Multi-Exponential Fit of Experimental Data: A Robust Method for Pore Size Distribution Determination

A. M. Niell, C. A. Martín*, E. Ramia
Facultad de Matemática, Astronomía y Física - Universidad Nacional de Córdoba - Ciudad Universitaria - X5016LAE Córdoba - Argentina
martin@famaf.unc.edu.ar

Keywords: pore size distribution, NMR relaxometry

Abstract: *A large variety of data, resulting from physical and chemical experiments, may be well described by a linear combination of decaying exponentials, such as those related to the distribution pore size of oil rocks. The relevant information sought is contained in the spectrum or mathematical transform, that is, the dependence of the coefficients in the linear combination with respect to the time constants of the exponentials in the basis. Computing the transform is a problem which, generally, requires regularization, and that has been previously analyzed by a number of authors. This work reports a comprehensive study of multi-exponential, fit where we have introduced two major improvements. First, the direct resolution of linear systems in the least-squares sense using the QR formalism instead of the normal equations, which efficiently reduces rounding errors, always present in the resolution of this type of ill–defined problems, and second, a robust and simple criterion to select the best value for the regularization parameter is given, which strongly improves the existing previous ones. It is important to remark that this method not only fits mult-exponential decays, but those related to natural processes where logo-normal exponential distributions are found rather than a collection of discrete exponentials.*

Introduction

We deal with the computation of a discrete Laplace transform of decaying functions of time (signals) for which a set of measured values is known. This situation is frequently found in physical and chemical data analyses. Nuclear Magnetic Resonance (NMR) experiments in certain cases (relaxometry), laser induced chemical reactions, etc., produce time dependent signals that may be described as a sum of decaying exponentials. Proper analysis of these data provides unique insight into, for example, pore size distribution in porous materials, oil rocks, the solid/liquid fraction in butter and animal fats, and various time constants involved in chained chemical reactions.[1, 7]

The computation of transforms is required because of their important theoretical and practical properties, but in the numerical practice some substantial shortcomings must be considered: the set of times for which the signal is known is finite, the known values are contaminated by errors, and furthermore the transform is not computed for a continuous range of exponents but rather only for a finite set of “time constants”.

Basic statement of the problem

Let n measured values of a signal $s(t)$ correspond to as many different observation times, as expressed by the set of pairs:

$$\{(t_i, s_i) \; ; \; i = 1, \ldots, n\} \qquad (1)$$

It is assumed that the t_i sequence is strictly increasing. This is not essential, but it provides some simplification.

A set of m decreasing exponential functions is selected by choosing as many different positive time constants, enumerated in

increasing order:

$$\{T_j \; ; \; j = 1,\ldots,m\} \qquad (2)$$

$$\varphi_j(t) = e^{-\frac{t}{T_j}} \qquad (j = 1,\ldots,m)$$

We look for a linear combination of these functions ("multi-exponential fit"):

$$\Phi(t) = \sum_{j=1}^{m} C_j \varphi_j(t) \qquad (3)$$

that in some sense to be precisely defined must fit the signal *s(t)*.

Baseline shift correction

The measured values s_i are assumed to approximate the values of a positive, strictly decreasing function $f(t)$ that tends to 0 as $t \to +\infty$. Errors affecting these values are of two kinds: systematic errors ("baseline shift") due to the measuring device and random errors.

In the experience gathered through the processing of many cases we have found that, while the systematic errors cannot be adequately represented by a constant term, a first-degree polynomial $At + B$ performs sufficiently well. Good estimates $\tilde{A}$ and $\tilde{B}$ are needed for the coefficient A ("drift") and the constant term B ("initial shift"). As for the random errors ε_i, in the absence of additional information, are assumed to come from a normal distribution with zero mean and unknown variance. Therefore, summing up, for every $i = 1,\ldots,n$:

$$s_i = f(t_i) + At_i + B + \varepsilon_i \qquad (4)$$

The baseline shift should be subtracted from the measured values before these values are used for any computation. In practice, the original s_i will be replaced by:

$$g_i = s_i - (\tilde{A}t_i + \tilde{B}) \qquad (5)$$

It is important to recall that all the available information comes from a set of measured values, apart from the theoretical hypotheses stated above.

We now describe the selected procedure to obtain reliable estimates for A and B whenever it is possible. The same algorithm leads to an estimation of the variance of the random observational errors.

Considering the ideal case where random observational errors are absent, and the coefficient A is exactly known so that $\tilde{A} = A$. The constant term B is not important here. Thus, since $f(t)$ is a strictly decreasing function, so is the g_i sequence.

In practice, and as a consequence of the observational errors, the g_i sequence ceases to be monotonically decreasing, starting with a certain value of i. From that point on, it oscillates erratically.

The preceding considerations can be taken as a loose definition of what may be termed the "tail" of the table of data values. For values of t in this tail, $f(t)$ is so close to zero that it becomes negligible in comparison to the random errors ε_i, and the s_i values are then nothing more than random approximations of $At_i + B$. The ensuing computation of the estimates for the baseline and for the variance of the random observational errors relies upon

the existence of a sufficiently long tail.

The baseline is then defined from the least-squares solutions, for each k, of the over-determined linear systems:

$$C_k\varphi_k(t_i) + At_i + B = s_i \quad (i = 1, \ldots, n) \quad (6)$$

with three unknowns $C_k, A,$ and $B.$ The value of k (between 1 and *m*) is finally selected to yield the least sum of squares of differences of both sides when considering all the m systems.

These systems are solved (in the sense of least squares) by the same algorithm used for the solution of the main problem, to be described in the following sections, with a special treatment for unknowns A and B. This is because the $C_j's$ will be subject to non-negativity conditions, while A and B are not restricted with respect to their signs.

Set of coefficients: optimization with regularization

In principle, the least squares criterion is adopted: the set $\{C_j;\ j = 1, \ldots, m\}$ is to be determined such that the quadratic functional:

$$S_0 = \sum_{i=1}^{n} [p(t_i) - g_i]^2 = \sum_{i=1}^{n} \left[\sum_{j=1}^{m} C_j\varphi_j(t_i) - g_i \right]^2 \quad (7)$$

takes its minimum value.

Problems of this kind are ill–defined,[3, 6, 8, 12] i.e. the presence of small errors (noise) in the data may produce drastic changes in the spectral distribution (coefficients). This is due to the fact that the linear independence of a basis consisting of such exponential functions is very weak in terms of the usual precision of floating-point operations. For this reason, the application of some smoothing procedure becomes mandatory.[3, 6, 8, 12] In our case, the following has been chosen: the functional to be minimized will be

$$S = S_0 + \alpha S_1 \quad (8)$$

where α is a positive parameter (degree of smoothing), and S_1 is defined as:

$$S_1 = \frac{q}{h^4}\left[C_1^2 + (C_2 - 2C_1)^2 + \sum_{j=2}^{m-1} (C_{j+1} - 2C_j + C_{j-1})^2 + (-2C_m + C_{m-1})^2 + C_m^2 \right] \quad (9)$$

The normalization parameters q (scaling factor) and h (mean logarithmic spacing) are conveniently chosen in order to obtain a uniform interpretation of the range of values of α when comparing different applications. The scaling factor q is defined as the quotient between the numbers of terms in each sum, that is, $n/(m + 2)$. If the time constants are uniformly spaced in the logarithmic scale (as it happens in our algorithm), h is just this spacing. In this case, S_1 corresponds to sums of squares of second-order divided differences

