

Résonance paramagnétique électronique, optimisation et différentiation automatique

Christèle Faure, Edgar Soulié, Théo Berclaz

► **To cite this version:**

Christèle Faure, Edgar Soulié, Théo Berclaz. Résonance paramagnétique électronique, optimisation et différenciation automatique. [Rapport de recherche] RR-3907, INRIA. 2000, pp.30. <inria-00072746>

HAL Id: inria-00072746

<https://hal.inria.fr/inria-00072746>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Résonance paramagnétique électronique,
optimisation et différentiation automatique*

Christèle Faure — Edgar Soulié — Théo Berclaz

N° 3907

Mars 2000

THÈME 1



*Rapport
de recherche*

Résonance paramagnétique électronique, optimisation et différentiation automatique

Christèle Faure^{*}, Edgar Soulié[†], Théo Berclaz[‡]

Thème 1 — Réseaux et systèmes
Projet TROPICS

Rapport de recherche n° 3907 — Mars 2000 — 30 pages

Résumé : La spectroscopie de résonance paramagnétique électronique permet d'accéder à des informations sur l'interaction des moments magnétiques électroniques (spins) avec le champ magnétique appliqué (effet Zeeman) et avec les moments magnétiques nucléaires des noyaux voisins (interactions hyperfines). Si l'on dispose d'un modèle mathématique pour simuler des spectres de RPE de poudre par le calcul, les spectres de poudre expérimentaux permettent de déduire des paramètres spectroscopiques. Dans ce but, on minimise la somme des carrés des écarts entre les valeurs échantillonnées des signaux observé et calculé. La différentiation automatique combinée à la double précision permet de faire converger la minimisation en un petit nombre d'itérations.

Mots-clés : Odyssee, moindres carrés, Levenberg-Marquardt, résonance paramagnétique électronique, spectre de poudre, ajustement de paramètres

^{*} Email : Christele.Faure@sophia.inria.fr, URL : <http://www.inria.fr/tropics/Christele.Faure>

[†] Service de Chimie Moléculaire, CEA-DSM-DRECAM-SCM, CEA/Saclay, 91191 Gif-sur-Yvette Cedex,
Email : bsoulie@drecam.cea.fr

[‡] Sciences II, Dept. de Chimie Physique, Université de Genève, 1211- Genève-4, Suisse.
Email : Theo.Berclaz@chiphy.unige.ch

Electron paramagnetic resonance, optimization and computational differentiation

Abstract: Electron paramagnetic resonance (EPR) spectroscopy is a method to investigate the behavior of samples containing unpaired electrons (free radicals or compounds comprising an ion whose outer electronic shell is incomplete) in an applied magnetic field. It consists in recording the microwave energy absorbed by the sample as a function of the applied magnetic field. Two kinds of information may usually be obtained by this technique. The first one is a tensor characterizing the so-called Zeeman interaction, or interaction of the applied magnetic field with the unpaired electron(s). The second consists of tensors characterizing the interaction of the unpaired electron(s) with each type of nucleus possessing an intrinsic magnetic moment or spin, provided that this interaction is reflected by specific lines in the EPR spectrum. The interpretation of the results obtained from EPR spectroscopy relies on a theory deriving from quantum mechanics. This theory describes the variation of the resonance field as a function of the orientation of the microscopic resonating species in the applied magnetic field and of adjustable parameters. The values of the magnetic field which fulfill the "resonance condition" may be determined by a numerical iterative procedure, or by approximate analytic formulae. Such analytic formulae are valid under generally verified assumptions which were published by Iwasaki. The quantitative analysis of the variation of the resonance field with orientation leads to accurate values of the adjustable parameters in the case of a "single crystal", that is a regular arrangement of atoms or molecules in three directions of space. However, many interesting materials which lend themselves to EPR investigations are disordered (powders, frozen solutions, etc.) and their EPR spectra do not vary with orientation in the magnetic field. For such materials, the spectrum of a sample is the sum of the spectra of the microscopic resonating species. Provided that the theory sketched above for a single crystal can be extended to encompass the intensities, shapes and widths of the resonance lines, a spectrum of the sample may be predicted for a given set of spectroscopic parameters.

As in many fields of science and technique, the inverse problem consists in determining the values of the spectroscopic parameters for which the calculated spectrum best matches the observed spectrum. The minimization of the sum of squares of differences between the intensities of the observed and calculated spectra provides the means of fitting the spectroscopic parameters. Automatic differentiation of algorithm (AD) was undertaken in order to avoid approximation and to obtain a better displacement of the current point in \mathbb{R}^n at each iteration of the optimizer. For that purpose, the simulator is differentiated using the automatic differentiator ODYSSEÉ in direct mode. ODYSSEÉ provides us with a new programming unit that computes both the spectrum and one directional derivative of the spectrum with respect to one adjustable parameter. Our choice is to modify the optimizer BSOLVE as little as possible: the modified version of BSOLVE calls this new programming unit or the original simulator to compute the Jacobian matrix line by line.

