambda_0^2 \le 10$. For larger values of λ_0^2, the order parameter slowly approaches its asymptotic limit of unity. This plot was generated using the subroutine DAWSON which is described in Section 4.7.

These considerations usually give sufficient information to begin the process of calculating spectra to model experimental results. The general procedure in fitting experimental spectra is as follows:

1. Obtain accurate values for the rigid-limit magnetic tensors from fitting a rigid-limit spectrum.
2. Decide on a rotational diffusion model.
3. Using a molecular model, define an approximate diffusion asymmetry parameter and diffusion tilt angle.
4. Guess at a starting value for the coefficients of the restoring potential if present.
5. Guess at a starting value for the rotational correlation time.
6. Determine the proper basis set for the parameters chosen.
7. Calculate a spectrum and compare with the experimental result.
8. If a satisfactory fit has not been obtained, refine the estimate of $\bar{R}$ and go back to step 5, otherwise refine estimates of previous parameters and go back to step 4, 3, or 2.

Obviously, the time it takes to reach a satisfactory fit is drastically reduced if one can start with good estimates of the final parameters. There are several ways to accomplish this. First, and most important, is to have a thorough understanding of the definitions of the parameters that go into the calculation and the physically reasonable values they could have for the system under study before attempting any calculations. This knowledge will help to avoid spending time doing calculations with physically unreasonable parameters. Second, previous spectral calculations done on similar systems can be very instructive. A survey of previously published experimental spectra and the input parameters used to fit them is given in Section 5.2. And third, other experimental data, such as NMR, neutron, or X-ray scattering, etc., on the same or a closely related system may prove useful in obtaining good estimates of order parameters and other structural information.

Once a reasonable set of input parameters has been decided upon, the proper basis set should be determined. This is most reliably and conveniently accomplished by using the programs EPRBL and TNLL to determine the MTS. The data in Tables 2 and 3 were obtained using the diagonally preconditioned conjugate-gradients algorithm (Vasavada *et al.*, 1987). These truncation parameters can be used as they stand, or they can be used to aid in deciding on input basis-set truncation parameters for EPRBL. We note that the input basis-set truncation parameters for EPRBL must always be larger than the expected MTS basis-set truncation parameters for the problem.

Several of the calculations outlined in Table 2 can be executed on the PC version of the programs supplied with this book. The calculation corresponding to the second line of Table 2 has already been extensively studied in Section 3.6 and will not be treated further. The remaining calculations (lines one and five through seven in Table 2) will be discussed here, although in less detail.

TABLE 3
Table of Truncation Parameters and MTS for 2D-ESE Spectra[a]

No.	Spin probe	R	λ	L^e_{max}	L^o_{max}	K_{max}	M_{max}	N	N_{min}
1	TEMPONE	10^7	0	10	7	6	2	123	92
2	TEMPONE	10^6	0	22	17	10	2	429	307
3	TEMPONE	10^5	0	44	37	18	2	1485	971
4	TEMPONE	10^4	0	88	71	28	2	4614	2506
5	TEMPONE	10^7	10	16	7	2	2	108	76
6	TEMPONE	10^6	5	20	15	8	2	333	209
7	TEMPONE	10^6	10	16	11	4	2	168	120
8	TEMPONE (90° tilt)	10^7	1	10	7	6	10	1440	586
9	TEMPONE (90° tilt)	10^7	10	16	15	6	6	2601	607
10	TEMPONE (90° tilt)	10^6	10	20	19	10	12	8196	2835
11	CSL	10^6	0	22	19	22	2	600	485
12	CSL	10^5	0	46	37	46	2	2310	1815

[a] All parameters have the same meaning as in Table 2, except for s_l which is taken to be 0.0003.

The first line in Table 2 corresponds to moderately slow isotropic Brownian diffusion of the TEMPONE spin probe dissolved in an isotropic liquid ($R_\parallel = R_\perp = 10^7\,s^{-1}$). The smoothness of the prominent spectral features (cf. Figure 9), together with the reduced overall width of the spectrum compared to Figure 5, are indicative of faster motion. The large breadth of the high- and low-field extrema also indicates that this spectrum is in the vicinity of the minimum of the T_2 curve.

