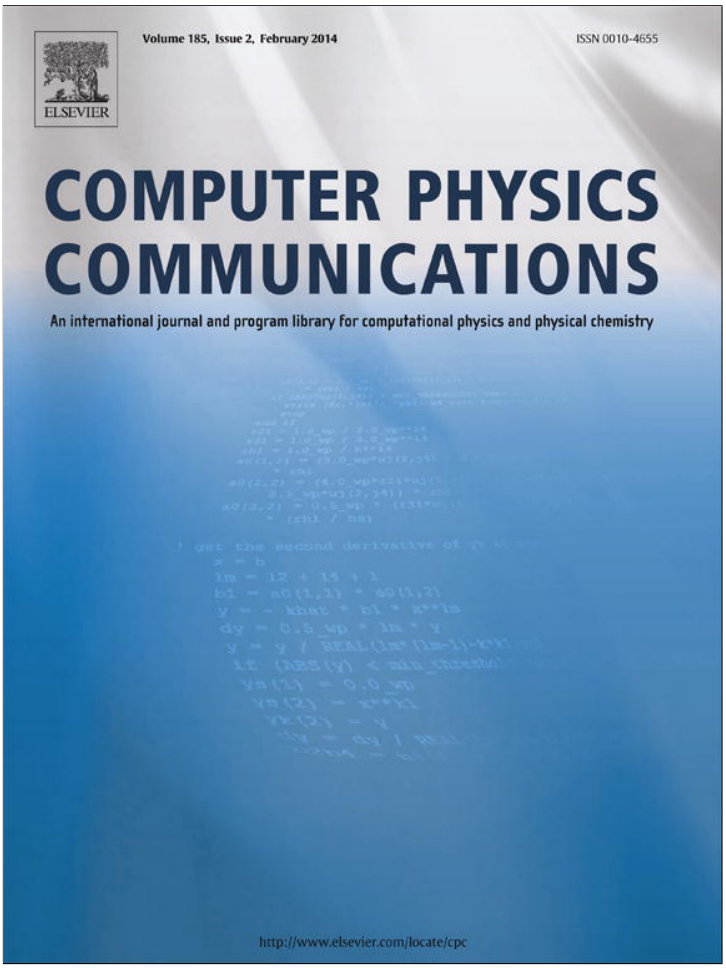

Computer Physics Communications 185 (2014) 599–603

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Development of the FUMILI minimization package☆

I.M. Sitnik*

Joint Institute for Nuclear Research, Russian Federation

ARTICLE INFO

Article history:
Received 16 April 2013
Received in revised form
26 July 2013
Accepted 1 August 2013
Available online 24 October 2013

Keywords:
Minimization
Fit
Least square method

ABSTRACT

The suggested FUMILI package development has the following advantages: a more convenient and friendly user interface; no limit on the number of parameters and experimental points; speed advantage when the number of parameters is high enough; there are options to ignore wrong experimental points and correct experimental errors. The preliminary scan is envisaged for complicated tasks.

All programs are written in FORTRAN-77.

The investigation has been performed at the Veksler and Baldin Laboratory of High Energy Physics, JINR.

Program summary

Program title: FUMILIM
Catalogue identifier: AEQF_v1_0
Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEQF_v1_0.html
Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland
Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html
No. of lines in distributed program, including test data, etc.: 17291
No. of bytes in distributed program, including test data, etc.: 346409
Distribution format: tar.gz
Programming language: Fortran-77.
Computer: Any computer with a Fortran 77 compiler.
Operating system: Any system capable of running Fortran 77 executables.
RAM: 500000 bytes
Classification: 4.9.

Nature of problem:
To minimize χ^2-functional.

Solution method:
Conjugate gradient method.

Running time:
From milliseconds to hours. The test runs provided only take seconds to complete.

1. Introduction

This work has been done in the memory of outstanding mathematician Prof. I.N. Silin. His famous program FUMILI [1–4] (FI) was the first one in the early sixties to provide a very advanced minimization method which gave physicists an effective tool to interpret experimental data. The so-called conjugate gradient method [5] was used in the program to find the functional minimum. The key algorithm of the error matrix transformation was several times improved by him. The last changes were done in 2002. On the other hand, the user interface, which was fixed in sixties as a beta version, was kept unchanged. The new user interface, suggested in this paper, had been previously in part discussed by the author with I.N. Silin, who supported it. The main events in this field are highlighted in Section 2.