(roughly, squares of curvatures). with:

The expansion of partial derivatives starts

$$\frac{\partial S_0}{\partial C_k} = 2\sum_{i=1}^{n}\left[\sum_{j=1}^{m} C_j \varphi_j(x_i) - g_i\right]\varphi_k(x_i)$$
$$= 2\sum_{j=1}^{m}\left\{\left[\sum_{i=1}^{n}\varphi_j(x_i)\varphi_k(x_i)\right]C_j - \sum_{i=1}^{n} g_i \varphi_k(x_i)\right\} \quad (10)$$

Here, defining the inner product of two vectors of dimension n,

$$\langle a, b\rangle = \sum_{i=1}^{n} a_i b_i \quad (11)$$

one may write for short

$$\frac{\partial S_0}{\partial C_k} = 2\sum_{j=1}^{m}\left(\langle \varphi_j, \varphi_k\rangle C_j - \langle g, \varphi_k\rangle\right) \quad (12)$$

with an obvious definition of the vectors included in this expression. On the other side, for $3 \le k \le m-2$,

$$\frac{\partial S_1}{\partial C_1} = \frac{2q}{h^2}(6C_1 - 4C_2 + C_3)$$
$$\frac{\partial S_1}{\partial C_2} = \frac{2q}{h^2}(-4C_1 + 6C_2 - 4C_3 + C_4)$$
$$\frac{\partial S_1}{\partial C_k} = \frac{2q}{h^2}(C_{k-2} - 4C_{k-1} + 6C_k - 4C_{k+1} + C_{k+2}) \quad (14)$$
$$\frac{\partial S_1}{\partial C_{m-1}} = \frac{2q}{h^2}(C_{m-3} - 4C_{m-2} + 6C_{m-1} - 4C_m)$$
$$\frac{\partial S_1}{\partial C_m} = \frac{2q}{h^2}(C_{m-2} - 4C_{m-1} + 6C_m)$$

Some additional conditions are still to be imposed. But, to the only purpose of completing a preliminary statement for reasons of a better comprehension, let us for the moment suppose we are dealing with a problem of optimization of the quadratic functional that has been defined, without any constraints. Then the (unique) solution of this unconstrained problem is characterized as the solution of the linear system in the C_j:

$$\frac{\partial S}{\partial C_k} = \frac{\partial S_0}{\partial C_k} + \alpha \frac{\partial S_1}{\partial C_k} = 0 \qquad (k = 1, \ldots, m) \qquad (15)$$

Matrix formulation

To express the problem in terms of matrix algebra, it is convenient to define:

$$\mathbf{V} = \begin{bmatrix} \varphi_1(1) & \cdots & \varphi_m(1) \\ \vdots & \ddots & \vdots \\ \varphi_1(n) & \cdots & \varphi_m(n) \end{bmatrix} \qquad n \times m \text{ - matrix ("basic values")}$$

$$\mathbf{C} = \begin{bmatrix} C_1 \\ \vdots \\ C_m \end{bmatrix} \qquad m \text{ - vector ("coefficients")}$$

$$\mathbf{A} = \mathbf{VC} = \begin{bmatrix} \sum_{j=1}^{m} C_j \varphi_j(1) \\ \vdots \\ \sum_{j=1}^{m} C_j \varphi_j(n) \end{bmatrix} = \begin{bmatrix} \Phi(1) \\ \vdots \\ \Phi(n) \end{bmatrix} \qquad n \text{ - vector ("approximations")}$$

$$\mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix} \qquad n \text{ - vector ("correctedsignal")}$$

$$\mathbf{r} = \mathbf{A} - \mathbf{g} = \mathbf{VC} - \mathbf{g} \qquad n \text{ - vector ("residuals")}$$

A matrix $\mathbf{B}$ ("band") related to second-order divided differences can be defined as

$$\mathbf{B} = \frac{1}{h^2} \begin{array}{c} \begin{matrix} 1 & 2 & 3 & \cdot & \cdot & \cdot & m \end{matrix} \\ \begin{bmatrix} +1 & 0 & \cdot & \cdot & \cdot & 0 \\ -2 & +1 & 0 & & & \cdot \\ +1 & -2 & +1 & 0 & & \cdot \\ 0 & +1 & -2 & +1 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & +1 & -2 & +1 \\ \cdot & & & 0 & +1 & -2 \\ 0 & \cdot & \cdot & 0 & 0 & +1 \end{bmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ \cdot \\ \cdot \\ \cdot \\ m \\ m+1 \\ m+2 \end{array} \qquad (n+2) \times m \text{ matrix}$$

The divided differences are:

$$\mathbf{BC} = \frac{1}{h^2}\begin{bmatrix} C_1 \\ C_2 - 2C_1 \\ C_3 - 2C_2 + C_1 \\ C_4 - 2C_3 + C_2 \\ \vdots \\ C_m - 2C_{m-1} + C_{m-2} \\ -2C_m + C_{m-1} \\ C_m \end{bmatrix} \qquad (n+2) \text{ vector ("curvatures")}$$

Then we have:

$$S_0 = \|\mathbf{r}\|^2 = \mathbf{r}^{\mathbf{T}}\mathbf{r} \qquad \text{scalar}$$

$$S_1 = q\|\mathbf{BC}\|^2 = q\mathbf{C}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{BC} \qquad \text{scalar}$$

$$\nabla_{\mathbf{C}}\,\mathbf{r} = \left[\frac{\partial r_i}{\partial C_k}\right] = \mathbf{V} \qquad n \times m \text{ - matrix}$$

$$\nabla_{\mathbf{C}}\,S_0 = \left[\frac{\partial S_0}{\partial C_k}\right] = 2\mathbf{V}^{\mathbf{T}}\mathbf{r} \qquad m \text{ - vector}$$

$$\nabla_{\mathbf{C}}\,S_1 = 2q\mathbf{B}^{\mathbf{T}}\mathbf{BC} \qquad m \text{ - vector}$$

$$S = \|\mathbf{VC} - \mathbf{g}\|^2 + \alpha q\,\|\mathbf{BC}\|^2 \qquad \text{scalar}$$

$$\nabla_{\mathbf{C}}\,S = 2\,\mathbf{V}^{\mathbf{T}}\mathbf{r} + 2\alpha q\,\mathbf{B}^{\mathbf{T}}\mathbf{BC} = 2\left[\left(\mathbf{V}^{\mathbf{T}}\mathbf{V} + \alpha q\,\mathbf{B}^{\mathbf{T}}\mathbf{B}\right)\mathbf{C} - \mathbf{V}^{\mathbf{T}}\mathbf{g}\right] \qquad m \text{ - vector}$$

The solution of the unconstrained problem is the coefficient vector $\mathbf{C}$ that satisfies the linear system:

$$\left(\mathbf{V}^{\mathbf{T}}\mathbf{V} + \alpha q\,\mathbf{B}^{\mathbf{T}}\mathbf{B}\right)\mathbf{C} = \mathbf{V}^{\mathbf{T}}\mathbf{g} \qquad m \text{ - vector} \qquad (16)$$

This is precisely the system of normal equations for the least-squares solution of an over-determined linear system which we now proceed to write. This is apparently a backward path, but there exist strong computational reasons for doing so. The normal equations do not yield an optimal algorithm because the system is often ill-conditioned. We have devised an algorithm adapted to our problem, based on the ***QR*** decomposition [13] and applied to an adequate over-determined system, not resorting to the normal equations. Moreover, the application of the required non-negativity constraints will then become extremely simple.

