Computer Physics Communications 74 (1993) 335–348
North-Holland

Computer Physics
Communications

Programs for signal recovery from noisy data using the maximum likelihood principle

v I. General description

V.I. Gelfgat, E.L. Kosarev and E.R. Podolyak

P.L. Kapitza Institute for Physical Problems, Moscow 117334, Russian Federation

Received 4 August 1992

A program package for nonnegative signal recovery from noisy experimental data distorted by the measuring device is presented. Noise may have a Gaussian, binomial or Poissonian distribution at each experimental point. Three examples are given which demonstrate the usage of the package: (a) reduction to an arbitrary instrumental function, which depends only on the difference of its argument (this is the typical spectroscopic problem); (b) the expansion of an exponentially decaying curve which has a continuous spectrum of decrements (this problem has applications in time dependent fluorescence, nuclear physics etc.); and (c) the ultrasoft X-ray spectrum recovery from absorption measurements. These programs are based on the maximum likelihood principle which allows the maximum value of resolution enhancement according to Shannon's information theory to be achieved.

1. Introduction

Because of the presence of noise and the finite resolving power of the measuring device any measurement of a physical quantity cannot provide us with the required information. Both problems: the noise suppression and reduction to the finite resolution are closely connected and should be solved simultaneously. In the simplest form the problem of signal recovery may be presented as an integral equation of the first kind

$$\int_a^b K(x, y)\, G_0(y)\, \mathrm{d}y = F_0(x), \quad c \le x \le d, \quad (1)$$

where $G_0(y)$ is the unknown function (the signal); $F_0(x)$ is the result of distortion (or transformation) by the measuring device which is described in terms of an instrumental function $K(x, y)$. The distorted function $F_0(x)$ is then spoiled by noise, giving the real result of measurement, $F(x)$. So only the random function $F(x)$ is available for signal reconstruction. We assume that the values of the random function $F(x)$ are independent at different x and they do not contain systematical errors, i.e

$$\overline{F(x)} = F_0(x) \quad (2)$$

and

$$\overline{\Delta F(x_i)\, \Delta F(x_j)} = \delta(x_i - x_j)\, \sigma^2(x_i), \quad (3)$$

where

$$\Delta F(x_i) = F(x_i) - F_0(x_i)$$

represents the noise at point x_i. The solid line above variables in eqs. (2) and (3) denotes the ensemble averaging and $\sigma^2(x_i)$ represents the dispersion (variance) of the noise at the point x_i.

The goal of the reconstruction procedure is to determine the estimate $G(y)$ of the unknown function $G_0(y)$ as accurate as possible from the random function $F(x)$.

Correspondence to: E.L. Kosarev, P.L. Kapitza Institute for Physical Problems, Moscow 117334, Russian Federation.
E-mail: kosarev@~~magnit.msk.su~~.
kapitza.ras.ru

Problems of this kind which often are called inverse problems are common in most fields of experimental physics so there are several reviews on the subject [1–5] and also papers containing the description and source code of programs to solve such problems [6–14].

We will only discuss in this paper the special case of seeking nonnegative signals $G_0 \geq 0$. Although this case is not general, it has a broad field of applications: e.g. all problems for energy spectra recovery fall into this category. A special monograph [6] is devoted to this case.

The principle feature of this work is that we are using the maximum likelihood (ML) method [15] to get the solution of eq. (1) with random right-hand side. It has been shown in refs. [16,17] that this method provides the ultimate resolution enhancement as compared with any other non-parametric methods.

The ML method requires not only knowledge of the kernel $K(x, y)$ of eq. (1) but also the statistical description of the noise properties, i.e. the noise distribution function. The kernel and the noise distribution function may be obtained either from some theoretical considerations or, which is better, from especially designed direct measurements. Since the reconstruction procedure essentially depends on the statistical properties of the noise we shall discuss separately the cases when the noise has a binomial (Poissonian) distribution at each experimental point and when the distribution function has the Gaussian form.

2. ML algorithm for the binomial (Poissonian) distribution

In this article we consider only those cases when the experimental data are given as a set of values $F(x_i)$ at some points $\{x_i\}$, $i = 1, 2, \dots, n$. In this section we use the notations $N_i = F(x_i)$ in order to indicate that the experimental data are positive integers. Introducing the set of independent variables $\{y_j\}$, $j = 1, 2, \dots, m$ at which we would like to calculate the values $G_j = G(y_j)$ of the unknown function, we can use any numerical quadrature formula to calculate the integral (1).

So we obtain a linear system of algebraic equations which is convenient for computer implementation,

$$\sum_{j=1}^{m} P_{ij} G_j = N_i, \quad i = 1, 2, \dots, n, \tag{4}$$

where

$$P_{ij} = K(x_i, y_j)(y_{j+1} - y_j).$$

Let us emphasize that the number of equations n in (4) may be greater or less than the number of unknowns n. In most cases the system (4) has no solution at all since its right-hand side is a random vector and the system matrix P_{ij} is practically degenerate.

In the case of binomial (Poissonian) distribution the right-hand side may acquire only nonnegative integer values (usually they are the numbers of pulses measured by some counter) which are mutually independent and can be treated as having a polynomial distribution.

Besides the requirement of nonnegativeness of the experimental data $\{N_i\}$ there is another requirement of nonnegativeness of the kernel P_{ik}. In this case the normalized quantities

$$s_i = \sum_{k=1}^{m} p_{ik} g_k, \quad g_k = \frac{G_k}{\sum_{l=1}^{m} G_l} \geq 0 \tag{5}$$

may be treated as probabilities and

$$p_{ik} = \frac{P_{ik}}{\sum_{j=1}^{n} P_{jk}} \tag{6}$$

as a stochastic matrix $p_{ik} \geq 0$, $\sum_{i=1}^{n} p_{ik} = 1$.

The joint probability of the measured set of values $\{N_i\}$ is then given by

$$\mathscr{P} = \frac{N!}{\prod_{i=1}^{n} N_i!} \prod_{i=1}^{n} s_i^{N_i}, \tag{7}$$

where $N = \sum_{i=1}^{n} N_i$. According to the maximum likelihood principle we call the vector $\{G_k\}$ which brings the probability (7) to its maximum, the solution to the system (4). The quantity (7) is known to be the likelihood function.

The maximum of the likelihood function always exists since it is continuous and the expression

$$L = \log \mathscr{P} = \text{constant} + \sum_{i=1}^{n} N_i \log s_i$$
$$= \text{constant} + N \sum_{i=1}^{n} f_i \log s_i, \qquad (8)$$

where $f_i = N_i/N$, has an upper bound since

$$\sum_i f_i \log s_i - \sum_i f_i \log f_i = \sum_i f_i \log(s_i/f_i) \leq 0, \qquad (9)$$

which follows from the inequality of the information theory (see ref. [15], No. 1e.6.1).

To find the maximum of the likelihood function L in the space of unknown vectors $\{G_k\}$ we use the same iterative procedure as described in ref. [18], which is the modified version of the procedure by M.Z. Tarasko [19]. In this method there are no other assumptions about the form of the unknown function $G(x)$ than $G(x) \geq 0$. No information is required on the initial guess, so we generally use the initial guess of the form $G(x) =$ constant.

This iterative procedure can be written in the form

$$\delta g_k^{(t)} = g_k^{(t)} \left[\sum_{i=1}^{n} p_{ik} \left(\frac{f_i}{\sum_{j=1}^{m} p_{ij} g_j^{(t)}} - 1 \right) \right], \qquad (10)$$
$$g_k^{(t+1)} = g_k^{(t)} + h \delta g_k^{(t)}.$$

Here $t = 1, 2, \ldots,$ is the iteration number, h is the step length in the space of unknown vectors $\{G_k\}$. When $h = 1$ iterative formula (10) coincides with that by M.Z. Tarasko.

The expression in the square brackets coincides with the kth component of the Lagrange function gradient

$$\Phi = L + \lambda \left(\sum_{k=1}^{m} g_k - 1 \right),$$
$$\frac{\partial \Phi}{\partial g_k} = N \sum_{i=1}^{n} p_{ik} \left(\frac{f_i}{\sum_{j=1}^{m} p_{ij} g_j} - 1 \right), \qquad (11)$$

and the Lagrange multiplier λ is equal to $\lambda = -N$. Hence the stationarity of iterations (10) is equivalent to the zero gradient condition on the Likelihood function providing $\sum_{k=1}^{m} g_k = 1$. Thus we have shown that the iterative process (10) leads to the maximum of the likelihood function, i.e. it brings the solution of eq. (1) when the experimental data have a polynomial distribution.

Note, that the formula (10) is essentially nonlinear with respect to the unknown vector $\{G_k\}$. The origin of the nonlinearity is the requirement of nonnegativeness of the components of vector $\{G_k\}$. This requirement is provided by the factor g_k in the definition of the direction of search (10) and by the appropriate choice of the value of the step satisfying the condition

$$h \leq \min_{\{k;\, \delta g_k < 0\}} (-g_k/\delta g_k). \qquad (12)$$

This nonlinearity of the ML method is its most important property, and it is the key to the ultimate resolution achievement as compared with any linear methods of signal recovery. It is shown in refs. [16,17] that this improvement of resolution by using (10) (or (27) for Gaussian noise, see below) reaches the theoretical limit derived from Shannon's theorem.