The new features of the package, called FUMILIM (FM), are described in Section 3. The greatest attention in this work was paid to the problem, related to a wide class of tasks, when the number of parameters essentially exceeds 100, which is declared as the maximum for FI. The acceleration algorithm was elaborated for such tasks.

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Tel.: +7 49621 21282.
E-mail addresses: sitnik@dubna.ru, sitnik@jinr.ru.

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.08.001

Also the intelligent procedure of fitting data ignoring wrong experimental points and the experimental error correction is considered in this section. The preliminary scan procedure is suggested for ambiguous tasks.

The beta versions of this package were described in [6].

2. User's interface

2.1. User's function

The program searches for the minimum of the functional

$$\sum_{i=1}^{n} \chi_i^2, \tag{1}$$

where n is a number of experimental points,

$$\chi_i = \frac{F(x_i, a, F'_a) - Y_i}{\sigma_i}. \tag{2}$$

Here Y_i and σ_i are the experimental value and error, respectively. $F(x, a, F'_a)$ is the user's function (UF), where x_i is the argument array, a is the adjusted parameter array and F'_a is the array of partial analytical derivatives, which can be defined in UF. Y_i, σ_i, x_i are the data of the experimental point.

By default, the following structure of the experimental point is implicit:

$$Y_i, \sigma_i, x_i(1)[, x_i(2) \ldots]. \tag{3}$$

The x-array is transmitted to UF. Sometimes it is more convenient, when the user completely defines the data structure of the experimental point. If this option is ordered, all the array of experimental point data is transmitted to UF and the value of χ_i (Eq. (2)) must be formed within UF.

Now UF should be referred in the program call. That makes, in particular, the use of several fits with different UF in one task easier. In each UF an arbitrary number of partial derivatives with respect to the parameters may be defined analytically, that is difficult while using FI.

2.2. Parameter step restriction

To fit by means of FI, the user should define the parameter step restriction pl, that is the maximum allowed parameter increment during one iteration. Sometimes this is not an easy task. If the functional F is strictly linear in all parameters, pl can be arbitrarily large. Too small pl leads to an excessive number of iterations. If F is non-linear in some parameters, and pl is sufficiently large, the fit can go in the wrong direction.

In FM there is an opportunity not to define these values. The adjusting of the parameter step restriction is based on the comparison of the first and second averaged functional derivatives with respect to the parameters. The first partial derivatives, if they are not defined in UF, are calculated in the program by the following formula:

$$F'_{a_j} = \frac{\partial F}{\partial a_j} = \frac{F[\ldots(a_j + s_j), a_{j+1}\ldots] - F[\ldots a_j, a_{j+1}\ldots]}{s_j} = \frac{\delta_j}{s_j}, \tag{4}$$

where j, a_j and s_j are parameter number, value and augment, respectively. pl is related to s as $pl = 100s$ (as in FI). A more accurate relation between δ and derivatives can be written by using the Taylor expansion:

$$\delta = F's + \frac{1}{2}F''s^2 + \cdots. \tag{5}$$

To find the both derivatives, two calls of UF with different values of s are necessary. In FM $s_1 = s$ and $s_2 = 2s$ are used. In this case we have

$$F' = \frac{4\delta_s - \delta_{2s}}{2s}, \qquad F'' = \frac{\delta_{2s} - 2\delta_s}{s^2}. \tag{6}$$

In the case when F' is defined by the user, we have

$$F'' = \frac{2(\delta_s - F's)}{s^2}. \tag{7}$$

Assuming that the second term in Eq. (5) should contribute to the value of δ less than 0.5%, we have the following equation for the optimal value of s:

$$s' = \frac{0.01\langle F'\rangle}{\langle F''\rangle}, \tag{8}$$

where $\langle F'\rangle$, $\langle F''\rangle$ are the averaged values of derivatives over all experimental points. The start value of s is defined from the start values or bounds of initial parameters. If nothing is defined, $s = 0.01$.