Key-words: Odyssee, least squares method, Levenberg-Marquardt, Electron paramagnetic resonance, powder spectrum

Table des matières

1	Introduction	7
2	Le simulateur	9
2.1	L'intensité du champ magnétique à la résonance	9
2.2	L'intensité de la résonance	11
2.3	La forme de la résonance	11
2.4	La largeur de la raie	11
2.5	Signal calculé	11
2.6	Le rôle du simulateur	12
3	Le processus d'optimisation	13
3.1	L'optimiseur	13
3.2	Différentiation automatique d'algorithme	14
3.3	Adaptation de l'optimiseur	14
3.3.1	Différentiation du simulateur	15
3.3.2	Modification de l'optimiseur	15
3.4	Résultats de complexité	15
4	Essais numériques	19
4.1	Premier essai	20
4.2	Second essai	22
5	Conclusion	25
A	Calcul du champ magnétique de résonance	27

Chapitre 1

Introduction

La spectroscopie de résonance paramagnétique électronique permet d'accéder à des informations sur l'interaction des moments magnétiques électroniques (spins) avec le champ magnétique appliqué (ou effet Zeeman) et avec les moments magnétiques nucléaires des noyaux voisins (interactions hyperfines). Dans le cas d'un monocristal, on enregistre le spectre de résonance paramagnétique électronique pour différentes orientations du monocristal. En ajustant les valeurs des paramètres intéressants dans l'opérateur appelé Hamiltonien effectif, on amène les valeurs du champ de résonance calculé correspondant aux différentes orientations à reproduire au mieux les valeurs du champ de résonance observé. Cette opération exige uniquement la connaissance d'une loi de variation du champ de résonance avec l'orientation. Dans le cas d'une poudre, le spectre de résonance paramagnétique électronique est la somme algébrique des spectres des micro cristaux (dimension de l'ordre du micron) dont la poudre est constituée. Le spectre de la poudre est normalement invariant par rotation de l'échantillon. Cette hypothèse est vérifiable en tournant l'échantillon dans le spectromètre. L'analyse quantitative d'un spectre de poudre a pour ambition de déterminer les paramètres spectroscopiques comme on le ferait sur un monocristal. Mais il existe une différence importante: dans le cas d'une poudre, on a besoin de connaître non seulement la loi de variation de la position de chaque résonance (c.a.d l'intensité du champ magnétique de résonance en fonction des angles (θ, ϕ) qui définissent l'orientation du cristal), mais encore les lois de variation de l'intensité, de la forme de raie et de la largeur de raie en fonctions de ces angles. Bien que l'orientation d'un solide dans l'espace exige la définition de trois angles, on peut se contenter des deux angles θ et ϕ , à condition de repérer l'orientation du champ magnétique appliqué par rapport aux "axes principaux" du tenseur g qui exprime l'Hamiltonien Zeeman électronique.

Dans le Chapitre 2, nous décrivons en détail le simulateur, un logiciel qui pour un ensemble donné de paramètres ajustables, calcule un spectre de poudre, auquel le spectre expérimental sera comparé. Dans le Chapitre 3.1, nous décrivons succinctement l'optimiseur, ou logiciel d'optimisation que nous avons utilisé. Ce logiciel ajuste les paramètres de telle façon que la somme des carrés des écarts entre les intensités observées et calculées soit

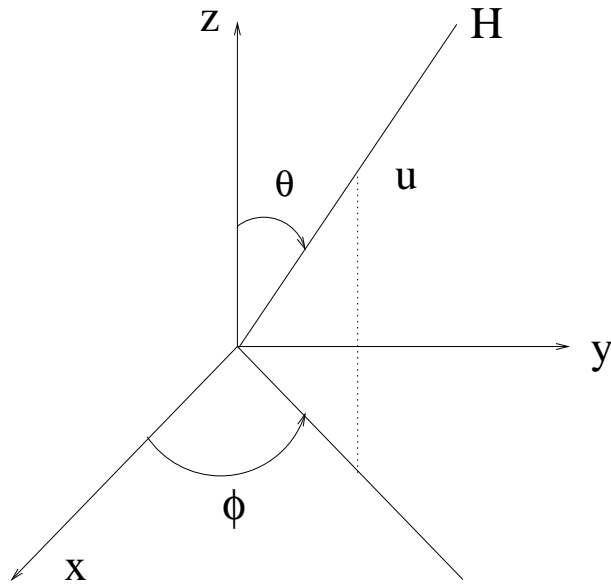


FIG. 1.1 – Orientation du champ magnétique statique par rapport aux axes principaux du tenseur g d'un micro-cristal.

minimale. La complexité du problème d'une part, l'importance du temps de calcul d'autre part, nous ont incités à explorer l'application de la Différentiation Automatique (DA) à ce problème. Nous avons substitué un calcul exact de dérivée directionnelle par DA à une approximation par différences divisées. Nous décrivons l'application du logiciel *Odyssée* au "simulateur de spectre de RPE" dans la Section 3.2, et les résultats des tests effectués au Chapitre 4.

Chapitre 2

Le simulateur

Le simulateur est un ensemble de sous-programmes (écrits en Fortran 77) permettant de calculer un spectre de RPE de poudre lorsque l'on suppose connus les paramètres ajustables. Une poudre est constituée par un grand nombre de microcristaux dont les orientations sont quelconques. En conséquence, un spectre de poudre est la somme des spectres des microcristaux qui la constituent. Un spectre de poudre est normalement invariant par rapport à une rotation de l'échantillon dans le spectromètre. Si cette hypothèse est vérifiée expérimentalement, on peut admettre que les orientations des microcristaux constituant la poudre sont equiréparties. L'orientation d'un solide par rapport à un repère est définie par trois angles (par exemple les angles d'Euler). Mais il suffit de deux angles pour définir l'orientation du champ magnétique appliqué par rapport à un repère tridimensionnel. Aussi utilisons-nous ci-dessus un repère différent pour chaque micro-cristal, à savoir celui constitué par les axes principaux du tenseur définissant l'interaction du moment magnétique électronique avec le champ magnétique appliqué (ou effet Zeeman électronique).

Pour chaque micro-cristal, on a donc besoin de calculer la variation du signal de résonance en fonction de l'intensité du champ magnétique appliqué. Le calcul repose sur quatre ingrédients: l'intensité du champ magnétique à la résonance, l'intensité de la résonance, la forme de la résonance, la largeur de la raie.

