

А.Д. Букин, С.И. Эйдельман

ЭВМ в физическом эксперименте

Новосибирск
2007?

УДК 519.28

Букин А.Д., Эйдельман С.И. **ЭВМ в физическом эксперименте.**
Учеб. пособие. Новосибирск, 2007?. 211 с.

Основная цель данного учебного пособия – ознакомить студентов вузов с элементами моделирования экспериментов и обработки экспериментальных данных. Приведены многочисленные примеры решения часто встречающихся на практике задач, которые могут быть полезными как для студентов, так и начинающих физиков-экспериментаторов.

Книга содержит 46 рис., 13 таблиц, библи. 79 назв.

Contents

Предисловие	7
1 Введение в операционную систему UNIX	9
1.1 Синтаксис командного языка	10
1.2 Файловая система	12
1.3 Редактор текстов	15
1.3.1 Основные команды редактора vi	15
1.3.2 Редактор emacs	17
1.4 Компиляция и запуск на счёт фортрановских программ	18
1.5 Командные процедуры	20
1.6 Получение справочной информации	21
2 Интерполяция и аппроксимация	23
2.1 Интерполяция полиномами	24
2.2 Интерполяция сплайнами	25
2.3 Аппроксимация экспериментальных данных	28
2.3.1 В-сплайны	29
2.3.2 Использование средне-квадратичной кривизны при аппроксимации экспериментальных данных	30
3 Метод Монте – Карло	33
3.1 Случайные величины. Общие характеристики распределения вероятностей	33
3.2 Генераторы некоторых стандартных распределений	40
3.2.1 Равномерное распределение	40
3.2.2 Распределение Гаусса	44
3.2.3 Распределение Пуассона	46

3.2.4	Биномиальное распределение	47
3.2.5	Экспоненциальное распределение	47
3.2.6	Распределение Коши	48
3.2.7	Основные характеристики распределений вероятности	48
3.3	Получение произвольного распределения случайной величины	48
3.3.1	Метод прямой выборки	50
3.3.2	Метод Неймана (реджекций)	51
3.3.3	Метод существенной выборки	51
3.3.4	Метод композиций	52
3.3.5	Метод композиций и реджекций	53
3.4	Генерация нескольких случайных величин при наличии корреляций	57
3.5	Интегрирование методом Монте – Карло	59
3.5.1	Метод среднего	59
3.5.2	Выделение главной части	60
3.5.3	Метод существенной выборки	61
3.5.4	Метод Метрополиса	61
3.6	Моделирование в ядерном эксперименте на примере регистрации событий в детекторе	65
4	Разработка программ прохождения излучения через вещество	71
4.1	Программы моделирования столкновения частиц в установках со встречными пучками	71
4.2	Моделирование взаимодействия частиц с веществом детектора	72
4.2.1	Непрерывные взаимодействия	72
4.2.2	Ионизационные потери заряженных частиц	73
4.2.3	Множественное рассеяние заряженных частиц	77
4.2.4	Точечные взаимодействия	77
4.2.5	Тормозное излучение	77
4.2.6	Рождение δ -электронов	77
4.2.7	Рождение электрон-позитронных пар	77
4.2.8	Комптон-эффект	77
4.2.9	Фотоэффект	78
4.3	Моделирование отклика детектора	78
4.4	Некоторые технические приемы	78

5	Численные методы минимизации функций нескольких переменных	79
5.1	Одномерная минимизация	81
5.2	Метод сеток	83
5.3	Случайный поиск	83
5.4	Метод последовательной минимизации	84
5.5	Метод Розенброка	84
5.6	Метод проб и ошибок	86
5.7	Симплекс-метод	88
5.8	Метод наискорейшего спуска	89
5.9	Метод Ньютона	90
5.10	Метод сопряжённых направлений	92
5.11	Метод сопряжённых градиентов	97
5.12	Метод переменной метрики пространства	98
5.13	Критерий нахождения минимума	102
5.14	Поиск глобального минимума	102
6	Определение параметров теоретической модели по экспериментальным данным	105
6.1	Метод максимального правдоподобия	105
6.2	Метод наименьших квадратов	113
6.3	Усреднение экспериментальных результатов	114
6.4	Пример нестандартной оценки параметра	117
6.5	Кинематическая реконструкция событий	118
6.5.1	Итерационный метод решения	119
6.5.2	Использование численных методов условной минимизации	120
6.5.3	Исключение «лишних» переменных	120
6.5.4	Реконструкция трёх-частичных событий	128
6.5.5	Реконструкция четырёх-частичных событий	136
7	Доверительные интервалы. Установление верхнего предела	139
7.1	Определение среднего для распределения Гаусса	140
7.2	Среднее время жизни	140
7.3	Измерение параметров Гауссова распределения	146
7.4	Среднее значение Пуассонова распределения	151
7.5	Верхний предел на параметр распределения Пуассона	156
7.6	Подгонка резонансной кривой	160

8	Разделение событий	165
8.1	Разделение событий двух классов	165
8.2	Использование корреляционной матрицы для разделения событий	167
9	Проверка гипотез. Критерии согласия теории с экспериментальными данными	177
9.1	Критерий χ^2	177
9.2	Критерий знаков	182
9.3	Распределение Стьюдента	183
9.4	Критерий Колмогорова	186
9.5	Критерий Мизеса – Смирнова	189
10	Система аналитических вычислений REDUCE	193
10.1	Числа, переменные, структура команд	193
10.2	Выражения	195
10.3	Операторы присвоения и подстановки	197
10.4	Групповые и составные операторы	198
10.5	Условные операторы, операторы цикла и массивы	199
10.6	Матрицы	200
10.7	Дифференцирование и интегрирование	201
10.8	Режимы работы	201
10.9	Некоторые дополнительные команды	202
	Список литературы	205

Предисловие

Данное пособие написано по материалам практикума, ведущегося в течение нескольких лет на кафедрах физико-технической информатики и физики элементарных частиц физического факультета НГУ. Цель практикума — познакомить студентов с некоторыми типичными задачами и способами их решения, с которыми сталкиваются в своей работе физики-экспериментаторы, точнее, экспериментаторы в области физики элементарных частиц или, как её принято сегодня называть, физики высоких энергий. Сложность экспериментов в физике высоких энергий определила необходимость интенсивного использования компьютеров при математическом моделировании экспериментов и обработке полученных экспериментальных данных.

С самого начала стало ясно, что в силу специфичности обсуждаемых задач заметно повысится эффективность практикума, если он будет сопровождаться лекциями, на которых студенты получают основы необходимой теоретической информации.

Мы не ставили целью дать подробное изложение вопросов математической обработки экспериментальной информации. Авторам скорее хотелось ввести студентов-информатиков в круг проблем, возникающих перед физиками-экспериментаторами, с которыми им приходится работать.

Это и определило круг рассматриваемых вопросов. Помимо собственно обработки информации мы сочли нужным дать здесь некоторые начальные сведения о работе с операционной системой UNIX, широко используемой в Институте ядерной физики СО РАН, а также о системе аналитических вычислений REDUCE. Ещё один важный вопрос для практикума — это язык программирования. Мы выбрали Фортран-

77. При создании программного обеспечения для физики высоких энергий традиционно использовался язык программирования Фортран. В последнее время появился успешный опыт использования объектно-ориентированного языка C^{++} (коллаборация детектора *VaBar* в США), но преимущество использования этого языка при создании больших программных комплексов по сравнению с Фортраном небесспорно. Возможно, если будет дальнейшее продвижение в этом направлении, и будет реализован принцип объектно-ориентированного программирования и «инкапсуляции данных» более последовательно, чем в C^{++} , то ситуация станет более простой. Фортран по-прежнему остаётся одним из основных средств при создании математического обеспечения экспериментальной физики высоких энергий. Накоплен большой опыт, многие методы реализованы в программах и подпрограммах, имеющихся в библиотеках стандартных научных программ в вычислительных центрах ведущих физических лабораторий. Ещё одно важное обстоятельство в пользу применения Фортрана — это то, что работа в большом коллективе, характерная для этой области физики, часто требует разобраться в программе, написанной коллегой из другого института или даже государства, внести в неё изменения. Опыт показывает, что Фортран вполне адекватен и с этой точки зрения. Существует множество учебников и монографий, подробно описывающих язык Фортран (можно порекомендовать, например, книги [1, 2]), так что в данном пособии нет необходимости останавливаться на описании языка.

Специально отметим, что это не конспект лекций или описание задач практикума, а, скорее, краткое руководство к действию, содержащее минимум необходимой информации для решения задач, возникающих при обработке эксперимента. Некоторые вопросы освещены более подробно, что отражает вкусы и личные пристрастия авторов. Всюду, где возможно, мы старались привести список литературы, доступной студенту по уровню сложности, и которая помогла бы ему получить более глубокую и подробную информацию о методах обработки экспериментальной информации. Многие примеры могут оказаться полезными и для начинающих физиков-экспериментаторов.

Chapter 1

Введение в операционную систему UNIX

Бурное развитие вычислительной техники, конкурентная борьба разных фирм-производителей и фирм-разработчиков операционных систем привели к последовательному появлению, расцвету и отмиранию нескольких типов операционных систем. Вначале на мощных машинах фирмы IBM получили распространение операционные системы VM/CMS (советские аналоги назывались СВМ – система виртуальных машин), затем их потеснили машины VAX и, соответственно, операционные системы VAX/VMS, и наконец, в последние годы начался расцвет операционной системы UNIX. Существует множество вариантов UNIX-подобных операционных систем разных фирм (SunOS, Solaris, IRIX, AIX и т.д.), которые могут отличаться, оставаясь всё же очень похожими друг на друга. На персональных компьютерах тоже было большое разнообразие операционных систем. На машинах одного из крупнейших производителей – фирмы IBM – сменились MS-DOS, OS/2, Windows, Windows-NT, Windows-2000. В последнее время получила распространение UNIX-подобная некоммерческая система LINUX. Трудно предсказать, долго ли продержится «эра UNIX», скоро ли появится для неё мощный конкурент.

Операционная система UNIX описывается в множестве книг. Можно прочитать классическую книгу [4], но лучше всего пользоваться той книгой, которая описывает систему, установленную на машине, на которой вы собираетесь работать. Если есть желание почувствовать разницу в общении с разными вариантами системы UNIX, то можно обратиться к книге,

например, [5], где рассматриваются особенности работы в разных версиях UNIX. Для пользователей, имеющих большой опыт работы с системой VAX/VMS, может оказаться интересной книга [6]. Здесь мы рассмотрим только основные принципы и команды системы UNIX. Первое отличие от VAX/VMS, которое сразу бросается в глаза, – большие и маленькие буквы считаются различными (в идентификаторах, паролях, именах директорий и файлов). Для удобства будем далее писать прямым шрифтом сообщения компьютера, курсивом – то, что обязан ввести пользователь. Сеанс работы начинается с идентификации пользователя:

```
login: username
password: password
```

1.1 Синтаксис командного языка

Если ввод идентификатора и пароля прошёл нормально, то после некоторых информационных сообщений системы появляется приглашение к команде. Обычно это приглашение задаётся системными программистами так, чтобы было понятно, на какой машине идёт работа (UNIX – система многопользовательская и многозадачная, кроме того, в «окнах», которые создаются для работы, можно по протоколу TCP/IP открыть сессию связи с какой-нибудь другой машиной. В общем, удобно видеть, с какой машиной идёт работа в каждом окне.) Далее в примерах команд для определённости будем использовать приглашение в виде «sky%».

Командная строка состоит из команды, модификаторов и параметров. Команду следует вводить полностью, без сокращений и в нужном регистре (большинство команд – маленькими буквами).

Модификаторы команд, или ключи, вводятся после пробела с начальным символом «-». Далее через пробелы перечисляются параметры. Рассмотрим пример.

Команда вывода списка файлов может быть выдана в нескольких вариантах:

```
ls
ls -l
ls -a
ls -l -a /usr
ls -la /usr
```

Вообще, у этой команды имеются десятки ключей. Последние два варианта использования команды «ls» приводят к одинаковому результату, то есть ключи можно объединять в группы.

В системе UNIX есть важное понятие *конвейера* (в английской терминологии *pipeline*). Если в командной строке подряд написаны две команды, разделённые символом «|» (с пробелами), то выходная информация первой команды является входной информацией для второй команды. Пусть список файлов в директории, высвечиваемый командой «ls», очень длинный, и нет возможности увидеть начало списка, быстро уходящего за пределы экрана. С другой стороны, пусть мы привыкли просматривать содержимое текстовых файлов командой «more» или «most» в виде

```
more filename
```

Тогда мы можем использовать *конвейер* для совместного использования двух команд:

```
ls | more
```

Работа этих двух команд выглядит так, как-будто результаты команды «ls» записаны в файл, который затем просматривается командой «more». Тем не менее никакого файла в директории не появляется (и не «расходуется» дисковая квота).

Имеется несколько интерпретаторов команд (к сожалению, такое разнообразие трудно считать благом для пользователя): так называемые *оболочки* B-shell, C-shell, TC-shell. Интерпретация команд и исполнение в разных оболочках несколько различаются. Пользователь может переключиться на любой интерпретатор команд с помощью команд bsh, csh, bash и tcsh. Обычно в интерактивном режиме по умолчанию запускается tcsh или bash.

Вместо команды может быть написано имя исполняемого модуля, например, имя исполняемого модуля, полученного компиляцией и редактированием связей вашей программы на языках Фортран или С. Тогда эта программа запускается на исполнение.

Для прекращения сеанса работы используется команда

```
exit
```

1.2 **Файловая система**

Как и во всех современных системах, в UNIXе файловая система организована в виде древовидной системы директорий. В составе каждой директории могут быть как файлы, так и другие (вложенные) директории. Команда «ls -l» высвечивает список файлов в текущей директории с перечислением всех атрибутов, например:

```
sky% ls -l
```

```
drwxr-xr-x    2 bukin    d1          4096 Nov  9 08:24 ILU10
drwxr-xr-x    2 bukin    d1           9 Aug 26 1996 PAW
drwxr-xr-x    2 bukin    d1          4096 Oct  3 08:14 PHIPSI
lrwxr-xr-x    1 root     root         22 Oct 17 16:26 WWW -> /var/httpd/h
tdocs/main
-rw-r--r--    1 bukin    d1           619 Sep 16 1997 calres.dat
-rw-r--r--    1 bukin    d1           394 Jan 14 1999 ddgaus.for
-rw-r--r--    1 bukin    d1          3336 Jan 14 1999 ddgaus.o
-rw-r--r--    1 bukin    d1          2397 Jul 23 1999 fitabomeps.for
drwx-----    2 bukin    d1           106 Nov  4 08:40 mail
drwxr-xr-x    3 bukin    d1          4096 Jan 31 2000 phi96
-rw-r--r--    1 bukin    d1          3750 Jan 30 1998 split.f
drwxr-x---    2 bukin    d1           9 Oct 17 22:38 spool
-rw-r--r--    1 bukin    d1           163 May 23 1997 tes1.f
drwxr-xr-x    2 bukin    d1         20480 Nov  3 19:05 unimod
```

Каждая строка здесь соответствует одному файлу (кроме тех случаев, когда строка очень длинная и часть информации переносится на следующую строку).

Самый первый символ в строке характеризует тип информации в данном файле:

d — файл является директорией,

l — link, ссылка на другой файл (или директорию), создаётся командой «ln»,

- — обычный файл.

Следующие 9 символов образуют список прав доступа к файлу трёх категорий пользователей (на каждую категорию пользователей – три символа): владельца файла, группе пользователей, к которой принадлежит владелец, и всех прочих пользователей. Отказ в праве доступа обозначается символом «-». Разрешение на полный доступ к файлу обозначается группой

из трёх символов «гwx». Разрешение на чтение «г» позволяет пользователю просматривать содержимое файла (или доступ к вложенным файлам для директории). Разрешение на запись «w» позволяет писать в файл и изменять его (для директории это соответствует созданию и удалению файлов в этой директории). Разрешение на исполнение «x» позволяет исполнять бинарные программы и командные файлы (для директории это означает возможность просматривать список файлов). Владелец файла может менять права доступа командой «chmod», описание команды можно прочитать в любой книге по UNIXу или в интерактивной справочной информации (см. разд. 1.6). Полностью, например, такая строка

```
-rw-r--r--    1 bukin    d1          619 Sep 16  1997 calres.dat
```

описания файла расшифровывается следующим образом:

1. Это обычный файл.
2. Владелец файла может читать и модифицировать файл, но не может исполнять.
3. Члены группы могут только читать файл.
4. Все остальные пользователи тоже могут только читать файл.
5. Количество ссылок на файл равно 1 (на один и тот же файл может быть несколько ссылок, и файл физически удаляется только в том случае, если удалены все ссылки).
6. Владелец файла – пользователь с именем bukin.
7. Правами группы обладают пользователи группы d1 (владелец принадлежит к группе d1).
8. Длина файла 619 байт.
9. Файл создан 16 сентября 1997 года.
10. Имя файла «calres.dat».

Для того чтобы перейти в другую директорию (сменить текущую директорию), используется команда «cd». Для текущей директории в вышеприведённом примере можно перейти во вложенную директорию командой

```
cd unimod
```

Самая главная директория (корневая) обозначается символом «/», то есть в неё можно перейти командой

```
cd /
```

Команда «cd» без параметров воспринимается как команда перехода в главную директорию пользователя (у каждого пользователя есть своя главная директория, которая создаётся системным программистом при регистрации пользователя, в документации на английском языке она обычно называется «home directory»). В главную директорию любого другого пользователя можно перейти (если позволяют права доступа) командой вида

```
cd ~идент_пользователя
```

например,

```
cd ~bukin
```

Узнать полное имя текущей директории можно командой «pwd», например:

```
sky% pwd
/u/d1/bukin
```

Полное имя файла, на которое можно ссылаться из любой директории, составляется из списка всех директорий, в которые он вложен, начиная с корневой, разделённых символом «/». Имя файла или директории может состоять из последовательности символов (не более 256), исключая «/». Символ «.» тоже может использоваться, причём может повторяться в имени несколько раз. Однако, если имя начинается с точки, то такой файл считается «спрятанным», системным. Команда «ls» такие файлы не показывает, и только если использовать форму команды «ls -a», тогда высвечивается весь список файлов в директории.

При использовании имени файла, символ «*» заменяет любое число последовательных символов, символ «?» – заменяет один любой символ. Поэтому команда

```
ls *.f
```

означает: высветить список файлов в текущей директории, имена которых заканчиваются на «.f».

Удалять файлы можно командой «rm», копировать – командой «cp», переименовывать и перемещать из директории в директорию – командой «mv». Создать новую директорию можно командой «mkdir», удалить – «rmdir».

Создать файл можно текстовым редактором или копированием уже существующего файла. Также можно создать файл командой «cat» следующим образом:

```
sky% cat >имя_файла
```

Далее вводятся с клавиатуры строки файла (последовательно), процесс ввода заканчивается командой:

```
dctrl
```

```
sky%
```

Этой же командой «cat» можно вывести на экран содержимое существующего файла. Основное же назначение этой команды – объединение файлов в один файл, например:

```
sky% cat file1.txt file2.txt file3.txt > newfile.txt
```

Здесь использована команда «>» направления стандартного вывода в файл. При этом, если такого файла не обнаружено, то он создаётся. Если в существующий файл стандартный вывод команды надо дописать в конец файла, то используется команда “>>”, например:

```
sky% cat file1.txt file2.txt file3.txt >> newfile.txt
```

Полезной командой при работе с файлами является команда «grep». Однако она не может претендовать на включение в минимальный набор команд, поэтому здесь мы её не рассматриваем. Желающие могут изучить её описание в книгах по операционной системе UNIX или получить описание в справочной системе, как описано в разд. 1.6.

Операционная система рассматривает любой файл как непрерывную последовательность символов (байтов). Какую-либо внутреннюю структуру файла могут воспринимать только специализированные программы, например, редакторы текстов.

1.3 Редактор текстов

Собственно стандартным экранным редактором текстов в системе UNIX является редактор «vi», очень простой и поэтому недостаточно удобный. Однако, у него есть неоспоримое преимущество перед другими редакторами – он есть в любой версии UNIX, на любой машине.





1.3.1 Основные команды редактора vi

Для того чтобы создать или редактировать существующий файл с именем *filename*, надо использовать команду

```
vi filename
```

При этом запускается экранный редактор **vi** в командном режиме. Пустые строки файла выглядят как

~

На самом деле в файле этих строк нет. В командном режиме можно перемещать курсор по тексту файла с помощью клавиш , , , . Команды в этом режиме состоят из одной или двух букв. Для перемещения курсора больше, чем на одну позицию, можно использовать команды:

w — на начало следующего слова,

b — на начало предыдущего слова,

0 — на начало текущей строки,

\$ — на конец строки,

F^{ctrl} — на экран вперед,


B^{ctrl} — на экран назад,

G — на конец файла,

1G — на первую строку файла,

12G — на двенадцатую строку файла.

Для удаления символа, находящегося под курсором, используется команда «x». Конец строки при этом сдвигается на место удалённого символа. Для удаления всей строки используется команда «dd». Для удаления слова, на котором стоит курсор, используйте команду «dw». Кроме того, команду удаления «d» можно комбинировать с командами перемещения курсора, например, «d\$» – удаление от курсора до конца строки, «dG» – удаление от курсора до конца файла. Замена одного символа – команда «r» (следом вводится один символ).

Кроме командного режима имеется режим вставки произвольного текста. Для перехода в этот режим используются команды «i» или «a». Командой «i» вставляют текст перед тем символом, на котором стоит курсор, а командой «a» – после. Закончить режим вставки и вернуться в командный режим можно клавишей .

Ещё один режим работы редактора (и последний) – «режим последней строки». Это тоже командный режим, переход в который осуществляется командой «:». Курсор при этом уходит на последнюю строку экрана и ждёт

продолжения команды (из одной строки). Окончанием команды является клавиша Enter. Основными командами в этом режиме являются команды выхода из редактора:

:wq — выход с сохранением результатов редактирования,

:q! — выход без сохранения результатов.

Командой «r» в этом режиме можно вставить в текст содержимое другого файла с именем *another_file*:

```
:r another_file
```

Командой «r!» можно вставить вывод указанной команды в файл, начиная с места, где находился курсор, например,

```
:r! ls -l
```

вставит в файл содержимое текущей директории.

1.3.2 Редактор emacs

Один из популярных редакторов текстов в среде UNIX – редактор **emacs**. По своим возможностям он гораздо мощнее редактора **vi**. Описание даже достаточно скромного подмножества команд выходит за рамки данной книги, бóльшая часть активных пользователей этого редактора, по-видимому, знает не больше половины всего набора команд этого редактора. Вариант этого редактора, работающий с X-приложениями, – **xemacs**, не подразумевает запоминание большого количества команд, основные команды доступны в *меню* и выбор команд может осуществляться с помощью мыши.

Описание редактора в Интернете можно найти, стартуя с адресов

<http://www.emacs.org>

или

<http://www.fsf.org>

Последний адрес может привести также к FTP-серверам, с которых можно скопировать бесплатно распространяемые версии emacs. О масштабе редактора можно судить по 600-страничному руководству [7].

Работая в редакторе emacs, можно получать интерактивно некоторую справочную информацию. Команда запроса информации начинается как h^{ctrl} , а дальнейшее продвижение зависит от версии редактора и установок в конфигурационных файлах. По-видимому, минимальные знания о командах независимо от версии можно свести к следующему набору:

$x^{ctrl}c^{ctrl}$ — выход из редактора. Если в редактируемый файл вносились изменения, то последует вопрос, надо ли сохранять изменения на диск.

g^{ctrl} — отмена незавершённой команды редактора.

$x^{ctrl}o$ — переход в другое окно. Экран может быть разделён на несколько окон. С помощью этой команды можно циклически делать активными разные окна.

$h^{ctrl}i$ — переход к чтению структурированного описания команд.

Кроме этого весьма сложного редактора могут быть установлены редакторы промежуточного уровня сложности: jed, joe и т.д. Как уже было сказано, все они (кроме vi) являются необязательным элементом в UNIX.

1.4 Компиляция и запуск на счёт фортрановских программ

Несмотря на то что реализация компилятора Фортран-77 у разных производителей может существенно различаться (в модификаторах команды компиляции и наборе расширений стандарта Фортрана-77), всё же в большинстве случаев компиляция программы возможна командой

```
sky% f77 fort_file_name -o output_module
```

где *fort_file_name* — имя файла, содержащего программу на языке Фортран-77 (лучше всего, если это имя заканчивается на «.f» или «.for»), *output_module* — имя, под которым надо записать исполняемый модуль. Если ключ «-o» с именем исполняемого модуля не написать, то будет использовано имя «a.out». Уточнить работу команды «f77» можно в соответствии с рекомендациями в разд. 1.6.

Запуск на счёт полученной программы (при условии, что не было зафиксировано ошибок при компиляции) производится просто вводом имени исполняемого модуля, например,

```
sky% a.out
```

или

```
sky% ./a.out
```

При этом до окончания счёта клавиатура не может быть использована для других работ (просмотр почты, редактирование файлов и т.п.). Если выполнение программы занимает значительное время, то это очень

неудобно. Можно запустить программу в фоновом режиме:

```
sky% a.out > job.log &
[1] 20153
sky%
```

Здесь вывод результатов программы в файл «job.log» заказан с той целью, чтобы вывод данных при исполнении не помешал другой нашей текущей работе. Исполнение в фоновом режиме определяет символ «&» в конце строки. При запуске задачи в фоновом режиме система откликнулась одной информационной строкой, в которой она в квадратных скобках сообщила номер работы [1] и уникальный для всей системы номер нового процесса. Теперь по команде «jobs» система будет выдавать список текущих работ пользователя, где будут использоваться номера работ (тоже в квадратных скобках), а по команде «ps» – список процессов данного пользователя. Последний список гораздо больше предыдущего и в нём уже используются номера процессов, т.е. в данном примере фоновый процесс появится в списке с номером 20153.

Любой из ваших процессов можно аварийно завершить командой «kill». В данном примере это могло бы выглядеть так:

```
sky% kill -9 20153
```

Если вы не хотите мешать другим своим (и чужим) работам, отнимая у них процессорное время, то можно запустить программу на исполнение с пониженным приоритетом командой «nice»:

```
sky% nice a.out > job.log &
[1] 20153
sky%
```

Для проверки, как идёт задание, сколько процессорного времени оно потребляет, в частности, в сравнении с другими заданиями, можно использовать команду «top», которая выводит список процессов, упорядоченный по потреблению процессорного времени. Прервать исполнение этой команды можно командой «*c^{ctrl}*».

Если прекратить сессию командой «exit», то все незавершённые задания аварийно завершатся, даже если им не надо выводить на экран какую-либо информацию. Для того чтобы запущенное задание выполнялось независимо от порождающего процесса и не прерывалось аварийно при вводе команды «exit», используется следующий вариант команды запуска:

```
sky% nohup nice a.out > job.log &
[1] 20153
sky%
```

Несмотря на свою «независимость», задание доступно команде «kill».

Кроме перечисленных возможностей запуска задания, на машинах под ОС UNIX часто устанавливают систему управления пакетными заданиями (batch-queue). Особенность таких систем заключается в том, что задание, поставленное в такую очередь, не обязательно сразу начинает исполняться. Иногда в таких системах поддерживается некоторая система приоритетов, и тогда может быть так, что задание не только не начало сразу исполняться, а даже может пойти на исполнение позже задания другого пользователя, которое было позже поставлено в очередь. Кроме того, если несколько машин объединены в так называемый *кластер*, то часто в такой системе нельзя предсказать, на какой конкретной машине будет исполняться задание из пакетной очереди. Поскольку на практике используют разные системы управления пакетными очередями, нет смысла описывать здесь какую-либо одну из них. Получить список команд, имеющих отношение к пакетной очереди в вашей системе, можно с помощью команды

```
sky% man -k batch
```

(см. разд. 1.6).

1.5 Командные процедуры

В ОС UNIX есть возможность создавать командные процедуры — файлы с группой команд на языке одного из интерпретаторов: B-shell, C-shell, TC-shell, BASH и т.д. Традиционно в практике использования UNIX их называют скриптами (англ. script). Мотивы для создания таких процедур очевидны: удобно часто повторяемые последовательности команд оформлять в виде отдельных файлов, которые впоследствии можно исполнять в интерактивном или фоновом режиме. Кроме того, есть некоторые командные процедуры с зарезервированными именами: «.login», «.cshrc», «.bashrc». Процедура «.login» запускается при входе пользователя в систему, а конфигурационные файлы типа «.cshrc» и «.bashrc» работают при запуске соответствующего интерпретатора (C-shell, BASH). Как уже говорилось, команды в разных интерпретаторах отличаются. Поэтому при разработке своей командной процедуры следует пользоваться описанием набора команд именно того интерпретатора, который вы выбрали для своей процедуры. В первой строке процедуры можно для надёжности указать, какой интерпретатор должен работать. Здесь мы не будем описывать правила написания командных процедур для каждого интерпретатора, потому что эту информацию нельзя отнести к насущно необходимой при использовании компьютера для обработки данных или моделирования

эксперимента.

1.6 Получение справочной информации

Получение интерактивной справочной информации в UNIXе является его слабым местом (по сравнению с VAX/VMS, конечно). Для получения справки по какой-либо команде *command_name* надо ввести команду

```
sky% man command_name
```

Неудобство такого способа получения информации, в первую очередь, заключается в том, что пользователь должен знать точное имя команды. В предыдущих разделах о некоторых командах было сказано только их назначение и имя в расчёте на использование команды *man*.

Если точное имя команды неизвестно, но есть идеи, что́ может быть ключевым словом для этой команды, то можно получить список команд для ключевого слова *key_word* с помощью

```
sky% man -k key_word
```

Полезной командой для получения интерактивной информации может быть команда «*info*», однако наполнение и команды управления могут различаться в разных системах.