As a rule, FM finds the optimal start values of pl better than the user.

The time necessary for this operation is equal approximately to the time of one regular iteration. In the case of a multiple call to FM (during event reconstruction, for example) this time loss becomes negligible, because the found values of pl are saved to the initial data array during the first call.

2.3. Output

FM starts with the verification of the initial conditions of the task. User error at the start of the task with a fit is a typical phenomenon. This diagnostics helps to save time when debugging. As a part of the verification, the brief analysis of the experimental point array is fulfilled. It is accepted that an experimental error should have a value within $(10^{-12}, 10^{12})$.

Apart from the standard output (like in FI), whose example is shown in Fig. 1, there is an opportunity to record a number of files to clear up the situation with the ambiguous fit or to provide the data convenient for graphic representation. The example of using the function table, describing the data, is shown in Fig. 2.

All possibilities for the extra output are listed in the user guide.

2.4. Main difference with FUMILI

The basic code of FI(the error matrix transform) was kept unchanged.

FI calculates the correlation coefficient of the j-th parameter with all the others. If it is not correlated with any other parameter, the correlation coefficient $R_j = 1$. Large values of R_j mean that there is no unique solution for the (UF). As a rule, the main contribution to $R_j \gg 1$ is provided by one of the pair correlations, r_{ij}. When we have a number of $R_j \gg 1$, the task to isolate the pairs becomes very unclear. FM calculates only pair correlations, the largest of them at the given j are placed into column ''correlation coef''., together with the parameter number i, which is responsible for this correlation.

The pair correlation is calculated by the following formula:

$$r_{ij} = \frac{M_{ij}}{\sqrt{M_{ii}M_{jj}}},$$

where $\|M\|$—an error matrix. r_{ij} satisfies Bunyakowsky's inequality:

$$-1 \le r_{ij} \le 1.$$

At the absence of correlations $r_{ij} = 0$. Lines with $|r_{ij}| > 0.9$ are marked, as it is seen in Fig. 1. There is an opportunity to see all pair correlations, described in the user guide.

```
Fit   3   by FUMILIM
  *No Initial Data File
Exp.Points:      1000
Free Param.:     7, User Defined Derivatives:     4
sets:     2,  Free Param./set:     4

Adjusted Parameter Step Restriction

Iter=  0: hi^2/NDF=1.98762D+10  hi^2=1.97570D+13  d(hi^2)/hi^2=-1.000D+00    Kapp=4.4D+06  Lamb=1.0D+00

     Parameter           Parameter        Standard      Parameter                  Max.Pair
     Number & Name       Value            Deviation     Increment                  Correl.Coef.

     1 common----------   1.00000D-03  --  2.27058D-07   0.00000D+00 (  0.0%) --    0.048    2
     2 a01_set001         1.00000D-03      1.38034D-03   0.00000D+00 (  0.0%)      0.870    3
     3 a02_set001-----    1.00000D-03  --  6.35168D-03   0.00000D+00 (  0.0%) --    0.970    4  >0.9
     4 a03_set001         1.00000D-03      6.18902D-03   0.00000D+00 (  0.0%)      0.970    3  >0.9
     5 a01_set002-----    1.00000D-03  --  1.40563D-03   0.00000D+00 (  0.0%) --    0.878    6
     6 a02_set002         1.00000D-03      6.29806D-03   0.00000D+00 (  0.0%)      0.968    7  >0.9
     7 a03_set002-----    1.00000D-03  --  6.06649D-03   0.00000D+00 (  0.0%) --    0.968    6  >0.9

Iter=  1: hi^2/NDF=1.00800D+00  hi^2=1.00195D+03  d(hi^2)/hi^2=-1.000D+00    Kapp=8.2D-09  Lamb=1.0D+00

     Parameter           Parameter        Standard      Parameter                  Max.Pair
     Number & Name       Value            Deviation     Increment                  Correl.Coef.

     1 common----------   1.00000D+00  --  2.27058D-07   9.99000D-01 ( 99.9%) --    0.048    2
     2 a01_set001         9.98347D-01      1.38034D-03   9.97347D-01 ( 99.9%)      0.870    3
     3 a02_set001-----    1.00326D+00  --  6.35168D-03   1.00226D+00 ( 99.9%) --    0.970    4  >0.9
     4 a03_set001         1.00069D+00      6.18902D-03   9.99686D-01 ( 99.9%)      0.970    3  >0.9
     5 a01_set002-----    9.99170D-01  --  1.40363D-03   9.98170D-01 ( 99.9%) --    0.878    6
     6 a02_set002         1.00370D+00      6.29806D-03   1.00270D+00 ( 99.9%)      0.968    7  >0.9
     7 a03_set002-----    9.96445D-01  --  6.06649D-03   9.95445D-01 ( 99.9%) --    0.968    6  >0.9

Iend= 1: Full Success                                   hi^2/NDF change: 1.98762D+10 -> 1.00800D+00

Exp.points,(weigths),<hi^2> change for each set
 All      1000    1.97570E+10 -> 1.00195E+00
 + over twice larger, - under twice less, than average

   1       482    4.09851E+10 -> 1.06437E+00
   2       518    4.16023E+06 -> 9.43860E-01
```

