



METHODOLOGY

Open Access

Implementing a new EPR lineshape parameter for organic radicals in carbonaceous matter

Mathilde Bourbin¹, Yann Le Du^{2*}, Laurent Binet² and Didier Gourier²

Abstract

Background: Electron Paramagnetic Resonance (EPR) is a non-destructive, non-invasive technique useful for the characterization of organic moieties in primitive carbonaceous matter related to the origin of life. The classical EPR parameters are the peak-to-peak amplitude, the linewidth and the g factor; however, such parameters turn out not to suffice to fully determine a single EPR line.

Results: In this paper, we give the definition and practical implementation of a new EPR parameter based on the signal shape that we call the R_{10} factor. This parameter was originally defined in the case of a single symmetric EPR line and used as a new datation method for organic matter in the field of exobiology.

Conclusion: Combined to classical EPR parameters, the proposed shape parameter provides a full description of an EPR spectrum and opens the way to novel applications like datation. Such a parameter is a powerful tool for future EPR studies, not only of carbonaceous matter, but also of any substance which spectrum exhibits a single symmetric line.

Reproducibility: The paper is a literate program—written using Noweb within the Org-mode as provided by the Emacs editor—and it also describes the full data analysis pipeline that computes the R_{10} on a real EPR spectrum.

Keywords: Electron paramagnetic resonance, Lineshape, Solid state chemistry, Carbonaceous matter, Exobiology, Literate programming, Python

Background: Necessity for a shape factor definition

In the field of exobiology, we need to determine the age of organic material in rock samples. Isotopic methods are commonly used to date the rock itself, but the organic matter may not be syngenetic with the rock. A novel solution based on Electron Paramagnetic Resonance (EPR) was proposed [1]; it requires the determination of a new EPR parameter, the R_{10} , from the EPR spectrum of the rock sample, from which the age can be computed from an empirical log-linear correlation that was uncovered in [1]. Knowing the distribution of the different parameters that contribute to the R_{10} , we may also provide a confidence interval for the age thus determined. In the following, we shall explain what the classical EPR parameters are and what the proposed new parameter brings to the table, and then describe the algorithm for the determination of the R_{10} : how to process the data files generated during

an EPR experiment, extract the classical EPR parameters and compute their distribution in order to have an estimate of their error; compute the new R_{10} parameter and its distribution from the preceding distributions. Thanks to this paper, scientists may themselves extract the R_{10} parameter from EPR data and use it not only for datation purposes but also to uniquely characterize the observed EPR spectrum lineshapes. Our goal is to automate a manual process that has proved scientifically successful yet cumbersome and tedious when applied on datasets that are getting larger. In that version of our code, some of our algorithmic choices just mirror the —successful— manual process. We have chosen the Python language because of its high level, ease of development and popularity; last but not least, it also provides powerful libraries for scientific development, and speed of execution turned out not to be a key factor for our goals^a. The Python code runs inside the Sage computing platform [2], which aims at providing a single computing environment both for numerical and symbolic computations.

*Correspondence: yann-ledu@chimie-paristech.fr

²Laboratoire de Chimie de la Matière Condensée de Paris, Ecole Nationale Supérieure de Chimie de Paris, UMR CNRS 7574, Paris, France
Full list of author information is available at the end of the article

Electron Paramagnetic Resonance (EPR) is a non-destructive and non-invasive technique which has indeed long been used for the study of paramagnetic defects (organic radicals) in carbonaceous materials. Such defects have been detected with high sensitivity in coals by pioneering EPR works [3]. These types of radicals were therefore used for the characterization of a wide range of carbonaceous objects, ranging from coals [4-6] to cherts [7] through meteorites [8-11]. The EPR signal of kerogen is a single line, due to the presence of aromatic radical moieties, with an unpaired electron spin delocalized in carbon p-type molecular orbitals [4,9,12,13]. Several parameters can be deduced from an EPR spectrum, based on the amplitude A_{pp} , the linewidth ΔB_{pp} and the resonance field B_{res} of the signal. However, for a single set of those three parameters, various lineshapes are possible (Figure 1); therefore, to fully determine the EPR line, a new EPR parameter, based on the lineshape, had to be defined.

The shape of the magnetic resonance absorption line of a system of interacting and randomly distributed spins depends on the nature of the interactions (dipole-dipole or exchange), on the spin concentration and on the dimensionality of the spatial distribution of the spins [14-18]. This study is restricted to the case of a dipole-dipole type interaction between electron spins, thus excluding exchange interaction occurring in very concentrated electron spin systems. Several limiting cases are distinguished in the literature, depending on the spin concentration and on the dimensionality of the distribution, cf. Table 1.

In the high concentration regime (generally considered when the fractional site occupation r by a paramagnetic centre exceeds 0.1), the lineshape is approximately Gaussian [17]. This regime also occurs when the line is broadened by unresolved hyperfine interaction. Given

Table 1 EPR lineshapes and lineshape parameter R_{10} for different limit regimes of dipolar broadening

Spin concentration	Distribution	Lineshape	R_{10}
High: dipolar and hyperfine broadening	3D	Gaussian to Lorentzian	≥ 0
Low: Lorentzian	0		
	2D	Stretched Lorentzian	-1.78
	1D	Stretched Lorentzian	-2.95

that EPR experimental spectra correspond to absorption derivatives, the Gaussian EPR line is described by:

$$F_G(B - B_{res}) = -A_{pp} \frac{B - B_{res}}{\Delta B_{pp}} \exp \left[-2 \left(\frac{B - B_{res}}{\Delta B_{pp}} \right)^2 + \frac{1}{2} \right] \quad (1)$$

where B is the applied magnetic field, B_{res} the field at the centre of the line (maximum of absorption), A_{pp} the peak-to-peak amplitude and ΔB_{pp} the peak-to-peak linewidth (Figure 1).