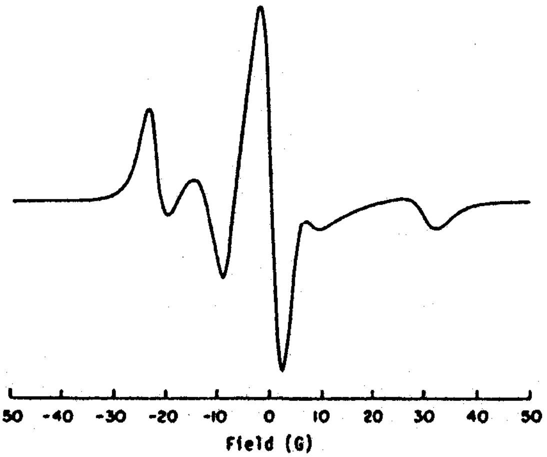

Figure 9. Plot of spectrum generated by input parameter file LBL1.DAT in the Appendix. The input parameters used to generate this plot correspond to line 1 of Table 2.

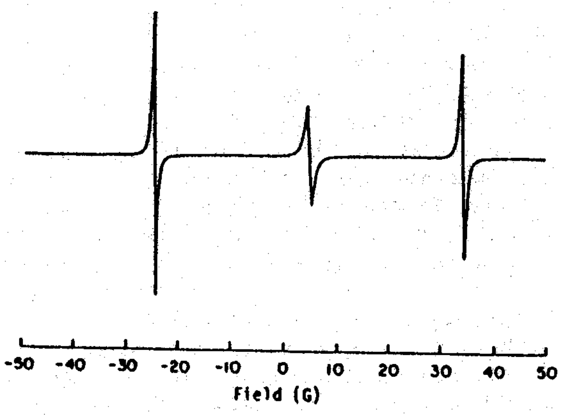

Figure 10. Plot of spectrum generated by input parameter file LBL5.DAT in the Appendix. The input parameters used to generate this plot correspond to line 5 of Table 2.

The calculation defined by the parameters on the fifth line of Table 2 is presented in Figure 10. This plot could be imagined to correspond to an experiment on the TEMPONE spin probe dissolved in a well-aligned smectic liquid crystal. The order parameter in this case is about 0.9 (cf. Figure 8). Note the unusual ratio of the heights of the three hyperfine lines. In addition, the hyperfine lines are well separated, despite the fact that the diffusion tensor is the same as in Figure 5. Both of these features are due to the fact that the probe is highly aligned by the restoring potential.

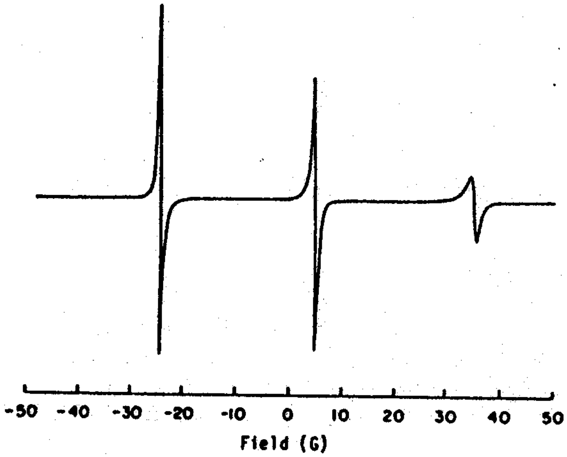

Figure 11. Plot of spectrum generated by input parameter file LBL6.DAT in the Appendix. The input parameters used to generate this plot correspond to line 6 of Table 2.

The plot in Figure 11 corresponds to line six of Table 2. When compared to Figure 10, this plot demonstrates the effects of lowering the order parameter to 0.8 and slowing down the motion by an order of magnitude. Note the widely separated hyperfine lines and shift in the ratio of peak heights.

The final plot in this set, Figure 12, corresponds to line seven of Table 2. This plot, when compared with Figure 11, shows the effect of increasing the order parameter from 0.8 to 0.9. The major changes are in the linewidths rather than the line positions. These last two plots emphasize the fact that not all "three line" nitroxide spectra are due to fast isotropic motion!

