Dielectric Spectra Analysis: Reliable Parameter Estimation Using Interval Analysis

Adrien Brochier, Maëlenn Aufray, and Wulff Possart

1 Introduction

Dielectric spectroscopy (DES) is widely applied to polymers, monomers and other insulating materials because it is an extremely effective method for characterizing the molecular dynamics over many orders of magnitude of time or frequency, respectively. In the measurement, the complex dielectric function

$$\varepsilon^*(\omega) = \varepsilon'(\omega) - i\varepsilon''(\omega) \tag{1}$$

with the angular frequency ω, is measured at constant temperature. This function is called the dielectric spectrum. Commonly, dielectric spectra are modelled by a sum of relaxation processes, but the choice of a reasonable physical model for the relaxator is critical. Most of the usual models, reviewed briefly in the introductory section, result from phenomenological considerations providing limited physical foundation. Moreover, the fitting algorithm turns out to be crucial in terms of reliability and unambiguity of the dielectric model function determined. As discussed in Sect. 2, common softwares use least square approximation fitting algorithms which need initial values for the fit parameters. This could imply some predestination of the fit results. In this work, a parameter estimation algorithm which is free of these limitations will be developed (See Sect. 3). The new algorithm S.A.D.E. not only provides the chosen dielectric model function by a confidence interval for each model parameter like the frequency position and the intensity of all relaxations: it also indicates the number of relaxations that are necessary to model the measured spectrum.

M. Aufray (✉)
Chair Adhesion and Interphases in Polymers, University of Saarland,
66041 Saarbrücken, Germany
e-mail: maelenn.aufray@ensiacet.fr

A. Öchsner et al. (eds.), *Materials with Complex Behaviour*, Advanced Structured Materials 6, DOI 10.1007/978-3-642-12667-3_7,

1.1 The Dielectric Spectroscopy and Its Models

1.1.1 Relaxations

The Debye relaxators [1] describe the dielectric relaxation response of an ideal, non-interacting population of freely rotating dipoles to an alternating external electric field

$$\varepsilon^* = \varepsilon_\infty + \frac{\varepsilon_S - \varepsilon_\infty}{1 + i\omega\tau_0} = \varepsilon_\infty + \frac{\Delta\varepsilon}{1 + i\omega\tau_0} \tag{2}$$

Where ε_S is the static permittivity ($\varepsilon_S = \lim_{\omega\to 0} \varepsilon'(\omega)$), ε_∞ is the optical dielectric constant ($\varepsilon_\infty = \lim_{\omega\to\infty} \varepsilon'(\omega)$) and τ_0 is the characteristic relaxation time of the medium. Let us note that the Debye model refers to a well-defined physical situation. All other relaxator models reported in the literature imply phenomenological modifications of the Debye relaxator without well–defined physical background. For example, the Havriliak–Negami (HN [2]) equation

$$\varepsilon^* = \varepsilon_\infty + \frac{\varepsilon_S - \varepsilon_\infty}{(1 + (i\omega\tau_0)^\alpha)^\beta} \tag{3}$$

is a mixture of the Cole–Cole [3,4] and the Cole–Davidson [5] equations, accounting for the asymmetry and broadness of the measured dielectric dispersion curve by the additional phenomenological parameters α and β. Developed to describe the dielectric relaxation of some polymers, the HN function is now one of the most popular models for dielectric relaxation although no exact physical meaning can still be given to the coefficients α and β.

Then, for dielectric spectra containing several relaxations, it is possible to sum a number of relaxation processes, according to Eq. (2) or (3), i.e. irrespective of the model used. Here is the example, for n relaxations represented by the Debye model,

$$\varepsilon^* = \varepsilon_\infty + \sum_{j=0}^{n} \frac{\Delta\varepsilon_j}{1 + i\omega\tau_j} \tag{4}$$

and the Havriliak–Negami model.

$$\varepsilon^* = \varepsilon_\infty + \sum_{j=0}^{n} \frac{\Delta\varepsilon_j}{(1 + (i\omega\tau_0)^{\alpha_j})^{\beta_j}} \tag{5}$$

1.1.2 DC-Conductivity

At high temperatures and $\omega\to 0$, a contribution of a DC-conductivity (σ_{DC}) can be observed in the dielectric spectra of real polymer samples. It contributes only to the imaginary part of the measured complex dielectric permittivity, as long as the

imaginary part of the generalized complex conductivity can be neglected in the low frequency region (i.e. $\sigma'' \ll \varepsilon'$). With this presumption, the following equation links the measured quantities $\tilde{\varepsilon}'$, $\tilde{\varepsilon}''$ to the true dielectric material quantities ε', ε'' and the DC electric conductivity (σ_{DC}):

$$\tilde{\varepsilon}'(\omega) = \varepsilon'(\omega) \text{ and } \tilde{\varepsilon}''(\omega) = \varepsilon''(\omega) + \frac{\sigma_{DC}}{\omega\varepsilon_0} \tag{6}$$

At high frequencies, the contribution from the DC-conductivity becomes negligible and hence

$$\lim_{\omega\to\infty} \tilde{\varepsilon}''(\omega) = \varepsilon'' \tag{7}$$

At low frequencies, the ε'' term is negligible compared to σ_{DC}, leading to the equation

$$\lim_{\omega\to 0} \log_{10}(\tilde{\varepsilon}''(\omega)) = \log_{10}\left(\frac{\sigma_{DC}}{\varepsilon_0}\right) - \log_{10}(\omega) \tag{8}$$

And hence, as a first approximation, the equation

$$\tilde{\varepsilon}''(\omega) \cdot \omega \cdot \varepsilon_0 = \sigma_{DC} \tag{9}$$

gives a good fit of the DC-conductivity [6–9]. However, some relaxations can take place even at these low frequencies. They are very difficult to fit using classical approaches (i.e. common algorithms) because they can be masked by the DC-conductivity.

As a consequence, the fitting process has to lead to a set of parameters which makes the model to fit both the real and the imaginary part, and has to detect hidden relaxations.

1.1.3 Polarization at Electrodes and Phase Boundaries

Electrode polarization is a parasitic effect in dielectric experiments which can mask the pure dielectric response of the sample material [10]. Moreover, as described by Maxwell, Wagner and Sillars (MWS polarization), phases in heterogeneous media are to be treated as macroscopic volume elements with different ε^* and conductivities σ^* [11–13]. The most basic geometrical situation was considered by Maxwell [11]. This consists of a plate capacitor filled with n dielectric sheets of non-complex dielectric properties and DC conductivities ε_1, σ_1, ε_2, σ_2, … ε_n, σ_n. This resulted in differential equations relating the field across the dielectric as a function of the current though the strata. Maxwell showed his model to be capable of explaining the observed data for dielectric relaxation in such systems. By considering small spheres with material properties ε_2, σ_2 dispersed so as to preclude electrostatic interaction with one another through a medium with properties ε_1, σ_1 Wagner [12] was able to develop Maxwell's analysis further. This analysis was developed by Sillars [13]

for a disperse 2-phase system by assuming that the matrix material behaved as a perfect dielectric (i.e. $\sigma_1 = 0$). The inclusions are spheroids with axis a in the field direction, and with b and c equal to one another. The geometry is wholly described by two variables, q, the volume fraction of dielectric 2, and the axial ratio a/b. The conducting inclusions behave as point dipoles in the dielectric matrix, and a full analysis yields relations similar to the Debye equations:

$$\varepsilon' = \varepsilon_\infty + \frac{\varepsilon_1}{1 + \omega^2\tau^2} \tag{10}$$

$$\varepsilon'' = \frac{\varepsilon_1\omega\tau}{1 + \omega^2\tau^2} \tag{11}$$

Where λ is a particular function of a/b, and ε_∞, N and τ are given by:

$$\tau = \frac{\varepsilon_1(\lambda - 1) + \varepsilon_2}{\sigma_2} \cdot \varepsilon_0 \tag{12}$$

$$N = \frac{\lambda^2}{\varepsilon_1(\lambda - 1) + \varepsilon_2} \tag{13}$$

$$\varepsilon_\infty = \varepsilon_1 \left(1 + \frac{q \cdot \lambda(\varepsilon_2 - \varepsilon_1)}{\varepsilon_1(\lambda - 1) + \varepsilon_2}\right) \tag{14}$$