2.1 L'intensité du champ magnétique à la résonance

On détermine théoriquement celle-ci en résolvant une équation transcendante exprimant que l'énergie transférée par une onde électromagnétique excitatrice (quantum d'énergie $h\nu$) est égale à l'énergie absorbée pour promouvoir le moment magnétique électronique de son état quantique fondamental (d'énergie plus basse) à un état quantique excité:

$$h\nu = E_2(B) - E_1(B)$$

Dans le second membre, $E_1(B), E_2(B)$ représentent les valeurs propres d'un opérateur (au sens de la mécanique quantique) appelé Hamiltonien effectif \mathcal{H} , pour la valeur B de l'intensité du champ appliqué. Dans le cas qui nous intéresse, cet opérateur Hamiltonien effectif comporte deux sortes de termes (qui s'ajoutent):

1. le terme d'interaction Zeeman électronique qui est de la forme

$$Z = \beta.B.\bar{g}.\hat{S}$$

où β désigne le magnéton de Bohr (une constante de la physique), B le champ magnétique appliqué, \bar{g} le tenseur "g effectif" de dimension trois constitué de nombres purs (sans unités), \hat{S} un opérateur vectoriel de dimension trois c'est à dire constitué des trois opérateurs scalaires S_x, S_y, S_z . Ces opérateurs agissent sur les "états de spin" d'un moment électronique qui sont notés traditionnellement $|\alpha\rangle$ et $|\beta\rangle$ (leur projection sur l'axe de quantification vaut respectivement $+1/2$ et $-1/2$). Si le terme d'interaction Zeeman électronique était seul présent, la différence d'énergie $E_2(B) - E_1(B)$ aurait pour expression analytique:

$$\beta B \sqrt{(g_x^2 \cos^2(\phi) + g_y^2 \sin^2(\phi)) \sin^2(\theta) + g_z^2 \cos^2(\theta)}$$

2. des termes d'interaction entre le moment magnétique électronique et les moments magnétiques de noyaux voisins, porteurs de spin nucléaire. Chaque terme est de la forme $\hat{I}\bar{A}\hat{S}$ où \hat{I} est un opérateur vectoriel de spin nucléaire. Si le spin nucléaire vaut I (nombre entier ou demi-entier), sa projection sur un axe de quantification peut prendre les valeurs $I, I-1, \dots, -I$. \bar{A} est un tenseur exprimé en unité d'énergie. Un suffixe s'attache à chaque opérateur \hat{I} ainsi qu'au tenseur \bar{A} correspondant.

L'équation aux valeurs propres de l'opérateur:

$$\mathcal{H} = Z + \sum_{i=1}^n I_i A_i \hat{S}$$

n'a pas de solution analytique dès que $n \geq 1$. On peut résoudre l'équation de résonance ou bien de manière numérique et itérative, ou bien de manière approchée par la technique du calcul de perturbations. Cette seconde manière présente le gros avantage de permettre l'utilisation de formules analytiques, évitant d'introduire un niveau supplémentaire de calcul itératif. De nombreuses formules ont été proposées pour différents cas particuliers. Le japonais Iwasaki[12] a proposé des formules dont la validité assez est générale. Dans la cas présent, les formules sont valides à condition que les termes d'interaction hyperfine soient petits par rapport au terme Zeeman. Ces formules sont vectorielles et tensorielles et de ce fait concises. Elles n'ont cependant pas été données dans un ordre correspondant à l'écriture d'un algorithme. Elles sont valables dans un cadre plus général que celui que nous envisageons ici. En Annexe A, nous les donnons pour le cas d'un terme Zeeman et de termes hyperfins.

2.2 L'intensité de la résonance

L'onde électromagnétique induit entre les états initial et final une transition dont on calcule la probabilité à l'aide d'une formule générale de la mécanique quantique appelée "règle d'or de Fermi". L'application de cette formule au cas d'un tenseur Zeeman dont les composantes principales sont deux à deux différentes conduit à une formule $P(\theta, \phi)$ qui fut calculée par Bleaney [1] et Pilbrow [18]. En toute rigueur, il faudrait utiliser les états propres de l'Hamiltonien effectif complet.

2.3 La forme de la résonance

La résonance n'est pas un phénomène de largeur nulle: l'intensité du signal diminue avec l'écart du champ appliqué au champ de résonance. Soient σ la demi largeur à mi-hauteur, et l'écart réduit x :

$$x = \frac{B - B_{res}}{\sigma}$$

les deux fonctions de forme les plus couramment rencontrées sont:

la fonction de Cauchy-Lorentz $f(B) = \frac{1}{1+x^2}$

la fonction de Gauss-Laplace $f(B) = \exp(-Ln(2).x^2)$

D'autres fonctions de forme se rencontrent, notamment la convolée de ces deux fonctions.

2.4 La largeur de la raie

Des expériences faites sur des mono-cristaux indiquent que la largeur exprimée en unité de champ varie beaucoup plus que la largeur exprimée en unité de fréquence. En l'absence d'une théorie satisfaisante, nous faisons l'hypothèse que la largeur exprimée en unité de fréquence varie avec l'orientation du micro-cristal selon la loi :

$$\sigma_\nu(\theta, \phi) = \sqrt{(\sigma_x^2 \cos^2(\phi) + \sigma_y^2 \sin^2(\phi)) \sin^2(\theta) + \sigma_z^2 \cos^2(\theta)}$$

et que la largeur exprimée en unité de champ a pour expression :

$$\sigma_B(\theta, \phi) = \frac{h\sigma_\nu(\theta, \phi)}{\beta g_{\text{eff}}(\theta, \phi)}$$

2.5 Signal calculé

Avec les ingrédients détaillés ci-dessus, pour une valeur donnée du champ magnétique, on calcule la contribution d'un micro-cristal au signal de RPE. Le signal dû à la poudre est la somme des signaux élémentaires dûs aux micro-cristaux et on le calcule comme une

intégrale double sur les angles. Compte tenu du fait que l'on observe expérimentalement non pas l'absorption d'énergie, mais la dérivée de cette absorption par rapport à B en fonction de B , le signal calculé a pour expression:

$$S(B) = \int \int P(\theta, \phi) \cdot \frac{\partial f}{\partial B} \left[\frac{B - B_{res}(\theta, \phi)}{\sigma_B(\theta, \phi)} \right] \cos \theta d\theta d\phi$$