5.2. Examples from the Literature

In I, a number of examples of calculated spectra of nitroxide spin probes in isotropic and liquid crystalline media are given. A summary of a number of more recent applications, where the present package of programs can be effectively employed, is given here.

An important test of the theory behind these programs is to properly interpret the spectra of nitroxide spin probes in oriented liquid crystalline media as a function of the director tilt angle. This was done for thermotropic liquid crystals by Meirovitch *et al.* (1982). The appendices to that work also present a slightly more general theory for slow-motional line shapes than is given in the present chapter. Further studies along these lines may be found in Meirovitch and Freed (1984) and Meirovitch (1983) in which a variety of nitroxide spin probes were utilized.

The application of these ideas to oriented model membranes, as well as dispersions, is discussed in detail by Tanaka and Freed (1984, 1985) and Kar *et al.* (1985). In these studies, various chain-labeled lipids and cholestane spin labels were utilized to determine the ordering and dynamics in the various lipid phases as well as the effects of such additives as cholesterol and the small polypeptide gramicidin A. These applications have recently been reviewed by Freed (1987).

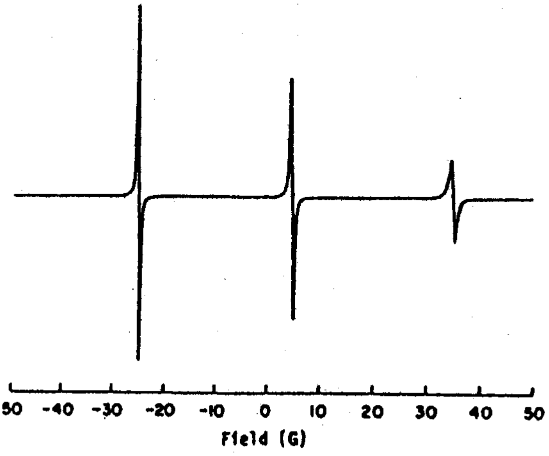

Figure 12. Plot of spectrum generated by input parameter file LBL7.DAT in the Appendix. The input parameters used to generate this plot correspond to line 7 of Table 2.

In dealing with dispersion samples of model membrane preparations, and macroscopic misalignment in general, it is necessary to obtain an average spectrum by properly weighting the spectra calculated for a range of different director tilt angles. This is just the MOMD model which was briefly mentioned in Section 5.1. Calculations based on this model are thoroughly described in a paper by Meirovitch *et al.* (1984). It is also employed in other recent work (Tanaka and Freed, 1984, 1985; Kar *et al.*, 1985).

An important development in the process of comparing calculated spectra to experimental data is the recent use of very general nonlinear least-squares fitting methods in conjunction with the powerful Lanczos and conjugate-gradients algorithms already described here. In this method, one simply allows the nonlinear least-squares algorithm to systematically vary the input parameters for the line-shape calculation program to minimize the deviations between the calculated and experimental spectra. This procedure can require 100 to 150 separate line-shape calculations, yet the overall running time is not inordinate owing to the efficiency of the algorithms chosen. Nevertheless, because of the extensive computations involved, these calculations are currently limited to large mainframe computers. In general, this appears to be a more unbiased, reliable, and accurate way to fit experimental data to theoretical models than the trial-and-error approach (Crepeau *et al.*, 1987).

In the past few years there have also been significant new developments which enhance the resolution and scope of studies of molecular dynamics by ESR spectrscopy. These are time-domain experiments based on electron-spin-echo (ESE) (Millhauser and Freed, 1984) (Millhauser *et al.*, 1987; Freed, 1987) or Fourier transform (FT) methods (Gorcester and Freed, 1986; 1988a; 1988b). The application of ESE techniques to the study of slow motions is discussed from a theoretical point of view by Schwartz *et al.* (1982) and examples of the interpretation of experimental data are given in papers by Millhauser and Freed (1984) and by Kar *et al.* (1984). The basic ESE technique has given rise to a new two-dimensional (2D) ESE technique wherein the homogeneous linewidth at each point in the spectrum is obtained as a function of the magnetic field. It shows a remarkable sensitivity to the details of the reorientational dynamics (Millhauser and Freed, 1984; Millhauser *et al.*, 1987; Freed, 1987) and has been successfully employed in the study of spin probe motion in model membranes (Kar *et al.*, 1984) and in the study of spin labeled proteins (Kar *et al.*, 1987). The present set of programs can be adapted to calculate this type of 2D-ESE spectra as discussed elsewhere (Millhauser and Freed, 1984; Vasavada *et al.*, 1987; Schneider and Freed, in press).

