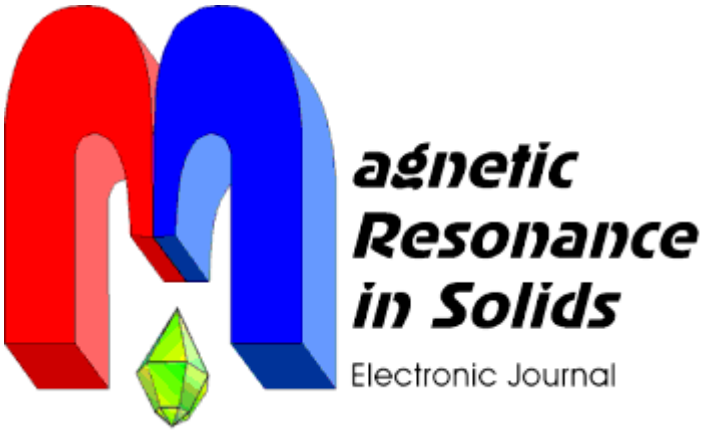
Application of Non-Linear Approximation for Electron Paramagnetic Resonance Spectrum Analysis: *Visual Spectrum Decomposer* Software

A.A. Levantovsky, A.V. Komolkin*

Saint Petersburg State University, Saint Petersburg 198504, Russia
* *E-mail*: levantovsky@mail.ru

Received February 22, 2008
Revised March 10, 2008
Accepted March 11, 2008



Volume **10**, *No.* **1**,
pages **20-24**, **2008**

http://mrsej.ksu.ru

Application of Non-Linear Approximation for Electron Paramagnetic Resonance Spectrum Analysis: *Visual Spectrum Decomposer* Software

A.A. Levantovsky*, A.V. Komolkin
Saint Petersburg State University, Ulianovskaya st., 1, Petrodvorets, Saint Petersburg 198504, Russia
**E-mail: levantovsky@mail.ru*

The new application of non-linear approximation for spectrum processing and multiplatform software for spectrum analysis with friendly user interface is suggested. Experimental spectrum is approximated by a sum of a number of functions. Each function can be approximated on selected interval separately when other functions are subtracted from experimental spectrum. All actions can be made using graphical user interface.

PACS: 87.80Lg, 76.30.-v

Keywords: spectrum analysis, spectrum processing, approximation, peak fitting, GUI, electron paramagnetic resonance, EPR

Introduction

The spectrum of electron paramagnetic resonance (EPR) consists of a number of spectrum lines. In common case these spectrum lines have different shapes and may overlap. The aim of the researcher is to single out separate lines and to find its parameters (position, amplitude, shape, width). In this paper the new spectrum analysis method and its implementation as graphical user interface (GUI) based application is suggested.

1. EPR spectrum analysis

For signal-to-noise ratio increase differential passing technique with double modulation is used in most spectrometers [1]. Therefore, the registered signal is the first derivative of EPR spectrum.

There are two basic line shape models (which are good for liquids): Gauss function and Lorentz function. Those functions differently agree with experimental spectrum lines of different samples. Spectrum lines of solids are asymmetric and have more complicated shape because of g-factor anisotropy.

For the spectrum analysis special software is used. Those software ordinary come with spectrometers, are very expensive and can be used only with spectra, recorded on one type of spectrometers. Freeware software has less functionality. Some separate operations of spectrum analysis can be made with general-purpose mathematical software, e.g. MatLab, gnuplot, Origin, QtiPlot, Fityk, but it takes a lot of time and may cause errors because of much handwork.

The aim of the effort is to develop techniques for separate lines determination and overlapping lines decomposition, and to implement it in user software with useful GUI.

2. Separate lines determination technique

Determination of a separate line is finding its shape, position, amplitude and width. This can be made by the non-linear approximation of an experimental spectrum by corresponding function (Gauss function, Lorentz function, etc.) In this work Levenberg-Marquardt algorithm is used as universal one. This algorithm is stable in most approximation functions and has reasonably convergence speed [2].

3. Overlapped lines decomposition technique

The EPR spectrum can enclose partly overlapped lines. In this case the following technique can be used. At the first step the broadest spectrum line is approximated. At the second step the obtained approximation function of the first line is subtracted from the experimental spectrum and the difference is used for approximation of the second line. To increase the accuracy, the first spectrum line is to be approximated while as the approximation function of second line is subtracted from the experimental spectrum. With this iteration algorithm more than two overlapping lines can be separated.

