REGULARIZED POSITIVE EXPONENTIAL SUM (REPES) PROGRAM – A WAY OF INVERTING LAPLACE TRANSFORM DATA OBTAINED BY DYNAMIC LIGHT SCATTERING

Jaromir JAKES

Institute of Macromolecular Chemistry,
Academy of Sciences of the Czech Republic, 162 06 Prague 6, Czech Republic

Received February 21, 1995
Accepted September 17, 1995

Dedicated to Dr Blahoslav Sedlacek on the occasion of his 70th birthday.

Computer program REPES, which has been used in treating dynamic light scattering data for eight years but never published, and the algorithm used within it are described in detail. A list of papers quoting REPES and a brief discussion of suitability and performance of regularizing methods in inverting Laplace transform are given.

A homodyne dynamic light scattering (DLS) experiment follows fluctuations of light scattered by a sample illuminated by a laser beam. A fluctuation treatment starts from forming the intensity autocorrelation function $g_2(t)$,

$$g_2(t) = \langle I(T)I(T+t)\rangle / \langle I(T)\rangle^2 - 1 \quad , \qquad (1)$$

where $\langle\ \rangle$ means averaging over time T, $I(T)$ is the scattered intensity at time T, and t is the time delay. $\langle I(T)\rangle^2$ is sometimes called the homodyne base. A direct relation to molecular dynamics is obtained for field autocorrelation function $g_1(t)$, which, for a Gaussian field, is related to $g_2(t)$ by Siegert relation

$$g_2(t) = g_1^2(t) \quad . \qquad (2)$$

When $g_2(t)$ means the autocorrelation function of the intensity measured with a detector of a non-zero detecting area, like in a real experiment, a correction factor $\beta < 1$ stands on the right-hand side of Eq. (*2*). Absolute value at $g_1(t)$ is omitted for working with $g_1(t)$ real. For molecular dynamics with a single relaxation process of relaxation time τ, the relation $g_1(t) = \exp(-t/\tau)$ holds; for several (r) relaxation processes of relaxation times τ_i the field autocorrelation function becomes

$$g_1(t) = \sum_{i=1}^{r} a_i \exp(-t/\tau_i) \qquad (3)$$

with positive a_i's and $\sum_{i=1}^{r} a_i = 1$. An infinite relaxation time is allowed; in this paper, τ_1 is reserved for it. Its amplitude (a_1) is sometimes called the heterodyne base (in relation to heterodyne DLS technique measuring $g_1(t)$ directly at the expense of worsening the signal-to-noise ratio). For a continuous distribution $w(\tau)$ of relaxation times, Eq. (*3*) is replaced by

$$g_1(t) = \int w(\tau) \exp(-t/\tau)\, d\tau \qquad (4)$$

with $w(\tau) \geq 0$ and $\int w(\tau)\, d\tau = 1$. With a log τ abscissa, commonly used when data cover a wide delay-time range, $\tau w(\tau)$ should be plotted over log τ for obtaining an equal-area representation of τ distributions. When the β factor is absorbed into the $g_1(t)$ function, we have $\sum_{i=1}^{r} a_i = \beta^{1/2}$ with Eq. (*3*) and $\int w(\tau)\, d\tau = \beta^{1/2}$ with Eq. (*4*).

Seeking $w(\tau)$ as a sum of δ-peaks yields Eq. (*3*); the least-squares approach to this problem is known as the positive exponential sum (PES) method. It was proven[1] that PES with $g_1(t)$ data yields unique solution with a finite number r of relaxation times τ_i. Later, PES solution was shown to be the least-squares solution of Eq. (*4*) over any $w(\tau)$ functions and a PES program using $g_2(t)$ data was described[2]. PES problem is well-posed (cf. ref.[3] Appendix, Theorem 6; a proof that Laplace kernel belongs to the set considered there is beyond the scope of this paper). Ill-posedness appears when considering statistically acceptable deviations from the least-squares solution. Usually, it turns out that a δ-peak may be spread in a wide variety of ways over a wide τ interval (a half decade or even more) without an appreciable error increase.

CONTIN program[4] seeks a smooth solution among the above variety by adding a regularizing term (an integral of squared solution's derivative of some order), multiplied by a factor (α), to the least-squares error term. The order of the derivative is selected between 0 (the solution itself) and 5 by the NORDER parameter. $g_1(t) = g_2^{1/2}(t)$ data and numerical integration (i.e., Eq. (*3*) with some fixed integration τ grid and non-negative a_i's) are used and derivatives are converted to differences. The α factor is selected so as to yield a reasonable compromise between smoothing and error increase.

F distribution function is used to calculate the probability of rejecting a solution. Probability 1 to Reject is defined by

$$I_x(p_0/2, (n - p_0)/2) = \mathrm{B}_x(p_0/2, (n - p_0)/2)/\mathrm{B}(p_0/2, (n - p_0)/2) \ , \qquad (5)$$

where I_x and B_x are the incomplete Beta functions[5], B is the complete Beta function[5], p_0 is the number of degrees of freedom in the reference solution, n is the number of data items, and $x = 1 - E_0/E$ with E and E_0 being the error term values for the actual and reference solutions, respectively. As the reference solution, that calculated with lowest α is used. Probability 2 to Reject is defined by

$$I_x((p_0 - p)/2, (n - p_0)/2) \ , \qquad (6)$$

where p is the number of degrees of freedom in the actual solution. The number of degrees of freedom is defined by

$$\mathrm{trace}\,((\boldsymbol{H} + \alpha\boldsymbol{R})^{-1}\boldsymbol{H}) = s - \alpha \times \mathrm{trace}\,((\boldsymbol{H} + \alpha\boldsymbol{R})^{-1}\boldsymbol{R}) \ , \qquad (7)$$

where ***H*** and ***R*** are matrices representing error and regularizing terms, respectively. In ***H*** and ***R***, only rows and columns corresponding to nonzero amplitudes a_i (s in number) are considered. (After necessary rearrangements, Eq. (*7*) is identical with Eq. (*A1*) of ref.[6].) An α grid is used to calculate a set of solutions; that with Probability 1 to Reject closest to 0.5 is chosen as the best. Inspecting other solutions may be sometimes useful.