Sillars includes the dimensionless quantity, q, which is a function of the ratio a/b. The non-linear variation of this quantity implies that little can be deduced about the dielectric properties of a heterogeneous material unless the shapes of the inclusions are known. Now, applying an electric voltage to the heterogeneous material, the mobile charges in phase 2 can be blocked and piled up at the phase boundaries. This picture also applies to the electrodes on the dielectric sample in the similar case of polarization at a blocking electrode. The experimental example presented in the last part of this work reveals that the MWS polarization causes a strong rise both in the real part (where electrode polarization is more visible) and in the imaginary part of the permittivity (where the electrode polarization superimposes to the DC-conductivity) with decreasing frequency. In this work, the electrode polarization will be modeled by one strong Debye relaxator (recall that the MWS equation is very similar to the Debye equation). The only way to separate all the phenomena (electrode polarization, DC-conductivity, and maybe low-frequency dielectric material relaxations) is a simultaneous fit of the real and imaginary part of the permittivity using a formula taking all these phenomena into account.

1.2 Modeling Problems: Simultaneous Fit and Choice of the Model

The main problem is that different physical processes can occur at the same frequency and temperature: the example of some small relaxations hidden by the DC-conductivity is very clear. Therefore, the only way to separate the phenomena

is a simultaneous fit of the real and imaginary part of the permittivity. So the simple fit of the imaginary part of the permittivity (used by most of researchers, except Axelrod et al. [8]) does not give a good solution as relaxations or other phenomena can be missed. The fit of the complex function of the permittivity will be done by our algorithm S.A.D.E. presented in the next paragraphs: the relaxations and the electrode polarization are fitted by the Debye model and the DC-conductivity by its specific function. Of course, we cannot determine *ex ante* the number of Debye relaxations needed to fit our curves, but the program S.A.D.E. will try to fit the curve using from *0* to *n* relaxations and one term due to DC-conductivity.

$$\varepsilon^* = \varepsilon_\infty + \sum_{j=0}^{n} \frac{\Delta\varepsilon_j}{1 + i\omega\tau_j} - i\frac{\sigma_{DC}}{\omega\varepsilon_0} \tag{15}$$

The goal of the data fitting is to find the parameter values of the applied physical model that match the data most closely. The models to which the data are fitted depend on adjustable parameters. Therefore, the fitting process requires both the choice of a physical model and the choice of a suitable computing algorithm. Most of the scientific softwares fit experimental data by using some variants of the least squares approximation method but the success of the fit is not guaranteed. The next section will summarize the least squares approximation method and its disadvantages. A different algorithm will be presented which makes the fit by the Debye model of experimental data with several relaxations possible in an efficient way, even if some of them are hidden.

2 Data Fit Methods: From Least Square Approximation to the Interval Analysis

2.1 Introduction

Let $(x_i, y_i)_{1\leq i\leq n}$ be a set of experimental data, and $f(x,\ p)$ be a physical model depending on a vector of parameters $p = (p_1, \ldots, p_k)$. If the measurement accuracy is known, it is natural to assume that a given vector of parameters p^* leads to a "good" fit if it makes the model consistent with the measurement error e_i on each data point, that is:

$$\forall 1 \leq i \leq n, y_i - e_i \leq f(x_i, p^*) \leq y_i + e_i$$

A graphical visualization of this criterion is shown in Fig. 1.

Interval analysis makes possible the computation with intervals rather than with real numbers, and thus it leads to a very similar criterion for checking whether a "vector of intervals" contains good parameters. Thus, by cutting the multidimensional space in which good parameters are searched into finitely many small pieces

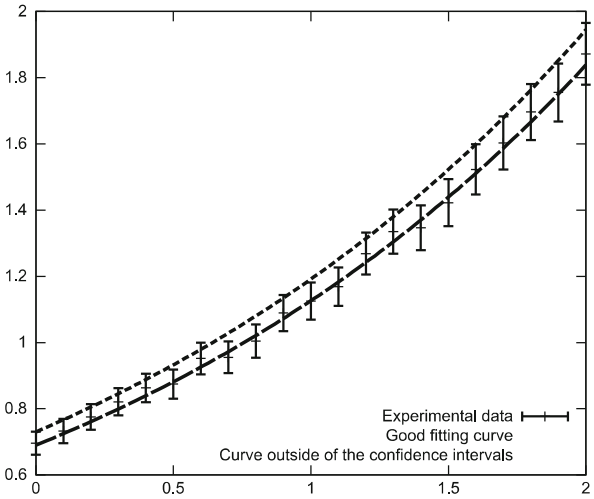

Fig. 1 Fit and confidence intervals

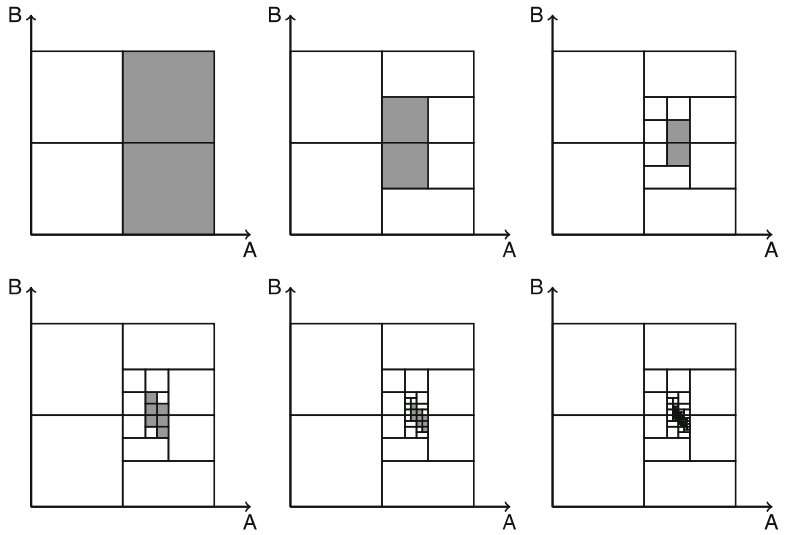

Fig. 2 Modelization of artificial data using a model depending on two parameters *A* and *B*. At every step of the algorithm, the set of feasible parameters is guaranteed to be contained is the area covered by gray squares. White squares doesn't contain any feasible parameters. Thus, at each step the approximation of the set of feasible parameters becomes more accurate

in a suitable way, it is possible to compute the best approximation of the set of good parameters. As shown in Fig. 2, this algorithm is *global*, as at each step the area covered by gray squares is guaranteed to contain the true set of good parameters.