Chapter 2

Интерполяция и аппроксимация

Распространённой задачей в работе со сложными функциями и экспериментальными данными является задача интерполяции или аппроксимации. Интерполяция нужна, когда мы имеем набор значений функции, достаточно точных, и хотим построить гладкую функцию, проходящую через эти значения. Такая потребность возникает, когда точное вычисление функции трудоёмко и практически может быть рассчитан только набор значений функции на какой-то сетке значений аргумента. Тогда для удобного использования этой функции в каких-то приложениях может оказаться пригодной процедура интерполяции.

Термин «аппроксимация» — приближение,— применим, когда мы не требуем прохождения аппроксимирующей функции через известные значения функции. Такое может быть, например, когда значения функции в заданных точках известны с какой-то точностью (скажем, из экспериментальных измерений). Очевидно, если табличные значения функции содержат статистическую ошибку, то требование прохождения аппроксимирующей функции точно через табличные значения является не только чрезмерным, но даже вредным, искажающим истинное поведение функции.

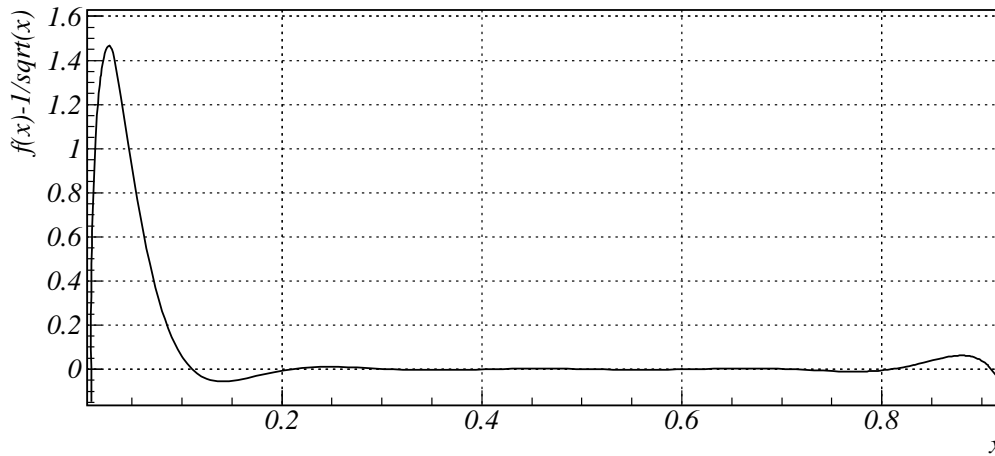


Figure 2.1: Отклонение интерполирующего полинома (2.1) $f(x)$ от $1/\sqrt{x}$. Опорные значения функции вычислены в точках $x_k = 0.01, 0.11, 0.21, 0.31, 0.41, 0.51, 0.61, 0.71, 0.81, 0.91$

2.1 Интерполяция полиномами

Идея интерполяции известных значений функции полиномами возникла очень давно. Очевидно, что для интерполяции n значений функции достаточно использовать полином степени $(n - 1)$. Если обозначить набор точек и значений функции в этих точках через x_i, y_i , $i = 1, \dots, n$, то для построения такого полинома можно использовать удобную формулу Лагранжа [12]:

$$f(x) = \sum_{k=1}^n y_k \cdot \left(\prod_{i \neq k} \frac{x - x_i}{x_k - x_i} \right). \quad (2.1)$$

Хотя такая интерполяция, очевидно, формально является гладкой (на всей оси x существуют непрерывные производные всех порядков, правда, начиная с порядка $n + 1$, производные тождественно равны нулю), практически она является неудобной для большого числа точек n . Интерполирующая функция может иметь сильно выраженный волнообразный характер, особенно усиливающийся, если табличные значения функции вычислены с ошибкой. Для иллюстрации возьмём функцию $1/\sqrt{x}$ в десяти точках $x_k = (k - 1)/10 + 0.01$, $k = 1, 2, \dots, 10$. На рис. 2.1 показано отклонение полинома 9-ой степени $f(x)$ от функции $1/\sqrt{x}$.

Показанный пример демонстрирует, что не обязательно математическое

понятие гладкости соответствует интуитивно ожидаемому поведению функции. Более углублённое описание интерполяции полиномами, в том числе оценку ошибки интерполяции, можно найти в [12].

2.2 Интерполяция сплайнами

Более распространённой в настоящее время является интерполяция табличных значений кубическими сплайнами [13]. Опорные точки x_k , в которых известно значение функции y_k , называются узлами сплайна (будем для удобства предполагать их упорядоченными $x_k < x_{k+1}$).

Общая идея такой интерполяции заключается в следующем: в каждом интервале между узлами функция представляется в виде кубического полинома, а в узлах полиномы «сшиваются» условиями непрерывности функции, а также первой и второй производной. Непрерывность функции будет обеспечена, кстати, автоматически приравниванием значений полиномов, определённых справа и слева от узла x_k , значению функции y_k . Обозначим значение второй производной сплайна в узле x_k через c_k . Тогда в интервале $x_k < x < x_{k+1}$ представление кубического полинома может быть записано в виде

$$f(x) = y_k \cdot \frac{x_{k+1}-x}{x_{k+1}-x_k} + y_{k+1} \cdot \frac{x-x_k}{x_{k+1}-x_k} + \\ + c_k \cdot \frac{(x-x_k)(x_{k+1}-x)(x+x_k-2x_{k+1})}{6(x_{k+1}-x_k)} - c_{k+1} \cdot \frac{(x-x_k)(x_{k+1}-x)(x+x_{k+1}-2x_k)}{6(x_{k+1}-x_k)} \quad (2.2)$$

Коэффициенты c_k не являются независимыми — условие непрерывности первой производной в каждом узле x_k приводит к уравнению

$$\frac{(2c_k + c_{k-1})(x_k - x_{k-1})}{6} + \frac{y_k - y_{k-1}}{x_k - x_{k-1}} = \frac{(2c_k + c_{k+1})(x_k - x_{k+1})}{6} + \frac{y_{k+1} - y_k}{x_{k+1} - x_k}. \quad (2.3)$$

Учитывая, что имеется два крайних узла, в которых нет требования непрерывности производных, получаем, что решений указанной задачи имеется бесконечно много. Чтобы устранить эту неопределённость, можно потребовать какие-то граничные условия. Разные варианты таких условий рассматриваются в [13]. С другой стороны, можно накладывать условия не на поведение сплайна в крайних узлах, а на какую-то среднюю характеристику сплайна. Например, потребуем, чтобы среднеквадратичная кривизна полученной кривой была минимальна. Выберем в качестве меры кривизны вторую производную и найдём,

при каких условиях достигается минимум интеграла от квадрата второй производной на интервале (x_1, x_n) . Вторая производная при $x_k < x < x_{k+1}$ равна

$$f''(x) = \frac{(x_{k+1} - x)c_k + (x - x_k)c_{k+1}}{x_{k+1} - x_k}. \quad (2.4)$$

Следовательно,

$$\int_{x_k}^{x_{k+1}} [f''(x)]^2 dx = \frac{x_{k+1} - x_k}{3} (c_k^2 + c_k c_{k+1} + c_{k+1}^2). \quad (2.5)$$

Оказывается, что минимум среднеквадратичной кривизны достигается при значениях коэффициентов $c_1 = c_n = 0$. Этот не слишком очевидный вывод наиболее просто может быть получен, следуя работе [14]. Заодно мы познакомимся с альтернативной параметризацией кубических сплайнов, в некоторых случаях более удобной. Вместо коэффициентов c_k , являющихся значениями второй производной сплайна в соответствующем узле, используем коэффициенты d_k — первые производные сплайна. В интервале $x_k < x < x_{k+1}$

$$\begin{aligned} f(x) = & y_k \frac{(x - x_{k+1})^2 (2x + x_{k+1} - 3x_k)}{(x_{k+1} - x_k)^3} + d_k \frac{(x - x_{k+1})^2 (x - x_k)}{(x_{k+1} - x_k)^2} + \\ & + y_{k+1} \frac{(x - x_k)^2 (3x_{k+1} - 2x - x_k)}{(x_{k+1} - x_k)^3} + d_{k+1} \frac{(x - x_k)^2 (x - x_{k+1})}{(x_{k+1} - x_k)^2} \end{aligned} \quad (2.6)$$

Если коэффициенты d_k являются свободными параметрами, то такой тип сплайна называется кубическим сплайном дефекта 2 (разрывными являются две старшие производные). Для сплайнов дефекта 1 условие непрерывности второй производной в узле x_k записывается следующим образом:

$$\begin{aligned} \frac{2d_{k-1}}{x_{k-1} - x_k} + \frac{4(x_{k+1} - x_{k-1})d_k}{(x_k - x_{k-1})(x_k - x_{k+1})} + \frac{2d_{k+1}}{x_k - x_{k+1}} - \\ - \frac{6y_{k-1}}{(x_k - x_{k-1})^2} + \frac{6(x_{k-1} + x_{k+1} - 2x_k)(x_{k+1} - x_{k-1})y_k}{(x_k - x_{k+1})^2 (x_k - x_{k-1})^2} + \frac{6y_{k+1}}{(x_k - x_{k+1})^2} = 0 \end{aligned} \quad (2.7)$$

Вторая производная сплайна на интервале (x_k, x_{k+1}) равна

$$\begin{aligned} f''(x) = & \frac{6(x_k + x_{k+1} - 2x)(y_{k+1} - y_k)}{(x_{k+1} - x_k)^3} + \\ & + \frac{3(2x - x_k - x_{k+1})(d_k + d_{k+1})}{(x_{k+1} - x_k)^2} + \frac{d_{k+1} - d_k}{x_{k+1} - x_k} \end{aligned} \quad (2.8)$$

Отсюда получаем выражение для интеграла от квадрата второй производной:

$$I_C = \int_{x_1}^{x_n} [f''(x)]^2 dx = \sum_{k=1}^{n-1} \left\{ \frac{(d_{k+1}-d_k)^2}{x_{k+1}-x_k} + \frac{3[(d_{k+1}+d_k)(x_{k+1}-x_k)+2(y_k-y_{k+1})]^2}{(x_{k+1}-x_k)^3} \right\} \quad (2.9)$$

Найдём минимум этой величины для сплайнов дефекта 2 (в этом случае все d_k являются независимыми). Частная производная по d_k , $k = 2, 3, \dots, n-1$ равна

$$\begin{aligned} \frac{\partial I_C}{\partial d_k} = & 2 \left[\frac{d_k-d_{k+1}}{x_{k+1}-x_k} + \frac{d_k-d_{k-1}}{x_k-x_{k-1}} \right] + \\ & + 6 \left[\frac{d_{k+1}+d_k}{x_{k+1}-x_k} + \frac{2(y_k-y_{k+1})}{(x_{k+1}-x_k)^2} + \frac{d_{k-1}+d_k}{x_k-x_{k-1}} + \frac{2(y_{k-1}-y_k)}{(x_k-x_{k-1})^2} \right] = 0 \end{aligned} \quad (2.10)$$

Легко убедиться, что это уравнение тождественно условию непрерывности (2.7) второй производной. Частные производные по d_1 и d_n равны

$$\frac{\partial I_C}{\partial d_1} = 2 \frac{d_1-d_2}{x_2-x_1} + 6 \left[\frac{d_2+d_1}{x_2-x_1} + \frac{2(y_1-y_2)}{(x_2-x_1)^2} \right] = 0, \quad (2.11)$$

$$\frac{\partial I_C}{\partial d_n} = 2 \frac{d_n-d_{n-1}}{x_n-x_{n-1}} + 6 \left[\frac{d_n+d_{n-1}}{x_n-x_{n-1}} + \frac{2(y_{n-1}-y_n)}{(x_n-x_{n-1})^2} \right] = 0.$$

Здесь также легко проверить, что первое уравнение соответствует равенству нулю второй производной в крайнем левом узле x_1 , а второе — равенству нулю второй производной в крайнем правом узле x_n . Итак, среди всех сплайнов дефекта 2, имеющих значения y_k в узлах x_k , наименьшую среднеквадратичную кривизну имеет сплайн дефекта 1, у которого крайние коэффициенты c_1 и c_n равны нулю. Посмотрим, как можно получить коэффициенты c_k такого сплайна. Перепишем условие непрерывности (2.3) первой производной в другом виде:

$$\begin{aligned} c_{k+1} = & \frac{6(y_{k-1}-y_k)}{(x_{k+1}-x_k)(x_k-x_{k-1})} + \frac{6(y_{k+1}-y_k)}{(x_{k+1}-x_k)^2} - \\ & - \frac{x_k-x_{k-1}}{x_{k+1}-x_k} c_{k-1} - 2 \frac{x_{k+1}-x_{k-1}}{x_{k+1}-x_k} c_k. \end{aligned} \quad (2.12)$$

Используя эту рекуррентную формулу, можно выразить значения всех коэффициентов c_k через c_1 и c_2 . Коэффициент $c_1 = 0$, а коэффициент c_2

можно найти, приравнивания $c_n = 0$. После этого все коэффициенты c_k определены, и значение сплайна в любой точке $x \in (x_1, x_n)$ может быть рассчитано по формуле (2.2).

2.3 Аппроксимация экспериментальных данных

В задачах сглаживания или аппроксимации несколько другая постановка задачи: на некоторой сетке значений x_k даны значения функции y_k со статистической ошибкой σ_k . В этих случаях аппроксимирующая функция не должна принимать значения y_k в точках x_k . Если есть какие-то теоретические основания выбрать конкретный вид функции со свободными параметрами, то этот вариант, обычно, предпочтительнее. Бывает, что нет никаких оснований выбрать какой-то конкретный вид функции. Тогда часто выбирают вид функции, исходя из несколько неопределённого принципа — «на что она похожа». Это может быть Гауссова кривая, экспонента, полиномы, сумма Гауссовых кривых с разными центрами и ширинами и т.п. В качестве меры допустимого отклонения аппроксимирующей функции от y_k чаще всего используют χ^2 (критерий статистического согласия χ^2 обсуждается в разделе 9.1):

$$\chi^2 = \sum_{k=1}^n \left(\frac{f(x_k) - y_k}{\sigma_k} \right)^2. \quad (2.13)$$

В работе [15] сделана попытка единого подхода к аппроксимации экспериментального распределения при отсутствии теоретической модели. Основной идеей этой работы является представление аппроксимирующей функции кубическим сплайном дефекта 1. Координаты крайних узлов сплайна определяются крайними точками в массиве x_k . Промежуточные узлы сплайна выбираются на одинаковом расстоянии, а количество узлов подбирается таким, чтобы минимальное значение χ^2 было удовлетворительным со статистической точки зрения. Рассмотренные ранее представления сплайна имеют тот недостаток, что коэффициенты d_k или c_k сильно коррелированы (изменение одного коэффициента меняет значения сплайна на всём интервала $x_1 < x < x_n$). Существенным техническим элементом предложенного алгоритма в [15] является представление сплайна через В-сплайны, которые представляют самостоятельный интерес.

2.3.1 В-сплайны

Понятие В-сплайнов введено в работе [16] и используется во многих учебниках, например, в [13]. В-сплайн является кубическим сплайном дефекта 1 специального вида и определяется на сетке узлов x_k следующими требованиями:

1. Сплайн $B_k(x)$ принимает ненулевые значения только на интервале $x_{k-2} < x < x_{k+2}$.
2. В узлах x_{k-2} и x_{k+2} равны нулю значения сплайна и его первых двух производных.

Указанных основных свойств В-сплайна недостаточно для полного определения функции — остаётся ещё одна степень свободы. Она фиксируется каким-либо дополнительным условием. Обычно это или значение сплайна в узле x_k или величина площади под кривой. Наиболее просто определяется В-сплайн на равномерной сетке узлов x_k , где $x_{k+1} - x_k = h$:

$$B_k(x) = \begin{cases} 0, & x < x_k - 2h, \\ \frac{1}{4h^3} (x - x_k + 2h)^3, & x_k - 2h < x < x_k - h, \\ 1 - \frac{3}{4h^3} (x - x_k)^2 (x - x_k + 2h), & x_k - h < x < x_k, \\ 1 - \frac{3}{4h^3} (x - x_k)^2 (x_k + 2h - x), & x_k < x < x_k + h, \\ \frac{1}{4h^3} (x_k + 2h - x)^3, & x_k + h < x < x_k + 2h, \\ 0, & x > x_k + 2h. \end{cases} \quad (2.14)$$

Здесь выбрана нормировка на значение сплайна в максимуме. Площадь под кривой равна $\frac{3}{2}h$, средне-квадратичное отклонение от среднего равно $\frac{h}{\sqrt{3}}$. На рис. 2.2 представлен график такого сплайна. В-сплайн по форме очень похож на Гауссову функцию распределения вероятностей. Любой кубический сплайн на интервале $x_1 < x < x_n$ можно однозначно представить в виде линейной комбинации В-сплайнов:

$$f(x) = \sum_{k=0}^{n+1} \mu_k \cdot B_k(x) \quad (2.15)$$

Видно, что для использования этой формулы надо имеющуюся сетку узлов x_1, x_2, \dots, x_n дополнить узлами x_{-2}, x_{-1}, x_0 и $x_{n+1}, x_{n+2}, x_{n+3}$. Для случая равномерной сетки это не представляет трудностей, для неравномерной

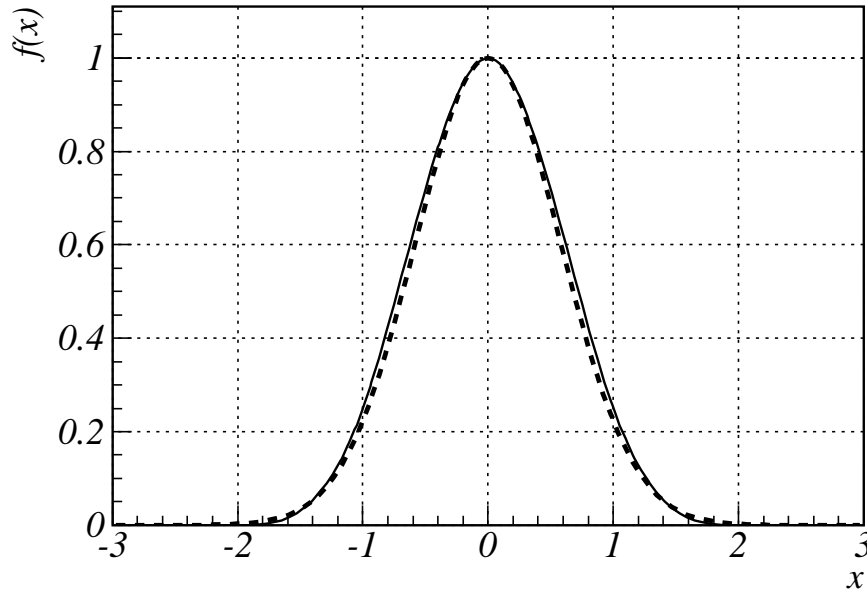


Figure 2.2: График В-сплайна $B_k(x)$ для случая $x_k = 0$, $h = 1$. Пунктиром показана Гауссова кривая с тем же максимальным значением и среднеквадратичным отклонением от среднего.

сетки можно выбрать любой вариант доопределения. Использование В-сплайнов наиболее эффективно в тех задачах, где необходимо проводить численную минимизацию какой-либо функции по параметрам μ_k , так как в этом случае параметры μ_k слабо коррелированы (значения сплайна на интервале $x_k < x < x_{k+1}$ зависят только от значений коэффициентов $\mu_{k-1}, \mu_k, \mu_{k+1}, \mu_{k+2}$).

2.3.2 Использование средне-квадратичной кривизны при аппроксимации экспериментальных данных

При аппроксимации экспериментальных данных в отсутствии теоретической модели можно попытаться использовать характеристику средней кривизны функции (2.9). В работе [14] предложено для аппроксимации экспериментальных значений минимизировать функцию L , объединяющую χ^2 и среднюю кривизну:

$$L = \chi^2 + q \cdot I_C = \sum_{k=1}^n \left(\frac{f(x_k) - y_k}{\sigma_k} \right)^2 + q \cdot \int_{-\infty}^{+\infty} [f''(x)]^2 dx. \quad (2.16)$$

Здесь q — некоторая положительная константа. В работе методами вариационного исчисления показывается, что в классе гладких функций $f(x)$ минимум L достигается функцией, у которой $f'''(x) = 0$ при $x \neq x_k$. Как легко заметить, это самое общее определение кубических сплайнов с набором узлов x_k . Для разных значений множителя q будут получаться разные решения. Устремляя $q \rightarrow 0$, мы получим стандартный кубический сплайн, проходящий точно через значения y_k . При этом $\chi^2 \rightarrow 0$. В противоположном случае $q \rightarrow \infty$ получается проведение прямой через экспериментальные точки методом наименьших квадратов. В данном случае мы имеем n степеней свободы, поэтому разумным значением χ^2 , со статистической точки зрения, является n . В работе [14] предлагается увеличивать q до тех пор, пока не будет достигнуто $\chi^2 = n - 2$ (здесь учитывается, что две степени свободы «съедаются» минимизацией функции по общему сдвигу и повороту).

Chapter 3

Метод Монте – Карло

Методом Монте–Карло называют любой метод расчёта с использованием генераторов случайных величин. Эти методы могут касаться разных областей науки и самых неожиданных задач, не имеющих, на первый взгляд, к случайности никакого отношения, как-то: решение интегральных и дифференциальных уравнений, вычисление определённых интегралов, задачи оптимизации и т.д. Большая часть этих методов описана в учебниках и справочниках [8, 9, 10, 11, 12]. Здесь мы коснёмся только двух задач: вычисление определённых интегралов и имитационное моделирование физических процессов.

3.1 Случайные величины. Общие характеристики распределения вероятностей

Случайные величины однозначно и полностью характеризуются своей функцией распределения вероятностей $f(r) \geq 0$, где r – случайная величина, определённая в интервале $[a, b]$, причём $\int_a^b f(r) dr = 1$. Интервал может быть конечным или бесконечным.

Функция

$$\Phi(r) = \int_a^r f(r) dr \quad (3.1)$$

тоже называется функцией распределения. Чтобы различать эти две функции, обычно $f(r)$ называют дифференциальной функцией распределения (по-английски *probability density function* — *p.d.f.*), а $\Phi(r)$ — интегральной функцией распределения (по-английски — *cumulative density function*).

Функция $\Psi(r)$ от случайной величины, естественно, сама является случайной величиной и имеет функцию распределения вероятностей. Правило получения функции распределения вероятностей по величине Ψ можно сформулировать следующим образом:

$$\Phi_{\Psi}(\mu) = \int_{\substack{\Psi(r) < \mu \\ a < r < b}} f(r) dr; \quad f_{\Psi}(\mu) = \frac{d\Phi_{\Psi}(\mu)}{d\mu}. \quad (3.2)$$

В качестве иллюстрации этого метода получим функцию распределения вероятностей случайной величины y , являющейся отношением двух независимых случайных величин, распределённых по закону Гаусса с нулевым средним и единичной дисперсией.

$$y = \frac{g_1}{g_2}; \quad \frac{dW}{dg_1} = \frac{1}{\sqrt{2\pi}} e^{-\frac{g_1^2}{2}}; \quad \frac{dW}{dg_2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{g_2^2}{2}}. \quad (3.3)$$

Совместная плотность вероятности для независимых величин равна

$$\frac{d^2W}{dg_1 dg_2} = \frac{1}{2\pi} \exp\left(-\frac{g_1^2 + g_2^2}{2}\right), \quad (3.4)$$

а интегральная функция распределения вероятностей по y равна:

$$\Phi(y) = \int_{g_1/g_2 < y} \frac{dg_1 dg_2}{2\pi} \exp\left(-\frac{g_1^2 + g_2^2}{2}\right). \quad (3.5)$$

Сделаем замену переменных под интегралом:

$$g_1 = \rho \cdot \sin \psi, \quad g_2 = \rho \cdot \cos \psi, \quad dg_1 dg_2 \rightarrow \rho d\rho d\psi. \quad (3.6)$$

Тогда

$$\Phi(y) = \int_{\text{tg}\psi < y} \frac{\rho d\rho d\psi}{2\pi} \exp\left(-\frac{\rho^2}{2}\right) = \frac{1}{2} + \frac{1}{\pi} \text{arctg } y. \quad (3.7)$$

Отсюда легко получается функция плотности вероятности

$$f(y) = \frac{dW}{dy} = \frac{d}{dy} \Phi(y) = \frac{1}{\pi \cdot (1 + y^2)}. \quad (3.8)$$

Словами этот результат можно сформулировать так: отношение двух независимых гауссовых величин с нулевым средним имеет распределение вероятностей Коши.

Среднее значение функции $\Psi(r)$ случайной величины определяется как $\int_a^b \Psi(r) f(r) dr$ и часто обозначается как $\langle \Psi(r) \rangle$ или $\overline{\Psi(r)}$.

Важной характеристикой распределения вероятностей являются моменты распределения M_n и центральные моменты C_n . Моментом n -го порядка распределения называется величина

$$M_n = \int_a^b r^n f(r) dr,$$

центральным моментом распределения называется величина

$$C_n = \int_a^b (r - M_1)^n f(r) dr.$$

Можно сказать, что распределение вероятностей полностью характеризуется набором всех моментов (если они существуют).

Некоторые моменты функции распределения ввиду их важности получили собственные имена. Так, первый момент функции распределения, являющийся средним значением самой случайной величины, называется также её математическим ожиданием. Второй центральный момент C_2 называется дисперсией и часто обозначается как $D(r)$. Квадратный корень из дисперсии называется среднеквадратичным отклонением от среднего (в английском варианте — r.m.s., сокращение от root-mean-square) и часто обозначается как $\sigma(r)$.

Эффективным средством получения любого момента распределения является производящая функция моментов $F(x) = \int_a^b e^{xr} f(r) dr$. Очевидно, что для получения момента n -го порядка достаточно n раз продифференцировать функцию $F(x)$ по x и положить x равным нулю.

Аналогично может использоваться производящая функция центральных моментов $G(x) = \int_a^b \exp[(r - M_1)x] f(r) dr = F(x) \cdot \exp(-xM_1)$.

В некоторых случаях могут отсутствовать какие-то (или все) моменты. Например, распределение Коши

$$f(r) = \frac{1}{\pi \cdot [(r - a)^2 + \sigma^2]} \quad (3.9)$$

не имеет ни одного момента. В этом случае не существует и производящей функции моментов.

Важное значение имеет понятие характеристической функции (см., например, [17, 19]):

$$\varphi(t) = \int_a^b e^{itr} f(r) dr. \quad (3.10)$$

Характеристическая функция существует для любого распределения, другое дело что часто её бывает трудно вычислить. С другой стороны, бывают случаи, когда для распределения вероятностей неизвестна явная запись функции распределения в элементарных функциях, а характеристическая функция известна и имеет довольно простой вид.

В тех случаях, когда моменты распределения существуют, их можно получить из характеристической функции по формуле

$$M_n = \frac{1}{i^n} \cdot \left. \frac{d^n \varphi(t)}{dt^n} \right|_{t=0} \quad (3.11)$$

Функция распределения вероятностей $f(r)$ может быть восстановлена из характеристической функции по так называемой *формуле обращения* [17, 19]

$$f(r) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \varphi(t) e^{-itr} dt. \quad (3.12)$$

В качестве примеров использования характеристической функции рассмотрим два распределения вероятностей.

Пример 1. Распределение вероятностей по величине $y = x_1 + x_2$, где x_1, x_2 – случайные числа, равномерно распределённые в $(0, 1)$, может быть получено традиционным способом, рассмотренным на стр. 34, и равно

$$f(y) = \begin{cases} y, & 0 < y < 1, \\ 2 - y, & 1 < y < 2. \end{cases} \quad (3.13)$$

А как это получить с помощью характеристической функции? Характеристическая функция в этом случае равна

$$\varphi(t) = \langle e^{iyt} \rangle = \int_0^1 dx_1 \int_0^1 dx_2 \cdot e^{it(x_1+x_2)} = \left(\frac{e^{it} - 1}{it} \right)^2. \quad (3.14)$$

3.1. СЛУЧАЙНЫЕ ВЕЛИЧИНЫ. ОБЩИЕ ХАРАКТЕРИСТИКИ РАСПРЕДЕЛЕНИЯ

Отсюда сразу можно получить все моменты распределения

$$M_k = \frac{1}{i^k} \cdot \left. \frac{d^k \varphi(t)}{dt^k} \right|_{t=0} = \sum_{m=0}^k \frac{k!}{(m+1)!(k-m+1)!} = 2 \cdot \frac{2^{k+1} - 1}{(k+1)(k+2)}. \quad (3.15)$$

Функцию распределения вероятностей можно найти по формуле обращения:

$$f(y) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \varphi(t) e^{-ity} dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left(\frac{e^{it} - 1}{it} \right)^2 e^{-ity} dt. \quad (3.16)$$

Наиболее просто интеграл можно найти по теории вычетов [12]. Функция $\varphi(t)$ аналитическая во всей комплексной плоскости, в том числе в точке $t = 0$. Если сместить путь интегрирования на величину δ вниз от действительной оси, то величина интеграла не изменится. Если теперь раскрыть скобки и рассмотреть каждое слагаемое, то легко заметить, что у всех у них появляется полюс второго порядка в точке $t = 0$.

Рассмотрим интеграл

$$R(\beta) = \frac{1}{2\pi} \int_{-i\delta - \infty}^{-i\delta + \infty} \frac{e^{i\beta t}}{t^2} dt. \quad (3.17)$$

Для того, чтобы воспользоваться теорией вычетов, надо замкнуть путь интегрирования на бесконечности полуокружностью. Где можно замкнуть, чтобы интеграл по этой полуокружности был равен нулю – в верхней полуплоскости или в нижней, – зависит от знака параметра β . Если $\beta > 0$, то замкнуть можно в верхней полуплоскости, если $\beta < 0$, то в нижней полуплоскости. В зависимости от этого полюс попадает или не попадает внутрь контура интегрирования. Вычет подынтегральной функции в точке $t = 0$ равен $i\beta/(2\pi)$. Поэтому по теории вычетов интеграл

$$R(\beta) = \begin{cases} 0, & \beta < 0, \\ 2\pi i \cdot \frac{i\beta}{2\pi} = -\beta, & \beta > 0. \end{cases} \quad (3.18)$$

В итоге получаем

$$f(y) = 2R(1-y) - R(2-y) - R(-y) = \begin{cases} 0, & y < 0, \\ y, & 0 < y < 1, \\ 2-y, & 1 < y < 2, \\ 0, & y > 2. \end{cases} \quad (3.19)$$

Легко убедиться, что результат не изменился бы, если мы путь интегрирования сместили бы вверх от действительной оси.