Performing calculations with my PES program[2], I soon found that obtaining the final solution was cumbersome and lengthy. Since I was unable to write an automatic program for refining both amplitudes and relaxation times at that time, I decided to fix relaxation times to a grid uniformly spaced on log τ scale and wrote a positive exponential sum on a grid (PESG) program. It turned out that under the double precision (16 significant digits) the algorithm used was numerically stable (with a single practically irrelevant exception[2]) without using the singular value decomposition procedure[7] used by CONTIN and without any regularization, simply by setting Hessian matrix and solving for corrections by Gauss elimination. I recognized that the result was nearly identical with the reference solution of CONTIN and felt that adding a regularization to PESG program would yield a program performing a regularized Laplace transform inversion in a much simpler and faster way than CONTIN does.

A need for such a program appeared when I was trying to remove artifacts of the CONTIN method left even after my modification of the regularizor[8] enabling the use of

CONTIN for very wide autocorrelation function data. I found[8] that suppressing regularization around a δ-function essentially removed artifacts (the result was published later[9]); a next attempt with data simulated for a two-bin distribution with one bin narrow and high and the other wide and low (Fig. 5*c* in ref.[8]) yielded a good result when the regularizor had different weights over either bin. Hence, artifacts seemed to be caused by an improper distribution of the penalization among individual degrees of freedom. Such an effect was rough prior to my modification of the regularizor, but still persisted to some extent afterwards. Indeed, in the results with the artifacts, the number of degrees of freedom and Probability 2 to Reject were large. So, my idea was to find a weighting scheme for the regularizor yielding lowest Probability 2 to Reject (i.e., the lowest number of degrees of freedom) for a given Probability 1 to Reject. This was supported by the fact that, in the above two-bin distribution, the weight ratio of the regularizor over individual bins yielding the lowest number of degrees of freedom gave a good regularization. Unfortunately, programming this approach was very difficult. Some preliminary attempts by the method of steepest descent to adjust the weighting function of the regularizor (with zero NORDER parameter) seemed to confirm my fear that such an approach may yield some “gaps” in the weighting function leaving some sharp peaks and failing to smooth the result properly. In view of these two difficulties, I gave up.

The CONTIN program was quite unsuitable for such a research. An essential difficulty was that the chosen solutions yielded different Probability 1 to Reject with different regularizor weighting schemes, so that their comparison could not be used for evaluating the scheme suitability properly. A technical difficulty consisted in working with a large program having about 60 subroutines and producing a huge output. Hence, I added a regularizor to the above PESG program and obtained a regularized positive exponential sum (REPES) program. The program was completed in autumn 1987. To remove the major drawback of CONTIN, an automatic adjusting of α to a preselected Probability 1 to Reject was programmed within REPES. However, nearly all artifacts of CONTIN (with exception of that shown in Fig. 1 below) persisted in REPES. This is why I decided to delay the publication until I find an appropriate weighting scheme for the regularizor. On the other hand, REPES was much simpler and faster than CONTIN and I put it at disposal of my colleagues. Now I see that REPES, despite of a limited access, was mentioned in about 60 scientific papers and no better routine seems to exist. This made me write the present paper.

PROGRAM DESCRIPTION

REPES program minimizes the expression

$$E + \alpha R \ , \qquad (8)$$

where

$$E = \sum_k [f(t_k) - f_{k,\mathrm{exp}}]^2 \tag{9}$$

is an error term, $f_{k,\mathrm{exp}}$ is the autocorrelation function $g_2(t)$ datum at the delay time $t = t_k$, R is a regularizor, and

$$f(t) = [a_1 + \sum_{i=2}^{r} a_i \exp(-t/\tau_i)]^2 \ . \tag{10}$$

Relaxation times τ_i are restricted to a τ grid spreading from τ_2 to τ_r and equidistantly spaced on log τ scale. A slight modification of the program allows for statistical weights of data items in the error term E. Minimization proceeds with respect to the $\boldsymbol{a} \equiv (a_1, a_2, \ldots, a_r)$ vector subject to the condition $a_i \geq 0$. α is adjusted so as to yield the Probability 1 to Reject preselected by the user. $\sum_{i=1}^{r} a_i$ is left unconstrained. For NORDER = 2, the regularizor is

$$R = a_2^2 + (2a_2 - a_3)^2 + \sum_{i=3}^{r-1} (a_{i-1} - 2a_i + a_{i+1})^2 + (a_{r-1} - 2a_r)^2 + a_r^2 + Ba_1^2 \tag{11}$$

(corresponding to two extra grid points with zero amplitudes assumed at each grid margin); for NORDER = 0,

$$R = \sum_{i=2}^{r} a_i^2 + Ba_1^2 \ . \tag{12}$$

A penalization of the heterodyne base (the term Ba_1^2) is introduced to prevent moving amplitudes of long relaxation times into the base in the regularization when this is undesirable. A non-negative B constant is selected by the user. Note that in changing the grid density D (the number of grid points per decade), the B constant should be proportional to D^{-5} with NORDER = 2 and to $1/D$ with NORDER = 0 for obtaining comparable relative penalization of the base and of the grid amplitudes. The B constant leading to lowest Probability 2 to Reject (at constant Probability 1 to Reject) should be considered as most appropriate. Optionally, the homodyne base position may be changed (by subtracting a user-supplied constant from $f_{k,\mathrm{exp}}$ data) when an incorrect estimate of $\langle I(T)\rangle^2$ is suspected. Adjusting both bases simultaneously frequently yields ill-conditioning and hence was not tried. $f_{k,\mathrm{exp}}$ data may be scaled, the square root of the scaling factor then scales the a_i amplitudes.

When a too low value $\sum_{i=1}^{r} a_i$ is obtained, the missing amplitude may be thought to be located at very short relaxation times, where it contributes essentially nothing to $f(t)$. A too large value may be caused by incorrect data or by extending the τ grid down beyond the lowest delay time where an artifact large a_2 may result. A check that unscaled $\sum_{i=1}^{r} a_i$ ("Total intensity" + "base" in the output file) does not appreciably exceed $\beta^{1/2}$ (or 1 when β is unknown) is therefore recommended. Seeing a_2 is also useful.

A program run may treat several data files, each of which may contain several data sets.