Let **U** be the $\mathbf{V}$ matrix augmented by the $m + 2$ rows of the matrix $\sqrt{\alpha q}\ \mathbf{B}$. Thus, **U** has $n + m + 2$ rows and m columns. Similarly, let **g'** be the column vector **g** augmented by $m + 2$ zeros. Then the over-determined linear system for the unconstrained problem is:

U C = g' $(n + m + 2)$ – vector (17)

which in the case $\alpha = 0$ is equivalent to:

V C = g n – vector (18)

Constraints

At this point, it is convenient to move on to the statement of the additional conditions still to be specified. These are, simply, that the coefficients $C_j;\ j = 1,\ldots,m$ of the exponentials must be nonnegative. In this case there is also a unique solution such that the functional S takes its minimum value, and the optimality conditions determining it are the following:

$$C_j > 0 \ \Rightarrow\ \frac{\partial S}{\partial C_j} = 0 \quad \text{(passive constraints)}$$
$$C_j = 0 \ \Rightarrow\ \frac{\partial S}{\partial C_j} \geq 0 \quad \text{(active constraints)} \qquad (19)$$

It is not known beforehand which of the coefficients must be positive and which ones must be zero. The essential part of the work reported here consists of an algorithm that, in a finite number of steps, determines such distribution and consequently the optimal multi-exponential fit.

In order to write down the matrix formulation of the situation just described, we consider the set of integers from 1 to m expressed as the union of two disjoint sets K_0 and K_1.

The integers k corresponding to the passive constraints belong to K_1. If we are dealing with the optimal solution we must have:

$$C_k > 0,\ \frac{\partial S}{\partial C_k} = 0 \Rightarrow k \in \mathrm{K}_1 \qquad (20)$$

In the case of the active constraints, there are two possibilities:

$$C_k = 0,\ \frac{\partial S}{\partial C_k} > 0 \quad \text{(strongly active constraints)} \qquad (21)$$

or

$$C_k = 0,\ \frac{\partial S}{\partial C_k} = 0 \quad \text{(weakly active constraints)} \qquad (22)$$

The integers k corresponding to the strongly active constraints belong to K_0. The remaining integers, corresponding to the weakly active constraints, may belong to either set. This introduces some ambiguity from the theoretical point of view, but it is inherent to the

algorithm, since a set defined by a system of equalities is ill-defined in floating-point computation. In fact, the algorithm efficiently overcomes this ambiguity, as will be later explained.

Let now $\mathbf{D}$ be a diagonal matrix of order m, its diagonal elements being defined as follows: $d_{kk} = 0$ for $k \in \mathrm{K}_0$ and $d_{kk} = 1$ for $k \in \mathrm{K}_1$.

Either K_0 or K_1 may be empty. If K_0 is empty, we have **D = I.** If K_1 is empty, **D = 0.**

Computational Algorithm

Summing up, the solution of the optimization problem for each value given to α is the vector **C** satisfying simultaneously:

- The following over-determined linear system, in the least squares sense

UDC = g' $(n + m + 2)$ - vector (23)

- The following systems of inequalities, for every component:

$$\left.\begin{array}{c} \mathbf{C} \geq 0 \\ U^T U\mathbf{C} \geq \mathbf{V^T g} \end{array}\right\} \qquad m \text{ - vectors} \tag{24}$$

for some particular definition of the diagonal matrix **D**, that is, for one of the possible partitions of the set of integers from 1 to m into two subsets K_0 and K_1, according to the previous specifications. As established by the relevant theory, this partition is unique, except for the possible ambiguity produced by one or more coefficients for which $c_j = \dfrac{\partial S}{\partial C_j} = 0$ holds. This ambiguity can be resolved in any way, not affecting the final results.

The computational algorithm that has been implemented starts with an arbitrary (but reasonable) initial definition of the sets K_0 and K_1. Herewith, the matrix **D** is defined too. The system of equations is solved in the least-squares sense by an algorithm based on the ***QR*** decomposition. If its solution, namely vector **C**, satisfies all the inequalities, then the solution to the optimization problem has been found. Otherwise, one or more indices are transferred from one set to the other, and the computational loop is repeated with the new definition of **D**.

Numerical experience has shown that, without proper precautions, repeated partitions sometimes occur, thereby inducing an infinite repetition of the computational loop. In some instances, this happened as a consequence of rounding errors, in situations close to the earlier mentioned ambiguity cases. For this reason, some safeguards of the following types have been incorporated into the algorithm

1) In each computational loop, only one index is transferred from one set to the other, even when the inequalities fail to be satisfied for more than one index.

2) A count is kept for the transfers of each index, giving priority for further changes to the indexes that have accumulated the fewest transfers.

3) The non-negativity conditions are gradually relaxed, thereby clearing the cases of ambiguity without undercutting the precision of the results, which are always expressed with more than enough significant digits.

Recommended Value of α (degree of

smoothing)

In our software, the computational algorithm just described is carried on for a fixed set of values of α. Except for α = 0, all other values are positive and uniformly spaced in a logarithmic scale. In every case, the coefficients of the exponentials (the C_j) are determined for $1 \leq j \leq m$, in correspondence with the time constants T_j .

Based on solid empirical evidence, we have developed an original criterion in order to determine the recommended degree of smoothing of the α-dependent function $C(T_j)$. Physical intuition calls for clipping sharp peaks and softening slopes, without over-smoothing.

Starting with the raw data, we first define approximate second derivatives $\tilde{f}''(t_i)$ for $1 \leq i \leq n$, from segmental least-squares polynomial approximations of the data values. Notice that the second derivatives are independent of the baseline.

For each value of α in turn, the second derivatives $\Phi''_\alpha(t_i)$ of the respective multi-exponential fit are computed for $1 \leq i \leq n$. Next, the mean-square difference of $\tilde{f}''$ and Φ''_α for all n times is defined as a function of α:

$$z(\alpha) = \sqrt{\frac{1}{n-m}\sum_{i=1}^{n}\left[\tilde{f}''(t_i) - \Phi''_\alpha(t_i)\right]^2} \quad (25)$$

Then, for all values of α in the fixed set except the first and the last, second differences of z with respect to α are computed:

$$\delta^2 z(\alpha_k) = z(\alpha_{k-1}) - 2z(\alpha_k) + z(\alpha_{k+1}) \quad (26)$$

Up to a constant factor, these second differences are approximations of the respective logarithmic second derivatives, but the factor may be disregarded because relative values are relevant.

In practical cases, the function $\delta^2 z(\alpha)$ has a very characteristic behavior. It is quite stable for small values of α, then followed by a steep rise (in absolute value). Recommended values of α are those just smaller than the starting abscissa of the sharp slope.