2.2 The Least Square Approximation

The main goal of data fitting is to find the parameters that make the model best describing the data. The commonly used method, namely the Least Square Approximation, tries to minimize the distance between the measured and the calculated points, i.e. to find:

$$\min_{p\in\mathbb{R}^k}\left(\sum_{i=0}^{n}(f(x_i,p)-y_i)^2\right) \tag{16}$$

If the model is linear this can be done analytically but if not, some classical minimization methods (generally based on gradient descent algorithms) are used. Such an algorithm takes a set of some initial parameter values as input and generates a sequence of model parameter vectors which are supposed to converge to one particular parameter vector producing the minimum according to Eq. (16). This method has some disadvantages however:

- The choice of initial values: The key for success of this algorithm is the proper choice of initial values but there is no general method to do that. In some cases, each single fit needs a patient observation of the data and a large amount of unsuccessful testings before finding good initial values. Moreover, when using HN or Debye models, the number of relaxations has to be known in advance. If the number of relaxations is higher than 1 and the relaxations are superimposed, it is very often impossible to find correct initial values for each relaxation.
- The convergence speed: As the algorithm works as an iterative process, it "jumps" at each iteration step from one value to another which is closer to a solution. One critical choice is the size of such a jump. Although this choice is partially done by the algorithm using the gradient and often some additional methods, for example the Levenberg–Marquardt algorithm [14,15], the step must be adjusted according to the nature and the order of magnitude of the parameters.
- The local minimum problem: The algorithm starts from a given vector of parameters and tries to "follow the slope" to find a minimum of the function. Thus, the solution provided by the algorithm could correspond to a local minimum which seems to be a good fit, whereas only the global minimum is physically meaningful.
- The complexity of the model: The model has to be well conditioned in order to make the algorithm work well. If the model is unstable (i.e. if a small variation of the parameters leads to a big variation of the corresponding computed value) or if the number of parameters is too large, convergence is not guaranteed. In particular, if there are some symmetries (i.e. if some permutation of the parameters does not change the value of the function) the algorithm could "hesitate" between different correct possibilities and not converge at all. This problem appears in all cases where we describe the measured dielectric function by a sum of individual physical model functions of the same mathematical type. There are two possibilities to solve this problem:

1. assuming a condition on the parameters which breaks the symmetry, for example that

$$\tau_1 < \tau_2 < \ldots < \tau_r \tag{17}$$

   but this is not possible using the least squares approximation, or
2. choose some perfect initial values which clearly distinguish the different relaxations, which, as it was already noticed, is often a complicated problem.

- The measurement accuracy: Although it is possible to attribute a weight factor to each measured data point according to the accuracy, the quality of the fit result is not directly linked to the measurement accuracy and there is no guarantee that the result will be consistent with it. In particular, a result is always provided, even if no parameter set is consistent with the measurement accuracy (for example if the chosen number of relaxations is too small to describe the data correctly). Moreover, the method supposes that the errors are randomly distributed, which is not always true.
- The bound on the parameters: In general, there is no way to impose some specific constraints for the parameters (for example that they must not be negative).

2.3 The Interval Analysis

Interval analysis was first introduced in order to have a true representation of real numbers for numerical computing: for example, assuming that $\pi = 3.1415$ generates some numerical error which propagates or amplifies during the computing process decreasing the quality of the result. By using an interval instead of a float precision number and assuming that $\pi = [3.1415; 3.1416]$ leads to an interval as the result of the calculation and this interval is guaranteed to contain the true result. Therefore, intervals could also be used in order to manipulate a large range of real numbers simultaneously, and thus to make an approximation of a complex set which is easily handled in computing.

Let $\mathbb{R}$ be the set of real numbers. An interval denoted with $[x]$ is a closed connected subset of $\mathbb{R}$. The lower and upper bound of $[x]$ are denoted by x^- and x^+, respectively. Let $\mathbb{IR}$ be the set of all real intervals, then elementary real operations are extended to intervals according to the following formula:

$$[x] \circ [y] = \{x \circ y \mid x \in [x],\ y \in [y]\} \text{ for } \circ \in \{+, -, *, /\} \tag{18}$$

leading to, for example:

$$[1; 3] + [2; 4] = [3; 7] \tag{19}$$

A vector of intervals is called a box, and $\mathbb{IR}^n$ denotes the set of all n–dimensional boxes. Arithmetic operations with boxes are defined componentwise. The size of an

interval $[x] = [x^-; x^+]$ is defined by:

$$\text{Size}([x]) = x^+ - x^- \tag{20}$$

The size of a box is the size of its greatest component. A bisection procedure will also be used, which cuts a box into two parts and returns the two parts. For a given box $[p] = ([p_1], [p_2], \ldots, [p_k])$, the procedure finds the index i of the greatest component, and returns 2 boxes:

$$([p_1], [p_2], \ldots, [p_i^-; p_i^- + (p_i^+ - p_i^-)/2], \ldots, [p_k]) \tag{21}$$

and

$$([p_1], [p_2], \ldots, [p_i^- + (p_i^+ - p_i^-)/2; p_i^+], \ldots, [p_k]) \tag{22}$$

Let us keep in mind that we do not always get an interval when we calculate the image of an interval by a real function. For example,

$$\sqrt{[4; 9]} = [-3; -2] \cup [2; 3] \tag{23}$$

provides not one but two intervals. Thus, for more complicated operations, some approximations have to be applied in order to keep a consistent representation for computing. In this case for example, the only possible choice is to take the "interval square root" of [4, 9] to be $[-3, 3]$.

More generally, let

$$f : \mathbb{R} \longrightarrow \mathbb{R} \tag{24}$$

be a real function. A so-called "inclusion function" for f is an interval function

$$[f] : \mathbb{IR} \longrightarrow \mathbb{IR} \tag{25}$$

which verifies

$$\forall [x] \in \mathbb{IR},\ f([x]) \subset [f]([x]) \tag{26}$$

In other word, the image of an interval (or a box) $[x]$ by an inclusion function for f is always still an interval (or a box) which contains the true image of $[x]$. Of course, there are infinitely many inclusion functions for a given real function. One of them is minimal but could be difficult to find. Then, the so-called natural inclusion function will be used. The natural inclusion function is simply obtained by replacing each operator in the associated real function by its interval equivalent, and each usual function $(\sin, \cos, \exp, \sqrt{\ })$ by a suitable interval counterpart. It is important to note that the natural inclusion function depends on how the real function is written and especially on the repetition of the same variable. For example, the two real functions

$$f_1(x) = x \text{ and } f_2(x) = x + x - x \tag{27}$$

are obviously equal, but their associated natural inclusion function are not. Indeed:

$$\begin{cases} [f_1]([1;2]) = [1;2] \\ [f_2]([1;2]) = [1;2] + [1;2] - [1;2] = [0;3] \end{cases} \tag{28}$$

To be sure that the inclusion function is an acceptable approximation of the real one, the inclusion function has to meet two conditions: It has to respect the inclusion

$$\forall [x], [y] \in \mathbb{IR},\ [x] \subset [y] \Rightarrow [f]([x]) \subset [f]([y]) \tag{29}$$

and, given a sequence of intervals (or boxes) with a size converging to *0*, the size of the image of this sequence by the inclusion function has to converge to 0 too.

$$\forall ([x_n])_{n\in\mathbb{N}} \in \mathbb{IR}^{\mathbb{N}},\ (\text{Size}([x_n]) \to 0) \Rightarrow (\text{Size}([f]([x_n])) \to 0) \tag{30}$$