Data Input

A data file name may be entered in the command line; if not, the program requests the user to enter it from the keyboard. The first data line should contain a data name, which is copied to the output. The next line should contain an integer n indicating the number of data items; n lines, each with a pair of (t_k, $f_{k,\mathrm{exp}}$) data when n is positive or a pair of ($\log t_k$, $f_{k,\mathrm{exp}}$) data when n bears the minus sign, should follow. When zero n is read in or an error is encountered at n reading, the program calls a READ subroutine which may be adapted by the user according to his needs. After terminating a data set calculation, the program tries to read another data set from the current data file. If an end of the file or an error is encountered, a new data file name is asked from the keyboard. Keying in "END" or "end" (without quotes) instead terminates the program run.

Setting a τ Grid

After successful reading in data, the program requests parameters DENSITY PER DECADE, FROM, TO, and BASE. The first parameter (D) means the number of grid points per one decade and is defaulted to four (to twelve in another program version); it is not necessarily an integer. The next two parameters define the grid extent, FROM is defaulted to t_1, TO to $3t_n$; t_1 and t_n are assumed to be the lowest and largest delay times, respectively. D should be at least one, TO at least twice the FROM; if not, a new request for parameters follows. $D \times \log \tau_2$ is obtained by rounding $D \times \log$ (FROM) to nearest integer and τ_r is obtained from TO similarly, so that unity always enters the grid and all integer powers of ten enter when D is an integer. More than one hundred grid points may be introduced with a PC of 640 kB memory. When the number of grid points allowed is exceeded, the program lowers D to largest acceptable integer and asks for confirmation. The user may confirm or limit the grid extent instead lowering *D*. The grid should not extend much beyond the lowest delay time.

The BASE parameter (defaulted to zero) is subtracted from $f_{k,\mathrm{exp}}$ data values allowing a shift of the homodyne base.

Regularization

The reference solution (i.e., non-regularized, with $\alpha = 0$ in Eq. (*8*)) is calculated first. Then three parameters PRRE, PEBA, and NORDER are asked. PRRE means user-selected Probability 1 to Reject and is defaulted to 0.5, PEBA is the *B* constant in Eq. (*11*) or (*12*). PEBA is defaulted to 0 and NORDER to 2 initially, afterwards they keep their current values. Entering a too large (≥1) Probability 1 to Reject value terminates the data set calculation, a too low (≤0) value allows to repeat the calculation with the grid and/or the homodyne base shift changed. When NORDER differs from both zero and two, it is set to two and parameters are re-asked.

After calculating a regularized solution, only Probability 1 to Reject (now named PRO1REJ) and PEBA are asked, a NORDER change is not allowed. When PRO1REJ is less than 0.5, it is defaulted to 0.5, otherwise it is defaulted to 1 (i.e., to terminate). When previous parameters are repeated, an extra refinement cycle is performed (a check of the iteration convergence). When only PEBA is changed and the base is absent ($a_1 = 0$), the parameters are re-asked. An improper Probability 1 to Reject value acts like above.

The desired Probability 1 to Reject values should be entered in an increasing order, otherwise convergence problems may arise.

Moment Calculation

Seven peak moments are calculated: the integral amplitude, the logarithmic (geometric) and arithmetic mean τ values, the relaxation rate $\Gamma = 1/\tau$ mean value, the root-mean-square (RMS) deviation of log τ from its mean, and the RMS deviations from means divided by means of τ and of Γ. A peak is considered to spread from a local minimum of a_i through the next local maximum to the next local minimum. A minimum amplitude a_i, when non-zero, is distributed into the neighbouring peaks in the ratio opposite to that of increases of a_i in going to respective peaks (e.g., when the increase $a_{i-1} - a_i$ is twice the increase $a_{i+1} - a_i$, one-third of a_i is counted to the peak containing a_{i-1} and two-thirds to the peak containing a_{i+1}). The moments are calculated also for all composite peaks spreading from a local minimum to another one, provided no zero a_i is embedded in the composite peak. This provides the user with moments of peaks wih a composite structure or when some local minima are considered as artifacts. Note that the logarithmic RMS deviation is more stable with respect to perturbations than the RMS deviations expressed in terms of τ or of Γ; therefore, logarithmic moments should be preferred. With a grid fine enough, all peaks in the reference solution are doublets or singlets separated by zero amplitudes on both sides (on one side when marginal). Their integral amplitudes and logarithmic mean τ values form a suitable starting iteration for PES program[2].

Results Output

An output file name may be entered in the command line together with the data file name; if not, it is generated from the first data file name by changing its extension to .RES, or to .RSL when the extension is .RES. First, the data set name (68 characters) and the number of data items is output. The homodyne base shift (BASE =) follows when nonzero. For each solution, the output starts with the number of nonzero a_i values, the grid density (D) rounded to nearest integer, a parameter IG ($\tau_i = 10^{(i+\mathrm{IG})/D}$, $i = 2, 3, \ldots, r$), the actual value of E (Eq. (*9*)), this actual E decreased by the E yielding required Probability 1 to Reject (an α value check, undefined and of no use in the reference solution), and α in one line. This is followed with a list of non-zero amplitudes (two triples of i, τ_i, a_i per line, τ_1 is set to 10^{30}). The heterodyne base a_1 (base =) follows when non-zero. The next line shows $\sum_{i=2}^{r} a_i$ (Total intensity =). Then peak moments are shown. All single peaks are numbered consecutively; for a composite peak, two integers show from which single peak to which one the composite peak extends; for a single peak just one integer is shown and the space for the other is left blank. For an isolated non-zero a_i, only the peak number, a_i, and τ_i are shown, followed by a zero to indicate zero second moment. The output of a reference solution is completed by PRO1REJ = 0 on a line, that of a regularized solution by showing Probability 2 to Reject, the number of degrees of freedom and NORDER in one line, and Probability 1 to Reject, PEBA (the B constant of Eq. (*11*) or (*12*)) and α in the next line. The output of all data sets treated within a single program run is stacked on in a single output file no output file change being enabled within a program run.

