Available online at www.sciencedirect.com
ScienceDirect
Journal of Magnetic Resonance 183 (2006) 152–159
JMR Journal of Magnetic Resonance
www.elsevier.com/locate/jmr

Communication

Nonlinear-least-squares analysis of slow motional regime EPR spectra ☆

Khaled Khairy [a,b,*], David Budil [a], Piotr Fajer [b]

[a] Department of Chemistry and Chemical Biology, Northeastern University, Boston, MA 02115, USA
[b] National High Magnetic Field Laboratory and Inst. Molecular Biophysics, Tallahassee, FL 32310, USA

Received 27 March 2006; revised 26 July 2006
Available online 24 August 2006

Abstract

A comparison between the full Newton-type optimization NL2SNO, the Levenberg–Marquardt method with the model-trust region modification, and the simplex algorithm is made in the context of the iterative fitting of EPR spectra. EPR lineshape simulations are based on the stochastic Liouville equation (SLE), with an anisotropic diffusion tensor and an anisotropic restraining potential describing the motional amplitude of the spin label. The simplex algorithm was found to be the most reliable, and an approach—incorporating both NL2SNO as well as the downhill simplex methods—is proposed as a strategy-of-choice.
© 2006 Elsevier Inc. All rights reserved.

Keywords: EPR; Spectrum fitting; Nonlinear optimization; Levenberg–Marquardt; Simplex

1. Introduction

The emergence of applications in the study of biological membranes and structural biology, for both NMR and EPR, have significantly increased the demand for robust, accurate, and efficient programs for the analysis of experimental data. A suitable minimization scheme can be used to elicit both structural and dynamic information from the spectroscopic lineshapes. Iterative analysis of spectra is based on: (a) a suitable physical model to simulate the theoretical spectrum, (b) an algorithm that improves a given set of model parameters, based on a suitable criterion such as minimizing χ^2 [1], and (c) a measure for the goodness of fit.

In this report, the simulation of EPR spectra involves the solution of the stochastic Liouville equation (SLE) of motion [2]. The SLE is of special importance in the slow-motion regime of the spin label and provides a versatile means for implementing detailed dynamical models that include fully anisotropic rates and amplitudes of motion [3]. SLE-based simulations are CPU-intensive, and—in all but the simplest cases—remain a time-consuming step in the analysis of experimental spectra. This is especially true at high magnetic fields [4,5], which necessitates finding an efficient and robust strategy that reduces the number of iterations needed for an acceptable fit.

Some of the more well-known parameter optimization algorithms are the downhill simplex [6], Powell [7], evolutionary Monte Carlo [8], and conjugate gradients methods [9], as well as Newton-type methods such as Gauss–Newton [1], Levenberg–Marquardt [10,11], and Levenberg–Marquardt with the model-trust-region modification [12]. A number of these have been applied to various curve fitting problems involving EPR spectra: simplex in powder patterns and orientational distribution of rigid samples [13,14], Levenberg–Marquardt for dynamically averaged line shapes [15], and simulations in the slow motional regime [16]. At the end of an optimization it is desirable to be able to test for the goodness of fit, a number of such tests is available, for example the reduced $\chi^2 = \chi^2/(n-p)$,

☆ This work was supported by National Science Foundation Grants MCB960094 (D.E.B.) and MCB0346650 (P.G.F.), and an IHRP grant from the National High Magnetic Field Laboratory (P.G.F.).
* Corresponding author. Present address: Max Planck Institute of Molecular Cell Biology and Genetics, Dresden 01307, Germany. Fax: +49 351 2102020.
E-mail address: khairy@mpi-cbg.de (K. Khairy).

1090-7807/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmr.2006.07.022

with n the number of data points and p the number of fitted parameters, which for a fit in which the only deviation between model and data is Gaussian noise, is distributed about unity with a standard deviation of $\sqrt{2/(n-p)}$.

Newton-type methods, which are the most popular in EPR spectral analysis, necessitate the calculation of first (and approximations of second) derivatives of χ^2 with respect to fitting parameters. Inaccuracies in the calculation of derivatives may lead to inefficient convergence requiring a large number of individual spectral simulations, as well as inaccuracies in the estimate of parameter uncertainties. SLE-based simulations are particularly prone to such inaccuracies when an insufficient basis set is used (see below) and/or in the presence of low signal to noise. Although increasing the step size in the calculation of derivatives may partially solve this problem, it is desirable to find a robust algorithm that is insensitive towards it.

Moreover, existing interactive programs for EPR spectral analysis provide little control over the detailed configuration of the optimization algorithms themselves, for example the step-size taken for (or the method of) calculating derivatives numerically, or the values for various converge criteria. We observe that a user-driven balance between control and automation of both spectral simulation and optimization would be invaluable to the community.

In this work, we introduced the adaptive full-Newton type algorithm NL2SNO [17] to EPR line-shape analysis and compared it to the model trust region modification of the Levenberg–Marquardt method (MTRMLM) and the downhill simplex algorithm (SIMPLEX). The latter can be categorized as a direct method since it does not depend on the calculation of derivatives. In addition, the SLE-based simulation, NL2SNO and SIMPLEX have been implemented in a unified computer program with an easy-to-use interface that runs on a conventional PC-workstation and is made available to the community.