The length of the step h is defined from the 1D maximization of the likelihood function which is quadratically extrapolated along the direction of search

$$h_{\text{opt.}} = - \left. \frac{\mathrm{d}L}{\mathrm{d}h} \right|_{h=0} \Big/ \left. \frac{\mathrm{d}^2 L}{\mathrm{d}h^2} \right|_{h=0}. \qquad (13)$$

The values of derivatives necessary for extrapolation are obtained from the accurate relations

$$\left. \frac{\mathrm{d}L}{\mathrm{d}h} \right|_{h=0} = N \sum_{k=1}^{m} \frac{\left(\delta g_k^{(t)}\right)^2}{g_k^{(t)}} > 0 \qquad (14)$$

and

$$\left. \frac{\mathrm{d}^2 L}{\mathrm{d}h^2} \right|_{h=0} = -N \sum_{i=1}^{n} \frac{f_i \left(\delta s_i^{(t)}\right)^2}{\left(s_i^{(t)}\right)^2} < 0, \qquad (15)$$

where

$$\delta s_i^{(t)} = \sum_{k=1}^{m} p_{ik}\, \delta g_k^{(t)}. \qquad (16)$$

The value of the step obtained from (12) and (13) is of the order of 10^3–10^4, which provides a much faster convergence rate compared to the original procedure by M.Z. Tarasko.

The norm of the vector $\| g_k \| = \Sigma_k g_k$ is conserved automatically during iterations at any value of the step h which directly follows from (10). The finite accuracy of computer calculations leads to unstable behavior of the algorithm so the stability is maintained artificially by additional normalizing of the vector $\{g_k^{(t)}\}$ at each iteration.

The iterations should be continued until the value

$$\chi^2 = \sum_{i=1}^{n} \frac{(\delta N_i)^2}{Ns_i}, \tag{17}$$

where $\delta N_i = N(s_i - f_i)$ reaches a value less than $\chi^2_{n-1}(P_1)$, where P_1 is the level of the χ^2 criterion.

In this case the deviation $\{\delta N_i\}$ should correspond to the experimental accuracy (3). This is an important principle in the solution of inverse problems.

In real physical applications it is necessary to know the accuracy of the solution. We use the method of statistical simulation where the reconstruction process is repeated several times for M different sets of random experimental data. This simulation gives the solution in the form of the mean value

$$\overline{G(y)} = \frac{1}{M} \sum_{j=1}^{M} G^{[j]}(y) \tag{18}$$

and the variance

$$\operatorname{var}[G(y)] = \frac{1}{M-1} \sum_{j=1}^{M} \left[(G^{[j]}(y) - \overline{G(y)}\right]^2, \tag{19}$$

which are calculated over M different sets of reconstructed data $G^{[j]}(y)$, $j = 1, 2, \ldots, M$. The number of simulations M usually is about 10–20.

3. ML algorithm for the Gaussian distribution

In the case of Gaussian distribution of noise the likelihood function may be written as

$$L = -\frac{1}{2} \sum_{i=1}^{n} \frac{(F_i - S_i)^2}{D_i} + \text{constant}, \tag{20}$$

where the values of D_i are equal to the noise dispersions at the ith experiment point

$$D_i = \sigma_i^2, \quad i = 1, 2, \ldots, n, \tag{21}$$

and the values of S_i are defined by the formula

$$S_i = \sum_{k=1}^{m} P_{ik} G_k, \quad i = 1, 2, \ldots, n. \tag{22}$$

Let us denote the sum of G_k by G

$$G = \sum_{k=1}^{m} G_k, \tag{23}$$

so the likelihood function may be defined in the extended $(n+1)$-dimensional space $\{g_k\}$, G. The extremum conditions in this space are as follows:

$$\frac{\partial L}{\partial g_k} = G \sum_{i=1}^{n} \frac{F_i - S_i}{D_i} P_{ik} = 0, \quad k = 1, 2, \ldots, m \tag{24}$$

and

$$\frac{\partial L}{\partial G} = G^{-1} \sum_{i=1}^{n} \frac{F_i - S_i}{D_i} S_i = 0. \tag{25}$$

From (25) one may obtain

$$G = \frac{\sum_{i=1}^{n} (F_i s_i)/D_i}{\sum_{i=1}^{n} s_i^2/D_i},$$

$$\text{where } s_i = \sum_{k=1}^{m} P_{ik} g_k, \quad s_i = S_i/G. \tag{26}$$

So for any given vector $\{g_k\}$ there is only one value of G maximizing the likelihood function.

The maximum of the function L will be sought for the case of Gaussian noise by means of an iterative procedure which is similar to that for

Poissonian noise. The direction of search $\delta g_k^{(t)}$ is defined by

$$g_k^{(t+1)} = g_k^{(t)} + h\ \delta g_k^{(t)},$$

$$\text{where } \delta g_k^{(t)} = g_k^{(t)} \sum_{i=1}^{n} \frac{F_i - G \sum_j P_{ij} g_j^{(t)}}{D_i}. \tag{27}$$

The calculation of the value of G by (26) on each iteration keeps the norm of the vector $\{g_k\}$ constant throughout the iteration process since

$$\sum_{k=1}^{m} \delta g_k^{(t)} = \sum_{i=1}^{n} \frac{F_i - S_i^{(t)}}{D_i} \left(\sum_{k=1}^{m} P_{ik} g_k^{(t)} \right)$$
$$= \sum_{i=1}^{n} \frac{F_i - S_i^{(t)}}{D_i} s_i^{(t)} = 0, \tag{28}$$

which follows from (25). An additional normalization is required at each iteration step in order to increase the stability against computational round-off errors.

The value of the optimal step h in the direction of search $\{\delta g_k\}$ is defined from the equation

$$\frac{\mathrm{d}}{\mathrm{d}h} L(G(h), \{g_k + h\ \delta g_k\}) = 0, \tag{29}$$

where

$$G(h) = \frac{\sum_{i=1}^{n} (F_i s_i)/D_i + h \sum_{i=1}^{n} (F_i\ \delta s_i)/D_i}{\sum_{i=1}^{n} (s_i + h\ \delta s_i)^2/D_i} \tag{30}$$

and

$$\delta s_i = \sum_{k=1}^{m} P_{ik}\ \delta g_k. \tag{31}$$

For the sake of simplicity we may introduce the scalar product notations

$$Fs = \sum_{i=1}^{n} \frac{F_i s_i}{D_i}, \quad F\delta = \sum_{i=1}^{n} \frac{F_i\ \delta s_i}{D_i},$$
$$Ss = \sum_{i=1}^{n} \frac{s_i^2}{D_i}, \quad S\delta = \sum_{i=1}^{n} \frac{s_i\ \delta s_i}{D_i}, \tag{32}$$
$$D\delta = \sum_{i=1}^{n} \frac{\delta s_i^2}{D_i}$$

and rewrite (30) in the form

$$G(h) = \frac{Fs + h\ F\delta}{Ss + 2h\ S\delta + h^2\ D\delta}, \tag{33}$$

and an expression for the optimal value of the step

$$h = \frac{F\delta - G \cdot S\delta}{G \cdot D\delta - F\delta \cdot S\delta/Ss}. \tag{34}$$

The number of scalar products used in the algorithm may be decreased by replacing (34) with the equivalent expression

$$h = \frac{A}{G\left[D\delta - (S\delta)^2/Ss\right] - A(S\delta)/(Ss)}, \tag{35}$$

where

$$A = \sum_{k=1}^{m} \frac{\delta g_k^2}{g_k}.$$

Since the change in the likelihood function after one iteration step

$$L(h) - L(0) = \tfrac{1}{2} G(h) \sum_{k=1}^{m} \frac{(\delta g_k)^2}{g_k} h \geq 0 \tag{36}$$

is always nonnegative and proportional to the length squared of the likelihood function gradient, the iterational process (27) with the step h defined by (34) and subject to (12) converges to the maximum of the likelihood function (20).

The iteration formula (27) and also (10) in the case of Poissonian noise is essentially nonlinear firstly because the unknown vector $\{g_k\}$ is used here as a factor in (27) and secondly through the dependence of S_i on g_k. The important thing, however, is the existence of nonlinearity in the ML algorithm which provides superresolution compared with the linear methods.

The iteration procedure (27) gives the solution to the reconstruction problem in the case of a Gaussian noise distribution for any instrumental function which is not restricted by the requirement of nonnegativeness which is essential in the case of polynomial distribution.

4. The convolution integral equation

The superresolution achievement may be illustrated by the problem of spectrum reduction to the instrumental function of the measuring device, when this function depends only on the difference of its arguments,

$$K(x, y) = K(x-y). \tag{37}$$

The integral equation (1) becomes the convolution integral

$$\int_a^b K(x-y)\, G_0(y)\, \mathrm{d}y = F_0(x), \quad c \le x \le d, \tag{38}$$

This problem has a lot of applications especially in spectroscopy. There are two special monographs [6,7] and computer source code [6–8,11–14] concerning this problem. None of those programs is able to achieve the superresolution available with the program described below.

