

РОССИЙСКАЯ АКАДЕМИЯ НАУК
Ордена Ленина Сибирское отделение
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ
им. Г.И. Будкера СО РАН

А.Д. Букин

ПРОГРАММА ЧИСЛЕННОЙ МИНИМИЗАЦИИ
ФУНКЦИИ МНОГИХ ПАРАМЕТРОВ
С ИСПОЛЬЗОВАНИЕМ АНАЛИТИЧЕСКИ
ВЫЧИСЛЯЕМОГО ГРАДИЕНТА

ИЯФ 2005-43

НОВОСИБИРСК
2005

Программа численной минимизации функции многих параметров с использованием аналитически вычисляемого градиента

А.Д. Букин

Институт ядерной физики им. Г.И.Будкера
630090, Новосибирск, РФ

Аннотация

Разработана программа численной минимизации функции многих параметров с использованием аналитически вычисляемого градиента. В программе реализованы алгоритмы: метод «золотого сечения» для минимизации функции одной переменной, симплекс-метод, метод переменной метрики пространства, метод сопряжённых направлений, метод Ньютона, метод проб и ошибок, случайный поиск. Возможна комбинация разных методов. Для выбранной комбинации методов возможна процедура многократной минимизации из разных начальных точек. Для тех алгоритмов, которые используют градиент функции, используется значение градиента, вычисляемое отдельной подпрограммой пользователя (предполагается, что вычисление производится по точным аналитическим формулам). Программа написана на языке Фортран-77. Общий объём кода около 3300 строк.

Subroutine for numerical minimization of function of many parameters using analytically evaluated gradient

A.D. Bukin

Abstract

A program of numerical minimization of function of many parameters using analytically evaluated gradient is carried out. There are several algorithms of minimization implemented in this program: "Golden section" method for one parameter minimization, simplex method, variable metrics method, conjugate directions method, Newton's method, "try and fail" method, random search. It is possible to combine different methods. For the chosen combination of methods there is possible a multiple minimization procedure from different start points. For those algorithms which use function gradient there is used a user supplied subroutine for gradient evaluation (supposed that exact analytical formulas are used). Program language is Fortran 77. Total size of source code is about 3300 rows.

© *Институт ядерной физики им. Г.И.Будкера СО РАН*

1 Введение

В экспериментальной физике элементарных частиц для минимизации функций многих переменных широко используется программа MINUIT [1], разработанная в ЦЕРН. Автор неоднократно пытался разработать эффективный алгоритм минимизации, начиная ещё с того времени, когда библиотеки программ из зарубежных ведущих центров по физике высоких энергий были недоступны в России. Однако, все попытки были безуспешны, и метод переменной метрики пространства, реализованный в программе MINUIT, является, по-видимому, наиболее мощным алгоритмом минимизации гладких функций.

Тем не менее, программа MINUIT не всесильна, и иногда возникают сомнения в правильности результата (конечно, это бывает, как правило, со сложными функциями). Кроме того, иногда может возникнуть необходимость провести минимизацию функции на вычислительной машине, где ещё не установлена библиотека программ ЦЕРНа, но есть компилятор Фортрана. Могут быть и другие причины использовать альтернативную программу минимизации. Для этих целей разработана компактная программа на языке Фортран-77, включающая в себя несколько перспективных алгоритмов, с возможностью произвольного сочетания этих алгоритмов, а также с возможностью многократного повторения минимизации для выбранного комплекта алгоритмов из разных начальных точек. Наиболее эффективные алгоритмы минимизации используют значение градиента функции. В общем случае градиент оценивается численно, но если имеются компактные формулы

для вычисления градиента, то вычисление градиента по этим формулам предпочтительнее. В программе MINUIT эти два варианта вычисления градиента объединены. По мнению автора, такое объединение неудобно для пользователя. Ранее была описана программа, использующая численно оцениваемый градиент [2].

Данная работа является подробным описанием версии программы, в которой используется предоставляемая пользователем подпрограмма для точного вычисления градиента по формулам, и сравнительным анализом качества её работы (в сравнении с MINUIT) на примере нескольких существенно различных функций.

При обработке экспериментальных данных со Сферического Нейтрального Детектора (СНД) [3] эти программы многократно использовались для получения надёжных результатов.

2 Структура программы

2.1 Вызов программы

Вызов программы оформлен в простейшем виде. Предполагается, что пользователем предоставлена подпрограмма, вычисляющая градиент. Допускается, что могут быть точки, где градиент функции не определён. В этих точках подпрограмма пользователя должна выдавать сигнал, что в данной точке градиент не существует.

Вызов:

```
call BUKMING(FCN,Grad,Npar,x,iFlMin,dfm,Ncal,Fmin)
```

```
integer *4 Npar, Ncal, iFlMin(Npar)
```

```
real *8 x(Npar), dfm, Fmin
```

```
external FCN,Grad
```

Минимизируемая функция F в точке x N -мерного пространства должна вычисляться программой FCN при вызове

```
call FCN(N,F,x)
```

```
integer *4 N
```

```
real *8 F,x(N)
```

Аналитически вычисляемый градиент $G(N)$ в точке $x(N)$ должен предоставляться подпрограммой $Grad$ при вызове

call Grad(N,x,G,iRet)

integer *4 N,iRet

real *8 x(N), G(N)

iRet — код возврата

iRet = 0 градиент успешно вычислен,

iRet=1 градиент не удалось вычислить (не существует в данной точке).

Входные параметры:

FCN — идентификатор подпрограммы, вычисляющей функцию.

Grad — идентификатор подпрограммы, вычисляющей градиент.

Npar — количество параметров, от которых зависит функция.

x — массив, содержащий начальную точку для минимизации.

iFlMin — массив, содержащий признаки оптимизируемых параметров. Если $iFlMin(k) = 0$, то минимизация по этому параметру не производится — он остаётся с первоначально определённым значением.

dfm — требуемая точность минимизации — оценка найденного минимума не должна отличаться от истинного минимума больше, чем на эту константу.

Ncal — переменная, содержащая максимально допустимое количество вызовов программ FCN и Grad.

Выходные параметры:

x — массив с координатами точки минимума.

Ncal — возвращается реальное количество вызовов FCN и Grad, произведённое при поиске минимума.

Fmin — минимальное значение функции.

2.2 Общие блоки

Общие блоки в большой степени определяют работу программы. Кроме того, полный список названий общих блоков удобно иметь для анализа возможных коллизий в больших программах по названиям модулей. Для уменьшения вероятности совпадения имён в данной программе с именами других модулей пользователя имена общих блоков и программ выбраны начинающимися с bukmig.

Так как в фортране нет динамического распределения памяти, то максимально возможное количество параметров функции определяется в тексте программы в самом начале в переменной NparMax. Для изменения максимального числа параметров следует изменить значение этой константы в строке

```
parameter (NparMax=...
```

Краткое описание общих блоков:

`common/bukmig00/Istr,Ivmm,Icgm,Isim,Ideb,Inew,Itaf,Iran`

integer *4 Istr, Ivmm, Icgm, Isim, Ideb, Inew, Itaf, Iran

Istr — стратегия минимизации:

- 0** — простейшая стратегия. В каждом из выбранных методов проверяются минимальные критерии достижения минимума. Если в цепочке программ минимизации какая-то программа сообщает о достижении минимума, то минимизация на этом заканчивается. Многократных заходов минимизации из разных начальных точек не производится.
- 1** — в каждом из выбранных методов применяются более надёжные критерии достижения минимума. Если в цепочке программ какая-то программа сообщает о достижении минимума (кроме методов с ненадёжным критерием достижения минимума: симплекс-метода, метода проб и ошибок, метода случайного поиска), то минимизация на этом заканчивается. Если этого не происходит, то минимизация проводится многократно из разных начальных

точек с алгоритмом выбора этих точек таким же, как при $Istr=2$.

- 2 — Всё как в предыдущем пункте, но многократная минимизация из разных начальных точек обязательно проводится, игнорируя сообщения о достижении минимума от отдельных подпрограмм минимизации (детали алгоритмов выбора начальных точек для следующего захода минимизации изложены в описании аналогичной программы [2]).
- 3 — Проводится многократная минимизация из разных начальных точек, игнорируя сообщения от отдельных подпрограмм минимизации о достижении минимума. Алгоритмы выбора каждой следующей начальной точки и критерий достижения минимума для $Istr=2$ и $Istr=3$ различаются.

$Ivmm$ — если больше нуля, то метод переменной метрики пространства используется при минимизации. Очередность вызова программ минимизации с разными алгоритмами определяется значениями этих сигнальных переменных. Если $Ivmm < Isim$, то вначале используется метод переменной метрики пространства, а затем симплекс-метод. При равных значениях сигнальных переменных порядок применения методов минимизации следующий: метод переменной метрики пространства, метод сопряжённых градиентов, симплекс-метод, метод Ньютона, метод проб и ошибок, метод случайного поиска.