Эту процедуру легко обобщить на сумму произвольного числа n случайных переменных, равномерно распределённых в интервале $(0, 1)$.

$$y_n = \sum_{m=1}^n x_m, \quad \varphi_n(t) = \left(\frac{e^{it} - 1}{it} \right)^n. \quad (3.20)$$

К сожалению, выписать общую формулу для произвольного момента распределения M_k представляется затруднительным. Однако, можно довольно просто вычислить любой конкретный момент:

$$\begin{aligned} M_1 &= \frac{n}{2}; & M_2 &= \frac{(3n+1)n}{12}; & M_3 &= \frac{(n+1)n^2}{8}; \\ M_4 &= \frac{(15n^3+30n^2+5n-2)n}{240}; & M_5 &= \frac{(3n^2+7n-2)(n+1)n^2}{96}; \\ M_6 &= \frac{(63n^5+315n^4+315n^3-91n^2-42n+16)n}{4032}. \end{aligned} \quad (3.21)$$

Также можно вычислить и центральные моменты:

$$\begin{aligned} C_1 &= C_3 = C_5 = \dots = 0; & C_2 &= \frac{n}{12}; & C_4 &= \frac{(5n-2)n}{240}; \\ C_6 &= \frac{(35n^2-42n+16)n}{4032}. \end{aligned} \quad (3.22)$$

Функция распределения $f(y_n)$ получается по той же схеме, что и для суммы двух случайных величин, только полюсы будут n -го порядка:

$$f(y_n) = \sum_{m=0}^n \frac{n!(-1)^m}{2\pi \cdot m!(n-m)!} \int_{-i\delta-\infty}^{-i\delta+\infty} \frac{e^{i(n-m-y_n)t} dt}{i^n t^n}. \quad (3.23)$$

Вспомогательный интеграл по теории вычетов

$$R_n(\beta) = \frac{(n-1)!}{2\pi} \int_{-i\delta-\infty}^{-i\delta+\infty} \frac{e^{i\beta t} dt}{i^n t^n} = \begin{cases} 0, & \beta < 0, \\ \beta^{n-1}, & \beta > 0. \end{cases} \quad (3.24)$$

Если ввести вспомогательное обозначение

$$(x)_+ = \begin{cases} 0, & x < 0, \\ x, & x > 0, \end{cases} \quad (3.25)$$

то функцию распределения можно записать в следующем виде:

$$f(y_n) = \sum_{m=0}^n \frac{(-1)^m n \cdot (n - m - y_n)_+^{n-1}}{m!(n - m)!}. \quad (3.26)$$

Фактически, эта функция представляет из себя полиномиальный сплайн порядка $(n - 1)$ [13].

Пример 2. Распределение вероятностей по величине ионизационных потерь заряженной частицы. Рассмотрим приближённую модель ионизационных потерь энергии заряженной частицы при прохождении слоя вещества. Если потери энергии много меньше начальной энергии частицы, что часто бывает для тяжёлых частиц, то процессы потерь от столкновения с разными атомами достаточно независимы, и можно считать, что при прохождении частицы в среднем порождаются n_a δ -электронов (выбиваются с атомных оболочек). Конкретное число δ -электронов получается выборкой из распределения Пуассона. Распределение по энергии каждого δ -электрона хорошо описывается функцией

$$f(\epsilon) \sim \frac{1}{(\epsilon + I)^2}, \quad (3.27)$$

где I – некоторый эффективный ионизационный потенциал. Энергия каждого δ -электрона не может достигать сколь угодно больших значений — законы сохранения энергии-импульса определяют некоторый кинематический предел $\epsilon \leq T_c$. Теперь можно полностью определить распределение вероятностей по энергии δ -электрона:

$$f(\epsilon) = \frac{(T_c + I)I}{(\epsilon + I)^2 T_c}, \quad 0 \leq \epsilon \leq T_c. \quad (3.28)$$

В действительности, при прохождении заряженной частицы через вещество большая часть энергии тратится на возбуждение атомов, не сопровождающееся рождением δ -электронов. Кроме того, для электронов с разных атомных оболочек имеется свой потенциал ионизации, что в основном модифицирует распределение вероятностей при малых потерях энергии. Поэтому более точно то распределение вероятностей, которое мы здесь будем вычислять, можно назвать распределением вероятностей по суммарной энергии δ -электронов, порождённых движущейся заряженной частицей.

Характеристическую функцию этого распределения можно записать так:

$$\begin{aligned} \varphi(t) &= \sum_{k=0}^{\infty} \frac{n_a^k e^{-n_a}}{k!} \cdot \left[\int_0^{T_c} \frac{e^{i\epsilon t} (T_c + I) I d\epsilon}{(\epsilon + I)^2 T_c} \right]^k = \\ &= \exp \left[-n_a + n_a \int_0^{T_c} \frac{e^{i\epsilon t} (T_c + I) I d\epsilon}{(\epsilon + I)^2 T_c} \right] = \exp \left[n_a \int_0^{T_c} \frac{(e^{i\epsilon t} - 1) (T_c + I) I d\epsilon}{(\epsilon + I)^2 T_c} \right]. \end{aligned} \quad (3.29)$$

Отсюда легко найти моменты распределения:

$$\begin{aligned} M_1 = \bar{E} &= n_a \cdot I \cdot \left[\left(1 + \frac{I}{T_c} \right) \ln \left(1 + \frac{T_c}{I} \right) - 1 \right] \approx n_a \cdot I \cdot \left[\ln \frac{T_c}{I} - 1 \right], \\ \sigma_E^2 = C_2 &= n_a \cdot \left[(T_c + 2I)I - 2I^2 \cdot \left(1 + \frac{I}{T_c} \right) \cdot \ln \left(1 + \frac{T_c}{I} \right) \right] \approx \\ &\approx n_a \cdot I \cdot \left[T_c - 2I \cdot \ln \frac{T_c}{I} \right]. \end{aligned} \quad (3.30)$$

Здесь приближённые выражения написаны для случая $T_c \gg I$.

По формуле обращения можно найти функцию распределения

$$f(E) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} dt \exp \left[-itE + n_a \int_0^{T_c} \frac{(e^{i\epsilon t} - 1) (T_c + I) I d\epsilon}{(\epsilon + I)^2 T_c} \right]. \quad (3.31)$$

Во всех работах по распределению вероятностей по ионизационным потерям заряженных частиц возникают подобные интегралы в комплексной плоскости, начиная с основополагающей работы Л.Д.Ландау [18]. Несмотря на то, что полученная функция распределения выглядит довольно сложной, всё же трудно себе представить, как можно получить точное выражение для этой функции распределения другим способом, не через характеристическую функцию.

В следующих разделах мы обсудим некоторые часто встречающиеся распределения вероятностей и способы генерации псевдослучайных чисел, распределённых по соответствующему закону.

3.2 Генераторы некоторых стандартных распределений

3.2.1 Равномерное распределение

Любой метод Монте – Карло использует некоторый генератор

случайных чисел. В принципе можно рассматривать генераторы истинно случайных чисел и псевдослучайных чисел. Истинно случайные числа следует получать с помощью устройств, работающих на основе физических случайных процессов. В этом случае необходимо дополнительно предусмотреть процедуру считывания готового случайного числа в ЭВМ. В некоторых случаях, когда нет никаких требований к качеству случайных чисел, используют младшие разряды внутреннего таймера ЭВМ. Однако для серьёзных расчётов этот способ не годится. В случае хорошо спроектированного генератора истинно случайных чисел (например, [20]) всё же остаётся проблема изготовления этого устройства, контроля за постоянством характеристик и т.д. Из-за этих трудностей с внешними генераторами истинно случайных чисел в практике использования генераторов случайных чисел давно сделан выбор в пользу псевдослучайных чисел.

Как правило, это программный генератор псевдослучайных чисел с равномерным распределением вероятностей в интервале $[0, 1]$, т. е. $f(r) = 1$. Как будет видно дальше, такой генератор может быть использован для генерации произвольных распределений вероятностей. В настоящее время имеется много программ-генераторов случайных величин с равномерным распределением вероятностей с разными алгоритмами и достаточно хорошими статистическими свойствами. Ни один из них нельзя считать идеальным. Запутанность алгоритма не гарантирует хороших статистических свойств, об этом хорошо написано в [21]. Все известные генераторы случайных чисел, производя некоторые арифметические операции над одним или несколькими предыдущими случайными числами, получают одно или несколько новых случайных чисел. В этом смысле в процессе получения ряда псевдослучайных чисел на ЭВМ нет ничего случайного. Более того, из-за конечной разрядной сетки ЭВМ любой ряд псевдослучайных чисел имеет конечный период T . Кажущийся недостаток генераторов псевдослучайных чисел (это то, что при повторном использовании генератора ряд случайных чисел в точности повторяется) на практике оборачивается очень большим достоинством, которое в полной мере применяется при отладке больших программ, использующих метод Монте – Карло.

Период T – основная характеристика ряда: после T случайных чисел ряд начинает в точности повторяться. Чем больше период ряда, тем лучше генератор при прочих равных условиях. Обычные значения периода ряда – 10^9 и больше, при меньших значениях генератор, вероятно, нельзя использовать в серьёзных расчётах. Большое

значение периода ряда, однако, не гарантирует хороших статистических свойств. Минимальный «джентльменский» набор статистических тестов для генераторов равномерного распределения обсуждается в той же книге [21]. Детальное описание алгоритмов нескольких генераторов можно найти в работах [22, 23, 24, 25]. Приведём алгоритм наиболее простого генератора – мультипликативного.

Существенным элементом такого генератора является производящая константа ряда k , от выбора которой зависят статистические свойства ряда. Начиная с некоторого стартового целого числа генерируется ряд целых положительных псевдослучайных чисел, не превосходящих 2^n . Новое случайное число ξ_{i+1} получается из предыдущего ξ_i по формуле $\xi_{i+1} = \xi_i \times k \bmod 2^n$, иначе говоря, новое случайное число равно младшим n двоичным разрядам произведения предыдущего случайного числа и производящей константы ряда.

В этом алгоритме исходное случайное число должно быть нечётным (и все остальные числа должны получаться нечётными), производящая константа ряда должна удовлетворять равенству $k = 3 \bmod 8$ или $k = 5 \bmod 8$, при этом достигается максимально возможный для этого алгоритма период $T = 2^{n-2}$. Хорошо зарекомендовавшие себя производящие константы для $n = 32$:

$$k_1 = 452807053 = 1AFD498D_{16}$$

и

$$k_2 = 69069 = 10DCD_{16};$$

для $n = 63$:

$$k_3 = 70369817985301 = 400040010115_{16}.$$

Вещественные случайные числа, равномерно распределённые в интервале $(0, 1)$, получают делением целых чисел ξ_i на 2^n .

Интересным свойством случайных чисел мультипликативного ряда является то, что векторы в r -мерном пространстве, компоненты которых составлены из r последовательных чисел ряда (для краткости будем называть их далее случайными векторами), попадают на параллельные эквидистантные гиперплоскости, расстояние между которыми сравнительно велико (определяется производящей константой ряда). Это свойство сначала было обнаружено экспериментально на примере генератора фирмы ИВМ (после этого все генераторы тщательно проверялись на корреляции в r -мерных пространствах) и было воспринято как большой недостаток для этого типа генераторов. Затем оно было

Table 3.1: Ограничение сверху на минимальное число N_q параллельных эквидистантных гиперплоскостей в n -мерном пространстве, на которые выпадают все случайные вектора, полученные с мультипликативным генератором. Размер мантиссы случайного числа – m бит.

$m \backslash n$	3	4	5	6	7	8	9	10
32	2953	566	220	120	80	60	48	41
63	3811000	122000	16170	4335	1731	884	531	357

исследовано теоретически [26], где было показано, что качественно это явление проявляется при любой производящей константе ряда, меняется лишь расстояние между гиперплоскостями. Если не интересоваться количеством плоскостей, на которые попадают случайные вектора, то это явление тривиально, поскольку координаты всех векторов дискретны и пропорциональны величине 2^{-m} . Поэтому для любого генератора такого типа имеются в n -мерном пространстве, как минимум, $\frac{n(n+1)}{2}$ наборов, параллельных каждой из координатных плоскостей, с шагом между плоскостями 2^{1-m} . Это целиком определяется дискретностью случайных чисел, и относится вообще ко всем генераторам. Однако, для мультипликативных генераторов в работе [26] было доказано сильное утверждение, что для случайных чисел этого типа, независимо от производящей константы ряда, найдётся по крайней мере один набор из N параллельных эквидистантных гиперплоскостей, на которые падают все случайные вектора, причём число плоскостей не превосходит

$$N \leq N_q = (n! \cdot 2^m)^{1/n}. \quad (3.32)$$

В табл. 3.1 приведены значения N_q для некоторых n и m . Знание этого теоретического предела позволило провести поиск параметров конкретных наборов плоскостей для указанных выше производящих констант k_1 , k_2 и k_3 (см. работу [27]). Упрощённо алгоритм поиска заключался в следующем.

Выбирался набор из n случайных векторов, производимых соответствующим генератором, таких, которые не лежат на одной плоскости. Затем перебирались все возможные наборы параллельных плоскостей, проходящих через эти точки и каждый из таких наборов проверялся на истинность с помощью достаточно большого числа дополнительных «контрольных» векторов (среди кандидатов в такой

набор большинство оказывается ложными). Из всех наборов плоскостей, успешно прошедших проверку, выбирался набор с наибольшим шагом H_{max} между плоскостями. В процессе такого перебора оказалось, что n -мерное пространство изрезано во всех направлениях такими наборами плоскостей с разными расстояниями H . В табл.3.2, где приведены результаты такого поиска, множественность наборов характеризуется в некоторой степени количеством разных наборов плоскостей с шагом H в интервале $(0, 5 \cdot H_{max}; H_{max})$.

Теоретическое исследование этого явления способствовало ещё большому недоверию к мультипликативным генераторам. Однако со временем было осознано, что наличие «пустот» при заполнении гиперпространства точками с координатами, образованными последовательными числами псевдослучайного ряда, – неизбежное зло, и форма этих «пустот» в виде пространства между гиперплоскостями ничем не хуже другой формы «пустот». Теперь мультипликативные генераторы снова вошли в употребление, просто производящие константы ряда интенсивно исследуют на предмет такого вида корреляции (см., например, работу [28]). Данные о характерном размере «пустот» в n -мерном пространстве случайных векторов могут представлять и практический интерес. Например, при генерации многомерной случайной величины методом Неймана (см. разд.3.3.2) в тех случаях, когда функция распределения вероятностей имеет узкие пики, генерация может быть ошибочной, если ширина пика много меньше размера «пустот». К сожалению, авторы не встречали исследований других алгоритмов генерации равномерного распределения на эту тему.

Все последующие алгоритмы генерации случайных величин, распределённых по другим законам, предполагают наличие какого-либо (хорошего) генератора случайных чисел, равномерно распределённых в интервале $(0, 1)$.

3.2.2 Распределение Гаусса

Распределение Гаусса (нормальное распределение) имеет большое значение в теории вероятностей. Центральная предельная теорема утверждает, что свёртка большого числа одинаковых распределений вероятности в пределе даёт распределение Гаусса. На практике это утверждение можно применять в следующей формулировке: распределение вероятностей по среднему значению большого числа независимых измерений одной и той же величины близко к распределению Гаусса

Table 3.2: Характеристики наборов параллельных плоскостей, на которые попадают все случайные вектора мультипликативного генератора в n -мерном пространстве.

n	Производящая константа	Максимальный шаг между плоскостями H_{max}	Количество плоскостей в наборе с шагом H_{max}	Количество наборов плоскостей с шагом в интервале $(0,5 \cdot H_{max}; H_{max})$
3	k_1	$1,42 \cdot 10^{-3}$	1021	4
	k_2	$2,78 \cdot 10^{-3}$	395	1
	k_3	$9,22 \cdot 10^{-7}$	$1,47 \cdot 10^6$	10
4	k_1	$1,13 \cdot 10^{-2}$	157	1
	k_2	$1,02 \cdot 10^{-2}$	156	3
	k_3	$5,45 \cdot 10^{-5}$	$3,4 \cdot 10^4$	2
5	k_1	$3,66 \cdot 10^{-2}$	57	1
	k_2	$2,30 \cdot 10^{-2}$	88	11
	k_3	$2,92 \cdot 10^{-4}$	$5,9 \cdot 10^3$	14
6	k_1	$3,86 \cdot 10^{-2}$	53	44
	k_2	$6,43 \cdot 10^{-2}$	35	3
	k_3	$1,76 \cdot 10^{-3}$	$1,2 \cdot 10^3$	1
7	k_1	$5,70 \cdot 10^{-2}$	41	130
	k_2	$7,67 \cdot 10^{-2}$	30	24
	k_3	$2,30 \cdot 10^{-3}$	$1,04 \cdot 10^3$	392
8	k_1	$1,31 \cdot 10^{-1}$	21	4
	k_2	$7,67 \cdot 10^{-2}$	30	378
	k_3	$5,98 \cdot 10^{-3}$	382	142
9	k_1	0,139	19	35
	k_2	0,100	26	723
	k_3	$8,98 \cdot 10^{-3}$	302	946
10	k_1	0,139	17	393
	k_2	0,139	19	424
	k_3	$1,39 \cdot 10^{-2}$	166	2120

со среднеквадратичным отклонением от среднего σ/\sqrt{n} , где σ – среднеквадратичное отклонение одного измерения, n – количество измерений.

Функция распределения Гаусса записывается как

$$f(r) = \frac{1}{\sqrt{2\pi}\sigma} \times \exp\left[-\frac{(r-a)^2}{2\sigma^2}\right],$$

где параметры распределения имеют следующий смысл: $a = M_1$ (математическое ожидание) и $\sigma = \sqrt{D(r)}$ (среднеквадратичное отклонение).

Приведём один из возможных алгоритмов генерации распределения вероятностей Гаусса с параметрами $a = 0$ и $\sigma = 1$.

- генерируется случайное число ξ_1 , равномерно распределённое в $[0, 1]$, и вычисляются две тригонометрические функции $s = \sin(2\pi\xi_1)$ и $c = \cos(2\pi\xi_1)$,
- генерируется ещё одно случайное число ξ_2 и вычисляется функция $R = \sqrt{-2\ln\xi_2}$,
- в итоге получаем сразу два независимых случайных числа, распределённых по закону Гаусса: $g_1 = R \cdot s$ и $g_2 = R \cdot c$.

3.2.3 Распределение Пуассона

Характерная ситуация, когда справедливо распределение Пуассона – измерение числа независимых событий, происшедших за фиксированное время, например, измерение количества срабатываний счётчика космических частиц за одну минуту. Функция распределения вероятностей

$$W_i = \frac{p^i}{i!} e^{-p} \quad (3.33)$$

характеризуется одним параметром $p = M_1 > 0$ — среднее значение числа событий. В пределе больших p распределение Пуассона переходит в распределение Гаусса с $a = p$ и $\sigma = \sqrt{p}$.

Один из возможных алгоритмов генерации распределения Пуассона:

1. Положить $i = -1, s = 1$;
2. Генерируется значение переменной ξ , равномерно распределённой в $[0, 1]$;

3. В счётчик i добавляется единица;
4. $s = s \cdot \xi$;
5. Если $s > \exp(-p)$, то переход на пункт 2;
6. Полученное значение счётчика i распределено по закону Пуассона со средним значением p .

3.2.4 Биномиальное распределение

Ещё один пример целочисленного (дискретного) распределения вероятностей кроме пуассоновского – биномиальное распределение. Это распределение реализуется, когда из N независимых событий n раз регистрируется результат, удовлетворяющий каким-то условиям отбора. Если вероятность реализации «нужного» результата в каждом событии равна q , то распределение вероятностей по n равно $W_n = \frac{N!q^n(1-q)^{N-n}}{n!(N-n)!}$. Если фиксировать произведение $Nq = p$ и одновременно устремить $N \rightarrow \infty$, то биномиальное распределение переходит в распределение Пуассона.

Если N не очень велико, то для генерации биномиального распределения можно использовать следующую процедуру.

- В счётчик «нужных» событий n заносится нуль.
- Начало цикла на N повторений.
 - Генерируется величина ξ , равномерно распределённая в $[0, 1]$.
 - Если $\xi < q$, то в счётчик n добавляется единица.
- Конец цикла.
- Переменная n распределена по искомому закону.

3.2.5 Экспоненциальное распределение

Экспоненциальное распределение используется при розыгрыше пробегов частиц в веществе до точки взаимодействия и времени жизни нестабильных частиц. Функция распределения вероятностей $f(r) = \alpha e^{-\alpha r}$. Случайная переменная ξ с равномерным распределением в $[0, 1]$ преобразуется в переменную $r = -\ln \xi / \alpha$ с экспоненциальной функцией распределения.

3.2.6 Распределение Коши

Распределение Коши имеет функцию распределения вероятностей

$$f(r) = \frac{\sigma}{\pi \cdot [\sigma^2 + (r - a)^2]}. \quad (3.34)$$

Это распределение замечательно тем, что не имеет ни одного момента. Случайная переменная этого распределения может быть получена по формуле

$$r = a + \sigma \cdot \operatorname{tg} \left(\pi \cdot \xi - \frac{\pi}{2} \right), \quad (3.35)$$

где ξ – случайная переменная, равномерно распределённая в интервале $(0, 1)$.

3.2.7 Основные характеристики распределений вероятности

После написания генератора какого-либо распределения для его проверки можно пользоваться не только визуальным контролем с помощью гистограммы, но и сравнить количественные характеристики распределений – моменты – с расчётными. Для оценки выборочных значений моментов можно пользоваться следующими формулами:

$$M_1(r) = \langle r \rangle = \frac{1}{n} \sum_{i=1}^n r_i \pm \frac{\sigma}{\sqrt{n}}, \quad (3.36)$$

$$\sqrt{C_2(r)} = \sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n r_i^2 - \frac{n}{n-1} \cdot \left[\frac{1}{n} \cdot \sum_{i=1}^n r_i \right]^2} \pm \frac{\sigma}{\sqrt{2n}}, \quad (3.37)$$

где n — объём выборки.

В табл. 3.3 приведены основные характеристики перечисленных в предыдущих подразделах распределений вероятности.

3.3 Получение произвольного распределения случайной величины

Методы генерации произвольных распределений в той или иной мере рассматриваются во многих учебниках [8, 9, 10]. Здесь изложение ближе всего к обзору, имеющемуся в работе [29].

Table 3.3: Основные характеристики некоторых часто используемых распределений вероятности

Распределение	Функция распределения	Интервал	Характеристическая функция	Производящая функция моментов	Математическое ожидание	Среднеквадратичное отклонение
Равномерное	1	$[0, 1]$	$\frac{\exp(it) - 1}{it}$	$\frac{\exp(x) - 1}{x}$	$\frac{1}{2}$	$\frac{1}{\sqrt{12}}$
Распределение Гаусса	$\frac{\exp[-\frac{(r-a)^2}{2\sigma^2}]}{\sqrt{2\pi}\sigma}$	$(-\infty, \infty)$	$e^{iat - \frac{t^2\sigma^2}{2}}$	$e^{ax + \frac{x^2\sigma^2}{2}}$	a	σ
Распределение Пуассона	$\frac{p^n}{n!}e^{-p}$	$[0, \infty)$	$\exp[p(e^{it} - 1)]$	$\exp[p(e^x - 1)]$	p	\sqrt{p}
Биномиальное	$\frac{N!q^n(1-q)^{N-n}}{n!(N-n)!}$	$[0, N]$	$[1 - q + qe^{it}]^N$	$[1 - q + qe^x]^N$	$N \cdot q$	$\sqrt{Nq(1-q)}$
Экспоненциальное	$\alpha e^{-\alpha r}$	$[0, \infty)$	$\frac{\alpha}{\alpha - it}$	$\frac{\alpha}{\alpha - x}$	$\frac{1}{\alpha}$	$\frac{1}{\alpha}$
Распределение Коши	$\frac{\sigma}{\pi [\sigma^2 + (r-a)^2]}$	$(-\infty, \infty)$	$\exp(iat - \sigma t)$	—	—	—

Не всегда есть возможность изобретать для какого-либо распределения вероятностей свой особый алгоритм генерации случайной величины с повышенной эффективностью работы (малым временем генерации). Иногда достаточно использовать какой-то из стандартных методов генерации. К тому же при изобретении своего особого алгоритма тоже полезно знать стандартные приёмы. В этом разделе мы рассмотрим несколько стандартных алгоритмов, которые дают статистически одинаковый результат, будучи применёнными к некоторому распределению вероятностей (если вообще алгоритм можно реализовать). Однако генераторы, написанные по разным алгоритмам, будут использовать разное количество времени CPU на генерацию очередного случайного числа. Поэтому для каждого конкретного случая необходимо выбрать наиболее подходящий для данного распределения алгоритм.

3.3.1 Метод прямой выборки

В случае непрерывного распределения вероятностей, описываемого функцией распределения $f(r)$ на интервале $[a, b]$, алгоритм включает следующие шаги.

- Находится интегральная функция распределения

$$\Phi(r) = \int_a^r f(\mu) d\mu.$$

Очевидно, что $\Phi(a) = 0$, а из условия нормировки

$$\Phi(b) = 1.$$

- Генерируется случайная величина ξ , равномерно распределённая в $[0, 1]$.
- Решается уравнение $\Phi(r) = \xi$. Значение корня этого уравнения r имеет распределение вероятностей $f(r)$.

В случае дискретного распределения W_i , где i меняется от 1 до m , алгоритм состоит из следующих шагов.

- Вычисляются частичные суммы $S_i = \sum_{j=1}^{j=i} W_j$ (интегральное распределение). Из условия нормировки $S_m = 1$.

3.3. ПОЛУЧЕНИЕ ПРОИЗВОЛЬНОГО РАСПРЕДЕЛЕНИЯ СЛУЧАЙНОЙ ВЕЛИЧИНЫ

- Генерируется случайная величина ξ , равномерно распределённая в $[0, 1]$.
- Определяется такой номер i , что $S_{i-1} < \xi \leq S_i$ (здесь принимается, что $S_0 = 0$). Этот номер и является искомой реализацией распределения.

Если распределение вероятностей является частично непрерывным, а частично дискретным, то эти два алгоритма необходимо комбинировать.

3.3.2 Метод Неймана (реджекций)

В некоторых случаях реализация метода прямой выборки является непростой задачей, требующей вычисления сложного интеграла – функции $\Phi(r)$, и последующего решения уравнения $\Phi(r) = \xi$, алгебраического с высокой степенью полинома или трансцендентного. Тогда можно попробовать применить метод Неймана.

Метод Неймана является наиболее простым в реализации, однако для некоторых функций распределения он оказывается весьма неэффективным, а в ряде случаев его вообще невозможно применить.

Функция распределения $f(r)$ в данном методе не обязана быть нормированной. С другой стороны, есть существенные требования: интервал изменения r должен быть ограничен $-\infty < a \leq r \leq b < +\infty$, и на всём интервале $r \in [a, b]$ должна быть ограничена функция распределения вероятностей: $0 \leq f(r) \leq M < \infty$.

1. Генерируется случайная величина r , равномерно распределённая в $[a, b]$. Из величины ξ_1 , равномерно распределённой в $[0, 1]$, можно смещением и растяжением получить $r = a + (b - a)\xi_1$.
2. Генерируется случайная величина μ , равномерно распределённая в интервале $[0, M]$: $\mu = \xi_2 \cdot M$, где ξ_2 равномерно распределена в $[0, 1]$.
3. Если $f(r) < \mu$, то возврат на пункт 1.
4. Значение r соответствует распределению $f(r)$.

3.3.3 Метод существенной выборки

В случае очень неравномерной функции распределения метод Неймана оказывается неэффективным. Тогда более перспективным может оказаться

метод существенной выборки, идея которого заключается в том, чтобы в модифицированном методе Неймана случайные точки по оси x выбирать не равномерно, а с большей вероятностью в области пика функции распределения.

Для применения этого метода необходимо функцию распределения представить в виде

$$f(r) = g(r) \cdot h(r), \text{ где } g(r) \geq 0, \quad 0 \leq h(r) \leq M.$$

Метод оказывается достаточно эффективным, если распределение $g(r)$ легко генерируется методом прямой выборки, а выражение

$$M \cdot \int_a^b g(r) dr / \int_a^b f(r) dr$$

имеет малое значение, близкое к единице.

Алгоритм состоит из следующих шагов:

1. Генерируется случайная величина r , имеющая распределение вероятностей $g(r)$ в интервале $[a, b]$.
2. Генерируется случайная величина μ , равномерно распределённая в интервале $[0, M]$.
3. Если $\mu > h(r)$, то возврат на пункт 1 и всё повторяется сначала.
4. Последнее значение r является искомым значением случайной величины, распределённой по закону $f(r)$.

3.3.4 Метод композиций

Этот метод эффективен тогда, когда функцию распределения удастся представить в виде суммы функций распределения вероятностей таких, что каждая из них легко генерируется методом прямой выборки: $f(r) = \sum_{i=1}^{i=k} \alpha_i g_i(r)$. При этом должно быть $\alpha_i > 0$. Тогда алгоритм выборки очередного случайного числа r представляется в следующем виде:

- В соответствии с дискретным распределением вероятностей α_i выбирается случайный номер i функции распределения $g_i(r)$. Это всегда возможно, так как в силу условия нормировки $\int_a^b f(r) dr = 1$, $\int_a^b g_i(r) dr = 1$, распределение α_i тоже нормировано на единицу: $\sum_{i=1}^{i=k} \alpha_i = 1$.