Fig. 1. The example of screen (or OUT-file) output.

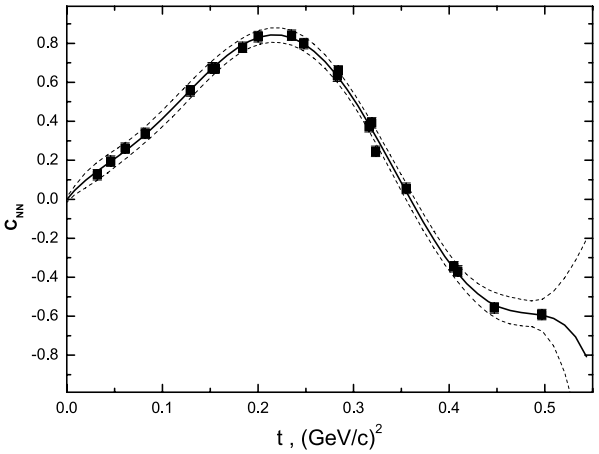

Fig. 2. Polynomial fit of data [7] with the calculation of the confidence band.

3. New features

3.1. Speed optimization

3.1.1. Problems with a large number of parameters

There is a wide range of tasks with a huge number of parameters, when experimental data are subdivided into a large number of sets, having a number of common parameters for the task (N_0) and also individual parameters for each set (N_1). Thus, we have

$$N_T = N_0 + N_1 N_s, \tag{9}$$

where N_T is the total number of parameters, and N_s is the number of sets. When a user faces the problem of adjusting big experimental setup offsets, such (or more complicated) a combination arises frequently with N_s large enough. In the task of adjusting straw offsets [8] the total number of parameters and experimental points achieved two hundreds and about 100 thousands, respectively.

Another example of a multi-set task is a study of the planet magnetic field with different satellites [9]. In perspective, several hundred sets of data, each about 500 experimental points, are expected for several planets.

The most time-consuming part of the program is accumulation of the error matrix [10]. The relevant sub-program has a cycle over all events. If the derivatives are not defined in UF, then $N_T + 1$ calls to UF are necessary for each experimental point. The first call is with the current settings, and then we need to make N_t calls with the parameter augments to find the partial derivatives with respect to the parameters. When N_s is large, the overwhelming majority of calls produces $F'_{a_j} = 0$. Each of these calls requires the same amount of time, as an effective call. To improve the efficiency in this segment of the program, FM performs a preliminary iteration, which analyzes the last arguments in the array of experimental points. If the user defines them as a set number, the program adjusts itself to make only (N_0+N_1+1) UF calls at each experimental point. This approach provides the following speed-up:

$$R = \frac{N_0 + N_1 + 1}{N_T + 1}.$$

The comparison of FI and FM speed for the tasks of this class is illustrated in Fig. 3. In this example N_0, N_1 are the same for all the sets and the enlargement of the total number of parameters is achieved by increasing N_s. The polynomial UF was used in this comparison. When UF is time consuming, the ratio has the dependence as for 500 events/set with the polynomial.

The solution time of the task [8] is about 20 s with the single 2.8 GHz processor.