$Icgm$ — такая же сигнальная переменная для метода сопряжённых градиентов.

$Isim$ — сигнальная переменная для симплекс-метода.

$Inew$ — сигнальная переменная для метода Ньютона.

$Itaf$ — сигнальная переменная для метода проб и ошибок.

$Iran$ — сигнальная переменная для метода случайного поиска.

По умолчанию установлены

Istr=1,Ivmm=1,Icgm=Itaf=Iran=Inew=0,Isim=2,Ideb=0.

Ideb — уровень отладочной печати. Каждый следующий уровень включает в себя все предыдущие.

0 — нет печати.

1 — начало и окончание каждого захода минимизации. Фиксация каждого нового минимального значения функции.

2 — значения градиентов, параметры и достижения каждого логического шага.

3 — максимально подробная печать.

**common/bukmig01/Npartot,Nfree,NcalTot,NcalMax,
indfree(Nfree)**

integer *4 Npartot, Nfree, NcalTot, NcalMax, indfree

Npartot — то же самое, что Npar — общее количество параметров.

Nfree — число свободных (оптимизируемых) параметров.

NcalTot — общее количество вызовов программ FCN и Grad.

NcalMax — максимально допустимое число вызовов программ FCN и Grad.

indfree — массив индексов оптимизируемых параметров в массиве x(Npar).

common/bukmig02/Fbest,xbest(Npar)

real *8 Fbest, xbest — текущее минимальное значение и точка минимума.

common/bukmig03/Nrun,NrunMax,iRets(NrunMax)

integer *4 Nrun, NrunMax, iRets

Nrun/NrunMax — текущее/максимальное число заходов минимизации из разных начальных точек. Если Nrun превышает NrunMax, то новые заходы могут быть, но информация о самых старых заходах теряется и Nrun=NrunMax.

iRets — массив кодов завершения заходов минимизации.

**common/bukmig04/Frun(NrunMax),xRun(NparMax,
NrunMax),R0(NparMax),wi(NrunMax),M(NparMax,
NparMax),v1(NparMax),v2(NparMax),ti(NparMax)**

real *8 Frun, xRun, R0, wi, M, v1, v2, ti

Frun — минимальные значения функции, достигнутые в результате данного захода минимизации.

xRun — точки, в которых заканчивался каждый заход минимизации.

R0, wi, M, v1, v2, ti — вспомогательные массивы для вычисления следующей начальной точки минимизации.

common/bukmig05/xLin(Npar)

real *8 xLin — рабочий массив для программы bukming0 (промежуточная программа для вычисления функции, осуществляющая переход из пространства минимизируемых параметров в полное пространство параметров, а также регистрацию самого минимального значения функции).

common/bukmig06/xStart(Npar),eDir(Npar),xMin(Npar)

real *8 xStart,eDir,xMin — рабочие массивы для программы bukming1 (одномерная минимизация).

common/bukmig07/h(Npar)

real *8 h — массив текущих шагов

**common/bukmig08/Vmm(Npar,Npar),grad1(Npar),
grad2(Npar),dG(Npar),xw1(Npar),xw2(Npar),
Vdg(Npar),Amat(Npar,Npar),Fsimsim(Npar,Npar),
dX(Npar),xsimsim(Npar,Npar+1),xw3(Npar),xw4(Npar)**

real *8 Vmm, grad1, grad2, dG, xw1, xw2, dX, Vdg, Amat, Fsimsim, xsimsim, xw3, xw4

Рабочие массивы для разных подпрограмм минимизации.

2.3 Подпрограммы

Все подпрограммы, кроме BUKMING0, работают с векторами и матрицами в сокращённом пространстве, состоящем из переменных параметров функции. Полное пространство переменным также используют программы пользователя, вычисляющие значение функции и градиента в заданной точке.

BUKMING0(FCN,Npar,xCur,Fcur)

```
integer *4 Npar
real *8 xCur(Npar), Fcur
external FCN
```

Программа вычисляет значение функции Fcur в указанной точке xCur в пространстве оптимизируемых параметров. Эта же программа регистрирует минимальное значение функции, достигнутое в данном счёте.

BUKMING1(FCN,Nfree,xStart,eDir,dfm,xMin,Fmin)

```
integer *4 Nfree
real *8 xStart(Nfree), eDir(Nfree), xMin(Nfree), dfm, Fmin
external FCN
```

Программа одномерной минимизации из начальной точки xStart вдоль направления eDir с требуемой точностью dfm. Предполагается, что в начальной точке значение функции известно и передаётся программе в переменной Fmin. В результате минимизации методом «золотого сечения» получается точка xMin с минимальным значением Fmin.

BUKMING2(FCN,GrFun,Nfree,iLev,x0,F0,grad,xwork)

```
integer *4 Nfree,iLev
real *8 x0(Nfree), F0, grad(Nfree), xwork(Nfree)
external FCN, GrFun
```

Промежуточная программа для вычисления градиента grad(.) в точке x0(.). Если программа GrFun вычислила градиент,

и установлен признак отладочной печати, то дополнительно оценивается численно градиент в этой же точке. Пользователь может сравнить напечатанные значения градиентов с целью обнаружения ошибок в подпрограмме GrFun (надо иметь в виду, что небольшие отличия обязательно должны быть, так как численная оценка градиента не может быть такой же точной, как вычисление по аналитическим формулам). F0 — значение функции в точке x0. xwork(.) — рабочий массив. Обращение к функции GrFun засчитывается, как вычисление функции при анализе достижения предела на количество вычислений функции.

На выходе из программы устанавливается код возврата в iLev: 0 — градиент вычислен, 1 — невозможно вычислить градиент.

**BUKMING3(FCN,GrFun,dfm,Nfree,iRet,xmin,Fmin,Vmm,
grad1,grad2,dG,xw1,xw2,dX,Vdg)**

integer *4 Nfree,iRet

real *8 dfm,xmin(Nfree),Fmin,Vmm(Nfree,Nfree),grad1(Nfree),
grad2(Nfree),dG(Nfree),xw1(Nfree),xw2(Nfree),dX(Nfree),
Vdg(Nfree)

Минимизация функции FCN Nfree параметров методом переменной метрики пространства. GrFun — подпрограмма пользователя для вычисления градиента. iRet — код возврата (0 — минимум найден, 1 — минимум не найден). xmin при обращении должен содержать начальную точку минимизации, по выходу будет содержать точку минимума (если найден). Fmin — значение функции в минимуме. Остальные массивы — рабочие.

**BUKMING4(FCN,Nfree,Nsim,iRet,xmin,Fmin,dfm,xwsim,
Fsim,xsim,xc,esim)**

integer *4 Nfree,Nsim,iRet

real *8 xmin(Nfree),Fmin,dfm,xwsim(Nfree),Fsim(Nsim),
xsim(Nfree,Nsim),xc(Nfree),esim(Nfree,Nfree)

Минимизация функции FCN Nfree параметров симплекс-методом. В xmin при входе содержится начальная точка, при выходе — точка с минимальным значением функции Fmin. iRet=0, если минимум найден, iRet>0, если условие достижения минимума не выполнено.

**BUKMING5(FCN,GrFun,dfm,Nfree,iRet,xmin,Fmin,pvec,
 evec,grcdm1,grcdm2,Cicdm,gicdm,pnorm,gpi1,
 gpi2,xw1,xw2,alphi,dXcdm)**

integer *4 Nfree,iRet

real *8 dfm,xmin(Nfree),Fmin,pvec(Nfree,Nfree),evec(Nfree,Nfree),
 grcdm1(Nfree),grcdm2(Nfree),Cicdm(Nfree), gicdm(Nfree),
 pnorm(Nfree),gpi1(Nfree),gpi2(Nfree),xw1(Nfree),xw2(Nfree),
 alphi(Nfree), dXcdm(Nfree)

Минимизация функции FCN Nfree параметров методом сопряжённых направлений. iRet — код возврата (0 — минимум найден, 1 — минимум не найден). xmin при обращении должен содержать начальную точку минимизации, по выходу будет содержать точку минимума. Fmin — значение функции в минимуме. Остальные массивы — рабочие.

**BUKMING6(FCN,GrFun,dfm,Nfr,iRet,xmin,Fmin,Amat,
 Awork,g,dx,xw1,xw2)**

integer *4 Nfr, iRet

real *8 dfm, xmin(Nfr), Fmin, Amat(Nfr,Nfr), Awork(Nfr,Nfr),
 g(Nfr), dx(Nfr), xw1(Nfr), xw2(Nfr)

Минимизация функции FCN по Nfr переменным параметрам методом Ньютона.