Other kinds of time-domain ESR spectroscopies exist, which cannot be treated by the present set of programs such as spin-echo ELDOR (Hornak and Freed, 1983; Dzuba *et al.*, 1984) and 2D-magnetization transfer ESE (Schwartz *et al.*, 1986). In general, the analysis of these experiments requires the more complete theory involving the slow-motional relaxation of both the longitudinal and transverse components of the magnetization. However, the Lanczos and conjugate-gradients algorithms can still be used to analyze these experiments—once the matrix elements are calculated (Schneider and Freed, 1989). This class of experiments is especially sensitive to very slow molecular motions.

A major breakthrough in ESR techniques has been the development of 2D-FT ESR spectroscopy. At the time of writing, its effectiveness in studying dynamics in the fast-motional regime has been demonstrated for nitroxides (Gorcester and Freed, 1986, 1988a, 1988b). These methods are expected, in the future, to be effectively employed for slow-motional spectroscopy providing valuable new information on the underlying dynamical processes. Such 2D spectra can also be described by a generalization of the methods described here.

Last of all, we wish to point out that these programs can be applied to experiments on slow-motional NMR spectroscopy (Meirovitch and Freed, 1979; Campbell *et al.*, 1979) and in more general form (Meirovitch *et al.*, 1982) to inorganic radicals such as the vanadyl (Campbell and Freed, 1980) and cupric (Subczynski *et al.*, 1987) ions which are often used in biophysical studies.

ACKNOWLEDGMENTS. We wish to thank Dr. R. H. Crepeau and Ms. Nancy Schneider for their extensive help in preparing and critically reading this manuscript. We also thank Mr. J. Gorcester and Dr. S. Rananavare for helpful discussions. This work was supported by NIH grant #GM25862 and NSF grants #DMR86-04200 and #CHE87-03014.

