

E11-2008-43

I. M. Sitnik\*

**MODIFICATION OF THE FUMILI  
MINIMIZATION PACKAGE**

In memory of Prof. I. N. Silin

Submitted to «Computer Physics Communications»

---

\*E-mail: [sitnik@sunhe.jinr.ru](mailto:sitnik@sunhe.jinr.ru)

Ситник И. М.

E11-2008-43

Модификация пакета программ минимизации FUMILI.

Памяти проф. И. Н. Силина

Предложена модификация пакета FUMILI, особенностями которой являются более удобный и дружелюбный пользовательский интерфейс, отсутствие ограничений на число параметров и экспериментальных точек, преимущество в скорости работы при большом числе параметров, возможность в функции пользователя определять аналитически произвольное число производных по параметрам. Предлагается также дополнительный пакет программ, включающий реконструкцию треков в дрейфовых трубках. Все программы написаны на FORTRAN-90.

Работа выполнена в Лаборатории высоких энергий им. В. И. Векслера и А. М. Балдина ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна, 2008

Перевод авторов

Sitnik I. M.

E11-2008-43

Modification of the FUMILI Minimization Package.

In memory of Prof. I. N. Silin

FUMILI renovation is suggested which allows the following: a more convenient and friendly user interface; no restriction on the number of parameters and experimental points; speed advantage when the number of parameters is high enough; a possibility to define analytically an arbitrary number of parameter derivatives. An additional package of subroutines, including track reconstruction in the straw tube chambers, is suggested. All programmes are written using FORTRAN-90.

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

Preprint of the Joint Institute for Nuclear Research. Dubna, 2008

## INTRODUCTION

This work has been done in memory of outstanding mathematician Prof. I.N.Silin, who went out on 21 November 2006. His famous program FUMILI [1–4] was the first one in the early sixties to provide the very advanced minimization method which gave physicists a quite effective tool to solve many sophisticated non-linear problems. 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 the sixties as beta-version, was kept unchangeable. The new user interface, suggested in this paper, was previously discussed between me and I.N.Silin in part and was supported by him, but at that time he was involved completely in other problems, such as constrained data fitting [5], or the combination of Monte-Carlo and Korobov's methods [6].

One more problem, related to the wide class of tasks, arises when the number of parameters exceeds essentially 100, which is declared as the maximum for FUMILI. Let us consider the simplest example of a source of the huge number of parameters:

$$N_T = N_0 + N_1 N_s, \quad (1)$$

where  $N_T$  is the total number of parameters,  $N_0$  is the number of parameters common for all events to be handled,  $N_1$  is the number of parameters common for some subgroup,  $N_s$  is the number of subgroups. When a user faces the problem of adjusting big experimental setup offsets, such a combination (or more complicated one) arises frequently with large enough  $N_s$ . In the author's practice the total number of parameters and events achieved several hundreds and about 2 millions, respectively. The last librarian version of the source program allows one to have only 70 free parameters, and only an advanced user can improve the situation, if he has the FORTRAN source. The matter is, in this case it is necessary to redefine not only user COMMON-blocks, but also several COMMON-blocks for only inner use. On the other hand, the huge length of COMMON-blocks (especially for error matrixes, with the size of about  $N_T^2$  elements) can be inconvenient for tasks with the small number of parameters. Therefore, all arrays in the new program, declared earlier in COMMON-blocks, have been replaced by adjustable ones. To compensate the unavoidable slowing down of the program in this case,

the programming style has been changed proceeding from the fact that one has the unlimited address-space. Recall that the memory of the USSR computer leader of the sixties, BESM-6, had about 192 kbytes for user tasks, and the FUMILI original was written with a special care of computer memory, often to the speed detriment. The basic code of FUMILI (the error matrix transform) during this optimization was kept unchangeable. The renewed version of FUMILI is named FUMILIM.

Below we assume that User is the name.

