**JINR preprint E11-2010-101**

# Modification of the FUMILI minimization package-2

*I.M. Sitnik*

Joint Institute for Nuclear Research
*e-mail: sitnik@sunhe.jinr.ru*

Submitted to Computer Physics Communications

## Abstract

FUMILI renovation-2 (FUMILIM) is suggested which has the following advantages: a more convenient and friendly user interface; no restriction to 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 the track reconstruction in straw chambers is suggested. The solution speed for the specific tasks was increased many times in comparison with the previous version of FUMILIM. All programmes are written using FORTRAN-77.

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

# 1 Introduction

This work has been done in memory of outstanding mathematician Prof. I.N.Silin, who went out on 21 Nov 2006. His famous program FUMILI[1-4] was the first one in the early sixties to provide the very advanced minimization method which gave to physicists the 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 sixties as beta-version, was kept unchangeable. The new user interface, suggested in this paper, was previously discussed between author and I.N.Silin in part and was supported by him.

Apart from the new user interface, the great attention was paid to the 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:

$$N_T = N_0 + N_1 N_g, \tag{1}$$

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 group, $N_g$ is the number of groups. When a user faces with the problem of adjusting big experimental setup offsets, such (or more complicated) combination arises frequently with large enough $N_g$. In author's practice the total number of parameters and events achieved several hundreds and about of 2 millions, respectively. The speed of solving tasks of this class was increased several times in FUMILIM-2 in comparison with FUMILIM-1[5].

In the new version of FUMILIM a number of services can be provided by changing of only one parameter in main program instead of making calls to additional programms with rather long list of parameters.

The problem of choosing of parameter step restrictions, which in FUMILI is the user's headache, is solved more or less successfully in FUMILIM. So, now the user is free from this nontrivial problem.

The basic code of FUMILI (the error matrix transform) during this optimization was kept unchangeable.

# 2 Program call

It is assumed below that all non-integer values are `*REAL8`.

The call has the form

```
HSQ=FUMILIM(A,NAM,NP,EXDA,LEXD,NEXD,UF,NIT,CF,CH).
```

Here

- `HSQ`=$\chi^2/NDF$ (Number of Degrees of Freedom, $NDF = NEXD - NP + 1$);
- `NP` – number of parameters (not restricted);
- `A(NP,6)` – parameters array;
- `NAM(NP)` – either character array of parameter names (up to 15 symbols) or name of file with initial parameters;
- `EXDA(LEXD,NEXD)` – experimental points array;
- `UF` – user's function;
- `NIT` – $abs(NIT)$ **is an iteration limit;**
- `CF` – output file name (`CF='*'` – screen output, `CF='-'` – no output);
- `CH` – character string to control the input/output.

Parameters array has the structure (corresponding common blocks in `FUMILI` are given in [...])

- `A(J,1)` – initial/final value of parameter [`A`];
- `A(J,2)` – parameter error (final)[`SIGMA`];
- `A(J,3)` – correlation factor (final) [`R`];
- `A(J,4)` – initial/final parameter step restriction [`PL`];
- `A(J,5)` – lower bound on parameter value(initial) [`AL`];
- `A(J,6)` – upper bound on parameter value (initial) [`AU`].

If the file with the name NAM exists, the initial data will be taken from this file. The file format is as follows:

```
1 name-1 A(1,1) A(1,4) A(1,5) A(1,6)
...
...
NP name-NP A(NP,1) A(NP,4) A(NP,5) A(NP,6)
```

If the initial parameter is not defined, then the string looks like:

```
i name-i 4*0.
```

Parameter number is defined by the string number in the initial data file. The numbers in the first column are ignored and are used only for better orientation of the user in his file.

The array of experimental points has the same, as in FUMILI, structure, namely

```
Y(1),ERR Y(1), X(1,1) [...X(LEXD-2,1)]
...
...
Y(NEXD),ERR Y(NEXD), X(1,NEXD) [...X(LEXD-2,NEXD)]
```

Here LEXD is the number of units belonging to one experimental point(LEXD$\geq$ 3), NEXD is the number of experimental points (not restricted). These values are equivalent to NED(2),NED(1) in FUMILI, respectively.