2.4 Data Fit and Set Inversion

As outlined in the introducting Sect. 2.1, modelization problems can be reformulated in the language of interval analysis. With $(x_i, y_i)_{1\leq i\leq n}$ being a set of measured data, $(e_i)_{1\leq i\leq n}$ being the corresponding measurement accuracy and $f(x,p)$ being a model depending on several parameters $p = (p_1, p_2, \ldots, p_k)$, each measured value is associated with an interval according to the measurement accuracy:

$$[y_i] = [y_i - e_i, y_i + e_i] \tag{31}$$

Thus, a vector of parameters p is called feasible if

$$\forall\ 1 \leq i \leq n,\ f(x_i, p) \in [y_i] \tag{32}$$

and is called unfeasible otherwise. This definition has to be extended to the intervals. Obviously, a box of parameters $[p]$ is feasible if and only if it contains only feasible parameters, and is unfeasible if it contains only unfeasible parameters. But there is one more possibility: a box can contain both feasible and unfeasible parameters (see Fig. 3). Such a box is called indeterminate. More formally, a box $[p]$ is called

- feasible if

$$\forall 1 \leq i \leq n,\ [f](x_i, [p]) \subset [y_i] \tag{33}$$

- unfeasible if

$$\exists i,\ [f](x_i, [p]) \cap [y_i] = \emptyset \tag{34}$$

- indeterminate otherwise

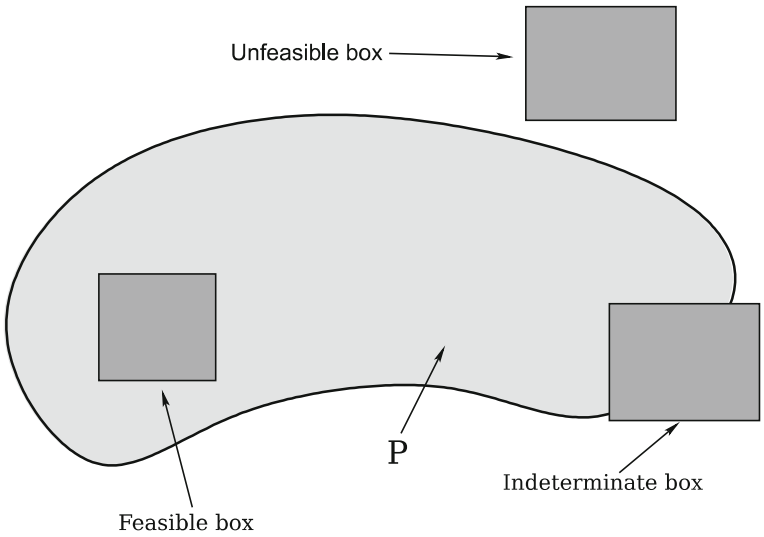

Fig. 3 Feasible, unfeasible and indeterminate boxes; P denotes the complete set of feasible parameters

Thus, the set P of all feasible parameters can be defined as the inverse image of the $[y_i]$ by a specific function. Thus, let

$$F : \mathbb{R}^k \to \mathbb{R}^n \tag{35}$$

$$p \to \begin{pmatrix} f(x_1, p) \\ f(x_2, p) \\ \vdots \\ f(x_n, p) \end{pmatrix} \tag{36}$$

be the function taking a vector of parameter P and returning all the corresponding calculated values and

$$[y] = \begin{pmatrix} [y_1] \\ [y_2] \\ \vdots \\ [y_n] \end{pmatrix} \tag{37}$$

the boxes of all measured intervals as defined in Eq. (31). So the set P of all feasible parameters is exactly:

$$P = \{p \mid F(p) \in [y]\} = F^{-1}([y]) = \bigcap_{i=1}^{n} f_i^{-1}([y_i]) \tag{38}$$

where $f_i(p) = f(x_i, p)$. Thus, the set of feasible parameters can also be considered as the set of parameters which satisfies a system of constraints.

Theoretically, it is very easy to test if a box of parameters $[p]$ is feasible or not but in practice it requires the ability to calculate $f(x_i, [p])$ which is not always an interval or easy to calculate. That is why the (natural) inclusion function $[f]$ will be used instead.

3 From S.I.V.I.A. to S.A.D.E.

S.I.V.I.A. (Set Inversion Via Interval Analysis) is a set inversion algorithm introduced by Jaulin [16, 17]. The algorithm is able to solve a large range of various problems by finding a list of boxes approximating the inverse image of a given set of real numbers or vectors (i.e. a subset of $\mathbb{R}^k$) by a given function. S.I.V.I.A. is a "branch and bound" algorithm: it first tests if a given box is feasible or unfeasible. If the box is indeterminate, the box is cut into 2 parts and each part is tested recursively. As S.I.V.I.A. has exponential complexity, a subroutine called contractor, which increases the speed of the main algorithm, is first presented.

3.1 Contractor

Given a box $[p]$, a contractor (C) is a subroutine which decreases the size of $[p]$ by removing some unfeasible parameters (as illustrated in the Fig. 4). So a function C is a contractor if and only if:

$$\forall [p] \in \mathbb{IR}^k,\ C([p]) \subseteq [p] \text{ and } C([p]) \cap P = [p] \cap P \tag{39}$$

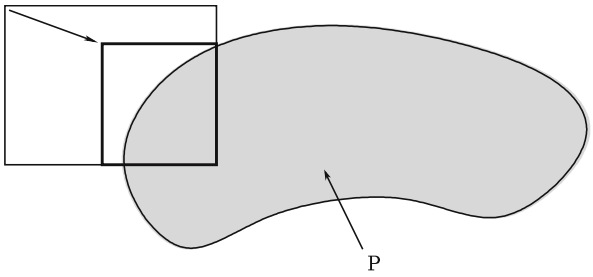

Fig. 4 Action of a contractor: it reduces the size of the current box without removing any feasible parameter. Thus, the resulting box fits the set P of feasible parameters more accurately

There are several ways to implement a contractor. One of them uses a "forward-backward propagation". Let $[p_0]$ be a box of parameters and

$$\begin{cases} f(x_1, p) \in [y_1] \\ f(x_2, p) \in [y_2] \\ \vdots \\ f(x_n, p) \in [y_n] \end{cases} \tag{40}$$

be the system of constraints corresponding to a fit problem. For each single constraint, the value of $f(x_i, [p_0])$ is calculated (forward propagation) and compared to the interval $[y_i]$. Then, the difference is propagated back to the parameter box $[p_0]$. Finally, the found box is contracted using the following constraint, and so on. This process is repeated as long as the contraction has a significant effect. For example, let

$$[p_1] \times [p_2] = [2; 3] \times [0; 1] \tag{41}$$

be a box of parameters,

$$[y] = [6; 10] \tag{42}$$

be an interval,

$$f(p_1, p_2) = p_1 \times \exp(p_2) \tag{43}$$

be a real function and

$$f(p_1, p_2) \in [y] \tag{44}$$

be a single constraint. The aim of the contractor is to remove some values from $[p_1]$ and $[p_2]$ which do not satisfy the constraint:

- Forward propagation: The expression is simply calculated but in order to make the backward propagation possible, each partial calculation is stored as some variable.

$$\begin{cases} [z] = \exp([p_2]) = [\exp(0); \exp(1)] = [1; 2.7183] \\ [y_p] = [p_1] \times [z] = [2; 3] \times [1; 2.7183] = [2; 8.1549] \end{cases} \tag{45}$$

- the values from $[y_\mathrm{p}]$ which do not satisfy the constraint are removed

$$[y^*] = [y_p] \cap [y] = [2; 8.1549] \cap [6; 10] = [6; 8.1549] \tag{46}$$

- backward propagation: the result is propagated back to remove any inconsistent value from the initial parameters:

$$\begin{cases} [p_1^*] = ([y^*]/[z]) \cap [p_1] = [2.2072; 3] \\ [z^*] = ([y^*]/[p_1^*]) \cap [z] = [2; 2.7183] \\ [p_2^*] = (\log([z^*])) \cap [p_2] = [0.6931; 1] \end{cases} \tag{47}$$

Thus, this process returns a new box $[2.2072; 3] \times [0.6931; 1]$ which is smaller than the initial one, but contains exactly the same feasible parameters.

