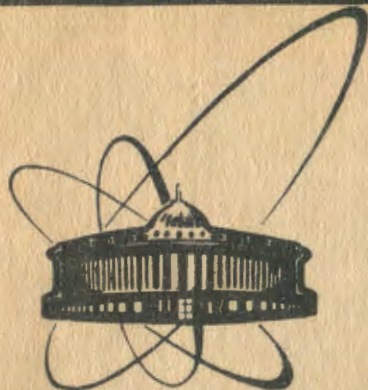


89-329



сообщения  
объединенного  
института  
ядерных  
исследований  
дубна

B 271

P11-89-329

Г. В. Велев, В. М. Маниев\*, Н. А. Русакович

ИНТЕРАКТИВНАЯ ПРОГРАММА МИНИМИЗАЦИИ  
ЛИНЕЙНОГО ФУНКЦИОНАЛА  
ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

---

\*Институт физики АН АзССР, Баку

1989

При обработке экспериментальных данных часто бывает необходимо оценить степень их отклонения от принятой гипотезы. Для этого во многих случаях используется критерий  $\chi^2$ . Величина  $\chi^2$  рассчитывается по формуле :

$$\chi^2 = \frac{1}{2} \sum_{j=1}^n \left\{ \frac{f(X_j, A) - F_j(X_j)}{\sigma_j} \right\}^2, \quad (1)$$

где  $f(X_j, A)$  — функция с параметрами  $A$  ( $A$  — вектор параметров), подгоняемая под измерения  $F_j \pm \sigma_j$  в точках  $X_j$ .

Нами разработана интерактивная программа FIT на языке BASIC для ПЭВМ типа IBM PC/AT/XT, которая минимизирует функционал (1). Эта программа основана на методе, описанном в работе<sup>1</sup>, и совпадает по конечным результатам с фортранной версией программы FUMILI<sup>2</sup>.

Программа FIT существует в двух версиях: для интерпретатора GWBASIC (версия I) и для компилятора BASIC (версия C). Основное отличие между ними состоит в следующем. В интерпретаторной версии необходимо заранее написать функцию  $f(X_j, A)$  (см. приложение А) в стандарте языка BASIC и запустить программу в командном интерпретаторе GWBASIC. Эта версия занимает около 90к дисковой памяти. При использовании компиляторной версии пользователь ждет во время компиляции и работы редактора связей, но исполняемый модуль выполняется на порядок быстрее. Эта версия занимает 235к на диске и работает с дополнительными

дискетами или твердым диском, потому что компилятор и редактор связей во время выполнения создают временные файлы размером около 140к.

### Работа с версией С

Эта версия включает в себя простой редактор, через который вводится функция Фита ( $f(X, A)$  в выражении (1)). Функция должна иметь следующий вид:

$$Y = f(X(J), A(I)), \quad (2)$$

где  $X(J)$  - аргументы функции ( $J = 1, 2, 3$ ),  $A(I)$  - параметры ( $I \leq 30$ ), определяемые из минимизации функционала (1),  $f$  - выражение на языках FORTRAN или BASIC\*.

Для увеличения быстродействия программы и более эффективного поиска минимума можно описать производные  $\frac{\partial Y}{\partial A(I)}$  по параметрам  $A(I)$  ( $\frac{\partial Y}{\partial A(I)}$ ) в аналитическом виде. Они должны иметь следующий вид:

$$DF(I) = \Phi(X(K), A(J)), \quad (3)$$

где  $\Phi$  - также выражение на языках FORTRAN или BASIC (см. приложение А). Если аналитический вид некоторых или всех производных  $\frac{\partial Y}{\partial A(I)}$  не задан пользователем, то производные вычисляются в программе численно.

Описание функции (2) и производных (3) не должно быть длиннее 23 строк.

Ниже дано короткое описание действия функциональных клавиш при работе редактора:

F1 - (Help) получить подсказку для описания функции и ее производных;

F2 - (Save) сохранить в файле набранный текст и изменить (если необходимо) начальное имя файла;

F3 - (Edit) сохранить в файле набранный текст и взять для редакции другой файл;

F4 - (Quit) выход из программы;

F5 - (Compile) вызов компилятора и редактора связей для создания исполняемого модуля. Для этого необходимо: 40 с для ПЭВМ типа IBM PC/AT и 250 с для ПЭВМ типа IBM PC/XT (Правец 16)\*\*;

F6 - (Setup) определение устройства или директории для временных файлов, которые создаются во время компиляции и работы редактора связей;

F9 - (DelLn) уничтожает строку в набираемом тексте;

F10 - (InsLn) вставляет строку в набираемом тексте.

После работы компилятора и редактора связей (клавиша F5) создается исполняемый модуль программы минимизации и начинается его исполнение.

Сначала необходимо ввести входные данные. Для этого имеется два способа :

а) программа получает с клавиатуры количество точек ( $\leq 400$ ), количество аргументов ( $\leq 3$ ), тип погрешностей (статистические, вычисляемые как квадратный корень из значения, или задаваемые пользователем) и границы изображения на дисплее. Затем для каждой точки вводятся аппроксимируемые значения  $F_j$  и значения аргументов  $x_j$ ;

б) данные читаются из файла, имеющего формат, указанный в приложении Б.