**There is also possible to organize the experimental points array as**

$$Y(i),\ ERR\,Y(i),\ Yth(i),\ X(1,i)\ [...X(LEXD-3,i)] \tag{2}$$

**where Yth(i) are values of fitted function after the fit. Initial values of Yth(i) can be arbitrary. In this case it is easy to calculate contribution of each experimental point to $\chi^2$ which is equal**

$$\chi^2 = \left(\frac{Y(i) - Yth(i)}{Err(i)}\right)^2 \tag{3}$$

**If the parameter Nit is negative the fast exit from the pro-**

**gram take place, when the fit has no perspectives (after 0th-iteration in case of $\chi^2/NDF > 500$, and after 1th-iteration in case of $\chi^2/NDF > 15$). This option is useful, when the several sets of the parameter initial values are used for the same set of experimental points.**

The name of UF may be chosen freely and must be declared EXTERNAL in the calling program. It should have three parameters: argument(s), parameters, derivatives, respectively. **An arbitrary number of derivatives can be defined**. Below is the example of UF for 5-th degree polynomial.

```
   FUNCTION POLI(X,C,DF)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION C(6) !  PARAMETERS
DIMENSION DF(6) !  DERIVATIVES

S=0.D0
DO I=1,6
S=S+C(I)*X**(I - 1)
DF(I)=X**(I - 1) !  NOT NECESSARY
ENDDO
POLI=S
RETURN
END
```

The simplest value for the parameter CH is '*'. The other values of this parameter are described in section 5.

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

```
HSQ=FUMILIM(A,'*',NP,EXDA,LEXD,NEXD,UF,NIT,CF,CH).
```

The following simplified call is possible:

```
HSQ = FUMILIR(HPAR,HEXDA,UF,NIT,CF,CH).
```

Here

- HPAR – name of file with initial data;
- HEXDA – name of file with experimental points, the first line of this file should contain LEXD, NEXD, then file structure should be the same as EXDA-array.

In this case, the user should not declare anything in the calling program, apart from `EXTERNAL UF`.

Special kind of optimization is applied for `NP=2` (typical for the track reconstruction), if to make call

```
H=FUMI2PAR(A,EXDA,LEXD,NEXD,UF,NIT).
```

This program ignores parameter fixing and restrictions. The print is not assumed. When all derivatives are user-defined, this program has speed advantage factor in comparison with `FUMILIM` about of 2.5.

# 3 Main differences with FUMILI

## 3.1 Initial data

`FUMILI` call has the form

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

The input parameters `EPS,N1,N2` now are internal ones in `FUMILIM`.

In `FUMILI` $\pm$huge values for parameter bounds are set by the `DATA`-operator. It is not possible to do so when the initial data arrays are not fixed in some common blocks. Now the same is set when `A(I,5)=A(I,6)=0`. **From this rule it follows that if one of bounds is set, the another one should be set too**.

In `FUMILI` the parameter step restriction `PL(I)`$\leq 0$ means the parameter is fixed. Now `A(I,4)`$< 0$ means the same, but `A(I,4)`$= 0$ means that the user trusts `FUMILIM` to find the optimal value of the parameter step restriction.

Parameters derivatives, if they are not defined in `UF`, are calculated by the formula

$$\frac{\partial f}{\partial a(i)} = \frac{f[...a(i) + s(i)] - f[...a(i)]}{s(i)}, \tag{4}$$

where $a(i)$ – parameter value, $s(i)$ – parameter increment. The value of $s(i)$ connected to `PL(I)` as $s(i)$=0.01 `PL(I)`.

If `UF` is strictly linear in the parameter, then $s(i)$ can be arbitrarily large. If $s(i)$ is small, it will lead to an excessive number of iterations. If `UF` strongly nonlinear in the parameter, and $s(i)$ is sufficiently large, then the result of the Eq.4 will be very vaguely resemble, what we call the first derivative, and the fit can go in the wrong direction.