Comments and Observations

The theory dealing with the minimization of a quadratic functional with non-negativity conditions is a simple case of application of the outstanding theory for convex functional and more general conditions. The optimality conditions that have been quoted are a particular instance of the ones known as Karush-Kuhn-Tucker (KKT) conditions.[10, 11]

Several authors[3, 7] have proposed different smoothing procedures, leaving a wide margin to experiment with other choices. The criterion recommended by Fordham, Sezginer and Hall,[12] in a different context, incorporates the use of the logarithmic derivative $\frac{d\log S_0}{d\log\alpha}$, which can be easily computed as a by-product of our algorithm.

Results

In order to test the method, above developed, a large variety of data were

produced and analyzed. In this section two representative cases are presented and fully discussed. In addition an experimental result of pore size distribution measurement, by means of NMR, is also presented.

The general procedure to produce the data –time decaying signals are considered- to be analyzed is as follows:

1) A varying number of decaying exponentials, characterized by their time constants, are multiplied by different sets of coefficients and the results are added up. The sum of each set of coefficients is, arbitrarily, set equal to 100. In this way the “pure” signals are produced.

2) These pure signals are corrupted by noise. The noises are generated using a resident subroutine in Fortran.[14] They follow a Gaussian distribution centered at 0 and with a standard deviation of 1. These noises are multiplied by factors spanning the range 1 through 5 and added up to the pure signals, thus obtaining signals with varying signal/noise ratios.

3) Finally various linear base-line drifts were added up. These base-lines were added to account for possible real experimental situations where this effect is present and must be accounted for in order to properly analyze the signals, therefore avoiding obtaining wrong results.

Thus, by adding pure signals, noises and base-line drifts the working signals to be analyzed are obtained. These working signals are built trying to simulate those obtained in a (Carr-Purcell-Meiboom-Gill CPMG-T_2 decay of a NMR signal[3,15]). For the ease of the analyses the data are generated at integer time values in the range 1-5000 ms.

Figures 1 to 3 and 4 to 6 depict the various contributions required to obtain the working signals *WSA* and *WSB*, whose relevant parameters are given in the following equations:

$$WSA(\,) = PSA(\,) + N(\,) + BL(\,)$$

$$PSA(\,) = 100 * \exp\left(-\frac{t_i}{50}\right) \qquad t_i = 1,2,3,....,5000\ ms$$

$$N(\,) = Noise\ A = \text{Gaussian noise centered at } 0 \text{ with a standar deviation of } 5$$

$$BL(\,) = 2 * 10^{-3} t_i \qquad (27)$$

$$WSB(t_i) = PSB(t_i) + N(t_i) + BL(t_i)$$

$$PSB(t_i) = 50^* \exp\left(-\frac{t_i}{10}\right) + 50^* \exp\left(-\frac{t_i}{100}\right) \qquad t_i = 1,\ 2,\ 3,\ldots,\ 5000\ ms \tag{28}$$

$$N(t_i) = NoiseB = \text{Gaussian noise centered at 0 with a standard deviation of 2.}$$

$$BL(t_i) = 10 + 2^*10^{-3} t_i$$

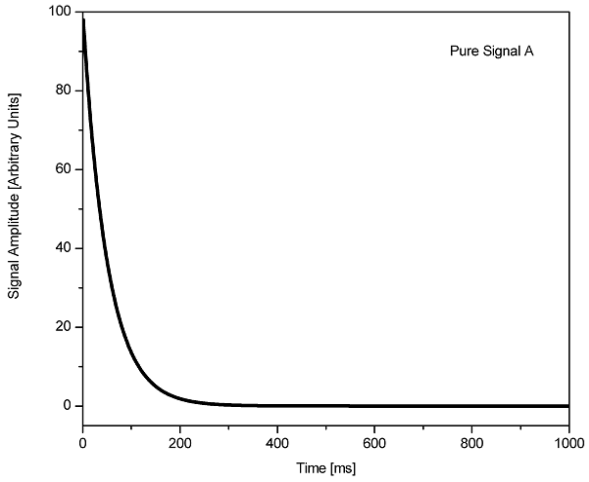


Figure 1. The pure signal A.

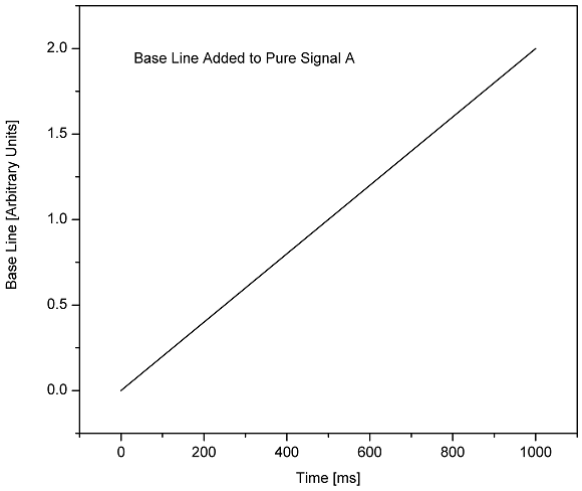


Figure 3 Base-line drift added to pure signal A, equation (26).

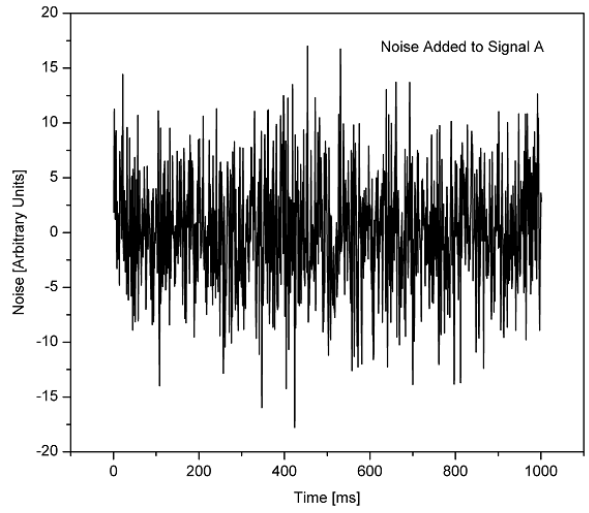


Fig. 2 Gaussian noise added to pure signal A (NoiseA), centered at 0 and with a standard deviation of 5.

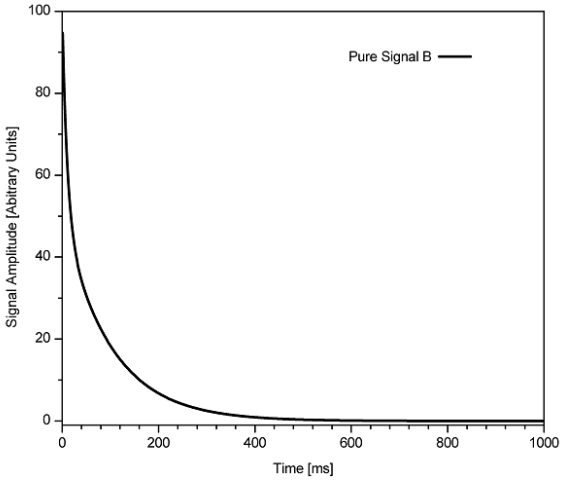


Figure 4. The pure signal B.