APPENDIX: PARAMETERS FOR EXAMPLE CALCULATIONS

```
##########################   file : 1bl1.dat   ##########################



g-tensor [gxx.gyy.gzz] :  2.00880        2.00610        2.00270

twice the nuclear spin [in2] :  2

A-tensor [axx,ayy,azz] (gauss) :  5.80000        5.80000        30.8000

static field [B0] (gauss) :  3200.00

diffusion parameter [ipdf] = 0

diffusion tensor [dxy.dzz] (1/sec) =  .100000E+08   .100000E+08

discrete jumps parameters [ist.djf] :  0   .000000

Heisenberg spin exchange frequency [oss] =   .000000

number of terms in the potential [ipt] = 0

angle between B0 and local director [psi] (degrees) :   .000000

diffusion tilt index [itd] = 0

truncation values  [lemx,lomx,kmx,mmx,ipnmx] :   6   3   2   2   2

number of Lanczos/CG steps [nstep] :  33

calculation type (0=Lanczos,1=CG) [itype] : 0



########################################################################
```

```
##########################   file : lb12.dat   ##########################


g-tensor [gxx,gyy,gzz] :   2.00880        2.00610        2.00270

twice the nuclear spin [in2] :  2

A-tensor [axx,ayy,azz] (gauss) :   5.80000        5.80000        30.8000

static field [B0] (gauss) :   3200.00

diffusion parameter [ipdf] = 0

diffusion tensor [dxy,dzz] (1/sec) =  .100000E+07    .100000E+07

discrete jumps parameters [ist,djf] :  0    .000000

Heisenberg spin exchange frequency [oss] =   .000000

number of terms in the potential [ipt] = 0

angle between B0 and local director [psi] (degrees) :   .000000

diffusion tilt index [itd] = 0

truncation values  [lemx,lomx,kmx,mmx,ipnmx] :  14    7    6    2    2

number of Lanczos/CG steps [nstep] : 100

calculation type (0=Lanczos,1=CG) [itype] : 0


#########################################################################
```

```
##########################   file : lb16.dat   ##########################


g-tensor [gxx,gyy,gzz] :   2.00880        2.00610        2.00270

twice the nuclear spin [in2] :  2

A-tensor [axx,ayy,azz] (gauss) :   5.80000        5.80000        30.8000

static field [B0] (gauss) :   3200.00

diffusion parameter [ipdf] = 0

diffusion tensor [dxy,dzz] (1/sec) =  .100000E+08    .100000E+08

discrete jumps parameters [ist,djf] :  0    .000000

Heisenberg spin exchange frequency [oss] =   .000000

number of terms in the potential [ipt] = 1

coefficients of the potential :

        ipt = 1   [l,k,coef.] = 2  0    10.0000

angle between B0 and local director [psi] (degrees) :   .000000

diffusion tilt index [itd] = 0

truncation values  [lemx,lomx,kmx,mmx,ipnmx] :  10    1    2    2    2

number of Lanczos/CG steps [nstep] :  63

calculation type (0=Lanczos,1=CG) [itype] : 0


#########################################################################
```

```
##########################   file : lb16.dat   ##########################


 g-tensor [gxx,gyy,gzz] :   2.00880        2.00610        2.00270

 twice the nuclear spin [in2] :  2

 A-tensor [axx,ayy,azz] (gauss) :   5.80000        5.80000        30.8000

 static field [B0] (gauss) :   3200.00

 diffusion parameter [ipdf] = 0

 diffusion tensor [dxy,dzz] (1/sec) =  .100000E+07    .100000E+07

 discrete jumps parameters [ist,djf] :  0    .000000

 Heisenberg spin exchange frequency [oss] =   .000000

 number of terms in the potential [ipt] = 1

 coefficients of the potential :

         ipt = 1  [l,k,coef.] = 2  0    5.00000

 angle between B0 and local director [psi] (degrees) :    .000000

 diffusion tilt index [itd] = 0

 truncation values  [lemx,lomx,kmx,mmx,ipnmx] :  12    3    2    2    2

 number of Lanczos/CG steps [nstep] :  78

 calculation type (0=Lanczos,1=CG) [itype] : 0



######################################################################
```

```
##########################   file : lb17.dat   ##########################


 g-tensor [gxx,gyy,gzz] :   2.00880        2.00610        2.00270

 twice the nuclear spin [in2] :  2

 A-tensor [axx,ayy,azz] (gauss) :   5.80000        5.80000        30.8000

 static field [B0] (gauss) :   3200.00

 diffusion parameter [ipdf] = 0

 diffusion tensor [dxy,dzz] (1/sec) =  .100000E+07    .100000E+07

 discrete jumps parameters [ist,djf] :  0    .000000

 Heisenberg spin exchange frequency [oss] =   .000000

 number of terms in the potential [ipt] = 1

 coefficients of the potential :

         ipt = 1  [l,k,coef.] = 2  0    10.0000

 angle between B0 and local director [psi] (degrees) :    .000000

 diffusion tilt index [itd] = 0

 truncation values  [lemx,lomx,kmx,mmx,ipnmx] :  10    1    0    2    2

 number of Lanczos/CG steps [nstep] :  33

 calculation type (0=Lanczos,1=CG) [itype] : 0



######################################################################
```

REFERENCES

Abramowitz, M. and Stegun, I. (eds.), 1964, *Handbook of Mathematical Functions*, Natl. Bur. Stand., Appl. Math. Ser., no. 55, U.S. Government Printing Office, Washington, D.C.

Bender, C. M., and Orszag, S. A., 1978, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York.

Biedenharn, L. C., and Louck, J. D., 1981, *Angular Momentum in Quantum Physics*, Addison-Wesley, Reading, Massachusetts.

Bruno, G. V., 1973, Application of the Stochastic Liouville Method in Calculating ESR Line Shapes in the Slow Tumbling Region and an ESR-ELDOR Study of Exchange, Ph.D. Thesis, Cornell University, Ithaca, New York.