As a rule, FUMILIM finds the optimal start values of $s(i)$ better than the user, so, **the recommended initial values for all parameter step restrictions are** A(I,4)=0.

## 3.2 Output data

The FUMILI output parameters S, AKAP, ALAM, IEND can be found in

```
COMMON/FUMOUTPUT/HISQ,IEND,IITER,AKAP,ALAM,NBAD
```

Here HISQ=$\chi^2$=2S, IITER is the number of real iterations, NBAD is the number of ignored experimental points, if the fit option was with taking off "bad points". The values of IEND are the following:

```
1 -- Full Success
2 -- Success When Some Parameters Are Fixed By Fumilim
3 -- Minimization Is Terminated, No Further Convergence
4 -- Minimization Is Terminated, Iteration Limit
5 -- Minimization Is Terminated, No Free Parameters
6 -- Fatal Initial Conditions Error
```

In the latter case the function value is -1.

## 3.3 Algorithm

The main code of FUMILI was kept unchangeable. Changes are connected to definition of the correlation factor only.

FUMILI calculates the correlation factor of $i$-th parameter with all the others. If it is not correlated with any other parameter, then the correlation factor $R(i) = 1$. Large values of $R(i)$ mean that no unique solution for UF. As a rule, the main contribution to $R(i) \gg 1$ gives one of pair correlations, $r(i, j)$. When we have a number of $R(i) \gg 1$, the task to isolate pairs becomes very unclear.

FUMILIM calculates only pair correlations, the largest of them at given $i$ are placed into column "Correlation Factor", together with the parameter number $j$, which is responsible for this correlation.

Pair correlation is calculated by the formula

$$r(i,j) = \frac{M_{ij}}{\sqrt{M_{ii}M_{jj}}} \ ,$$

where$\|M\|$ - error matrix. $r(i,j)$ satisfies Bunyakowsky's inequality:

$$-1 \leq r(i,j) \leq 1.$$

In the absence of correlations $r(i,j) = 0$. $|r(i,j)| > 0.9$ means that `UF` needs to be rethought.

# 4 Fit with taking off "bad points"

It is widely spread a double fit procedure, when the second fit is fulfilled after removing of 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 $\chi^2$-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 $i$-th experimental point, $y_i$ is the experimental value. **Experimental points with bound values of arguments can be defended from taking off.**

Frequently, the second fit gives the parameter values, which differ meaningfully from those in the previous fit, and they are more correct.

`FUMILIM` allows one to fulfill such a procedure using one call. During the first fit `FUMILIM` finds $\chi_0^2 = \chi^2/NEXD$. In subsequent fits the experimental points with $\chi^2 > C\chi_0^2$ are ignored. Recommended values are $7 < C < 12$. It is possible up to four cycles of the fit with different values of $C$.

One can set the values of $C$ in two ways.
The first way.
To define `C(M)` in

```
COMMON/FUMINPUT/C(4)
```

The second way.
In the initial data file write:

```
...
NP name-NP A(NP,1) A(NP,4) A(NP,5) A(NP,6)
C(1) [ C(2) C(3) C(4)]
```

Sometime the removing process become model dependent. For example, the polinomial fit leads to removing of points with bound values of the argument. The user can forbid removing of such points. The way to do it is described in the next section.

# 5 Input/Output control

All cases, considered in this section work when `CF` is not `'-'`.

`CF` file has the structure, resembled to `FUMILI` output one. Here the column with parameter names is included. The column `"Correlation Factor"` is slightly changed, as mentioned in section 3.3. Also the column `"Parameter Increment"` is added to show difference between initial and current parameter values.

In some tasks the minimzation process jumps over the minimum. In this case the program retreats to the parameter values of the previous iteration and recalculate parameter increments with 4 times less $s(i)$ (See Eq. 4). In this case the output string, devoted iteration, is **marked by star**.