We found that NL2SNO offers significant advantages over MTRMLM in terms of both efficiency and accuracy of parameter estimates. Surprisingly, SIMPLEX is significantly more efficient in locating a global minimum than either of the above Newton-type methods. However, SIMPLEX does not allow for a precise estimation of solution uncertainties in a straight-forward way (commonly a bootstrap method is used). We advocate a combined method in which SIMPLEX is used to identify the general region of the solution, and NL2SNO is used to refine parameter estimates and calculate fitting statistics.

2. Methods

2.1. EPR-lineshape simulation

The physical model for the EPR simulations we used here is based on the work of Freed and co-workers [3,18]. The model spans a wide range of motional correlation times for the spin label, and can also account for the label's microscopic order but macroscopic disorder (MOMD). The MOMD model assumes microscopic molecular ordering with respect to a local director that is itself randomly oriented. This model is particularly useful for biological samples where motion occurs in a locally ordered environment but the sample is macroscopically disordered, e.g., membrane dispersions, protein surfaces, or DNA [19]. For a detailed description of the stochastic Liouville calculation we refer the reader to Schneider and Freed [20].

The SLE used to carry out a slow-motional calculation is constructed in a basis set consisting of spin functions that represent the magnetic resonance properties, and generalized spherical harmonic functions used to represent the rotational diffusion of the probe molecule. The SLE basis functions are specified by the quantum numbers L, K, and M. The slow-motion EPR calculation utilizes truncation parameters, L_{emx}, L_{omx}, K_{mn}, K_{mx}, M_{mn}, and M_{mx}, that determine the maximum even L, maximum odd L, minimum K, maximum K, minimum M, and maximum M allowed, respectively [4]. When these parameters are set to the minimum necessary for acceptable convergence of the calculated spectrum, they are referred to as a minimum truncation set (MTS). Simulated EPR lineshapes that include molecular motion are thus approximations, since they use a finite number of component functions.

Since the CPU-time needed for a simulation increases with the size of the basis set, it is desirable to find the MTS [4,21]. This has implications when performing optimizations with Newton-type methods since they depend on the evaluation of first and second order partial derivatives of the simulated model spectra w.r.t. fitting parameters (see below), both of which are sensitive to the details of the lineshape. An insufficient basis set or a restricted number of orientations in the MOMD model causes oscillations (ringing) in the high field portion of the EPR spectrum (Fig. 1). There is no fixed number for the minimum size of a basis set, the proper size increases with slower motions of the spin label. This is important in the case of fitting for motional parameters, since the optimization algorithm may drive their values to a region where the basis set is insufficient, introducing ringing.

2.2. Newton-type optimization

MTRMLM and NL2SNO fall under the broad category of Newton-type optimization methods. The optimization problem is to minimize χ^2 the sum of squared residuals between the simulated $\boldsymbol{f}(\boldsymbol{x})$ and the experimental $\boldsymbol{h}$ spectra containing N data points, for a given parameter vector $\boldsymbol{x}$ of length M. In case of the Newton-type methods of nonlinear least squares, the current estimate $\boldsymbol{x}_0$ is taken and a refined estimate $\boldsymbol{x}_+$ is calculated according to the formula [22,23],

$$\boldsymbol{x}_+ = \boldsymbol{x}_0 + \boldsymbol{\alpha}^{-1}(\boldsymbol{x}_0) \cdot \boldsymbol{\beta}(\boldsymbol{x}_0), \tag{1}$$

where the ith component of $\boldsymbol{\beta}$ is given by,

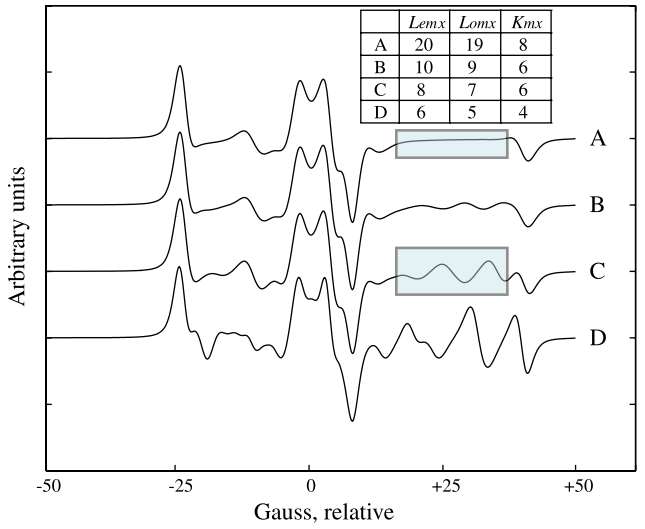

Fig. 1. Oscillations due to an insufficient choice of basis set. The simulations shown are based on the SLE calculation at X-band at a center field of 3500 Gauss and a sweep width of 100 Gauss, by applying increasingly severe truncation set restrictions detailed in the table (A–D). For symbol meanings see Table 1. The insufficient-basis-set pathology can already be observed in (B) and gets more severe as the basis set is increasingly restricted as shown in (C and D) (compare the spectrum section in the gray box (A) to that in (C)). Other parameters are $g_x = 2.009$, $g_y = 2.005$, $g_z = 2.002$, $a_x = a_y = 7.5$ Gauss, $a_z = 35.5$ Gauss, $R_{\text{iso}} = 6.1$, and line-width broadening = 0.6 Gauss, K_{mn}, M_{mx}, and M_{mn} were 0, 2, and 0, respectively.