The PC screen displays hints to operate the program and the results of the calculation, the latter in a way similar to that in the output file. In 1991, I equipped REPES program with a graphic display on the screen, which optionally may be copied to a printer by the PrintScreen key. Data are displayed after reading in and selected data items may be discarded, e.g., when incorrect items appear. The number of discarded items and the list of their indices are shown in the output file. On completing a solution, a graph of the resulting τ distribution and a graph of residuals (i.e., $f_{k,\mathrm{exp}} - f(t_k)$ values) are displayed. Then the user may in turn request re-inspecting either the data or the τ distribution. When the discarded data set is changed in inspecting the residuals or re-inspecting the data, a repetitive calculation (starting from the reference solution) with the changed set is done. Keying in a negative value for Probability 1 to Reject causes a display of the data with an opportunity of changing the discarded data set before repeating the calculation; zero value causes skipping the display with no such opportunity.

ALGORITHM DESCRIPTION

Two methods are used for the least-squares refinement. The Newton–Raphson method[10] (NR) calculates Hessian matrix $\boldsymbol{H}$ of halved second derivatives of the minimized expression with respect to iterated parameters p_i and the gradient vector $\boldsymbol{g}$ of first derivatives (also halved) and uses $-\boldsymbol{H}^{-1}\boldsymbol{g}$ for correcting a trial parameter set. The Gauss–Newton method (GN) neglects, in the $\boldsymbol{H}$ matrix, second derivatives of the expressions for data items. When the expressions are linear in p_i (the linear case), both methods are identical and the result is obtained within a single refinement cycle. NR yields a very fast convergence, but usually needs a good starting estimate to converge.

With the non-negativity constraint, $p_i \geq 0$, such a set (the zero set) of p_i parameters should be found that, when the p_i's entering the zero set are fixed at zero (i.e., corresponding elements of $\boldsymbol{H}$ and $\boldsymbol{g}$ are discarded), the least-squares solution yields all remaining p_i's non-negative and with this solution, all g_i's corresponding to the p_i's from the zero set are positive. In the linear case, the zero set is unique and is sought by a non-negative least-squares (NNLS) algorithm[7].

For the error term E in Eq. (*8*), the $\boldsymbol{g}$ vector elements g_i are

$$g_i = \sum_k 2F_k v_{ik}(f_k - f_{k,\mathrm{exp}}) \ , \qquad (13)$$

where $f_k = f(t_k)$, $F_k = f_k^{1/2}$, $v_{1k} = 1$, and $v_{ik} = \exp(-t_k/\tau_i)$ for $i = 2, 3, \ldots, r$. The $\boldsymbol{H}$ matrix elements h_{ij} are

$$h_{ij} = \sum_k (6f_k - 2f_{k,\mathrm{exp}}) v_{ik} v_{jk} \qquad (14)$$

with NR; with GN the factor $6f_k - 2f_{k,\mathrm{exp}}$ should be replaced by $4f_k$. $\boldsymbol{H}$ is positively definite with GN, which may not be with NR if a data item is underestimated more than three times by the trial parameter set. Hessian matrix of the regularizor is denoted by $\boldsymbol{R}$ and the gradient vector by $\boldsymbol{r}$, so that corrections are calculated by $-(\boldsymbol{H} + \alpha\boldsymbol{R})^{-1}(\boldsymbol{g} + \alpha\boldsymbol{r})$. There is $r_{11} = B$ and $r_1 = Ba_1$. With NORDER = 0, $r_{ii} = 1$ and $r_i = a_i$, $i = 2, 3, \ldots, r$. With NORDER = 2, $r_{ii} = 6$, $r_{i,i+1} = r_{i+1,i} = -4$, $r_{i,i+2} = r_{i+2,i} = 1$, and $r_i = a_{i-2} - 4a_{i-1} + 6a_i - 4a_{i+1} + a_{i+2}$ where a_0, a_1, a_{r+1}, and a_{r+2} are set to zero; the indices of r_{ij} and r_i are restricted to $2 \leq i,j \leq r$. All r_{ij} elements not mentioned above are zero. Note that $\boldsymbol{r} = \boldsymbol{Ra}$.

To obtain a strting trial $\boldsymbol{a}$ vector for the reference solution, the problem is linearized using

$$g_i = \sum_k 2F_k v_{ik}\,(f_k^{1/2} + f_{k,\mathrm{exp}}^{1/2})\,(f_k^{1/2} - f_{k,\mathrm{exp}}^{1/2}) \approx 4\sum_k f_{k,\mathrm{exp}}\,(f_k^{1/2} - f_{k,\mathrm{exp}}^{1/2})v_{ik}\ ,$$

replacing f_k in h_{ij} by $f_{k,\mathrm{exp}}$, and removing a common factor of four. Then

$$g_i' = \sum_k f_{k,\mathrm{exp}}\,(f_k^{1/2} - f_{k,\mathrm{exp}}^{1/2})v_{ik} \quad \text{and} \quad h_{ij}' = \sum_k f_{k,\mathrm{exp}}\,v_{ik}v_{jk}$$

holds. This is equivalent to taking square roots of $f_{k,\mathrm{exp}}$ data in CONTIN. Using $f_{k,\mathrm{exp}}$ instead of its expectation value for the weight factor is good enough for this purpose when the factor exceeds the noise level. However, this may not be true in the opposite case. Therefore, when the factor is smaller than a threshold (arbitrarily chosen as 1/1024), it is replaced by the threshold. Following Provencher[11], $f_{k,\mathrm{exp}}^{1/2}$ is taken as $-(-f_{k,\mathrm{exp}})^{1/2}$ when $f_{k,\mathrm{exp}}$ is negative. With this approximation, a NNLS run is performed starting from zero trial $\boldsymbol{a}$ vector. The resulting $\boldsymbol{a}$ vector is iterated until convergence is reached: using it in Eqs (*13*) and (*14*), $\boldsymbol{g}$ and $\boldsymbol{H}$ are reset and another NNLS run is accomplished starting with this trial $\boldsymbol{a}$; during the NNLS run, $\boldsymbol{H}$ is kept fixed and $\boldsymbol{g}$ is being updated just with $\Delta\boldsymbol{g} = \boldsymbol{H}\Delta\boldsymbol{a}$ as in the linear case. The $\boldsymbol{a}$ convergence is assumed when all a_i's change less than $16^{-8} \times \sum_{i=1}^{r} a_i$; this test may become inappropriate when a very large a_2 value is obtained due to an improper τ grid. Surprisingly, no need for ensuring convergence was ever observed with proper $f_{k,\mathrm{exp}}$ data. Improper data (a data item quite off the autocorrelation curve) caused a failure sometimes. For the sake of simplicity and speed, I developed my own NNLS algorithm avoiding the singular value decomposition procedure.