BUKMING7(FCN,dfm,Nfr,iRet,xmin,Fmin,xw1)

integer *4 Nfr, iRet

real *8 dfm, xmin(Nfr), Fmin, xw1(Nfr)

Минимизация функции FCN по Nfr переменным параметрам методом проб и ошибок.

BUKMING8(FCN,dfm,Nfr,iRet,xmin,Fmin,xw1,dX,g)

integer *4 Nfr, iRet

real *8 dfm, xmin(Nfr), Fmin, xw1(Nfr), dX(Nfr), g(Nfr)

Минимизация функции FCN по Nfr переменным параметрам методом случайного поиска.

BUKMING9(g)

real *8 g

Простой генератор нормальной случайной переменной.

3 Вычисление градиента

В данной версии программы минимизации градиент вычисляется в подпрограмме пользователя. Следует иметь в виду, что ошибки в вычислении градиента дают катастрофические последствия в градиентных методах минимизации. К сожалению, производить какую-то автоматическую проверку правильности вычисления градиента не представляется возможным. Дело в том, что решающее влияние градиента на ход минимизации, в основном, начинается вблизи минимума. Если функция сложная, то численная оценка градиента не дает надежного значения — оно всегда будет как-то отклоняться от точного значения. Поэтому при включении отладочной печати можно визуально сравнить распечатки значений градиентов, полученных из программы пользователя и при численной оценке. Такое сравнение может дать основания для отдельной проверки подпрограммы вычисления градиента.

Учитывая высокую точность вычисления градиента по аналитическим формулам, в критерии достижения точки минимума в градиентных методах используется в качестве обязательного требования близость к нулю величины градиента: абсолютная величина не должна превышать 10^{-3} .

4 Метод переменной метрики пространства (VMM)

Известный метод переменной метрики пространства [4, 5] в данной программе реализован в том же самом виде, как в программе BUKMIN [2], только градиент получается из подпрограммы пользователя.

5 Метод сопряжённых направлений (CDM)

Метод реализован в соответствии со схемой из работы [6]. Алгоритм без изменения перенесен из программы [2], только используется не численная оценка градиента, а точное значение из программы пользователя.

6 Симплекс-метод

Симплекс-метод [5, 7] позволяет эффективно спускаться по крутому склону в область минимума, не требуя вычисления градиента, но скорость сходимости к точке минимума у него гораздо ниже, чем у метода переменной метрики пространства.

Данная версия алгоритма подробно описана в [2]. В принципе, если есть аналитически вычисляемый градиент и мощные градиентные методы минимизации (метод переменной метрики пространства и метод сопряженных направлений), то необходимости в неградиентных методах не возникает. Однако, допущение, что градиент может существовать не во всех точках пространства, требует сохранения в данном комплекте и неградиентных методов минимизации с возможностью автоматического переключения на них в особых точках.

7 Метод Ньютона

Наиболее существенно при возможности использовать аналитическое значение градиента изменился алгоритм метода Ньютона. Параметры квадратичной формы, аппроксимирующей функцию в некоторой окрестности текущей точки, можно получить из значений градиента.

$$\Delta F = \sum_{i=1}^n g_i \Delta x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \Delta x_i \Delta x_j,$$

где a_{ij} — симметричная матрица. Градиент аппроксимируется этими же коэффициентами:

$$\mathbf{G}_i = g_i + \sum_{j=1}^n a_{ij} \Delta x_j.$$

Тогда

$$g_i = \mathbf{G}_i(\Delta \mathbf{x} = 0), \quad a_{ij} = \frac{\mathbf{G}_i(\Delta x_j = h) - g_i}{h}.$$

Делая поочерёдно по каждой координате шаг h , можно определить таким образом все коэффициенты a_{ij} . При этом, однако, матрица получится несимметричной, так как функция в общем случае не является квадратичной формой. В нашем случае матрицу надо симметризовать (вместо вычисленных значений a_{ij} и a_{ji} взять их среднее значение).

Если $\sum_{i=1}^n g_i^2 > 0$, то текущая точка не может быть минимумом.

Стационарную точку аппроксимирующей квадратичной формы (в ней градиент равен нулю) можно найти из системы n линейных уравнений

$$\sum_{j=1}^n a_{ij} \Delta x_j = -g_i, \quad i = 1, 2, \dots, n,$$

которую можно решить любым способом, например, последовательным исключением неизвестных. Если в этой точке предполагаемого минимума значение функции окажется меньше, чем в исходной, то перемещаемся в эту точку и всё повторяем сначала. Если

же значение функции окажется больше, то проводим линейную минимизацию вдоль направления на эту точку.

Удобство этого метода в том, что каждый следующий шаг делается независимо от предыдущей информации, поэтому вблизи точки минимума восстановление параметров квадратичной формы проводится точно, и сходимость вблизи минимума должна быть быстрой.

В качестве критерия достижения точки минимума с заданной точностью трудно что-нибудь здесь придумать, кроме совпадения предсказанного значения функции (по квадратичной форме) с действительным значением в новой точке. Кроме случая, конечно, когда градиент точно обращается в нуль.

8 Метод проб и ошибок

Метод основан на поочерёдных пробах вдоль каждой координаты с некоторым шагом h_i . Алгоритм сохранен в точности таким же, как в программе [2]. В таблицах этот метод будем обозначать как *T&F* (Try and Fail).

9 Случайный поиск

Вариант случайного поиска оставлен таким же, как в программе [2].

10 Многократное повторение заходов минимизации

Многократное повторение заходов минимизации из разных начальных точек может предприниматься только при стратегии $Istr > 0$. Детальное описание процедуры выбора новой начальной точки для очередного захода минимизации и критерия достижения минимума имеется в [2].

11 Отладка программы

Для отладки программы и проверки эффективности алгоритмов были выбраны несколько тестовых функций, описанных в таблице 1. Хотя набор функций точно такой же, какой был использован для программы [2], все описания и комментарии приведены здесь полностью, так как исследование эффективности программы минимизации является основной темой данной работы. Далее везде будут ссылки на тестовые функции по их порядковому номеру в этой таблице. Все функции имеют единственный минимум, и значение функции в минимуме всегда равно нулю. Начальные значе-

Таблица 1: Тестовые функции для проверки эффективности алгоритмов минимизации. Минимальное значение функций равно нулю.

№	Кол. пар.	Функция	Точка минимума
1	5	$(1 + x_1)^2 + 2^2 \cdot (2 + x_1 + 2^2 x_2)^2 +$ $+ 3^2 \cdot (3 + x_1 + 2^3 x_2 + 3^3 x_3)^2 +$ $+ 4^2 \cdot (4 + x_1 + 2^4 x_2 + 3^4 x_3 + 4^4 x_4)^2 +$ $+ 5^2 \cdot (5 + x_1 + 2^5 x_2 + 3^5 x_3 + 4^5 x_4 + 5^5 x_5)^2$	$(-1, -0.25, 0, \frac{1}{256}, -\frac{1}{3125})$
2	2	$100 \cdot (x_2 - 0.01 \cdot x_1^2 + 1)^2 + 0.01 \cdot (x_1 + 10)^2$	$(-10, 0)$
3	2	$100x_2^2 + 0.01 \cdot x_1 + 10 $	$(-10, 0)$
4	2	$100 \sqrt{ x_2 - 0.01x_1^2 } + 0.01 x_1 + 10 $	$(-10, 1)$
5	2	$1000 \cdot x_1^2 + x_2^2 - 800 + x_1 + x_2 + 40 $	$(-20, -20)$
6	4	$[1000 x_2 - 0.001x_1^3 + x_2 + x_1 + 11] \times$ $\times [1 + 1000 x_3^2 + x_4^2 - 800 + x_3 + x_4 + 40] +$ $+ 1000 x_3^2 + x_4^2 - 800 + x_3 + x_4 + 40 $	$(-10, -1, -20, -20)$
7	8	$1000 \times \left\{ (x_1 + 1 - \rho \cos 5\rho)^2 + \right.$ $+ (x_2 + 2 - \rho \sin 5\rho \cos 6\rho)^2 +$ $+ (x_3 + 3 - \rho \sin 5\rho \sin 6\rho \cos 7\rho)^2 +$ $+ (x_4 + 4 - \rho \sin 5\rho \sin 6\rho \sin 7\rho \cos 8\rho)^2 +$ $+ (x_5 + 5 - \rho \sin 5\rho \sin 6\rho \sin 7\rho \sin 8\rho \cos 9\rho)^2 +$ $+ (x_6 + 6 - \rho \sin 5\rho \sin 6\rho \sin 7\rho \sin 8\rho \sin 9\rho \times$ $\quad \times \cos 10\rho)^2 +$ $+ (x_7 + 7 - \rho \sin 5\rho \sin 6\rho \sin 7\rho \sin 8\rho \sin 9\rho \times$ $\quad \times \sin 10\rho \cos 11\rho)^2 +$ $+ (x_8 + 8 - \rho \sin 5\rho \sin 6\rho \sin 7\rho \sin 8\rho \sin 9\rho \times$ $\quad \times \sin 10\rho \sin 11\rho)^2 \left. \right\} + 0.1\rho,$ $\rho = \sqrt{\sum_{i=1}^8 (x_i + i)^2}.$	$(-1, -2, -3, -4, -5, -6, -7, -8)$

ния всех параметров выбраны равными 1. Результаты минимизации этих функций известной программой MINUIT [1] приведены в таблице 2.