In the low concentration regime (generally considered when $r < 0.01$) with no hyperfine broadening, the lineshape depends on the dimensionality of the spatial distribution of the paramagnetic centres [16]. When the distribution is random, the resonance line may be calculated from the relaxation function:

$$G(d, t) = B \exp \left(-a \cdot t^{\frac{d}{3}} \right) \quad (2)$$

This function describes the decay with time t of the spin magnetization, perpendicular to the magnetic field, after an infinitely short microwave pulse. Parameter a is a

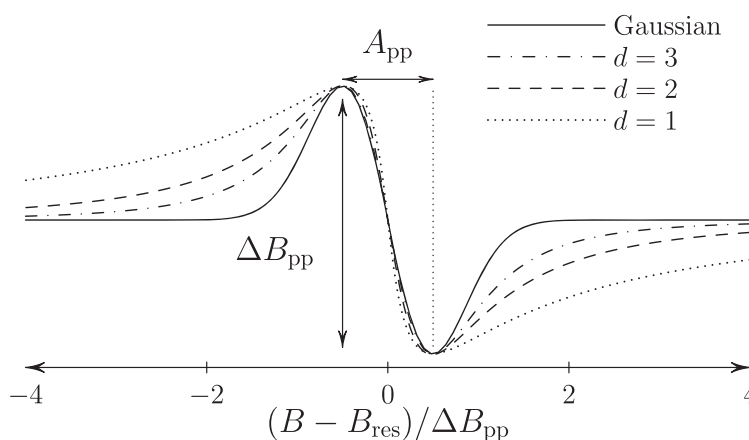


Figure 1 Theoretical EPR lines corresponding to upper limit cases of dipolar broadening. Continuous line: high spin concentration regime (Gaussian lineshape); Mixed line: diluted spin regime and 3D distribution (Lorentzian lineshape); Dashed line: diluted regime, 2D distribution (stretched Lorentzian); Dotted line: diluted regime, 1D distribution (stretched Lorentzian).

constant that depends linearly on the spin concentration and parameter d represents the dimensionality of the spin distribution: $d = 1$ for a linear distribution, $d = 2$ for a distribution in a plane and $d = 3$ for a distribution in a volume. The EPR absorption is the Fourier transform of the relaxation function, and thus the EPR spectrum is the field derivative of this Fourier transform:

$$F_d(B - B_{\text{res}}) = \Re \left[\int_0^{+\infty} G(d, t) \left(-it \frac{g\beta}{\hbar} \right) \times \exp \left[-i(B - B_{\text{res}})t \frac{g\beta}{\hbar} \right] dt \right] \quad (3)$$

where \Re stands for the real part. In the case of a three dimensional distribution ($d = 3$), the EPR lineshape function can be analytically calculated and corresponds to the field derivative of a Lorentzian function:

$$F_3(B - B_{\text{res}}) = -\frac{16}{9} A_{\text{pp}} \frac{\frac{(B - B_{\text{res}})}{\Delta B_{\text{pp}}}}{\left[1 + \frac{4}{3} \left(\frac{(B - B_{\text{res}})}{\Delta B_{\text{pp}}} \right)^2 \right]} \quad (4)$$

For lower dimension of spin spatial distribution ($d < 3$), the Fourier transform can only be calculated numerically. Figure 1 shows the theoretical EPR spectra corresponding to the Gaussian, Lorentzian ($d = 3$) and low dimensional ($d = 1$ and 2) cases. The wings of a Gaussian line fall off faster than those of a Lorentzian line while the wings of an EPR spectrum corresponding to a low-dimensional distribution fall off more slowly, giving rise to a so-called stretched Lorentzian lineshape. Originally, the R_{10} lineshape factor was imagined after studying the spectra in a coordinate system (x, y) in which the difference between the lineshapes stands out more clearly [14], and where the Lorentzian becomes a straight line:

$$f_L(x) = x + \frac{3}{4} \quad (5)$$

and the Gaussian shape by an increasing exponential:

$$f_G(x) = \exp\left(x - \frac{1}{4}\right) \quad (6)$$

with $f_G(x) \geq f_L(x), \forall x$, cf. Figure 2. That coordinate system can be obtained thanks to the following transformations as given in [14]:

$$x_{\text{Benc}} = \left(\frac{B - B_{\text{res}}}{\Delta B_{\text{pp}}} \right)^2 \text{ and} \quad (7)$$

$$y_{\text{Benc}} = f(x_{\text{Benc}}) = \sqrt{\frac{A_{\text{pp}}}{|F(B - B_{\text{res}})|} \frac{|B - B_{\text{res}}|}{\Delta B_{\text{pp}}}}$$

where $F = F_G$ or F_d . We shall thus define two functions, one that creates the new abscissas from the old $x \equiv B$ and the other that creates the new ordinates from the old x and $y \equiv F(B - B_{\text{res}})$:

```

3
<Functions to transform coordinates 3> ≡ (11a)
def xTransform(x, Bres, DeltaBpp):
    return (x - Bres) / DeltaBpp ^ 2

def yTransform(x, y, App, Bres, DeltaBpp):
    return sqrt(App * abs(x - Bres) /
    (DeltaBpp * abs(y)))
    
```