La double intégration sur la sphère ne peut pas être réalisée par le calcul formel [19]. Le calcul numérique de cette intégrale peut être fait de nombreuses manières. Des expériences numériques ont montré que la substitution de la variable $z = \cos \theta$ à la variable θ , et l'utilisation de la méthode d'intégration de Gauss à 96 points [2] permettaient d'améliorer nettement la qualité de la simulation. Par contre, l'introduction de cette méthode pour l'angle ϕ a donné des résultats incorrects. Afin de limiter le "bruit numérique" dans le spectre calculé, nous avons pris une origine aléatoire pour l'intégration sur ϕ . Le spectre ainsi calculé doit être mis à l'échelle du spectre observé et sa "ligne de base" ajustée à celle du spectre observé. Ces deux opérations se font en résolvant un système linéaire.

2.6 Le rôle du simulateur

Étant donné un ensemble de N paramètres ajustables (paramètres spectroscopiques) représenté par un vecteur \vec{x} de dimension N , le simulateur calcule un spectre sous la forme de M valeurs numériques du signal correspondant aux M valeurs échantillonnées du champ magnétique statique appliqué. Le simulateur s'insère dans un ensemble de trois parties: l'"exécutant" (première partie) chargé de la conduite de l'ensemble des calculs et notamment de l'appel au sous-programme ayant pour rôle la conduite de l'optimisation appelé "optimiseur" et enfin le "simulateur". Il faut noter que le simulateur peut être appelé un grand nombre de fois par l'optimiseur. Les paramètres ajustables qui composent le vecteur \vec{x} sont :

$x(1 - 6)$ les six composantes du tenseur symétrique g ,

$x(7 - 9)$ les trois composantes de la demi-largeur de raie,

$x(10 - 15)$ les six composantes du tenseur hyperfin du premier noyau (en Megahertz),

$x(16 - 21)$ les composantes du tenseur hyperfin du second noyau (en Megahertz),

$x(22 - 27)$ les composantes du tenseur hyperfin du troisième noyau (en Megahertz) ...

Chapitre 3

Le processus d'optimisation

3.1 L'optimiseur

Pour la minimisation d'une somme de carrés de fonctions non linéaires deux fois dérivables, rencontrée dans de nombreux problèmes en physico-chimie et notamment en spectroscopie, il est opportun de recourir à un logiciel mettant en oeuvre l'algorithme de Levenberg-Marquardt [14, 15]. En pratique, il est important de pouvoir assigner des bornes inférieure et supérieure à chaque paramètre ajustable et de pouvoir bloquer ou débloquer chaque paramètre. Le logiciel `BSOLVE` de William Ball [3] repris par Kuester et Mize [13] offre ces fonctionnalités. Malgré la présence de constantes numériques qui devraient normalement dépendre du processeur utilisé, nous avons utilisé ce logiciel qui met en oeuvre l'algorithme de Levenberg-Marquardt de façon souple, en ce sens qu'il permet de calculer chaque ligne de la matrice Jacobienne soit par différences divisées, soit en utilisant des formules analytiques codées à la main.

À chaque itération, le logiciel indique le nombre des paramètres concernés par l'ajustement. Lorsque l'optimisation se termine normalement, ce nombre vaut zéro. Mais compte tenu de la complexité des calculs, les erreurs d'arrondi s'accumulent et il arrive souvent que l'optimisation s'interrompt avec impression du diagnostic `Amélioration de la fonction impossible`. Nous soupçonnions depuis longtemps que ce diagnostic pouvait résulter du fait que les différences divisées fournissent une matrice Jacobienne dont la précision est insuffisante. Ainsi après une itération, l'optimiseur serait "mal guidé" par la matrice Jacobienne vers l'itération suivante. Cette idée nous a incités à recourir à la différentiation automatique pour obtenir à partir du code du simulateur, un code dérivé qui calcule la matrice Jacobienne de façon exacte. La formulation exacte des dérivées mise en oeuvre par la DA doit nous permettre d'améliorer la qualité du calcul du déplacement du vecteur \vec{x} dans l'espace \mathbb{R}^N .

3.2 Différentiation automatique d'algorithme

La différentiation automatique (DA) décrite dans [11] et [5] se compose d'un ensemble de techniques permettant de calculer des dérivées d'une fonction décrite par un programme (c.a.d en des points arbitraires). La DA est fondée sur deux observations: l'exécution d'un programme peut être considérée comme la composition de fonctions élémentaires et peut donc être différenciée en utilisant la règle de dérivation des fonctions composées. Les dérivées des fonctions élémentaires sont calculées en utilisant les règles standard de dérivation des expressions: par exemple "la dérivée d'une somme est la somme des dérivées" ...

Deux modes de DA ont été étudiés dans divers travaux et mis en oeuvre dans les outils actuels: le **mode direct** permet le calcul de dérivées directionnelles et des valeurs originales simultanément, le **mode inverse** calcule d'abord les valeurs originales, puis leurs dérivées associées dans l'ordre inverse. Le mode inverse est particulièrement efficace pour calculer des gradients car par cette méthode le coût en temps du gradient est indépendant du nombre d'entrées (taille du gradient).