To increase the accuracy of the approximation it is essential to approximate one line of the spectrum only on the part of the experimental spectrum, where contribution of other spectrum lines is negligible. This requires setting of the approximation interval (or a number of intervals) for each spectrum line.

4. Implementation: *Visual Spectrum Decomposer* software

The outlined algorithm was implemented in Java language as the application *Visual Spectrum Decomposer* with GUI (old name is *EPRMultiPeak* software). Java is multiplatform interpreted programming language; therefore the application can be run on multiple operating systems (Microsoft Windows, Unix-like systems, MacOS) [3]. It is very useful because softwares of different spectrometers are running on different operating systems.

Each line of the spectrum (approximation sum component) is presented as a *curve* in terms of application. All curves are presented in table (fig. 1). After creation of the new curve the user have to set initial conditions of the approximation function. This can be done by moving the approximation function near to the experimental line using mouse (fig. 2). On the next step user sets the one or more approximation interval (fig. 3) for the line and run the approximation. When the approximation is done, the approximation function of new line can be subtracted from experimental spectrum and the other lines can be approximated.

The feature of the application is to evaluate the sum of all curves which approximates all spectrum. This sum can be compared with the experimental spectrum on one plot and the difference between experimental spectrum and the approximation sum (only noise in ideal) sum can be seen.

Sum of selected curves can be also approximated. The approximation of sum of all curves is traditional technique for spectrum analysis [4].