Regardless of whether the initial data file exists or not, the final data file is always created in a format that is prescribed for the initial data file.

The output file name is `H.fpN` ("fp" means final parameters), where `H` - the name of the initial data file (if any) or the name of the first parameter, `N` – one digit number. If parameter names are not defined, `H='FUMI'`. If the file, named `H.fp1` exists, the file `H.fp2` will be created etc.. Thus, one can produce up to ten files with different versions of the fit. The length of `H` is a number of symbols in the input file name (or in the first parameter name) before the first blank or '.', but not more then 15 symbols.

The output are controlled by the character string CH(1-5 symbols):

- `CH='f'` (full output) – full information about each iteration goes to the file `CF`. By default, full information is provided only for the first(0) and the last iterations.

- `CH='h'` (histogram) – the 100-channel histogram of $\lg(\chi^2/\chi_0^2)$ is created in the file `H.hiN`. The string format is as follows: $x, N1, N2$, where $N1$ – number of events below the constraint, $N2$ – above.

- `CH='c'`(curve) – the 100-channel table of function, describing the data, is created in the file `H.cuN`. Its format is as follows: $x, y, y-er, y+er$,

where $y \pm er$ - confidence level.

- `CH='a'` (all points) – the list of all points is created in the file `H.apN`. This list is similar to output of `ERORF` from the `FUMILI` package.
- `CH='y'` – **in this case it is assumed that experimental points array is organized as in Eq. (2).**

Any combination of mentioned above symbols is possible in the string `CH`. Each record of files is accompanied by the message line in the `CF` file.

When `CH='c'`, but UF has more than one argument, the user should define the argument number `IX`, which will be scanned, in

```
COMMON/FUMX/X(20),IX
```

and set values of other arguments. By default, `IX=1`, the other arguments are taken from the first experimental point.

To forbid removing of bound points, described in the previous section, the user should introduce symbol ('1'... '9') which is treated as argument number, for which this forbiddance is applied. For example `CH='b13'`or, say, `CH='1b3'` means that the file with the list of "bad points" is ordered and it is forbidden to remove points with bound values of first and 3th arguments.

Example for one of the tasks with a constraint on $\chi^2$ is shown in Fig.1. When choosing a proper $x$ in this Fig., one should then use the constraint $C = 10^x$ in the program.

The range $(x_1, x_2)$ of the curve, describing data, is defined by the relations

$$x_1 = x_{min} - 0.1(x_{max} - x_{min}), \quad x_2 = x_{max} + 0.1(x_{max} - x_{min}),$$

where $x_{min}$( $x_{max}$) is the smallest(greatest) argument in the array of experimental points. Example of usage of this table in the ORIGIN graphic package shown in Fig. 2.

# 6 Problems with large number of parameters

The most time consuming part of the program is accumulation of error matrix [6]. 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` take place at each event. Here $N_T$ is the total number of parameters (see Eq.1). The first call is with the