Deux classes d'outil de DA ont été développées: ceux qui travaillent par *surcharge d'opérateurs* et ceux qui travaillent par *génération de code*. Dans le premier cas, l'utilisateur n'a accès qu'à un exécutable qui calcule les dérivées; dans le second cas l'utilisateur obtient un code source qu'il peut intégrer à une chaîne logicielle telle qu'un optimiseur. Les logiciels *Odyssée*, *Adifor*[6], et *TAMC*[10] appartiennent à la seconde classe d'outils. *Odyssée* (voir [9]) est un outil de différentiation automatique, développé à l'INRIA et à l'Université de Nice - Sophia Antipolis, qui différencie des unités de programme écrites en *Fortran 77*. Ce logiciel présente deux caractéristiques principales par rapport aux outils de la même classe: il peut différencier un programme même si certaines unités du code source ne peuvent pas être lues par le système et le mode inverse qu'il implémente est applicable à des codes opérationnels de grande taille (100 000 lignes) [8, 7]. Pour pouvoir traiter un programme dont toutes les unités ne sont pas disponibles (provenant de bibliothèques compilées par exemple), le système a besoin que l'utilisateur décrive les unités manquantes dans un langage particulier. De plus, l'utilisateur devra fournir lui-même les dérivées de ces unités manquantes. Si une fonction est mise en oeuvre sous la forme d'un ensemble d'unités de programme, *Odyssée* est capable de la différencier dans sa totalité par rapport aux variables d'entrée spécifiées par l'utilisateur. En mode direct, à partir de cet ensemble d'unités, *Odyssée* utilise l'algorithme linéaire tangent pour produire un code *Fortran 77* qui calcule une dérivée directionnelle. En mode inverse, le code généré automatiquement par *Odyssée* est un code adjoint, c'est-à-dire qu'il calcule le produit d'un vecteur par la matrice Jacobienne transposée.

3.3 Adaptation de l'optimiseur

Comme nous l'avons dit précédemment, la version de l'optimiseur *BSOLVE* que nous utilisons offre la possibilité de calculer la matrice Jacobienne par différences divisées ou dérivées analytiques. Nous avons adapté l'optimiseur pour pouvoir calculer les lignes de la matrice Jacobienne par des appels au code dérivé du simulateur.

Dans le cas de l’ajustement des paramètres spectroscopiques en RPE, il y a beaucoup plus de fonctions calculées (M de l’ordre de 500 à 4000) que de paramètres ajustables ($N = 21$ pour deux tenseurs, $N = 27$ pour trois tenseurs, ...); nous avons donc gardé le calcul de la matrice Jacobienne par lignes, ce qui nous a amené à utiliser le mode direct de DA.

3.3.1 Différentiation du simulateur

Nous avons d’abord différentié le simulateur `simula` afin d’obtenir le code dérivé qui calcule la dérivée du vecteur `sigcal` représentant les valeurs échantillonnées du signal (spectre discrétisé) par rapport aux paramètres ajustables regroupés dans la variable `x`.

En mode direct, le logiciel `Odyssée` a généré le programme `simulat1` qui calcule une dérivée directionnelle:

$$sigcalttl(i) = \sum_j \frac{\partial sigcal(i)}{\partial x(j)} xttl(j).$$

Les commandes `Odyssée` permettant de fabriquer le sous-programme `simulat1` à partir du sous-programme `simula` lu dans le fichier `simula` et de l’écrire dans le fichier `simulat1` sont :

```
load simula
diff -tl -vars x -h simula -o simulat1
```

3.3.2 Modification de l’optimiseur

Une nouvelle option a été ajoutée à l’optimiseur, dont la version modifiée peut appeler ou bien le code produit automatiquement `simulat1` ou bien le code du simulateur originel `simula` pour calculer une ligne de la jacobienne. Ce comportement est contrôlé pour chaque paramètre ajustable désignée par son rang `rang` au moyen d’un `aiguilleur` entier `BV(rang)`. Si la valeur de cet aiguilleur est 0, le paramètre est bloqué. Si cette valeur est positive, l’optimiseur calcule la dérivée directionnelle par différences divisées en appelant le simulateur originel `simula`. Si cette valeur est négative, l’optimiseur appelle le sous-programme `simulat1` pour calculer la dérivée directionnelle.

La Figure 3.1 montre que les modifications apportées à l’optimiseur sont très réduites.

3.4 Résultats de complexité

Pour calculer N dérivées directionnelles en utilisant les différences divisées, on fait $N + 1$ appels au simulateur. Pour faire le même calcul en recourant au mode direct de DA, on appelle N fois la dérivée du simulateur. Par ailleurs, on sait qu’en pratique la dérivée du simulateur coûte entre deux et trois fois plus que le simulateur lui-même. En théorie, le rapport devrait être de 5 (si toutes les opérations effectuées dans le code étaient des divisions), comme l’ont montré Baur et Strassen[4] et Morgenstern[16, 17]. Si on note T le temps d’exécution du simulateur, les N dérivées directionnelles sont calculées en $(N+1)T$ par différences divisées et $3NT$ par différentiation automatique. Ainsi, le temps d’exécution d’un

```

C----- Computation of the directional derivatives
      DO J1 = 1, K
          XTTL(J1) = 0.
      END DO
      DO 34 J1 = 1, K
          N1 = (J1 - 1)* N
C----- non adjustable parameter
          IF( BV(J1) .EQ. 0 ) GO TO 34

C----- partial derivative w.r.t. J1-ème parameter by finite differences
          IF( BV(J1) .GT. 0 ) GO TO 24

C---- partial derivative w.r.t. J1-ème parameter by AD

          XTTL(J1) = 1.
C-----
C P( KBI1 + 1) = vecteur des parametres ajustables
C P( N1 + 1) = vecteur des fonctions calculees
C-----

          CALL SIMULATL( AFINT , ALM , ALMI , ALMIN , X ,  FREQ ,
&                      NALF , NBNOY , NBTR , NBTRT , NPFI , NN , Y ,
&                      SPIN , P(KBI1+1) , NAPSIM , BETA , AUX ,
&                      XTTL , BETATTL , P(N1+1))
          XTTL(J1) = 0.
          GO TO 34

C----- partial derivative w.r.t. J1-ème parameter by finite differences
C      Scd call to the simulator

      24 CONTINUE

...
      CALL SIMULA( AFINT , ALM , ALMI , ALMIN , X ,  FREQ ,
&                NALF , NBNOY , NBTR , NBTRT , NPFI , NN , Y ,
&                SPIN , P(KBI1+1) , NAPSIM , BETA , P(N1+1))

...
      34 CONTINUE

```

FIG. 3.1 – *Modification de l'optimiseur*

pas d'optimisation utilisant un code produit par différentiation automatique est supérieur au temps d'exécution lorsque les différences divisées sont utilisées.