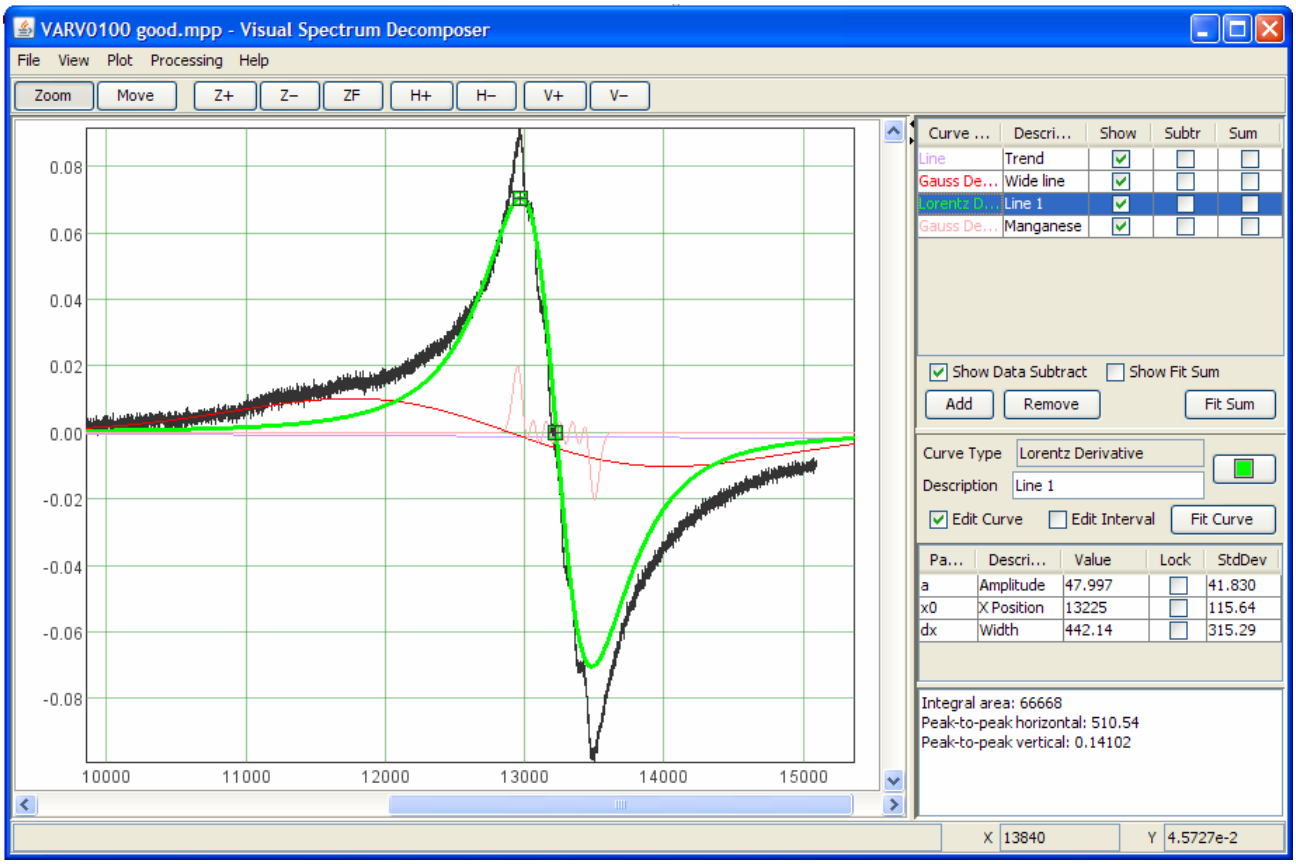


Fig. 1. *Visual Spectrum Decomposer* window

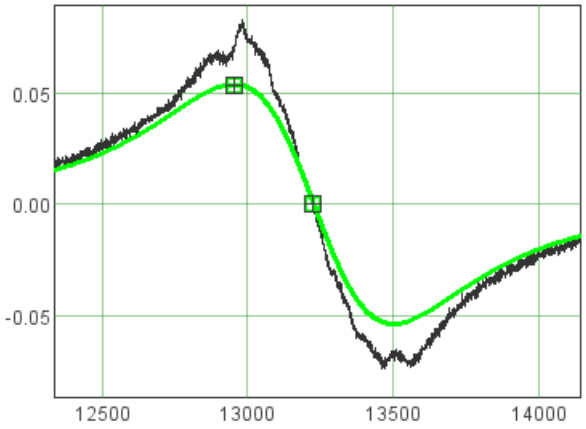


Fig. 2. Initial conditions setting by mouse

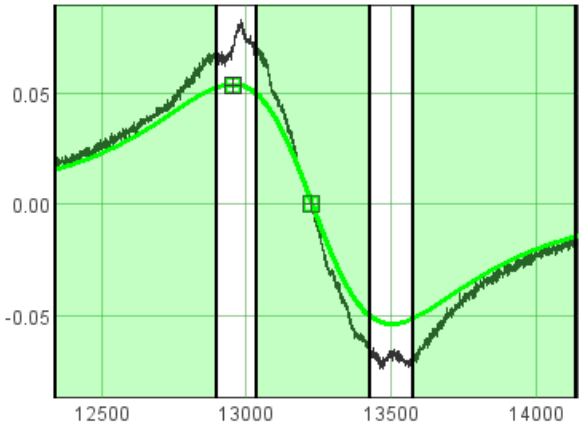


Fig. 3. Approximation interval setting by mouse

In the current version of the application the following types of lines are implemented: Gauss and Lorentz functions (for non-derivative spectra), derivatives of Gauss and Lorentz functions, Gauss derivative multiplet, Lorentz derivative multiplet, the line function, the parabolic function, the constant. The three latter are used for trend removing on the first stage of analysis.

The application is able to import spectra, which was recorded on different spectrometers. It can be used not only to process EPR spectra, but also for NMR spectra, optical spectra and other. To save results of the work experimental spectrum and approximation curves can be saved into the project file which can be opened later. The difference between the experimental spectrum and the selected curves can be exported as text file. Application provides also export of the plot in png format.

5. Spectrum processing example

The functioning of application was tested on a standard EPR spectrum of DPPH (di-phenyl-picrylhydrazyl), which spectrum line is good approximated by Lorentz function. The test showed agreement of the obtained line parameters with other sources and the right working of the application in a simple case. One of the processed spectra is the spectrum of archeological bone, which is shown in fig. 1. Fig. 1 shows also all curves.

Conclusion

The new technique of EPR spectrum processing and its realization is proposed. The algorithm and the Java-application allow to process spectrum visually and very easy. The application was tested on several experimental spectra.

In the current version of the application only Gauss and Lorentz functions are implemented. It is planned to implement the Foight function and asymmetric lines for solid state spectra.

The trial version of Visual Spectrum Decomposer can be downloaded from author's site (site is in Russian): http://electriq.nm.ru/vsd.html

References

1. Poole C.P. *Electron Spin Resonance. Comprehensive Treatise on Experimental Techniques*. Interscience Publ. (John Wiley & Sons) (1967).
2. Nosatch V. V. *Solving approximation problems on personal computers*. Moscow, "MIKAP" Publ. (1994) (*in Russian).*
3. http://java.sun.com
4. http://www.unipress.waw.pl/fityk/