3.3. ПОЛУЧЕНИЕ ПРОИЗВОЛЬНОГО РАСПРЕДЕЛЕНИЯ СЛУЧАЙНОЙ ВЕЛИЧИНЫ

- Для выбранной функции $g_i(r)$ методом прямой выборки генерируется случайная величина r .
- Полученная таким способом переменная r имеет распределение вероятностей $f(r)$.

3.3.5 Метод композиций и реджекций

Этот метод является комбинацией метода композиции и метода Неймана. Пусть функцию распределения можно представить в виде суммы функций распределения: $f(r) = \sum_{i=1}^{i=k} \alpha_i \cdot h_i(r)$, где $0 \leq h_i(r) \leq M_i$, $\alpha_i > 0$. Тогда метод композиций и реджекций может быть применён по следующей схеме.

- Генерируется случайный номер функции i в соответствии с дискретным распределением $\beta_i = \alpha_i \cdot M_i / \left(\sum_{j=1}^{j=k} \alpha_j \cdot M_j \right)$.
- Генерируется случайная переменная r , равномерно распределённая в интервале $[a, b]$.
- Генерируется случайная переменная μ , равномерно распределённая в интервале $[0, M_i]$.
- Если $\mu > h_i(r)$, то всё повторяется сначала.
- Значения переменной r , прошедшие отбор в предыдущем пункте, распределены в соответствии с функцией распределения $f(r)$.

Этот метод, реализованный в строгом соответствии с этим простым алгоритмом, не имеет никакой практической ценности, так как по эффективности уступает простому методу Неймана. Это можно заметить, вычислив среднее количество раз прохождения указанного цикла до получения очередного случайного числа r , распределённого по $f(r)$: $\langle n \rangle = \left(\sum_{j=1}^{j=k} \alpha_j M_j \right) / \left(\int_a^b f(r) dr \right)$. Отсюда становится очевидной модификация алгоритма, которая может давать хороший эффект. Для этого разложение функции распределения должно быть выполнено в следующем виде: по-прежнему ветвь алгоритма с функцией распределения $h_1(r)$ предназначена для моделирования методом Неймана, поэтому $0 \leq h_1(r) \leq M_1$, однако все остальные функции должны использоваться для генерации методом прямой

выборки. Так как коэффициенты α_k не являются вероятностями перехода на соответствующую ветвь алгоритма и не упрощают описания, положим их равными единице, т.е. пусть ненормированная функция распределения $f(r)$ представляется на интервале (a, b) в виде

$$f(r) = \sum_{i=1}^k h_i(r),$$

где все $h_i(r) \geq 0$ на интервале (a, b) . Кроме того, интегралы от всех $h_i(r)$, $i = 2, 3, \dots, k$ выражаются в элементарных функциях

$$H_i(x) = \int_a^x h_i(r) dr$$

и позволяют решать соответствующее уравнение в рамках метода прямой выборки. Введём ещё константы

$$\gamma_i = H_i(b), \quad i = 2, 3, \dots, k.$$

Теперь алгоритм приобретает следующий вид:

1. Генерируется случайный номер i функции $h_i(r)$ в соответствии с дискретным распределением β_i , где

$$\beta_1 = M_1(b - a) / \left(M_1(b - a) + \sum_{j=2}^{j=k} \gamma_j \right),$$

$$\beta_i = \gamma_i / \left(M_1(b - a) + \sum_{j=2}^{j=k} \gamma_j \right) \text{ при } i > 1.$$

2. Если $i > 1$, то переход на пункт 6 для генерации по методу прямой выборки.
3. Генерируется случайная переменная r , равномерно распределённая в интервале $[a, b]$.
4. Генерируется переменная μ , равномерно распределённая в интервале $[0, M_1]$.

3.3. ПОЛУЧЕНИЕ ПРОИЗВОЛЬНОГО РАСПРЕДЕЛЕНИЯ СЛУЧАЙНОЙ ВЕЛИЧИНЫ

5. Если $\mu > h_1(r)$, то переход на пункт 1, и всё повторяется сначала. В противном случае значение r считается принятым и делается переход на пункт 7.
6. Методом прямой выборки генерируется случайная переменная r в соответствии с распределением $h_i(r)$, т.е. решается уравнение

$$H_i(r) = \xi \cdot \gamma_i,$$

где ξ — случайное число с равномерным распределением в интервале $(0, 1)$.

7. Полученное значение переменной r выдаётся в качестве очередного случайного числа, имеющего распределение вероятностей $f(r)$.

Для модифицированного алгоритма снова можно посчитать среднее число повторений цикла, пока не будет получено «полезное» случайное число:

$$\langle n \rangle = \left[1 - \beta_1 \cdot \left(1 - \frac{\gamma_1}{M_1(b-a)} \right) \right]^{-1}.$$

Видно, что алгоритм тем эффективнее, чем ближе M_1 к своему пределу $\gamma_1/(b-a)$ (минимальное возможное значение $\min M_1 = \gamma_1/(b-a)$ для функции распределения $h_1(r)$ достигается при $h_1(r) \equiv \text{const}$).

Пример. Пусть ненормированная функция распределения на интервале $0 < x < 1$ равна

$$f(x) = \frac{1}{\sqrt{x}} + \frac{e^x}{\sqrt{1-x^2}}.$$

Выберем $h_2(x) = \frac{1}{\sqrt{x}}$, $h_3 = \frac{e \cdot x}{\sqrt{1-x^2}}$ — функции, удобные для генерации случайной переменной x :

$$H_2(x) = \int_0^x \frac{d\xi}{\sqrt{\xi}} = 2\sqrt{x}, \quad H_3(x) = \int_x^1 \frac{e \cdot \xi d\xi}{\sqrt{1-\xi^2}} = e \cdot \sqrt{1-x^2}.$$

Здесь для удобства решения получающегося уравнения для интегральной функции распределения $H_3(x)$ выбран интеграл не от нижнего предела до текущей координаты, а от x до верхнего предела — это вполне допустимо. Отсюда получаем $\gamma_2 = 2$, $\gamma_3 = e$,

$$h_1(x) = \frac{e \cdot (e^{x-1} - x)}{\sqrt{1-x^2}}.$$

Анализируя поведение функции $h_1(x)$, можно убедиться, что

$$M_1 = \max_{x \in (0,1)} h_1(x) = h_1(0) = 1, \quad \lim_{x \rightarrow 1} h_1(x) = 0.$$

Соответствующие вероятности для переходов на три ветви алгоритма получаются равными

$$\beta_1 = \frac{1}{3+e}, \quad \beta_2 = \frac{2}{3+e}, \quad \beta_3 = \frac{e}{3+e}.$$

Теперь алгоритм генерации случайной величины x с распределением вероятностей $f(x)$ может проводиться по следующей схеме (здесь последовательные независимые случайные числа с равномерным распределением в $(0, 1)$ обозначены как ξ_k).

1. Делается выборка случайного числа ξ_1 . Если $\xi_1 < \beta_1$, то выполняется следующий пункт, иначе переход на пункт 3.
2. Делается выборка чисел ξ_2 и ξ_3 . Если $h_1(\xi_2) < \xi_3$, то возврат на пункт 1, иначе $x = \xi_2$ и переход на пункт 5.
3. Если $\xi_1 > \beta_1 + \beta_2$, то переход на пункт 4, иначе выборка очередного ξ_4 , $x = \xi_4^2$ и переход на пункт 5.
4. Выборка очередного ξ_5 . Полагаем $x = \sqrt{1 - \xi_5^2}$ и переходим на пункт 5.
5. В x содержится очередное значение случайной переменной, распределённой по закону $f(x)$.

Оценив численное значение интеграла

$$\gamma_1 = \int_0^1 h_1(x) dx = \int_0^1 \frac{e \cdot (e^{x-1} - x) dx}{\sqrt{1-x^2}} \approx 0.386,$$

получаем среднее число повторений цикла до генерации очередного случайного числа по заданному распределению:

$$\langle n \rangle = \left[1 - \beta_1 \cdot \left(1 - \frac{\gamma_1}{M_1(b-a)} \right) \right]^{-1} \approx 1.12$$

Получается, что почти всегда случайная величина x генерируется за один проход цикла генерации.

3.4 Генерация нескольких случайных величин при наличии корреляций

На практике часто встречаются случаи, когда задано совместное распределение вероятностей для нескольких случайных величин $f(x_1, x_2, \dots, x_k)$, причём каждая случайная величина $x_i \in (a_i, b_i)$, и распределение вероятностей по каждой величине зависит от значения остальных величин. В этом случае говорят о наличии корреляции между случайными величинами.

Некоторые из рассмотренных методов легко обобщаются на случай нескольких случайных величин с учётом корреляции (например, метод Неймана; ввиду очевидности этой процедуры мы не будем её рассматривать). В любом случае, по крайней мере теоретически, задачу всегда можно свести к последовательной генерации случайных величин следующей процедурой.

Введём функции условного распределения вероятностей

$$f_i(x_1, x_2, \dots, x_i) = \frac{\int_{a_{i+1}}^{b_{i+1}} dx_{i+1} \cdots \int_{a_k}^{b_k} f(x_1, x_2, \dots, x_k) dx_k}{\int_{a_i}^{b_i} dx_i \cdots \int_{a_k}^{b_k} f(x_1, x_2, \dots, x_k) dx_k}. \quad (3.38)$$

Очевидно, что каждая функция $f_i(x_1, x_2, \dots, x_i)$ нормирована на единицу ($\int_{a_i}^{b_i} f_i(x_1, \dots, x_i) dx_i = 1$), если исходная функция f нормирована. Теперь возможна поочерёдная генерация случайных величин в следующей последовательности.

- Генерируется значение случайной величины x_1 в соответствии с распределением вероятностей $f_1(x_1)$.
- Генерируется значение случайной величины x_2 в соответствии с распределением $f_2(x_1, x_2)$ при фиксированном значении x_1 .
- Продолжается генерация следующих величин, причём всегда в условном распределении вероятностей ранее полученные случайные величины рассматриваются как фиксированные параметры распределения.

Количественно степень корреляции можно характеризовать ковариационной матрицей $v_{ij} = \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rangle$ [12], в которой диагональные элементы равны дисперсии соответствующих величин, а недиагональные характеризуют парную корреляцию случайных

величин. Иногда для характеристики корреляции вводят коэффициент линейной корреляции $\rho_{ij} = \frac{v_{ij}}{\sqrt{v_{ii}v_{jj}}}$, который интересен тем, что принимает значения в интервале $[-1, 1]$. Когда $\rho_{ij} = 1$, говорят о 100%-й положительной корреляции, когда $\rho_{ij} = -1$, то о 100%-й отрицательной корреляции. Если $\rho_{ij} = 0$, то это говорит об отсутствии линейной корреляции.

Полностью характеризует корреляцию переменных только совместная плотность распределения вероятностей. Если ковариационная матрица недиагональна, то корреляция определённо есть, однако обратное неверно: можно привести много примеров распределения вероятностей, когда есть корреляция, а ковариационная матрица диагональная, т.е. коэффициент линейной корреляции равен нулю. Например, функция распределения $f(x_1, x_2) = \delta(x_1^2 + x_2^2 - R^2)/\pi$ даёт коэффициент корреляции $\rho_{12} = 0$, в то время как корреляция, можно сказать, 100%-я, так как при любом значении $x_1 \in (-R, +R)$ x_2 может принимать только фиксированные значения $\pm\sqrt{R^2 - x_1^2}$.

Иногда возникает необходимость генерировать набор Гауссовых случайных переменных $x_k, k = 1, \dots, n$, степень корреляции которых задана ковариационной матрицей H_{ik} . Элементы матрицы равны

$$H_{ik} = \overline{(x_i - \bar{x}_i)(x_k - \bar{x}_k)}.$$

Симметричная матрица \mathbf{H} , если она соответствует реальным ковариациям некоторых случайных величин, по определению должна иметь неотрицательные собственные значения, что даёт возможность построить набор гауссовых переменных, имеющих такую же ковариационную матрицу. Очевидно, что линейная корреляция между переменными x_k может быть обеспечена в общем виде линейным преобразованием

$$x_k = \bar{x}_k + \sum_{i=1}^n T_{ki} g_i, \quad (3.39)$$

где переменные $g_i, i = 1, \dots, n$ — независимые Гауссовы переменные с единичной дисперсией. В матричном виде

$$\Delta\mathbf{X} = \mathbf{T}\mathbf{G}, \quad \mathbf{H} = \mathbf{T}\mathbf{T}^T. \quad (3.40)$$

В этом матричном уравнении неизвестных элементов t_{ij} больше, чем уравнений (из-за симметрии матрицы \mathbf{H}). Количество неизвестных равно n^2 , а количество уравнений — $\frac{n(n+1)}{2}$. Поэтому решений имеется

бесконечное множество. Одно из них можно получить, используя преобразование поворота \mathbf{U} , приводящее \mathbf{H} к диагональному виду $\mathbf{\Lambda}$:

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \implies \mathbf{T} = \mathbf{U}\sqrt{\mathbf{\Lambda}}. \quad (3.41)$$

Любая матрица, полученная умножением \mathbf{T} на любую матрицу поворота, тоже будет решением. Очевидно, что можно подобрать эту матрицу таким образом, что решение будет представляться треугольной матрицей. Такое решение в виде треугольной матрицы однозначно, и может быть получено следующим алгоритмом.

1. $T_{1,1} = \sqrt{H_{1,1}}$
 $k = 1$
2. $k = k + 1$
 $i = 1$
3. $T_{k,i} = \frac{H_{k,i} - \sum_{j=1}^{i-1} T_{k,j}T_{i,j}}{T_{i,i}}$
 $T_{i,k} = 0$
4. $i = i + 1$
Если $i < k$, то возврат на п. 3.
5. $T_{k,k} = \sqrt{H_{k,k} - \sum_{j=1}^{k-1} T_{k,j}^2}$
6. Если $k < n$, то возврат на п. 2.

Если в п. 5 корень квадратный невозможно извлечь (по причине отрицательного подкоренного выражения), то это свидетельствует о том, что ковариационная матрица не положительно определённая и точного решения не существует.

3.5 Интегрирование методом Монте – Карло

3.5.1 Метод среднего

Метод среднего значения является простейшим способом интегрирования произвольной функции одной или нескольких переменных. Обоснование этого метода продемонстрируем на примере функции одной

переменной. Пусть требуется найти определённый интеграл $R = \int_a^b \Phi(r) dr$. Перепишем подынтегральное выражение таким образом, чтобы интеграл представлял собой запись среднего значения некоторой функции случайной величины $R = \int_a^b [(b - a)\Phi(r)] \cdot \frac{dr}{b-a}$. Здесь выражение $\frac{dr}{b-a}$ представляет собой дифференциал функции распределения вероятности переменной r , равномерно распределённой в интервале $[a, b]$, а значит, весь интеграл представляет собой среднее значение функции $(b - a)\Phi(r)$ от случайной переменной r . Оценка среднего значения функции может быть сделана по следующему алгоритму (обобщаем на случай n аргументов):

- Зануляется переменная $R = 0$.
- Начало цикла по генерации случайной функции на N повторений.
 - генерируются случайные величины r_i , равномерно распределённые в интервалах $[a_i, b_i]$, $i = 1, 2, \dots, n$;
 - в накопитель R суммируется значение случайной функции

$$R = R + \left[\prod_{i=1}^{i=n} (b_i - a_i) \right] \cdot \Phi(r_1, r_2, \dots, r_n).$$

- Конец цикла.
- Искомая оценка среднего значения равна R/N .

Ошибка расчёта методом Монте – Карло оценивается обычными статистическими методами, рассматриваемыми в разд. 6.

3.5.2 Выделение главной части

Если интегрируемая функция $\Phi(r)$ очень неоднородна, т.е. может сильно меняться в пределах области интегрирования, то статистическая ошибка может быть значительной. Ошибку расчёта можно существенно уменьшить, если удастся разбить функцию на два слагаемых $\Phi(r) = \Phi_1(r) + \Phi_2(r)$ таким образом, что первая функция $\Phi_1(r)$ интегрируется аналитически, поэтому не имеет статистической ошибки, а вторая функция $\Phi_2(r)$ более однородная, чем исходная, и при интегрировании методом среднего даёт меньшую статистическую ошибку.

3.5.3 Метод существенной выборки

Искомый интеграл можно переписать следующим образом:

$$R = \int_a^b [\Phi(r)/g(r)] \cdot g(r) dr,$$

что совпадает с записью выражения для среднего значения функции $\Phi(r)/g(r)$ случайной величины r , распределённой по закону $g(r)$. Из того, что $g(r)$ должна быть функцией распределения вероятностей, вытекает существенное ограничение на её выбор: $g(r) > 0$; кроме того, функция $g(r)$ должна быть «удобной» функцией распределения для генерации случайной переменной r по методу прямой выборки (см. разд. 3.3.1). После выбора аппроксимирующей функции распределения $g(r)$ такой, что $\Phi(r)/g(r) \approx \text{const}$, схема алгоритма может выглядеть так:

- Зануляется переменная $R = 0$.
- Начало цикла на N повторений;
 - генерируется случайная величина r с распределением вероятностей $g(r)$ в интервале $[a, b]$ методом прямой выборки;
 - Значение усредняемой функции добавляется в сумматор $R = R + \Phi(r)/g(r)$;
- Конец цикла.
- Полученное среднее значение $R/N \approx \int_a^b \Phi(r) dr$.

3.5.4 Метод Метрополиса

В работе [30] предложен метод «случайного блуждания» в n -мерном пространстве в соответствии с заданным распределением вероятностей $f(\mathbf{x})$. Если искомый интеграл удаётся представить в виде выражения для среднего от некоторой функции $g(\mathbf{x})$, то среднее значение в виде интеграла можно заменить на среднее значение функции g по последовательности точек:

$$\int_{\Omega} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{i=N} g(\mathbf{x}^{(i)}), \quad \text{где} \quad \int_{\Omega} f(\mathbf{x}) d\mathbf{x} = 1. \quad (3.42)$$

Метод Метрополиса описан в работах [31, 32]. Приведём алгоритм генерации очередной точки последовательности $\mathbf{x}^{(i+1)}$, если уже получена точка $\mathbf{x}^{(i)}$. Самая первая точка выбирается достаточно произвольно.

- Генерируется случайная точка с координатами $x_j^{(i+1)}$, каждая из которых имеет независимое от других координат равномерное распределение в интервале $[x_j^{(i)} - \delta_j, x_j^{(i)} + \delta_j]$, где δ_j – некоторые малые константы.
- Если полученная точка выходит за область интегрирования, то новая точка принимается равной старой.
- Если пробная точка находится в области интегрирования, то вычисляется отношение $k = \frac{f(\mathbf{x}^{(i+1)})}{f(\mathbf{x}^{(i)})}$. Генерируется случайное число ξ , равномерно распределённое в интервале $[0, 1]$. Если $\xi < k$, то пробная точка принимается в качестве новой точки последовательности, в противном случае новая точка принимается равной старой $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)}$.

Полученная последовательность точек при достаточно большом N в среднем распределена в соответствии с функцией $f(\mathbf{x})$.

Метод внешне похож на метод существенной выборки, однако имеет сравнительные достоинства и недостатки. К достоинствам следует отнести возможность реализации при любой функции распределения f и простоту алгоритма независимо от числа переменных и области интегрирования. При очень большом числе переменных, как в некоторых задачах статистической физики, этот метод, по-видимому, является единственно возможным из методов Монте – Карло. Недостатком является проблема выбора констант δ_j (нет чётких рекомендаций на этот счёт), отсутствие надёжных методов оценки ошибки расчёта. Кроме того, если распределение вероятностей имеет несколько максимумов, то переход из области одного максимума в область другого в процессе блуждания затруднён.

Рисунок 3.1 демонстрирует скорость сходимости оценки интеграла к точному значению для случая, когда ответ известен:

$$\begin{aligned} R &= \int_0^4 dx \int_0^4 dy \int_0^4 dz (x^2 + y^2 + z^2 - 12)^2 x^2 y^2 z^2 e^{-x-y-z} = \\ &= 192(51 - 3694e^{-4} + 55787e^{-8})(1 - 13e^{-4}) \approx 300,83098. \end{aligned} \quad (3.43)$$

С учётом требования нормировки выберем функцию распределения

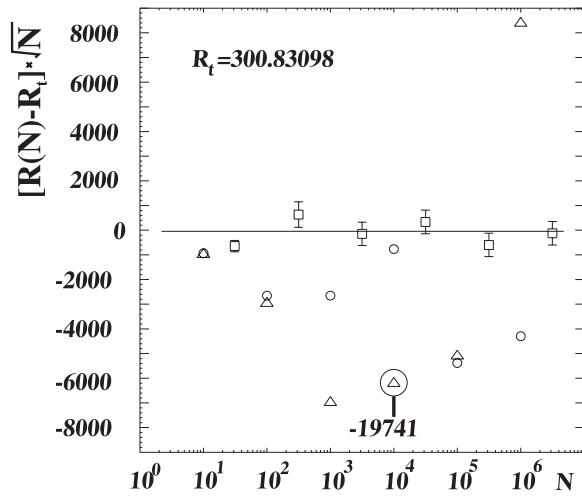


Figure 3.1: Отклонение оценки интеграла в методе Метрополиса (\circ и \triangle) и методе среднего (\square) от точного значения, умноженное на \sqrt{N} , в зависимости от числа испытаний N

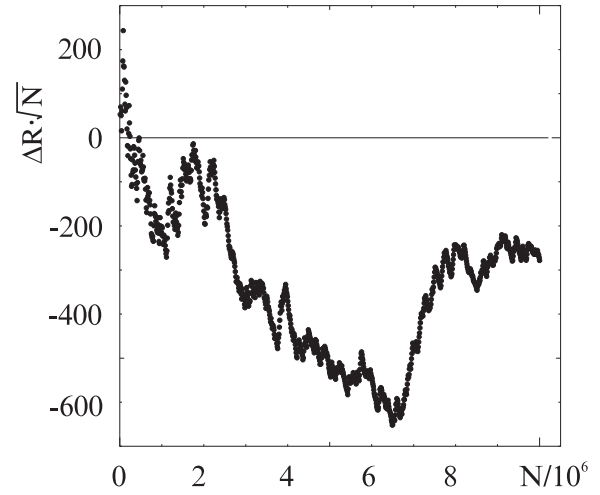


Figure 3.2: Отклонение оценки интеграла в методе Метрополиса от точного значения, умноженное на \sqrt{N} , в зависимости от числа испытаний N . Функция распределения вероятностей имеет три максимума.

вероятностей в виде

$$f(x, y, z) = \frac{x^2 y^2 z^2 \exp(-x - y - z)}{8(1 - 13e^{-4})^3}, \quad (3.44)$$

а усредняемая функция тогда приобретает вид

$$g(x, y, z) = 8(1 - 13e^{-4})^3 (x^2 + y^2 + z^2 - 12)^2. \quad (3.45)$$

Две серии расчётов (кружочки и треугольники) различаются начальным случайным числом генератора случайных чисел. В ранее обсуждавшихся алгоритмах вычисления интегралов методом Монте – Карло сходимость результата к точному решению всегда имела зависимость $\sim \frac{1}{\sqrt{N}}$, где N – полное число испытаний. Для того чтобы расчёты показывали скорость сходимости относительно этой стандартной, по оси ординат откладывается отклонение результата от точного значения, умноженное на \sqrt{N} . Кроме того, на этом же рисунке квадратиками с ошибками приведены результаты вычисления этого же интеграла методом среднего (оценка ошибки проводится стандартными статистическими методами). Видно, что

для данного случая сходимость метода Метрополиса к точному результату имеет место, но существенно хуже, чем сходимость метода среднего.

Возьмём другой пример, иллюстрирующий проблему сходимости при наличии нескольких максимумов в функции распределения. Пусть функция распределения вероятностей равна

$$f(x, y, z) = \frac{1}{3(\sqrt{2\pi}\sigma)^3} \times \left[e^{-\frac{(x-1)^2+y^2+z^2}{2\sigma^2}} + e^{-\frac{x^2+(y-1)^2+z^2}{2\sigma^2}} + e^{-\frac{x^2+y^2+(z-1)^2}{2\sigma^2}} \right]. \quad (3.46)$$

При малых значениях σ функция имеет три равноудалённых максимума с примерной высотой

$$\max f(x, y, z) \approx \frac{1}{3(\sqrt{2\pi}\sigma)^3},$$

в то же время на пути от одного максимума к другому неизбежен «провал» до уровня

$$\frac{2}{3(\sqrt{2\pi}\sigma)^3} e^{-\frac{1}{4\sigma^2}}.$$

Возьмём значение $\sigma = 0,2$ и усредняемую функцию

$$g(x, y, z) = 3x - 2y - z \quad (3.47)$$

и посмотрим, как будет меняться оценка интеграла методом Метрополиса по мере увеличения статистики. Точное значение интеграла легко вычисляется:

$$\int f(x, y, z)g(x, y, z) dx dy dz = 0. \quad (3.48)$$

На рис. 3.2 представлена зависимость отклонения оценки интеграла от точного значения, умноженной на нормирующий множитель \sqrt{N} , от количества шагов N в траектории случайного блуждания. Если бы все $N = 10^7$ точек вычисления функции распределялись случайно по окрестностям максимумов, то в области каждого максимума было бы 3333333 ± 1491 , однако, в действительности в районе X -максимума оказалось 3158613 точек вычисления функции, в районе Y -максимума – 3528816, и Z -максимума – 3312571, причём переходов из области одного максимума в другую область было всего 3669.

3.6 Моделирование в ядерном эксперименте на примере регистрации событий в детекторе

Рассмотрим задачу об определении вероятности регистрации пары заряженных каонов, рождающихся при аннигиляции электрона с позитроном, летящих навстречу друг другу с равными по величине и противоположными по направлению импульсами. Рождающиеся каоны регистрируются детектором, состоящим из двух счётчиков, симметрично расположенных относительно места встречи. Каждый счётчик представляет собой плоскую прямоугольную пластину длиной $2a$ (вдоль оси z) и шириной $2b$ (вдоль оси x) на расстоянии h от плоскости xz (рис. 3.3). Под вероятностью регистрации понимается отношение числа событий, в которых оба каона зарегистрированы детектором, к полному числу событий рождения пары каонов.

Точки рождения каонов распределены вдоль оси пучков по Гауссу со среднеквадратичным отклонением σ_z и средним в центре симметрии системы.

Распределение по углам вылета каонов даётся дифференциальным сечением процесса, которое в случае неполяризованных электрона и позитрона зависит только от полярного угла θ между направлением вылета каона (например, K^-) и осью z (ось e^+e^- пучков) и даётся формулой

$$\frac{d\sigma}{d\Omega} \sim \sin^2 \theta.$$

Так как каон нестабильная частица, распадающаяся в покое со средним временем жизни $\tau = 1,24 \cdot 10^{-8}$ с, то случайная длина пробега до распада каона на лету даётся экспоненциальным законом со средней длиной пробега

$$\lambda = c\tau \sqrt{\left(\frac{E}{m}\right)^2 - 1},$$

где E – полная энергия каона, совпадающая в нашем случае с энергией начального электрона (позитрона), m – масса каона. Легко сообразить, что на математическом языке задача вычисления эффективности регистрации сводится к вычислению многомерного интеграла (размерность не менее 10) с достаточно простой подынтегральной функцией и весьма сложными пределами интегрирования, так как диапазон изменения одной переменной

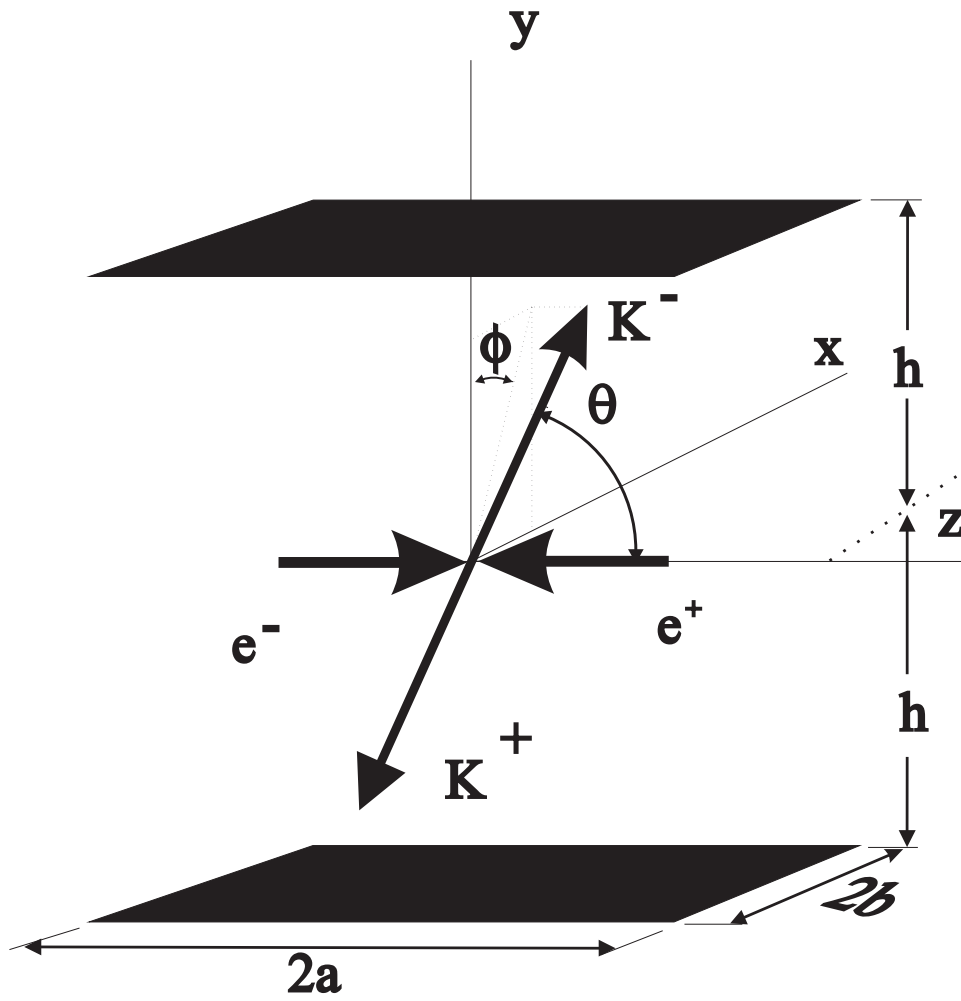


Figure 3.3: Моделирование процесса рождения пары заряженных каонов при столкновении электрона и позитрона

является функцией других переменных. Идеальным для решения этой задачи является имитационный метод Монте – Карло, в котором с помощью случайных чисел, распределённых по нужному закону в соответствии с природой явления, моделируется (имитируется) шаг за шагом «судьба» каждого из каонов. Примерный алгоритм решения методом Монте – Карло выглядит следующим образом:

1. Моделируется случайная точка z_0 столкновения электрона с позитроном (по закону Гаусса).
2. Моделируется случайный угол θ для отрицательного каона (по закону $\sin^2 \theta$).
3. Моделируется случайный угол φ , равномерно распределённый на интервале $[0, 2\pi]$.
4. Находится точка пересечения прямой, по которой летит каон, с плоскостью счётчиков и проверяется, находится ли она внутри границ счётчика. Если каон не попал ни в один из счётчиков, идём на п. 1, если попал, то идём на п. 5.
5. Моделируем случайную длину пробега до распада по экспоненциальному закону. Если она меньше расстояния до счётчика, определённого в п. 4, то каон считается распавшимся до регистрации, т. е. данное событие не даёт срабатывания детектора и нужно перейти на моделирование следующего события (п. 1). Если каон не распался до счётчика, то начинаем рассмотрение второго каона, для которого, как видно из рис. 3.3,

$$\theta = \pi - \theta, \quad \varphi = \varphi + 2\pi,$$

и можно начинать с п. 4.