3.2 Set Inversion Via Interval Analysis (SIVIA)

It is now possible to describe the S.I.V.I.A. algorithm. An initial box $[P_0]$ which represents the initial search space is taken as input: The initial box is chosen large enough to be sure that the expected solution set P of parameters satisifes the condition $P \subset [P_0]$. The second input value is a small positive real number η which determines the precision of the algorithm. If a box is indeterminate but smaller than η, this box is still accepted. This guarantees that the algorithm terminates. The fact that the inclusion function satisfies the conditions (29) and (30) guarantees also the correctness of the algorithm, i.e. that by choosing the value of η small enough, it is possible to make the approximation of P as precise as desired. The original version of S.I.V.I.A. returns an inner and an outer approximation of the set P of all feasible parameters, but for the sake of simplicity, the algorithm presented in this article only returns an outer approximation. Some basic structures used in the algorithm are defined here:

- A List is a structure which can store a list of elements. An element is added to the list with the function Push (List, element).
- A Stack is a LIFO (Last In, First Out) structure. The function Push (Stack, element) adds an element to the top of the Stack, and the function Pop (Stack) returns the element which is on the top of the stack and removes it from the stack.

The precise description of the algorithm is given Fig. 5. S.I.V.I.A. (Set Inversion Via Interval Analysis) has some advantages compared to LSA (Least Squares Approximation):

- This is a global algorithm. If there are some parameters which satisfy the constraints, they will all be found.
- Conversely, if there is no parameter which satisfies the constraints, an empty set will be returned. Therefore, S.I.V.I.A. also gives a strong criterion to estimate the number of relaxations for a given dielectric spectrum.
- All the returned parameters belong to the initial search space. So there is no risk to find non-sense parameters, such as negative values.

- It is possible to add some arbitrary constraints that the parameters have to satisfy. It is also possible to use in only one step, several systems of constraints coming from different sources (for example to fit simultaneously the real and imaginary part of a set of data) or to use several sets of data coming from repeated measurements (data accumulation) in order to increase the quality of the result.
- The returned list of boxes can be reinterpreted to find an interval for each parameter, which is guaranteed to contain the true parameter value, and which size is directly linked to the measurement accuracy.

Fig. 5 S.I.V.I.A. (Set Inversion Via Interval Analysis)

```
List L
Stack S
Push (S, [P0])
While Not ( is Empty (S) )
      [pc] ← Pop (S)
       Contract ( [pc] )
       If [pc] is feasible
            Push ( L, [pc] )
       End If
       If [pc] is indeterminate
            If Size ( [pc] ) ≤ η
                 Push ( L, [pc] )
            Else
                 { [p1], [p2]} = Bisection ( [pc] )
                 Push ( S, [p1] )
                 Push ( S, [p2] )
            End If
      End If
End While
Return L
```

Figure 6 shows a symbolic set of parameters approximated by a list of boxes which entirely cover the parameter set with an accuracy determined by the chosen value for the algorithm parameter η (the maximal size of indeterminate boxes which are still accepted). The list of boxes has to be post-processed in order to

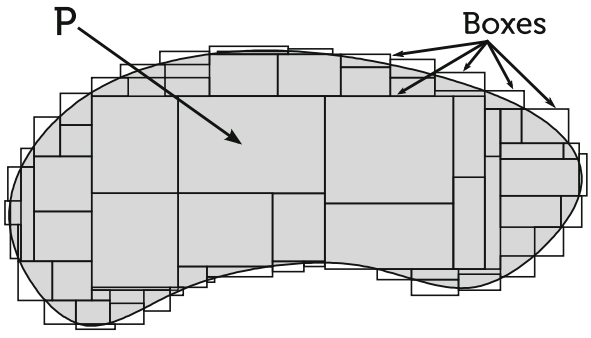

Fig. 6 The complicated set P is approximated by many boxes which are easier to handle

obtain an interval for each parameter of the applied physical model. Thus, S.I.V.I.A. is really efficient for a broad range of hard problems, such as robots localization, parameter estimation or stability analysis, and especially for very badly conditioned problems. For such problems, S.I.V.I.A. provides a detailed description of the part of the parameter space which solves the problem. However, in the specific case of experimental data fitting using a physical model with a lot of parameters, this feature of S.I.V.I.A. results in some redundancy of the parameters. This is caused by the symmetry of the model, i.e. the set P is not a connected set but it contains several connected components that correspond to all possible permutations of the parameters which let the model invariant. Then, the post-processing of the list could be time-consuming. So it is necessary to find a way to select a single connected component.

3.3 How to Modify S.I.V.I.A. for Dielectric Spectroscopy

In dielectric spectroscopy, the dielectric function ε^* measured as a function of frequency provides the experimental data set which consists of the two sub-sets for the real and the imaginary parts ε' and ε'', respectively. For dielectric data fit, the results given by S.I.V.I.A. are too precise: the list S.I.V.I.A. returns is not easy to process and a perfect approximation of P is not necessary as only an interval for each parameter is needed. Thus, it is possible to modify this algorithm to make it applicable to most complex cases.

3.3.1 Returned Values

As only an interval is needed for each parameter, the first idea is to approximate the bounding box of P, instead of P itself. The bounding box of P is simply the smallest box $[P]$ containing P. Each time the algorithm finds a feasible box of parameters, it will store the convex union of the current result and the newly found box instead of pushing this new box into a List. The convex union of 2 intervals is defined by:

$$\big[[x] \cup [y]\big] = [Min(x^-, y^-); Max(x^+, y^+)] \tag{48}$$

The convex union of 2 boxes is defined componentwise. In other words, the convex union of 2 boxes is the bounding box of the classical union of the 2 boxes. Thus, the returned value is not a List anymore, but a single box. The occupied memory space depends only on the size of the Stack, which is generally negligible.

Another positive consequence is, as the convex union of 2 boxes is bigger (in the sense of inclusion) than the standard union, that some parameters can already belong to the currently found box without having been processed before. This situation appears very frequently, so before contracting and testing a box, the algorithm will first check if the box which will be tested, already belongs to the currently found box. If yes, then the algorithm has nothing to do and can move along to the following box in the Stack saving a lot of computing time (see Fig. 7).