Таблица 2: Результат минимизации тестовых функций в виде F_{min}/N_{cal} программой MINUIT (несколько основных режимов) и программой BUKMING (default mode, т.е. ISTR=1, Ivmm=1, Isim=2).

№	MINUIT			BUKMING default
	Команда MINUIT	Status	Result	
1	simplex 1000000	PROGRESS	24.01/515	$\frac{2.0 \cdot 10^{-21}}{38}$
	migrad 1000000	CONVERGED	$4.8 \cdot 10^{-12}/75$	
	minimize 1000000	CONVERGED	$4.8 \cdot 10^{-12}/75$	
	minos 1000000	SUCCESSFUL	$4.8 \cdot 10^{-12}/115$	
2	simplex 1000000	PROGRESS	0.86/ 87	$\frac{3.0 \cdot 10^{-9}}{76}$
	migrad 1000000	CONVERGED	$2.0 \cdot 10^{-7}/85$	
	minimize 1000000	CONVERGED	$2.0 \cdot 10^{-7}/85$	
	minos 1000000	SUCCESSFUL	$2.0 \cdot 10^{-7}/195$	
3	simplex 1000000	PROGRESS	$4.6 \cdot 10^{-4}/33$	$\frac{4.4 \cdot 10^{-14}}{1512}$
	migrad 1000000	CONVERGED	$7.5 \cdot 10^{-12}/126$	
	minimize 1000000	CONVERGED	$7.5 \cdot 10^{-12}/126$	
	minos 1000000	PROBLEMS	$7.5 \cdot 10^{-12}/234$	
4	simplex 1000000	PROGRESS	0.22/103	$\frac{4.0 \cdot 10^{-6}}{57329}$
	migrad 1000000	FAILED	0.12/139	
	minimize 1000000	FAILED	0.12/265	
	minos 1000000	FAILURE	0.12/269	
5	simplex 1000000	PROGRESS	69.30/77	$\frac{6.9 \cdot 10^{-11}}{14420}$
	migrad 1000000	FAILED	80.00/106	
	minimize 1000000	CONVERGED	80.00/153	
	minos 1000000	FAILURE	80.00/264	
6	simplex 1000000	PROGRESS	1650.14/259	$\frac{7.4 \cdot 10^{-7}}{260579}$
	migrad 1000000	FAILED	898.4/236	
	minimize 1000000	FAILED	895.5/576	
	minos 1000000	FAILURE	470.3/89516	
7	simplex 1000000	PROGRESS	78020.94/1037	$\frac{7.4 \cdot 10^{-12}}{230981}$
	migrad 1000000	CONVERGED	1.79/698	
	minimize 1000000	CONVERGED	1.79/698	
	minos 1000000	PROBLEMS	1.79/144801	

Использована версия программы 96.03, для более устойчивой минимизации установлено “set strategy 2”. Для того, чтобы исполь-

зовался градиент из программы пользователя, выдана команда “set GRAD 1”. Для некоторых функций существуют точки, где градиент не существует. Но для программы MINUIT нет возможности сообщения, что данная точка особенная. Поэтому градиент вычисляется всегда, но если в формулах знаменатель обращается в нуль, то он заменяется на некоторую ненулевую константу, и вычисление градиента продолжается. Такое неправильное значение градиента, несомненно, может испортить результат минимизации, но это также является предметом данного исследования. В этой же таблице приведены результаты минимизации этих функций программой BUKMING в режиме, который устанавливается по умолчанию. По результатам видно, что некоторые функции представляют проблему даже для такой программы, как MINUIT.

В табл. 3 приведены результаты минимизации тестовых функций программой BUKMING с помощью каждого алгоритма отдельно при двух стратегиях ($Istr=0$ и $Istr=1$).

Результаты получились неоднозначные вследствие очень разных свойств функций. Попробуем рассмотреть их для каждой функции отдельно.

1. Первая функция — простая квадратичная функция, представляющая сложность только из-за очень сильно различающихся собственных значений матрицы квадратичной формы ($\lambda_1 = 1.88$, $\lambda_2 = 124.9$, $\lambda_3 = 12951.3$, $\lambda_4 = 2.08 \cdot 10^6$, $\lambda_5 = 5.44 \cdot 10^8$). MINUIT здесь показала великолепный результат в методе переменной метрики пространства. Методы минимизации в BUKMING разбиваются на две группы, которые условно можно назвать градиентными (VMM, CDM и Newton) и неградиентными (Simplex, T&F, Random). Первые должны хорошо работать с гладкими, дифференцируемыми функциями и быстро приближаться к минимуму после достижения области, где минимизируемая функция уже хорошо аппроксимируется квадратичной формой. Поэтому закономерно, что и в программе BUKMING метод переменной метрики пространства, метод сопряжённых направлений и метод Ньютона достигли хорошего результата. Метод Нью-

Таблица 3: Результаты минимизации тестовых функций в виде $\frac{F_{min}}{N_{cal}}$ с помощью каждого алгоритма отдельно при двух стратегиях (Istr=0 и Istr=1). Ограничение на число вызовов функции FCN установлено $N_{cal} < 5 \cdot 10^6$, требование на точность минимизации $\Delta F_m = 10^{-3}$.

Номер функ.	Результаты для разных алгоритмов					
	VMM	CDM	Simplex	Newton	T&F	Random
Istr = 0						
1	$\frac{2.9 \cdot 10^{-11}}{35}$	$\frac{4.9 \cdot 10^{-5}}{117}$	$\frac{12.9}{667}$	$\frac{3.4 \cdot 10^{-25}}{37}$	$\frac{1.1 \cdot 10^6}{1389}$	$\frac{36312.6}{471}$
2	$\frac{1.9 \cdot 10^{-5}}{71}$	$\frac{7.6 \cdot 10^{-5}}{92}$	$\frac{5.2 \cdot 10^{-6}}{234}$	$\frac{8.6 \cdot 10^{-8}}{249}$	$\frac{0.36}{2733}$	$\frac{1.10}{76}$
3	$\frac{1.4 \cdot 10^{-4}}{65}$	$\frac{0.110}{13}$	$\frac{8.6 \cdot 10^{-6}}{79}$	$\frac{3.4 \cdot 10^{-8}}{199}$	$\frac{0.00015}{48}$	$\frac{0.105}{47}$
4	$\frac{0.110}{196}$	$\frac{17.37}{30}$	$\frac{0.119}{192}$	$\frac{0.110}{128}$	$\frac{0.201}{108}$	$\frac{0.351}{83}$
5	$\frac{80.00}{302}$	$\frac{80.00}{1669779}$	$\frac{78.6}{308}$	$\frac{80.0}{177}$	$\frac{78.46}{115}$	$\frac{77.7}{104}$
6	$\frac{7.9 \cdot 10^6}{363}$	$\frac{1.0 \cdot 10^7}{353}$	$\frac{543.5}{1375}$	$\frac{1.0 \cdot 10^7}{169}$	$\frac{1832.1}{237}$	$\frac{5061.97}{110}$
7	$\frac{0.134}{146}$	$\frac{0.130}{162}$	$\frac{300179.1}{1421}$	$\frac{20694.9}{310835}$	$\frac{433951.4}{1685}$	$\frac{333.6}{1838}$
Istr = 1						
1	$\frac{2.0 \cdot 10^{-21}}{38}$	$\frac{1.9 \cdot 10^{-11}}{482}$	$\frac{0.00011}{9331}$	$\frac{3.4 \cdot 10^{-25}}{37}$	$\frac{0.0031}{139385}$	$\frac{1004.8}{6729}$
2	$\frac{3.0 \cdot 10^{-9}}{76}$	$\frac{3.6 \cdot 10^{-10}}{100}$	$\frac{5.1 \cdot 10^{-14}}{903}$	$\frac{8.6 \cdot 10^{-8}}{249}$	$\frac{3.4 \cdot 10^{-6}}{1866}$	$\frac{8.4 \cdot 10^{-9}}{875}$
3	$\frac{9.6 \cdot 10^{-15}}{858}$	$\frac{9.9 \cdot 10^{-15}}{314}$	$\frac{1.8 \cdot 10^{-7}}{654}$	$\frac{2.9 \cdot 10^{-12}}{1447}$	$\frac{2.8 \cdot 10^{-5}}{605}$	$\frac{0.0032}{414}$
4	$\frac{0.0025}{5 \cdot 10^6}$	$\frac{2.9 \cdot 10^{-5}}{95253}$	$\frac{0.00072}{152105}$	$\frac{4.2 \cdot 10^{-5}}{13517}$	$\frac{0.00023}{158965}$	$\frac{0.0328}{15410}$
5	$\frac{1.9 \cdot 10^{-8}}{6206}$	$\frac{1.6 \cdot 10^{-9}}{1675600}$	$\frac{4.2 \cdot 10^{-7}}{8995}$	$\frac{9.2 \cdot 10^{-9}}{4768}$	$\frac{2.2 \cdot 10^{-9}}{11855}$	$\frac{0.00017}{278342}$
6	$\frac{248.3}{59528}$	$\frac{7.4 \cdot 10^6}{1286}$	$\frac{2.8 \cdot 10^{-5}}{525489}$	$\frac{9.8 \cdot 10^6}{671}$	$\frac{0.00025}{10^6}$	$\frac{128.9}{457}$
7	$\frac{1.4 \cdot 10^{-12}}{109367}$	$\frac{4.3 \cdot 10^{-13}}{296037}$	$\frac{0.875}{502437}$	$\frac{8056.0}{1.6 \cdot 10^6}$	$\frac{1.653}{5 \cdot 10^6}$	$\frac{1.132}{17252}$