6. Если оба каона попали в детектор, он срабатывает, и такое событие считается полезным. Содержимое счётчика полезных событий увеличивается на один, и начинаем моделирование следующего события.

Ниже приведён примерный вариант программы на Фортране, реализующий описанный алгоритм и снабжённый подробными комментариями. В программе используются некоторые стандартные подпрограммы из библиотеки научных подпрограмм ЦЕРН. Заметим,

что при любых вычислениях методом Монте – Карло мы рекомендуем не ограничиваться выводом (печатью) лишь окончательных результатов. Для контроля полезно разбить всю набираемую статистику на независимые группы и выводить промежуточные результаты для статистики, набранной в данной группе событий. Сравнение результатов по отдельным группам позволяет оценить эффективность выбранного алгоритма моделирования, и если отдельные группы дают статистически согласующиеся результаты, объединить их и воспользоваться результатами по всей статистике.

```

PROGRAM KAON
COMMON/PART/P(3,2)
DIMENSION D(2)
DATA ZO,SIGZ,BGCT/0.,1.,95.34/
DATA A,B,H/30.,30.,20./
DATA NPART,NGROUP,NSTAT/2,10,500/
IDET=0
DO 100 I=1,NGROUP
  KDET=0
  DO 90 K=1,NSTAT
C
C-----МОДЕЛИРОВАНИЕ ТОЧКИ ВЫЛЕТА КАОНОВ ПО Z-----
C  ZO - ЦЕНТР СТОЛКНОВЕНИЯ ЭЛЕКТРОНА С ПОЗИТРОНОМ,
C  SIGZ - РАЗМЕР МЕСТА ВСТРЕЧИ,
C  NORRAN - ПОДПРОГРАММА, ГЕНЕРИРУЮЩАЯ СЛУЧАЙНОЕ ЧИСЛО,
C  РАСПРЕДЕЛЕННОЕ ПО ГАУССУ СО СРЕДНИМ 0 И ДИСПЕРСИЕЙ 1
C
      CALL NORRAN(Z)
      Z=ZO+Z*SIGZ
C
C ----- МОДЕЛИРОВАНИЕ НАПРАВЛЕНИЯ ВЫЛЕТА ЧАСТИЦЫ -----
C  СТ = COS(ТЕТА), ST=SIN(ТЕТА),
C  RNDM - ДАТЧИК СЛУЧАЙНЫХ ЧИСЕЛ, РАВНОМЕРНО РАСПРЕДЕЛЕННЫХ НА [0,1]
C
10      СТ=-1.+2.*RNDM(R)
          ST2 = 1.-СТ**2
          IF(RNDM(R).GT.ST2) GO TO 10
          ST = SQRT(ST2)
C
C  SPHI = SIN(PHI), CPHI = COS(PHI)
C  RAN2VS - ПОДПРОГРАММА, ГЕНЕРИРУЮЩАЯ СИНУС И КОСИНУС УГЛА PHI,
C  РАВНОМЕРНО РАСПРЕДЕЛЕННОГО УГЛА НА [0,6.2832]
C

```

3.6. МОДЕЛИРОВАНИЕ В ЯДЕРНОМ ЭКСПЕРИМЕНТЕ НА ПРИМЕРЕ РЕГИСТ

```

                CALL RAN2VS(1.,SPHI,CPHI)
C
C-----НАПРАВЛЯЮЩИЕ КОСИНУСЫ ВЫЛЕТА-----
C   P(1,I) - ПО X, P(2,I) - ПО Y, P(3,I) - ПО Z ДЛЯ I-ОЙ ЧАСТИЦЫ
C
                P(1,I)=ST*SPHI
                P(2,I)=ST*CPHI
                P(3,I)=CT
                DO 15 L=1,3
                    P(L,2)=-P(L,1)
15             CONTINUE
                NDET=0
                DO 80 L=1,NPART
C
C----- ПРОВЕРКА ПОПАДАНИЯ ЧАСТИЦЫ В СЧЕТЧИК -----
C
                D(1)=H/P(2,L)
                D(2)=-H/P(2,L)
                DO 20 M=1,2
                    X=D(M)*P(1,L)
                    IF(ABS(X).GT.B)GO TO 20
                    Z=Z0+D(M)*P(3,L)
                    IF(ABS(Z).GT.A)GO TO 20
C
C----- МОДЕЛИРОВАНИЕ ПРОБЕГА ДО РАСПАДА -----
C   RANGE - ДЛИНА ПРОБЕГА ДО РАСПАДА,
C   ABS(D(M)) - ДЛИНА ПУТИ ДО СЧЕТЧИКА
C
                RANGE=-BGCT*LOG(RNDM(R))
                IF(RANGE.LT.ABS(D(M))GO TO 20
                NDET=NDET+1
20             CONTINUE
80             CONTINUE
                IF(NDET.LT.2)GO TO 90
                KDET=KDET+1
                IDET=IDET+1
90             CONTINUE
C
C   ПРОМЕЖУТОЧНЫЕ РЕЗУЛЬТАТЫ: KDET - ЧИСЛО СОБЫТИЙ, ДАВШИХ ЗАПУСК
C   В ОЧЕРЕДНОЙ ГРУППЕ, ERSV - ВЕРОЯТНОСТЬ РЕГИСТРАЦИИ ПО
C   СТАТИСТИКЕ ДАННОЙ ГРУППЫ, ERSE - ЕЕ ПОГРЕШНОСТЬ
C

```

```

      EPCV=KDET*1./NSTAT
      EPCE=SQRT(EPCV*(1.-EPCV)/NSTAT)
      WRITE (6,*) K,KDET,EPCV,EPCE
100 CONTINUE
C
C ОКОНЧАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ: IDET - ЧИСЛО СОБЫТИЙ, ДАВШИХ ЗАПУСК
C ПО ВСЕЙ СТАТИСТИКЕ, DEPCV - ВЕРОЯТНОСТЬ РЕГИСТРАЦИИ ПО
C ВСЕЙ СТАТИСТИКЕ, DEPCE - ЕЕ ПОГРЕШНОСТЬ
C
      DEPCV=IDET*1./(NSTAT*NGROUP)
      DEPCE=SQRT(DEPCV*(1.-DEPCV)/(NSTAT*NGROUP))
      WRITE (6,*) IDET,DEPCV,DEPCE
      STOP
      END

```

Данная программа допускает сравнительно простое обобщение на случай любого другого процесса e^+e^- взаимодействия, иной конфигурации детектора, дополнительного учёта процессов взаимодействия частиц с веществом детектора, запоминания промоделированной информации на некоем магнитном носителе в нужном формате и т. д. Высокая актуальность данной задачи привела к созданию больших стандартных пакетов программ, позволяющих моделировать практически любые процессы и конфигурации детекторов. Примерами таких пакетов являются GEANT в ЦЕРН [33], UNIMOD в ИЯФ СО РАН [34]. В этих пакетах программ имеются подпрограммы моделирования большинства процессов взаимодействия элементарных частиц с веществом, процессов распада нестабильных частиц. Современные детекторы являются сложными инженерными сооружениями. Для моделирования прохождения частиц через детектор в указанных пакетах программ имеются подпрограммы моделирования большого набора геометрических элементов: параллелепипед, конус, цилиндр и т. д., а также разнообразные сегменты этих стандартных форм. Поэтому сложную геометрию детектора, как правило, можно представить в виде набора стандартных элементов. Всё это позволяет при разработке программы моделирования нового детектора сосредоточиться на моделировании специфических элементов: нестандартных геометрических фигур, нестандартных способов регистрации пролетающих частиц и т. п.

Chapter 4

Разработка программ прохождения излучения через вещество

Разработка программ прохождения излучения через вещество является трудоёмкой задачей.

4.1 Программы моделирования столкновения частиц в установках со встречными пучками

...

4.2 Моделирование взаимодействия частиц с веществом детектора

Обработку данных с современных детекторов элементарных частиц невозможно представить без использования программ детального моделирования прохождения частиц через вещество детектора. В первую очередь это определяется высокой вероятностью изменения параметров частиц вплоть до полного поглощения или рождения новых частиц на ядрах или электронных оболочках атомов.

Формально, вообще говоря, все процессы взаимодействия частиц с атомами вещества являются «точечными», т.е. в некоторой точке траектории частицы происходит взаимодействие частицы с ядром или электроном оболочки отдельного атома. При этом может измениться направление движения частицы, энергия, из атома может вылететь нуклон ядра или электрон из оболочки, может также произойти рождение дополнительных частиц в поле ядра. Однако, некоторые процессы настолько интенсивны и одновременно эффект в них настолько мал, что моделировать эти акты взаимодействия один за другим очень нерационально. Традиционно в программах моделирования учитывают суммарный эффект влияния таких процессов на параметры частицы на некотором участке траектории. Такие процессы условно можно назвать «непрерывными».

4.2.1 Непрерывные взаимодействия

Непрерывные взаимодействия присущи заряженным частицам, у которых, с одной стороны, взаимодействия с другими заряженными частицами — не самые сильные в мире элементарных частиц, с другой стороны — дальнедействующие. Первое определяет сравнительно небольшой эффект влияния на параметры движущейся частицы в каждом отдельном акте взаимодействия, второе — большую вероятность и, следовательно, большое число таких актов на единицу траектории. И хотя в каждом акте столкновения с атомом вещества одновременно происходит и изменение направления движения начальной частицы, и изменение энергии частицы, но моделируют эти процессы отдельно — ионизационные потери и многократное рассеяние.

4.2.2 Ионизационные потери заряженных частиц

Этот процесс более точно можно назвать «потери энергии заряженной частицы на ионизацию и возбуждение атомов». Передача энергии электрону атома в каждом акте взаимодействия имеет широкий диапазон — от десятков электрон-вольт до существенной доли кинетической энергии движущейся частицы. При этом если энергия такого электрона велика, то он после своего рождения начинает жить своей жизнью — рассеиваясь на других атомах, движется по своей траектории, в свою очередь возбуждая другие атомы на своём пути. Поэтому, если мы хотим иметь высокую точность моделирования, не следует усреднять эффекты с большими передачами импульса и эффекты с малыми передачами. Общепринятая практика в этой проблеме — вводить некоторую пороговую передачу энергии η электрону атома, и процессы с меньшей передачей энергии суммировать на элементе длины траектории, а выбивание электронов с атомной оболочки с передачей, большей, чем η (δ -электроны), рассматривать как точечный процесс и моделировать индивидуально.

Для частицы с ненулевым зарядом q , массой M и импульсом P при перемещении на расстояние Δx в веществе с атомным номером Z и атомным весом A имеется некоторое распределение вероятностей по суммарной потере энергии на ионизацию и возбуждение атомов с максимальной передачей энергии η в каждом столкновении. Это распределение даже в некотором приближении может быть представлено только в виде сложного интеграла в комплексной плоскости. Кинетическое уравнение для этого распределения вероятностей было получено Л.Д.Ландау [18]. В этой же работе Л.Д.Ландау привёл красивое решение кинетического уравнения при некоторых достаточно сильных упрощениях. Основное допущение, которое не позволяет использовать решение Ландау для точного моделирования прохождения заряженных частиц через вещество — снятие ограничения на передачу энергии в каждом столкновении. Фактически, приходится использовать одну — более точную формулу для вычисления среднего значения потерь энергии на ионизацию, и какую-либо формулу для моделирования флуктуаций потерь. Средние потери энергии на ионизацию с максимальной передачей энергии η вычисляются по формуле [68]

$$-\frac{dE}{dx} = \frac{0.307075 \left[\frac{\text{MeV} \cdot \text{cm}^2}{\text{g}} \right] q^2 Z \rho}{A \cdot \beta^2} \cdot \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_m}{I^2} - \frac{T_{kin} + T_m}{2T_{kin}} \beta^2 - \frac{\delta}{2} \right], \quad (4.1)$$

где q — заряд движущейся частицы, Z, A — атомный номер и вес

Table 4.1: Расчётное значение среднего ионизационного потенциала для некоторых веществ

Вещ-во	I , эВ	Вещ-во	I , эВ	Вещ-во	I , эВ	Вещ-во	I , эВ
H	19	Be	64	C	78	N	88
Ne	130	Al	166	Ar	210	Cu	323
Br	372	Kr	381	Ag	487	Xe	555
Pb	826	U	923				

вещества, ρ — плотность вещества, $m_e c^2 \approx 0.511$ МэВ — энергия покоя электрона, $\gamma = E/M$, $\beta = P/E$. Заряды q и Z измерены в единицах заряда протона.

I — средний ионизационный потенциал, который может быть получен довольно сложными вычислениями [35], очень приближённая формула для него может быть записана в виде

$$I \sim 9.8 \text{ eV} \times Z \cdot \left[1 + \frac{6}{Z^{0.2}} \right]$$

(точность порядка 100%). Однако, величина I не является гладкой функцией параметра Z . Для некоторых веществ расчётное значение среднего ионизационного потенциала приведено в табл. 4.1.

Кинематический предел для передачи энергии налетающей частицей свободному электрону (в пренебрежении энергией связи с атомом) равен

$$T_{kin} = \frac{2m_e c^2 \beta^2 \gamma^2}{1 + 2\gamma m_e/M + (m_e/M)^2}. \quad (4.2)$$

В формуле (4.1) в качестве предельной передачи энергии T_m используется минимум из двух величин

$$T_m = \min \{T_{kin}, \eta\}.$$

И наконец, в формуле (4.1) введена важная поправка δ — поправка на эффект плотности, которая уменьшает потери энергии частицей за счёт поляризации среды. При очень больших энергиях эта поправка имеет вид

$$\frac{\delta}{2} \xrightarrow{E \rightarrow \infty} \ln(\hbar\omega_p/I) + \ln\beta\gamma - 1/2, \quad (4.3)$$

где

$$\hbar\omega_p = 28.816 [\text{эВ}] \cdot \sqrt{\frac{Z\rho \left[\frac{\text{г}}{\text{см}^3} \right]}{A}}.$$

При конечных энергиях частиц расчёт поправки на эффект плотности достаточно сложный и обычно используют некоторую аппроксимацию [36, 37, 68]:

$$\delta = \begin{cases} 0, & \xi < \xi_0 (\text{вещество — изолятор}), \\ \delta_0 \cdot 10^{2(\xi - \xi_0)}, & \xi < \xi_0 (\text{вещество — проводник}), \\ B + 4.606\xi + a \cdot (\xi_1 - \xi)^k, & \xi_0 < \xi < \xi_1, \\ B + 4.606\xi, & \xi > \xi_1, \end{cases} \quad (4.4)$$

где $\xi = 0.434 \ln \beta\gamma$, $B = 2 \ln \frac{\hbar\omega_p}{I} - 1$. Для достижения высокой точности параметры $\xi_0, \xi_1, \delta_0, k$ подбирают для каждого вещества таким образом, чтобы наилучшим образом аппроксимировать точные расчёты данного эффекта. Неплохая точность получается (лучше 2% в расчёте потерь энергии частицы) при выборе следующих параметров [37]:

$$\delta_0 = 0, \quad a = -\frac{B + 4.606\xi_0}{(\xi_1 - \xi_0)^k},$$

для плотных веществ (жидких и твёрдых):

$$\text{для } I < 100 \text{ эВ } \xi_1 = 2, \quad k = 3, \quad \xi_0 = \begin{cases} 0.2, & B > -3.681, \\ -1 - 0.326B, & B < -3.681, \end{cases}$$

$$\text{для } I \geq 100 \text{ эВ } \xi_1 = 3, \quad k = 3, \quad \xi_0 = \begin{cases} 0.2, & B > -5.215, \\ -1.5 - 0.326B, & B < -5.215, \end{cases}$$

для газов при нормальной температуре и давлении:

$$k = 3,$$

$$1) B > -12.25 : \xi_1 = 4, \quad \xi_0 = \begin{cases} 1.6, & B > -10, \\ 1.7, & -10.5 < B < -10, \\ 1.8, & -11 < B < -10.5, \\ 1.9, & -11.5 < B < -11, \\ 2.0, & -12.25 < B < -11.5, \end{cases}$$

$$2) -13.804 < B < -12.25 : \xi_1 = 5, \quad \xi_0 = 2,$$

$$3) B < -13.804 : \xi_1 = 5, \quad \xi_0 = -2.5 - 0.326B.$$

Если температура и давление газа отличаются от нормальных, то все коэффициенты при расчёте δ надо поправить, используя отношение Q_ρ плотности вещества в газе к его плотности при нормальных условиях. Обозначив поправленные значения индексом ρ , можно использовать следующие зависимости:

$$B_\rho = B + \ln Q_\rho, \quad \xi_{1,\rho} = \xi_1 - \frac{\ln Q_\rho}{4.606}, \quad \xi_{0,\rho} = \xi_0 - \frac{\ln Q_\rho}{4.606}.$$

Рассчитанные таким образом потери энергии частицы можно непосредственно использовать для моделирования прохождения заряженных частиц через вещество, и во многих случаях точность моделирования будет приемлемой. Например, расчёт электромагнитного ливня в толстых слоях вещества вполне допускает учёт потерь энергии электронов и позитронов в среднем, так как основные флуктуации энерговыделения происходят от других случайных процессов — рождения пар и тормозного излучения. Однако, для программ общего назначения важно моделировать и флуктуации ионизационных потерь на каждом элементарном перемещении заряженной частицы. Ранее упомянутое приближённое решение Ландау [18] недопустимо использовать, так как оно не учитывает введённое нами ограничение η на передачу энергии при каждом столкновении с атомом. Более того, в нём проигнорирован даже кинематический предел T_{kin} на эту передачу. Поэтому потери энергии на рождение δ -электронов будут учитываться дважды. Распределение Вавилова [38] учитывает кинематический предел T_{kin} и с этой точки зрения более правильное, чем распределение Ландау, но граничную энергию η для разделения малых и больших передач энергии не использует. Поэтому оно тоже не пригодно для точных программ моделирования. Возможны следующие варианты алгоритмов моделирования флуктуаций:

- Из кинетического уравнения Ландау легко получается формула для дисперсии распределения по потере энергии заряженной частицей:

$$\sigma_E^2 = \frac{0.153 \left[\frac{\text{МэВ} \cdot \text{см}^2}{\Gamma} \right] Z T_m \rho \Delta x}{A \beta^2} \quad (4.5)$$

Если теперь использовать любое распределение вероятностей, желательно, похожее на распределение Ландау, которое имеет среднее значение (4.1) и средне-квадратичное отклонение от среднего σ_E в (4.5), то при моделировании движения заряженной частицы

4.2. МОДЕЛИРОВАНИЕ ВЗАИМОДЕЙСТВИЯ ЧАСТИЦ С ВЕЩЕСТВОМ ДЕТЕКТОРА

на любом участке траектории частицы будет воспроизводиться правильное среднее значение и дисперсия потерь энергии, так как при суммировании независимых флуктуаций на последовательных прямолинейных участках траектории среднее значение равно сумме средних и дисперсия суммы равна сумме дисперсий. При этом форма распределения, вообще говоря, не будет правильной.

- Кинетическое уравнение Ландау можно решить при условии ограничения передач энергии на каждом столкновении. Решение записывается в виде интеграла в комплексной плоскости, которое прямо не может быть использовано при моделировании из-за большой трудоёмкости вычисления интеграла. В работе [39] эта функция была аппроксимирована кубическим сплайном во всём диапазоне изменения своих параметров. Использование коэффициентов сплайна позволяет написать эффективную программу генерации флуктуаций ионизационных потерь. В этом случае правильно воспроизводится не только среднее значение и дисперсия ионизационных потерь, но и все остальные моменты (и форма распределения). Точность аппроксимации при любых параметрах вещества и частицы лучше 0.5% в интегральной функции распределения, достигая 1% в считанных точках, точность воспроизведения среднего значения и дисперсии тоже порядка или меньше 1%.

4.2.3 Многократное рассеяние заряженных частиц

4.2.4 Точечные взаимодействия

4.2.5 Тормозное излучение

4.2.6 Рождение δ -электронов

4.2.7 Рождение электрон-позитронных пар

4.2.8 Комптон-эффект

4.2.9 Фотоэффект

4.3 Моделирование отклика детектора

4.4 Некоторые технические приемы

Построение дополнительных орт к данному единичному вектору.

Иногда требуется построить координатную систему из трех ортогональных единичных векторов, причем один из этих векторов \mathbf{e}_1 задан (как правило, это текущее значение вектора скорости частицы). Требуется добавить к этому вектору еще два единичных вектора, которые вместе составляли бы правильную декартову систему координат. Быстро справиться с такой задачей поможет следующий прием:

$$\begin{aligned} e_{2x} &= \frac{e_{1x}^2}{1+e_{1z}} - 1, & e_{2y} &= \frac{e_{1x}e_{1y}}{1+e_{1z}}, & e_{2z} &= e_{1x}, \\ e_{3x} &= \frac{e_{1x}e_{1y}}{1+e_{1z}}, & e_{3y} &= \frac{e_{1y}^2}{1+e_{1z}} - 1, & e_{3z} &= e_{1y}. \end{aligned} \quad (4.6)$$

Нетрудно убедиться, что при условии $\mathbf{e}_1^2 = 1$ два других вектора — тоже единичной длины, ортогональны между собой и с первым вектором, и смешанное произведение

$$(\mathbf{e}_1, [\mathbf{e}_2 \times \mathbf{e}_3]) = +1.$$

Может так оказаться, что величина $(1 + e_{1z})$ близка к нулю, что может привести к техническим трудностям. Тогда можно использовать другие два вектора с такими же свойствами:

$$\begin{aligned} e_{2x} &= -\frac{e_{1x}e_{1y}}{1-e_{1z}}, & e_{2y} &= 1 - \frac{e_{1y}^2}{1-e_{1z}}, & e_{2z} &= e_{1y}, \\ e_{3x} &= 1 - \frac{e_{1x}^2}{1-e_{1z}}, & e_{3y} &= -\frac{e_{1x}e_{1y}}{1-e_{1z}}, & e_{3z} &= e_{1x}. \end{aligned} \quad (4.7)$$

Chapter 5

Численные методы минимизации функций нескольких переменных

Задача определения экстремумов функции нескольких переменных является весьма актуальной в экспериментальной физике (см. раздел 6). Как правило, в реальных задачах невозможно аналитически найти точки экстремума даже для функции двух или трёх переменных, не говоря уже о десятках переменных. Поэтому возникает задача численного поиска экстремумов на ЭВМ. Мы не будем касаться своеобразных приёмов линейного программирования, т. е. определения экстремумов линейной функции с линейными условиями, но даже без этого задача представляется достаточно сложной. Имеется много учебников, специально посвящённых этому вопросу, например, [40]. Методы оптимизации функции при наличии каких-либо ограничений на параметры рассматриваются в работе [41]. Эта сложная проблема является чрезвычайно актуальной в обработке экспериментальных данных в физике высоких энергий. О живом интересе к этой проблеме говорит неиссякающий поток оригинальных работ на данную тему [42, 43, 44, 45, 46, 47]. Здесь этой проблемы мы не касаемся ввиду её сложности. Изложение в данном пособии ближе всего соответствует обзору в работе [48].

Очевидно, что задачу поиска максимума функции всегда можно свести к задаче минимизации, взяв исследуемую функцию с обратным знаком.

Имеется много программ численной минимизации, решающих эту проблему различными способами. В обработке данных физических экспериментов наиболее часто используют программу MINUIT [49], разработанную в ЦЕРНе. Может создаться неверное впечатление, что не требуется знать достоинства и недостатки различных известных схем минимизации, если есть хорошие, проверенные на практике, готовые программы. Это совсем не так. Возьмём даже такой простой вопрос, как выбор алгоритма минимизации в программе MINUIT (в ней предлагается несколько алгоритмов). Чтобы сделать такой выбор осознанно, надо представлять хотя бы в общем эти алгоритмы, их слабые места.

Кроме того, по разным причинам не всегда возможно использовать готовую программу. Например, программу MINUIT не рекомендуется использовать для задач конвейерной обработки данных, таких как реконструкция событий в трековых детекторах. Для этого лучше использовать специализированные программы, учитывающие характер минимизируемой функции. Поэтому краткий обзор методов минимизации, приводимый ниже, не представляется лишним.

Иллюстрации к методам будут всегда приводиться на примере одной и той же простой функции двух переменных

$$f(x, y) = (50 - x^2 - y^2)^2 + (x + y + 10)^2$$

с единственным очевидным минимумом в точке $x = -5$, $y = -5$. Стартовая точка для поиска минимума всегда будет $x = 2$, $y = 2$ (если алгоритм в принципе может достичь минимума из этой точки, в противном случае начальная точка будет $x = 1,5$, $y = 2$), начальный шаг (если такой требуется для алгоритма) будет равным 1. Линии уровня нарисованы в логарифмическом масштабе: первый уровень $f(x, y) = 5$, второй – $f(x, y) = 25$, третий – $f(x, y) = 125$, четвёртый – $f(x, y) = 625$.

Точки вычисления функции помечены кружочками, если значение функции в них меньше, чем в ранее проверенных точках (сюда же включена самая первая точка), причём эти точки соединены отрезками прямой в порядке появления. Остальные точки отмечаются квадратиками. Всего в каждом методе фиксируется 100 первых вычислений функции.

Кроме скорости приближения из далёкой начальной точки к точке минимума эффективность алгоритма характеризуется также скоростью приближения к минимуму, когда программа минимизации уже близка к этой точке. В тех случаях, когда будет демонстрироваться практическая работа какого-либо алгоритма, будет указываться порядковый номер $n_{0.1}$ вычисления значения функции, на котором было достигнуто отклонение

значения функции от минимального меньше, чем 0,1, и аналогичные значения $n_{0.01}$ и $n_{0.001}$.

В тех случаях, когда требуется значение градиента, он будет вычисляться аналитически, что, однако, в большинстве практических задач невозможно.

5.1 Одномерная минимизация

Задача минимизации функции $f(x)$ одной переменной является наиболее исследованной. Хотя практические задачи, как правило, требуют оптимизации нескольких параметров одновременно, алгоритм минимизации функции одной переменной часто является составной частью других, более сложных алгоритмов. Методы минимизации функции нескольких переменных можно так или иначе применять и для минимизации функции одной переменной, и здесь мы не будем их дублировать. Однако для одномерной минимизации возможен «метод золотого сечения», который в других случаях неприменим.

Задачу можно сформулировать таким образом: пусть имеется интервал $a_1 \leq x \leq a_2$, внутри которого находится локальный минимум функции $f(x)$. Требуется максимально быстро уточнить положение минимума.

Чтобы определить, что минимум находится внутри интервала, необходима одна внутренняя точка, кроме границ интервала. Для того чтобы сократить интервал, необходимо вычислить значение функции ещё в одной внутренней точке и в результате сравнения значений функции в двух внутренних точках a_3 и a_4 (рис. 5.1) решить, какую из двух внешних точек отбросить, сократив тем самым интервал локализации минимума. Для экономии времени желательно оставшиеся три точки использовать для следующего шага сокращения интервала.

Если даже сузить задачу условием симметричного выбора двух внутренних точек, задача в такой постановке не имеет однозначного решения. Известно, что если поставить задачу для максимального сокращения интервала за фиксированное число шагов, то пропорции при выборе внутренних точек должны быть переменными и основываться на числах ряда Фибоначчи. Действительно, условие симметричности расположения точек и вычисление на каждом шаге только одной точки определяет связь между длинами последовательных интервалов локализации минимума:

$$d_i = d_{i+1} + d_{i+2}. \quad (5.1)$$

Легко заметить, что это ряд Фибоначчи (в обратном порядке). Если на последнем шаге осуществить максимально возможное сокращение

$$d_n = \frac{1}{2}d_{n-1},$$

то ряд восстанавливается:

$$d_{n-2} = 3 \cdot d_n; \quad d_{n-3} = 5 \cdot d_n; \quad \dots \quad d_1 = \left[\left(\frac{1 + \sqrt{5}}{2} \right)^{n+1} - \left(\frac{1 - \sqrt{5}}{2} \right)^{n+1} \right] \cdot \frac{d_n}{\sqrt{5}}.$$

Мы не останавливаемся здесь на этом методе. Другой вариант решения, когда накладывается требование сохранения пропорций при сокращении интервала, называется методом «золотого сечения». Определим

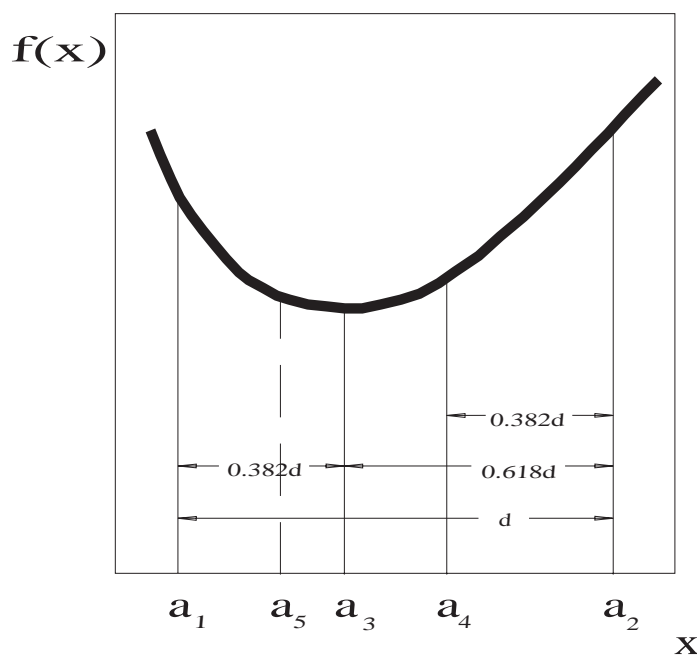


Figure 5.1: Одномерная минимизация. Метод

«золотого сечения».