En contrepartie de l'augmentation en temps de ce calcul, la précision de la matrice Jacobienne est améliorée et l'optimiseur peut calculer un déplacement meilleur depuis le point courant \vec{x} vers le point suivant, et finalement converger plus vite. C'est ce que nous montrons dans le chapitre suivant.

Chapitre 4

Essais numériques

Dans ce chapitre, nous comparons le comportement de l'optimiseur selon qu'il fait appel aux différences divisées ou à la différentiation automatique. Nous examinons les liens entre ce comportement et la précision numérique du calcul dans son ensemble. Nous avons réalisé deux optimisations différentes, utilisant des données différentes pour les paramètres ajustables. Chacune de ces deux optimisations a été effectuée en utilisant quatre options possibles résultant de la combinaison d'un choix de précision (simple ou double précision), et d'un choix de la méthode de calcul des dérivées (différences divisées ou différentiation automatique).

Pour faciliter la lecture des résultats qui suivent, nous avons associé un sigle à chacun des quatre cas examinés :

DD-R4 différences divisées, optimisation en simple précision,

DA-R4 différentiation automatique, optimisation en simple précision,

DD-R8 différences divisées, optimisation en double précision,

DA-R8 différentiation automatique, optimisation en double précision.

Il est difficile de faire la différence entre les solutions obtenues et donc d'évaluer la "qualité" du résultat. Nous n'essayons donc pas de comparer les spectres obtenus en utilisant les différentes options. Nous analysons le déroulement de l'optimisation en observant certains indicateurs fournis par l'optimiseur à chaque itération. Les indicateurs les plus directement interprétables sont :

icon nombre de paramètres restant à ajuster,

napsim nombre cumulé d'appels au simulateur si on ne compte pas le nombre d'appels dus au calcul de dérivées directionnelles,

somcar valeur de la fonction objectif,

gamma angle entre la direction de Newton-Raphson modifiée et la direction de plus grande pente.

Les tests décrits dans la suite ont été effectués sur une station SUN OS5 possédant 512 Megaoctets de mémoire.

4.1 Premier essai

Ce cas se rapporte à un radical libre ayant des interactions hyperfines avec un noyau de phosphore-31 ($I = 1/2$) et un noyau de carbone-13 ($I = 1/2$). Le spectre de RPE est échantillonné sur 4096 points et il y a 21 paramètres dont 12 sont bloqués. Il reste donc 9 paramètres ajustables, ce qui signifie que la matrice jacobienne nécessaire au déroulement de l'optimisation peut être calculée en utilisant 9 dérivées directionnelles dans les directions de la base canonique.

Les diagnostics émis par l'optimiseur dans les quatre cas comparés pour le premier essai sont :

DD-R4 Amélioration de la fonction impossible après 11 itérations,

DA-R4 Solution obtenue après 26 itérations,

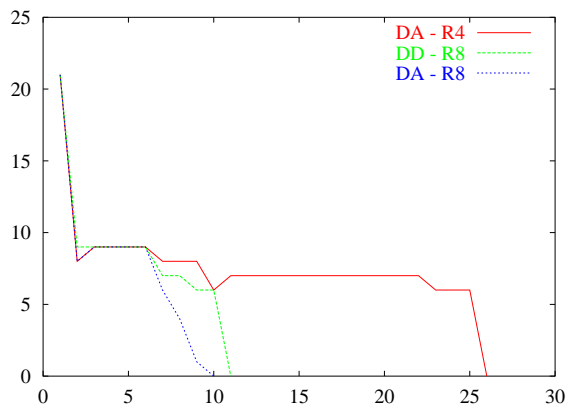
DD-R8 Solution obtenue après 11 itérations.

DA-R8 Solution obtenue après 10 itérations.

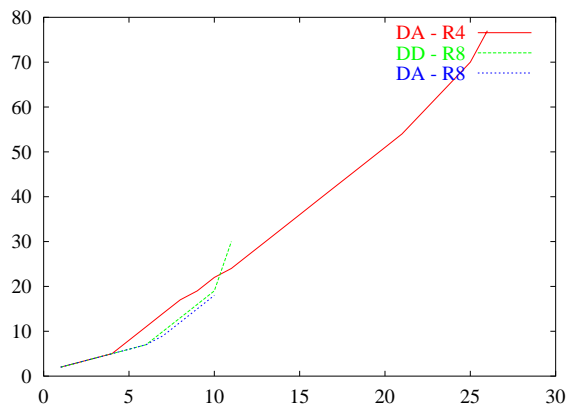
La Figure 4.1 présente l'évolution des indicateurs que nous avons retenus au cours des itérations de l'optimisation. Le cas DD-R4 n'aboutissant à aucune solution, nous avons choisi de ne pas le présenter. Cette évolution montre que le déroulement de l'optimisation est meilleur si la matrice Jacobienne est calculée par différentiation automatique et non approximée par différences divisées : le nombre de paramètres restant à ajuster Figure 4.1(a) diminue beaucoup plus vite, le nombre d'appels au simulateur Figure 4.1(b) est plus petit, la valeur finale de la fonction objectif Figure 4.1(d) est plus faible et l'angle entre la direction de Newton-Raphson modifiée et la direction de plus grande pente Figure 4.1(c) évolue de façon moins chaotique. L'utilisation de la double précision combinée avec la DA permet encore d'améliorer l'optimisation. On peut noter que la valeur finale de la fonction objectif est plus faible en simple qu'en double précision, ce phénomène pourrait résulter de la sensibilité du calcul du simulateur à la précision des calculs.