$$\beta_i(\boldsymbol{x}_0) = -\frac{1}{2}\frac{\partial\chi^2(\boldsymbol{x}_0)}{\partial x_i} = \sum\nolimits_{k=1}^{N}(h_k - f_k(\boldsymbol{x}_0))\frac{\partial f_k(\boldsymbol{x}_0)}{\partial x_i} \tag{2}$$

with $i = 1, 2, \ldots, M$, and assuming unit standard deviation for all measured data points. $\boldsymbol{\alpha}$, which is one half of the Hessian matrix, has components $\boldsymbol{\alpha}_{ij}$ given by,

$$\begin{aligned}\boldsymbol{\alpha}_{ij}(\boldsymbol{x}_0) &= \frac{1}{2}\frac{\partial^2\chi^2(\boldsymbol{x}_0)}{\partial x_i \partial x_j} \\ &= \sum\nolimits_{k=1}^{N}\frac{\partial f_k(\boldsymbol{x}_0)}{\partial x_i}\frac{\partial f_k(\boldsymbol{x}_0)}{\partial x_j} - (h_k - f_k(\boldsymbol{x}_0)) \\ &\quad \times \frac{\partial^2 f_k(\boldsymbol{x}_0)}{\partial x_i \partial x_j}\end{aligned} \tag{3}$$

so $\boldsymbol{\alpha}$ can be written as a sum of two terms

$$\boldsymbol{\alpha}(\boldsymbol{x}_0) = \boldsymbol{J}(\boldsymbol{x}_0)\cdot\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_0) + \boldsymbol{S}(\boldsymbol{x}_0), \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$, and $\boldsymbol{S}(\boldsymbol{x})$ a matrix that contains all second derivative information. The iterative solution of Eq. (1) leads to better estimates $\boldsymbol{x}$ until a set of parameters ($\boldsymbol{x}^*$) is found that minimizes χ^2 according to some termination criterion (see below).

In all but the simplest cases, calculating $\boldsymbol{S}(\boldsymbol{x})$ is computationally impractical. Newton-type algorithms differ in their method of treating/approximating $\boldsymbol{S}(\boldsymbol{x})$. For instance, in the Gauss–Newton method, $\boldsymbol{S}(\boldsymbol{x})$ is ignored altogether. In Levenberg–Marquardt and MTRMLM, $\boldsymbol{S}(\boldsymbol{x})$is replaced by a diagonal linear scaling matrix ($\lambda\cdot\mathrm{diag}(\boldsymbol{J}(\boldsymbol{x}_0)\cdot\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{x}_0))$), which makes the algorithm a compromise between Gauss–Newton and steepest descent because the magnitude of the scaling parameter λ determines the diagonal dominance of $\boldsymbol{\alpha}$. In such cases $\boldsymbol{\alpha}$ depends on first derivative information only. This is equivalent to assuming a random distribution of errors which renders the contribution of second derivatives negligible. The success of the Gauss–Newton and Levenberg–Marquardt methods would clearly depend on the importance of $\boldsymbol{S}(\boldsymbol{x})$.

In our new program we chose to implement the Newton-type algorithm NL2SNO [17]. It adaptively decides on the form of the approximation to the Hessian: Gauss–Newton-like, MTRMLM-like, or by using a so-called secant approximation of $\boldsymbol{S}(\boldsymbol{x})$. Secant methods are those that build increasingly better approximations of derivative information based on the history of iterations. For the detailed formula of the secant Hessian approximation in NL2SNO the reader is referred to the original publication [17]. The procedure for switching models is based on the comparison of the actual reduction in function value (regardless which model was used) to the reduction predicted by each individual model. The model with the best match in function reduction prediction is used in the next step. This adaptive modeling causes NL2SNO to use Gauss–Newton or MTRMLM steps in the beginning until it builds up enough second-order information (i.e., better approximations of $\boldsymbol{S}(\boldsymbol{x})$), after which the secant Hessian approximation can be used.