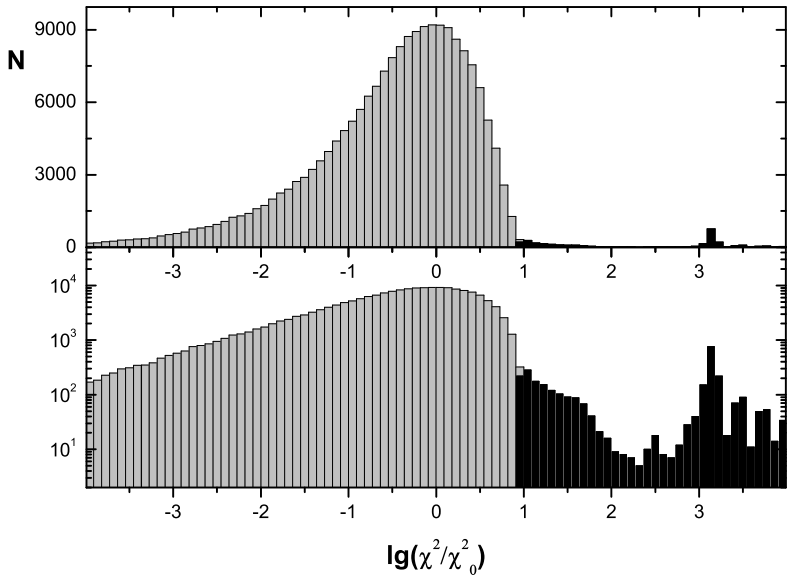


Figure 1: $\chi^2$-distributions for one of tasks in linear and logarithmic scales. Black columns are events, rejected by $\chi^2$-criterium.

current settings, then each parameter is given an increment, and again there is a call to find the partial derivative of the parameter. When $N_g$ is large, the overwhelming majority of calls produces $\partial f/\partial a(i) = 0$. Each such a call requires the same amount of time, as an effective call. To improve efficiency in this segment of the program, `FUMILIM` conducts preliminary iteration, which analyzes the last arguments in the array of experimental points. If each value differs from the integer less than $10^{-10}$, it is treated as this argument refers to a group number. Then the program adjusts itself to make only $(N_0 + N_1 + 1)$ `UF` calls at each event. This approach provides speed-up

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

We illustrate the comparison of `FUMILI` and `FUMILIM` speed for the task of this class in Fig.3. In our example $N_1$ is the same for all groups and increase the total number of parameters is achieved by increasing of $N_g$.

If the group contains subgroups, the recommendation is the following. Let we have a number of groups and $M_I$ subgroups in each $I$-th group. It is

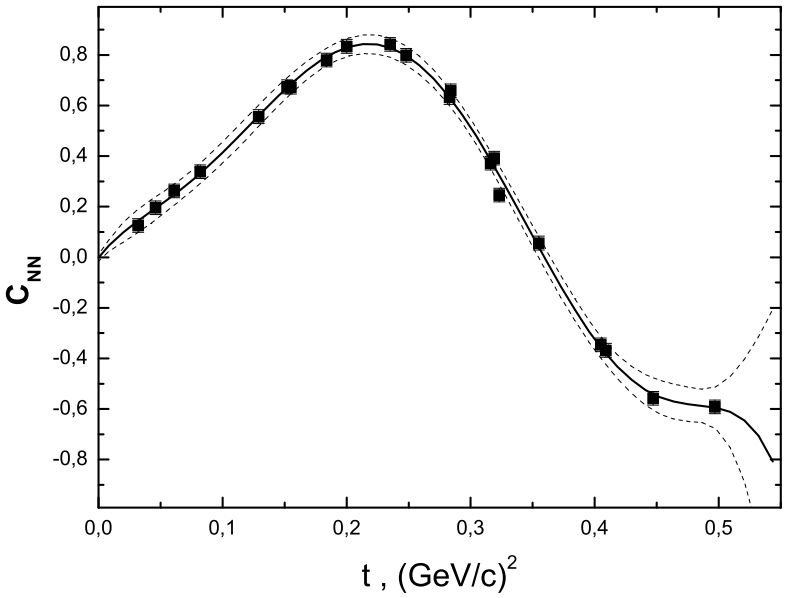


Figure 2: Polinomial fit of data with calculation of the confidence band.

suggested to construct the last argument of an experimental point as follows:

$$N_{Arg} = (I-1) * K + J,$$

where $I$ - number of the group, $J$ - number of subgroups in group $I$, K must be not less than the greatest $M_I$. The decoding of $N_{Arg}$ in `UF` is assumed.

# 7 The tracks via straw chambers

Additional programs in the `FUMILIM` package predestinated to reconstruct tracks via straw (drift) chambers.