In the case of an instrumental function depending only on the difference of its arguments (37) all the sums of the form

$$s_i = \sum_{j=1}^{m} p_{ij} x_j, \quad i = 1, 2, \ldots, n, \tag{39}$$

$$y_j = \sum_{i=1}^{n} p_{ij} f_i, \quad j = 1, 2, \ldots, m, \tag{40}$$

in the iterative formulae (10) or (27) are replaced by the discrete convolutions

$$s_i = \sum_{j=1}^{m} p_{ij'} x_j, \quad \text{where } j' = i - j + 1 \ (\mathrm{mod}\ m), \tag{41}$$

$$y_j = \sum_{i=1}^{n} p_{i'j} f_i, \quad \text{where } i' = i + j - 1 \ (\mathrm{mod}\ n), \tag{42}$$

and the latter in turn are calculated via FFT routines. To perform this replacement one should pad the vectors involved in (39) and (40) by additional zeroes. The matrix representing the instrumental function contains $n+m-1$ distinguished values so the convolution should have the length of $n+m-1$.

In the cases when the right-hand side data become zeros at the ends of the interval (c, d), the convolutions may have a reduced length.

To prove the resolution, corresponding to the spatial frequencies which are absent in the experimental data, i.e. the superresolution not available for any of the linear methods of signal reconstruction, we present examples of the reconstruction for two types of instrumental functions: Gaussian $k_1(s) = \exp(-s^2)$ which has an infinite Fourier spectrum and $k_2(s) = ((\sin s)/s)^2$ with a finite Fourier spectrum.

In this formula we use the dimensionless parameter $s = (x-y)/D$, where the parameter D, defining the width of the instrumental function, is chosen so that the corresponding instrumental functions become zero at the ends of the interval (a, b). In this case eq. (38) may be treated as having infinite integration limits. All the functions involved in the computations are given at the discrete mesh $x_i = i$ for $i = 1, 2, \ldots, n$ and the integral is replaced by a finite sum of the form (39). The total number of points is equal to $n = 512$.

Following the papers [16,17] we choose the true solution $G_0(y)$ as a sum of two or three narrow lines with Gaussian profile

$$G_0(y) = \sum_{k=1}^{M} A_k \exp(-u_k)^2, \quad M = 2 \text{ or } 3,$$

where $u_k = (y - y_k)/D_1$, $D_1 = D/40$ and then calculate the integral (38). Next we add the noise $N(x_i)$ with Gaussian distribution and dispersion σ^2 to the value of integral $F_0(x_i)$ at each point x_i.

The value of dispersion σ^2 is defined by the ratio of signal energy $P_s = \int_{-\infty}^{\infty} F_0^2(x)\, \mathrm{d}x$ to the energy of noise $P_n = n\sigma^2$ using the formula

$$\sigma^2 = \frac{1}{n} \frac{P_s}{10^{\mathrm{dB}/10}},$$

where the variable dB is equal to the signal-to-noise ratio expressed in decibels dB = 10 $\log(P_s/P_n)$.

The input data for the ML algorithm are the values $F(x_i) = F_0(x_i) + N(x_i)$ for $i = 1, 2, \ldots, n$, the value of σ^2 and the instrumental function

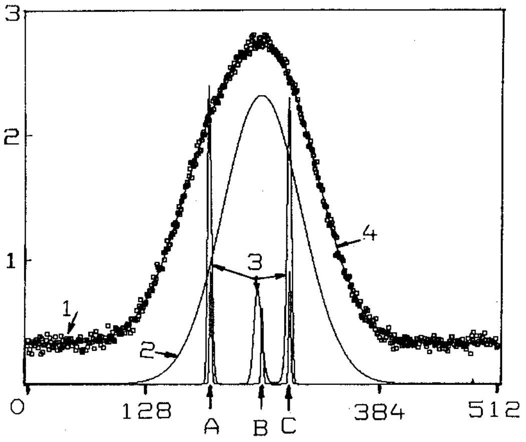

Fig. 1. Example of a three lines reconstruction by the Dconv program for a Gaussian instrumental function with width $D = 60$ and number of data ponts $N = 512$. 1 – input data with the signal-to-noise ratio 30 dB; 2 – instrumental function; 3 – result of recovery after 450 iterations; 4 – convolution of recovery result 3 with the instrumental function 2. The obtained χ^2-test value $\chi^2/N = 0.934$. Arrows A, B and C indicate the true locations of the recovered lines. The Y-axis scale is linear, but for clarity curves 1 and 4 been shifted upwards vertically.

$K(x-y)$. No more information is necessary for the algorithm to work.

Figure 1 shows the result of three lines reconstruction with the amplitude ratio 1.5 : 1 : 1.5 for a Gaussian instrumental function with the width $D = 60$ and signal-to-noise ratio 30 dB. It may be seen from this figure that all the lines are well resolved with correct positions and amplitude ratio. The qualitative characteristics of corresponding deviations are given in table 1.

Note, that no information about the linear structure of the unknown signal is supplied to the algorithm. The program is looking for the function $G_0(y)$ or, which is more correct, for the

Table 1
The actual deviation of line positions and the area under each of three peaks in the example presented at fig. 1.

	Relative deviation of the reconstruction (%)		
Position	−0.41	−1.74	−0.27
Area under the peak	−3.41	−7.55	+8.44
Number of line	1	2	3

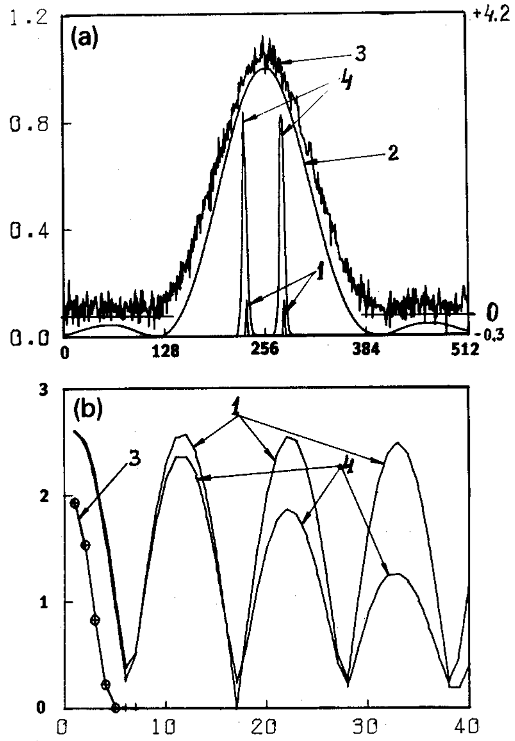

Fig. 2. Restoration of two lines, convolved with instrumental function $((\sin x)/x)^2$ for $D = 45$ and data points $N = 512$. (a) Original lines – 1; instrumental function – 2; input data with SNR = 22.5 dB – 3; the result of restoration – 4. (b) Fourier spectra in linear scale: original lines – 1; input data – 3; result of restoration – 4. The obtained χ^2-test value $\chi^2/N = 0.96057$.

vector $\{G_k\}$ with m components which corresponds to the "experimental" data best of all.

If the approximate solution is found to be like that presented in fig. 1, it is meaningful to start *a parametric* algorithm to find the positions and amplitudes of the three lines. In this case the accuracy of the estimation of these 6 parameters (or 5 if the normalized values are searched) may be much better than Shannon's limit (see refs. [16,17]) and will be limited only by the Cramer–Rao inequality [15].

Figure 2 shows the result of the reconstruction for the instrumental function $((\sin s)/s)^2$, which has a limited Fourier spectrum. The input data in this example do not contain spatial frequencies higher than the cutoff frequency

$$|\omega| > \omega_0 = 2/D. \tag{43}$$

Nevertheless the data shown in this figure demonstrate the ability of ML algorithm to reconstruct the absent frequencies. This feature is inherent only to nonlinear algorithms and is principally unavailable for any linear method since the linear methods are capable only to modify the harmonics which are present in experimental data but not to create the absent ones.

The programs described in this section to solve the convolution integral equations have been used in refs. [16,17] along with the experimental study of superresolution with respect to the signal-to-noise ratio.

The results demonstrated in figs. 1 and 2 are obtained using the iterative procedure (27) designed for Gaussian noise and arbitrary instrumental function. Similar results are obtained using formula (10) for a positive instrumental function and bynomial or Poissonian noise.

5. Superresolution exponential analysis via the maximum likelihood method

The exponential analysis problem is to recover the spectral function $g(\lambda)$ from the function $f(t)$, which is the Laplace transform of the $g(\lambda)$,

$$f(t) = \int_0^\infty g(\lambda) \exp(-\lambda t)\, d\lambda. \tag{44}$$

We always have the experimental data $F(t)$ together with the noise $n(t)$,

$$F(t) = f(t) + n(t), \tag{45}$$

so the input data is $F(t)$, and not only $f(t)$.

Lancos [29] has pointed out the inherent difficulties in the even more simple problem of exponential analysis with a discrete numbers of exponentials,

$$f(t) = \sum_{j=1}^{M} A_j \exp(-\lambda_j t), \tag{46}$$

where decrements λ_j, amplitudes A_j and the numbers of exponents M are unknown. These difficulties are even more serious for a continuous spectrum of $g(\lambda)$.

The standard formula for the Laplace transform inverse is:

$$g(\lambda) = \frac{1}{2\pi \mathrm{i}} \int_{c-\mathrm{i}\infty}^{c+\mathrm{i}\infty} f(t)\, \mathrm{e}^{\lambda t}\, dt$$

(see, for example, Titchmarsh [30]), and it is not applicable in our case since knowledge of $f(t)$ for complex values of t is required for this formula, whereas $f(t)$ is known only for real positive values of t, rather than for complex values of t.

From the variety of different methods for the Laplace transform inversion it is worth to note the method of Pike et al. [31–34], which these authors have applied for example to the light scattering polydispersion analysis of molecular diffusion.