Following the Noweb literate programming style as described in [19], the above code is called a *code chunk*, with a unique name given between angle brackets and followed with an equal sign, together with a corresponding unique number made up of the page number and a letter starting at *a* and increasing alphabetically on a given page;

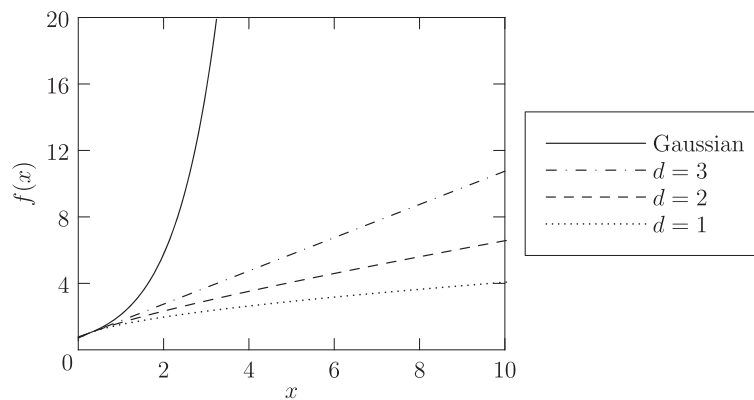


Figure 2 Representation of the EPR spectra in the new $(x_{\text{Benc}}, y_{\text{Benc}})$ coordinates system described by [14] and given in equation (7). Continuous line: Gaussian; mixed line: 3D distribution (Lorentzian); dashed line: 2D distribution (stretched Lorentzian); dotted line: 1D distribution (stretched Lorentzian).

that number is mirrored in the left margin for easy reference. The number on the end of line after the code chunk name indicates the code chunk where the current code chunk is used. Often, we shall add some code to an already existing code chunk, and that will appear in two different ways: first, the name between angle brackets will be followed by an equal sign attached to a plus sign (instead of a lone equal sign), and the numbers on the end of line will also indicate where the code chunk gets some new code (a small triangle is added to that number, i.e. < for previous existing definition, and > for the next new code).

For diluted spin systems with low-dimensional distribution, the representative function f lies below the line corresponding to a Lorentzian shape. To quantitatively characterize the lineshape for systems intermediate between the above four ideal cases [Gaussian, Lorentzian ($d = 3$), one-dimensional ($d = 1$) and two-dimensional ($d = 2$)], we define a lineshape parameter measuring the deviation from a Lorentzian line as described in [7]:

$$R_{10} = \frac{1}{10} \int_{x=0}^{x=10} [f(x) - f_L(x)] dx \quad (8)$$

This parameter corresponds to the algebraic surface between the curve f representing an experimental EPR spectrum and the curve f_L representing a Lorentzian line. R_{10} is negative for a low-dimensional distribution ($d < 3$) and positive for an EPR line intermediate between Lorentzian and Gaussian lines (Table 1). The integration in equation (8) must be restricted to a finite range of x -values for the integral may not converge when $x \rightarrow \infty$. In practice, the range is limited to $x \leq 10$, since in most cases encountered the signal-to-noise ratio of the EPR spectra is poor for $x \geq 10$, inducing strong fluctuations in f and consequently in the lineshape parameter. Also, because of spectra with left/right asymmetry, the final R_{10} is the average of the values computed on the left and right of the resonance field, i.e.

$$R_{10} = \frac{1}{2} (R_{10} + R_{-10}) \quad (9)$$

To compute the integral in equation (8), we shall follow the method originally used: a simple top-left corner rectangular approximation. That allows full reproducibility with the original manual method that was used before automation with a program; in the future we may replace it with a more accurate algorithm if there is a general agreement on the need to depart from the manual processing. We shall thus consider a matrix `matrixXYL`—a numpy array—made up of the abscissas of the spectrum in the first column, the ordinates of the spectrum in the second column, and the ordinates of the

ideal Lorentzian in the third column, with the number of lines corresponding to the number of data-points on the curves:

```
4a <R10 rectangular integral 4a> ≡ (5)
Xspacing = (matrixXYL[:,0][1:]
            - matrixXYL[:,0][:-1])
Ydifference = (matrixXYL[:,1][:-1]
              - matrixXYL[:,2][:-1])
Xspan = (matrixXYL[:,0][-1]
         - matrixXYL[:,0][0])
R10 = sum(Xspacing*Ydifference)
     / Xspan
```

The `matrixXYL` will be defined as a numpy array, and we use the `sum` function from the same library:

```
4b <Import useful pylab functions 4b> ≡ (12) 5b>
from pylab import sum
```