This track reconstruction in straw chambers, where $x = A * z + B$, is not linear in $A$ 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. The call has the form:

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

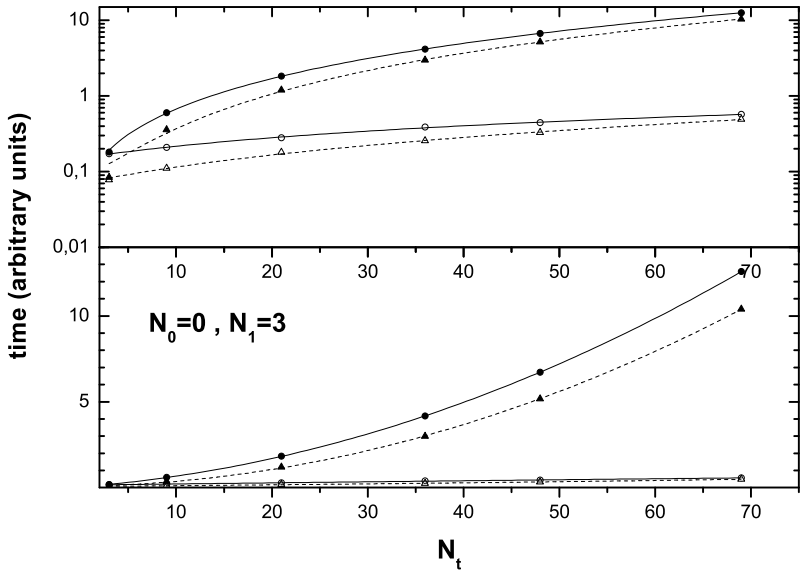


Figure 3: Speed comparison for FUMILI(top 2 curves) FUMILIM(bottom 2 curves) in linear and logarithmic scales. Solid curves – derivatives are not defined in UF, dashed ones – all derivatives are defined.

where H= $\chi^2$/NDF, A, B are mentioned above track parameters, EXDA is input array of experimental points:

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

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 or vice versa. Each FUMI2PAR call within FUMSTRAW is making with the references to UF, named FSTRAWD, which is included in the FUMILIM package.

**This program provides 250 000 solutions/sec for 4 straw planes using 2400 Mhz processor.**

When the number of straw planes is not large enough, two solutions with close values of $\chi^2$ are possible. It The solution with better $\chi^2$ is not always right one (so-called mirror solutions). In this case it would be better to use

```
I=IFUMSTRAW(A,B,H,EXDA,NEX,RH,DH)
```

Here `A, B, H` have the same sense as in previous call, but they should be declared in the calling program as

```
real*8 A(2),B(2),H(2)
```

The solutions are arranged to provide `H(1)<H(2)`.

If there are two close to each other solutions, `I=2`, otherwise `I=1`. The conditions of proximity of two solutions are the following:

$$\frac{\chi_2^2}{\chi_1^2} \leq RH, \quad or \quad \chi_2^2 - \chi_1^2 \leq DH.$$

These conditions should be defined by the user. Recommended values are `RH=1.5, DH=1`.

Two conditions of the proximity are used for the following reasons. There are examples where $\chi_1^2 \simeq 10^{-4}$, $\chi_2^2 \simeq 10^{-3}$. Here both values of $\chi^2$ are brilliant, and they pass the second criterion, although their ratio - a dozen. Another case where $\chi^2 \simeq 5$, $\chi^2 \simeq 7$. None of these solutions is good enough, but the second solution is hardly much worse than the first one. These solutions pass the first criterion.

When two solutions are close, additional information is needed to choose the right one.

# 8 Acknowledgements

The author is grateful to A.P. Sapozhnikov for fruitful discussions and to T.F. Sapozhnikova for help in spreading of this package.

# References

[1] S.N. Sokolov and I.N. Silin, Preprint JINR 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, NIM A 314 (1992) 578.

[4] V.S. Kurbatov, I.N.Silin, NIM A 345 (1994) 346.

[5] I.M. Sitnik, Preprint JINR E11-2008-43(2008).

[6] A.P. Sapozhnikov, http://wwwinfo.jinr.ru/programs/publ/pfumili.htm