тона здесь наиболее эффективен, так как за один шаг точно определяет матрицу квадратичной формы и переходит точно в точку минимума.

Неградиентные методы (Simplex, T&F, Random) в данном случае не могут конкурировать с методами первой группы.

2. Вторая функция тоже является гладкой везде дифференцируемой функцией, но не является квадратичной формой. Поэтому для неё важно, как преобразуются параметры квадратичной формы при приближении к минимуму. В данном случае опять хорошо справились с задачей методы VMM, Ньютона и метод сопряжённых направлений. Симплекс-метод тоже показал неплохой результат.
3. Третья функция везде имеет ограниченный градиент, кроме дна ущелья, где градиент не определён. Поэтому для этой функции среди успешных оказался симплекс-метод. Метод CDM пришел в точку, где градиент не определён. Неожиданно успешным оказался здесь метод Ньютона, хотя вблизи дна ущелья функция не может быть представлена поверхностью второго порядка.
4. Четвёртая функция отличается от предыдущей тем, что «дно ущелья» представляет собой не прямую линию, как в предыдущей функции, а параболу. Кроме того, при приближении к линии дна ущелья градиент возрастает неограниченно (стремится к бесконечности). Поэтому ни один из алгоритмов программы MINUIT не достиг минимума. В программе BUKMING тоже ни один из алгоритмов не достиг точки минимума. Однако, при стратегии ISTR=1 за счет большого числа заходов минимизации до минимума с заданной точностью дошли метод Ньютона, метод сопряженных направлений, симплекс-метод и метод проб и ошибок. Метод переменной метрики пространства не дошел до минимума из-за ограничения на количество вызовов функции.
5. В пятой функции градиент везде ограничен, а на самом дне ущелья не определён. В программе MINUIT ни один из методов не достиг минимума. Интересно, что команда MINIMIZE, которая должна быть более надежной, чем MIGRAD, хоть и тоже не достигла минимума, но сообщила о выполнении

критериев достижения минимума (ложный минимум). В программе BUKMING при стратегии $ISTR=0$ ни один из методов не дошел до минимума, но при стратегии $ISTR=1$, наоборот, все методы достигли точки минимума, причем меньше всего времени потратил на это метод Ньютона. Следующим за ним был метод переменной метрики пространства.

6. Шестая функция имеет схожие свойства с пятой функцией, но определена в четырёхмерном пространстве. В программе MINUIT ни один из методов не привёл к минимуму, также как и в программе BUKMING. Правда, при многократном повторении заходов минимизации симплекс-метод все же достиг точки минимума, а также метод проб и ошибок при несколько большем времени.
7. Седьмая функция интересна тем, что градиент в ней ограничен и определён во всём восьмимерном пространстве, кроме самой точки минимума. Кроме того, путь вдоль ущелья очень извилист и долог. Степень извилистости может характеризовать, например, такой факт, что если начать движение на расстоянии $\rho = 10$ от минимума (по прямой), то суммарный криволинейный путь по дну ущелья будет около 419.5, то есть более, чем в 40 раз превышает прямолинейное расстояние. Трудность минимизации этой функции повышается также малым значением градиента на дне «ущелья» (правда, сложно говорить о профиле функции в восьмимерном пространстве, как об ущелье). График зависимости минимального значения градиента от расстояния до точки минимума приведён на рис. 1. Функция оказалась настолько трудна, что ни один из алгоритмов ни в программе MINUIT, ни в BUKMING не достиг цели, но всё же предпочтительнее оказались градиентные методы VMM, CDM, которые за счет многократного повторения заходов минимизации достигли точки минимума (при стратегии $ISTR=1$).

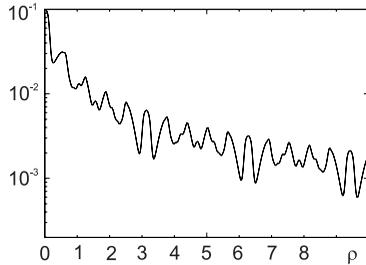


Рис. 1: Минимальное значение градиента для функции номер 7 в зависимости от расстояния ρ от точки минимума.

Для первых трех функций, где MINUIT достигла успеха, в программе BUKMING таким же успешным оказался метод переменной метрики пространства.

В среднем несколько менее эффективными оказались метод Ньютона и метод сопряженных направлений, хотя тоже достигли точек минимума. В качестве режима минимизации «по умолчанию» для программы BUKMING можно выбрать стратегию ISTR=1 и метод переменной метрики пространства. При этом на простых функциях программа BUKMING будет сравнима с MINUIT по эффективности. Для повышения надежности минимизации (повышения вероятности, что программа действительно пришла в точку минимума), хорошо бы дополнить алгоритм переменной метрики пространства каким-то другим.

Включение стратегии Istr=1 в программе BUKMING (возможность многократного повторения минимизации из разных начальных точек в случае осознанных проблем) позволяет достичь минимума для всех функций, кроме шестой, методами переменной метрики пространства или методом сопряженных направлений. Минимизация шестой функции оказалась непосильной задачей и для MINUIT, и для всех методов в BUKMING, кроме симплекс-метода. В среднем, неплохие результаты у симплекс-метода, правда, при худшей эффективности для гладких функций, чем у градиентных методов, — при Istr=1 только седьмую функцию не удалось минимизировать с заданной точностью. Посмотрим, можно ли улучшить результат, используя разные «танделы» методов минимизации.

ции (аналогично команде MINIMIZE в программе Minuit, когда после применения метода MIGRAD в случае неудачи используется метод SIMPLEX). В таблице 4 приведены результаты таких экспериментов.

Результат, как всегда, неоднозначный. Функции 6 и 7 по-прежнему представляют наибольшую трудность. Если рассматривать только первые 5 функций, то при стратегии Istr=1 справились с задачей все проверенные варианты. Однако, шестая и седьмая функция оказалась по силам только «тандемам» VMM→Simplex и CDM→Simplex. Учитывая, что на простых функциях первый тандем быстрее приходит в точку минимума, в качестве стандартного режима была выбрана стратегия Istr=1 и тандем алгоритмов VMM→Simplex. Так программа BUKMING будет минимизировать, если пользователь не поменяет значения соответствующих переменных в общем блоке /BUKMIG00/. Такой выбор можно охарактеризовать коротко — «не хуже MINUIT, быстро и достаточно точно для функций простых и средней сложности» (на основании данных о результате минимизации в таблицах).

Стратегии Istr=2 и Istr=3 призваны проверять надежность найденного минимума. В них полностью игнорируется «мнение» программ, реализующих конкретные алгоритмы минимизации, и вывод о достижении точки минимума делается только на основе нескольких последовательных заходов минимизации из разных начальных точек. Эти две стратегии различаются алгоритмом выбора новой стартовой точки для очередного захода минимизации.

При Istr=2 надёжнее всего минимизирует «связка» программ VMM→Simplex, или CDM→Simplex, или VMM→Newton→Simplex — минимизирует все функции, правда, при этом на минимизацию более простых функций тратит неоправданно много процессорного времени.

Таблица 5 демонстрирует результаты минимизации этих же семи функций при стратегии Istr=3.