In order to construct the matrix `matrixXYL`, we need the data abscissas and ordinates and we use equation 5 for the `yL` coordinates of the ideal Lorentzian curve:

```
5a <Matrix of XYL values in new coordinates 5a> ≡ (5)
matrixXYL = array([[xm,ym,yL] for x,y
                  in zip(abscissas,ordinates)\
                  for xm,ym in [(xTransform(x, Bres,
                  DeltaBpp), yTransform(x, y, App,
                  Bres, DeltaBpp))])
                  for yL in [xm + 3/4.] if xm < 10 and
                  <Same side of Bres 5c >])
```

Again, we need to use the array data-structure, so we import it:

```
5b <Import useful pylab functions 4b>+ ≡ (12) <4b 6f>
from pylab import array
```

Operationally, the R_{10} was only defined separately for the parts of the curve which abscissas x are larger or smaller than the resonance field `Bres`, and we thus define an operator `testSameSideofBres` that will enable us to build two matrices `matrixXYL`, one for each side:

```
5c <Same side of Bres 5c> ≡ (5a)
testSameSideofBres(x, Bres)
```

In the case of the left hand side, we look for x lower than `Bres`, and the opposite for the right hand side:

```
5d <Left matrix of XYL values 5d> ≡ (5h) 6a>
testLeftSideofBres = lambda x,
                      Bres: x < Bres
testSameSideofBres = testLeftSideofBres
<Matrix of XYL values in new coordinates 5a >
```

5e *<Right matrix of XYL values 5e>* \equiv (5i)
`testRightSideofBres = lambda x,
 Bres: x >
 Bres testSameSideofBres =
 testRightSideofBres
 <Matrix of XYL values in new coordinates 5a>`

We shall thus obtain two values of R_{10} , one for each side of B_{res} ,

5f *<Compute R10 5f>* \equiv (11a)
`<R10 on the left 5h>
 <R10 on the right 5i>
 <Average R10 5g>`

and we shall then use their average as the final value for the spectrum under study, cf. equation (9):

5g *<Average R10 5g>* \equiv (5f)
`R10 = (R10_left + R10_right) / 2`

5h *<R10 on the left 5h>* \equiv (5f)
`<Left matrix of XYL values 5d>
 <R10 rectangular integral 4a>
 <Save R10 left value 5j>`

5i *<R10 on the right 5i>* \equiv (5f)
`<Right matrix of XYL values 5e>
 <R10 rectangular integral 4a >
 <Save R10 right value 5k>`

5j *<Save R10 left value 5j>* \equiv (5h)
`R10_left = R10`

5k *<Save R10 right value 5k>* \equiv (5i)
`R10_right = R10`

We need to be careful with the order of the values in the matrix giving the coordinates in the new coordinate system defined in equation (7): if we start from small values of x in the original frame, then, for the left hand side of B_{res} , values in the new frame will decrease, whereas values on the right hand side will increase. Thus, values on the left side must be reversed, whereas that will not be necessary for the right hand side.

6a *<Left matrix of XYL values 5d>* \equiv (5h) <5d
`<Reverse matrix 6b>` \equiv

6b *<Reverse matrix 6b>* \equiv (6a)
`matrixXYL = matrixXYL[::-1]`

Methods

All the relevant discussion about the experimental part of the work, that involves collecting EPR data on the

rock samples, can be found in [1]. In the current paper, we focus on the specific data handling and processing in order to extract the R_{10} parameter from an EPR spectrum and estimate the associated error. All computations were made in the Sage computing environment [2], with imports from the Numeric Python library [20].

In the spirit of reproducible research [21], the paper is written in the literate programming style [22]: the code and its explanation^b are intertwined in a single place, and a particular program is then used to extract either the source code for execution on a computer or the literate paper for reading by humans. Literate programming tools exist, and we use Noweb [19] and Org-mode [23,24] within Emacs with Evil mode to enable vi commands. We also make use of the Sagetex package that comes with the Sage distribution, that allows Sage code to be executed when compiling the LaTeX source of the paper^c, and we have a home-built script that manages to combine Org-mode with Sagetex together with a Noweb output. Figures are produced either with Sage and Sagetex, or with Asymptote: it allows us to program figures, and thus make them executable, and embeddable in the LaTeX source code. The code will be made available through the team's website^d.

Processing data from an EPR file

Removing the background signal

EPR spectra on which the R_{10} factor was to be measured were selected for their symmetric and well-defined single absorption derivative signal. As usual in EPR studies, the large scale background signal was subtracted with a third degree polynomial fitted on the smooth parts of the spectrum where the signal variations are only due to noise, which in practice correspond to the first and last 10% data points in a typical spectrum.

6c *<Remove background 6c>* \equiv (12)
`<Compute number of data points in tails 6d>
 <Define points in tails 6e>
 <Compute polynomial on tails 6g>
 <Subtract polynomial from spectrum 7a>`

6d *<Compute number of data points in tails 6d>* \equiv (6c)
`numPointsTails = ceil(len(abscissas) / 10.)`

6e *<Define points in tails 6e>* \equiv (6c)
`tailX = concatenate
 ((abscissas[:numPointsTails],
 abscissas[-numPointsTails:]))
 tailY = concatenate
 ((ordinates[:numPointsTails],
 ordinates[-numPointsTails:])))`

```
6f <Import useful pylab functions 4b>+ ≡ (12) <5b 6h>
    from pylab import concatenate
```

```
6g <Compute polynomial on tails 6g> ≡ (6c)
    degreePoly = 3
    backPoly = polyfit(tailX, tailY,
        degreePoly)
```

```
6h <Import useful pylab functions 4b>+ ≡ (12) <6f 7b>
    from pylab import polyfit
```