NNLS Algorithm

Starting a NNLS iteration cycle with zero $\boldsymbol{a}$ vector at the very beginning, the maximum of g_i^2/h_{ii} is found and the respective a_i is set to $-g_i/h_{ii}$ to minimize expression (*8*) with respect to a_i. $\boldsymbol{g}$ is updated with $\Delta g_j = h_{ij}a_i$, s (the number of a_i's outside the zero set) is set to one, and h_{ii} is moved to h_{11}'. Starting an iteration cycle with a non-zero trial $\boldsymbol{a}$ (such that it minimizes expression (*8*) with respect to non-zero a_j's, s in number), the Gauss-eliminated $s \times s$ submatrix of $\boldsymbol{H}$ matrix is left stored in $\boldsymbol{H}'$ from the previous cycle; a single two-dimensional array is used for storing both $\boldsymbol{H}$ and $\boldsymbol{H}'$ as upper and lower triangles (with shifting the latter by one row). Now, a minimum g_i is found. When non-negative, the NNLS run terminates. When negative, h_{ii} is moved to $h_{s+1,\,s+1}'$; h_{ij}'s with a_j's outside the zero set are moved to $h_{u,s+1}'$ and Gauss elimination is extended to the $(s + 1)$-th column. When $h_{s+1,s+1}'$ is positive afterwards, a_i is accepted for leaving the zero set and s is incremented by one, otherwise NNLS rejects this a_i and repeats the minimum g_i search with this a_i neglected. When again a non-positive $h'_{s+1,s+1}$ results, the minimum search is repeated in turn with all former "bad" a_i's neglected; in the opposite case, a_i found is accepted instead of primary "bad" a_i. When every negative g_i yields a "bad" a_i, $\boldsymbol{H}$ matrix is reset with GN using the current trial $\boldsymbol{a}$ vector and another NNLS run is started with this trial $\boldsymbol{a}$. Since a GN $\boldsymbol{H}$ matrix is positively definite, its Gauss elimination may yield a non-positive diagonal element merely due to round-off errors. REPES terminates calculation at such an event and asks for a new data file; choosing a less dense τ-grid may be helpful in this case. After completing a NNLS run with GN, NR is restored for the next run.

After an a_i left the zero set, corrections Δa_j to the a_j's outside this set are calculated using Gauss-eliminated $\boldsymbol{H'}$; the correction to the a_i ($-g_i/h'_{s,s}$ – after incrementing s) is positive. When $-\Delta a_j < a_j$ always holds, a_j's are corrected, $\boldsymbol{g}$ updated with $\Delta\boldsymbol{g} = \boldsymbol{H}\Delta\boldsymbol{a}$, and a next NNLS iteration cycle started. If not, minimum positive $-a_j/\Delta a_j$ is found, corrections scaled with this minimum, $\boldsymbol{a}$ and $\boldsymbol{g}$ updated, the respective a_j re-entered to the zero set, a_k in the s-th position moved to the position (u-th) where a_j lies, and s decremented by one. Now, $\boldsymbol{H'}$ elements with both indices not less than u must be restored from $\boldsymbol{H}$ and Gauss elimination repeated starting from the u-th position. It is therefore desirable to have a_j as close to the s-th position as possible. In this respect, a_k's where $-a_j/\Delta a_j \leq -a_k/\Delta a_k < -2a_j/\Delta a_j$ are considered as candidates for re-entering the zero set in next steps and, when lying beyond the u-th position, are moved to last positions before $\boldsymbol{H'}$ restoring, starting from a_k with the lowest $-a_k/\Delta a_k$ value. A non-positive diagonal $\boldsymbol{H'}$ matrix element may now appear due to round-off errors only and causes program termination as above. After Gauss-eliminated $\boldsymbol{H'}$ is set, control returns to the calculation of corrections Δa_j at the beginning of this paragraph with the difference that no zero a_i appears now. When $\boldsymbol{H}$ is just reset or updated, Gauss elimination of $\boldsymbol{H'}$ starts from the first position and control again returns to the corrections calculation. A non-positive diagonal element appearing in this elimination causes resetting $\boldsymbol{H}$ by GN unless it is already set by GN in which case calculation terminates.

Every step within a NNLS run decreases E, so that NNLS can never return to any former iteration. Hence, cycling is impossible. Each scaling decreases the number of non-zero a_i's, so that only a finite number of scalings may be done within a NNLS iteration cycle. Now, a cycle starts with an $\boldsymbol{a}$ minimizing expression (*8*); such an $\boldsymbol{a}$ is unique for a given set of zero a_i's. Since the number of zero sets is finite (2^r), the number of NNLS iteration cycles within a run is finite. This proves the convergence of the NNLS run.

Regularization

The probabilities to reject and the number of degrees of freedom are defined by Eqs (*5*)–(*7*). The error term E is given by Eq. (*9*). p_0 equals the number s of the non-zero amplitudes in the reference solution. Note that p_0 is τ-grid-dependent since a PES peak is represented by a singlet when it falls close to a grid point and by a doublet otherwise. However, this seems to be of little importance in view of usually little sensitivity of the regularized solution to the actual Probability 1 to Reject value[12]. In Eq. (*7*), $\boldsymbol{H}$ of NR (Eq. (*14*)) is used rather than that of GN since it better represents E in the vicinity of the minimum of expression (*8*); $\boldsymbol{R}$ follows from Eq. (*11*) or (*12*).

For the α iteration, x is calculated from the selected Probability 1 to Reject first. The first approximation to x is obtained using Eqs 26.5.22 and 26.2.23 of ref.[5] and is re-iterated by the Newton method. At this, I_x is calculated by repetitive use of Eq. 26.5.16 of ref.[5], considering that $I_x(a,b)$ converges to zero when $a \to \infty$ and b and $x < 1$ are fixed; $\mathrm{d}I_x(a,b)/\mathrm{d}x = x^{a-1}(1-x)^{b-1}\Gamma(a+b)/(\Gamma(a)\Gamma(b))$. For $\Gamma(z)$, Eq. 6.1.41 of ref.[5] with terms up to z^{-7} is used when $z > 18$, Eq. 6.1.15 of ref.[5] is used to meet this with a lower z and in the repetitive use of Eq. 26.5.16. The x iteration converges unless the Probability 1 to Reject is either very small (below 10^{-7}) or very close to unity (above $1 - 10^{-5}$). Next, $E =$ $= E_0/(1-x)$ is calculated and α is iterated to yield this E value in minimizing expression (*8*). Hence, REPES minimizes the regularizor R among solutions with a given error term value E by the method of Lagrange multiplier (equal to $1/\alpha$).