In this method the right-hand side of eq. (44) and the solution $g(\lambda)$ are expanded over the eigenfunctions of the Laplace transform (44). This method is proved to be effective and it can resolve in principle two or three exponents having a moderate number of experimental data.

These authors have discovered that the accuracy of recovery of an unknown function $g(\lambda)$ would be better if the experimental data points t_i, $i = 1, 2, \ldots, n$ were placed uniformly in the logarithmic scale $x = \log t$, rather than in the original scale t. While this reasoning agrees with common sense, the complicated procedure for the generation of the eigenfunctions does not explain the origin of the efficiency increase obtained by using the logarithmic scale.

Another method that certainly should be kept in mind is known in physics literature as Gardner's method after the first author of the paper [35] in spite of the face that this method was earlier described in Titchmarsh's book [30]. There are new publications [36–40] using Gardner's method.

Here is a brief outline of this method. We start with the change of variables

$$x = \ln t, \quad y = \ln(1/\lambda) = -\ln \lambda, \tag{47}$$

which means the transformations to a logarithmic scale on both the t axis and the λ axis. One obtains, in the new variables,

$$f(\mathrm{e}^x) = \int_{-\infty}^{\infty} g(\mathrm{e}^{-y}) \exp(-\mathrm{e}^{x-y})\, \mathrm{e}^{-y}\, dy. \tag{48}$$

Multiplying both sides of (48) by e^x we obtain

$$tf(t) = e^x f(e^x) = \int_{-\infty}^{\infty} g(e^{-y}) \exp[-e^{(x-y)} + (x-y)]\, dy. \tag{49}$$

Equation (49) is obviously the convolution type of integral equation for an unknown function $g(e^{-y})$ with kernel (or apparatus function)

$$K(x-y) = \exp[-e^{(x-y)} + (x-y)]. \tag{50}$$

The convolution integral equation (49) is solved in a standard way – using the Fourier transform to both sides of the equation. As a result we have

$$\mathscr{F}(\mu) = G(\mu)\ \mathscr{K}(\mu). \tag{51}$$

Here $\mathscr{F}$, G and $\mathscr{K}$ are denoting the Fourier transforms of the given and sought for functions

$$\mathscr{F}(\mu) = \int_{-\infty}^{\infty} e^x f(e^x)\, e^{i\mu x}\, dx,$$

$$G(\mu) = \int_{-\infty}^{\infty} g(e^{-y})\, e^{i\mu y}\, dy, \tag{52}$$

and also the kernel

$$\mathscr{K}(\mu) = \int_{-\infty}^{\infty} K(s)\, e^{i\mu s}\, ds = \int_{-\infty}^{\infty} \exp(-e^s + s)\, e^{i\mu s}\, ds = \Gamma(1 + i\mu). \tag{53}$$

$\Gamma(1 + i\mu)$ in this formula is the Euler gamma-function of complex argument $1 + i\mu$.

From (51) we have

$$g(e^{-y}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{\mathscr{F}(\mu)}{\Gamma(1 + i\mu)} e^{-i\mu y}\, d\mu. \tag{54}$$

As the sought for function $g(\lambda)$ is really the density distribution of different values λ, so from the second relation (47) we obtain

$$g(\lambda)\, d\lambda = g(y)\, dy \left| \frac{d\lambda}{dy} \right| = g(y)\, dy\, |\lambda|, \tag{55}$$

and therefore, briefly, in conclusion

$$g(y)\, dy = \frac{g(\lambda)}{\lambda} d\lambda.$$

Gardner's method is distinguished by its elegance and it is mathematically rigorous if one applied this method for a given function $f(t)$ with no noise, and besides this function should be known at the full real axis $t \in [0, \infty)$.

But instead of that we only know the function $F(t)$ together with the noise $n(t)$, and in a limited range of $t \in (t_{\min}, t_{\max})$. It follows from this that there appear spurious high-frequency components of the spectral function $F(\mu)$, and the integral (54) diverges in high frequencies. These difficulties are not specific for Gardner's method, but they are always common for inverse problems dealing with noisy experimental data. All variety of methods for the solution of inverse problems is divided according to the noise filtering methods which are used. All the paper mentioned above [31–40] use really different kinds of noise protection filters.

As one can understand from this brief outline, the Gardner method is linear according to the standard definition: the sum of two input independent signals corresponds to the sum of two output independent results of restoration. That is the main reason why this method can resolve and restore only those signal harmonics which are not lost in the input noise. This is the origin of the resolution limit for linear methods.

But the principal resolution limit for any non-parametric restoration method (including of course nonlinear ones) is determined by the Shannon theorem [17] and this limit is obviously higher in comparison with the only linear methods. The authors of this paper were influenced by these consideration when they started testing the ML method for exponential analysis problems.

It is very easy, in principle, to implement this idea. We apply the Gardner transform (47) and come to the convolution integral equation (49) with the kernel function (50) instead of the Laplace transform equation (44). The Gardner apparatus function or PSF (50) corresponding to this kernel is unsymmetric and its range is equal to

$$\Delta = \int_{-\infty}^{\infty} K^2(s)\, ds / K^2(0) = \tfrac{1}{4} e^2 \approx 1.85, \tag{56}$$

where

$$K(s) = \exp(-e^s + s).$$

After that the convolution integral equation is solved by the ML method using the fast Fourier transform described in section 4.

Test computations revealed, however, that the convergence rate of the algorithm, based on iteration formula (10) or (27), was not fast enough in difficult cases: e.g. there are close values of decrements λ, or there is a small input interval of available values t. The enhanced version of the ML method algorithm is described below.

Because the direction of looking for the maximum of the likelihood function is computed in both formulas (10) and (27) by multiplication of the gradient vector by the nonnegative factors g_k, both of these algorithms can be denoted as "quasigradient" ones.

In the enhanced version of the algorithm there is a new way of finding the direction of looking for the maximum over one iteration. This direction is the linear combination of the "quasigradient" vector and the search direction in the previous iteration.

The CPU time for one iteration is larger with about 20–25% in the new enhanced algorithm in comparison with the old one. But because of the faster convergence of the new algorithm, the total time, which is proportional to the total number of iterations, is 2–3 decimal orders smaller. We call the new enhanced algorithm a quasi-conjugate-gradient one.

The example of using the ML recovery algorithm for the well-known test of 6 exponents recovery [37] is shown in fig. 3. In contrast to ref. [37] we use input data having noise with a signal-to-noise ratio of 60 dB, whereas in ref. [37] input data without noise was used. The number of input and output data point was equal to 256, and

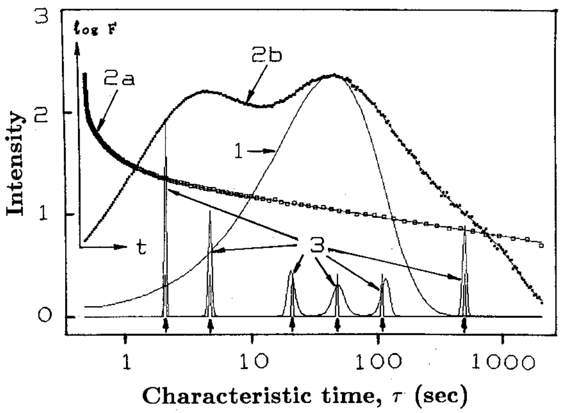

Fig. 3. Recovery of six exponential functions with unknown decrements and amplitudes at a 60 dB signal-to-noise ratio. $N = 256$ input and output data points. 1 – the Gardner instrumental function (see eqs. (50) and (56)); 2a – input data in semilogarithmic scale (see coordinate axes in inset); 2b – same input data after the Gardner transform eq. (47); 3 – recovery result and original lines, indicated by arrows from below. The number of iterations equals 1200 and $\chi^2/N = 0.885$. $g(\lambda)/\lambda$ is intensity on the Y-axis and $\tau = 1/\lambda$ is time on the X-axis. The true solution (indicated by arrows from below) is expressed by the formula

$$f(t) = 10\,e^{-0.002t} + 45\,e^{-0.009t} + 100\,e^{-0.02t} + 225\,e^{-0.045t} + 1000\,e^{-0.2t} + 2250\,e^{-0.45t}.$$

the variable t lay in the range 0.5–2000 s. It is seen from this figure, that all 6 exponents are well resolved. The actual restoration accuracy is presented in table 2.

For the evaluation of the place of the ML algorithm for exponential analysis among the other algorithms, it should be noted, that the ML algorithm (as well as the Gardner and Pike methods) are nonparametric methods and because of that, these algorithms should always be used if there is no specific information that the parametric model (46) is adequate for signal which is being sought.

Table 2
The actual recovery accuracy of the line positions and areas under peaks for the example shown in fig. 3.

	Relative accuracy for recovering (%)					
Position	0.33	0.06	−0.36	0.51	0.91	−0.09
Area under peaks	0.19	−2.19	−2.31	11.96	−7.13	−0.53
Peak no.	1	2	3	4	5	6

We agree with the conclusion of the authors of papers [35–40] that the two-stage recovery procedure would probably be optimal, although the nonparametric algorithm should be applied first. If and only if it is found that the parametric model (46) is adequate, the parametric algorithm should be applied secondly.

From the three methods by Pike et al., Gardner and ML, the first one has probably the least resolution which is determined by the number of basis function actually used, but it is quite effective for a moderate number of experimental data. If is reasonable to apply the Gardner or ML algorithms for a number of experimental data not less than 100–200. The speed of the Gardner algorithm (especially based on the FFT) is larger than the speed of the ML method, but the resolution is better for the ML algorithm. It would be interesting to compare the actual usefulness of the three methods using the same experimental data sets.