Коэффициент сокращения интервала на одном шаге:

$$\xi = \frac{a_4 - a_1}{a_2 - a_1} = \frac{a_2 - a_3}{a_2 - a_1},$$

$$\frac{a_3 - a_1}{a_2 - a_1} = 1 - \xi,$$

$$\frac{1 - \xi}{\xi} = \xi,$$

$$\xi = \frac{\sqrt{5} - 1}{2} \approx 0,618.$$

коэффициент сокращения интервала на одном шаге

$$\xi = \frac{a_4 - a_1}{a_2 - a_1} = \frac{a_2 - a_3}{a_2 - a_1}.$$

Отсюда

$$\frac{a_3 - a_1}{a_2 - a_1} = 1 - \xi.$$

При сокращении интервала влево мы должны будем добавить новую точку a_5 такую, чтобы пропорции в группе точек $\{a_1, a_5, a_3, a_4\}$ остались такими же, как в группе $\{a_1, a_3, a_4, a_2\}$, т. е. $\frac{1-\xi}{\xi} = \xi$. Такое же уравнение мы получили бы, подставляя в (5.1) выражение для d_k в виде $d_k = d_0 \cdot \xi^k$. Решая квадратное уравнение, получаем $\xi = \frac{\sqrt{5}-1}{2} \approx 0,618$.

Сокращение исходного интервала в этом методе экспоненциальное. Например, в 10^6 раз интервал сокращается меньше, чем за 30 шагов. Проигрыш в эффективности по сравнению с выбором пропорций в соответствии с рядом Фибоначчи можно характеризовать отношением конечных длин интервалов для двух этих методов:

$$\frac{\xi^{n-1}}{\sqrt{5}} \cdot \left[\frac{1}{\xi^{n+1}} - (-\xi)^{n+1} \right] \xrightarrow{n \rightarrow \infty} \frac{1}{\xi^2 \sqrt{5}} \approx 1,17.$$

Видно, что при большом сокращении исходного интервала эффективность этих методов сравнима, хотя и метод «золотого сечения» всегда несколько проигрывает.

5.2 Метод сеток

Этот метод самый простой в применении, но крайне неэффективен при большом числе аргументов функции. Алгоритм заключается в том, чтобы покрыть весь исследуемый фазовый объём в пространстве n переменных регулярной сеткой с постоянным шагом по каждой переменной, вычислить значения функции в узлах сетки и выбрать среди них минимальное.

5.3 Случайный поиск

Обычно под случайным поиском подразумевается следующий алгоритм.

- Генерируется заданное количество случайных точек N в n -мерном пространстве. Обычно используется равномерное распределение вероятностей по каждой координате в интервале $(x_i - h_i, x_i + h_i)$, где x_i — i -я координата начальной точки, h_i — значение начального шага поиска по i -й координате.
- Среди этих N точек выбирается точка с минимальным значением функции. Пусть её координаты равны y_i .

- Эффективность алгоритма будет меньше зависеть от неоптимального выбора шагов поиска h_i , если будет предусмотрена некоторая процедура настройки шага, например, $h_i = h_i \cdot (0.5 + \frac{|y_i - x_i|}{h_i})$, причём если не было найдено точки со значением функции меньшим, чем в исходной точке, то полагается $y_i = x_i$.
- Координаты исходной точки x_i заменяются координатами y_i , и всё повторяется сначала.

На рис. 5.2 представлен ход минимизации при случайном поиске, выполненном в соответствии с представленным алгоритмом. Для сравнения на рис. 5.3 показан ход минимизации в программе MINUIT, написанной в ЦЕРНе [49], где для выбора случайных пробных точек используется более сложный алгоритм Метрополиса. Лучшие из методов минимизации функций, обсуждаемые в данной работе, реализованы в этой программе и могут использоваться или по выбору пользователя, или на усмотрение головной программы MINUIT.

5.4 Метод последовательной минимизации

На первый взгляд кажется привлекательным метод поочерёдной однокоординатной минимизации, в котором, отправляясь от начальной точки, находим минимум по первой координате, оставляя прочие постоянными, затем, зафиксировав первую координату, находим минимум по второй координате и т.д. После завершения цикла минимизации по всем координатам, как правило, придётся повторить его ещё не один раз до достижения минимума, что демонстрируется рис. 5.4. Более того, легко подобрать минимизируемую функцию и соответствующую начальную точку так, что процесс поиска вообще не сойдётся к минимуму. Рисунок демонстрирует как раз такой случай. Несколько лучше выглядит модифицированный алгоритм однокоординатной минимизации, описанный в разд. 5.5, однако и он не гарантирует сходимости к минимуму.

5.5 Метод Розенброка

В методе Розенброка [50] цикл однокоординатной минимизации по всем координатам выполняется так же, как в разд. 5.4, а затем координатные оси переопределяются так, чтобы одна из осей имела направление из начальной

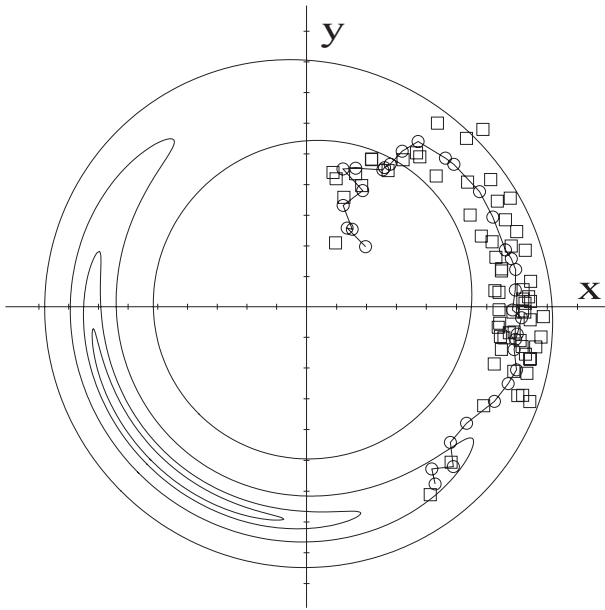


Figure 5.2: Метод случайного поиска при равномерном распределении пробных точек внутри гиперкуба. $n_{0.1} = 356$, $n_{0.01} = 873$, $n_{0.001} = 1772$, $\min_{i=1,100} f(x_i, y_i) = f_{99}(4, 31; -5, 89) = 81, 18$

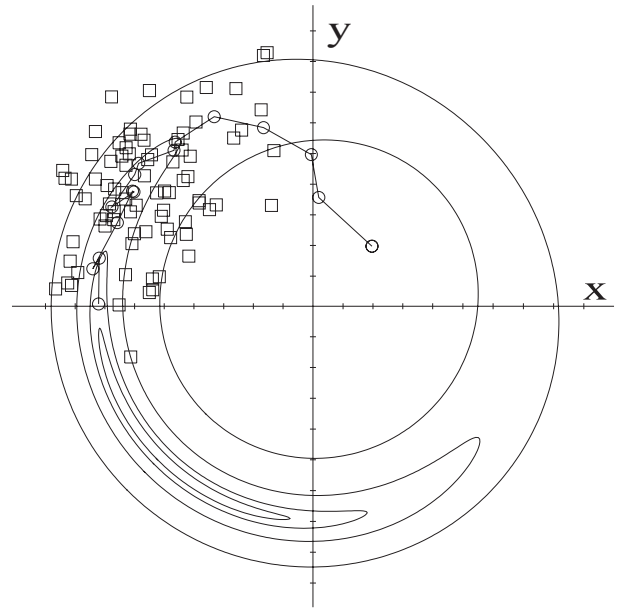


Figure 5.3: Метод случайного поиска с выбором точек по алгоритму Метрополиса. $n_{0.1} = 463$, $n_{0.01} = 704$, $n_{0.001} = 704$, $\min_{i=1,100} f(x_i, y_i) = f_{97}(-7, 08; 0, 08) = 9, 03$

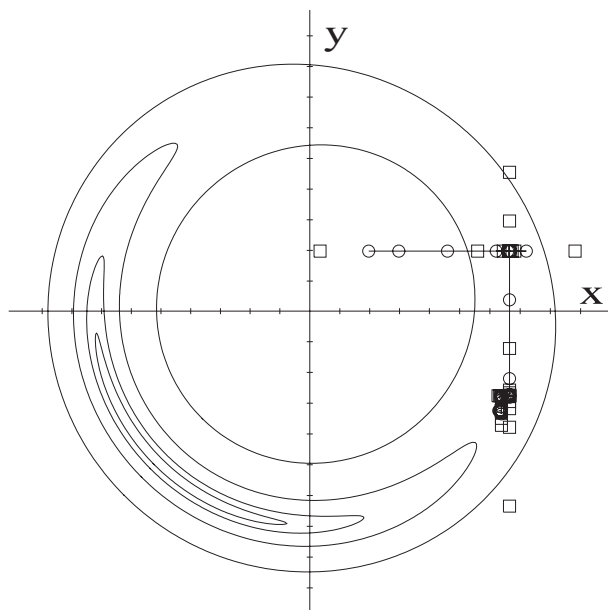


Figure 5.4: Метод последовательной минимизации. $n_{0.1}, n_{0.01}, n_{0.001} > 2 \cdot 10^6$, $\min_{i=1,100} f(x_i, y_i) = f_{100}(6, 318; -3, 303) = 170, 07$

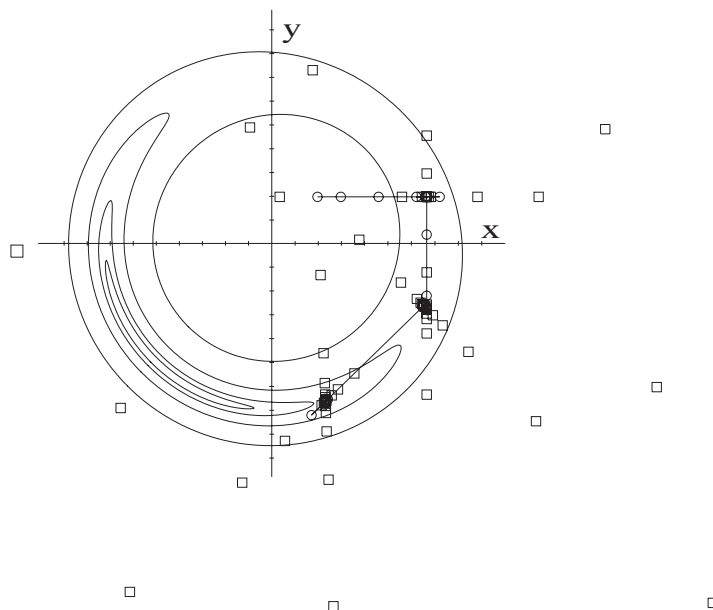


Figure 5.5: Метод Розенброка. $n_{0.1} = 164, n_{0.01} = 204, n_{0.001} = 244$, $\min_{i=1,100} f(x_i, y_i) = f_{96}(2, 338; -6, 703) = 31, 91$

точки в точку, достигнутой в результате цикла минимизации (ожидая, что это есть направление вдоль некоторого «ущелья», ведущего к минимуму). Несмотря на большое усложнение алгоритма, в большинстве случаев это не даёт существенного выигрыша (рис. 5.5).

5.6 Метод проб и ошибок

Явным недостатком методов последовательной минимизации (покоординатного спуска) и Розенброка является требование локализовать минимум вдоль некоторого направления, которое проходит далеко от искомого минимума. В методе Розенброка имеются также значительные накладные расходы по построению вспомогательной системы координат. Рассматриваемый в данном разделе метод проб и ошибок не требует времени на излишнее уточнение вдоль какого-либо направления, а использует информацию о поведении функции сразу же после пробного шага. Алгоритм минимизации может выглядеть следующим образом.

1. Перед началом минимизации заданы координаты x_i исходной точки и

начальные шаги h_i по каждой координате.

2. Начало цикла по всем координатам.

- (a) Для очередной координаты i делается шаг h_i и вычисляется значение функции $f(x_i + h_i)$.
- (b) Если значение функции в новой точке меньше, чем в исходной ($f(x_i + h_i) < f(x_i)$), то координата x_i заменяется на $x_i + h_i$, увеличивается шаг $h_i = h_i \alpha$ и делается возврат в пункт 2а.
- (c) В целях более успешного прохождения узких «ущелий» можно вычислять пробные значения функции при парных смещениях $f(x_i + h_i, x_j \pm h_j)$ в сочетании i -й координаты со всеми $j \neq i$. Начало вложенного цикла по $j \neq i$.

I. Начало вложенного цикла на два повторения $k = -1; 1$.

- i. Вычисляется значение $f(x_i + h_i, x_j + k \cdot h_j)$.
- ii. Если $f(x_i + h_i, x_j + k \cdot h_j) < f(x_i, x_j)$, то обе координаты x_i, x_j заменяются на координаты успешной точки, шаги h_i, h_j по этим координатам увеличиваются в α раз и пункт 2(c)Ii повторяется с этими же i, j, k .

II. Конец цикла по k .

- (d) Конец цикла по j
- (e) Шаг по i -й координате уменьшается: $h_i = -\beta h_i$, и всё повторяется для следующей координаты.

3. Конец цикла по i .

4. Если критерий достижения минимума не выполняется, то минимизация продолжается с пункта 2.

Работа этого алгоритма с параметром увеличения шага $\alpha = 3$ и параметром уменьшения шага $\beta = 0,4$ демонстрируется на рис. 5.6.

Видно, что алгоритм позволяет достаточно быстро прийти в область минимума, но затем медленно уточняет его положение. Неоспоримым достоинством этого алгоритма является простота реализации и надёжная сходимость, т.е. рано или поздно этот алгоритм приведёт в точку минимума.

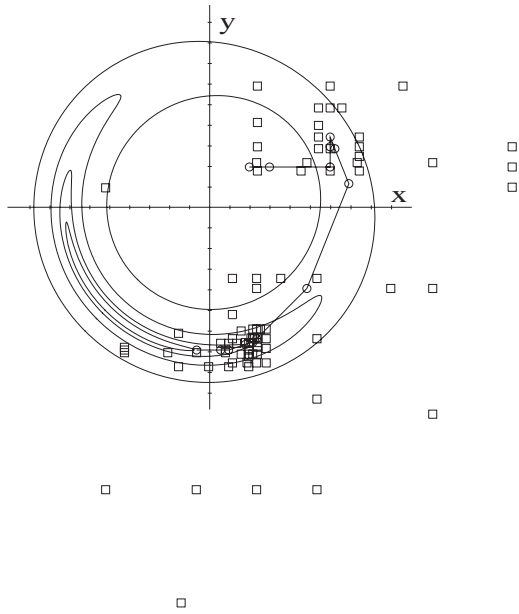


Figure 5.6: Метод проб и ошибок.
 $n_{0.1} = 308$, $n_{0.01} = 1578$, $n_{0.001} = 1689$,
 $\min_{i=1,100} f(x_i, y_i) = f_{93}(-0,609; -7,054) = 5,478$

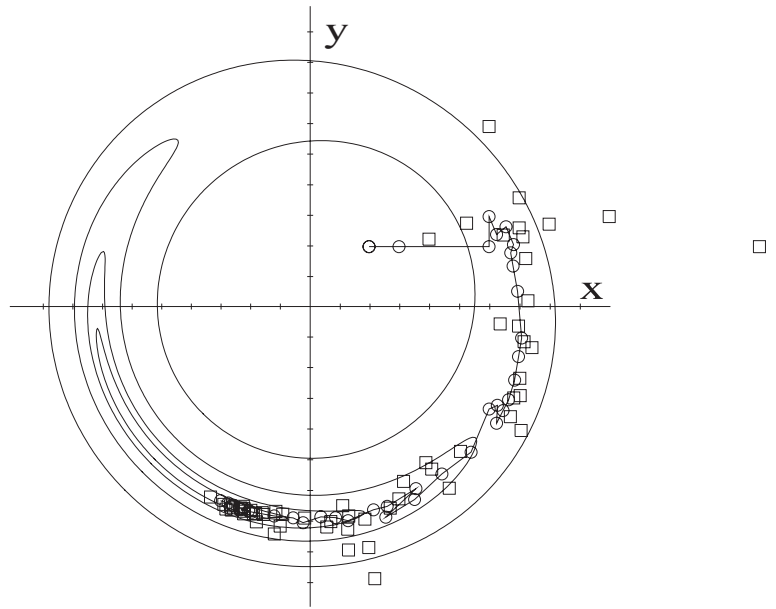


Figure 5.7: Симплекс-метод (программа MINUIT). $n_{0.1} = 112$,
 $n_{0.01} = 132$, $n_{0.001} = 145$,
 $\min_{i=1,100} f(x_i, y_i) = f_{99}(-2,94; -6,46) = 0,53$

5.7 Симплекс-метод

Одним из наиболее мощных пошаговых методов в n -мерном пространстве является симплекс-метод [51]. Он заключается в преобразовании набора $(n + 1)$ точек в пространстве путём замены самой «худшей» точки на более перспективную.

1. Выбираются начальные координаты $(n + 1)$ точек (например, случайным образом или поочерёдным сдвигом по каждой координате). Обозначим набор n координат i -й точки вектором \mathbf{x}_i , а значение функции в ней — f_i .
2. Среди всех точек определяется точка $i = h$ с наибольшим значением функции и точка $i = l$ с наименьшим значением функции.
3. По всем точкам, кроме $i = h$, вычисляется вектор \mathbf{x}_a :

$$\mathbf{x}_a = \frac{1}{n} \sum_{i \neq h} \mathbf{x}_i.$$

4. Вычисляются значения функции в одной или более точках, лежащих на прямой, проходящей через точки \mathbf{x}_a и \mathbf{x}_h .

(а) Вычисляется значение функции f_o в точке $\mathbf{x}_o = 2\mathbf{x}_a - \mathbf{x}_h$.

(б) Если $f_o < f_l$, то вычисляется значение функции f_s в точке

$$\mathbf{x}_s = 3\mathbf{x}_a - 2\mathbf{x}_h.$$

(с) Если $f_o > f_h$, то вычисляется значение функции f_t в точке

$$\mathbf{x}_t = (\mathbf{x}_a + \mathbf{x}_h)/2.$$

По этим трём значениям функции f_h , f_o и f_t проводится парабола, определяется предполагаемая точка минимума \mathbf{x}_p и в ней тоже вычисляется значение функции f_p .

5. Если f_o , f_t и f_p превышают f_h , то делается переход в пункт 1, образуется новый симплекс около точки \mathbf{x}_l , и всё повторяется сначала.

6. Из новых точек (от одной f_o до максимум трёх f_o , f_t и f_p) выбирается точка с наименьшим значением функции, и точка \mathbf{x}_h заменяется на эту точку, образуя новый симплекс. Если критерий достижения минимума не выполняется, то всё повторяется с пункта 2.

Рисунок 5.7 демонстрирует ход минимизации для симплекс-метода, реализованного в программе MINUIT [49].

5.8 Метод наискорейшего спуска

В тех случаях, когда для исследуемой функции возможно аналитическое вычисление градиента, поиск минимума может быть значительно ускорен. В методе наискорейшего спуска в направлении, определяемом вектором градиента, проводится одномерная минимизация, в полученной точке снова вычисляется градиент, и поиск минимума продолжается в новом направлении. Рис. 5.8 демонстрирует работу данного алгоритма на примере нашей тестовой функции.

Как видим, пробная функция оказалась очень неудобной для обсуждаемого метода. Во-первых, если начать минимизацию в соответствии с данным алгоритмом из стандартной начальной точки $\mathbf{x} =$

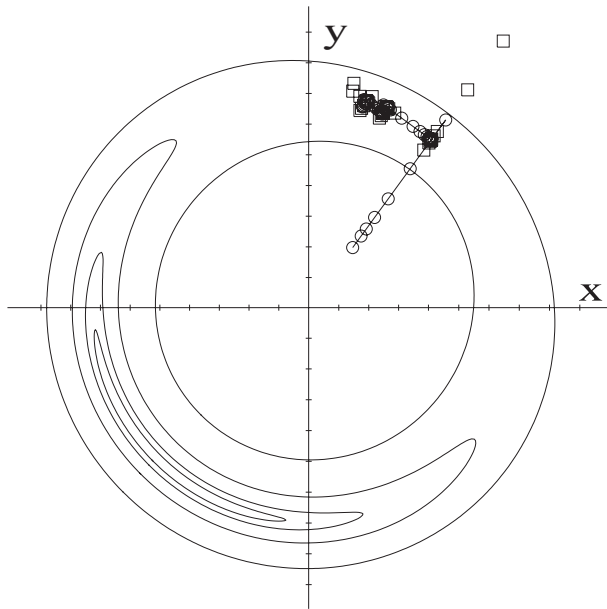


Figure 5.8: Метод наискорейшего спуска. $n_{0.1} = 6627$, $n_{0.01} = 15082$, $n_{0.001} = 42310$, $\min_{i=1,100} f(x_i, y_i) = f_{100}(1, 839; 6, 683) = 346, 89$

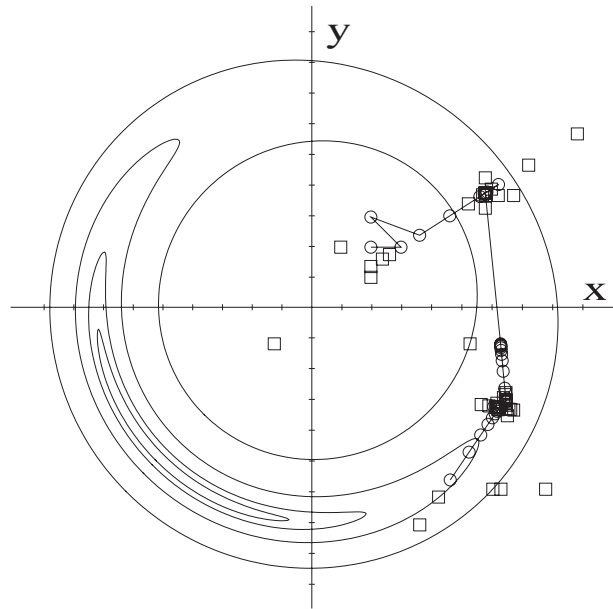


Figure 5.9: Метод Ньютона. $n_{0.1} = 1144$, $n_{0.01} = 3499$, $n_{0.001} = 8389$, $\min_{i=1,100} f(x_i, y_i) = f_{98}(4, 634; -5, 727) = 97, 591$

(2; 2), то сразу же произойдёт спуск в седловую точку в области $\mathbf{x} = (5; 5)$, и в рамках данного алгоритма невозможно из неё выйти. Поэтому минимизация была начата из точки $\mathbf{x} = (1, 5; 2)$. Однако и в этом случае, как видно из рисунка, результаты неудовлетворительные. В процессе поиска минимума происходит зигзагообразное движение вдоль дна «ущелья» с очень мелким шагом.

5.9 Метод Ньютона

Одной из перспективных идей в поисках эффективного алгоритма минимизации является использование разложения минимизируемой функции в районе исходной точки \mathbf{x}_0 в виде квадратичной формы, т. е.

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \mathbf{g}_0^T \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{M}_0(\mathbf{x} - \mathbf{x}_0), \quad (5.2)$$

где $f(\mathbf{x}_0)$, \mathbf{g}_0 , \mathbf{M}_0 – значения минимизируемой функции, вектора градиента и симметричной матрицы вторых производных в начальной точке $\mathbf{x} =$

\mathbf{x}_0 . Верхним индексом T здесь и далее обозначаются транспонированные векторы и матрицы.

Если бы функция $f(\mathbf{x})$ действительно представляла собой положительно определённую квадратичную форму, то мы сразу же могли бы вычислить точку минимума

$$\mathbf{x}_{min} = \mathbf{x}_0 - \mathbf{M}^{-1}\mathbf{g} = \mathbf{x}_0 - \mathbf{V}\mathbf{g} \quad (5.3)$$

и значение функции в минимуме

$$f_{min} = f(\mathbf{x}_0) - \frac{1}{2}\mathbf{g}^T\mathbf{V}\mathbf{g}, \quad (5.4)$$

где введено новое обозначение $\mathbf{V} = \mathbf{M}^{-1}$ для ковариационной матрицы – обратной матрицы к матрице вторых производных. Эта матрица названа так же, как матрица в разд. 3.4, описывающая корреляцию случайных величин, по следующим соображениям. Как будет видно в разд. 6.1, наиболее эффективный метод оценки параметров модели связан с минимизацией функции $\{-\ln P(\mathbf{x})\}$, где $P(\mathbf{x})$ – полная плотность вероятности зарегистрировать полученные экспериментальные данные при параметрах модели \mathbf{x} . При этом ковариационная матрица в минимуме функции описывает флуктуации искомых параметров в гауссовом приближении (с учётом линейной корреляции) при многократном повторении этого эксперимента.

Для практического применения этого алгоритма необходимо решить, что делать в случае, когда матрица квадратичной формы отрицательно определённая, или матрица положительно определённая, но значение функции в ожидаемом минимуме превышает значение функции в исходной точке, или, наконец, определитель матрицы равен нулю, и обратную матрицу вообще невозможно найти.

Возможный алгоритм минимизации по методу Ньютона можно представить следующим образом (строго говоря, такое название метода более подходит к одномерному случаю, но данный алгоритм является естественным обобщением на многомерный случай с использованием матричной алгебры).

- В начальной точке вычисляется градиент и матрица вторых частных производных.
- Если определитель матрицы ненулевой, то вычисляется обратная матрица, затем вычисляется точка предполагаемого минимума \mathbf{x}_{min}

и в ней вычисляется функция $f(\mathbf{x}_{min})$. Если значение функции меньше, чем в исходной точке, то делается переход в эту точку, и всё повторяется сначала.

- Если значение функции в предполагаемой точке минимума больше, чем в начальной точке, то производится одномерная минимизация вдоль направления из начальной точки в точку \mathbf{x}_{min} , и всё повторяется сначала.
- Если определитель матрицы нулевой, то производится одномерная минимизация вдоль направления градиента из начальной точки, найденная точка минимума принимается за начальную, и всё повторяется сначала.

На рис. 5.9 демонстрируется ход минимизации по одной из возможных реализаций приведённого алгоритма. За рамками нашего анализа осталась процедура численной оценки матрицы вторых производных, ошибки вычисления которой могут существенно повлиять на скорость приближения к минимуму (напомним, что градиент в наших тестах вычисляется аналитически и одновременно со значением функции).

Устойчивость данного алгоритма невелика, на практике приходится минимизировать функции гораздо более сложные, и тогда этот алгоритм совершенно не сравним, скажем, с симплекс-методом или методом проб и ошибок. Кроме того, здесь нужно находить обратную матрицу к матрице вторых производных, что отнимает значительное процессорное время, а вдали от минимума не имеет никакого смысла, так как функция плохо описывается квадратичной формой.

5.10 Метод сопряжённых направлений

Для того чтобы избежать вычисления матрицы вторых производных и её обращения, можно попытаться использовать свойства квадратичной формы, не находя её представления в виде матрицы. Дело в том, что в пространстве n параметров функции, если бы она была положительно определённой квадратичной формой, можно было бы найти систему векторов \mathbf{d}_i , определяющих сопряжённые направления:

$$\mathbf{d}_i^T \mathbf{M} \mathbf{d}_j = 0 \quad \text{для} \quad i \neq j. \quad (5.5)$$

Такая система векторов не определена однозначно, но может быть легко построена, если стартовать с какого-нибудь произвольного вектора, который определит первое из этих направлений. Далее для истинной квадратичной формы достаточно было бы в целях нахождения минимума из любой начальной точки провести последовательно n одномерных минимизаций вдоль сопряжённых направлений.

Пусть у нас имеется k векторов \mathbf{d}_i , задающих сопряжённые направления для данной квадратичной формы, и пусть точка \mathbf{x}_0 является минимумом функции, найденным путём последовательной однокоординатной минимизации по этим сопряжённым направлениям, начиная с некоторой начальной точки. Теперь можно построить $(k + 1)$ -й вектор, задающий направление, сопряжённое со всеми предыдущими.

Сместимся из точки \mathbf{x}_0 на вектор $\Delta\mathbf{x}$, который не может быть представлен в виде линейной комбинации k векторов $\mathbf{d}_i, i = 1, \dots, k$. Теперь из этой начальной точки k раз проведём одномерную минимизацию вдоль имеющихся сопряжённых направлений. Пусть минимум функции при этом найден в точке \mathbf{x}_1 . Тогда вектор $\mathbf{d}_{k+1} = \mathbf{x}_1 - \mathbf{x}_0$ определяет новое сопряжённое направление. Теперь осталось провести одномерную минимизацию вдоль этого направления, и мы придём к состоянию, которое было вначале, только уже с $k + 1$ вектором. Когда максимально возможное количество n сопряжённых направлений будет найдено, то для истинной квадратичной формы будет достигнута точка минимума, а для функции общего вида надо будет продолжать всю эту процедуру, пока минимум не будет достигнут.