For a linear case without the non-negativity constraint and when $\boldsymbol{g}$ is set with $\boldsymbol{a} = \boldsymbol{0}$, $\mathrm{d}E/\mathrm{d}(\alpha^2) = \tilde{\boldsymbol{g}}(\boldsymbol{H}+\alpha\boldsymbol{R})^{-1}\boldsymbol{R}(\boldsymbol{H}+\alpha\boldsymbol{R})^{-1}\boldsymbol{R}(\boldsymbol{H}+\alpha\boldsymbol{R})^{-1}\boldsymbol{g} = \tilde{\boldsymbol{r}}(\boldsymbol{H}+\alpha\boldsymbol{R})^{-1}\boldsymbol{r} = \sum_i r_i'^2/h_{ii}'$, where r_i' and h_{ii}' are elements of $\boldsymbol{r}$ and $\boldsymbol{H} + \alpha\boldsymbol{R}$ after Gauss elimination. So, $\Delta(\alpha^2) = (E - E_{\mathrm{calc}})/(\mathrm{d}E/\mathrm{d}(\alpha^2))$ and $\alpha + \Delta\alpha = (\alpha^2 + \Delta(\alpha^2))^{1/2}$ are used to iterate α by the Newton method. Using a linear transformation bringing both $\boldsymbol{H}$ and $\boldsymbol{R}$ to a diagonal form, $E = E_0 + \alpha^2 \sum_i c_i/(d_i + \alpha)^2$ with some positive c_i and d_i constants is found, which means that the Newton method underestimates $\Delta(\alpha^2)$ when positive and overestimates it when negative. To prevent oscillations, $\alpha + \Delta\alpha = 2\alpha^3/(2\alpha^2 - \Delta(\alpha^2))$ is used for the negative case.

With the non-negativity constraint, the current zero set (from the reference or previous regularized solution in the beginning) is used for the above α correction. Current $\boldsymbol{H} + \alpha\boldsymbol{R}$ and $\boldsymbol{g} + \alpha\boldsymbol{r}$ are updated with $\boldsymbol{R}\Delta\alpha$ and $\boldsymbol{r}\Delta\alpha$ (with a correction for a change in the B constant when changed; however, the B change is neglected in the first $\Delta\alpha$ correction by using the old $\mathrm{d}E/\mathrm{d}(\alpha^2)$ value) and used in place of $\boldsymbol{H}$ and $\boldsymbol{g}$ for a NNLS run started with the current $\boldsymbol{a}$ as the trial $\boldsymbol{a}$ vector. This is repeated until α converges. In a non-linear case, the solution for a particular α may be found by resetting $\boldsymbol{H} + \alpha\boldsymbol{R}$ using Eq. (*14*) and $\boldsymbol{g} + \alpha\boldsymbol{r}$ using Eq. (*13*) after a NNLS run and repeating NNLS until $\boldsymbol{a}$ converges. However, it is unpractical to calculate the exact solution in each α iteration step, since several steps are often necessary to get proper zero set and setting $\boldsymbol{H}$ is the most time-consuming step of REPES. Hence, α is iterated with $\boldsymbol{H}$ fixed and $\boldsymbol{g}$ updated just with $\boldsymbol{H}\Delta\boldsymbol{a}$ until two subsequent NNLS runs yield no change in the zero set and only then $\boldsymbol{H}$ and $\boldsymbol{g}$ are reset and the whole procedure repeated until $\boldsymbol{a}$ converges. Convergence problems were never observed in the α iteration. However, due to using the previous regularized solution to start a new one, the problems may arise when the latter solution is of a lower Probability 1 to Reject value than the former one.

For calculating the number of degrees of freedom for a regularized solution, the latter of expressions (*7*) is used. $\boldsymbol{G} = (\boldsymbol{H} + \alpha\boldsymbol{R})^{-1}$ is calculated starting from the Gauss-eliminated $\boldsymbol{H} + \alpha\boldsymbol{R}$. trace $((\boldsymbol{H} + \alpha\boldsymbol{R})^{-1}\boldsymbol{R})$ is calculated using

$$Bg_{11} + \sum_{i>1} g_{ii} \quad \text{for NORDER} = 0 \ \text{ and}$$

$$Bg_{11} + 6\sum_{i>1} g_{ii} - 8\sum_{i>1} g_{i,i+1} + 2\sum_{i>1} g_{i,i+2} \quad \text{for NORDER} = 2 \ ;$$

in both cases, the first term is absent when a_1 enters the zero set.

Probability 2 to Reject is calculated from Eq. (*6*) by repetitive using Eq. 26.5.16 of ref.[5]. When $p \geq p_0$, it is arbitrarily set to 2.

DISCUSSION

Running REPES with simulated data[8] yields results almost identical to those of CONTIN, with the only exception shown in Fig. 1 (reproduced Figs 2*d* and 3 of ref.[8]), where CONTIN and REPES results obtained from the data simulated for a Gaussian distribution of $1/\tau$ are compared. Apparently, the difference seen is caused by a very high density of simulated data; data are in fact averaged in the Laplace transform inversion and with high-density data, the averaging decreases the data variance below the square root taking bias[8] of CONTIN. As a much lower data density has been used since, no such difference has been found again. Hence, REPES results are expected to be essentially identical to those of CONTIN unless data are very dense. Nevertheless, REPES is considerably faster and needs considerably less computer memory than CONTIN. An influence of changing Probability 1 to Reject (*P*) is considered in ref.[12]. For a simulated broad Pearson *V* distribution of τ, even a very small *P* value (0.01) yields a proper smoothing and in an example of an experimental autocorrelation curve, the number of peaks is found unchanged within a broad *P* range (0.01–0.999). This indicates that the *P* value chosen may not be very critical for the result; for a thorough analysis, however, a calculation for a few different *P* values should be recommended. The *P* values of 0.125, 0.5, and 0.875 were often used in the Institute and differences in the results were revealed sometimes. A discussion of REPES may also be found in refs[9,13].