Finally, we shall now simply explain the conclusion of Pike et al. about the advantage of the logarithmic uniform spacing. This is true because in such a spacing the distribution of the sampling is uniform over the PSF (50), but not condensed in the right side of the PSF. It results in a better resolution of the lines sought if they are also uniformly distributed in the logarithmic scale.

6. High-resolution soft X-ray spectrum reconstruction by MWPC attenuation measurements

It was shown in paper [18] that this problem is reduced to the solution of the integral equation

$$N(x) = \int_{\Delta E} I(E)[1 - e^{-\alpha(E)\Delta}]\, e^{-\alpha(E)x}\, dE. \tag{57}$$

In this equation x is the depth of an attenuation layer from the entrance point up to coordinate x, $N(x)$ is the measured intensity as a function of depth, $I(E)$ is the unknown energy spectrum at the entrance point of the attenuation material, $\alpha(E)$ is the linear absorption coefficient as a function of energy E, Δ is the size of one section

Table 3
The experimental data of the X-ray tube spectral measurements by the MWPC filled with molecular hydrogen at room temperature.

Cathode number in MWPC	Number of pulses for 100 s		
	experiment A	experiment B	experiment C
1	753100	876334	850281
2	259949	243066	240589
3	96431	84524	85179
4	40552	38256	41309
5	19889	21050	24440
6	10674	12816	16796
7	6539	8218	11518
8	3055 [a)]	5426	8337
9	2625	3406	5902
10	1764	–	4337

Parameters for experiments:
A: Anode–cathode voltage 410 volts, hydrogen pressure 20 atm.
B: 448 volts, 25 atm.
C: 470 volts, 25 atm.
[a)] This is probably an outlyer. This value was actually replaced by 4143 which was obtained by linear interpolation of the data logarithms.

of the multiwire proportional counter (MWPC) which is used for attenuation measurements. The factor

$$w(E) = 1 - \exp[-\alpha(E)\Delta] \tag{58}$$

represents the registration efficiency, and this factor is proportional to the probability of absorption of a photon having energy E. The integration in (57) is extended over the energy ΔE, where $W(E) \geq 0$.

The integral equation (57) is the special case of the general equation (1), and the experimental data $N(x)$ usually have the Poissonian distribution at every x. Because of that we can apply the iteration procedure (10) which is described in section 2.

Extensive numerical results for the testing of the efficiency of the iteration procedure (10) have been given in paper [18]. Table 3 shows the original experimental data, which were used in ref. [18] for experimental tests in X-ray tube spectra measurements.

The upper bound of energy spectra, recovered from the experimental data presented in table 3, is equal to the tube voltage, and the carbon line C_{K_α} having energy 275 eV appears in each measurement. All these features, obtained with the program Datten, are clearly seen in fig. 4.

7. On the fast Fourier transform programs

Among many present-day FFT programs for $N = 2^n$, $n = 1, 2, 3, \ldots$ data points there should be noted those ones, which are based on the "split-radix" algorithm [20]. These programs probably have a minimal total number of arithmetic operations (additions and multiplications together). One of the first programs of this kind, based on the algorithm by I.E. Kaporin [21], was compared in paper [10] with the IBM FFT program [22] and the FFT program by R.C. Singlton e ✓ [23]. It was shown in ref. [10] that for 512 data points the first program is about two times faster than the second program and four times faster than the third one.

Table 4 shows the FFT execution times (in seconds) for 1024 real values with single precision for programs by J.B. Martens [24], H.V. Sorensen [25] and G. Cabras et al. [26] in comparison with the program by I.E. Kaporin and the new one based on the algorithm of ref. [27]. This algorithm also belongs to the "split-radix" class but it uses a faster digit-reversal permutation algorithm than the one in the papers by D.M. Evans [28].

It is seen from this table that our new FFT program is the fastest one of all mentioned here. As an important fact should also be noted that our new program requires 2.5 times less size of working array in comparison with the program by I.E. Kaporin.

This size can be reduced to even more than twice, for the costs of a speed decrease of 3–4%. If the inner loop in the FFT is prepared using

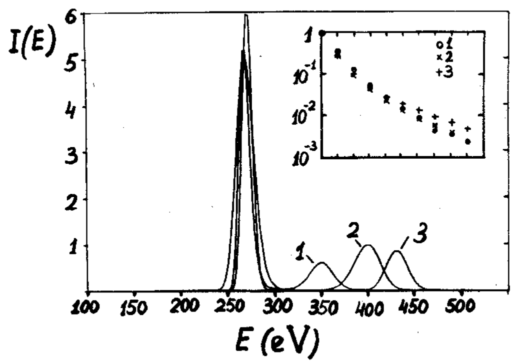

Fig. 4. The X-ray tube spectrum reconstruction. The insert shows the input data of three independent experiments: 1 – the tube voltage is 410 V; 2 – 448 V, 3 – 470 V. The strong line is the carbon C_{K_α} emission line at 275 eV. The number of photons in each of the three experiments can be seen in table 3. The resolution of the C_{K_α} line is about 15 eV (5%) (see paper [18]).

assembly
✓ Assembler language instead of Fortran, the speed of the program with a small memory model can be increased with 1.5 times in comparison with the GFT4 program.

The computing times shown in the table 4 were measured on a personal computer IBM ✓ PS/2, model 50 equiped p with a 80287 coprocessor using the IBM Fortran/2 compiler with option I and Y2, working under the operation system DOS 3.30. The operations for initialization (that is computation of the sin and cos tables) and normalization of the FFT transform results are not included in all programs mentioned in table 4.

These times can vary on different computers and different compilers, but the main conclusion is not changed: our new fast Fourier transform program is the best one among the ones mentioned, due to its speed and small size of working arrays.

Table 4
FFT computing time (in seconds) for 1024 real*4 values on an IBM PS/2 personal computer, equiped P with 286 and 287 processors. ✓

Martens [24] RCF2 (real and unsc.)	Sorensen [25] RV1FFT	Duhamel [20] SPLITDIT	Cabras [26] RFFT	Kaporin [21] FFT87	Gelfgat [27] GFT4
0.76	0.52	0.49	0.53	0.52	0.46

8. Conclusion

Different parts of the software for deconvolution RECOVERY have been used at the P.L. Kapitza Institute for Physical Problems starting from 1980, and this software naturally evolved all the time. All programs of this software package are based on the maximum likelihood principle. This is a short outline of its history: use the steepest ascent method for maximization of the likelihood function instead of the quasigradient one with the constant step length h for the Poissonian experimental data; choose of the optimal step length h by the parabolic extrapolation of the likelihood function instead of the bisection method used before; compute the convolution via the fast Fourier transform; the generalization of the steepest ascent method for experimental data having the Gaussian distribution; use of the ML method for the Gardner transformed experimental data in exponential analysis and use of the quasi-conjugate gradient method for maximization of the likelihood function.

The authors hope to continue this work and will be grateful to users of the RECOVERY software for possible comments and suggestions and for information about its applications in different problems.

Acknowledgement

The authors wish to thank Prof. P. Duke of Daresbury Laboratory, UK, for his interest in this work and for his many valuable suggestions to improve the English style of this paper.

References

[1] V.F. Turchin, V.P. Kozlov and M.S. Malkevich, The use of mathematical statistics methods in the solution of incorrectly posed problems, Sov. Phys. Uspekhi, 13 (1971) 681.

[2] B.R. Frieden, Image enhancement and restoration, in: Picture Processing and Digital Filtering, ed. T.S. Huang, (Springer, Berlin, 1979) pp. 179–248.

[3] E.L. Kosarev, Application of the 1-st kind integral equations in experimental physics. Comput. Phys. Commun. 20 (1980) 69.

[4] A.N. Tichonov and V.Ya. Arsenin, Solution of Ill-posed Problems (Wiley, New York, 1977).

[5] G.I. Vasilenko and A.M. Taratorkin, Restoration and Images, (Radio i swjaz, Moscow, 1986) (in Russian).

[6] W.E. Blass and B.W. Halsey, Deconvolution of Absorption Spectra (Academic, New York, 1981).

[7] P.A. Jansson, ed. Deconvolution with Application in Spectroscopy (Academic, New York, 1984).

[8] A.N. Tichonov, A.V. Gontcharskii, V.V. Stepanov and A.G. Jagola, Numerical Solution of Ill-posed Problems (Fizmatgiz, Moscow, 1990) (in Russian).

[9] E.L. Kosarev and E. Pantos, Optimal smoothing of noisy data by fast Fourier transform, J. Phys. E. 16 (1983) 537.

[10] E.L. Kosarev and E. Pantos, Optimal smoothing of data plus noise using fast Fourier transform, in: Instruments and Experimental Techniques. Vol. 28, No. 3, part 1, (Consultants Bureau, New York, 1985) p. 600.

[11] V.N. Vapnik, ed. Algorithms and Programs for Reconstructing of the Dependences (Fizmatgiz, Moscow, 1984) (in Russian).

[12] A.F. Verlan and V.S. Sizikov, Integral Equations: Methods, Algorithms and Programs (Naukova Dumka, Kiev, 1986) (in Russian).

[13] S.W. Provencher, CONTIN: A general purpose constrained regularization program for inverting noisy linear algebraic and integral equations, Comput. Phys. Commun. 27 (1982) 229.