From now on, the spectrum will be understood as the baseline corrected raw spectrum.

```
7a <Subtract polynomial from spectrum 7a> ≡ (6c)
    ordinates -= polyval(backPoly, abscissas)
```

```
7b <Import useful pylab functions 4b>+ ≡ (12) <6h 7f>
    from pylab import polyval
```

Reading the data for the spectra

EPR Spectra are given as .txt files, with a name made up of the following informations:

- sample-name
- temperature-of-acquisition
- microwave-power
- number-of-scans

For example, `gunflint_ambient_2mW_1scan.txt` corresponds to a sample named `gunflint`, studied at ambient temperature with a microwave power of 2mW using 1 scan^e.

```
7c <Load data 7c> ≡ (12)
    <Define DATA directory 7g>
    <Define filename 7d>
    <Extract abscissas and ordinates 7e>
```

```
7d <Define filename 7d> ≡ (7c)
    fileName =
        'MB_gunflint_ambient_2mW_1scan.txt'
```

The first two lines must be skipped when loading data: they provide the EPR acquisition parameters and the file description. EPR text files comprise three columns, giving respectively the point index (starting from one and running to the total number of points recorded), the datapoint abscissa —the magnetic field B — and the datapoint ordinate —the intensity in arbitrary units. To ease data manipulation we extract two lists, `abscissas` and `ordinates`.

```
7e <Extract abscissas and ordinates 7e> ≡ (7c)
    data = loadtxt(DATA+fileName, skiprows=2)
    #Skip first two comment lines
```

```
abscissas = data[:,1].copy()
ordinates = data[:,2].copy()
```

and the load function `loadtxt` will be taken from the `pylab` library.

```
7f <Import useful pylab functions 4b>+ ≡ (12) <7b 8h>
    from pylab import loadtxt
```

We also have to make sure that the `DATA` variable is defined, which is normally automatic within Sage:

```
7g <Define DATA directory 7g> ≡ (7c)
    try:
        DATA
    except NameError:
        DATA = 'data/'
```

In order to plot the spectrum as in Figure 3, we use Sage builtin plot function `list_plot`.

```
7h <Plot spectrum 7h> ≡ (10g 12)
    spectrumPlot = list_plot(zip(abscissas,
        ordinates))
    spectrumPlot.save
        (DATA+'spectrumPlot.png')
```

The distribution of the classical EPR parameters

To uncover the underlying Lorentzian curve which will be compared to the original spectrum for the R_{10} computation, we need to find the three parameters that determine the latter: the peak-to-peak amplitude A_{pp} , the linewidth ΔB_{pp} and the resonance field B_{res} . We define the peaks (positive and negative) as the extrema of the spectrum ordinate values, and the A_{pp} and ΔB_{pp} as the difference between the peaks' ordinates and abscissas, respectively.

```
8a <Compute App 8a> ≡ (8d)
    Amax = max(ordinates)
    Amin = min(ordinates)
    App = Amax - Amin
```

```
8b <Compute DeltaBpp 8b> ≡ (8d)
    indexAmax = list(ordinates).index(Amax)
    Bmin = abscissas[indexAmax]
    indexAmin = list(ordinates).index(Amin)
    Bmax = abscissas[indexAmin]
    DeltaBpp = Bmax-Bmin
```

The resonance field B_{res} was defined as the value at which the EPR lineshape crosses the baseline of the spectrum, which corresponds to the zero axis since the spectra are baseline corrected.