Given $g(\boldsymbol{x})$ as the function to be minimized, convergence criteria in the implementation for NL2SNO were: (1) x-convergence; occurs when the current iterate $\boldsymbol{x}_0$ is within a prescribed tolerance ε_x of a strong local minimizer $\boldsymbol{x}^*$ (i.e., close to the solution which of course may still represent a local minimum), in which case the following criteria are satisfied: (a) the scaled relative difference ($\mathrm{SRD} = \max(|\boldsymbol{x}_0 - \boldsymbol{x}_-|)/\max(|\boldsymbol{x}_0| + |\boldsymbol{x}_-|)$) between the current scaled step $\boldsymbol{x}_0$ and the previous one $\boldsymbol{x}_-$ is smaller than ε_x, (b) the current step yields no more than twice the predicted function decrease, and (c) the Hessian is positive definite, (2) relative function convergence; occurs if the current function value $g(\boldsymbol{x})$ is close to its value $g(\boldsymbol{x}^*)$ at a strong local minimizer $\boldsymbol{x}^*$, in that case (a) the Hessian is positive definite, (b) the current step yields no more than twice the predicted function decrease, and (c) $g(x_0) - g(\boldsymbol{x}^*) < \varepsilon_R g(x_0)$, where ε_R is the relative function convergence tolerance, (3) absolute function convergence; occurs if the condition $g(\boldsymbol{x}_0) < \varepsilon_A$ is satisfied, where ε_A is the absolute function convergence tolerance, this is for the rare case when $\boldsymbol{x}^*$ is the zero vector and $g(\boldsymbol{x}^*) = 0$, since x-convergence and relative function convergence tests do not work in this case, (4) singular convergence; occurs when the least-squares Hessian is singular or nearly so, which signifies that the model is over-specified, (5) false convergence (convergence to a non-critical point); is detected when (a) none of the previous convergence criteria is satisfied, (b) the algorithm is detecting convergence in the sense that the SRD is smaller than a prescribed tolerance ε_F, (this is similar to x-convergence, but ε_F should always be smaller that ε_x), and (c) the current step yields more than twice the predicted function decrease. For a more

detailed explanation of convergence criteria the reader is referred to [17].

For the implementation of MTRMLM the convergence criteria were: (1) relative function convergence (as above), (2) x-tolerance (as above) and (3) a convergence test for the gradient of $g(\boldsymbol{x}_0)$ w.r.t. the fitting parameters, with its respective tolerance ε_G.

2.3. Downhill simplex

SIMPLEX (also called amoeba) due to Nelder and Mead [6] is also iterative. It does not include an implicit model for the derivatives, and so only depends on function values. This makes it particularly suitable for cases where the calculation of the derivatives might be influenced by noise or other spectral pathologies such as an insufficient basis set or, as can be the case with MOMD calculations, an insufficient number of orientations.

The coordinates of each vertex of the simplex are the fitting parameters, the value associated with an individual vertex is χ^2 corresponding to the parameter set defining the coordinates. The algorithm proceeds by attempting to move the vertices of the simplex in parameter space into a minimum of χ^2. At each step it either reflects the vertex associated with the highest χ^2 value through the $(n-1)$-dimensional plane defined by the remaining points (n being the number of simplex vertices), reflects and expands to be able to take larger steps, contracts to shrink the overall volume after reaching a valley floor, or shrinks (multiple contractions). Four parameters, that control the respective behaviours of the simplex, must be specified to define a complete Nelder–Mead method, $\alpha > 1$, $\beta > 1$, $0 < \gamma < 1$, and $0 < \sigma < 1$, where $\beta > \alpha$. For instance a reflection point is calculated as $\boldsymbol{x}_r = \bar{\boldsymbol{x}} + \alpha(\bar{\boldsymbol{x}} - \boldsymbol{x}_{\mathbf{Hi}})$, where $\boldsymbol{x}_{\mathbf{Hi}}$ is the vertex associated with the highest $\boldsymbol{x}$, and $\bar{\boldsymbol{x}}$ is the centroid of the other points. The other parameters enter the algorithm in a similar way to scale the step sizes taken by the corresponding action of the simplex. The behavior of the simplex during the course of a minimization is that of an amoeba crawling around parameter space, creeping down valleys and shrinking to get to the very bottom of narrow valleys or through the eye of a needle.

The termination criterion of the original algorithm was based on a relative tolerance computed as the fractional range from highest to lowest function values associated with the vertices of the simplex. As in Fajer et al. [13], we have extended the criteria to include an absolute χ^2 tolerance (equivalent to the absolute function convergence tolerance above), a maximum number of iterations and a minimum step-size for each parameter to make sure that the steps are still within the resolution of the search.

2.4. WinMOMD

As mentioned above, we have incorporated the SLE-spectral simulation program of Freed and coworkers together with the optimization algorithms NL2SNO and SIMPLEX in the unified user-friendly program called WinMOMD that runs on a conventional PC-workstation. The simulation of a single spectrum is based on the original FORTRAN code at double precision, and has been wrapped with C++ for calculating MOMD spectra and for communication with the C++-based interface. The SIMPLEX algorithm has been programmed in C and NL2SNO in FORTRAN, both at double precision. All code was compiled using Microsoft Visual C++ (and Digital FORTRAN). WinMOMD allows control over all settings of the spectral simulation, and optimization engines. The user may choose between SIMPLEX or NL2SNO. For automation, a Monte Carlo shell has been added that enables performing a large number of minimizations in succession with starting values chosen randomly within parameter-prespecified bounds. A download URL for WinMOMD is http://www.chem.neu.edu/d.budil/webfolder/WinMOMD.htm or http://fajerpc.magnet.fsu.edu/Programs/WinMOMD/WinMOMD.html.