При этой стратегии тоже есть успешные последовательности программ, которые сумели найти минимумы всех функций — VMM→Simplex, VMM→Newton→Simplex, VMM→CDM→Simplex.

Таблица 4: Результаты минимизации тестовых функций в виде $\frac{F_{min}}{N_{cal}}$ с помощью разных сочетаний алгоритмов отдельно при двух стратегиях (Istr=1 и Istr=2). Ограничение на число вызовов функции FCN установлено $N_{cal} < 10^7$, требование на точность минимизации $\Delta F_m = 10^{-3}$.

Набор алгор.	Результаты для разных функций						
	1	2	3	4	5	6	7
Istr = 1							
VMM ↓ Simplex	$\frac{2.0 \cdot 10^{-21}}{38}$	$\frac{3.0 \cdot 10^{-9}}{76}$	$\frac{4.4 \cdot 10^{-14}}{1512}$	$\frac{4.0 \cdot 10^{-6}}{57329}$	$\frac{6.9 \cdot 10^{-11}}{14420}$	$\frac{7.4 \cdot 10^{-7}}{260579}$	$\frac{7.4 \cdot 10^{-12}}{230981}$
Newton ↓ Simplex	$\frac{3.4 \cdot 10^{-25}}{37}$	$\frac{8.6 \cdot 10^{-8}}{249}$	$\frac{9.2 \cdot 10^{-13}}{943}$	$\frac{2.3 \cdot 10^{-6}}{195860}$	$\frac{2.0 \cdot 10^{-11}}{15015}$	$\frac{0.0043}{218141}$	$\frac{1.6 \cdot 10^{-7}}{10^7}$
Newton ↓ VMM	$\frac{3.4 \cdot 10^{-25}}{37}$	$\frac{8.6 \cdot 10^{-8}}{249}$	$\frac{2.4 \cdot 10^{-15}}{2025}$	$\frac{4.2 \cdot 10^{-5}}{18362}$	$\frac{3.3 \cdot 10^{-9}}{5764}$	$\frac{9836186}{899}$	$\frac{1.788}{358356}$
VMM ↓ Newton	$\frac{2.0 \cdot 10^{-21}}{38}$	$\frac{3.0 \cdot 10^{-9}}{76}$	$\frac{1.8 \cdot 10^{-12}}{1461}$	$\frac{6.3 \cdot 10^{-6}}{15069}$	$\frac{5.4 \cdot 10^{-9}}{7308}$	$\frac{544.8}{2143}$	$\frac{7.4 \cdot 10^{-14}}{109730}$
VMM ↓ CDM	$\frac{2.0 \cdot 10^{-21}}{38}$	$\frac{3.0 \cdot 10^{-9}}{76}$	$\frac{2.4 \cdot 10^{-19}}{1586}$	$\frac{4.2 \cdot 10^{-5}}{14059}$	$\frac{1.2 \cdot 10^{-9}}{7276}$	$\frac{445.5}{1838}$	$\frac{2.0 \cdot 10^{-12}}{109922}$
CDM ↓ Simplex	$\frac{2.1 \cdot 10^{-16}}{755}$	$\frac{3.6 \cdot 10^{-10}}{100}$	$\frac{1.8 \cdot 10^{-17}}{4376}$	$\frac{1.0 \cdot 10^{-6}}{192493}$	$\frac{6.1 \cdot 10^{-12}}{1682988}$	$\frac{1.4 \cdot 10^{-4}}{183990}$	$\frac{4.7 \cdot 10^{-13}}{44821}$
VMM ↓ CDM ↓ Simplex	$\frac{2.0 \cdot 10^{-21}}{38}$	$\frac{3.0 \cdot 10^{-9}}{76}$	$\frac{2.3 \cdot 10^{-40}}{3293}$	$\frac{5.7 \cdot 10^{-6}}{60862}$	$\frac{2.9 \cdot 10^{-11}}{15024}$	$\frac{0.086}{10^7}$	$\frac{7.4 \cdot 10^{-12}}{230983}$
Istr = 2							
VMM ↓ Simplex	$\frac{8.4 \cdot 10^{-26}}{3412}$	$\frac{2.8 \cdot 10^{-18}}{42597}$	$\frac{4.4 \cdot 10^{-14}}{1512}$	$\frac{4.0 \cdot 10^{-6}}{57329}$	$\frac{6.9 \cdot 10^{-11}}{14420}$	$\frac{7.4 \cdot 10^{-7}}{260579}$	$\frac{1.1 \cdot 10^{-12}}{235589}$
Newton ↓ Simplex	$\frac{4.6 \cdot 10^{-27}}{3801}$	$\frac{6.3 \cdot 10^{-18}}{3280}$	$\frac{9.2 \cdot 10^{-13}}{943}$	$\frac{2.3 \cdot 10^{-6}}{195860}$	$\frac{2.0 \cdot 10^{-11}}{15015}$	$\frac{0.0043}{218141}$	$\frac{1.6 \cdot 10^{-7}}{10^7}$
CDM ↓ Simplex	$\frac{7.3 \cdot 10^{-24}}{3605}$	$\frac{1.9 \cdot 10^{-14}}{3614}$	$\frac{1.8 \cdot 10^{-17}}{4376}$	$\frac{1.0 \cdot 10^{-6}}{192493}$	$\frac{6.1 \cdot 10^{-12}}{1682988}$	$\frac{0.00014}{183990}$	$\frac{4.7 \cdot 10^{-13}}{44821}$
Newton ↓ VMM	$\frac{2.2 \cdot 10^{-26}}{302}$	$\frac{1.4 \cdot 10^{-16}}{846}$	$\frac{2.4 \cdot 10^{-15}}{2025}$	$\frac{4.2 \cdot 10^{-5}}{18362}$	$\frac{3.3 \cdot 10^{-9}}{5764}$	$\frac{9836186.0}{899}$	$\frac{4.6 \cdot 10^{-12}}{10^7}$
VMM ↓ Newton	$\frac{4.6 \cdot 10^{-27}}{254}$	$\frac{4.8 \cdot 10^{-21}}{8083716}$	$\frac{1.8 \cdot 10^{-12}}{1461}$	$\frac{6.3 \cdot 10^{-6}}{15069}$	$\frac{5.4 \cdot 10^{-9}}{7308}$	$\frac{544.8}{2143}$	$\frac{7.4 \cdot 10^{-14}}{109730}$
VMM ↓ CDM	$\frac{1.0 \cdot 10^{-28}}{275}$	$\frac{6.3 \cdot 10^{-21}}{40366}$	$\frac{2.4 \cdot 10^{-19}}{1586}$	$\frac{4.2 \cdot 10^{-5}}{14059}$	$\frac{1.2 \cdot 10^{-9}}{7276}$	$\frac{445.5}{1838}$	$\frac{2.0 \cdot 10^{-12}}{109922}$
VMM ↓ Newton ↓ Simplex	$\frac{6.3 \cdot 10^{-28}}{4523}$	$\frac{1.4 \cdot 10^{-20}}{950261}$	$\frac{8.7 \cdot 10^{-16}}{1970}$	$\frac{3.1 \cdot 10^{-5}}{66982}$	$\frac{3.1 \cdot 10^{-11}}{14661}$	$\frac{2.2 \cdot 10^{-8}}{502335}$	$\frac{2.6 \cdot 10^{-12}}{235805}$
VMM ↓ CDM ↓ Newton	$\frac{2.4 \cdot 10^{-26}}{3523}$	$\frac{5.5 \cdot 10^{-20}}{5927495}$	$\frac{2.3 \cdot 10^{-40}}{3533}$	$\frac{5.7 \cdot 10^{-6}}{64380}$	$\frac{1.4 \cdot 10^{-11}}{16775}$	$\frac{15.0}{10^7}$	$\frac{1.0 \cdot 10^{-12}}{236190}$

Таблица 5: Результаты минимизации тестовых функций в виде $\frac{F_{min}}{N_{cal}}$ с помощью разных сочетаний алгоритмов при стратегии Istr=3. Ограничение на число вызовов функции FCN установлено $N_{cal} < 10^7$, требование на точность минимизации $\Delta F_m = 10^{-3}$.