REPES was used in studying dilute and semidilute solutions of polymers[12,14–23], concentrated solutions of polymers[24–27], diffusion of a polymer in a matrix of another polymer[28–37], an influence of a surfactant on polymers[38–40], sol–gel transition[41], polymer blends[42], polyelectrolytes[43,44], zwitterions[45], crosslinked gels as compared to corresponding linear polymers[46–51], block copolymers and micelles[52–63], and an influence of an antibiotic on micelles of cholesterol[64]. It was also used in studying the effect of multiple light scattering[65–67]. Some of the above papers pursued both REPES and CONTIN with identical data. None of them found any essential difference between the results of these two methods.

The fact that REPES mostly gives essentially the same results as CONTIN means that nearly all CONTIN artifacts[8,68] persist in REPES. The Maximum Entropy (MAXENT)

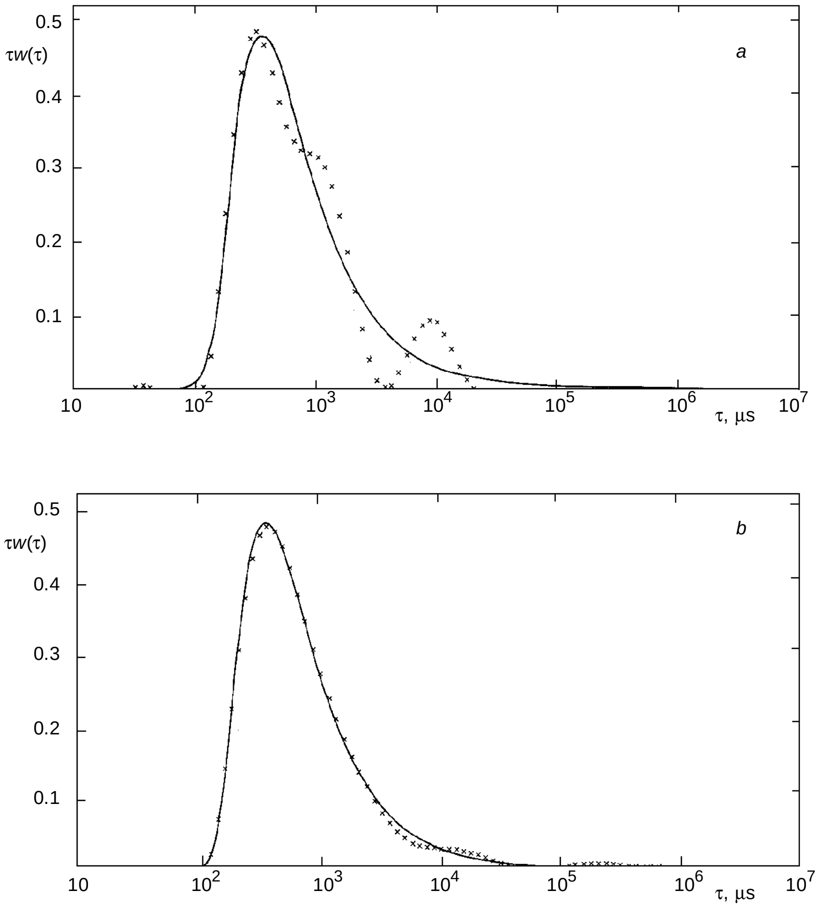

FIG. 1

Comparison of the relaxation time τ distributions obtained by CONTIN and REPES methods from data simulated with a Gaussian distribution of $1/\tau$ ($w(\tau) = 2\tau_0 \exp(-(\tau_0/\tau)^2)/(\pi^{1/2}\tau^2)$ with $\tau_0 = 250$ µs). NORDER = 2. Full lines: simulated distribution; × calculated results (zero values not shown). *a* CONTIN chosen solution, *b* REPES with Probability 1 to Reject of 0.5

method seems to be[68] even below CONTIN and hence below REPES. Methods restricting the number of maxima in the relaxation time distribution function instead of regularizing (the so called peak-constraining methods or methods with forced number of peaks) should be therefore preferred to regularizing methods. A flaw of these methods is unaesthetic histogram bin edges appearing in the results; however, the methods produce no artifact side peaks or shoulders[68,69]. Another alternative is to use a trial relaxation time distribution function with some parameters adjusted. The generalized exponential distribution[70,71] seems preferential among them and was used in considering the polydispersity index[72] and in a study of polystyrene dissolved in toluene[18].

The paper was supported by the Grant Agency of the Czech Republic (Grant No. 203/94/0817 (1994)).

It should be mentioned here that Dr Blahoslav Sedlacek, in honour of whom this issue appears, introduced the light scattering methods at the Institute of Macromolecular Chemistry. His effort in this field is gratefully appreciated.

REFERENCES

1. Cantor D. G., Evans J. W.: SIAM J. Appl. Math. *18*, 380 (1970).
2. Jakes J., Stepanek P.: Czech. J. Phys., B *40*, 972 (1990).
3. Jakes J.: Collect. Czech. Chem. Commun. *60*, 1815 (1995).
4. Provencher S. W.: Comput. Phys. Commun. *27*, 213 (1982).
5. Abramowitz M., Stegun I.: *Handbook of Mathematical Functions*. Dower, New York 1965.
6. Provencher S. W.: Makromol. Chem. *180*, 201 (1979).
7. Lawson C. L., Hanson R. J.: *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs 1974.
8. Jakes J.: Czech. J. Phys., B *38*, 1305 (1988).
9. Stepanek P. in: *Dynamic Light Scattering: The Method and Some Applications* (W. Brown, Ed.), Chap. 4. Clarendon Press, Oxford 1993.
10. Pope A. J.: Can. Surveyor *28*, 663 (1974).
11. Provencher S. W.: Comput. Phys. Commun. *27*, 229 (1982).
12. Nicolai T., Brown W., Johnsen R. M., Stepanek P.: Macromolecules *23*, 1165 (1990).
13. Johnsen R. M. in: *Laser Light Scattering in Biochemistry* (S. E. Harding, D. B. Sattelle and V. A. Bloomfield, Eds). Royal Society of Chemistry, Cambridge 1992.
14. Brown W., Nicolai T., Hvidt S., Stepanek P.: Macromolecules *23*, 357 (1990).
15. Pu Z., Brown W.: Macromolecules *23*, 1131 (1990).
16. Nicolai T., Brown W.: Macromolecules *23*, 3150 (1990).
17. Brown W., Nicolai T.: Colloid Polym. Sci. *268*, 977 (1990).
18. Brown W., Johnsen R. M., Konak C., Dvoranek L.: J. Chem. Phys. *95*, 8568 (1991).
19. Brown W., Pu Z.: Macromolecules *24*, 5151 (1991).
20. Brown W., Fundin J.: Macromolecules *24*, 5171 (1991).
21. Brown W., Stepanek P.: Macromolecules *26*, 6884 (1993).
22. Brown W., Nicolai T.: Macromolecules *27*, 2470 (1994).
23. Stepanek P., Johnsen R. M.: Collect. Czech. Chem. Commun. *60*, 1941 (1995).
24. Brown W., Stepanek P.: Macromolecules *24*, 5484 (1991).