Méthode	Précision	Processus d'optimisation		
		temps total (s)	nombre d'itérations	temps unitaire (s)
DD	R4	1757.10	11	159.7
DA	R4	7723.57	26	297.0
DD	R8	3119.29	11	283.6
DA	R8	4555.98	10	455.6

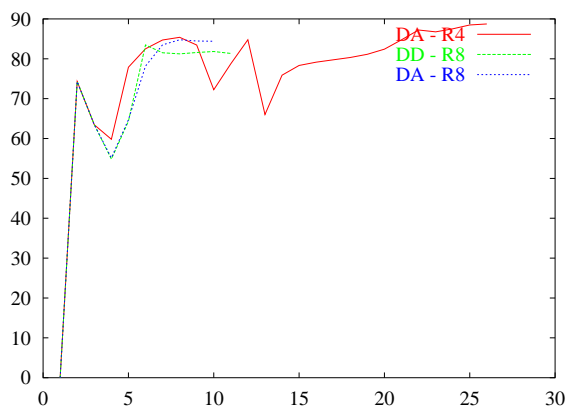
La table ci-dessus présente les temps d'exécution des quatre optimisations correspondant à ce premier essai. Pour chaque optimisation trois valeurs sont présentées : le **temps total** d'exécution en secondes, le **nombre d'itérations** d'optimisation et le **temps unitaire** qui est le temps moyen nécessaire à l'exécution d'une itération. Il faut noter que l'optimisation (DD-R4) se termine par un échec après 1757.10 secondes d'exécution. Pour ce calcul (DD-R4), la seule valeur de temps à considérer est le temps d'exécution d'une itération. On fait l'hypothèse que le temps unitaire est exactement le temps de calcul de la Jacobienne



(a) icon



(b) napsim



(c) gamma

Valeur finale de la fonction objectif	
DD-R4	0.11042409E+06
AD-R4	0.11039414E+06
DD-R8	0.11062249E+06
AD-R8	0.11062246E+06

(d) somcar

FIG. 4.1 – Premier essai

Méthode	Précision	Nombre de paramètres ajustés	
		12	15
DD	R4	0.10719038E+06	0.10670412E+06
DA	R4	0.10700169E+06	0.10662119E+06

TAB. 4.1 – Valeurs du critère en fin d'exécution

en négligeant le temps nécessaire au reste du calcul. On peut remarquer alors, que le surcoût dû à l'utilisation de la différentiation automatique est de $1.86 = 297/159.7$ en simple précision et $1.61 = 455.6/283.6$ en double précision. Ce facteur est nettement inférieur au rapport généralement observé (compris entre deux et trois) et à plus forte raison au rapport théorique qui est de cinq.

En conclusion, on peut dire que sur cet exemple, utiliser une matrice Jacobienne calculée par différentiation automatique à la place d'une approximation par différences divisées permet d'obtenir une solution quelle que soit la précision des calculs choisie. De plus, si on associe la double précision à la différentiation automatique, la convergence de l'optimiseur est accélérée : on passe de 26 itérations de l'optimiseur à 10. Bien qu'une itération en double précision coûte 1,54 fois une itération exécutée en simple précision, le temps total d'optimisation en double précision est 1,69 fois plus petit qu'en simple précision. La contrepartie à l'exactitude de la matrice Jacobienne est en double précision une augmentation du temps total d'optimisation d'un facteur 1,46.

4.2 Second essai

Ce cas se rapporte au même radical libre que celui de l'essai précédent. L'interprétation correcte du spectre nécessite la prise en compte d'une troisième interaction hyperfine du radical libre avec un proton ou noyau d'hydrogène ($I = 1/2$) comme le suggéraient des expériences d'électrolyse d'une part et des calculs de chimie quantique d'autre part. Le spectre de RPE est échantillonné sur 4096 points et il y a 27 paramètres ajustables. Pour ces données, deux calculs ont été effectués : l'un en bloquant 15 paramètres ce qui laisse 12 paramètres à ajuster par l'optimiseur, et l'autre en bloquant 12 paramètres ce qui laisse 15 paramètres à ajuster.

Les diagnostics émis par l'optimiseur (exécuté sur station SUN) en simple précision pour le deuxième essai avec 12 ou 15 paramètres ajustés sont :

DD-R4 12 paramètres ajustés Amélioration de la fonction impossible après 8 itérations,

DA-R4 12 paramètres ajustés Solution obtenue après 37 itérations,

DD-R4 15 paramètres ajustés Amélioration de la fonction impossible après 14 itérations,

DA-R4 15 paramètres ajustés Solution obtenue après 61 itérations.

On peut remarquer que la valeur finale de coût (somme des carrés des écarts entre spectre calculé et spectre observé) présenté Table 4.1 sont plus faible en utilisant la différentiation automatique que les différences divisées. Ceci signifie que l'optimiseur est mieux guidé par les directions calculées par différentiation automatique.

Méthode	Précision	P. ajustés	Processus d'optimisation		
			temps total (s)	nb. d'itérations	temps unitaire (s)
DD	R4	12	3170.25	8	396.3
DA	R4	12	27523.05	37	743.9
DD	R4	15	6619.44	14	472.8
DA	R4	15	55812.81	61	915.0

Comme pour l'essai précédent, on peut remarquer que le surcoût dû à l'utilisation de la différentiation automatique est de $1.88 = 743.9/396.3$ (resp. $1.93 = 915.0/472.8$) en simple précision. Ce facteur est nettement inférieur au rapport généralement observé (compris entre deux et trois) et à plus forte raison au rapport théorique qui est de cinq.