Набор алгор.	Результаты для разных функций						
	1	2	3	4	5	6	7
VMM	$\frac{2.6 \cdot 10^{-27}}{186}$	$\frac{4.9 \cdot 10^{-23}}{1050}$	$\frac{3.7 \cdot 10^{-18}}{5782}$	$\frac{4.4 \cdot 10^{-5}}{11626}$	$\frac{3.0 \cdot 10^{-7}}{18345}$	$\frac{39.2}{5227977}$	$\frac{4.5 \cdot 10^{-12}}{45828}$
CDM	$\frac{5.7 \cdot 10^{-20}}{6927}$	$\frac{1.1 \cdot 10^{-17}}{560}$	$\frac{1.8 \cdot 10^{-17}}{4047}$	$\frac{8.7 \cdot 10^{-6}}{68689}$	$\frac{9.6 \cdot 10^{-8}}{1698623}$	$\frac{57.2}{115427}$	$\frac{6.1 \cdot 10^{-12}}{10727}$
Simplex	$\frac{3.2 \cdot 10^{-13}}{344810}$	$\frac{1.1 \cdot 10^{-22}}{3621}$	$\frac{3.8 \cdot 10^{-14}}{2807}$	$\frac{2.4 \cdot 10^{-7}}{105314}$	$\frac{1.6 \cdot 10^{-5}}{39868}$	$\frac{0.400}{108266}$	$\frac{1.790}{10^7}$
Newton	$\frac{3.4 \cdot 10^{-25}}{62}$	$\frac{6.3 \cdot 10^{-24}}{2965}$	$\frac{2.8 \cdot 10^{-15}}{5767}$	$\frac{2.7 \cdot 10^{-4}}{9004}$	$\frac{1.0 \cdot 10^{-8}}{10679}$	$\frac{316084.5}{15247}$	$\frac{2.251}{10^7}$
T&F	$\frac{2.7 \cdot 10^{-5}}{968129}$	$\frac{3.5 \cdot 10^{-9}}{77287}$	$\frac{2.9 \cdot 10^{-11}}{6764}$	$\frac{0.00020}{15207}$	$\frac{3.0 \cdot 10^{-6}}{20401}$	$\frac{80.0}{154977}$	$\frac{1.655}{841178}$
Random	$\frac{21.92}{439501}$	$\frac{6.8 \cdot 10^{-12}}{13105}$	$\frac{1.2 \cdot 10^{-9}}{3953}$	$\frac{0.081}{3342}$	$\frac{0.031}{5583}$	$\frac{101.17}{4843}$	$\frac{1.493}{62686}$
VMM ↓ Simplex	$\frac{8.1 \cdot 10^{-26}}{2501}$	$\frac{7.4 \cdot 10^{-21}}{2464}$	$\frac{5.2 \cdot 10^{-16}}{1587}$	$\frac{4.0 \cdot 10^{-7}}{100802}$	$\frac{3.0 \cdot 10^{-8}}{88986}$	$\frac{0.0010}{233384}$	$\frac{2.4 \cdot 10^{-12}}{60918}$
CDM ↓ Simplex	$\frac{4.5 \cdot 10^{-19}}{27670}$	$\frac{8.5 \cdot 10^{-26}}{3219}$	$\frac{8.5 \cdot 10^{-16}}{1903}$	$\frac{2.3 \cdot 10^{-7}}{329966}$	$\frac{2.0 \cdot 10^{-12}}{1743157}$	$\frac{0.0032}{177669}$	$\frac{8.9 \cdot 10^{-13}}{17118}$
Newton ↓ Simplex	$\frac{1.8 \cdot 10^{-26}}{1899}$	$\frac{1.1 \cdot 10^{-24}}{7877}$	$\frac{1.6 \cdot 10^{-13}}{5237}$	$\frac{4.2 \cdot 10^{-6}}{120406}$	$\frac{1.1 \cdot 10^{-7}}{46980}$	$\frac{0.00016}{205881}$	$\frac{1.788}{10^7}$
VMM ↓ Newton ↓ Simplex	$\frac{3.4 \cdot 10^{-27}}{2209}$	$\frac{3.8 \cdot 10^{-23}}{6958}$	$\frac{9.9 \cdot 10^{-16}}{1395}$	$\frac{3.7 \cdot 10^{-7}}{135250}$	$\frac{1.3 \cdot 10^{-10}}{58716}$	$\frac{2.7 \cdot 10^{-6}}{356871}$	$\frac{2.3 \cdot 10^{-13}}{58178}$
VMM ↓ CDM ↓ Simplex	$\frac{1.4 \cdot 10^{-26}}{2661}$	$\frac{5.8 \cdot 10^{-24}}{3697}$	$\frac{2.3 \cdot 10^{-40}}{1628}$	$\frac{3.1 \cdot 10^{-7}}{140870}$	$\frac{2.7 \cdot 10^{-9}}{80488}$	$\frac{2.0 \cdot 10^{-5}}{190034}$	$\frac{1.5 \cdot 10^{-12}}{58185}$

Для полноты картины осталось проверить, как справляется программа BUKMING с минимизацией функций при очень большом числе параметров (возьмём для проверки $n = 100$). Выберем для минимизации следующие функции:

Таблица 6: Результат минимизации тестовых функций с $n = 100$ в виде F_{min}/N_{cal} программой MINUIT (несколько основных режимов) и программой BUKMING ($N_{cal} < 10^7$, default mode, т.е. ISTR=1, Ivmm=1, Isim=2).

№	MINUIT			BUKMING default
	Команда MINUIT	Status	Result	
8	simplex 1000000	PROGRESS	$8.1 \cdot 10^7/251284$	$\frac{1.1 \cdot 10^{-26}}{301}$
	migrad 1000000	CONVERGED	$9.1 \cdot 10^{-12}/10901$	
	minimize 1000000	CONVERGED	$9.1 \cdot 10^{-12}/10901$	
	minos 1000000	PROBLEMS	$9.1 \cdot 10^{-12}/11754$	
9	simplex 1000000	PROGRESS	1256.8/113975	$\frac{5.7 \cdot 10^{-13}}{2485}$
	migrad 1000000	CONVERGED	$5.7 \cdot 10^{-5}/23236$	
	minimize 1000000	CONVERGED	$5.7 \cdot 10^{-5}/23236$	
	minos 10000000	PROBLEMS	$5.7 \cdot 10^{-5}/50224$	
10	simplex 1000000	PROGRESS	880.6/85543	$\frac{1.6 \cdot 10^{-6}}{5.2 \cdot 10^6}$
	migrad 1000000	FAILED	71.6/19088	
	minimize 1000000	FAILED	71.6/39813	
	minos 10000000	PROBLEMS	71.6/7167010	
11	simplex 1000000	PROGRESS	1034.5/75089	$\frac{0.0038}{10^7}$
	migrad 1000000	FAILED	73.2/18229	
	minimize 1000000	FAILED	73.2/39061	
	minos 10000000	FAILURE	8.98/323379	

8. Квадратичная форма

$$F_8 = \sum_{k=1}^{n-1} \left[\sum_{j=1}^k (x_j + j) - k \cdot (x_{k+1} + k + 1) \right]^2 + \left[\sum_{j=1}^n (x_j + j) \right]^2.$$

Одно собственное значение равно n , остальные выражаются формулой $\lambda_k = k \cdot (k + 1)$, $k = 1, 2, \dots, n - 1$.

9. Гладкая функция с извилистым ущельем

$$F_9 = (x_1 + 10)^2 + \left[1000 + (x_1 + 10)^2 \right] \left[\sum_{k=2}^n \left(x_k + 10 \sin \frac{k \cdot x_1}{k - 1} \right)^2 \right].$$

10. Функция с извилистым ущельем, у которой градиент везде существует и ограничен, но не определён на дне ущелья

$$F_{10} = (x_1 + 10)^2 + [1000 + (x_1 + 10)^2] \cdot \sqrt{\sum_{k=2}^n \left(x_k + 10 \sin \frac{k \cdot x_1}{k-1} \right)^2}.$$

11. Функция с извилистым ущельем, у которой градиент неограниченно возрастает при приближении ко дну ущелья

$$F_{11} = (x_1 + 10)^2 + [1000 + (x_1 + 10)^2] \cdot \sqrt[4]{\sum_{k=2}^n \left(x_k + 10 \sin \frac{k \cdot x_1}{k-1} \right)^2}.$$

Обычно программа MINUIT устанавливается с ограничением $n \leq n_{max} = 50$ на число переменных параметров функции. Для того, чтобы проверить успешность работы программы MINUIT на этих функциях, была специально проведена установка программы MINUIT с бóльшим ограничением $n_{max} = 200$ (тот же самый Release 96.03). В табл. 6 приведены результаты минимизации этих четырёх функций программой MINUIT, а также программой BUKMING в режиме, устанавливаемом по умолчанию. Как и ожидалось, первые две функции оказались вполне по силам методу переменной метрики пространства.

В таблице 7 приведены результаты минимизации этих функций программой BUKMING в разных режимах.

Для пространства большой размерности метод Ньютона оказался не такой успешный, как для пространства малой и средней размерности. Это объясняется тем, что вычисление матрицы квадратичной формы вдали от минимума при большой размерности страдает от накопления ошибок округления, кроме того, для вычисления матрицы и совершения очередного шага в сторону минимума требуется вычислить гораздо больше опорных значений функции, чем в методе переменной метрики пространства. Для пространства большой размерности тоже довольно успешны оказываются сочетания алгоритмов программы $VMM \rightarrow \text{Newton}$ и $VMM \rightarrow \text{Newton} \rightarrow \text{Simplex}$.