3. Results

The efficiencies of MTRMLM, NL2SNO, and SIMPLEX were compared on simulated test spectra generated with different magnitudes and types of noise. The following implementations were used:

(1) *NLSL*: It is an implementation of MTRMLM with the addition of a separation of variables [4].
(2) *NLSL(s)*: The same as *NLSL* but with the spectral shifting option enabled. This feature includes the trace of the g-tensor as an additional separable parameter to be optimized. Shifting is useful in cases where either the spectrometer frequency or the absolute magnetic field are not known to sufficient accuracy. We include it as a separate method to test whether shifting has detrimental effects on the efficiency and accuracy of the spectral analysis under conditions of low S/N.
(3) *NL2SNO*: It is an implementation of NL2SNO [17,22].
(4) *SIMPLEX*: It is based on SIMPLEX [6]. Our implementation is modified for termination criteria as described above [13].

3.1. Test cases

Four simulated test cases are given in Fig. 2. The spectra, which were normalized to the double integral, were simulated with the rotational diffusion rates, $R_\perp$ and $R_\parallel$ (perpendicular and parallel to the principal axes of diffusion of the probe) with the values $\log_{10}(R_\perp) = 7.28$ and $\log_{10}(R_\parallel) = 8.13$. In case 1, normally distributed noise was added to the simulated spectrum to obtain a signal to noise ratio (S/N) of 27. In case 2, uniformly distributed noise was added, and in case 3, Gaussian noise with $S/N = 6$. Case 4 was simulated as case 1 above except that before addition

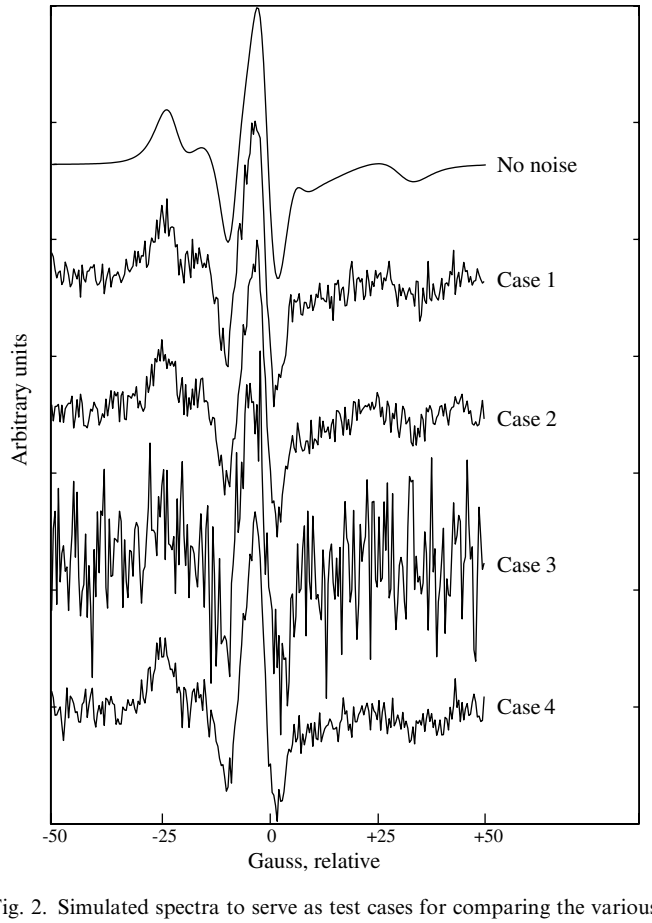

Fig. 2. Simulated spectra to serve as test cases for comparing the various optimization algorithms. The original spectrum was simulated at X-band (at a center field of 3400 Gauss and a sweep width of 100 Gauss) with $\log_{10}(R_\perp)$ and $\log_{10}(R_\parallel)$ of 7.28 and 8.13, respectively. Case 1: normally distributed noise with $S/N = 27$. Case 2: uniformly distributed noise with $S/N = 27$. Case 3: normally distributed noise with $S/N = 6$. Case 4: the model does not fit the data anymore. It uses a different physical model than that in the simulations. All other values are given in Table 1.

of noise the spectrum was artificially stretched horizontally by 2%. Thus case 4 does not fit the model anymore. This was done to test the optimization methods in the large residual case. NL2SNO was specifically developed to meet the need for a nonlinear least-squares algorithm which would be reliable in the presence of large residuals, at the minimum, but would still be more efficient than the variable metric methods such as Davidon–Fletcher–Powell which are intended for general function minimization.

The spectral parameters which were kept constant during fitting are shown in Table 1. Here, g_x, g_y, and g_z are the Cartesian components of the g-tensor for the electronic Zeeman interaction, a_x, a_y, and a_z are the components of the electron/nuclear hyperfine tensor in Gauss. w_x, w_y, and w_z are principal values of the orientation-dependent Lorentzian inhomogeneous line broadening tensor, gib_0 specifies the inhomogeneous (Gaussian) broadening that is added to the spectrum after the calculation of the slow-motional lineshape.

All values for convergence tolerances used for our minimizations using *NLSL*, *NL2SNO*, and *SIMPLEX* are listed in Table 2. Convergence tolerances for *NLSL* and *NLSL(s)* were identical.