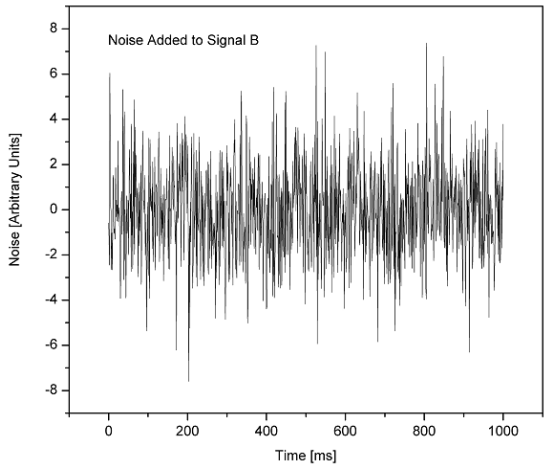


Figure 5. Gaussian noise added to pure signal B (NoiseB), centered at 0 and with a standard deviation of 2.

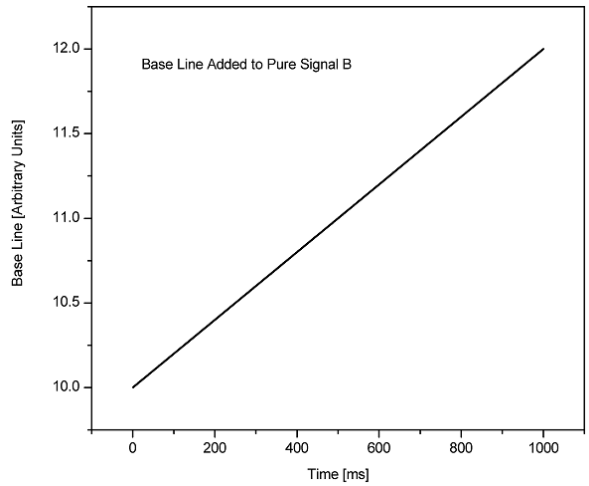


Figure 6. Base-line drift added to pure signal B, equation (27).

WSA and *WSB* are the working signals *A* and *B* respectively, which are shown in figures 7 and 8; *PSA* and *PSB* are the pure signals *A* and *B*; *NoiseA* and *NoiseB* are Gaussian noises centered at 0 and with standard deviations of 5 and 2, respectively.

In order to check the behavior of the method developed above a set of 101 time constants is chosen spanning the rage 1 – 10000 *ms* equally spaced on a logarithmic scale.

Once the set of time constants is chosen what is left is to choose the regularization parameter (RP) and to calculate the base-line and the coefficients. The proper determination of the base-line is important particularly for the correct determination of long time constants coefficients. Also, it is important to determine a correct value for the RP since the time constants distribution depends on the chosen value. The “optimum” value for the RP depends on the expertise of the operator, who is strongly guided by the numerical help previously given in the text. Let us discuss the results obtained for both cases.

Case A

As shown in equation (27) this is the case of a dominant exponential with significant noise.

Some points are worth mentioning:

. As depicted in figure 9, obtained with no regularization, the coefficients show violent changes in the vicinity of 50 ms, this is the kind of unwanted situations to expect the regularization will solve.

. A spike shows up at the shortest time constant used, i.e. 1 ms. This spike arises from the region where the data points have a spacing similar or comparable to that of one of the time constants employed, and the noise which in this time region has the maximum impact.

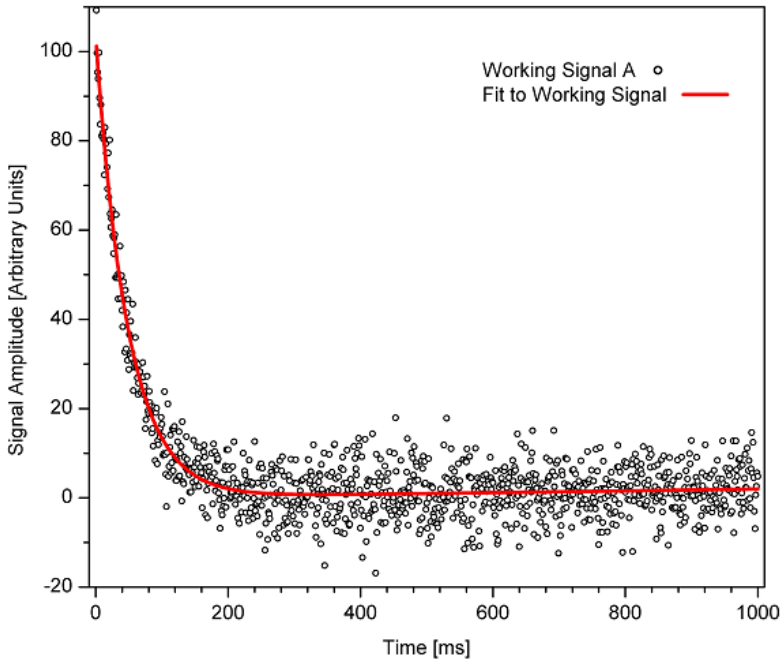
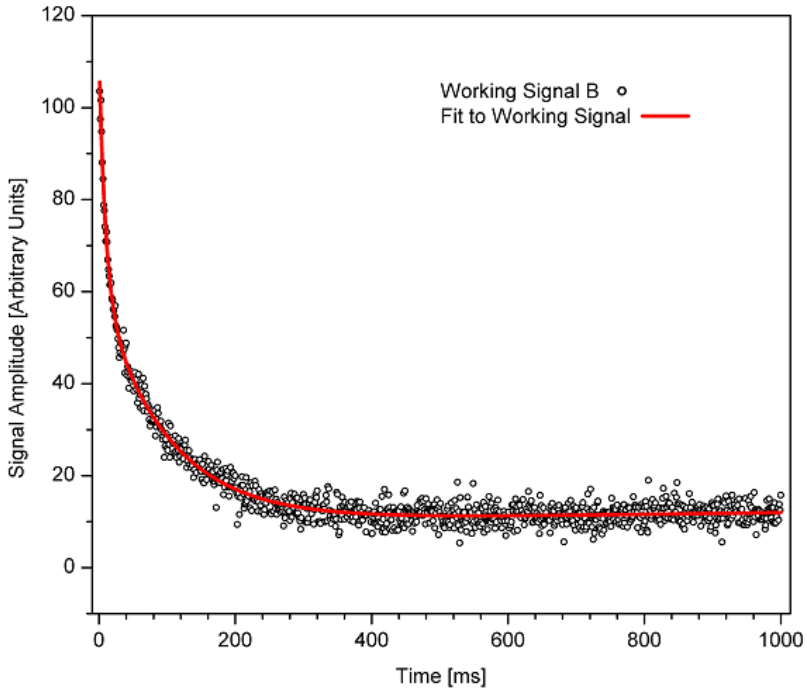


Figure 7. Working signal A obtained by adding up the above three shown contributions. The red line is the fit obtained to the data using the values given in figure 6. The base-line parameters determined are: *Slope* = 0.0019939 Arbitrary Units/ms and the *Intercept* = 0.0046282 Arbitrary Units.



Figur 8. Working signal B obtained by adding up the above three shown contributions. The red line is the fit obtained to the data using the values given in figure 12. The base-line parameters determined are: *Slope* = 0.0018762 Arbitrary Units/ms and the *Intercept* = 10.4192705 Arbitrary Units.