После этого программа начинает минимизацию функционала (1). При этом на экране изображаются результат каждой итерации, значения параметров и оценка их погрешностей.

В это время нажатием клавиши F10 можно остановить программу и выйти из нее, а нажатие F1 приостанавливает программу и высвечивает меню.

Короткое описание меню :

F1 - (Scales) изменение шкалы (линейная или логарифмическая) и границ изображения. Некоторые из точек  $F_j(x_j)$  могут оказаться за пределами выбранного окна, однако во всех случаях они учитываются при минимизации;

F2 - (LoadSave) считывание из файла или сохранение в файле параметров, их погрешностей и матрицы ошибок;

F3 - (Params) вызов редактора параметров. Возможно изменение значений параметров, их шагов и пределов изменения (если значение шага изменения параметра  $\Delta A \leq 0$ , этот параметр фиксируется);

**F4** - (CovMat) вывод на экран матрицы ошибок ;

**F5** - (Data) вызов редактора данных. Возможно изменение данных и их погрешностей. Если для значения номера точки  $N_j$  введено число  $N_j \leq 0$  (см. приложение B), эта точка убирается из набора данных. Возможно добавление новых точек;

**F6** - (Save D) сохранение данных в дисковом файле;

**F7** - (ErProp) включает и выключает изображение на экране коридора ошибок функции. Следует учесть, что включение изображения коридора ошибок увеличивает время итерации.

**F8** - (Funct) вывод на экран функции пользователя. Если необходимо изменить функцию, можно вновь вызвать редактора функции. После этого необходима повторная компиляция (см. описание функциональной клавиши **F5** в меню редактора);

**F9** - (CheckD) вывод на экран производных по параметрам, вычисленных аналитически в функции пользователя и численно в программе;

**F10** - (SliceD) в случае, когда функция зависит от нескольких аргументов, изменение показываемого среза.

#### Работа с версией I

В этой версии редактор функции отсутствует. Функция  $f(X_j, A)$  должна быть создана заранее на языке BASIC и спасена на диск командой SAVE "I имя функции I", а. Числа параметров, аргументов и экспериментальных точек могут быть любыми и ограничены лишь объемом памяти ЭВМ.

Текст функции  $f(X_j, A)$  должен обязательно начинаться с номера строки 10000 (см. приложение A). Количество строк для описания функции и производных не ограничено.

Отличия от версии C в назначении функциональных клавиш:

**F2** - (Load P) считывает значения параметров из файла;

**F4** - (Save P) сохраняет в файле параметры и их погрешности;

**F8** - (FunctD) выводит на экран текст используемой функции (но не дает возможности ее изменить).

\*Основная часть программы написана на языке BASIC и поэтому для полной совместимости функций  $f$  и  $F$  с выражениями на

языке FORTRAN, нами был создан простой препроцессор, который перед трансляцией переводит арифметические выражения в стандарт языка BASIC (напр.  $** \rightarrow ^$ , SORT  $\rightarrow$  SQRT и др.)

\*\* Существует версия программы FIT, которая работает только с твердым диском. В этой версии время, необходимое для создания исполняемого модуля после вызова компилятора и редактора связей, равно: 7 с для ПЭВМ типа IBM PC/AT и 15 с для ПЭВМ типа IBM PC/XT (Правец 16).

#### Приложение А

Пример описания функции минимизации (нормальное распределение):

а) для компиляторной версии

$$Z = (X(1) - A(3))$$

$$Y = A(1) * \exp(-0.5 * (Z/A(2)) ** 2)$$

$$DF(1) = Y/A(1)$$

$$DF(3) = Y * Z/A(2) ^ 2$$

б) для интерпретаторной версии

```
10000 Z = (X(1) - A(3))
```

```
10010 Y = A(1) * exp(-0.5 * (Z/A(2)) ^ 2)
```

```
10020 DF(1) = Y/A(1)
```

```
10030 DF(3) = Y * Z/A(2) ^ 2
```

```
10040 RETURN
```

#### Приложение Б

Формат файла с входными данными должен иметь следующий вид (пример):

```
50 2
1. 1.
0.42 0.73
5. 2.236
0.428 0.81
..... и т.д.
```

Числа разделяются пробелами. Первая строка - число точек и число аргументов. Вторая строка - аппроксимируемые значения и их погрешности. Третья строка - значения аргументов функции (в примере их 2) и т.д.

#### Приложение В

#	Value	Error	Argument
1	0.100E+01	0.100E+01	0.420E+00
2	0.500E+01	0.224E+01	0.424E+00
3	0.300E+01	0.173E+01	0.428E+00
4	0.500E+01	0.224E+01	0.432E+00
5	0.600E+01	0.245E+01	0.436E+00

..... и т.д.

Первый столбец - это номер точки. Второй - значение аппроксимируемого измерения. Третий - погрешность, четвертый и т.д. - аргументы функции. Нужно иметь в виду, что после изменения данных они считываются с экрана дисплея и при этом сохраняются лишь три значащие цифры.

Аналогичное замечание относится и к случаю, когда пользователь работает с редактором параметров.

#### Литература

1. Соколов С.Н., Силин И.Н. ОИЯИ, Д-810, Дубна, 1961 .
2. Silin I.N. CERN, Program Library D510, 1988.

Рукопись поступила в издательский отдел  
10 мая 1989 года.