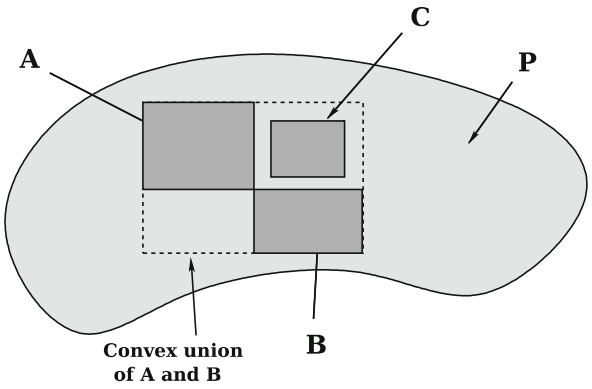

Fig. 7 *A* and *B* are feasible boxes. *C* is accepted without having been processed as it already belongs to the convex union of *A* and *B*

The computing time decreases a lot by using this simple technique. However, this technique is not justified if P is a non-connected parameter set. Indeed, the bounding box of a non-connected set includes all the connected components of the set which are generally far from each other. Thus, having a method to select one single connected component is now not only a question of saving computing time, but also a question of relevance of the result (as illustrated by Fig. 8).

3.3.2 Dealing with Symmetry

It is now necessary to find a way to select one single connected component from the set of feasible parameters. The easiest way is to assume that, for example (in the case of r relaxation times τ_i associated with r dielectric relaxations),

$$\forall i \in \{1 \dots r-1\}, \quad \tau_i < \tau_{i+1} \tag{49}$$

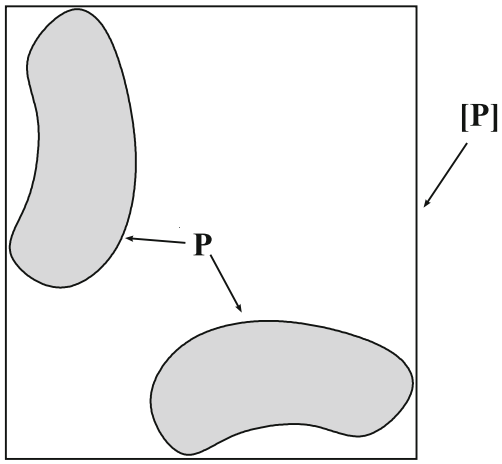

Fig. 8 Bounding box of a non-connected set. Because of the symmetries, the set of feasible parameters is divided into two components which are far one from the other. Thus, the bounding box of the whole set is not relevant at all, whereas the bounding box of one of the components is

But, as the algorithm deals with intervals, this constraint cannot be applied directly. However, it is obvious that this relation implies that the upper bound of $[\tau_i]$ cannot be greater than the upper bound of $[\tau_{i+1}]$, and conversely the lower bound of $[\tau_{i+1}]$ cannot be smaller than the lower bound of $[\tau_i]$. So the algorithm will remove some (and potentially all) values from $[\tau_i]$ by using:

$$\text{For } i = 1 \text{ To } r-1, \quad [\tau_i] \leftarrow \left[\tau_i^-; \min(\tau_i^+, \tau_{i+1}^+)\right] \tag{50}$$

and

$$\text{For } i = 1 \text{ To } r-1, \quad [\tau_{i+1}] \leftarrow \left[\max(\tau_i^-, \tau_{i+1}^-); \tau_{i+1}^+\right] \tag{51}$$

Again, there is a positive consequence of that. The size of $[\tau_i]$ could decrease by this process, or $[\tau_i]$ could become empty. Obviously, there are much more parameters which do not satisfy this constraint than parameters which do, so a lot of boxes will be simply removed during this process (namely, when $\tau_{i+1}^+ < \tau_i^-$, or when $\tau_i^- > \tau_{i+1}^+$ for some i) even if they are mathematically feasible, decreasing considerably the computing time. Moreover, the proportion of boxes which do not satisfy this constraint increases when the number of parameters increases. Then, the increase of computing time due to the increase of the number of parameters is partially compensated by the decrease of computing time due to this process.

3.3.3 Bisection

As the parameter τ_i (relaxation times) and the corresponding interval $[\tau_i]$ could cover many orders of magnitude it makes no sense to cut this interval in the middle. Their size will often be greater than the size of the measured parameter $[\Delta\varepsilon_i]$ [see Eq. (15)]. Thus, plenty of computing time would be wasted by making a lot of useless bisections. Instead, it will be useful to define a specific size and a specific bisection for τ_i. So, by rewriting

$$[\tau_i] = [10^{T_i^-}; 10^{T_i^+}] \tag{52}$$

the size of τ_i is defined by:

$$\text{Size}([\tau_i]) = T_i^+ - T_i^- \tag{53}$$

and the bisection by:

$$\text{Bisection}([\tau_i]) = [10^{T_i^-}; 10^{T_i^- + (T_i^+ - T_i^-)/2}], [10^{T_i^- + (T_i^+ - T_i^-)/2}; 10^{T_i^+}] \tag{54}$$

So, exactly for the same reason that it is reasonable to use a logarithmic axis for the frequency to plot a function, the algorithm will use a "logarithmic bisection" for $[\tau_i]$.

3.4 S.A.D.E. Algorithm

By using these modifications, it is possible to give a new algorithm, S.A.D.E. (as S.I.V.I.A. Applied to DiElectric spectroscopy) which will also find the optimum number of relaxations. This algorithm is described in Fig. 9. The algorithm will loop until it finds the minimum number of relaxations which make the result being non-empty. Note that if it is possible to fit the data with r relaxations, trying to fit it with more than r relaxations will give too many degrees of freedom and will lead to a very bad result. In fact, if there are more than r relaxations, the additional relaxations are masked by the measurement errors. Then, the numbers of relaxations provided by S.A.D.E. should be considered as optimum.

4 S.A.D.E. Examples

4.1 A First Test by Using Home-Made Data

First, the widely used Havriliak-Negami function was tested. Using fixed parameters for only one relaxation ($\varepsilon_\infty = 4$, $\Delta\varepsilon = 2$, $\tau = 1s$, $\alpha = 0.5$ and $\beta = 0.5$), a perfect data set was created using the Havriliak–Negami model [according to Eq. (3)]. Of course, as it was a "home-made" data set, the experimental error on data points would be zero. Hence, this data set was tested as a real measured data set by adding

```
Box [P] ← ∅
Integer relaxation Number ← 1
Stack S
While [P] = ∅
    Push ( S, [P0] )
    While Not ( is Empty (S) )
        [pc] ← Pop (S)
        If Not ( [pc] ⊂ [P] )
            Break Symetries ( [pc] )
            Contract ( [pc] )
            If [pc] is feasible,
                [P] ← [[P]∪ [pc]]
            End If
            If [pc] is indeterminate
                If Size ( [pc] ) ≤ η
                    [P] ← [[P] ∪ [pc]]
                Else
                    {[p1],[p2]}= Bisection ([pc])
                    Push (S, [p1] )
                    Push (S, [p2] )
                End If
            End If
        End If
    End While
    relaxation Number ← relaxation Number + 1
End While
Return [P]
```

Fig. 9 S.A.D.E. (as Set inversion via interval analysis Applied to DiElectric spectroscopy)

Table 1 Calculated parameters, from a "home made" Havriliak–Negami relaxation

Parameter	Interval	
ε_∞	3.96	4.04
$\Delta\varepsilon$	1.9853	2.0538
τ	0.983181 s	1.013406 s
α	0.4847	0.5112
β	0.4836	0.5059

an accuracy of 1%. Then, with a 0.01 box size (η, described in Sect. 3.2) and within less than 1 second computing time on a common desktop computer, the resulting parameter intervals were obtained by S.A.D.E. and presented in Table 1.