[14] H.J.H. te Riele, A program for solving first kind Fredholm integral equations by means or regularization, Comput. Phys. Commun. 36 (1985) 423.

[15] C. Radharkishna Rao, Linear Statistical Inference and its Applications (Wiley, New York, 1965).

[16] E.L. Kosarev, Superresolution limit for signal recovery, in: Maximum Entropy and Bayesian Methods, ed. J. Skilling (Kluwer, Dordrecht, 1989) pp. 475–480.

[17] E.L. Kosarev, Shannon's superresolution limit for signal recovery, Inv. Probl. 6 (1990) 55.

[18] E.L. Kosarev, V.D. Peskov and E.R. Podolyak, High resolution soft X-ray spectrum reconstruction by MWPC attenuation measurements, Nucl. Instrum. Methods 208 (1983) 637.

[19] M.Z. Tarasko, On the method for solution of the linear system with stochastic matrixes, Preprint of the Physics and Energy Institute PEI-156, Obninsk (1969) (in Russian).

[20] P. Duhamel and M. Vetterli, Improved Fourier and Hartley transform algorithms: Application to cyclic convolution of real data, IEEE Trans. Acoust. 35 (1987) 818.

[21] I.E. Kaporin, A new algorithm for fast Fourier transform, Sov. Comput. Math. Math. Phys. 20 (1980) 1054 (in Russian).

[22] System/360 Scientific Subroutine Package, Fourth Edition, IBM (1968).

✓ [23] R.C. Singlton, ALGOL procedures for the fast Fourier transform, Commun. ACM, 11 (1968) 773.

Singleton

[24] J.B. Martens, Recursive cyclotomic factorization – A new algorithm for calculating the discrete Fourier transform, IEEE Trans. Acoust. 32 (1984) 750.

[25] H.V. Sorensen, D.L. Jones, M.T. Heideman and C.S. Burus, Real-valued fast Fourier transform algorithms, IEEE Trans. Acoust. 35 (1987) 849.

[26] G. Cabras, V. Roberto and G. Salemi, A package of optimized discrete Fourier transforms, Comput. Phys. Commun. 47 (1987) 113.

[27] V.I. Gelfgat, Economical algorithms for the discrete Fourier transform, in: Numerical Methods in Modern Acoustics Wave Problems (Acoustic Institute, Moscow, 1988) pp. 60–61 (in Russian).

[28] D.M. Evans, An improved digit-reversal permutation algorithm for the fast Fourier and Hartley transforms, IEEE Trans. Acoust. 35 (1987) 1120; A second improved digit-reversal permutation algorithm for fast transforms, IEEE Trans. Acoust. 37 (1989) 1288.

[29] C. Lanczos, Applied Analysis (Prentice-Hall, Englwood Cliffs, NJ, 1956).

[30] E.C. Titchmarsh, Introduction to the Theory of Fourier Integrals (Oxford University Press, Oxford, 1937).

[31] J.G. McWhirter and E.R. Pike, On the numerical inversion of the Laplace transform and similar Fredholm integral equation of the first kind, J. Phys. A. 11 (1978) 1729.

[32] E.R. Pike, J.G. McWhirter, M. Bertero and C. de Mol, Generalized information theory for inverse problems in signal processing, Proc. IEE, Pt. F, 131 (1984) 660.

[33] M. Bertero, P. Brianzi and E.R. Pike, On the recovery and resolution of exponential relaxion rates from experimental data: Laplace transform inversions in weighted spaces, Inv. Probl. 1 (1985) 1.

[34] M. Bertero and E.R. Pike, Exponential-sampling method for Laplace and other dilationally invariant transforms: I. Singular-system analysis, Inv. Probl. 7 (1991) 1; II. Examples in photon correlation spectroscopy and Fraunhofer diffraction, Inv. Probl. 7 (1991) 21.

[35] D.G. Gardner, J.C. Gardner, G. Laush and W.W. Meinke, Method for the analysis of multicomponent exponential decay curve, J. Chem. Phys. 31 (1959) 978.

[36] J. Schlesinger, Fit to experimental data with exponential functions using the fast Fourier transform, Nucl. Instrum. Methods 106 (1973) 503.

[37] M.R. Smith and S. Cohn-Sfetch, Comment on "Fit to experimental data with exponential functions using the fast Fourier transform", Nucl. Instrum. Methods 114 (1974) 171.

[38] H.J. Stockmann, A new method to analyse composite exponential decay curves, Nucl. Instrum. Methods 150 (1978) 273.

[39] B. Kober, W. Gruhle and M. Hillen, Gardner analysis of decay components with strong fluctuations, Nucl. Instrum. Methods (1980) 173.

[40] M.R. Smith and S.T. Nichols, Improved resolution in the analysis of multicomponent exponential signals, Nucl. Instrum. Methods 205 (1983) 479.

Computer Physics Communications 74 (1993) 349–357
North-Holland

Computer Physics Communications

Programs for signal recovery from noisy data using the maximum likelihood principle

II. Program implementation

V.I. Gelfgat, E.L. Kosarev and E.R. Podolyak

P.L. Kapitza Institute for Physical Problems, Moscow 117334, Russian Federation

Received 4 August 1992

Three subroutines intended for nonnegative signal recovery from noisy experimental data distorted by the measuring device are presented. The use of these subroutines is illustrated by a test program.

PROGRAM SUMMARY

Title of programs: MLG8, MLU8, MLP8 from the RECOVERY package

Catalogue number: ACLJ

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Licensing provisions: Persons requesting the program must sign the standard CPC nonprofit use licence.

Computer: IBM PS/2 mod.50

Operating system under which the program has been tested: PC DOS 3.30

Programming language used: IBM FORTRAN/2

Memory required to execute with typical data: 200 Kb

No. of bits in a word: 16

No. of lines in distributed programs, including test data, etc.: 4888

Correspondence to: E.L. Kosarev, P.L. Kapitza Institute for Physical Problems, Moscow 117334, Russian Federation.
E-mail: kosarev@~~magnit.msk.su~~.
kapitza.ras.ru

Separate documentation available: user's manual of the RECOVERY software; *no. of pages*: 9

Keywords: ill-posed problems, maximum likelihood, steepest descent

Nature of physical problem
The programs allow reconstruction of nonnegative signals from noisy experimental data distorted by the measuring device [1]. Noise may have a Gaussian, binomial or Poissonian distribution at each experimental point. These programs may be used for: (a) reduction to an arbitrary instrumental function, which depends only on the difference of its arguments (this is the typical spectroscopic problem); (b) the expansion of an exponentially decaying curve which has a continuous spectrum of decrements (this problem has applications in time-dependent fluorescence, nuclear physics etc.); and (c) the ultrasoft X-ray spectrum recovery from absorption measurements.

Method of solution
These programs are based on the maximum likelihood principle and use an algorithm of steepest descent class to find the maximum of the likelihood function.

Restrictions on the complexity of the problem
There are no principle restrictions on the usage of the programs. The only practical restriction is posed by the memory available on the computer.

Typical running time
It takes less than one minute for test examples mentioned in this paper and some tens of minutes for the examples presented in ref. [1].

Unusual features of the program
Programs do not finish the work automatically. They just perform the required number of iterations. The users should check the quality of the solution according to the specifications of their own problems.

Reference
[1] V.I. Gelfgat, E.L. Kosarev and E.R. Podolyak, Programs for signal recovery from noisy data using the maximum likelihood principle. I. General description, Comput. Phys. Commun. 74 (1993), this issue.

LONG WRITE-UP

1. Program description

The subroutines presented in this paper are intended to solve the linear problem which may be written as a system of linear algebraic equations

$$\sum_{k=1}^{M} A_{ik} G_k^0 = F_i^0, \quad i = 1, \ldots, N. \tag{1}$$

providing $G_k^0 \geq 0$, $k = 1, \ldots, M$. This linear system may represent a dicrete form of integral equation of the first kind or any other linear operator equation which describes the process of physical measurement. Here $\{G_k^0\}$ is the unknown physical quantity; $\{F_i^0\}$ is the result of the transformation (or distortion) by the measuring device with the instrumental (point spread) function A_{ik}. There is always a finite measurement accuracy due to some sort of noise, so in practice one does not obtain the values of $\{F_i^0\}$ but the values $\{F_i\}$ spoiled by noise. And only random vector $\{F_i\}$ is available for problem solution. A solution is searched on the basis of the maximum likelihood principle which requires the knowledge of statistical properties of noise [1]. Here we assume that the components of random vector $\{F_i\}$ corresponding to different i are independent and are described either by a Gaussian, multinomial or Poissonian distribution function.

This paper presents three subroutines MLG8, MLU8 and MLP8 handling different noise distribution functions P:

(a) Subroutine MLG8 processes Gaussian noise in general form

$$P\left(F_i \mid F_i^0\right) = \text{constant} \times \exp\left[-\frac{1}{2}\left(\frac{F_i - F_i^0}{\sigma_i}\right)^2\right],$$
$$\text{for } i = 1, \ldots, N, \tag{2}$$

and σ_i denotes standard deviation of F_i from F_i^0;

(b) Subroutine MLU8 expects Gaussian noise with equal standard deviations at different points i, i.e. $\sigma_i = \sigma = \text{constant}$,

$$P\left(F_i \mid F_i^0\right) = \text{constant} \times \exp\left[-\frac{1}{2}\left(\frac{F_i - F_i^0}{\sigma}\right)^2\right]. \tag{3}$$

This case may of course be processed by the MLG8 routine but it will work slower than MLU8 due to unnecessary operations with constant values of σ_i and, moreover, the general routine MLG8 requires additional memory space to hold the array of standard deviations.