Таблица 7: Результаты минимизации тестовых функций с большим числом параметров $n = 100$ в виде F_{min}/N_{cal} с помощью разных сочетаний алгоритмов и разных стратегиях Istr. Ограничение на число вызовов функции FCN установлено $N_{cal} < 10^6$, требование на точность минимизации $\Delta F_m = 10^{-3}$.

Набор алгор.	Результаты для разных функций			
	8	9	10	11
Istr=0				
VMM	$1.1 \cdot 10^{-26}/301$	0.879/420	0.752/1445	2.84/1853
CDM	578.5/1239	0.825/3923	1.18/3467	4.10/5674
Simplex	$3.7 \cdot 10^7/36334$	2951.1/39311	1431.3/21618	1130.8/19440
Newton	0.706/302770	92.0/116893	98.14/1853	100.2/1874
T&F	247649.0/34196	97.0/12428	97.0/14832	97.0/18450
Random	3750.1/311929	431.6/17873	753.1/17838	592.2/17781
Istr=1				
VMM	$1.1 \cdot 10^{-26}/301$	$5.7 \cdot 10^{-13}/2485$	$7.9 \cdot 10^{-5}/26243$	$0.094/10^6$
CDM	$1.4 \cdot 10^{-18}/3510$	$5.2 \cdot 10^{-12}/55150$	$7.5 \cdot 10^{-8}/332080$	0.694/30458
Simplex	$2.5 \cdot 10^7/10^6$	$88.5/10^6$	$94.2/10^6$	$91.7/10^6$
Newton	$0.135/10^6$	$13.4/10^6$	$2.07/10^6$	$3.38/10^6$
VMM ↓ Newton	$1.1 \cdot 10^{-26}/301$	$5.7 \cdot 10^{-13}/2485$	$8.7 \cdot 10^{-7}/66615$	$0.104/10^6$
VMM ↓ CDM	$1.1 \cdot 10^{-26}/301$	$5.7 \cdot 10^{-13}/2485$	$5.1 \cdot 10^{-7}/59516$	$0.042/10^6$
VMM ↓ Simplex	$1.1 \cdot 10^{-26}/301$	$5.7 \cdot 10^{-13}/2485$	$6.3 \cdot 10^{-4}/10^6$	$0.217/10^6$
Istr=2				
VMM	$1.1 \cdot 10^{-26}/2916$	$3.5 \cdot 10^{-14}/355078$	$7.9 \cdot 10^{-5}/26243$	$0.094/10^6$
CDM	$7.6 \cdot 10^{-23}/6448$	$4.0 \cdot 10^{-18}/52638$	$6.9 \cdot 10^{-6}/10^6$	0.694/30458
Newton	$0.135/10^6$	$13.44/10^6$	$2.07/10^6$	$3.38/10^6$
VMM ↓ CDM	$1.1 \cdot 10^{-26}/3096$	$8.4 \cdot 10^{-21}/435121$	$5.1 \cdot 10^{-7}/59516$	$0.042/10^6$
VMM ↓ Newton	$1.1 \cdot 10^{-26}/4838$	$6.1 \cdot 10^{-15}/242502$	$8.7 \cdot 10^{-7}/66615$	$0.104/10^6$
VMM ↓ Simplex	$1.1 \cdot 10^{-26}/897261$	$9.6 \cdot 10^{-21}/322447$	$6.3 \cdot 10^{-4}/10^6$	$0.217/10^6$

Istr=3				
VMM	0/4069	4.1 $10^{-22}/102890$	$8.6 \cdot 10^{-7}/72927$	0.697/67766
CDM	$7.4 \cdot 10^{-21}/6292$	2.5 $10^{-17}/123600$	0.531/420457	32.0/96267
Newton	$0.530/10^6$	$0.495/10^6$	54.59/125382	98.18/70053
VMM ↓ CDM	0/4249	1.3 $10^{-21}/261934$	$8.8 \cdot 10^{-6}/193344$	0.778/143618
VMM ↓ Newton	0/5995	$2.8 \cdot 10^{-22}/29331$	$2.5 \cdot 10^{-6}/125979$	0.658/192403
VMM ↓ Simplex	0/893241	$4.0 \cdot 10^{-22}/10^6$	$0.750/10^6$	$0.636/10^6$

12 Заключение

Разработана программа BUKMING безусловной численной минимизации функции многих переменных с использованием аналитически вычисляемого градиента. Предполагается её использование при обработке экспериментальных данных на детекторах СНД, КМД, КЕДР. Для достаточно простых функций качество минимизации этой программы сравнимо с качеством широко известной программы MINUIT. Сравнительные достоинства и недостатки этой программы можно кратко сформулировать следующим образом.

Достоинства:

- Программа написана на языке Фортран и не использует никаких специальных библиотек. Несмотря на большое количество алгоритмов минимизации в программе BUKMING (переменной метрики пространства, метод сопряжённых направлений, метод Ньютона, симплекс-метод, метод проб и ошибок, метод случайного поиска), которые можно использовать в любых сочетаниях, программа достаточно компактна — около 3000 строк исходного кода. Это означает, что необходимым и достаточным условием использования данной программы является наличие компилятора Fortran-77.

- Простая модульная организация программы позволяет легко добавлять новые алгоритмы минимизации.
- Некоторые сложные функции программа BUKMING минимизирует более надёжно, чем MINUIT. Это даёт возможность даже в тех случаях, когда MINUIT доступна для использования, проверить качество минимизации альтернативной программой.

Недостатки:

- Любая новая программа, не отличающаяся кардинально от ранее написанных, имеет естественный недостаток — малый опыт эксплуатации, ненулевая вероятность присутствия какой-либо ошибки. Правда, этот недостаток со временем может исчезнуть.
- Следующий недостаток является продолжением достоинства — компактность и простота программы обуславливает отсутствие многих приятных функций программы MINUIT. Это отсутствие процедуры оценки статистических ошибок для случая минимизации логарифмической функции правдоподобия, невозможность нарисовать карту линий уровня на двухкоординатном срезе профиля функции, невозможность ограничить область изменения параметров и т.п. Часть этих недостатков можно компенсировать, построив свою аналогичную процедуру, используя для минимизации программу BUKMING, но в любом случае это дополнительная работа.
- Отсутствие идентичной версии программы на популярном в настоящее время языке C++ существенно ограничивает возможность её использования.

Этот препринт можно рассматривать как подробное описание программы.

Список литературы

- [1] *F.James, M.Roos.* “MINUIT” a system for function minimization and analysis of the parameter errors and correlations. *Computer Physics Communications* 10 (1975) 343–367.
- [2] *А.Д. Буккин.* Программа численной минимизации функции многих параметров. Препринт ИЯФ 2004–78, Новосибирск, 2004.
- [3] *M.N.Achasov, V.M.Aulchenko, S.E.Baru et al.* Spherical Neutral Detector for VEPP-2M collider. *Nucl. Instr. and Meth. A* 449 (2000) 125–139. e-Print: hep-ex/9909015.
- [4] *R.Fletcher, M.J.D.Powell.* A rapidly converging descent method for minimization. *Comput. J.* 6 (1963) 163
- [5] *А.Д. Буккин, С.И. Эйдельман.* ЭВМ в планировании и обработке эксперимента. Учеб. пособие. Изд. 2-е. Новосиб. гос. ун-т, Новосибирск, 2002.
- [6] *A.I.Manevich, E.Boudinov.* An efficient conjugate directions method without linear minimization. *Nucl. Instr. and Meth. in Phys. Research A* 455 (2000) 698–705.
- [7] *J.A.Nelder, R.Mead.* A simplex method for function minimization. *Comput. J.* 7 (1965) 308

А.Д. Бужин

**Программа численной минимизации
функции многих параметров с использованием
аналитически вычисляемого градиента**

A.D. Bukin

**Subroutine for numerical minimization
of function of many parameters
using analytically evaluated gradient**

ИЯФ 2005-43

Ответственный за выпуск А.М. Кудрявцев

Работа поступила 22.09.2005 г.

Сдано в набор 22.09.2005 г.

Подписано в печать 23.09.2005 г.

Формат бумаги 60×90 1/16 Объем 2.0 печ.л., 1.6 уч.-изд.л.

Тираж 100 экз. Бесплатно. Заказ № 43

Обработано на IBM PC и отпечатано на
роталпринте ИЯФ им. Г.И. Будкера СО РАН

Новосибирск, 630090, пр. академика Лаврентьева, 11.