25. Brown W., Johnsen R. M., Konak C., Dvoranek L.: J. Chem. Phys. *96*, 6274 (1992).
26. Brown W., Stepanek P.: Macromolecules *25*, 4359 (1992).
27. Konak C., Brown W.: J. Chem. Phys. *98*, 9014 (1993).
28. Brown W., Pu Z.: Macromolecules *22*, 3508 (1989).
29. Brown W., Pu Z.: Macromolecules *22*, 4031 (1989).
30. Brown W., Konak C., Johnsen R. M., Pu Z.: Macromolecules *23*, 901 (1990).
31. Nicolai T., Brown W., Hvidt S., Heller K.: Macromolecules *23*, 5088 (1990).
32. Brown W., Pu Z.: Macromolecules *23*, 5097 (1990).
33. Brown W., Pu Z.: Polymer *31*, 772 (1990).
34. Konak C., Tuzar Z., Jakes J.: Polymer *31*, 1866 (1990).
35. Brown W., Pu Z.: Macromolecules *24*, 1820 (1991).
36. Pajevic S., Bansil R., Konak C.: J. Non-Cryst. Solids *131*, 630 (1991).
37. Pajevic S., Bansil R., Konak C.: Macromolecules *26*, 305 (1993).
38. Brown W., Fundin J., Miguel M. da G.: Macromolecules *25*, 7192 (1992).
39. van Stam J., Brown W., Fundin J., Almgren M., Lindblad C.: ACS Symp. Ser. *532*, 194 (1993).
40. Fundin J., Brown W.: Macromolecules *27*, 5024 (1994).
41. Fang L., Brown W., Konak C.: Macromolecules *24*, 6839 (1991).
42. Stepanek P., Lodge T. P., Kedrowski C., Bates F. S.: J. Chem. Phys. *94*, 8289 (1991).
43. Sedlak M., Konak C., Labsky J.: Polymer *32*, 1688 (1991).
44. Sedlak M.: J. Chem. Phys. *101*, 10140 (1994).
45. Konak C., Rathi R. C., Kopeckova P., Kopecek J.: Macromolecules *27*, 1992 (1994).
46. Fang L., Brown W.: Macromolecules *23*, 3284 (1990).
47. Fang L., Brown W., Konak C.: Polymer *31*, 1960 (1990).
48. Konak C., Jakes J., Brown W., Fang L.: Polymer *32*, 1077 (1991).
49. Brown W., Fang L., Stepanek P.: Macromolecules *24*, 3201 (1991).
50. Fang L., Brown W., Hvidt S.: Macromolecules *25*, 3137 (1992).
51. Fang L., Brown W.: Macromolecules *25*, 6897 (1992).
52. Tuzar Z., Konak C., Stepanek P., Plestil J., Kratochvil P., Prochazka K.: Polymer *31*, 2118 (1990).
53. Balsara N. P., Stepanek P., Lodge T. P., Tirrell M.: Macromolecules *24*, 6227 (1991).
54. Konak C., Podesva J.: Macromolecules *24*, 6502 (1991).
55. Brown W., Schillen K., Almgren M., Hvidt S., Bahadur P.: J. Phys. Chem. *95*, 1850 (1991).
56. Brown W., Schillen K., Hvidt S.: J. Phys. Chem. *96*, 6038 (1992).
57. Almgren M., Bahadur P., Jansson M., Li P., Brown W., Bahadur A.: J. Colloid Interface Sci. *151*, 157 (1992).
58. Bahadur P., Li P., Almgren M., Brown W.: Langmuir *8*, 1903 (1992).
59. Castanho M. A. R. B., Brown W., Prieto M. J. E.: Biophys. J. *63*, 1455 (1992).
60. Schillen K., Brown W., Konak Ç.: Macromolecules *26*, 3611 (1993).
61. Mortensen K., Brown W.: Macromolecules *26*, 4128 (1993).
62. Schillen K., Brown W., Johnsen R. M.: Macromolecules *27*, 4825 (1994).
63. Konak C., Stepanek P., Vlcek P., Johnsen R. M.: Macromolecules *28*, 2852 (1995).
64. Castanho M. A. R. B., Brown W., Prieto M. J. E.: Biopolymers *34*, 447 (1994).
65. Konak C., Jakes J., Petras F., Karska M., Perina J.: Collect. Czech. Chem. Commun. *55*, 1022 (1990).
66. Konak C., Jakes J., Stepanek P., Petras F., Karska M., Krepelka J., Perina J.: Appl. Opt. *30*, 4865 (1991).
67. Stepanek P.: J. Chem. Phys. *99*, 6384 (1993).

68. Provencher S. W. in: *Laser Light Scattering in Biochemistry* (S. E. Harding, D. B. Sattelle and V. A. Bloomfield, Eds). Royal Society of Chemistry, Cambridge 1992.
69. Jakes J.: Czech. J. Phys., B *43*, 1 (1993).
70. Kubin M.: Collect. Czech. Chem. Commun. *32*, 1505 (1967).
71. Jakes J., Saudek V.: Makromol. Chem. *187*, 2223 (1986).
72. Jakes J.: Collect. Czech. Chem. Commun. *56*, 1642 (1991).