This result is very promising since the boundaries of the intervals deviate by less than 4% from the exact parameter values and hence it validates the program S.A.D.E..

Secondly, the Debye function was tested. This function is the only one having a physical meaning and will be used for our experiments in the next part. Using fixed parameters for only one relaxation ($\varepsilon_\infty = 2$, $\Delta\varepsilon = 3$, $\tau = 10^{-6}$s), a perfect data set for the Debye model (see equation (2)) was created. Of course, as it was a "home-made" data set, the experimental error on data points would be zero. Again, this data set was tested as a real measured data set by adding an arbitrary error of 1%. With a 0.1 box size and with less than 1 second computing time on a common desktop computer, the resulting parameter intervals were calculated and presented in Table 2.

As for the previous example, this result is very promising since the boundaries of the intervals deviate by less than 2% from the exact parameter values. Moreover, as this model was simpler than the HN model (less parameters), S.A.D.E. was able to calculate the parameters with a smaller error although the box size was ten times bigger.

Table 2 Calculated parameters, from a "home made" Debye relaxation

Parameter	Interval	
ε_∞	1.97963	2.02038
$\Delta\varepsilon$	2.93991	3.05322
τ	9.89954×10^{-7} s	1.01024×10^{-6} s

4.2 Test with Real Experimental Curves

4.2.1 Experimental Details

The pure diglycidylether of bisphenol A (DGEBA) DER 332 from DOW Chemical was studied by dielectric broadband measurements in a frequency range of *0.1* Hz to 10^6 Hz using a Novocontrol High Resolution Dielectric Alpha Analyser with automatic temperature control by a Quatro cryosystem. For the examples presented here, 200 data points were measured in the frequency range at −90 and 60°C.

The viscous DGEBA was placed between stainless steel electrodes, with a teflon spacer in order to have a well-defined geometry.

As the number of relaxations is higher for real data points than for Home-made curves, the common desktop computer was changed to a Dual Opteron machine containing two 2.4 GHz/64 Bits CPUs and at least 4 GB of RAM. The operating system was Suse Linux 10.0. A parallelized version of the algorithm was implemented (in the C++ programming language) in order to take advantage of the two CPUs.

4.2.2 Application of S.A.D.E. to Multiple Relaxations Data

Figures 10 and 11 show ε' and ε'' as measured at −90°C for pure DGEBA. By eye, only two relaxation processes would be recognized for the curves. However, the shape of the spectra is complex. We choose a sum of Debye relaxators to describe it [see Eq. (15)]. Then, S.A.D.E. identifies the five relaxations represented in Figs. 10 and 11. The five relaxation curves were drawn using the middle point of each computed interval (in the logarithmic sense for the parameters τ_i). With a 0.1 box size and with 382 s computing time on the cluster, the resulting parameter intervals were calculated and presented in Table 3.

This result is very promising since the boundaries of each of the intervals are not far apart. Of course, the width of the intervals can be reduced further as they depend on the error of the experimental data points and on the given box. But the smaller the box η is made, the higher the calculation time will be. Let us note that S.A.D.E. tries first to fit these data points by only one Debye relaxator. As the fit failed, it tries with two to five relaxations (using five Debye relaxators lead to the determination of eleven parameter intervals!). If we try to fit the data points with more than 5

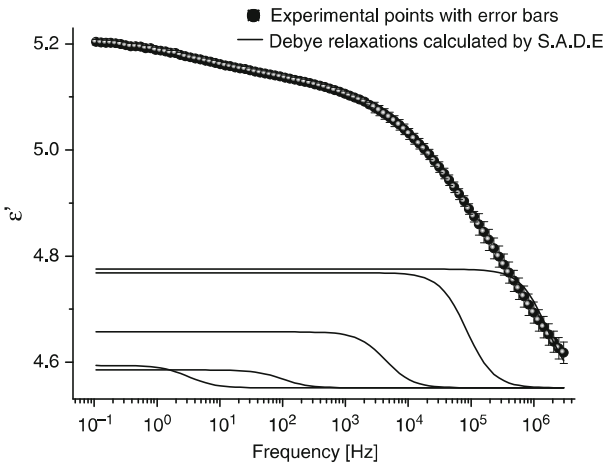

Fig. 10 Dielectric spectroscopy real permittivity of the DGEBA DER 332 prepolymer at −90°C and the corresponding five Debye relaxations as calculated by S.A.D.E.

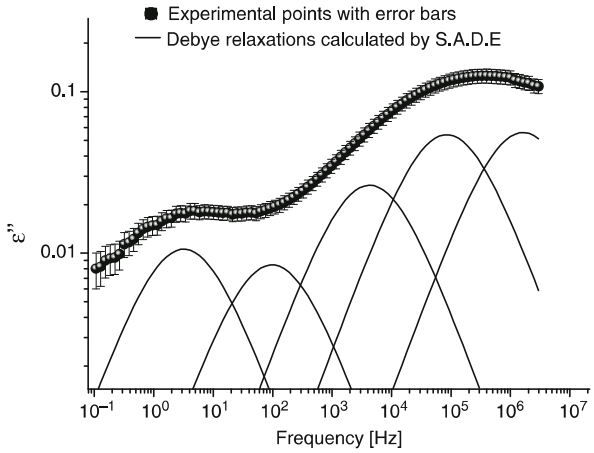

Fig. 11 Dielectric spectroscopy imaginary permittivity of the DGEBA DER 332 prepolymer at −90°C and the corresponding five Debye relaxations as calculated by S.A.D.E.

Table 3 Calculated parameters, using the equation (4), from experimental data set at −90°C presented in the Figs. 10 and 11

Parameter	Interval	
ε_∞	4.50144	4.60168
$\Delta\varepsilon_1$	0.176763	0.270755
τ_1	7.23×10^{-8} s	1.24×10^{-7} s
$\Delta\varepsilon_2$	0.164731	0.269318
τ_2	1.54×10^{-6} s	2.21×10^{-6} s
$\Delta\varepsilon_3$	0.0771269	0.133832
τ_3	2.74×10^{-5} s	4.70×10^{-5} s
$\Delta\varepsilon_4$	0.0239589	0.0437055
τ_4	1.19709×10^{-3} s	2.05353×10^{-3} s
$\Delta\varepsilon_5$	0.0321849	0.0524569
τ_5	3.69994×10^{-2} s	6.26434×10^{-2} s

relaxations, the additional relaxations will be masked by the measurement errors. Then, the numbers of relaxations provided by S.A.D.E. is considered as optimum. It is worth noting that these parameters were not found by classical least square approximation fitting routines.

4.2.3 Application of S.A.D.E. to the Global Problem: Data with Relaxations, Conductivity and Electrode Polarization

Figures 12 and 13 show ε' and ε'' measured at 60°C for the pure DGEBA DER 332. At first glance, only conductivity, electrode polarization and the beginning of a relaxation process (at high frequencies) can be identified for that measured dielectric spectrum. But two relaxations are found by S.A.D.E., and the DC-conductivity is well defined. As for the previous figures, the relaxation curves were drawn using the

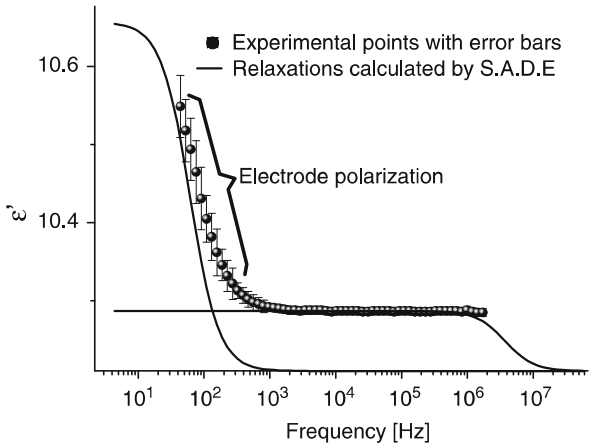

Fig. 12 Dielectric spectroscopy real permittivity of the DGEBA DER 332 prepolymer at 60°C and the corresponding two Debye relaxations as calculated by S.A.D.E.