Other than termination criteria, the *SIMPLEX* settings were kept at their nearly universal choices $\alpha = 1$, $\beta = 2$, and $\gamma = \sigma = 0.5$ [23]. *NL2SNO* had not been used for this type of calculation before, we searched for reasonable settings for forward difference step size in the Jacobian calculation (*DLTFDJ*) and suitable convergence tolerances, that minimized the number of function evaluations needed to reach a minimum without encountering false or singular convergence. *NL2SNO* has many settings that control the behaviour of the algorithm, all of which were kept at their default values except for *DLTFDJ* which was set to 5×10^{-3}. Both the *NLSL* and *NLSL(s)* implementations were introduced in Budil et al. [4]. NL2SNO and SIMPLEX were implemented in WinMOMD as stated above.

Random starting guesses for all tests were generated by a uniform sampling within prescribed parameter intervals for $\log_{10}(R_\perp)$ and $\log_{10}(R_\parallel)$; 6(slow motion) $< \log_{10}(R) <$ 9(fast motion). For consistency, the same starting parameters were used for all the Newton-based calculations.

Figs. 3–5 show our results when comparing the four algorithms for the cases considered in Fig. 2. A total of 100 optimizations, each starting with random guesses, were performed on each test case. Fig. 3 shows the number of optimization runs that gave χ^2 values within 5% of the lowest χ^2. This is a test for how consistent the optimization algorithms are in finding close minima given a range of starting values. For the Newton algorithms, none of the fits within those 5% terminated with false convergence (for NL2SNO) or absolute function convergence, which is further assurance that we have found suitable settings for *NL2SNO*, *NLSL*, and *NLSL(s)*. *SIMPLEX* has shown a clear superiority in its ability to find the same (low) minimum repeatedly. On average 94% of *SIMPLEX* minimizations resulted in a correct answer as compared to 62% for *NL2SNO* and 52% for *NLSL(s)* and *NLSL*. Importantly, the trend was independent of the noise level; case 3 has 4-times more noise yet the efficiency of finding a minimum was the same as for case 1.

The real price of any fitting method is the CPU-time needed to perform a successful optimization. In the EPR case the slowest step is the lineshape simulation, thus the number of simulations (function calls to the simulation routine) gives an accurate estimate of the relative length of optimization. Fig. 4 shows a comparison of the number of function calls that are needed to reach a (any) minimum. Here, we see that both *SIMPLEX* and *NL2SNO* outperform *NLSL(s)* and *NLSL*, with *SIMPLEX* consistently being the fastest. Thirty-three spectral simulations were needed on average for *SIMPLEX*, 42 for *NL2SNO* compared to 93 for *NLSL*. This is a factor of 3 in the number of simulations, which shows a significant practical gain when using *SIMPLEX*. For NLSL(s) the extremely low S/N of case 3 appears to be detrimental to its performance, compared to cases 1 and 2. The large residual case (case 4), shows that NL2SNO is indeed more robust, compared to

Table 1
Non-fitting variables for the simulation

g-tensor				
g_x	2.009	Max. even L quantum number	L_{emx}	10
g_y	2.006	Max. odd L quantum number	L_{omx}	9
g_z	2.002	Min. K quantum number in basis set	K_{mn}	0
Hyperfine tensor				
a_x	5	Max. K quantum number in basis set	K_{mx}	8
a_y	5.5	Min. M quantum number in basis set	M_{mn}	0
a_z	33	Max. M quantum number in basis set	M_{mx}	2
Gaussian inhomogenous broadening				
gib_0	1.0	Lorenzian linewidth broadening	w_x,w_y,w_z	0, 0, 0

Table 2
Termination criteria and tolerances

NLSL and NLSL(s)		NL2SNO		SIMPLEX	
ε_x	1.0×10^{-4}	ε_x	5.3×10^{-4}	ε_A	1.0×10^{-10}
ε_A	1.0×10^{-4}	ε_A	1.0×10^{-10}	ε_{HL}[c]	100
ε_G	1.0×10^{-6}	ε_R	1.0×10^{-6}	—	—
—	—	ε_F	2.8×10^{-15}	—	—
MXFUN[a]	300	MXFUN[a]	300	MXFUN[a]	200
MXITER[b]	150	MXITER[b]	150	—	—

[a] Maximum number of function evaluations.
[b] Maximum number of fitting iterations.
[c] Fraction of parameter step size below which iterations terminate.

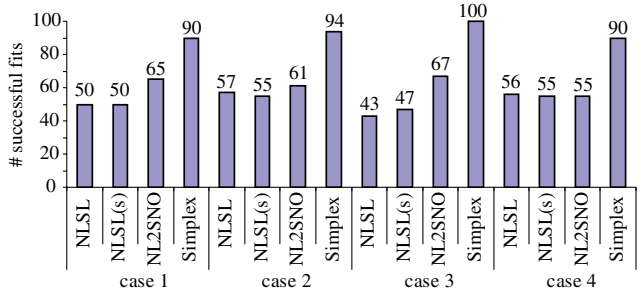

Fig. 3. Comparison of the four algorithms for the cases shown in Fig. 1 in terms of the number of successful fits. One hundred fits were performed for each implementation for each case (a total of 16). A fit was regarded as successful when the values of χ^2 at the minimum was within a 5% range of the smallest χ^2 value.