На рис. 5.10 представлен ход минимизации по этому алгоритму для пробной функции. В нашем случае двух переменных основной недостаток этого метода не очень заметен, но при большем числе переменных потери времени при минимизации в $n - 1$ ненужных нам «поперечных» направлениях для того, чтобы найти n -е сопряжённое направление, станут обременительными.

Производит впечатление скорость сходимости к минимуму данного алгоритма в окрестности точки минимума, где функция уже хорошо аппроксимируется квадратичной формой.

Недавно этот метод был существенно улучшен [52], приобретая черты методов сопряжённых градиентов и переменной метрики пространства (см. разд. 5.11 и 5.12). По эффективности этот метод сравним с методом переменной метрики пространства. По сравнению с первоначальным вариантом метода сопряжённых направлений в новом варианте не

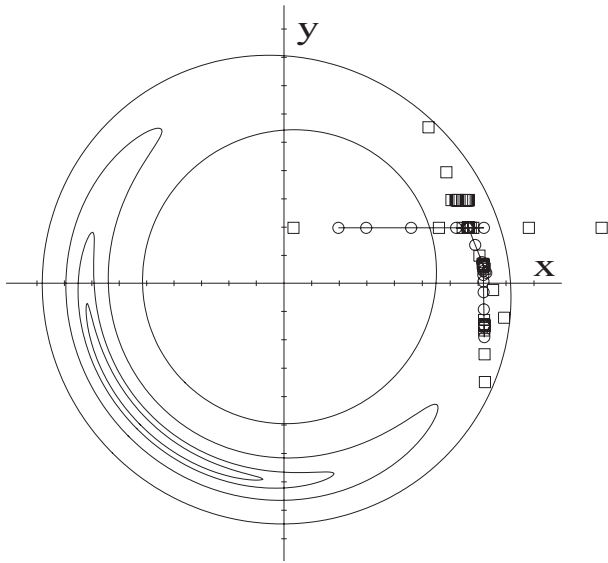


Figure 5.10: Метод сопряжённых направлений. $n_{0.1} = 306$,
 $n_{0.01} = 308$, $n_{0.001} = 360$,
 $\min_{i=1,100} f(x_i, y_i) = f_{100}(7, 241;$
 $-1, 510) = 269, 596$

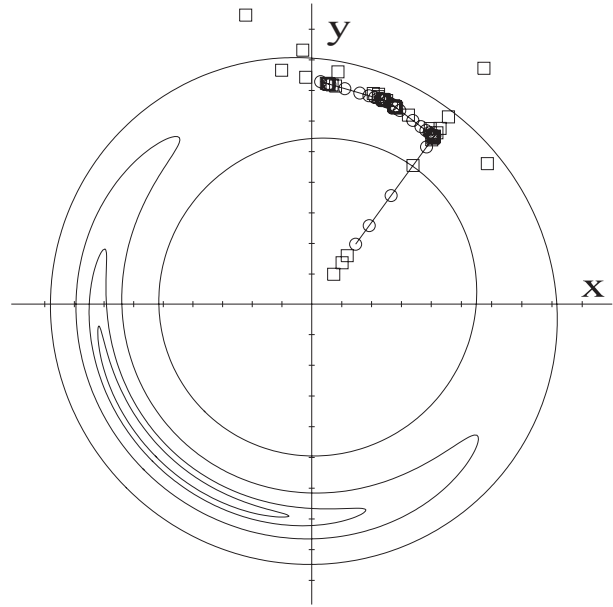


Figure 5.11: Метод сопряжённых градиентов. $n_{0.1} = 358$,
 $n_{0.01} = 388$, $n_{0.001} = 427$,
 $\min_{i=1,100} f(x_i, y_i) = f_{97}(0, 599;$
 $7, 333) = 338, 565$

требуется линейная минимизация вдоль каждого из сопряжённых направлений.

Как и в методе переменной метрики, все вычисления основаны на локальном представлении функции в виде квадратичной формы.

$$F(\mathbf{X}) = F_0 + \mathbf{B}^T \mathbf{X} + \frac{1}{2} \mathbf{X}^T \mathbf{A} \mathbf{X}. \quad (5.6)$$

В процессе минимизация происходит построение двух систем векторов $\mathbf{p}_i, \mathbf{e}_i, i = 1, 2, \dots, n$. Вектора \mathbf{e}_i ортогональны векторам \mathbf{p}_i :

$$\mathbf{e}_i^T \mathbf{p}_j = |\mathbf{p}_i| \cdot \delta_{ij} \cdot \gamma_i,$$

кроме того,

$$\mathbf{e}_i = \mathbf{A} \mathbf{p}_i. \quad (5.7)$$

Теперь, если мы разложим перемещение из начальной точки по векторам \mathbf{p}_i :

$$\Delta \mathbf{X} = \sum_{i=1}^n \alpha_i \frac{\mathbf{p}_i}{|\mathbf{p}_i|},$$

то получим значение функции

$$\begin{aligned} F(\mathbf{X} + \Delta\mathbf{X}) &= F(\mathbf{X}) + \mathbf{B}^T \Delta\mathbf{X} + \mathbf{X}^T \mathbf{A} \Delta\mathbf{X} + \frac{1}{2} \Delta\mathbf{X}^T \mathbf{A} \Delta\mathbf{X} = \\ &= F(\mathbf{X}) + \sum_{i=1}^n \left[\mathbf{B}^T \mathbf{p}_i \frac{\alpha_i}{|\mathbf{p}_i|} + \mathbf{X}^T \mathbf{A} \mathbf{p}_i \frac{\alpha_i}{|\mathbf{p}_i|} + \frac{\alpha_i^2 \gamma_i}{2|\mathbf{p}_i|} \right]. \end{aligned}$$

Видно, что переменные α_i разделяются, и минимизировать можно по каждой из них отдельно. Удобно определять значения α_i , приводящие к минимуму, используя условие равенства нулю градиента.

$$\mathbf{g}(\mathbf{X} + \Delta\mathbf{X}) = \mathbf{g}(\mathbf{X}) + \mathbf{A} \Delta\mathbf{X} = \mathbf{g}(\mathbf{X}) + \sum_{i=1}^n \frac{\alpha_i}{|\mathbf{p}_i|} \mathbf{e}_i = 0,$$

что даёт

$$\alpha_i = -\frac{1}{\gamma_i} \mathbf{p}_i^T \mathbf{g}(\mathbf{X}).$$

Свойства матрицы \mathbf{A} можно определять, используя соотношение, что изменение градиента на перемещении $\Delta\mathbf{X}$ связано с матрицей \mathbf{A} следующим образом

$$\Delta\mathbf{g} = \mathbf{A} \Delta\mathbf{X}$$

и для квадратичной формы не зависит от самой точки \mathbf{X} . Каждое очередное перемещение в n -мерном пространстве формируется как линейная комбинация векторов \mathbf{p}_i :

$$\Delta\mathbf{X} = \sum_{i=1}^n \alpha_i \frac{\mathbf{p}_i}{|\mathbf{p}_i|}.$$

Введём верхние индексы как обозначение номера шага, тогда получаем

$$\Delta\mathbf{g}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} = \sum_{i=1}^n \frac{\alpha_i}{|\mathbf{p}_i|} \mathbf{A} \mathbf{p}_i = \sum_{i=1}^n \frac{\alpha_i}{|\mathbf{p}_i|} \mathbf{e}_i.$$

Отсюда

$$\gamma_i = \frac{1}{\alpha_i} \mathbf{p}_i^T \Delta\mathbf{g}^{(k)}.$$

Теперь можно предложить последовательность шагов, которая позволяет построить систему векторов и одновременно эффективно спускаться к минимуму, причём для функции — истинной квадратичной формы,— за $(n+1)$ шагов прийти точно в точку минимума. Пусть при подготовке k -ого

шага в точке \mathbf{X}_k вычисленное значение градиента равно $\mathbf{g}^{(k)}$ и уже имеются вектора $\mathbf{p}_i, \mathbf{e}_i, i = 1, 2, \dots, k-1$. Выберем новый вектор

$$\mathbf{p}_k = -\mathbf{g}^{(k)} + \sum_{i=1}^{k-1} \beta_i^{(k)} \mathbf{p}_i,$$

$$\beta_i^{(k)} = \frac{1}{\gamma_i |\mathbf{p}_i|} \mathbf{e}_i^T \mathbf{g}^{(k)}, \quad \gamma_i = \frac{1}{|\mathbf{p}_i|} \mathbf{e}_i^T \mathbf{p}_i.$$

Здесь γ_i выражают именно скалярное произведение векторов. Такой выбор коэффициентов β_i обеспечивает ортогональность \mathbf{p}_i и \mathbf{e}_j при $i \neq j$. Делаем новый шаг

$$\Delta \mathbf{X}_k = \sum_{i=1}^n \alpha_i^{(k)} \frac{\mathbf{p}_i}{|\mathbf{p}_i|}, \quad \alpha_i^{(k)} = -C_i^{(k)} \frac{\mathbf{p}_i^T \mathbf{g}^{(k)}}{|\mathbf{p}_i|},$$

где константы $C_i^{(k)}$ вначале заполнены произвольными числами, например,

$$C_k^{(k)} = 0.1 \cdot \left| \frac{\mathbf{p}_i^T \mathbf{g}^{(k)}}{|\mathbf{p}_i|} \right|^{-1}.$$

После этого делается переход в новую точку и в ней вычисляется новое значение градиента $\mathbf{g}^{(k+1)}$ и вектор изменения градиента

$$\Delta \mathbf{g}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}.$$

Это даёт возможность вычислить сопряжённый вектор

$$\mathbf{e}_k = \frac{|\mathbf{p}_k|}{\alpha_k^{(k)}} \cdot \left(\Delta \mathbf{g}^{(k)} - \sum_{i=1}^{k-1} \alpha_i^{(k)} \frac{\mathbf{e}_i}{|\mathbf{p}_i|} \right)$$

и правильное значение констант

$$C_i^{(k+1)} = \frac{\alpha_i |\mathbf{p}_i|}{\mathbf{p}_i^T \Delta \mathbf{g}^{(k)}}, \quad i = 1, 2, \dots, k$$

Последовательно выполнив n шагов, мы построим все пары сопряжённых векторов и следующим шагом попадём точно в минимум, если минимизируемая функция — квадратичная форма. Однако, для большинства функций на практике минимум не будет достигнут. Надо делать новые шаги, заменяя при этом какие-то из старых векторов. Проблема выбора «жертвы» не такая простая. В работе [52] предлагается следующий критерий, можно ли заменять вектор \mathbf{p}_k на новый \mathbf{p}_k^* :

- Вычисляется новый вектор \mathbf{p}_k^* , используя значение градиента и вектора, кроме k -ого.
- Если $\mathbf{e}_k^T \mathbf{p}_k^* > \mathbf{e}_k^T \mathbf{p}_k$, замена производится, иначе делается попытка заменить другой вектор.

Эта процедура достаточно трудоёмкая, и нет гарантии, что можно будет заменить какой-нибудь вектор.

Другой важный вопрос — когда прекращать минимизацию. Когда уже сформированы все n векторов, то последний шаг предполагает переход точно в минимум, причём изменение функции на этом шаге известно:

$$\Delta F_k = - \sum_{i=1}^n \frac{C_i (\mathbf{p}_i^T \mathbf{g}^{(k)})^2}{2 |\mathbf{p}_i|^2} \cdot \left| 2 - \frac{\gamma_i C_i}{|\mathbf{p}_i|} \right|.$$

Если реальное изменение функции отличается от этой оценки больше, чем на $\frac{1}{2} \Delta F_m$, минимум не достигнут. Если предсказание было удачным, то ещё проверяется, не слишком ли велико предсказание для следующего изменения функции. Если

$$\sum_{i=1}^n \frac{C_i (\mathbf{p}_i^T \mathbf{g}^{(k+1)})^2}{2 |\mathbf{p}_i|^2} < \frac{1}{10} \Delta F_m,$$

то минимум считается достигнутым. Дополнительно можно проверять, насколько самосогласованы C_i и γ_i . Для квадратичной формы должно выполняться соотношение $\gamma_i C_i = |\mathbf{p}_i|$. Чтобы проверять его в сравнении с требуемой точностью ΔF_m , можно потребовать

$$\sum_{i=1}^n \frac{C_i (\mathbf{p}_i^T \mathbf{g}^{(k)})^2}{2 |\mathbf{p}_i|^2} \cdot \left| 1 - \frac{\gamma_i C_i}{|\mathbf{p}_i|} \right| < \frac{1}{10} \Delta F_m.$$

5.11 Метод сопряжённых градиентов

В работе [53] предложен метод построения нового сопряжённого направления без вспомогательных минимизаций в «поперечных» направлениях. Пусть у нас имеется новое направление, задаваемое вектором \mathbf{d}_1 , которое является сопряжённым ко всем ранее определённым сопряжённым направлениям (по всем этим направлениям была проведена одномерная минимизация). Пусть в первый момент, когда мы определили

направление \mathbf{d}_1 , мы находились в точке \mathbf{x}_1 и градиент в этой точке был равен \mathbf{g}_1 , затем мы провели одномерную минимизацию вдоль этого направления и пришли в точку \mathbf{x}_2 , где градиент равен \mathbf{g}_2 . Спрашивается, можно ли составить новое сопряжённое направление из имеющихся векторов?

Легко проверить, что любое направление, ортогональное к

$$\Delta \mathbf{g} = \mathbf{g}_2 - \mathbf{g}_1,$$

является направлением, сопряжённым с \mathbf{d}_1 . Выберем новый вектор направления в виде $\mathbf{d}_2 = -\mathbf{g}_2 + \beta \mathbf{d}_1$ и подберём β таким образом, чтобы $(\mathbf{d}_2 \Delta \mathbf{g}) = 0$. Используя тот факт, что $(\mathbf{d}_1 \mathbf{g}_2) = 0$, получаем $\beta = -\frac{(\mathbf{g}_2 \Delta \mathbf{g})}{(\mathbf{g}_1 \Delta \mathbf{g})}$. Рис. 5.11 демонстрирует ход минимизации по этому алгоритму. Естественно, нельзя было начинать минимизацию из точки $(2; 2)$, так как в рамках данного алгоритма невозможно было бы сойти с оси симметрии, проходящей через седловую точку, поэтому была взята начальная точка $(1, 5; 2)$.

5.12 Метод переменной метрики пространства

Наиболее мощным методом, позволяющим не прибегать к одномерной минимизации вдоль выбранных направлений и не вычислять точно матрицу вторых производных, а также не обращать её, является метод переменной метрики пространства. Общая идея этого метода состоит в том, что матрица \mathbf{M} (см. разд. 5.9) рассматривается как метрический тензор пространства переменных, по которым надо минимизировать функцию. Если функция представляет собой квадратичную форму, то этот тензор постоянный, и пространство можно считать пространством с постоянной метрикой. В общем случае матрица вторых производных зависит от точки, в которой её вычисляют, поэтому можно говорить о пространстве с переменной метрикой.

Общая схема алгоритма такова:

1. В начальной точке вычисляется градиент \mathbf{g}_0 и формируется некоторое начальное состояние матрицы \mathbf{V}_0 (обозначения такие же, как в 5.9), например, равное единичной матрице.

2. Вычисляется функция в точке предполагаемого минимума

$$\mathbf{x}_t = \mathbf{x}_0 - \mathbf{V}_0 \mathbf{g}_0. \quad (5.8)$$

Если $f(\mathbf{x}_t) < f(\mathbf{x}_0)$, то точка $\mathbf{x}_1 = \mathbf{x}_t$ принимается, в противном случае проводится одномерная минимизация в направлении вектора $-\mathbf{V}_0 \mathbf{g}_0$, и в качестве следующей опорной точки \mathbf{x}_1 принимается точка минимума функции на этой прямой.

3. Вычисляется смещение в направлении новой точки:

$$\Delta \mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0.$$

4. В новой точке вычисляется градиент \mathbf{g}_1 и делается преобразование матрицы

$$\mathbf{V}_1 = \Psi(\mathbf{g}_1, \mathbf{V}_0, \mathbf{g}_0, \Delta \mathbf{x}). \quad (5.9)$$

После этого всё повторяется с п. 2.

Различные варианты метода переменной метрики пространства различаются в основном правилом преобразования ковариационной матрицы \mathbf{V}_0 . Желательно, чтобы в процессе минимизации истинной квадратичной формы восстанавливалось реальное значение ковариационной матрицы, а в случае функции общего вида алгоритм был устойчив, а матрица отслеживала локальное значение ковариационной матрицы.

Один из первых способов преобразования ковариационной матрицы был предложен Давидоном и позже в упрощённой форме был опубликован в работе [54]:

$$\mathbf{V}_1 = \mathbf{V}_0 + \frac{\Delta \mathbf{x} \Delta \mathbf{x}^T}{\Delta \mathbf{x}^T \Delta \mathbf{x}} - \frac{\mathbf{V}_0 \Delta \mathbf{g} \Delta \mathbf{g}^T \mathbf{V}_0}{\Delta \mathbf{g}^T \mathbf{V}_0 \Delta \mathbf{g}}, \quad (5.10)$$

где $\Delta \mathbf{g} = \mathbf{g}_1 - \mathbf{g}_0$. Для успешной работы этого алгоритма надо обязательно проводить одномерную минимизацию в п. 2.

Алгоритм, не требующий обязательной минимизации на каждом шаге, задаётся преобразованием

$$\mathbf{V}_1 = \mathbf{V}_0 + \frac{(\Delta \mathbf{x} - \mathbf{V}_0 \Delta \mathbf{g})(\Delta \mathbf{x} - \mathbf{V}_0 \Delta \mathbf{g})^T}{\Delta \mathbf{g}^T (\Delta \mathbf{x} - \mathbf{V}_0 \Delta \mathbf{g})}, \quad (5.11)$$

однако применение этого алгоритма не даёт хорошего результата, так как знаменатель дроби довольно часто получается близок к нулю, и тогда приращения матрицы получаются большими.

Результаты многочисленных исследований подходящих преобразований ковариационной матрицы были обобщены в работе [55], практические выводы которой мы здесь приведём.

Обозначим преобразование в уравнении (5.10) как $\mathbf{V}_1 = \Psi_1(\mathbf{V}_0)$. Обратное к этому преобразование $\mathbf{V}_1 = \Psi_2(\mathbf{V}_0)$ записывается следующим образом:

$$\mathbf{V}_1 = \left(1 - \frac{\Delta \mathbf{x} \Delta \mathbf{g}^T}{\Delta \mathbf{x}^T \Delta \mathbf{g}}\right) \mathbf{V}_0 \left(1 - \frac{\Delta \mathbf{g} \Delta \mathbf{x}^T}{\Delta \mathbf{x}^T \Delta \mathbf{g}}\right) + \frac{\Delta \mathbf{x} \Delta \mathbf{x}^T}{\Delta \mathbf{x}^T \Delta \mathbf{g}}. \quad (5.12)$$

Оказывается, существует широкий класс преобразований, определяемый параметром α :

$$\mathbf{V}_1 = (1 - \alpha)\Psi_1(\mathbf{V}_0) + \alpha\Psi_2(\mathbf{V}_0), \quad (5.13)$$

который обеспечивает монотонную сходимость \mathbf{V}_i к матрице \mathbf{V} квадратичной формы при значениях α в интервале $[0, 1]$. Преобразование (5.11) тоже попадает в этот класс с

$$\alpha_1 = \frac{\Delta \mathbf{x}^T \Delta \mathbf{g}}{(\Delta \mathbf{x}^T \Delta \mathbf{g} - \Delta \mathbf{g}^T \mathbf{V}_0 \Delta \mathbf{g})}. \quad (5.14)$$

Легко заметить, что α_1 никогда не попадает в интервал $[0, 1]$. На практике хорошо работает алгоритм, реализованный в программе MINUIT: если на очередном шаге $\alpha_1 < 0$, то используется преобразование (5.13) с $\alpha = 0$, а если $\alpha_1 > 1$, то преобразование (5.13) с $\alpha = 1$. Рис. 5.12 демонстрирует ход минимизации по этому алгоритму из точки $(1, 5; 2)$.

Как видно, скорость нахождения минимума самая высокая среди всех рассмотренных методов. Однако и этот алгоритм может ошибаться. Если начать минимизацию из точки $(2; 2)$, то программа уверенно сообщает, что нашла минимум в седловой точке $\mathbf{x} = (4, 898; 4, 898)$. Кроме того, мы использовали возможность аналитически вычислять градиент, если же предлагать программе численно оценивать градиент, то минимизация идёт гораздо медленнее (рис. 5.13), однако в этом случае симметрия нарушается и минимум находится и из начальной точки $(2; 2)$.

Алгоритм минимизации методом переменной метрики пространства хорош ещё и тем, что, как и в методе Ньютона, зная матрицу \mathbf{V} , можно предсказать не только положение минимума (5.8), но и значение функции в этой точке:

$$F_{min} = F_0 - \frac{1}{2} \mathbf{g}_0^T \mathbf{V} \mathbf{g}_0. \quad (5.15)$$

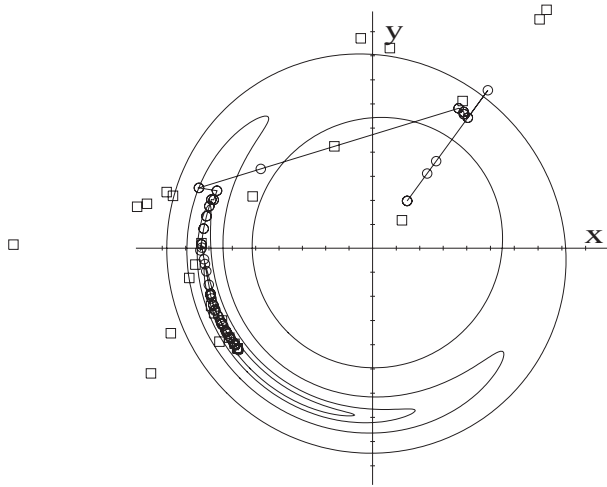


Figure 5.12: Метод переменной метрики пространства. Программа MINUIT с аналитическим вычислением градиента.

$$n_{0.1} = 87, \quad n_{0.01} = 101, \\ n_{0.001} = 111, \quad \min_{i=1,100} f(x_i, y_i) = \\ f_{98}(-5, 661; -4, 244) = 0, 013$$

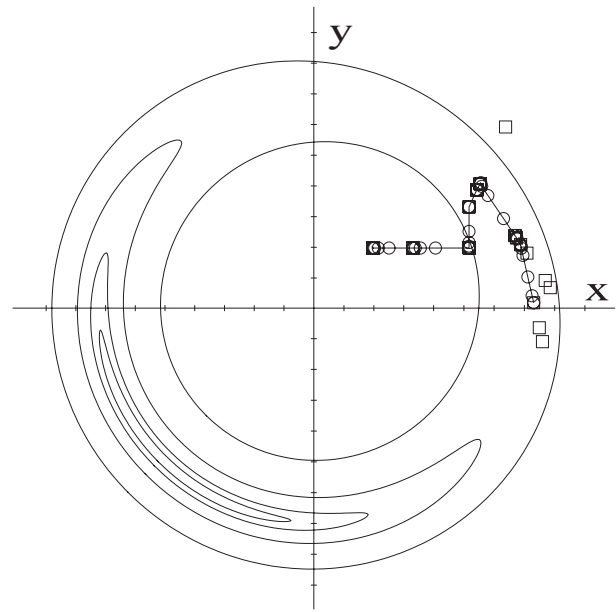


Figure 5.13: Метод переменной метрики пространства. Программа MINUIT с численной оценкой градиента. $n_{0.1} = 242$, $n_{0.01} = 272$, $n_{0.001} = 309$, $\min_{i=1,100} f(x_i, y_i) = f_{100}(7, 335; 0, 184) = 321, 686$

Эту возможность можно использовать при выработке критерия достижения минимума.

5.13 Критерий нахождения минимума

Во всех рассмотренных алгоритмах мы приводили только правила смещения из некоторой точки в следующую и совсем не рассматривали критерий окончания этого итерационного процесса. Задача является нетривиальной, и в трудных случаях любой алгоритм может ошибаться. Во-первых, надо сразу разделить два подхода в целях минимизации: локализация координат минимума с заданными допусками по каждому направлению или локализация минимума с предельным отклонением функции в найденной точке от минимального значения. В задачах оптимальной оценки параметров, как будет видно дальше, необходим второй критерий – по допустимому отклонению функции от минимума. Единственный вариант реализации такого критерия – следить за изменениями функции в процессе минимизации, и если на заданном количестве итерационных шагов подряд изменение функции не превысило заданный уровень, то считать минимум найденным. Дополнительные возможности предоставляют алгоритмы, производящие оценку ковариационной матрицы в процессе минимизации. Тогда имеется возможность сравнить предсказанное значение функции в минимуме с действительным значением функции в этой точке.

Следует помнить, что критерий нахождения минимума может давать сбой, поэтому полезно проводить несколько процессов минимизации из разных начальных точек.

5.14 Поиск глобального минимума

Все рассмотренные алгоритмы позволяют найти какой-либо локальный минимум. Чаще всего, однако, представляет интерес глобальный минимум, т.е. наименьший минимум из всех локальных.

Алгоритма надёжного поиска глобального минимума не найдено. В работе [56] предложен способ выбора новых начальных точек для последовательных попыток найти новый локальный минимум — предлагается исходную точку для поиска нового минимума выбирать на прямой линии, соединяющей точки ранее найденных двух локальных минимумов. Малая эффективность такого алгоритма очевидна, так как

предположение, что локальные минимумы «выстроены» по прямой линии, ни на чём не основывается, хотя для некоторых функций это может оказаться и справедливым. Не менее перспективным может оказаться и просто случайный выбор начальной точки с широким разбросом по всем переменным. Кроме того, нет никакого предложения, когда следует прекращать поиск новых локальных минимумов.

Более интересный способ предложен в работе [57]. Пусть мы нашли некоторый локальный минимум \mathbf{x}_1 и имеем оценку матрицы вторых производных \mathbf{M} в точке минимума. Образует вспомогательную функцию

$$f_1(\mathbf{x}) = \frac{2[f(\mathbf{x}) - f(\mathbf{x}_1)]}{(\mathbf{x} - \mathbf{x}_1)^T \mathbf{M} (\mathbf{x} - \mathbf{x}_1)}. \quad (5.16)$$

По построению новой функции мы «выключили» найденный минимум, поэтому, если в результате минимизации $f_1(\mathbf{x})$ будет найден другой минимум в \mathbf{x}_2 , то это может свидетельствовать о наличии другого локального минимума функции $f(\mathbf{x})$, который, правда, придётся уточнять, начиная минимизацию $f(\mathbf{x})$ с точки \mathbf{x}_2 . В случае, когда $f_1(\mathbf{x}_2) < 0$, успех в поиске нового локального минимума гарантирован, так как это означает $f(\mathbf{x}_2) < f(\mathbf{x}_1)$. Этот процесс поиска новых локальных минимумов можно продолжать и дальше, и, как всегда, нет критериев для окончания этого процесса.

Chapter 6

Определение параметров теоретической модели по экспериментальным данным

Эксперименты в физике элементарных частиц часто проводятся для определения свободного параметра какой-либо теоретической модели. Рассмотрим сначала общий подход к этой задаче.

6.1 Метод максимального правдоподобия

Пусть распределение вероятностей по значениям величин, измеряемых в эксперименте, зависит известным образом от искомого параметра α : $dW = f(\mathbf{x}, \alpha) d\mathbf{x}$. Функция $f(\mathbf{x}, \alpha)$ называется функцией правдоподобия. Рассмотрим некоторую оценку параметра, являющуюся функцией измеренных величин $y(\mathbf{x})$. Очевидно, что если повторять этот эксперимент многократно в неизменных условиях, то значения величин y будут как-то распределены вокруг истинного значения параметра α . Если среднее значение оценок параметра совпадает с истинным значением параметра

$$\int y(\mathbf{x})f(\mathbf{x}, \alpha) d\mathbf{x} = \alpha, \quad (6.1)$$

то оценка называется несмещённой. Оценка параметра называется *состоятельной*, если при увеличении статистики оценка параметра сходится к истинному значению по вероятности, т.е. для любых малых величин $\delta > 0$, $\epsilon > 0$ найдётся N такое, что вероятность

$$P(|y - \alpha| > \delta) < \epsilon$$

для любого $n > N$. Количество возможных несмещённых и состоятельных оценок для данного параметра всё ещё велико, однако среди них есть оценки, которые дают наименьший разброс $D(y)$ значений относительно истинного значения параметра. Такие оценки называются эффективными.

Принцип максимального правдоподобия позволяет получить асимптотически (при большой статистике) несмещённую, состоятельную и эффективную оценку [12, 58, 59]. В соответствии с этим принципом необходимо найти значение параметра α , которое при измеренных значениях \mathbf{x} обеспечивает максимум функции правдоподобия $f(\mathbf{x}, \alpha)$. На практике чаще используется логарифмическая функция правдоподобия

$$L(\mathbf{x}, \alpha) = -\ln f(\mathbf{x}, \alpha).$$

В соответствии с определением необходимо искать минимум функции L . Поиск минимума аналитическими методами часто невозможен или очень затруднён, в этом случае приходится применять численные методы минимизации на ЭВМ, рассмотренные в разд. 5.

В экспериментальных результатах принято всегда приводить статистическую и систематическую ошибку. Для получения статистической ошибки результата применяется следующий метод. Пусть в результате минимизации получено значение $\min_{\alpha} L(\mathbf{x}, \alpha) = L(\mathbf{x}, \alpha_0) = L_0$ (рис. 6.1).