Campbell, R. F., Meirovitch, E., and Freed, J. H., 1979, Slow-motional NMR line shapes for very anisotropic rotational diffusion, Phosphorus-31 NMR of phospholipids, *J. Phys. Chem.* 83:525-533.

Campbell, R. F., and Freed, J. H., 1980, Slow-motional ESR spectra for vanadyl complexes and their model dependence, *J. Phys. Chem.* 84:2668-2680.

Choudhury, D., and Horn, R. A., 1986, The Analog of the Gram-Schmidt Algorithm for Complex Bilinear Forms and Diagonalization of Complex Symmetric Matrices, Technical Report No. 454, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, Maryland.

Cody, W. J., 1983, Algorithm 597: Sequence of modified Bessel function of the first kind, *ACM Trans. Math. Soft.* 9:242-245.

Crepeau, R. H., Rananavare, S., and Freed, J. H., 1987, Automated Least-Squares Fitting of Slow Motional ESR Spectra, Abstracts of the 10th International EPR Symposium, Rocky Mountain Conference, Denver, CO.

Cullum, J. K., and Willoughby, R. A., 1985, *Lanczos Algorithms for Large Sparse Eigenvalue Computations*, Vols. I and II, Birkhäuser, Basel.

Dammers, A. J., 1985, Numerical Simulation of Electron Spin Resonance Spectra in the Slow Motion Regime, Ph.D. Thesis, Utrecht, The Netherlands.

Dijkstra, D., 1977, A continued fraction expansion for a generalization of Dawson's integral, *Math. Comp.*, 31:503-510.

Dzuba, S. A., Maryaso, A. G., Salikhov, K. M., and Tsvetkov, Yu. D., 1984, Superslow rotations of nitroxide radicals studied by pulse EPR spectroscopy, *J. Magn. Reson.* 58:95-117.

Edmonds, A. R., 1957, *Angular Momentum in Quantum Mechanics*, Princeton University Press, Princeton, New Jersey.

Fano, U., 1957, Description of states in quantum mechanics by density matrix and operator techniques, *Rev. Mod. Phys.* 29:74-93.

Freed, J. H., Bruno, G. V., and Polnaszek, C. F., 1971a, Electron spin resonance lineshapes and saturation in the slow motional regime, *J. Phys. Chem.* 75:3385-3399.

Freed, J. H., Bruno, G. V., and Polnaszek, C. F., 1971b, ESR line shapes for triplets undergoing slow rotational reorientation, *J. Chem. Phys.* 55:5270-5281.

Freed, J. H., 1976, Ch. 3, Theory of Slow Tumbling ESR Spectra for Nitroxides, in *Spin Labeling: Theory and Applications* (L. Berliner, ed.), Vol. I, pp. 53-132, Academic Press, New York.

Freed, J. H., 1987, Molecular Rotational Dynamics in Isotropic and Oriented Fluids Studied by ESR, in *Rotational Dynamics of Small and Macromolecules in Liquids*, Lecture Notes in Physics, Vol. 293 (T. Dorfmüller and R. Pecora, eds.), pp. 89-142, Springer-Verlag, Berlin.

Fried, B. D., and Conte, S. D., 1961, *The Plasma Dispersion Function*, Academic Press, New York.

Gantmacher, F. R., 1959, *The Theory of Matrices*, Chelsea, New York.

Goldman, S. A., Bruno, G. V., Polnaszek, C. F., and Freed, J. H., 1972, An ESR study of anisotropic rotational reorientation and slow tumbling in liquid and frozen media, *J. Chem. Phys.* 56:716-735.

Golub, G. H., and Van Loan, C., 1983, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland.

Gorcester, J., 1985, unpublished report, Cornell University, Ithaca, New York.

Gorcester, J., and Freed, J. H., 1986, Two-dimensional Fourier transform ESR spectroscopy, *J. Chem. Phys.* 85:5375-5377.

Gorcester, J., and Freed, J. H., 1988a, Two-dimensional Fourier transform ESR correlation spectroscopy, *J. Chem. Phys.* 85:4673-4693.

Gorcester, J., and Freed, J. H., 1988b, Linear prediction and projection of pure absorption line shapes in two-dimensional FTESR correlation spectroscopy, *J. Magn. Reson.* 78:291-301.