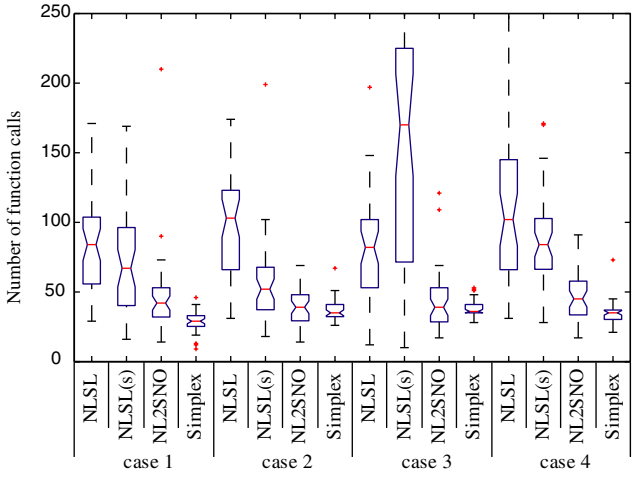

Fig. 4. Quartile plot comparison of the four algorithms for the cases given in Fig. 1, showing the number of function calls needed to reach a successful fit (as defined in Fig. 3 caption). The N values used to produce the individual quartile plots are those corresponding to Fig. 3. For each box the lower and upper lines are the 25th and 75th percentiles of the sample. The distance between top and bottom is the interquartile distance. The line in the box is the median. The whiskers (dashed lines) indicate the extent of the rest of the sample excluding outliers, which are data points more than 1.5 times the interquartile range away from the top or bottom of the box. The notches in the box are confidence intervals (95%) about the median.

NLSL and NLSL(s), when using a model that is not correct. SIMPLEX appears to also be unaffected by slight inconsistencies in the model.

The mark of a successful optimization is its accuracy. Fig. 5 shows the relative accuracy of the different algorithms in the form of $\frac{(\chi^2-\chi^2_{true})}{\chi^2_{true}}\times 100$, the deviation of the minimum value found for χ^2 for each algorithm from χ^2_{true} (the χ^2 obtained by using the original values of fitting parameters). Also plotted is the deviation % of $\log_{10}(R_\perp)$ and $\log_{10}(R_\parallel)$ at that minimum, from their original values before the addition of noise. *SIMPLEX*, *NL2SNO*, and *NLSL(s)* find an almost identical minimum in all cases (for χ^2, $\log_{10}(R_\perp)$ as well as for $\log_{10}(R_\parallel)$). However, except for case 3, *NLSL* is less reliable. For the same number of attempts as the other three methods, *NLSL* did not find the correct global minimum. As expected with the increase in noise, the ability of the optimization to resolve $\log_{10}(R_\parallel)$ is diminished, to the extent that in case 3 no reliable infor-

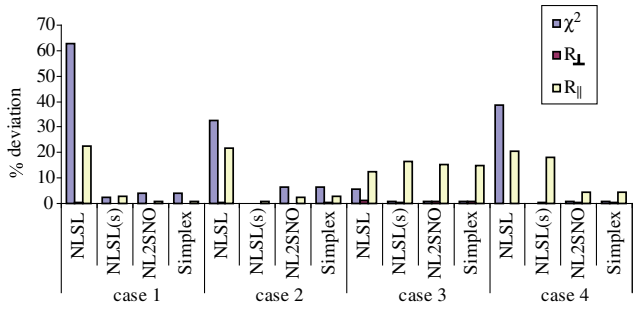

Fig. 5. Comparison of the four algorithms for the cases given in Fig. 1 in terms of the % deviation of the best (lowest χ^2) from the true minimum χ^2 value, and the average deviation % of the corresponding values of $\log_{10}(R_\perp)$ and $\log_{10}(R_\parallel)$ from their true values 7.28 and 8.13, respectively.

mation regarding $R_\parallel$ can be obtained. *NLSL* has done poorly on the determination of $R_\parallel$ in all cases. $R_\perp$, which is determined by the most prominent feature in the first derivative spectra, was successfully found in all cases.

4. Discussion

A comparison of four iterative approaches to the analysis of EPR spectra in the slow-motion regime was performed with the purpose of finding an efficient and accurate strategy. The comparison focused on efficiency, reproducibility and accuracy. As a test, four different spectra with varying *S*/*N* ratios, types of noise, and a non-random deviation were used to approximate various experimental conditions. The optimal strategy is to use SIMPLEX, to reproducibly approach a global minimum, followed by NL2SNO for refinement of the parameter fit and the calculation of the relevant statistics. Automated repetition using random starting values (Monte Carlo) is used to sample full parameter space.

Consistent with the literature, the Newton-based methods, which do well when starting close to the minimum [24], have shown a general high sensitivity towards initial values, which would explain their lower percentage of successful fits. Still, *NL2SNO* seems less sensitive to starting values than *NLSL* and *NLSL(s)*, as would be expected, since it starts out with a Gauss–Newton model, which is known to have better initial convergence.

SIMPLEX clearly outperformed both *NL2SNO* and the MTRMLM-based implementations with regards to the success rate of finding a correct solution as well as requiring fewer time-consuming spectral simulations. SIMPLEX has been used to find the orientational distribution and magnetic tensors before [13,24], however, not the more complex motionally narrowed EPR spectra.