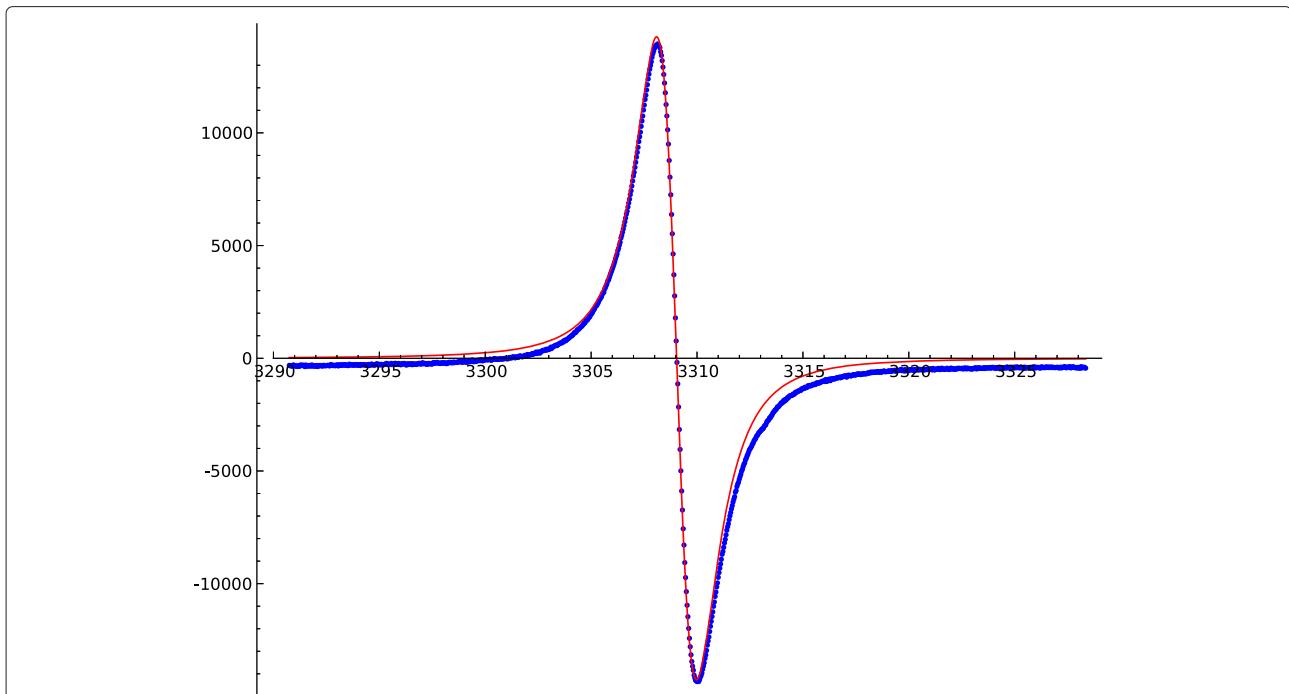


Figure 3 The loaded EPR spectrum (dots) and the corresponding theoretical Lorentzian (continuous): the R_{10} factor is based on the integral difference between the two, cf. equation (8).

```
8c <Compute Bres 8c> ≡
    for y in
        <The ordinates between the two extrema 8e>:
        <Find when ordinate crosses baseline 8f>
        Bres =
            <Mean of the two ordinates above and below
            baseline 8g>
```

```
8d <Compute the classical EPR parameters 8d> ≡ (8d)
    <Compute App 8a>
    <Compute DeltaBpp 8b>
    <Compute Bres 8c>
```

```
8e <The ordinates between the two extrema 8e> ≡ (8c)
    ordinates[indexAmax:indexAmin+1]
```

```
8f <Find when ordinate crosses baseline 8f> ≡ (8c)
    if y < 0:
        yCross = y
        break
```

The resonance field Bres is thus the mean of the two ordinates lying above and below the baseline respectively:

```
8g <Mean of the two ordinates above and below
baseline 8g> ≡ (8c)
    mean(map(lambda v: abscissas[list
        (ordinates).index(yCross) + v],
        [-1, 0]))
```

```
8h <Import useful pylab functions 4b>+ ≡ (12)<7f 9b>
    from pylab import mean
```

Knowing the distributions of the classical EPR parameters App, DeltaBpp and Bres, we may check visually their normality thanks to a histogram plot; if normal, we may propagate their standard deviation in the global R_{10} error calculation.

```
9a <Plot histograms of the classical EPR
parameters 9a> ≡ (9g)
    setEPRparams = [listApp, listDeltaBpp,
        listBres]
    for i in range(len(setEPRparams)):
        clf()
        hist(setEPRparams[i])
        savefig(DATA+'distrib'+ str(i))
```

```
9b <Import useful pylab functions 4b>+ ≡ (12)<8h 9d>
    from pylab import clf, hist, savefig
```

```
9c <Compute the moments of the classical EPR
parameters 9c> ≡ (9g)
    [App, DeltaBpp, Bres] =
        map(mean, setEPRparams)
    [sigmaApp, sigmaDeltaBpp, sigmaBres] =
        map(std, setEPRparams)
```

```
9d <Import useful pylab functions 4b>+ ≡ (12)<9b 9f>
    from pylab import std
```



```
lorenzPlot.save
    (DATA+'lorenzPlot.png')
```

```
10g <Plot spectrum and Lorentzian 10g> ≡ (12)
    <Plot spectrum 7h>
    <Plot Lorentzian 10f>
    dlPlot = spectrumPlot + lorentzPlot
    dlPlot.save (DATA+'dlPlot.png')
```

Now the R_{10} parameter is computed relatively to the theoretical Lorentzian having the same set of classical EPR parameters, so we could compute the error on the former by propagating analytically the errors of the latter, which we now know thanks to the previous application of the Monte Carlo error propagation method. However, we found it easier and somewhat more in line with the computational approach to use a Monte Carlo approach to propagate the errors. We thus need to repeat the R_{10} computation for a series of values of B_{res} , ΔB_{pp} and A_{pp} to which we add a random error compatible with their distributions^g:

```
11a <R10 distribution computation 11a> ≡ (12)
    <Functions to transform coordinates 3>
    <Create R10 list 11e>
    <Backup classical EPR parameters 11b>
    <Repeat a large number of times 10b>
    <Add noise to classical EPR parameters 11d>
    <Compute R10 5f>
    <Store R10 11f>
    <Retrieve original classical EPR parameters 11c>
    <Compute mean and standard deviation
    for R10 11g>
```