Gordon, R. G., and Messenger, T., 1972, Ch. 13, Magnetic Resonance Line Shapes in Slowly Tumbling Molecules, in *Electron-Spin Relaxation in Liquids* (L. T. Muus and P. W. Atkins, eds.), pp. 341-381, Plenum Press, New York.

Hestenes, M. R., and Stiefel, E., 1952, Methods of conjugate gradients for solving linear systems, *J. Natl. Bur. Stand.* 49:409-436.

Horn, R. A., and Johnson, C. A., 1985, *Matrix Analysis*, Cambridge University Press.

Hornak, J. P., and Freed, J. H., 1983, ELDOR spin echoes and slow motions, *Chem. Phys. Lett.* 101:115-119.

Householder, A. S., 1964, *The Theory of Matrices in Numerical Analysis*, Blaisdell Publ. Co., New York (reprinted by Dover Publ., Inc. 1975).

Jeener, J., 1982, On operators, superoperators, Hamiltonians and Liouvillians, *Int. J. Quantum. Chem.*, Symp. 16, 485-560.

Kar, L., Millhauser, G. L., and Freed, J. H., 1984, Detection of slow motions in oriented lipid multilayers by two-dimensional electron-spin-echo spectroscopy, *J. Phys. Chem.* 88:3951-3956.

Kar, L., Ney-Igner, E., and Freed, J. H., 1985, Electron spin resonance and electron-spin-echo study of oriented multilayers of *Lα*-dipalmitoylphosphatidylcholine water systems, *Biophys. J.* 48:569-595.

Kar, L., Johnson, M. E., and Bowman, M. K., 1987, Electron spin-echo techniques for the study of protein motion, *J. Magn. Reson.* 75:397-413.

Kivelson, D., 1972, Ch. 10, Electron Spin Resonance in Liquids. Selected Topics, in *Electron-Spin Relaxation in Liquids* (L. T. Muus and P. W. Atkins, eds.), pp. 213-277, Plenum Press, New York.

Korst, N. N., and Antsiferova, L. I., 1979, Study of slow molecular motions by stable-radical EPR, *Sov. Phys. Usp.* 21:761-778.

Lajzerowicz-Bonneteau, J., 1976, Ch. 6, Molecular Structure of Nitroxides, in *Spin Labeling: Theory and Applications* (L. Berliner, ed.), Vol. I, pp. 239-249, Academic Press, New York.

Lanczos, C., 1950, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J. Natl. Bur. Stand.* 45:255-282.

Löwdin, P. O., 1982, On operators, superoperators, Hamiltonians and Liouvillians, *Int. J. Quantum Chem.*, Symp. 16:485-560.

Löwdin, P. O., 1985, Some aspects on the Hamiltonian and Liouvillian formalism, the special propagator methods, and the equation of motion approach, *Adv. Quantum Chem.* 17:285-334.

Mason, R. P., Polnaszek, C. F., and Freed, J. H., 1974, Comments on the interpretation of electron spin resonance spectra of spin labels undergoing very anisotropic reorientation, *J. Phys. Chem.* 78: 1324-1329.

McLachlan, N. W., 1961, *Bessel Functions for Engineers*, 2nd edn., Oxford University Press, New York.

Meirovitch, E., and Freed, J. H., 1979, Slow motional NMR lineshapes for very anisotropic diffusion: $I = 1$ nuclei, *Chem. Phys. Lett.* 64:311-316.

Meirovitch, E., Igner, D., Igner, E., Moro, G., and Freed, J. H., 1982, Electron-spin relaxation and ordering in smectic and supercooled nematic liquid crystals, *J. Chem. Phys.* 77:3915-3938.

Meirovitch, E., 1983, Electron spin resonance observations on the annealing process of low-water-content dipalmitoylphosphatidylcholine bilayers, *J. Phys. Chem.* 87:845-850.

Meirovitch, E., and Freed, J. H., 1984, Analysis of slow-motional electron spin resonance spectra in smectic phases in terms of molecular configuration, intermolecular interactions, and dynamics, *J. Phys. Chem.* 88:4995-5004.