In particular we stress that *SIMPLEX*, which is traditionally known for its robustness but not for its efficiency, has done surprisingly well on the number of function evaluations (Fig. 4) as compared to *NLSL*, *NLSL(s)*, and *NL2SNO*. This could be explained by inaccurate calculations of the Hessian for the gradient-dependent techniques, especially in the presence of noise, which lies in contrast to *SIMPLEX* where no gradient information is needed. Inaccuracies of the Hessian do not generally affect the final values of the fitted parameters, but only the iterative route that is taken to reach the minimum, hence the advantage of *SIMPLEX* in the number of function evaluations. This conclusion is consistent with the results in Fig. 5, where it is shown that *NL2SNO* as well as *SIMPLEX* reach a global minimum (albeit at different rates). Another possible source of inefficiencies in the gradient-based methods is their use of the model trust region. Eventhough the trust region assures convergence, it can significantly hinder the optimization algorithm from approaching the minimum in a small number of function evaluations because of the small the step size.

In summary, we propose a method where Monte Carlo/SIMPLEX is used to approach the global minimum quickly and NL2SNO subsequently refines the found parameter values and is used to obtain necessary statistics, such as inference regions and parameter error estimates.

References

[1] Y. Bard, Nonlinear Parameter Estimation, Academic Press, New York, 1974.
[2] R. Kubo, K. Tomita, A general theory of magnetic resonance absorption, J. Phys. Soc. (Jpn) 9 (1954) 888.
[3] J.H. Freed, Theory of slow tumbling ESR spectra of nitroxides, in: L.J. Berliner (Ed.), Spin Labeling: Theory and Applications, Academic Press, New York, 1976, p. 53.
[4] D.E. Budil, S. Lee, S. Saxena, J.H. Freed, Nonlinear-least-squares analysis of slow-motion EPR spectra in one and two dimensions using a modified Levenberg–Marquardt algorithm, J. Magn. Res. A 120 (1996) 155.
[5] D.E. Budil, K.A. Earle, J.H. Freed, Full determination of the rotational diffusion tensor by electron paramagnetic resonance at 20 GHz, J. Phys. Chem. 97 (1993) 1294.
[6] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308.
[7] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, Comput. J. 7 (1964) 155.
[8] I. Rechenberg, Evolutionsstrategie, Friedrich Frommann Verlag, Stuttgart, 1973.
[9] G.H. Golub, C.F. Van Loan, Matrix Computations, John Hopkins University Press, Baltimore, 1996.
[10] D.W. Marquardt, R.G. Bennett, E.J. Burrell, Least squares analysis of electron paramagnetic resonance spectra, J. Mol. Spectrosc. 7 (1961) 269.
[11] D.W. Marquardt, An algorithm for least squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math 11 (1963) 431.
[12] J.J. Moré, Numerical Analysis, in: G.A. Watson (Ed.), Lecture Notes in Mathematics, vol. 630, Springer, Berlin, 1977, p. 105.
[13] P.G. Fajer, R.L.H. Bennett, C.F. Polnaszek, E.A. Fajer, D.D. Thomas, General method for multiparameter fitting of high-resolution EPR spectra using a simplex algorithm, J. Magn. Res. 88 (1990) 111.
[14] P.G. Fajer, Method for the determination of myosin head orientation from EPR spectra, Biophys. J. 66 (1994) 2039.
[15] K. Jacobsen, S. Oga, W.L. Hubbell, T. Risse, Determination of the orientation of T4-lysozyme vectorially bound to a planar-supported lipid bilayer using site-directed spin labeling, Biophys. J. 88 (2005) 4351.

[16] S.V. Kolaczkowski, A. Perry, A. McKenzie, F. Johnson, D.E. Budil, P.R. Strauss, A spin-labeled abasic DNA substrate for AP endonuclease, Biochem. Biophys. Res. Commun. 288 (2001) 722.
[17] J.E. Dennis, D.M. Gay, R.E. Welsch, An adaptive nonlinear least-squares algorithm, ACM Transactions on Mathematical Software 7 (1981) 348.
[18] E. Meirovitch, D. Igner, E. Igner, G. Moro, J.H. Freed, Electron-spin relaxation and ordering in smectic and supercooled nematic liquid crystals, J. Chem. Phys. 77 (1982) 3915.
[19] E. Meirovitch, A. Nayeem, J.H. Freed, Analysis of protein lipid Interactions based on model simulations of electron spin resonance spectra, J. Phys. Chem. 88 (1984) 3454.
[20] D.J. Schneider, J.H. Freed, Calculating slow motional magnetic resonance spectra: a users guide, in: L.J. Berliner, J. Reuben (Eds.), Biological Magnetic Resonance, vol. 8, Plenum Press, New York, 1989, p. 1.
[21] K.V. Vasavada, D.J. Schneider, J.H. Freed, Calculation of ESR spectra and related Fokker–Planck forms by the use of the Lanczos algorithm. II. Criteria for truncation of basis sets and recursive steps utilizing conjugate gradients, J. Chem. Phys. 86 (1987) 647.
[22] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C, Cambridge University Press, Cambridge, MA, 1992.
[24] B. Kirste, Methods for automated analysis and simulation of EPR spectra, Anal. Chim. Acta 265 (1992) 191.