Because we modify the classical parameters during the R_{10} computation, we need to store the values and retrieve them before and after each iteration of the Monte Carlo:

```
11b <Backup classical EPR parameters 11b> ≡ (11a)
    backupEPR = [App, DeltaBpp, Bres]
```

```
11c <Retrieve original classical EPR parameters 11c> ≡ (11a)
    [App, DeltaBpp, Bres] = backupEPR
```

```
11d <Add noise to classical EPR parameters 11d> ≡ (11a)
    App += randn()*sigmaApp
    DeltaBpp += randn()*sigmaDeltaBpp
    Bres += randn()*sigmaBres
```

```
11e <Create R10 list 11e> ≡ (11a)
    listR10 = []
```

```
11f <Store R10 11f> ≡ (11a)
    listR10.append(R10)
```

```
11g <Compute mean and standard deviation
    for R10 11g> ≡ (11a)
    R10 = mean(listR10)
    sigmaR10 = std(listR10)
```

Results and conclusion

We now have extracted the R_{10} parameter together with its distribution and may proceed to use it, for example to determine the age of organic matter inside rock samples [1]. Given the distribution, we may then check if the mean and standard error do indeed properly characterize the parameter, and eventually assign a probability to a range of ages for the rock sample. The code runs in only a few minutes, if we take into account all the Monte Carlo computations. In [1], we demonstrate that the data processing as reported here can indeed provide us with a reasonable estimate for the age of rock samples older than 1 billion years.

The complete code

```
12 <r10.py 12> ≡
    <Import useful pylab functions 4b>
    <Define global variables 10a>
    <Load data 7c>
    <Remove background 6c>
    <Plot spectrum 7h>
    <Distribution of the classical EPR parameters 9g>
    <Plot spectrum and Lorentzian 10g>
    <R10 distribution computation 11a>
```

Endnotes

^aAnyway, tools exist to go faster when needed, as Cython inside Sage that allows easy variable typing.

^bOr maybe the explanation and its code... literate programming is really a whole new approach to writing, thinking and coding.

^cThis means that the outputs of some code need not be pasted inside the paper, but can be computed on the fly as needed.

^dThe url is <http://hpu4science.org>.

^eThis sample is part of the study where the R_{10} parameter was proposed as a datation method [1].

^fThis corresponds to a normal distribution arising from a Poisson distribution, and is the common practice in EPR because of the underlying counting process when measuring the absorption giving the spectrum. We can indeed check it is so by studying the noise on the flat tails of EPR spectra.

^gUsing the Monte Carlo approach would also allow us to draw the values for the classical parameters according to their computed distribution.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MB carried out the experiments and did the analysis; MB and YLD developed the code and adapted the existing analysis framework to the new design, and wrote the paper. YLD initiated the move to a fully automated analysis framework for extracting the R_{10} and proposed the Python language within the Sage environment. LB and DG designed the original R_{10} parameter, corresponding manual extraction procedure and application to datation; they also took part in the analysis and helped with the experiments and the paper. All authors read and approved the final manuscript.

Acknowledgements

All the people who contributed substantially to the work are co-authors.

Chunks

{Add noise to classical EPR parameters 11d}
{Add noise to data 9e}
{Average R10 5g}
{Backup classical EPR parameters 11b}
{Backup data 10c}
{Compute App 8a}
{Compute Bres 8c}
{Compute DeltaBpp 8b}
{Compute mean and standard deviation for R10 11g}
{Compute number of data points in tails 6d}
{Compute polynomial on tails 6g}
{Compute R10 5f}
{Compute the classical EPR parameters 8d}
{Compute the moments of the classical EPR parameters 9c}
{Create R10 list 11e}
{Create the lists for the classical EPR parameters 9h}
{Define DATA directory 7g}
{Define filename 7d}
{Define global variables 10a}
{Define points in tails 6e}
{Define theoretical Lorentzian 10e}
{Distribution of the classical EPR parameters 9g}
{Extract abscissas and ordinates 7e}
{Find when ordinate crosses baseline 8f}
{Functions to transform coordinates 3}
{Import useful pylab functions 4b}
{Left matrix of XYL values 5d}
{Load data 7c}
{Matrix of XYL values in new coordinates 5a}
{Mean of the two ordinates above and below baseline 8g}
{Plot histograms of the classical EPR parameters 9a}
{Plot Lorentzian 10f}
{Plot spectrum 7h}
{Plot spectrum and Lorentzian 10g}
{R10 distribution computation 11a}
{R10 on the left 5h}
{R10 on the right 5i}
{R10 rectangular integral 4a}
{r10.py 12}
{Remove background 6c}
{Repeat a large number of times 10b}
{Retrieve original classical EPR parameters 11c}
{Retrieve original data 10d}
{Reverse matrix 6b}
{Right matrix of XYL values 5e}
{Same side of Bres 5c}
{Save R10 left value 5j}
{Save R10 right value 5k}
{Store R10 11f}
{Store the parameters 9i}
{Subtract polynomial from spectrum 7a}
{The ordinates between the two extrema 8e}

Author details

¹Biogéochimie et Ecologie des Milieux Continentaux, UMR CNRS 7618, Université Pierre et Marie Curie, Paris, France. ²Laboratoire de Chimie de la

Matière Condensée de Paris, Ecole Nationale Supérieure de Chimie de Paris, UMR CNRS 7574, Paris, France.