## 1. USER GUIDE

**1.1. FUMILI.** All programs described below use «**IMPLICIT REAL\*8(A-H,O-Z)**».

For better understanding of the new program, let us recall the user guide for FUMILI. The call has the form

```
CALL FUMILI(S,NPAR,N1,N2,NIT,EPS,AKAP,ALAM,IPR,IEND).
```

Here **input parameters:**

NPAR — the number of parameters,  
N1=2, N2=1 (recommended values),  
NIT — restriction for the number of iterations,  
EPS — desirable accuracy (recommended values are 0.01–0.1),  
IPR — to print each  $i$ th iteration (IPR < 0 — no print);

**output parameters:**

S= $\chi^2/2$ ,  
AKAP, ALAM, NC — parameters characterizing the iteration convergence,  
IEND — flag of conditions of the exit from the program.

An array of experimental points should be placed at

```
COMMON/EXDA/EXDA(LEXD,NEXD),
```

where LEXD and NEXD are the number of words in each experimental point and the number of experimental points, respectively. These values should be defined in

```
COMMON/NED/NED(2)
```

as NED(1)=NEXD and NED(2)=LEXD. The content of one experimental point is the following:

EXDA(1,I) —  $y_i$ -value of an experimental point;  
EXDA(2,I) —  $\sigma_i$ , its error;  
(3:LEXD,I) —  $x_i$ -array of an experimental point.

Initial values of parameters (which will be replaced after the fit by final ones), and possible restrictions for them should be defined in

```
COMMON/A/A(NP)/AL/AMIN(NP)/AU/AMAX(NP),
```

respectively. Here should be  $\text{NPAR} \leq \text{NP} \leq 100$ . User should also define restrictions for parameter steps during one iteration in

```
COMMON/PL/PL(NP).
```

If they are not defined (equal to 0.d0) or negative, the corresponding parameters will be treated as fixed. After fit, the parameter errors and correlation factors are placed in

```
COMMON/SIGMA/SIGMA(NP) and  
COMMON/R/R(NP), respectively.
```

User should write the function FUNCT(X), depending on parameters A(1:NPAR) (the corresponding common-block should be declared in this function). Here X (either a scalar or an array) is arguments of experimental points. The numerical computation of derivatives is used in this case.

If User prefers to define also parameter derivatives, he should write the subroutine ARITHM(Y), which will replace the program with the same name in the librarian package. The derivatives array should be declared in

```
COMMON/DF/DF(NEXD).
```

In some linkers conflicts between User's and librarian subroutines are possible. **Here we have the following alternative: either no one (FUNCT) or all derivatives (ARITHM) are to be defined.**

**1.2. FUMILIM.** The call has now the form

```
H=FUMILIM(A,NAM,NPAR,EXDA,LEXD,NEXD,USFUN,NIT,LUN).
```

Here (input parameters only) NPAR and NIT are the same as in the FUMILI call, the sense of LEXD and NEXD are described above, A(NPAR,1:6) — array of initial (and final) values for parameters. The equivalence between old and new arrays is the following:

A(J,1) – A(J);	A(J,4) – PL(J);
A(J,2) – SIGMA(J);	A(J,5) – AMIN(J);
A(J,3) – R(J) (correlation factor);	A(J,6) – AMAX(J).

NAM(NPAR) — character array of the parameter names (up to 15 symbols) (this idea is taken from the famous MINUIT program [7]);

EXDA(LEXD,NEXD) — array of experimental points, having the structure described above;

USFUN(X,AF,DF) — User's function for data fit (should be declared via EXTERNAL in the calling program);

X,AF,DF — argument(s), parameters and their derivatives, respectively;

(AF(1:NPAR) — array in USFUN is equivalent to A(1:NPAR,1)-slice in the main program),

LUN — the output unit number (lun = 0, 6 means the screen output, other numbers should be associated with file names). Output is switched off when LUN < 0.

The value of the function is  $\chi^2/NDF$  (Number of Degrees of Freedom).

If User does not want to deal with the parameter names, he can write

H=FUMILIM(A,'\*',NPAR,EXDA,LEXD,NEXD,USFUN,NIT,LUN).

**The suggested user interface simplifies programming appreciably when several FUMILIM calls are used in the main program** (the new or old A- or EXDA-arrays can be used, the different USFUN can be written instead of complicated bifurcations in the only possible FUNCT in the FUMILI case).