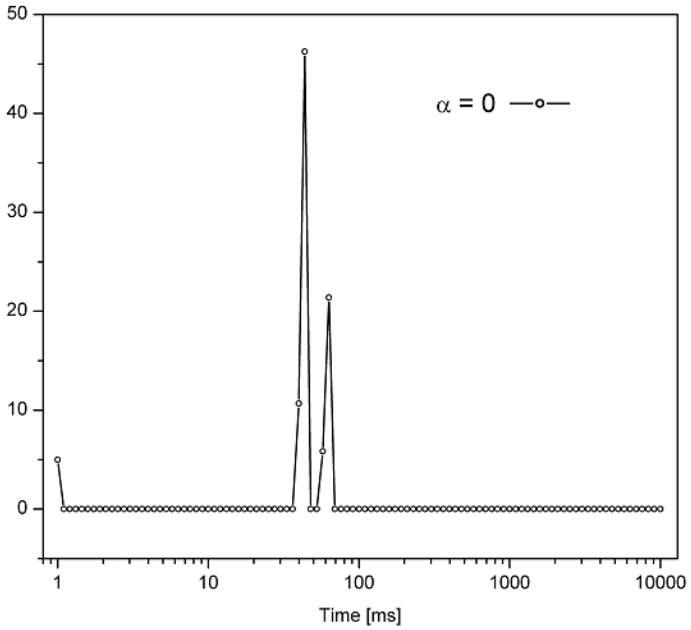


Figure 9. The values obtained for the coefficients and their corresponding time constants are depicted. The typical violent oscillations of the amplitudes, when no regularization is applied, corresponding to neighboring time constants are clearly seen. Also, spurious coefficients are generated at the shortest time constants, clearly lacking of any physical meaning. Both situations are satisfactorily solved when regularization is applied as may be seen in the results shown in figure 6. The addition of all the amplitudes is 100.

The regularization adequately copes with these two situations as may be seen in figure 10, where it is depicted the coefficient distribution which shows in, in the first place, that a smooth distribution is obtained in the neighborhood of 50 ms, and, in the second place, the spike, with a time constant of 1 ms, shown in figure 9 becomes unnoticeable.

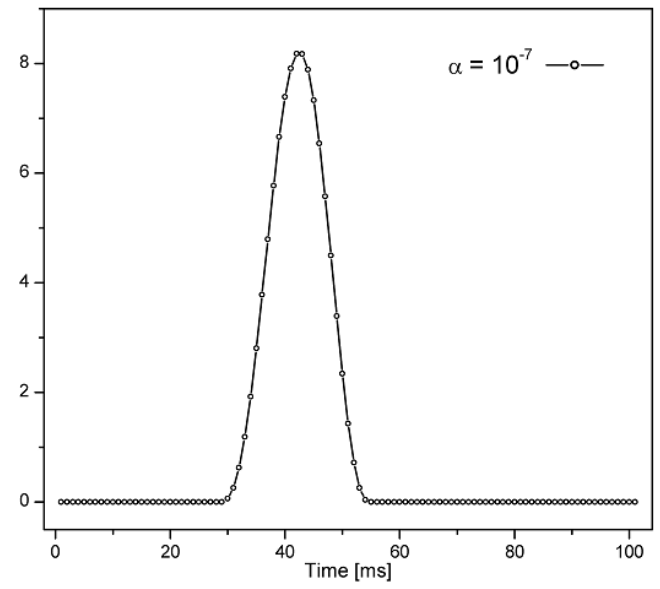


Figure 10. The time constants and their corresponding amplitudes, when regularization is applied, are shown. Comparing with figure 5 it may be seen that the violent oscillations and that the spurious signal at the shortest time constants have disappeared. The addition of all the amplitudes is 100.

The base-line determined shows an excellent agreement as may be seen by comparing the values given in equation (26) with those shown in the caption to figure 9.

Thus, by proper determination of the base-line and of the regularization parameter a bell-shaped coefficient distribution is determined, peaked at about 50 ms and which is able to closely reproduce the working signal as may be seen in figure 7.

Case B

The construction of this working signal is given in equation (27). Figure 11 depicts the coefficient distribution clearly showing the presence of two well defined time constants, 10 and 100 ms, with equal weights (amplitudes) of about 50 arbitrary units. Figure 12 depicts the much smoother distribution, looking like a double bell centered at about 10 and 100 ms. The addition of the coefficients of each bell adds up to about 50, as expected.

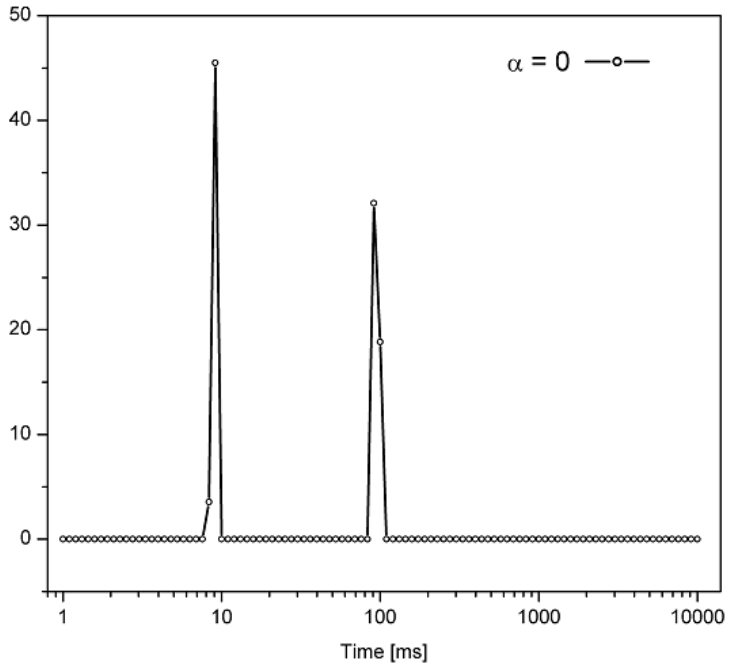


Figure 11. The values obtained for the coefficients and their corresponding time constants, when no regularization is applied, are depicted. The addition of all the amplitudes is 100.

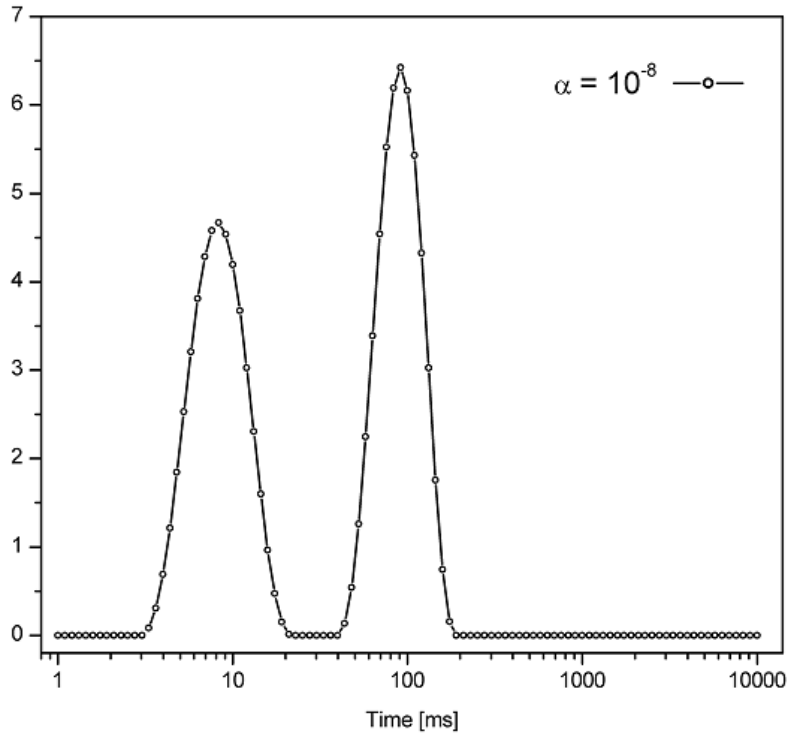


Figure 12. The time constants and their corresponding amplitudes, when regularization is applied, are shown. The addition of all the amplitudes is 100.