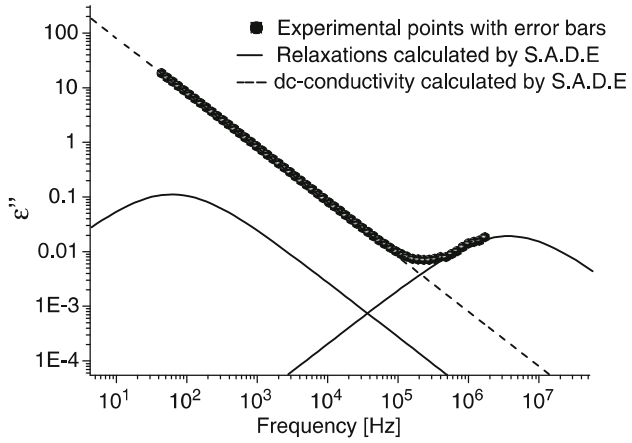

Fig. 13 Dielectric spectroscopy imaginary permittivity of the DGEBA DER 332 prepolymer at 60°C and the corresponding two Debye relaxations as calculated by S.A.D.E.

middle point of each computed interval (in the logarithmic sense for the parameters τ_i). With a 0.1 box size and with 4336 s computing time on the cluster, the resulting parameter intervals were calculated and presented in Table 4.

The first relaxation (at high frequencies, with the smaller relaxation time τ_1) corresponds to a dipole relaxation of the material. By just looking at the curve, it is not possible to say if some relaxators take place as only conductivity and electrode polarization could be identified, but S.A.D.E. clearly identified this material relaxation. In addition, the second relaxation (at low frequencies, with the higher

Table 4 Calculated parameters, using the Eq. (15), from experimental data set at 60°C presented in the Figs. 12 and 13

Parameter	Interval	
ε_∞	10.169	10.2691
$\Delta\varepsilon_1$	0.0308334	0.123296
τ_1	1.56023×10^{-8} s	6.97831×10^{-8} s
$\Delta\varepsilon_2$	0.222722	0.671
τ_2	1.00487×10^{-3} s	4.17685×10^{-3} s
σ_{DC}	3.95878×10^{-8}S $\times$ m^{-1}	4.9343×10^{-8}S $\times$ m^{-1}

relaxation time τ_2) corresponds to the electrode polarization. As mentioned in the paragraph "Polarization at electrode and phase boundaries" (see Sect. 1.1.3), the electrode polarization is well modeled as a strong Debye relaxator. In fact, close to the electrodes the mobile charges can be blocked and piled up at the phase boundaries, as described by Maxwell, Wagner, and Sillars [11–13], whose relation is similar to the Debye equation (see Eqs. (2) and (10)). Electrode polarization is reasonable since the DGEBA contains traces of ions from synthesis. Let us note that these relaxations were not visible in the data plot (Figs. 12 and 13) and not possible to fit by classical ways. Finally, the DC-conductivity was calculated simultaneously with the Debye relaxations: it perfectly fits the rise observed in the imaginary part of the material permittivity (Fig. 13).

In conclusion, as S.A.D.E. found some Debye-like relaxations we can be sure that a Debye relaxator also describes electrode polarization reasonably well.

5 Conclusion

Dielectric spectroscopy is an extremely versatile method for characterizing the molecular dynamics over a large range of time scales. Unfortunately, the extraction of model parameters by data fitting is still a crucial problem which is now solved by our program S.A.D.E.

S.A.D.E. is based on the algorithm S.I.V.I.A. which was proposed and implemented by Jaulin [16,17] in order to solve constraint satisfaction problems. The problem of dielectric data analysis is reduced to a problem of choosing the appropriate physical model. In this article, Debye relaxations were used and validated to fit the relaxations of a DGEBA prepolymer and the polarization of the spectrometer electrodes. The conductivity was evaluated too.

Acknowledgement The authors acknowledge Professor Springborg who granted the access to the computing cluster and Michael Bauer for his help in using it. We also thank Luc Jaulin for his most useful advices.

Software Availability

S.A.D.E. is freely available on the website of M. Aufray (http://maelenn.aufray.free.fr).

References

1. P. Debye, *Polar Molecules* (Chemical Catalog, Reprinted by Dover Publications, New York, 1929)
2. Havriliak, S. Negami, Polymer **8**, 161 (1967), http://www.sciencedirect.com/science/article/B6TXW-48FBVVM-4T/2/ffb6daa2b39f773a1666f9d2cb8d4ed8
3. K.S. Cole, R.H. Cole, J. Chem. Phys. **9**, 341 (1941)
4. K.S. Cole, R.H. Cole, J. Chem. Phys. **10**, 98 (1942)
5. D. Davidson, R.H. Cole, J. Chem. Phys. **18**, 1417 (1950)
6. M. Mangion, M. Wang, G. Johari, J. Polym. Sci. B Polym. Phys. **30**(5), 433 (1992), http://www3.interscience.wiley.com/cgi-bin/fulltext/104051384/PDFSTART
7. Corezzi, S. Capaccioli, G. Gallone, A. Livi, P. Rolla, J. Phys. Condens. Matter 9, 6199 (1997), http://www.fisica.unipg.it/~corezzi/
8. N. Axelrod, E. Axelrod, A. Gutina, A. Puzenko, P. Ben Ishai, Y. Feldman, Meas. Sci. Technol. **15**, 1 (2004)
9. F. Stickel, E. Fischer, R. Richert, J. Chem. Phys. **104**(5), 2043 (1996)
10. R.M. Fuoss, J.G. Kirkwood, J. Am. Chem. Soc. **63**(2), 385 (1941)
11. J. Maxwell, *A Treatise on Electricity and Magnetism,* vol 1. (Dover Publications, New York, 1954)
12. K.W. Wagner, Archiv für Elektrotechnik **2**(9), 371 (1914). URL http://www.springerlink.com/content/xr0617448810/?p=a72f93d829804c93afa018301302e470&pi=624
13. R. Sillars, Proc. Inst. Electr. Eng. **80**, 378 (1937)
14. D.W. Marquardt, J. Soc. Ind. Appl. Math. **11**(2), 431 (1963)
15. K. Levenberg, Q. J. Appl. Math. **2**, 164 (1944)
16. Jaulin, E. Walter, Automatica **29**(4), 1053 (1993), http://www.sciencedirect.com/science/article/B6V21-47WVRTR-5T/2/35ab6e2405d16f0886be1fcfc21d687c
17. Jaulin, E. Walter, Math. Comput. Simul. **35**(2), 123 (1993), http://www.sciencedirect.com/science/article/B6V0T-45DHW9D-2/2/f8024c6b00508f6f3ce7087b39f78f59