If the set contains subsets, it is offered to construct the last argument of an experimental point as follows:

$$M = (I - 1)K + J,$$

where I—the set number, J—the subset number, and K must not be less than $J_{\max}$.

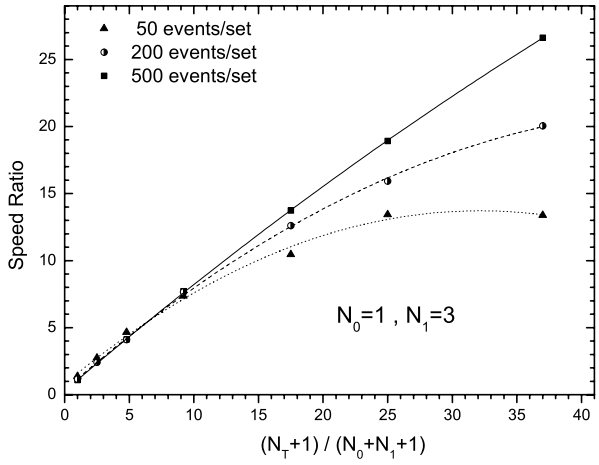

Fig. 3. Speed comparison for FI and FM using polynomial UF.

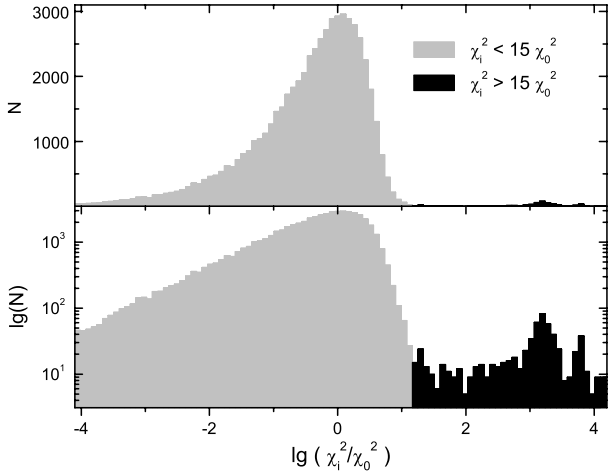

Fig. 4. Right/wrong point selection in linear and logarithmic scales for the task of adjusting of straw offsets [8]. This figure is prepared by using FM service.

3.1.2. Event reconstruction

When FM is involved into the event reconstruction, the multiple call with the option "no print" is usually applied. When this option is ordered, the program analyzes initial conditions of the task during the first call. If they satisfy certain conditions (see the user guide), the program starting from the second call transits to the optimized for the speed algorithm (no verification of the initial conditions, no adjustment, etc.). It is assumed that the same array of initial parameter values is referred in each call. The extra optimization is applied for two-parameter tasks (track reconstruction).

In the latter case (depending on whether the derivatives are defined or not by the user) the speed advantage in comparison with FI is about a factor of 3–6. The solution time with mentioned above processor is $300 \cdot 10^3$ (defined derivatives) and $150 \cdot 10^3$ tracks per second for the polynomial UF.

3.2. Fit ignoring wrong points

It is widely spread to use a double fit procedure, when the second fit is fulfilled after removing the wrong experimental points in a manual mode. When adjusting physical detector offsets, one deals with a huge number of events, and wrong points form statistically meaningful tail of the χ^2-distribution (Eq. (2)), as it is seen in Fig. 4.

Frequently, the second fit gives the parameter values, which differ meaningfully from those in the previous fit, and they are more correct. FM allows one to fulfill the fit ignoring wrong points (IWP) by using one call. After the standard fit as the first step, we know $\chi_0^2 = \chi^2/n$, where n is the number of experimental points. In subsequent fits the experimental points with $\chi_i^2 > \chi_m^2$, where

$$\chi_m^2 = C\chi_0^2, \tag{10}$$

are ignored. The default value $C = 15$ can be redefined by the user.

It is seen from Fig. 5 that the IWP fit in comparison with standard one gives the much better agreement between regular experimental points and theory in vicinity of the ambiguous points.