Figure 13 depicts a numerical generated function ($\delta^2 z(\alpha)$), equation (25), depending upon the α (the RP) which runs over 10 orders of magnitude from 10^{-10} to 1. Sudden changes are clearly seen as a function of α. First, a relatively smooth behavior is exhibited, followed by a sudden change that is reached while increasing α. The "best" value for α is to be taken towards the left of the sudden change and where the first premonitory effects are detected. This type of behavior is similar in all of the cases we have dealt with. There is some imprecision in the sense that the transition from smooth region to that of the sudden change is somewhat subjective and not precisely defined. However, on the one hand, a reasonably trained operator will determine the value of α within an order of magnitude, and on the other hand this order of magnitude variation does not significantly affect the coefficient distribution function. Anyway, this method to determine the RP α greatly improves on previous methods.[4,12, 16, 18]

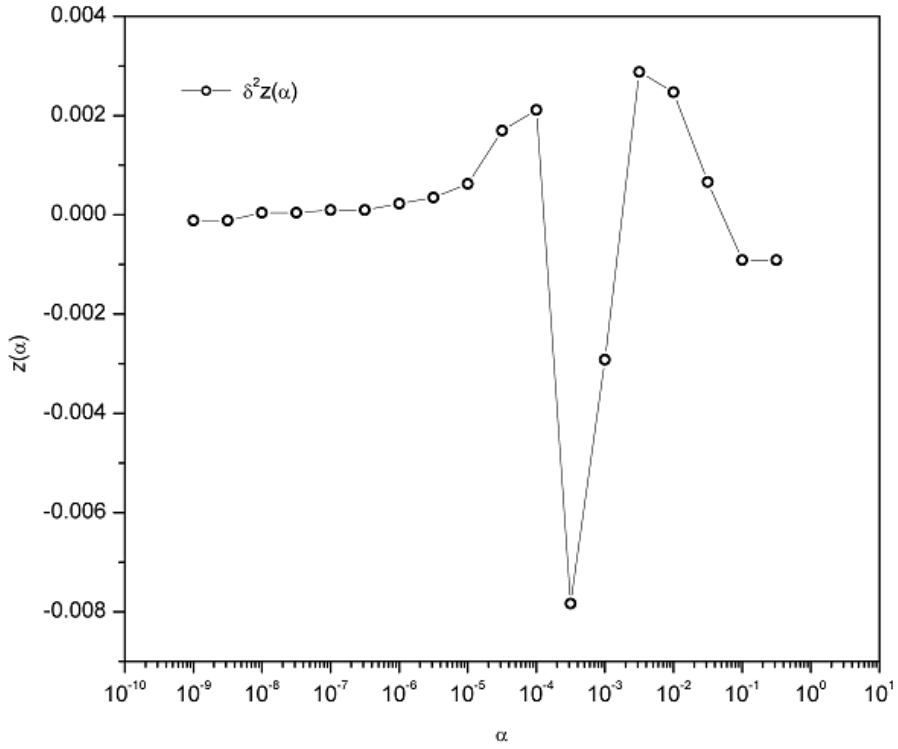


Figure 13. $\delta^2 z(\alpha)$, equation (25), versus the regularization parameter α.

An experimental result of pore size distribution measurement, by means of NMR-T_2,[3, 5] of a natural brine saturated rock and after centrifugation is presented in figures 14 and 15. The difference between the two fits shows the movable fluids and the pore size distribution affected.

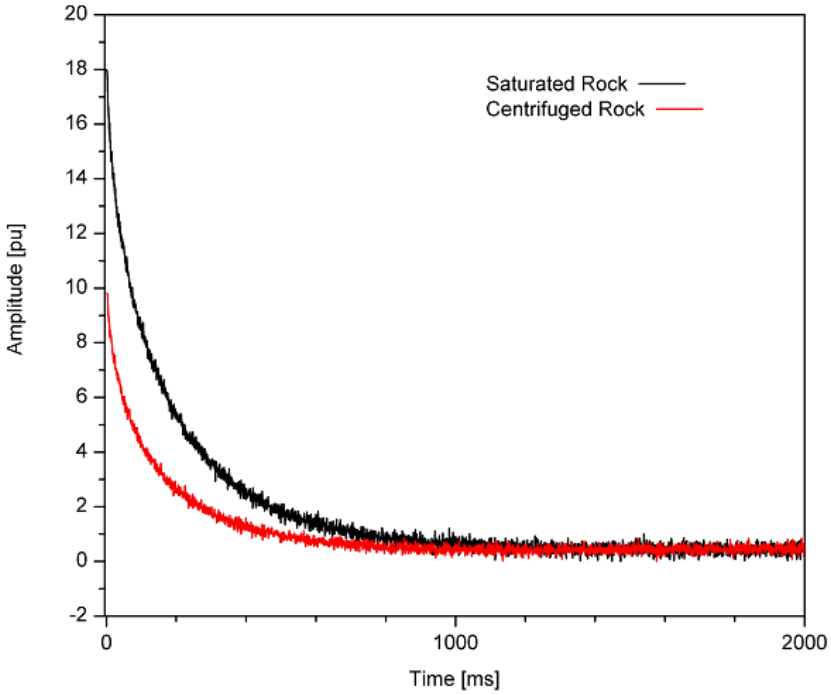


Figure 14. ^{1}H NMR-T_2 decay signal, of water in saturated sandstone rock, both in the saturated and centrifuged conditions.