**User is free to define any number of parameter derivatives in USFUN.**

In the new program initial parameter restrictions are  $\pm$ huge values, exactly as in the old one. The difference is: in FUMILI these values are defined via DATA-operator, but in FUMILIM the A(J,5)- or A(J,6)-values, equal to 0.d0, are considered as undefined, and the redefinition (huge values) follows in this case.

The treatment of PL(I) is slightly changed. Now A(I,4)==0.d0 is considered as undefined value, and the redefinition follows (rather large value). The advantage of such an approach is the following. **If, for example, USFUN is linear with respect to AF(J)  $\left(\frac{\partial F(x, A)}{\partial A(J)} = f(x)\right)$ , the definition of A(J,4) is rather harmful than senseless (the same result may be with the larger number of iterations). In particular, if USFUN is linear with respect to all parameters, User is free to define nothing in the A(1:NPAR,1:6)-array. And a task will be solved during one iteration.** The definition of A(J,4) is hardly useful, when the corresponding first derivative is defined analytically. There are many other cases, when User is free from definition of A(J,4). So, it is recommended not to define A(1:NPAR,4) always when starting the fit. And only if the result is suspicious, one can start to play with only those A(J,4) for which USFUN is not linear with respect to corresponding AF(J), and, simultaneously, DF(J) is not defined analytically.

One can see that the input parameters N1,N2,EPS from FUMILI now are not in the list of parameters. They are defined inside the program via DATA-operator by recommended values, but advanced User can redefine them in

COMMON/FUMINPUT/EPS, N1, N2, NPAROPT. (2)

The omitted output parameters AKAP,ALAM,IEND can be found in  
COMMON/FUMOUTPUT/HISQ,IEND,NITER,AKAPPA,ALAMBD.

Here  $HISQ=\chi^2$ , NITER is the number of real iterations, values of IEND are the following:

- 1 – MINIMIZATION IS FINISHED SUCCESSFULLY,
- 2 – MINIMIZATION IS FINISHED SUCCESSFULLY IN PART WHILE SOME PARAMETERS ARE FIXED ON BOUND,
- 3 – MINIMIZATION IS TERMINATED AS ALL PARAMETERS ARE FIXED ON BOUND,
- 4 – ITERATION LIMIT IS REACHED,
- 5 – MINIMIZATION IS FINISHED SUCCESSFULLY IN PART WHILE SOME PARAMETERS ARE UNDEFINED,
- 6 – MINIMIZATION IS TERMINATED AS NO FURTHER DECREASE IN CHISQ IS OBTAINABLE,
- 7 – ALL PARAMETERS ARE FIXED,
- 8 – MINIMIZATION IS TERMINATED AS ALL PARAMETERS EITHER FIXED ON BOUND OR HAVE INFINITE ERROR ESTIMATION.

The speed optimization is used, when parameter boundaries are completely absent, and the number of free parameters is greater than 15. This value can be redefined in NPAROPT (Eq. (2)). Speeding up is appreciable, when  $N_T \gg N_0 + N_1$  (Eq. (1)). The error matrix has many empty elements in this case. The author is not sure that this optimization is 100% correct. So, if results are suspicious, User should set up  $NPAROPT > NPAR$ .

Special kind of optimization is applied for  $NPAR=2$  and  $LUN < 0$  (typical for the track reconstruction). In such a case the control is delivered to FUMI2PAR. The direct call

H=FUMI2PAR(A,EXDA,LEXD,NEXD,USFUN,NIT)

is also possible. This program ignores parameter boundaries (A(J,5),A(J,6)).

FUMILIM starts to work from User's error check:

1. The number of experimental points should not be less than the number of parameters:  $NEXD \geq NPAR$ ;
2. Length of Block of one experimental point should be  $LEXD \geq 3$ ;
3.  $EXDA(2,I)$  (experimental error) should be positive;
4. Initial value of parameter should be within up- and down-limits for it:  $A(J,5) \leq A(J,1) \leq A(J,6)$ .