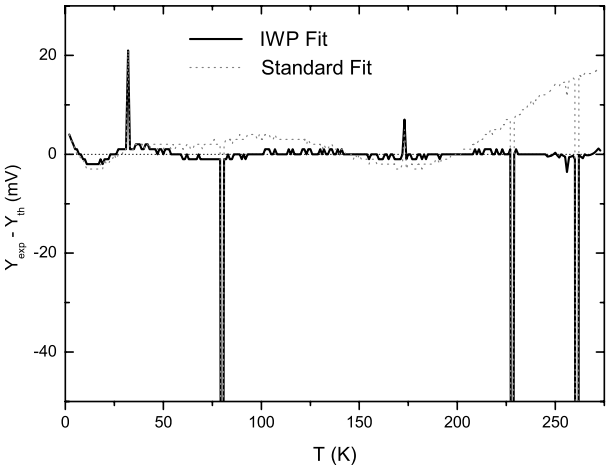

Fig. 5. Differences between experimental and fitted values of thermocouple calibration for standard and IWP fits. This figure is prepared by using FM service.

While removing the wrong points in the manual mode in the thermocouple task is possible (6 out of 272 wrong points), this approach is hardly realistic for the straw task ($7 \cdot 10^2/6 \cdot 10^4$).

In the standard fit the convergence condition (as in FI),

$$\kappa = \max\left[\frac{|da_j|}{\delta(a_j)}\right]_{j=1,N_t} < \varepsilon, \tag{11}$$

is a signal to finish the job. Here $\varepsilon = 0.01$, da_j, $\delta(a_j)$ are the parameter increment and confidence band, obtained in the current iteration. This condition is checked at the end of each iteration.

In the considered mode the condition to finish the job is

$$\kappa < \varepsilon \wedge \frac{\chi_s^2}{\chi^2} < 1.05, \tag{12}$$

where χ_s is the value of χ at the start of the iteration cycle. The condition (11) now is only a signal to redefine (always to reduce) χ_m by using Eq. (10). Thus, a sequence of iteration cycles with decreasing values of χ_m^2 may occur. For example, in the thermocouple task during the first iteration cycle only three worst points were ignored. These points are easily distinguishable in Fig. 5. The other three points were ignored during the next two iteration cycles.

In such tasks, as a polynomial fit, the points with extreme values of the argument are at the increased risk to be interpreted as wrong. The user can forbid ignoring of these points.

3.3. Unequal accuracy measurements

When one deals with several sets of measurements of the same observable, he/she can face with the unequal accuracy of different sets of data. An example of such a situation took place while measuring the Mercury magnetic field by means of different satellites [9]. The χ^2-dispersion for 15 sets of these data before and after error alignment is shown in Fig. 6.

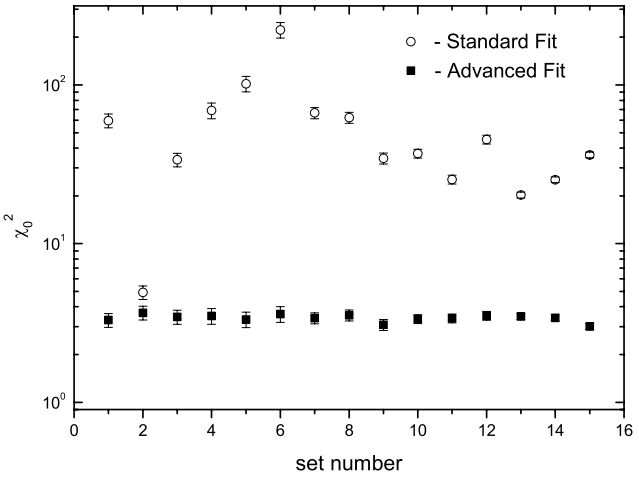

Fig. 6. χ^2_0 values for different sets of data [9] before and after the error-correction procedure.

The solution of this problem is as follows. After the standard fit as the first step, the averaged values of χ^2 for i-th set, $\chi^2_{0,i}$ become known. During the second fit each experimental point, belonging to the i-th set, is taken into account with the weight of $\chi^2_{0,m}/\chi^2_{0,i}$, where $\chi^2_{0,m}$ is the least of $\chi^2_{0,i}$.