(c) Subroutine MLP8 expects the multinomial distribution function

$$P(\{F\} \mid \{F^0\}) = \frac{\left(\sum_j F_j\right)!}{\prod_j F_j!} \prod_i \left(\frac{F_i^0}{\sum_j F_j^0}\right)^{F_i} \tag{4}$$

and it can be used also in the case of a Poissonian distribution if the total sum of measured values F_i is much greater than 1, i.e. $\Sigma F_i \gg 1$.

These subroutines are intended to find the vector $\{G^*\}$ maximizing the likelihood function $\mathscr{P}(\{G^*\})$ providing

$$G_k^* \geq 0, \quad k = 1,\dots,M. \tag{5}$$

The form of likelihood function $\mathscr{P}$ depends on the noise distribution function, so in case of Gaussian distribution it is as follows,

$$\mathscr{P}(\{G^*\}) = \prod_{i=1}^{N} P(F_i \mid F_i^*) = \text{constant} \times \exp\left[-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{F_i - F_i^*}{\sigma_i}\right)^2\right],$$

where

$$F_i^* = \sum_{k=1}^{M} A_{ik} G_k^*, \tag{6}$$

and in the case of multinomial distribution,

$$\mathscr{P}(\{G^*\}) = \text{constant} \times \exp\left[\sum_{i=1}^{N} F_i \log\left(\frac{F_i^*}{\sum_j F_j^*}\right)\right]. \tag{7}$$

Other noise statistics may also be processed using the same maximum likelihood technique which requires only the redefinition of the likelihood function and rewriting several related expressions in the maximization algorithm.

2. Characteristics of the subroutines

The detailed description of the algorithm can be found elsewhere [1] so here we present only the main characteristics of the subroutines. The search of the likelihood function maximum is performed by the steepest ascent method which is a sign inverted variant of steepest descent method. The implementation contains the following differences from the standard scheme:

- The vector of unknowns $\{G_k\}$ is normalized, i.e. it is replaced by new variables Gsum and $\{g_k\}$ defined by Gsum $= \sum_{k=1}^{M} G_k$ and $g_k = G_k/\text{Gsum}$. Maximizing of the likelihood function with respect to Gsum is done analytically which results in decrease of the dimension of the system and provides a much better convergence rate as compared to the unnormalized case.
- The search of the likelihood function maximum at each iteration is performed along the direction of the so called "conditional" gradient. The kth component of the "conditional" gradient is obtained by multiplication of the corresponding component of the gradient by a nonnegative factor g_k. This modification of search direction along with appropriate choice of step length (see below) provides the fulfillment of constraints (5). The "conditional" gradient can be thought of as a result of projection of the gradient to keep the vector of unknowns in the first octant of the corresponding space.
- The value of the step in the search direction is calculated from 1D maximization of the likelihood function along the "conditional" gradient. The value of the step is constrained to fulfill eq. (5). This 1D maximization is done analytically rather than numerically. Since most of the computational time is spent to solve the direct problem (1), i.e. to perform matrice-by-vector multiplication, it is necessary to reduce the number of such multiplications. Analytical calculation of the step implemented in subroutines requires only one additional matrice-by-vector multiplication to find an optimal value.

All these changes to standard steepest ascent method concern only accounting for constraints (5) and they are aimed at improvement of the convergence rate.

3. Input parameters

The types of all input and output parameters correspond to IBM PC clone architecture, i.e. all REAL variables and vectors are the REAL*8 ones since the difference in REAL*8 and REAL*4 arithmetic performance is negligible but the convergence rate of the algorithm is much better using the higher computation accuracy. All INTEGER variables are of type INTEGER*2, because a typical RAM capacity will hardly allow large system dimensions requiring INTEGER*4 range.

The notations of arguments in the subroutine slightly differ from those used in formulae (1)–(5)

and paper [1]. It is caused by the case insensitivity of the FORTRAN language which makes no difference between lower and upper case symbols.

The calling sequence is very similar for all three routines so we present here the description of parameters for MLG8 and emphasize the differences in calling of other subroutines:

call MLG8 (SUBR8,NF,F,P,Disp,NG,DG,Limit, Chi2,IFlag,WS,NWS)

Here SUBR8 is the name of the user defined external subroutine (see below) containing the description of the problem to be solved, i.e. eq. (1). NF is the number N of experimental data $\{F_i\}$ which are passed in the REAL*8 array F. Disp is the REAL*8 array of squared standard deviations $\{\sigma_i^2\}$. The representation of standard deviations is the main difference in the calling sequence of the subroutines which are presented. Subroutine MLU8 uses not an array but a scalar REAL*8 value of $\sigma^2 = \sigma_i^2 =$ constant which is passed in the same place Disp. Subroutine MLP8 does not use standard deviations at all so Disp is excluded from the list of its parameters. The subroutines also require the number M of unknowns, passed as the INTEGER*2 variable NG and possibly initial guess $\{G_k^*\}$ of the solution vector, which is the REAL*8 array DG with NG elements. Each component of this initial guess must be nonnegative. The INTEGER*2 parameter IFlag controls the usage of the initial guess $\{G_k^*\}$: it is used only when IFlag = 0. If IFlag > 0, then the initial guess is not required and the program uses internal variables to continue the maximization process. Limit is the number of iterations to be performed. The program does not allocate any fixed sized arrays so the user must supply an additional REAL*8 array, WS, to be used as working storage. Dimension NWS of this working storage must not be less than (NF + NG).

We have already mentioned that the only difference between the MLG8 and MLU8 routines is that unnecessary operations with constant values of $\sigma_i = \sigma =$ constant have been removed. This removing markedly speeds up the computation. In the case of multinomial distribution of experimental data there is another opportunity to improve the subroutine performance due to the fact that the value of Gsum is really fixed Gsum = ΣF_i and optimization with respect to Gsum is not needed. But this improvement is at the cost of normalization of some input and output arrays. This redefinition is explained in ref. [1] and can be easily traced in the example below.

4. Output parameters

The subroutines check all input parameters for legality and if any of these checks fails then subroutines immediately return to caller with the INTEGER*2 parameter IFlag set to negative error code, otherwise parameter IFlag remains unchanged on exit. Possible error codes are summarized in the following:

IFlag = −1	number of data points, NF < 2,
IFlag = −2	number of unknowns, NG < 2,
IFlag = −3	working storage is too small, NWS < (NF + NG),
IFlag = −4	illegal component, Disp(I) ≤ 0,
IFlag = −5	illegal component in vector of initial guess, DG(K) < 0,
IFlag = −6	can not find nonnegative solution, i.e. Gsum < 0.

According to the different usage of parameter Disp, which is stated in the previous section, subroutines MLP8 never returns parameter IFlag = −4 and subroutine MLU8 returns this value in case of an illegal scalar value Disp.

The obtained solution is returned via the REAL*8 array DG and corresponding approximation $\{F_i^*\}$ of the right-hand side via the REAL*8 array P. The quality of the approximation can be estimated by the REAL*8 value of

$$\text{Chi2} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{F_i - F_i^*}{\sigma_i} \right)^2 ,$$

which can be tested using the χ^2 criterion to decide to continue or not the iteration process.

5. External subroutines

In the course of iterations it is necessary to solve a direct problem, i.e. to calculate the sum

(1) for a given vector $\{G_k\}$. It gives an opportunity for the user to formulate the original linear problem in his own fashion. For example, matrice-by-vector muliplication in eq. (1) may be replaced by fast Fourier transform convolution in case of a translation invariant instrumental function. The user-defined implementation of the direct problem (1) should be done via subroutine SUBR8, which must be declared EXTERNAL in the calling program.

The calling sequence is as follows:

call SUBR8 (NX,X,NY,Y,IDIR)

Input parameters of this subroutine NX, X,NY,IDIR should remain unchanged on return:

NX	number of input data,
X(NX)	REAL$*$8 array to be multiplied by matrix,
NY	number of output data,
IDIR	IDIR $= 1$, use original matrice $Y_i = \sum_{k=1}^{\mathrm{NX}} A_{ik} X_k$, IDIR $= -1$, use transposed matrice $Y_k = \sum_{i=1}^{\mathrm{NX}} A_{ik} X_i$.

Note, that when IDIR $= 1$, the input array represents vector $\{G\}$ with M components and SUBR8 should return vector $\{F\}$ with N components. ✓ And, vise versa, when IDIR $= -1$, the input array represents vector $\{F\}$ or $\{F*\}$ or their linear combination with N components which should be multiplied by transposed matrix A_{ij} producing a vector with M components. The subroutine SUBR8 is never called with parameter IDIR other than ± 1, so other values of IDIR may be used for various user purposes, i.e. matrice initialization etc. Note, that the matrix A_{ij} is not supplied as an actual parameter to any of the subroutines. It should be either defined in SUBR8 as a fixed array or passed to this routine via a COMMON block. The only output parameter of this subroutine is the REAL$*$8 array Y which is the result of multiplication.