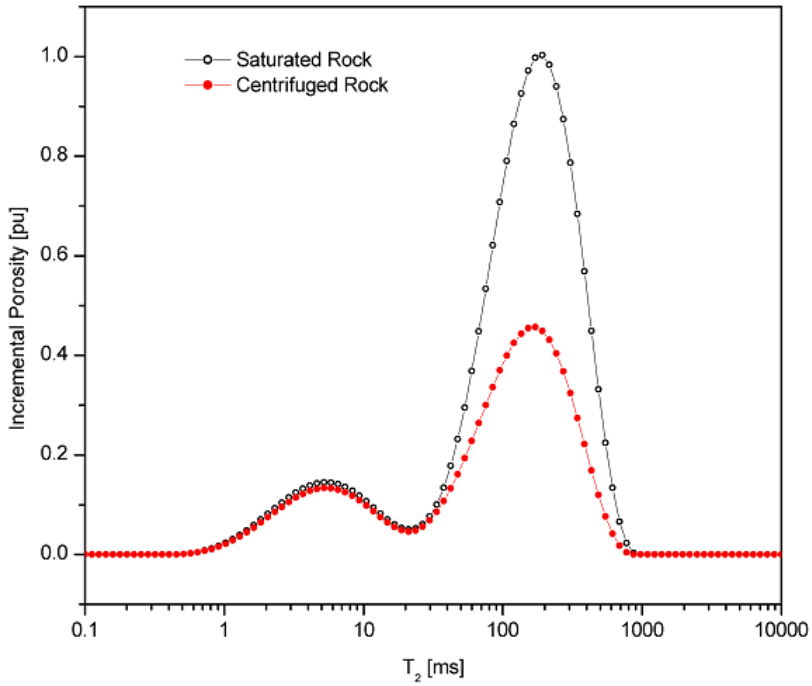


Figure 15. Pore size distribution corresponding to the saturated sandstone, measured in figure 14, for both the saturated and centrifuged conditions.

Conclusions

The final conclusions of the method developed in this work are:

Includes a proper treatment of base-line drifts and shifts;

The QR decomposition employed to find the solution of the least-squares linear system greatly reduces the characteristic rounding up errors presented in solving these kinds of ill-defined problems; and

Provides a clear procedure to precisely determine the regularization parameter □. Therefore, the distribution coefficient function determined is a reliable one, which is not the case in previous works.

Acknowledgements

Support provided by Universidad Nacional de Córdoba, by the Consejo Nacional de Investigaciones Científicas y Técnicas (through grant PIP 02588) and by the firm MR Technologies S.A., are gratefully acknowledged.

References

1. ISO 8292-1, Animal and vegetable fats and oils - Determination of solid fat content by pulsed nuclear magnetic resonance method - Part 1: Direct method.

2. ISO 8292-2, Animal and vegetable fats and oils - Determination of solid fat content by pulsed nuclear magnetic resonance method - Part 2: Indirect method.

3. Coates G.R., Xiao L. and Prammer, M.G.: NMR Logging: Principles and Applications, Halliburton Energy Services Publication. (1999).

4. Davies S. and Packer K.J.: J. Appl. Phys. **67** (1999) 3163.

5. Straley C., Rossini D., Vinegar H., Tutunjian P. and Morris C.: The Log Analyst **84** (1997).

6. Kleinberg R.L., Kenyon W.E. and Mitra P.P.: J. Magn. Resonance A **108** (1994) 206.

7. Pino G.A., Rinaldi C.A. and Ferrero J.C.: J. Phys. Chem. A **107** (2003) 6761.

8. Fletcher R.: Practical Methods of Optimization. John Wiley & Sons, Chichester (1987).

9. Lawson C.L. and Hanson R.J.: Solving Least Squares Problems, Englewood Cliffs, NJ. Prentice-Hall (1974).

10. Fletcher R.: Practical Methods of Optimization, 2nd Edition, John Wiley & Sons, Chichester (2000).

11. Bonnans J. F., Gilbert J. C., Lemaréchal C., et Sagastizábal C. : Optimisation Numérique, aspects théoriques et pratiques. Springer-Verlag, Berlin-Heidelberg (1997).

12. Fordham E. J., Sezginer A. , and Hall L.D.: J. Magn. Resonance, Series A **113** (1995) 139.

13. Watkins D. S.: Fundamentals of Matrix Computations, John Wiley & Sons (1991).

14. Fortran Microsoft Developer Studio, Microsoft corporation (1995).

15. Slichter C.P.: Principles of Nuclear Magnetic Resonance, Springer Verlag (1990).

16. Davies S., Kalam M.Z. and Zelaya O.F.: J. Appl. Phys. **67** (1990) 3171.

17. Kleinberg R.L. and Horsfield M.A.: J. Mag. Res. **88** (1990) 9.

18. Whittall K.P. and Mackey A.L.: J. Mag. Res. **84** (1989) 134.

Appendix

Least-squares solution of linear systems using *QR* decompositions

The title evokes an extensive subject to which we want to refer only briefly, with application to our particular problem. Let ***Ax = b*** be the system for which we want a least-squares solution. Here (as in **VC = g**), matrix ***A*** is $n \times m$, ***x*** and ***b*** are vectors of dimensions m and n respectively. Normally, $n > m$.

It is always possible, in different ways, to find a decomposition ***A = QR*** where ***Q*** is an orthogonal matrix of order n while ***R*** is $n \times m$ and upper triangular; that is, the element in row i and column j is 0 whenever $i > j$. In particular, all rows for which $i > m$ are composed by zeros.

By definition, ***Q*** is orthogonal if $\boldsymbol{Q}^{-1} = \boldsymbol{Q}^{T}$. Then the system ***Ax = b*** is equivalent to $\boldsymbol{Rx} = \boldsymbol{Q}^{T}\boldsymbol{b}$. This equivalence is not only algebraic but also metric, because an orthogonal transformation preserves distances and angles. The Euclidean norm is the same for both residual vectors

$$\|\boldsymbol{Ax} - \boldsymbol{b}\| = \|\boldsymbol{Rx} - \boldsymbol{Q}^{T}\boldsymbol{b}\|$$

but the second system can be easily solved by back substitution.

The transformation ***Q*** is obtained as a product of orthogonal transformations: $\boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2 \ldots \boldsymbol{Q}_m$. In our algorithm we have chosen to use “reflectors” [13].

If the matrix ***R*** has full rank m, the least-squares solution is unique. If ***R*** is rank-defective (rank < m), as shown by the value 0 of one or more of its diagonal elements, the respective unknowns can be given any value. For our application it makes sense to give them the value 0.

Both sides of ***Ax = b*** are successively pre-multiplied by $\boldsymbol{Q}_1^{T}, \boldsymbol{Q}_2^{T} \ldots, \boldsymbol{Q}_m^{T}$. After the k–th pre-multiplication, the transformed matrix in the left side is upper triangular in its first k columns. Then $\boldsymbol{Q}_{k+1}^{T}$ is defined as a function of the next column, which was previously selected as the column vector of maximum norm among those $m - k$ remaining. This implies a reordering of columns and unknowns, conveniently expressed by a permutation matrix ***P***. Recall that any permutation matrix is orthogonal.

Thus, the decomposition of the original matrix ***A*** is in fact

$$\boldsymbol{AP} = \boldsymbol{QR}$$

and the algorithm proceeds as follows:

$$\boldsymbol{Ax} = (\boldsymbol{AP})(\boldsymbol{P}^{T}\boldsymbol{x}) = \boldsymbol{QR}(\boldsymbol{P}^{T}\boldsymbol{x}) = \boldsymbol{b}$$

$$\boldsymbol{R}(\boldsymbol{P}^{T}\boldsymbol{x}) = \boldsymbol{Q}^{T}\boldsymbol{b}$$

The partial derivatives of the sum of squares of the residuals are expressed as components of the gradient:

$$\nabla \boldsymbol{S} = 2\,\boldsymbol{P}\boldsymbol{R}^{T}(\boldsymbol{R}\boldsymbol{P}^{T}\boldsymbol{x} - \boldsymbol{Q}^{T}\boldsymbol{b})$$