Error number 3 arises frequently when the real structure of EXDA-array does not match with its definition via LEXD,NEXD. Each negative check is **FATAL** for User's program. Error check is switched off, when  $LUN < 0$ .

Below is the recommendation for users who want to replace FUMILI by FUMILIM, but have many old programs. The sequence of such lines as

```
COMMON/A/A(100)/PL/PL(100)/AU/AMX(100)/AL/AMN(100)
COMMON/NED/NED(2)
NED(1)=250
NED(2)=4
NPAR=20
eps=0.01D0
N1=2
N2=1
NIT=25
IPR=1
...
CALL FUMILI(S,NPAR,N1,N2,NIT,EPS,AKAP,ALAM,IPR,IEND)
```

can be replaced by

```
PARAMETER(NPAR=20,NED1=250,NED2=4)
COMMON/A/A(NPAR),SIG(NPAR),R(NPAR),PL(NPAR),AMN(NPAR),
      AMX(NPAR)
EXTERNAL USFUN
NIT=25
...
H=FUMILIM(A,'*',NPAR,EXDA,NED2,NED1,USFUN,NIT,6).
```

In this case one can keep unchangeable all definitions of A(J), PL(J), AMN(J) and AMX(J) in the old program.

**1.3. FUMIDOUB.** A procedure is widely spread such as the double fit, when the second step is fulfilled after removing so-called «bad points». When adjusting physical detectors offsets, one deals with the huge number of events and «bad points» form statistically meaningful wings of the  $\chi$ -distribution, where

$$\chi_i = \frac{y_i - Y(x_i)}{\sigma_i}.$$

Here  $Y(x_i)$  is the value of fitted function for  $x_i$  arguments of the  $i$ th experimental point. The advanced method of «bad points» removing is used in FUMIDOUB, which gives much better results in comparison with the simplest criterium of  $\chi_i^2 < C$ . The algorithm is described in Section 2. The FUMIDOUB call is identical with the FUMILIM call.

The work of FUMIDOUB is organized as follows. Firstly, the ordinary FUMILIM call is fulfilled. Then, the parameters  $\chi_0(j), \sigma(j)$  of the  $\chi$ -distribution are calculated for each AF(J)-parameter, and the new EXDA-array is formed (re-



placing initial one) with the condition  $|\chi_i - \chi_0(j)| < k\sigma(j)$ , for each  $j$  for which  $\partial Y(x_i)/\partial A(J) \neq 0$ . The value of  $k = 2.5$  is defined in COMMON/FDOUB/k via DATA-operator and can be redefined by User. Then the second FUMILIM call is fulfilled with the redefined value of NEXD. If one wishes to repeat FUMIDOUB-procedure, he can write one more line, identical to the first one.

**1.4. FUMSTRAW.** The track reconstruction in the straw tube chambers, where  $x = A * z + B$ , is not linear with respect to  $A$  and  $B$  parameters. So, there is a large probability to find the local minimum. To find the global minimum, four sets of initial parameter values are analyzed first in FUMSTRAW (one-iteration FUMI2PAR call for each set), then for the set, provided the least  $\chi^2$ , the FUMI2PAR call with rather large iteration limit is being done. But, as a rule (99%), the convergence is achieved during 0–1 iterations. Four possible sets of parameters for two tubes with the smallest and the largest  $z$ -coordinates are used as initial conditions. The call have the form

$$H=FUMSTRAW(A,B,EXDA,NEXD),$$

where EXDA is input array of experimental points:

- EXDA(1,I) — space radius (not time);
- EXDA(2,I) — radius error;
- EXDA(3,I) —  $z$ -coordinate of a wire (along the beam);
- EXDA(4,I) —  $x$ -coordinate of a wire (transversal to  $z$ ).

All values in EXDA should be in the same units. The ascending (descending) order of  $z$ -coordinates is assumed. At least, the tubes with the largest and the smallest  $z$ -coordinates should be the first and the last. Each FUMI2PAR call within FUMSTRAW is done with the reference to the user function, named FSTRAW, which is included in the FUMILIM-package.