Received: 13 June 2012 Accepted: 18 June 2013
Published: 17 July 2013

References

1. Bourbin M, Gourier D, Derenne S, Binet L, Le Du Y, Westall F, Kremer B, Gautret P: **Dating carbonaceous matter in archean cherts by electron paramagnetic resonance.** *Astrobiology* 2013, **13**(2):151–162.
2. Stein WA, et al: *Sage Mathematics Software (Version 4.8.0)*: The Sage Development Team, 2012. [http://www.sagemath.org]
3. Uebersfeld J, Erb E: **A new effect of oxygen on the paramagnetic resonance of charcoal.** *CR Acc Sci* 1956, **243**:363–364.
4. Mrozowski S: **EsR studies of carbonization and coalification processes part i: carbonaceous compounds.** *Carbon* 1988, **26**:521–529.
5. Mrozowski S: **EsR studies of carbonization and coalification processes part ii: biological materials.** *Carbon* 1988, **26**:531–541.
6. Retcofsky HL, Stark JM, Friedel RA: **Electron spin resonance in american coals.** *Anal Chem* 1968, **40**:1699–1704.
7. Skrzypczak-Bonduelle A, Binet L, Delpoux O, Vezin H, Derenne S, Robert F, Gourier D: **Epr of radicals in primitive organic matter: a tool for the search of biosignatures of the most ancient traces of life.** *App Magn Reson* 2008, **33**:371–397.
8. Binet L, Gourier D, Derenne S, Robert F: **Heterogeneous distribution of paramagnetic radicals in insoluble organic matter from the orgueil and murchison meteorites.** *Geochim Cosmochim Acta* 2002, **66**:4177–4186.
9. Binet L, Gourier D, Derenne S, Robert F, Crofini I: **Occurrence of abundant diradicaloid moieties in the insoluble organic matter from the orgueil and murchison meteorites: a fingerprint of its extraterrestrial origin?** *Geochim Cosmochim Acta* 2004, **68**:881–891.
10. Delpoux O, Gourier D, Vezin H, Binet L, Derenne S, Robert F: **Biradical character of the d-rich carriers in the insoluble organic matter of carbonaceous chondrites: a relic of the protoplanetary chemistry.** *Geochim Cosmochim Acta* 2011, **75**:326–336.
11. Gourier D, Robert F, Delpoux O, Binet L, Vezin H, Moissette A, Derenne S: **Extreme deuterium enrichment of organic radicals in the orgueil meteorite: revisiting the interstellar interpretation?** *Geochim Cosmochim Acta* 2008, **72**:1914–1923.
12. Dickneider TA, Scull S, Whelan JK, Blough NV: **Epr study of kerogens from middle valley, northern juan de fuca ridge.** *Organic Geochem* 1997, **26**:341–352.
13. Uebersfeld J, Etienne A, Combrisson J: **Paramagnetic resonance, a new property of coal-like materials.** *Nature* 1954, **174**:614–614.
14. Bencini A, Gatteschi D: **Exchange narrowing in lower dimensional systems.** In *EPR of exchange coupled systems*. Edited by Lewin RA. Berlin: Springer-Verlag; 1990.
15. Drabold DA, Fedders PA: **Dipolar broadening in magnetically diluted lattices.** *Phys Rev B* 1988, **37**:3440–3447.
16. Fel'dman EB, Lacelle S: **Configurational averaging of dipolar interactions in magnetically diluted spin networks.** *J Chem Phys* 1996, **104**:2000–2010.
17. Kittel C, Abrahams E: **Dipolar broadening of magnetic resonance lines in magnetically diluted crystals.** *Phys Rev* 1953, **90**:238–239.
18. Van Vleck JH: **The dipolar broadening of magnetic resonance lines in crystals.** *Phys Rev* 1948, **74**:1168–1183.
19. Ramsey N: **Literate programming simplified.** *IEEE Software* 1994, **11**:97–105. ISSN 0740-7459. [http://dx.doi.org/10.1109/52.311070]
20. Oliphant TE: **Python for scientific computing.** *Comput Sci & Engnr* 2007, **9**(3):10–20. [http://doi.ieeecomputersociety.org/10.1109/MCSE.2007.58]
21. Donoho DL: **An invitation to reproducible computational research.** *Biostatistics* 2010, **11**(3):385–388.
22. Knuth DE: **Literate programming.** *Comput J* 1984, **27**:97–111.
23. Schulte E, Davison D, Dye T, Dominik C: **A multi-language computing environment for literate programming and reproducible research.** *J Stat Softw* 2012, **46**(3):1–24. ISSN 1548-7660. [http://www.jstatsoft.org/v46/i03]

24. Dominik D: *The Org-Mode 7 Reference, Manual: Organize Your Life with GNU Emacs*. UK: Network Theory; 2010. with contributions by David O'Toole, Bastien Guerry, Philip Rooke, Dan Davison, Eric Schulte, and Thomas Dye.
25. BIPM, IEC and IFCC, ILAC and ISO, IUPAC, IUPAP and OIML: **"Evaluation of measurement data—supplement 1 to the 'Guide to the expression of uncertainty in measurement'—propagation of distributions using a Monte Carlo method"**. *Joint Committee for Guides in Metrology, JCGM* 2008, **101**.

doi:10.1186/1751-0473-8-15

Cite this article as: Bourbin *et al.*: Implementing a new EPR lineshape parameter for organic radicals in carbonaceous matter. *Source Code for Biology and Medicine* 2013 **8**:15.

**Submit your next manuscript to BioMed Central
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