6. Test run

Test examples presented in this paper are aimed not to demonstrate the power of the maximum likelihood technique but rather to facilitate implementation of presented subroutines in various user programs. So we chose simple linear problems with "known answer" which can be easily reproduced and understood. The dimensions of the system, $N = 20$ equations with $M = 20$ unknowns, are chosen to minimize the amount of print out. A Lorentz matrix,

$$A_{ik} = 1/\left(1 + (i-k)^2/\alpha^2\right),$$

is taken as an instrumental function. The parameter α describing the width of the Lorentz profile is taken $\alpha = 4$. The true solution of the system is taken in the form

$$G_k^0 = 0,\ k \neq 7,\ 13 \text{ and } G_7^0 = G_{13}^0 = 10\,000, \qquad (8)$$

This true soltion is kept in the array G0 and the current approximation of solution $\{G_k^*\}$ is kept in the array G. The right-hand side $\{F_i^0\}$ is calculated according to eq. (1) and then the noise is added in a different way for each of the subroutines:

- the noisy right-hand side for the general subroutine MLG8 is calculated with fixed relative accuracy of 1%, i.e. $\sigma_i = 0.01\ F_i^0$;
- the noisy right-hand side for subroutine MLU8 is calculated with a constant absolute value of $\sigma_i = \sigma = 20.0$;
- and the noise for subroutine MLP8 is calculated according to multinomial distribution around mean values of $\{F_i^0\}$.

For the sake of simplicity the values of the noisy right-hand side $\{F_i\}$ are precalculated and saved in the DATA statement. The test program prints out the results of the calculation of every 100th iteration and terminates after iteration number 500. χ^2 test is not performed and values of Chi2 variables are only printed for reference as well as iteration counter.

7. The RECOVERY software for deconvolution

The three main subroutines MLU8, MLG8 and MLP8 described just above are used as a basis for a unique software deconvolution package RECOVERY. Nowadays the package consists of three main programs for:

- deconvolution, i.e. a spectral resolution enhancement,
- exponential analysis, i.e. a decay-time analysis, and
- radiation absorption processing.

There are two versions of the programs for deconvolution and for exponential analysis: Dconv, Dconv2 and Dexpa, Dexpa2, respectively. The first versions of them are designed for the number of experimental data N equal to some power of 2

$$N = 2^m, \quad m = 2, 3, \ldots.$$

The second ones are able to use any number of experimental data. The program for radiation absorption processing Datten also has the ability to use any number of experimental data.

For the sake of efficiency the fast Fourier transform subroutine GFt4 [1] is used for the circle convolution calculations in the programs Dconv, Dconv2 and Dexpa, Dexpa2. The logic diagram of the RECOVERY software is presented in fig. 1. A more detailed description of the input and output data format and instructions how to use the RECOVERY software can be found elsewhere [2].

There also are enhanced versions of the subroutines MLU8, MLG8 and MLP8 which use the conjugate gradient method for maximization of the likelihood function instead of the gradient one [3]. The ready-to-use (executable) form of this enhanced software for IBM compatible PC's is available from the second author (E.L.K.) on a commercial basis. The only difference between the ordinary and enhanced versions of these programs is the computation time and hence it is essential for the PC's but not for more powerful computers. The full listings in source form of the programs Dconv, Dconv2, Dexpa, Dexpa2, Datten * and subroutines MLU8, MLG8, MLP8, MATX8, CNVprim, GFt4, RASF8, WRASF8 and GRAPHS are available from the CPC Program Library.

* And also the test program CPC.TST, which was described in section 6.

8. Conclusions

The subroutines presented in this paper demonstrate good efficiency in solving linear problems with noisy right-hand side. The only prior information about the solution is the nonnegativeness of its components. This prior knowledge is not incorporated in the objective function but it is used in the nonlinear projection operator implemented in the steepest descent algorithm. This scheme is especially effective when the solu-

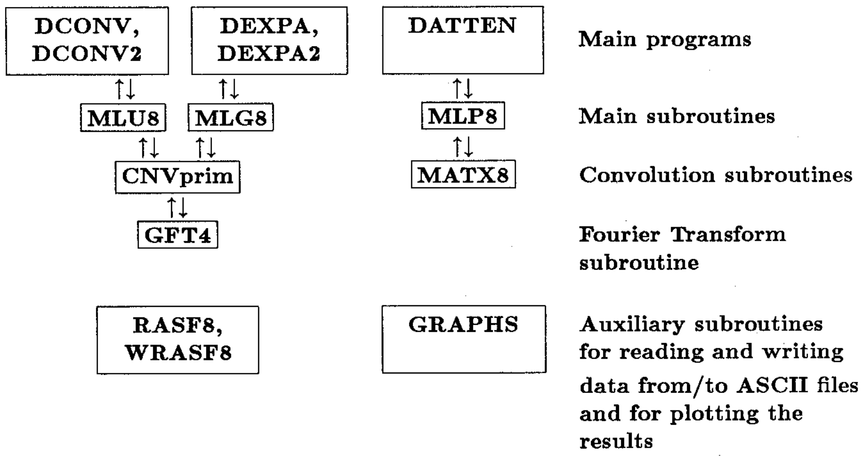

Fig. 1. Logical diagram of the RECOVERY package.

tion vector consists of several narrow nonzero structures separated by zero-valued components. In such cases the nonnegativeness constraints make the maximization of the likelihood function a well-posed problem and the routines presented in this paper can be efficiently used to solve it.

References

[1] V.I. Gelfgat, E.L. Kosarev and E.R. Podolyak, Programs for signal recovery from noisy data using the maximum likelihood principle. I. General description, Comput. Phys. Commun. 74 (1993) 335, this issue.

[2] E.L. Kosarev, User's Manual of the RECOVERY Software. Available from the author or the CPC Program Library.

[3] V.I. Gelfgat, E.L. Kosarev and E.R. Podolyak, Programs for signal recovery from noise data by maximum likelihood method, Instrum. Exp. Techn. 34 (1992) 1070.

TEST RUN OUTPUT

```
                         results of MLG8 routine

iter       100          200          300          400          500
===================================================================
01        .027         .000         .000         .000         .000
02        .058         .000         .000         .000         .000
03        .683         .008         .000         .000         .000
04      18.783        1.152         .113         .015         .002
05     382.565      119.225       43.092       17.569        7.849
06    2454.740     2266.279     1974.039     1712.287     1493.888
07    3864.712     4924.718     5653.207     6209.409     6655.010
08    2125.138     2045.600     1928.277     1797.774     1666.320
09     859.525      514.132      340.204      236.163      169.137
10     556.225      262.636      145.180       86.661       54.370
11     827.526      483.596      313.040      213.032      149.816
12    2056.191     1952.375     1817.426     1677.295     1541.958
13    3812.947     4844.549     5547.083     6080.726     6506.926
14    2502.201     2359.459     2111.188     1881.951     1686.643
15     437.601      161.575       71.251       35.803       19.819
16      24.013        2.262         .371         .086         .025
17        .614         .012         .000         .000         .000
18        .014         .000         .000         .000         .000
19        .000         .000         .000         .000         .000
20        .000         .000         .000         .000         .000
===================================================================
Chi2 5.643E+00   3.294E+00    2.474E+00    2.056E+00    1.802E+00
```

```
                         results of MLU8 routine

iter       100          200          300          400          500
===================================================================
01        .000         .000         .000         .000         .000
02        .000         .000         .000         .000         .000
03        .002         .000         .000         .000         .000
04        .237         .000         .000         .000         .000
05      43.432         .217         .003         .000         .000
06    1886.436      659.733      274.287      169.702      147.643
07    6204.331     8649.858     9422.517     9630.657     9670.910
08    1715.582      709.289      322.957      219.548      203.512
09     171.953        6.947         .642         .224         .217
10      53.492         .624         .022         .004         .003
11     161.385        5.667         .386         .078         .036
12    1652.636      625.446      236.886      117.961       72.706
13    6286.847     8743.351     9512.785     9739.939     9819.327
14    1864.443      632.646      251.442      140.433      103.264
15      37.525         .185         .003         .000         .000
16        .148         .000         .000         .000         .000
17        .000         .000         .000         .000         .000
18        .000         .000         .000         .000         .000
19        .000         .000         .000         .000         .000
20        .000         .000         .000         .000         .000
===================================================================
Chi2 1.951E+01   2.070E+00    7.952E-01    6.943E-01    6.847E-01
```

```
                    results of MLP8 routine

iter      100         200         300         400         500
=====================================================================
01         .000        .000        .000        .000        .000
02         .005        .000        .000        .000        .000
03         .238        .000        .000        .000        .000
04       13.528        .462        .022        .000        .000
05      378.978      97.669      28.107       8.985       3.140
06     2709.308    2458.622    2098.401    1786.858    1538.081
07     4063.127    5230.988    6065.067    6705.949    7209.212
08     1897.131    1715.630    1506.199    1294.191    1096.948
09      656.024     335.175     185.728     106.463      62.592
10      424.717     169.527      77.083      37.332      18.967
11      750.998     409.154     241.158     148.237      94.448
12     2216.747    2137.504    1995.975    1842.158    1698.402
13     4180.226    5296.818    6051.810    6611.117    7038.891
14     2393.626    2071.019    1732.901    1459.564    1246.274
15      347.822     109.536      43.203      20.345      11.121
16       17.264       1.367        .204        .047        .016
17         .488        .009        .000        .000        .000
18         .013        .000        .000        .000        .000
19         .000        .000        .000        .000        .000
20         .000        .000        .000        .000        .000
=====================================================================
Chi2 3.436E+00   1.700E+00   1.158E+00   9.144E-01   7.863E-01
```