The joint `IWP` and error alignment fit were applied in this task. The difference between the standard and advanced modes of the fit achieved 30% for some parameter values.

3.4. Preliminary scan

When F has a number of local minima and the initial parameter values are not evident, scanning to find a preliminary global minimum is envisaged. This procedure needs the time proportional to $m^k n$, where m is a number of scanned parameters, k is a number of values for each scanned parameter, and n is a number of experimental points.

To speed up this very time-consuming procedure, the following steps are provided. The first of them is the restriction of the number of used experimental points: $n_u \ll n$. In this case the taken experimental points are distributed evenly in the experimental data array with interval n/n_u. $n_u = 100$ per set is the default value, which can be redefined by the user.

Also, the prospect of the current combination of parameter values is evaluated using only several experimental points ($n_{\text{few}} = 0.1n_u$). The set of the parameter values is treated as unpromising when $\chi^2_{0,\text{few}} > 9\chi^2_{0,u}$. Here $\chi^2_{0,u}$ is the minimum value, achieved at the moment. For the start value of χ^2_u the combination with the central values of all scanned parameters is used. This approach allows one to speed up the scan procedure by about 10 times.

The scanned parameters are regularized in the descending order along gradient values:

$$G_j = \sum_{i=1}^{n} \frac{F'_{a_j}}{\sigma_i} \chi_i, \tag{13}$$

where j is the parameter number. Other denotations are defined in Eqs. (1), (2). The parameter with the highest gradient value is scanned first.

When the number of candidates to scan (defined in the user guide) is too large to solve the task for a reasonable time, there is an opportunity to restrict the number of the scanned parameters. In this case the parameters with the lowest gradient values will be excluded from scanning. The small gradient value means that the parameter value is close to optimal or F (Eq. (2)) has a weak dependence on this parameter.

By default, 3 values for each parameter are scanned. 5 loops are envisaged. After each loop the interval of parameter values is shrunk. This mode has the speed advantage in comparison with more scanning values for each parameter, but ambiguous solution is possible, when a strong correlation between parameters takes place. The number of scanning values for each parameter and the number of iteration loops can be redefined by the user.

The preliminary scan proved to be necessary in the task [9].

4. Summary

There is a wide class of many parameter time-consuming tasks which the suggested package solves much faster in comparison with existing minimization packages. The package has also a distinct speed advantage in tasks typical for the event reconstruction.

The set of options allows one to use one call instead of complicated many-step procedures with removing the ambiguous experimental points and error alignment for different sets of data, necessary when the standard call to the minimization program is used.

The scanning procedure incorporated into the package makes it possible to extend the class of solvable tasks in comparison with the `FUMILI` package.

The modern user interface is elaborated for this package.

Acknowledgments

The author is grateful to I.I. Alexeev for collaboration in the task concerning the study of the Mercury magnetic field, to G.A. Ososkov, A.P. Sapozhnikov and S.V. Chubakova for fruitful discussions, J. Musinsky for testing this package under different compilers and to T.F. Sapozhnikova for help in spreading out this package.

References

[1] S.N. Sokolov, I.N. Silin, JINR Preprint D-810, Dubna, 1961.
[2] I.N. Silin, CERN Program Library, D510, FUMILI, 1983.
[3] A.J. Ketikian, E.V. Komissarov, V.S. Kurbatov, I.N. Silin, Nucl. Instrum. Methods Phys. Res. A 314 (1992) 578.
[4] V.S. Kurbatov, I.N. Silin, Nucl. Instrum. Methods Phys. Res. A 345 (1994) 346.
[5] M.R. Hestenes, E. Stiefel, J. Res. Natl. Bur. Stand. 49 (1952) 409. http://en.wikipedia.org/wiki/Conjugate_gradient_method.
[6] I.M. Sitnik, JINR Preprints E11-2008-43 and E11-2010-101.
[7] V. Ghazikhanian, et al., Phys. Rev. C 43 (1991) 1532.
[8] I.M. Sitnik, et al., Phys. Rev. C 84 (2011) 034006.
[9] I.I. Alexeev, et al., Icarus 209 (2010) 23.
[10] A.P. Sapozhnikov, http://wwwinfo.jinr.ru/programs/publ/pfumili.htm.