Meirovitch, E., Nayeem, A., and Freed, J. H., 1984, Analysis of protein-lipid interactions based on model simulations of electron spin resonance spectra, *J. Phys. Chem.* 88:3454-3465.

Messiah, A., 1962, *Quantum Mechanics*, Wiley, New York.

Millhauser, G. L., and Freed, J. H., 1984, Two-dimensional electron spin echo spectroscopy and slow motions, *J. Chem. Phys.* 81:37-48.

Millhauser, G. L., Gorcester, J., and Freed, J. H., 1987, New Time-Domain ESR Methods for the Study of Slow Motions on Surfaces, in *Electron Magnetic Resonance of the Solid State* (J. A. Weil, ed.) pp. 571-597, Canadian Chemical Society, Ottowa, Canada.

Moro, G., 1980, EPRLF and EPRLP Routines, unpublished report, Cornell University, Ithaca, New York.

Moro, G., 1980, Implementation of the Lanczos Algorithm in the Calculation of Spectral Functions, unpublished report, Cornell University, Ithaca, New York.

Moro, G., and Freed, J. H., 1981, Calculation of ESR spectra and related Fokker-Planck forms by the use of the Lanczos algorithm, *J. Chem. Phys.* 74:3757-3773.

Moro, G., and Freed, J. H., 1986, The Lanczos Algorithm in Molecular Dynamics: Calculation of Spectral Densities, in *Large-Scale Eigenvalue Problems, Mathematical Studies Ser.*, Vol 127 (J. Cullum and R. Willoughby, eds.), North-Holland, Amsterdam.

Nordío, P. L., 1976, Ch. 2, General Magnetic Resonance Theory, in *Spin Labeling: Theory and Applications* (L. J. Berliner, ed.), Vol. I, pp. 5-52, Academic Press, New York.

Paige, C. C., 1976, Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix, *J. Inst. Math. Appl.* 18:341-349.

Paige, C. C., 1980, Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem, *Linear Algebra & Appl.* 34:235-258.

Parlett, B. N., 1980, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, N.J.

Parlett, B. N., and Nour-Omid, B., 1985, The use of a refined error bound when updating eigenvalues of tridiagonals, *Linear Algebra & Appl.* 68:179-219.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, *Numerical Recipes*, Cambridge University Press, New York.

Rose, M. E., 1957, *Elementary Theory of Angular Momentum*, Wiley, New York.

Schneider, D. J., 1989, Molecular Dynamics and Spin Relaxation: Theory, Computational Methodologies and Experimental Applications, Ph.D. Thesis, Cornell University, Ithaca, New York.

Schneider, D. J., and Freed, J. H., 1989, Spin relaxation and motional dynamics, *Adv. Chem. Phys.* 73:387-527.

Schwartz, L. J., Stillman, A. E., and Freed, J. H., 1982, Analysis of electron spin echoes by spectral representation of the stochastic Liouville equation, *J. Chem. Phys.* 77:5410-5425.

Schwartz, L. J., Millhauser, G. L., and Freed, J. H., 1986, Two-dimensional electron spin echoes: Magnetization transfer and molecular dynamics, *Chem. Phys. Lett.* 127:60-66.

Sookne, D. J., 1973a, Bessel functions *I* and *J* of complex argument and integer order, *J. Res. Natl. Bur. Stand.* 77B:111-124.

Sookne, D. J., 1973b, Bessel functions of real argument and integer order, *J. Res. Natl. Bur. Stand.* 77B:125-132.

Subczynski, W. K., Antholine, W. E., Hyde, J. S., and Petering, D. H., 1987, Orientation and mobility of a copper square-planar complex in a lipid bilayer, *J. Am. Chem. Soc.* 109:46-52.

Tanaka, H., and Freed, J. H., 1984, Electron spin resonance studies on ordering and rotational diffusion in oriented phosphatydlcholine multilayers: Evidence for a new chain-ordering transition, *J. Phys. Chem.* 88:6633-6644.

Tanaka, H., and Freed, J. H., 1985, Electron spin resonance studies of lipid-gramicidin interactions utilizing oriented multilayers, *J. Phys. Chem.* 89:350-360.