**1.5. ERREXDA.** This program is an analog of the original program ERRORF. It calculates theoretical confidence band for each experimental point. The results, together with the experimental point data, will be written to the file associated with LUN. Here Silin's original algorithm is used. The call has the form

$$CALL ERREXDA(A,EXDA,USFUN,LUN),$$

where all parameters are described above.

**1.6. ERRFUNCT.** This program calculates the confidence band as a set of points, which allows one to plot data. The algorithm used is the same as in ERREXDA. The call has the form

$$CALL ERRFUNCT(ALIM,BLIM,N,A,EXDA,USFUN,LUN).$$

Here ALIM and BLIM are the lowest and the highest values of the argument, respectively,  $N$  is the number of points within  $\{ALIM,BLIM\}$ -interval. The other parameters are described above. It is assumed that experimental points have only

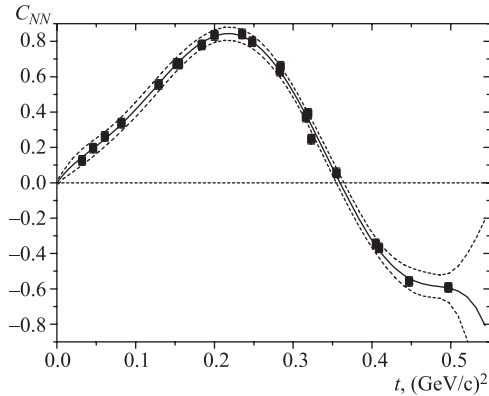


Fig. 1. Polynomial fit of data with calculation of the confidence band

one argument. The structure of the output array is the following:

$$x(1), y(1), y_{\min}(1), y_{\max}(1) \dots$$

The work of this program is illustrated in Fig. 1.

When the number of arguments is higher, the following program is applied:

CALL ERRFMULT(ALIM,BLIM,NDOT,A,EXDA,USFUN,IX,X,LUN).

Here IX is the argument number, which will be scanned. The values of other arguments should be defined by User in the X(1:(LEXD-2))-array.

## 2. CALCULATION OF DISTRIBUTION PARAMETERS FOR FUMIDOUB

The distribution parameters  $\chi_0$  and  $\sigma$  are defined as

$$\chi_0 = \frac{1}{\text{NEXD}} \sum \chi_i, \quad (3)$$

$$\sigma^2 = \frac{1}{\text{NEXD}^2} \sum \chi_i^2 - \chi_0^2. \quad (4)$$

Here

$$\chi_i = \frac{y_i - Y(x_i)}{\sigma_i},$$

$y_i = \text{EXDA}(1, I)$ ,  $\sigma_i = \text{EXDA}(2, I)$ ,  $Y(x_i)$  is the fitted function value. In the first approximation we use  $\chi_0 = 0$  and  $\sigma_0 = \sqrt{\frac{\chi^2}{\text{NDF}}}$ .

Then 100-bin histograms with edges  $\pm 10\sigma_0$  are filled by  $\chi_i$ -values. It is easy to see that the bin width in this case is  $0.2\sigma_0$ . In cases when  $N_T \gg N_0 + N_1$ , it

is useful to have the separate histogram for each parameter. That allows one to collect in histograms only events correlated with the considered parameter. The condition of the presence of the correlation is  $\partial Y(x_i)/\partial A(J) \neq 0$ .

To calculate the distribution parameters in the second approximation, only histogram bins close to the bin with the maximum value are collected. The condition is  $w_k > w_m$ , where  $w_m = w_{\max}e^{-2}$  (it is equivalent to the condition  $|\chi_k - \chi_0| < 2\sigma$  for Gaussian). Here  $\chi_k$  is the bin argument,  $w_k$  is the bin value. More exactly, the  $k$ th bin is included, if both inequalities  $w_{k+1} < w_m$  and  $w_{k+2} < w_m$  are true (for the histogram right wing), or  $w_{k-1} < w_m$  and  $w_{k-2} < w_m$  (for the left wing). The latter double conditions prevent one from effects of the thin structure of a histogram. Approximately, about 20 bins are involved into calculations. The following formulae are used for the selected bins:

$$\chi_0 = \frac{\sum \chi_k w_k}{\sum w_k}, \quad (5)$$

$$\sigma^2 = \frac{\sum \chi_k^2 w_k}{\sum w_k} - \chi_0^2. \quad (6)$$

The value of  $\sigma$  calculated in this approach is underestimated. For the ideal Gaussian we have  $\sigma = 0.879\sigma_G$ , where  $\sigma_G$  is the true value. This factor is taken into account for the final value of  $\sigma$ . This approach helps to suppress the contribution of the distribution wings and of their asymmetry. It was tested many thousands of times in the author's own histogram package, similar to the famous HBOOK. It gives results close to the ones fitted by Gaussian+background function. For histograms with many peaks, this method gives right values of  $\chi_0$ ,  $\sigma$  for the highest of them. The subsidiary program is used to calculate  $\chi_0$ ,  $\sigma$ . Its call has the form

CALL HISMOM8(HIST,NH,A,B, $\chi_0$ , $\sigma$ ).

Here HIST(1:NH) is the histogram array, A and B are the left and right boundaries of the histogram, respectively.

In this way the  $\chi_0(1:NPAR)$ - and  $\sigma(1:NPAR)$ -arrays are formed. As a rule, some elements of these arrays are identical, but minimization of the number of histograms is a rather complicated combinatoric task, which produces no result, apart from saving some address space. For example, the number of independent histograms in the simplest case, described by Eq. (1), is  $N_s + 1$ .

### 3. TESTING

A number of testing programs and dozens of real ones were used for comparable testing of FUMILI and FUMILIM. No one case with nonidentical results was found. When all parameters are common for all events and  $NPAR > 2$ , the new

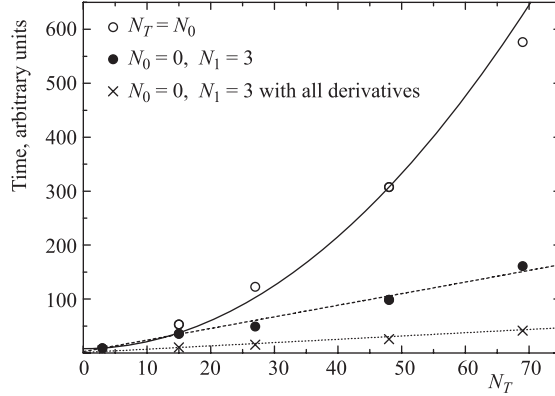


Fig. 2. Dependence of the solution time on the number of parameters. Open and solid circles show the time solution when no one parameter derivative is defined

program works 5–15% faster. The solution time grows as  $t \simeq a + c * \text{NPAR}^2$ . At  $\text{NPAR} = 2$  (and  $\text{LUN} < 0$ ) the speed advantage of the new program is about of factor 2.5(1.5) when all parameter derivatives are defined (not defined). When  $N_0 + N_1 \ll N_T$  (Eq. (1)), the solution time grows as  $t \simeq a + c * \text{NPAR}$ . The speed of FUMILI is the same in both cases.

The more parameter derivatives are defined analytically, the shorter the solution time. For the linear task with respect to all parameters the solution is 2.5–4 times faster when all derivatives are defined in comparison with the case when no one derivative is defined. The test results for different cases of solution time vs  $N_T$  for FUMILIM are shown in Fig. 2. Taking into account that at  $N_t = N_0$  the solution times for FUMILI and FUMILIM are more or less the same, the extrapolations of curves in Fig. 2 predict that the speed advantage of FUMILIM in comparison with FUMILI at  $N_T \gg N_0 + N_1$  can achieve one order of magnitude at 100 parameters and two orders of magnitude at 1000 parameters.

When the experimental points array is subdivided into several sets, which are not correlated to each other ( $N_0 = 0$ ), the global fit with all parameters is nevertheless preferable in comparison with doing cycle from 1 to  $N_s$  (and filtering is, as a rule, necessary, because events belonging to different sets are frequently distributed randomly in the common array of experimental points). The programming in this case is simpler, and the solution time is slightly better (because of no filtering).