Chapitre 5

Conclusion

Alors que la détermination à la main des formules analytiques de calcul de la jacobienne aurait été beaucoup trop longue et difficile à faire et à vérifier, l'utilisation du logiciel *Odyssée* nous a permis de fabriquer automatiquement une unité de programme calculant cette jacobienne par lignes. Les essais réalisés montrent que la substitution de la différentiation automatique aux différences divisées permet d'éviter une terminaison sans convergence de la méthode d'optimisation. En outre, le passage en double précision sur SUN permet dans certains cas de faire converger l'optimisation en un petit nombre d'itérations.

Annexe A

Calcul du champ magnétique de résonance

Ce chapitre présente le calcul du champ magnétique de résonance au moyen des formules d'Iwasaki[12]. Nous résumons ici l'algorithme valide dans le cas d'un terme Zeeman électronique et d'un, deux ou plusieurs termes d'interaction hyperfine dans l'Hamiltonien hyperfin. Le vecteur unitaire de composantes U_x, U_y, U_z est parallèle au champ magnétique (voir Figure 1.1).

$$\begin{aligned}
 U_x &= \sin \theta \cdot \cos \phi \\
 U_y &= \sin \theta \cdot \sin \phi \\
 U_z &= \cos \theta \\
 \bar{p} &= \bar{g} \cdot \bar{U} \\
 g &= \text{norme}(\bar{p}) \\
 \bar{q} &= \bar{A} \cdot \bar{p} / g \\
 r &= \text{norme}(\bar{q}) \\
 \bar{B} &= \bar{A} \cdot \bar{A} \\
 \bar{C} &= \bar{B} \cdot \bar{B} \\
 s &= \bar{p}^T \cdot \bar{C} \cdot \bar{p} / g^2 \cdot r^2 \\
 t &= \text{trace}(\bar{B}) \\
 w &= s - r^2 \\
 v &= t - s
 \end{aligned}$$

L'énergie au premier ordre (en unité de fréquence) a pour expression :

$$E^{(1)} = v - r \cdot I_z$$

L'énergie au second ordre (en unité de fréquence) a pour expression :

$$E^{(2)} = E^{(1)} - \frac{w \cdot I_z^2}{2 \cdot E^{(1)}} - v \frac{I(I+1) - I_z^2}{4 \cdot E^{(1)}}$$

Finalement, le champ magnétique à la résonance vaut :

$$B = \frac{\hbar \cdot E^{(2)}}{\beta \cdot g}$$

où \hbar est la constante de Planck et β le magnéton de Bohr.

Bibliographie

- [1] A. Abragam and B. Bleaney. *Résonance paramagnétique électronique des ions de transition*, chapter 3.2 L'effet de l'anisotropie sur le facteur g, formule (3,10). Presses Universitaires de France, Paris, 1972.
- [2] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55, page 919. U.S. Government Printing Office, Washington, D.C., 20402, 1964.
- [3] W. E. Ball. chapter BSOLVE: Solution to a set of nonlinear equations by Marquardt's method, in *Material and energy balance computations* by E. J. Henley and E. M. Rosen, pages 560–566. John Wiley and Sons, Inc, 1969.
- [4] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical Comp. Sci.*, 22:317–330, 1983.
- [5] M. Berz, C.H. Bischof, G.F. Corliss, and A. Griewank, editors. *Computational Differentiation: Applications, Techniques, and Tools*. SIAM, Philadelphia, 1996.
- [6] C. Bischof, A. Carle, P. Khademi, A. Mauer, and P. Hovland. Adifor2.0 user's guide. Argonne National Laboratory Technical Memorandum and CRPC Technical Report ANL/MCS-TM-192/CRPC-TR95516-S.
- [7] I. Charpentier and M. Ghemires. Efficient adjoint derivatives: Application to the atmospheric model Meso-NH. *Optimization Methods and Software*, 13(1):35–63, 2000.
- [8] C. Faure. Le gradient de THYC3D par Odysée. Rapport de recherche 3519, INRIA, October 1998.
- [9] C. Faure and Y. Papegay. Odysée User's Guide. Version 1.7. Rapport technique 0224, INRIA, September 1998.
- [10] R. Giering. *Tangent linear and Adjoint Model Compiler , Users manual*, 1997. Unpublished, available from <http://puddle.mit.edu/~ralf/tamc>.
- [11] A. Griewank and G.F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Applications*. SIAM, Philadelphia, 1991.
- [12] M. Iwasaki. Second-order perturbation treatment of the general spin hamiltonian in an arbitrary coordinate system. *Journal of Magnetic Resonance*, 16:417–423, 1974.
- [13] J. L. Kuester and J. H. Mize. *Optimization techniques with Fortran*, chapter Least squares objective functions, pages 240–250. McGraw-Hill, New-York, 1973.

- [14] W. D. Marquardt. An algorithm for least squares estimation of non linear parameters. *J.Soc.Indust.Appl.Math*, 11(2):413–441, 1963.
- [15] J.J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson, editor, *Lecture Notes in Mathematics*, volume 630, pages 105–116. Springer Verlag, Berlin, 1977.
- [16] J. Morgenstern. Algorithmes linéaires tangents et complexité. *Compte Rendu à l'Académie des Sciences*, t.277:367, September 1973.
- [17] J. Morgenstern. How to compute fast a function and all its derivatives, a variation on the theorem of Baur-Strassen. *Sigact News*, 16:60–62, 1985.
- [18] J.R. Pilbrow. *Transition ion electron paramagnetic resonance*, chapter 5.3.1 Powder averaged transition probability, formula (5,22). Clarendon Press, Oxford, 1990.
- [19] E. J. Soulié and P. C. Lesieur. Quantitative analysis of the electron paramagnetic resonance spectrum of a uranium (III) compound. *J. Chem. Soc. Faraday Trans. I*, 85(12):4053–4062, 1989.



Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399