The programs FUMSTRAW and FUMIDOUB were tested in the real exp.305 data processing (Saturne-2, Saclay). More exactly, they were created to improve the rather complicated data analysis of this experiment. The apparatus offsets

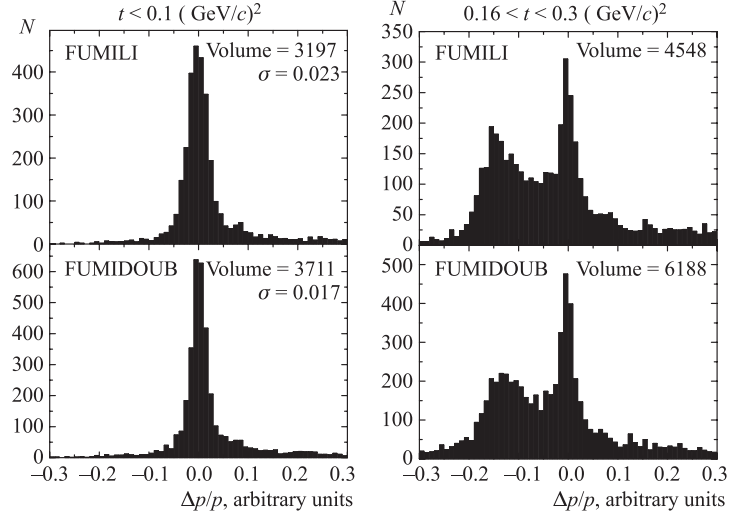


Fig. 3. Comparison of the elastic peak separation for two squared 4-momentum transfer intervals

were first adjusted with the help of FUMILI. The preliminary results of this experiment are presented in [8]. Thereafter the same job was fulfilled with the help of FUMIDOUB. In particular, when adjusting straw tubes offsets (time shifts (drifting), wire eccentricity and so on), about 250 parameters and two million events were involved into the fit. As the result, with new offsets the number of events passed through the set of criteria increased appreciably. The elastic peak resolution also became better. The comparison of the old results [8] and present ones are demonstrated in Fig.3. In the latter case the elastic peak resolution ( $\sigma = 0.017$ ) is proved to be close to the theoretical expectations ( $\sigma_{th} = 0.014$ ).

**Acknowledgements.** The author is grateful to Prof. G.A.Ososkov and A.P.Sapozhnikov for fruitful discussions and the correction of the text; to T. F. Sapozhnikova for help in preparing the seminar in memory of Prof. I. N. Silin.

## REFERENCES

1. Sokolov S.N., Silin I.N. JINR Preprint D-810. Dubna, 1961.
2. Silin I.N. CERN Program Library, D510, FUMILI.
3. Ketikian A.J., Komissarov E. V., Kurbatov V. S., Silin I.N. // NIM A. 1992. V. 314. P. 578.

4. *Kurbatov V. S., Silin I. N.* // NIM A. 1994. V. 345. P. 346.
5. *Dymov S. N., Kurbatov V. S., Silin I. N., Yashenko S. V.* // NIM A. 2000. V. 440. P. 431.
6. *Sitnik I. M.* // NIM A. 2004. V. 527. P. 278.
7. *James F., Roos M.* CERN Program Library, MINUIT, D506.
8. *Sitnik I. M. et al.* // Proc. of XI Advanced Research Workshop on High Energy Spin Physics (DUBNA-SPIN-05) / Edited by A. V. Efremov and S. V. Goloskokov. Dubna, 2006. P. 431.

Received on March 28, 2008.

Редактор *Е. И. Кравченко*

Подписано в печать 07.05.2008.

Формат 60 × 90/16. Бумага офсетная. Печать офсетная.

Усл. печ. л. 0,94. Уч.-изд. л. 1,16. Тираж 310 экз. Заказ № 56162.

Издательский отдел Объединенного института ядерных исследований  
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: [publish@jinr.ru](mailto:publish@jinr.ru)

[www.jinr.ru/publish/](http://www.jinr.ru/publish/)