Проведём теперь на уровне $L_0 + 0,5$ горизонтальную прямую, пересекающую график функции правдоподобия в двух точках, которые определяют 68%-й доверительный интервал. Другими словами, с вероятностью 68% истинное значение α содержится в интервале $(\alpha_0 - \Delta_1, \alpha_0 + \Delta_2)$. Часто результат эксперимента записывают в виде

$$\alpha = \alpha_0 \begin{matrix} +\Delta_2 \\ -\Delta_1 \end{matrix}.$$

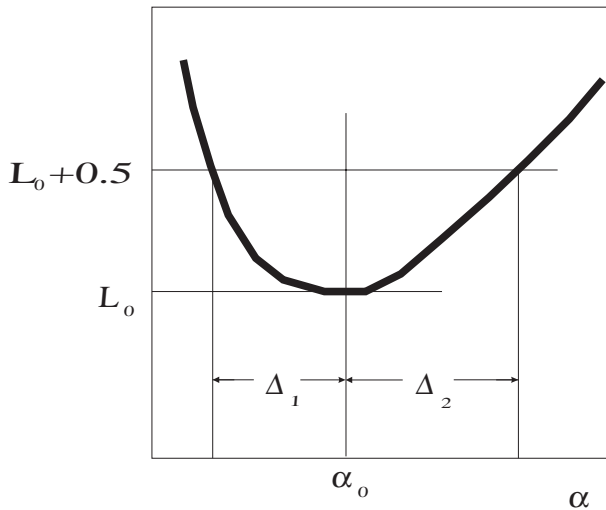


Figure 6.1: Определение доверительного интервала для оценки параметра α по логарифмической функции правдоподобия

Горизонтальная прямая на уровне $L_0 + 0,5$, пересекающая график функции правдоподобия в двух точках, определяет 68%-й доверительный интервал. Результат эксперимента записывают в виде

$$\alpha = \alpha_0 \begin{matrix} +\Delta_2 \\ -\Delta_1 \end{matrix} .$$

В методе максимума правдоподобия с доверительными интервалами, полученными по графику функции правдоподобия, есть прямая аналогия с распределением Гаусса. Если мы провели горизонтальную прямую на уровне $L_0 + \Delta L$, то получаем интервал значений α с таким уровнем доверия, с какой вероятностью случайная величина ξ , распределённая по закону Гаусса, попадает в интервал $(\xi_0 - \sqrt{2\Delta L}\sigma, \xi_0 + \sqrt{2\Delta L}\sigma)$. Если имеются таблицы или графики $P_n(\chi^2)$ (см. разд. 9.1), то уровень доверия $C.L.$ (Confidence Level) можно вычислить следующим образом: $C.L. = 1 - P_1(2\Delta L)$. Действительно, для стандартного случая $\Delta L = 0,5$ получаем $C.L. = 1 - P_1(1) = 1 - 0,3173 = 0,6827$.

В случае, если в теоретической модели несколько свободных параметров, то правило нахождения оптимальной оценки и доверительного интервала одного из параметров остаются прежними, только теперь функция правдоподобия зависит от нескольких параметров, и в каждой точке графика $L(\alpha)$ функцию правдоподобия надо минимизировать по всем свободным параметрам, кроме α .

Надо иметь в виду, что при наличии нескольких параметров модели их экспериментальные оценки могут иметь значительные статистические корреляции. Для учёта этих корреляций в первом приближении достаточно знать элементы ковариационной матрицы в минимуме логарифмической

функции правдоподобия. При некоторых способах численной минимизации (см. разд. 5) оценка ковариационной матрицы производится попутно при поиске минимума. В простых случаях можно получать оценку ковариационной матрицы, которая равна обратной матрице к матрице вторых производных логарифмической функции правдоподобия.

Если необходим более полный учёт корреляций при вычислении некоторой функции $R(\alpha, \beta, \dots)$ от оцениваемых параметров, то самый простой и надёжный способ заключается в следующем. Надо исключить из списка параметров модели один из параметров, выразив его значение в общем виде через значение функции R и остальные параметры. После этого R станет одним из параметров модели, и, проделав описанные ранее действия с функцией правдоподобия, мы получим правильные оценки доверительного интервала для функции R . Приведём два простых примера.

Пример 1. Пусть мы провели несколько одинаковых измерений одной и той же величины и получили набор экспериментальных значений $x_i, i = 1, \dots, N$. Предполагается, что величина x распределена по закону Гаусса

$$dW = \frac{dx}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(x - a_0)^2}{2\sigma_0^2}\right], \quad (6.2)$$

где a_0, σ_0 — параметры распределения. Составляем функцию правдоподобия:

$$f(x_1, x_2, \dots, x_N, a, \sigma) = \frac{\exp\left(-\sum_{i=1}^N \frac{(x_i - a)^2}{2\sigma^2}\right)}{(\sqrt{2\pi}\sigma)^N} \quad (6.3)$$

и логарифмическую функцию правдоподобия

$$L(x_1, x_2, \dots, x_N, a, \sigma) = \left(\sum_{i=1}^N \frac{(x_i - a)^2}{2\sigma^2}\right) + N \ln(\sqrt{2\pi}\sigma). \quad (6.4)$$

Находя частную производную $\frac{\partial L}{\partial a}$ и приравнивая её к нулю, получаем оценку параметра a :

$$a = \frac{1}{N} \sum_{i=1}^N x_i. \quad (6.5)$$

Легко проверить, что эта оценка является несмещённой, т. е. $\langle a \rangle = a_0$. Решая уравнение $\frac{\partial L}{\partial \sigma} = 0$, получаем оценку для σ :

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - a)^2 = \frac{1}{N} \sum x_i^2 - \left(\frac{1}{N} \sum_{i=1}^N x_i \right)^2. \quad (6.6)$$

Проверка показывает, что эта оценка получается смещённой, так как $\langle \sigma^2 \rangle = \frac{N-1}{N} \sigma_0^2$. Правда, асимптотически эта оценка всё же несмещённая:

$$\lim_{N \rightarrow \infty} \frac{\langle \sigma^2 \rangle}{\sigma_0^2} = 1. \quad (6.7)$$

Введением дополнительного множителя $\frac{N}{N-1}$ оценка (6.6) превращается в несмещённую:

$$\hat{\sigma}^2 = \frac{N}{N-1} \sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - a)^2. \quad (6.8)$$

Вычислим матрицу вторых производных

$$\begin{aligned} M &= \begin{vmatrix} \frac{\partial^2 L}{\partial a^2} & \frac{\partial^2 L}{\partial a \partial \sigma} \\ \frac{\partial^2 L}{\partial a \partial \sigma} & \frac{\partial^2 L}{\partial \sigma^2} \end{vmatrix} = \begin{vmatrix} \frac{N}{\sigma^2} & \frac{2}{\sigma^3} \sum_{i=1}^N (x_i - a) \\ \frac{2}{\sigma^3} \sum_{i=1}^N (x_i - a) & \frac{3}{\sigma^4} \sum_{i=1}^N (x_i - a)^2 - \frac{N}{\sigma^2} \end{vmatrix} \Rightarrow \\ &\Rightarrow \begin{vmatrix} \frac{N}{\sigma^2} & 0 \\ 0 & \frac{2N}{\sigma^2} \end{vmatrix} \end{aligned} \quad (6.9)$$

Обратная к ней ковариационная матрица

$$V = M^{-1} = \begin{vmatrix} \frac{\sigma^2}{N} & 0 \\ 0 & \frac{\sigma^2}{2N} \end{vmatrix} \quad (6.10)$$

определяет дисперсию оценки каждого параметра и корреляцию (в данном случае корреляция равна нулю). Окончательно можно выписать оценки параметров с ошибками:

$$a = \frac{1}{N} \sum_{i=1}^N x_i \pm \frac{\sigma}{\sqrt{N}}, \quad \sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - a)^2} \pm \frac{\sigma}{\sqrt{2N}}. \quad (6.11)$$

Пример 2. Пусть мы регистрируем число событий n_i в точках x_i , $i = 1, \dots, k$, причём для простоты примем $x_i = x_0 + ih$. Допустим, математическое ожидание числа событий

$$\langle n_i \rangle = b_0 \cdot \exp \left[-\frac{(x_i - a_0)^2}{2\sigma_0^2} \right]. \quad (6.12)$$

Требуется получить оценки параметров a_0, b_0, σ_0 , которые далее мы будем обозначать теми же буквами без индексов.

Предполагая, что события независимы друг от друга, получим, что распределение в каждой точке x_i подчиняется закону Пуассона

$$W(n_i) = \frac{\langle n_i \rangle^{n_i}}{n_i!} \exp[-\langle n_i \rangle].$$

Тогда логарифмическую функцию правдоподобия можно записать в виде

$$L = \sum_{i=1}^k [\langle n_i \rangle + \ln(n_i!) - n_i \ln \langle n_i \rangle] \Rightarrow \sum_{i=1}^k [\langle n_i \rangle - n_i \ln \langle n_i \rangle]. \quad (6.13)$$

Здесь отброшено слагаемое $\ln(n_i!)$ как совсем не зависящее от искомым параметров.

Вычислив частные производные по всем трём параметрам и приравняв их к нулю, получим систему трёх алгебраических уравнений для определения оптимальных оценок параметров a, b, σ :

$$\begin{aligned} \frac{\partial L}{\partial b} = 0 &\Rightarrow \sum_{i=1}^k \left\{ \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] - \frac{n_i}{b} \right\} = 0, \\ \frac{\partial L}{\partial a} = 0 &\Rightarrow \sum_{i=1}^k \left\{ \left(b \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] - n_i \right) \frac{x_i - a}{\sigma^2} \right\} = 0, \\ \frac{\partial L}{\partial \sigma} = 0 &\Rightarrow \sum_{i=1}^k \left\{ \left(b \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] - n_i \right) \frac{(x_i - a)^2}{\sigma^3} \right\} = 0. \end{aligned} \quad (6.14)$$

Несмотря на то что этот пример по сложности уступает типичным задачам в обработке экспериментальных данных, тем не менее очевидно, что аналитическое решение системы (6.14) невозможно. Однако если принять допущение, что $h \ll \sigma_0$, $|a_0 - x_0| \gg \sigma_0$, $|x_0 + kh - a_0| \gg \sigma_0$, то некоторые суммы вблизи минимума L неплохо аппроксимируются интегралами

$$\begin{aligned}
\sum_{i=1}^k \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] &= \frac{1}{h} \sum_{i=1}^k h \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] \approx \\
&\approx \frac{1}{h} \int_{-\infty}^{+\infty} \exp \left[-\frac{(x - a)^2}{2\sigma^2} \right] dx = \sqrt{2\pi} \frac{\sigma}{h}, \\
\sum_{i=1}^k (x_i - a) \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] &\approx 0, \\
\sum_{i=1}^k (x_i - a)^2 \exp \left[-\frac{(x_i - a)^2}{2\sigma^2} \right] &\approx \sqrt{2\pi} \frac{\sigma^3}{h},
\end{aligned} \tag{6.15}$$

что позволяет существенно упростить систему (6.14) и провести анализ ковариационной матрицы:

$$\begin{aligned}
\sqrt{2\pi} \frac{\sigma}{h} - \frac{1}{b} \sum_{i=1}^k n_i &= 0, \quad \sum_{i=1}^k n_i (x_i - a) = 0, \\
\sqrt{2\pi} \frac{b}{h} - \frac{1}{\sigma^3} \sum_{i=1}^k n_i (x_i - a)^2 &= 0.
\end{aligned} \tag{6.16}$$

Введя обозначение $N = \sum_{i=1}^k n_i$, получим решение системы

$$a = \frac{1}{N} \sum_{i=1}^k n_i x_i, \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^k n_i (x_i - a)^2}, \quad b = \frac{hN}{\sqrt{2\pi}\sigma}. \tag{6.17}$$

Оценку матрицы вторых производных можно существенно упростить, если предположить, что экспериментальные измерения близки к теоретическому описанию и $\sum_{i=1}^k n_i (x_i - a)^2 \approx \sum_{i=1}^k \langle n_i \rangle (x_i - a)^2$. Для идентификации столбцов и строк матриц будем считать, что вектор параметров модели имеет в качестве своих компонентов (a, b, σ) .

$$M = \left\| \begin{array}{ccc} \frac{N}{\sigma^2} & 0 & 0 \\ 0 & \frac{N}{b^2} & \frac{\sqrt{2\pi}}{h} \\ 0 & \frac{\sqrt{2\pi}}{h} & \frac{3N}{\sigma^2} \end{array} \right\|, \quad V = M^{-1} = \left\| \begin{array}{ccc} \frac{\sigma^2}{N} & 0 & 0 \\ 0 & \frac{3b^2}{2N} & -\frac{h}{2\sqrt{2\pi}} \\ 0 & -\frac{h}{2\sqrt{2\pi}} & \frac{\sigma^2}{2N} \end{array} \right\|. \tag{6.18}$$

Теперь мы можем переписать результаты (6.17) вместе с оценкой ошибки:

$$a = \frac{1}{N} \sum_{i=1}^k n_i x_i \pm \frac{\sigma}{\sqrt{N}}, \quad b = \frac{hN}{\sqrt{2\pi\sigma}} \times (1 \pm \sqrt{\frac{3}{2N}}),$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^k n_i (x_i - a)^2} \times (1 \pm \frac{1}{\sqrt{2N}}).$$
(6.19)

В рассматриваемом примере ковариационная матрица получилась недиагональная. Можно сказать, что между параметрами b и σ имеется отрицательная корреляция. Посмотрим, как это сказывается на вычислении ошибок величин, являющихся функциями определённых нами параметров. Допустим, что нас интересует величина $S = \frac{\sqrt{2\pi\sigma b}}{h}$ («площадь под кривой»). Как подсчитать оценку этого параметра и ошибку по результатам эксперимента? Оптимальная оценка параметра S получается подстановкой оценок параметров в функцию: $S = \frac{\sqrt{2\pi\sigma}}{h} \frac{hN}{\sqrt{2\pi\sigma}} = N$. Однако неправильно оценивать относительную ошибку этого параметра, как относительную ошибку произведения независимых случайных величин – корень квадратный из суммы квадратов относительных ошибок

$$\frac{\Delta S}{S} \neq \sqrt{\left(\sqrt{\frac{3}{2N}}\right)^2 + \left(\frac{1}{\sqrt{2N}}\right)^2} = \sqrt{\frac{2}{N}}.$$

Правильная процедура использования ковариационной матрицы следующая. Если мы обозначим $\alpha_1 = a$, $\alpha_2 = b$, $\alpha_3 = \sigma$, а новый параметр $S = \Psi(\alpha)$, то дисперсия S в первом приближении равна

$$D(S) = \langle (S - \langle S \rangle)^2 \rangle = \sum_{j,m} v_{jm} \frac{\partial \Psi}{\partial \alpha_j} \frac{\partial \Psi}{\partial \alpha_m}. \quad (6.20)$$

Применяя формулу (6.20) к нашему случаю, получаем

$$D(S) = \frac{3b^2}{2N} \cdot \left(\frac{\sqrt{2\pi\sigma}}{h}\right)^2 - 2 \cdot \frac{h}{2\sqrt{2\pi}} \cdot \frac{\sqrt{2\pi\sigma}}{h} \cdot \frac{\sqrt{2\pi b}}{h} + \frac{\sigma^2}{2N} \cdot \left(\frac{\sqrt{2\pi b}}{h}\right)^2 = N.$$

Отсюда для относительной ошибки получаем $\frac{1}{\sqrt{N}}$. Итак, ошибочно пренебрегая корреляцией, получаем относительную ошибку $\sqrt{\frac{2}{N}}$, а с учётом корреляции – $\frac{1}{\sqrt{N}}$.

6.2 Метод наименьших квадратов

Частным случаем метода максимума функции правдоподобия можно рассматривать метод наименьших квадратов. Пусть в нескольких точках $x_i, i = 1, \dots, k$ произведены измерения некоторой величины y_i с известными ошибками σ_i . Предполагаем, что распределение измеренной величины относительно точного значения $y(x_i, \alpha_1, \alpha_2, \dots)$ гауссово, параметры $\alpha_1, \alpha_2, \dots$ надо определить по результатам этих измерений.

Логарифмическая функция правдоподобия получается равной

$$\begin{aligned} L &= -\ln \left\{ \prod_{i=1}^k \left[\frac{1}{\sqrt{2\pi}\sigma_i} \exp \left[-\frac{(y_i - y(x_i, \alpha_1, \dots))^2}{2\sigma_i^2} \right] \right] \right\} = \\ &= \sum_{i=1}^k \left[\frac{[y_i - y(x_i, \alpha_1, \dots)]^2}{2\sigma_i^2} + \ln [\sqrt{2\pi}\sigma_i] \right]. \end{aligned} \quad (6.21)$$

Отбросив не зависящие от искоемых параметров α_i слагаемые $\ln [\sqrt{2\pi}\sigma_i]$ и умножив всю функцию на 2, получим модифицированную функцию правдоподобия

$$\tilde{L} = \sum_{i=1}^k \left[\frac{[y_i - y(x_i, \alpha_1, \dots)]^2}{\sigma_i^2} \right], \quad (6.22)$$

минимизация которой приводит к тем же оптимальным параметрам, что и минимизация L . Цель оптимизации параметров имеет здесь наглядный смысл – надо подобрать такие параметры α_i , чтобы минимизировать сумму квадратов отклонений (нормированных на ошибки) результатов измерений от теоретической кривой. При поиске ошибок параметров по графику функции \tilde{L} не надо забывать о добавленном множителе 2, поскольку 68%-й доверительный интервал определяется на уровне +1 от минимума функции \tilde{L} .

Пример. Проведены измерения величины y_i в ряде точек $x_i, i = 1, \dots, k$ с ошибками σ_i . Требуется провести прямую наилучшим образом. Прямая должна однозначно задаваться двумя параметрами. Выберем параметризацию прямой в виде $y(x_i, b, c) = b \cdot (x_i - a) + c$, где константа a не является подгоняемым параметром, её значение мы выберем позднее из соображений удобства формул. Минимизируемая функция запишется в виде

$$\tilde{L} = \sum_{i=1}^k \frac{1}{\sigma_i^2} [b \cdot (x_i - a) + c - y_i]^2. \quad (6.23)$$

Условия минимума функции:

$$\begin{aligned} \frac{\partial \tilde{L}}{\partial b} = 0 &\Rightarrow \sum_{i=1}^k \frac{x_i - a}{\sigma_i^2} [b \cdot (x_i - a) + c - y_i] = 0, \\ \frac{\partial \tilde{L}}{\partial c} = 0 &\Rightarrow \sum_{i=1}^k \frac{1}{\sigma_i^2} [b \cdot (x_i - a) + c - y_i] = 0. \end{aligned} \quad (6.24)$$

Легко заметить, что если мы выберем константу a из условия

$$\sum (x_i - a) / \sigma_i^2 = 0,$$

то система уравнений (6.24) существенно упрощается и параметры b, c становятся статистически независимы (ковариационная матрица диагональна). В результате получаем

$$\begin{aligned} a &= \frac{\sum_{i=1}^k \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^k \frac{1}{\sigma_i^2}}, \quad c = \frac{\sum_{i=1}^k \frac{y_i}{\sigma_i^2}}{\sum_{i=1}^k \frac{1}{\sigma_i^2}} \pm \frac{1}{\sqrt{\sum_{i=1}^k \frac{1}{\sigma_i^2}}}, \\ b &= \frac{\sum_{i=1}^k \frac{y_i}{\sigma_i^2} (x_i - a)}{\sum_{i=1}^k \frac{(x_i - a)^2}{\sigma_i^2}} \pm \frac{1}{\sqrt{\sum_{i=1}^k \frac{(x_i - a)^2}{\sigma_i^2}}}. \end{aligned} \quad (6.25)$$

6.3 Усреднение экспериментальных результатов

Пусть проведено несколько измерений одной и той же величины с разными ошибками. Требуется усреднить результаты этих измерений. В этом случае $y(x_i, \alpha) = \alpha$. Приведём попутно пример несмещённой оценки параметра $\hat{\alpha} = \frac{1}{k} \sum_{i=1}^k y_i$, которая не является эффективной. Легко проверить, что среднее значение этого параметра является правильным,

но дисперсия $D(\hat{\alpha}) = \frac{1}{k^2} \sum_{i=1}^k \sigma_i^2$ превышает дисперсию оптимальной оценки, которую мы сейчас получим. Приравнивая к нулю частную производную $\frac{\partial \tilde{L}}{\partial \alpha}$ и вычисляя ковариационную матрицу для получения ошибки параметра, получаем несмещённую и эффективную оценку

$$\alpha = \frac{\sum_{i=1}^k \frac{y_i}{\sigma_i^2}}{\sum_{i=1}^k \frac{1}{\sigma_i^2}} \pm \frac{1}{\sqrt{\sum_{i=1}^k \frac{1}{\sigma_i^2}}}. \quad (6.26)$$

Теперь мы можем сравнить дисперсии двух оценок:

$$\begin{aligned} \frac{D(\hat{\alpha})}{D(\alpha)} &= \frac{\frac{1}{k^2} \sum_{i=1}^k \sigma_i^2}{\frac{1}{\sum_{j=1}^k \frac{1}{\sigma_j^2}}} = \frac{1}{k^2} \left(\sum_{i=1}^k \sigma_i^2 \right) \left(\sum_{j=1}^k \frac{1}{\sigma_j^2} \right) = \frac{1}{k^2} \left(k + \sum_{i < j} \left[\frac{\sigma_i^2}{\sigma_j^2} + \frac{\sigma_j^2}{\sigma_i^2} \right] \right) \geq \\ &\geq \frac{1}{k^2} \left(k + \sum_{i < j} 2 \right) = 1. \end{aligned} \quad (6.27)$$

Не всегда случайные величины, участвующие в изучаемом процессе, имеют Гауссово распределение вероятностей. В таких случаях, обычно, границы доверительного интервала несимметричны относительно оценки параметра, и результат измерения представляется в виде

$$x = a_{-\sigma_-}^{+\sigma_+}, \text{ где } \sigma_+ > 0, \sigma_- > 0. \quad (6.28)$$

В этом случае рассмотренная выше процедура усреднения не может быть использована. Величины ошибок определяют с некоторой степенью точности зависимость функции правдоподобия от параметра. Типичная зависимость функции правдоподобия вблизи минимума – парабола, однако, асимметрия ошибок определяет поправки следующего порядка. Наиболее простая и надёжная модель функции правдоподобия с несимметричными ошибками может быть создана по аналогии с процедурой оценки среднего значения Пуассонова распределения по однократному измерению:

$$L \sim a_1 \cdot x + a_2 \cdot \ln(x + a_3). \quad (6.29)$$

Для того, чтобы эта функция имела минимум в точке $x = a$, набор параметров следует модифицировать следующим образом:

$$L = \frac{1}{b} \cdot [z \cdot (x - a) - \ln(1 + z \cdot (x - a))], \quad b > 0. \quad (6.30)$$

Для нахождения коэффициентов b и z надо решить систему уравнений:

$$\begin{cases} \frac{1}{b} \cdot [z \cdot \sigma_+ - \ln(1 + z \cdot \sigma_+)] = \frac{1}{2}, \\ \frac{1}{b} \cdot [-z \cdot \sigma_- - \ln(1 - z \cdot \sigma_-)] = \frac{1}{2}. \end{cases} \quad (6.31)$$

Переменные разделяются, и для определения параметра z надо решить уравнение:

$$z \cdot (\sigma_+ + \sigma_-) - \ln \frac{1 + z \cdot \sigma_+}{1 - z \cdot \sigma_-} = 0. \quad (6.32)$$

При $\sigma_+ \neq \sigma_-$ решение $z = 0$ является ложным. Другое решение легко может быть найдено численными методами. При $\sigma_+ > \sigma_-$ решение находится в интервале

$$\frac{1}{\sigma_-} - \frac{1}{\sigma_+} < z < \frac{1}{\sigma_-},$$

а в случае $\sigma_+ < \sigma_-$ — в интервале

$$-\frac{1}{\sigma_+} < z < \frac{1}{\sigma_-} - \frac{1}{\sigma_+}.$$

После определения z параметр b может быть найден из любого из двух уравнений системы (6.31).

При близких значениях σ_{\pm} параметры z, b стремятся к нулю, и решение может быть найдено путём разложения в ряд логарифмов:

$$z = \frac{3}{2} \cdot \frac{\sigma_+ - \sigma_-}{\sigma_+^2 - \sigma_+ \sigma_- + \sigma_-^2}; \quad b = \frac{9}{8} \cdot \frac{(\sigma_+^2 - \sigma_-^2)^2 (\sigma_+^2 + \sigma_-^2)}{(\sigma_+^3 + \sigma_-^3)^2}. \quad (6.33)$$

Нетрудно убедиться, что при малых z функция правдоподобия (6.30) превращается в

$$L \approx \frac{z^2 \cdot (x - a)^2}{2b} = \frac{(x - a)^2}{\sigma_+^2 + \sigma_-^2} \approx \frac{(x - a)^2}{2\sigma_{\pm}^2} \quad (6.34)$$

Теперь, когда мы реконструировали для каждого экспериментального результата вид функции правдоподобия с некоторой точностью, можно использовать суммарную функцию правдоподобия для получения усреднённого результата. Операции численного нахождения минимума функции одной переменной и корней уравнения $L(x) = L_{min} + 1/2$ не составляет трудностей.

6.4 Пример нестандартной оценки параметра

Пусть у нас есть распределение событий по некоторому параметру m , описываемое функцией распределения

$$\frac{dN}{dm} = F(m) \cdot f(m), \quad 0 < m < \infty, \quad (6.35)$$

где $f(m)$ — известная функция. Надо получить оценку среднего значения неизвестной функции $F(m)$ на некотором интервале (a, b) , имея массив событий достаточно большого объёма n .

Непонятно, как можно было бы достаточно просто провести оценку стандартными методами без существенных упрощений или допущений. Хорошо бы ввести искомый параметр в формулу, описывающую распределение вероятностей, составить функцию правдоподобия и методом максимального правдоподобия провести оценку этого параметра. Однако, как сделать это в данном примере не очень понятно.

Введём вспомогательный параметр

$$\rho = \int_0^m f(\xi) d\xi. \quad (6.36)$$

Так как функция $f(m)$ положительная, то зависимость ρ и m взаимно однозначная и

$$\frac{d\rho}{dm} = f(m). \quad (6.37)$$

Распределение событий по параметру ρ описывается функцией

$$\frac{dN}{d\rho} = \frac{dN}{dm} \bigg/ \frac{d\rho}{dm} = F(m). \quad (6.38)$$

Нас интересует оценка среднего значения

$$R = \frac{1}{b-a} \int_a^b F(m) dm = \frac{1}{b-a} \int_{\rho_a}^{\rho_b} F(m) \frac{d\rho}{f(m)} \quad (6.39)$$

Рассмотрим среднее значение

$$\left\langle \frac{1}{f(m)} \right\rangle = \int_{\rho_a}^{\rho_b} \frac{1}{f(m)} \cdot \frac{dN}{d\rho} d\rho \bigg/ \int_{\rho_a}^{\rho_b} \frac{dN}{d\rho} d\rho = \frac{b-a}{N_{ab}} \cdot R. \quad (6.40)$$

Отсюда

$$R = \frac{N_{ab}}{b-a} \left\langle \frac{1}{f(m)} \right\rangle, \quad (6.41)$$

где обозначено N_{ab} — число событий в интервале $a < m < b$. Введём дополнительные обозначения

$$\mu_1 = \frac{1}{N_{ab}} \sum_{a < m < b} \frac{1}{f(m)}, \quad \mu_2 = \frac{1}{N_{ab}} \sum_{a < m < b} \left[\frac{1}{f(m)} \right]^2. \quad (6.42)$$

Учитывая, что

$$\left\langle \frac{1}{f(m)} \right\rangle = \mu_1 \pm \sqrt{\frac{\mu_2 - \mu_1^2}{N_{ab}}}, \quad (6.43)$$

получаем

$$R = \frac{N_{ab} \pm \sqrt{N_{ab}}}{b-a} \cdot \left[\mu_1 \pm \sqrt{\frac{\mu_2 - \mu_1^2}{N_{ab}}} \right] = \frac{N_{ab} \cdot \mu_1}{b-a} \cdot \left[1 \pm \frac{\sqrt{\mu_2}}{\mu_1 \sqrt{N_{ab}}} \right]. \quad (6.44)$$

Здесь проведено квадратичное сложение относительных ошибок от двух источников: флуктуации числа событий, попавших в интервал $a < m < b$, и эффект случайного перераспределения событий по параметру m внутри интервала.

6.5 Кинематическая реконструкция событий

В экспериментах по физике элементарных частиц часто можно применить процедуру, называемую «кинематической реконструкцией», которая позволяет улучшить энергетическое и угловое разрешение детектора. Формально процедуру кинематической реконструкции можно представить следующим образом.

Пусть в детекторе для данного события измерен набор экспериментальных величин x_i , $i = 1, 2, \dots, n$ (импульсы частиц, углы и т.п.). Для каждой из этих величин известна ошибка измерения или, в более общем случае, распределение вероятностей по отклонению от истинного значения. Обозначим истинные значения этих величин через α_i . Если попытаться оценить эти истинные значения методом максимального правдоподобия или наименьших квадратов, введя логарифмическую функцию правдоподобия или χ^2 :

$$\mathcal{L}(x_1, x_2, \dots, x_n; \alpha_1, \alpha_2, \dots, \alpha_n), \quad (6.45)$$

то в результате минимизации этой функции по параметрам α_i мы не получим ничего нового – решение будет $\alpha_i = x_i$. Ситуация меняется, если между этими истинными значениями существуют какие-либо связи, определяемые законами сохранения энергии-импульса или наличием промежуточных нестабильных частиц с известной массой. В общем виде эти связи можно ввести следующим образом:

$$F_j(\alpha_1, \alpha_2, \dots, \alpha_n) = 0, \quad j = 1, \dots, k \quad (6.46)$$

С учётом этих связей решение уже не будет таким тривиальным, кроме того, величина отклонений α_i от истинных значений $\bar{\alpha}_i$ будет меньше, чем отклонения исходных величин x_i . Как правило, аналитическая минимизация функции \mathcal{L} с учётом связей является трудно выполнимой или невыполнимой задачей.

Наиболее эффективным способом минимизации при наличии связей является исключение «лишних» переменных и минимизация по оставшимся переменным уже без ограничений. Однако, в общем случае эта задача не всегда выполнима.

Очевидно, что крайне низкая эффективность у такого простого способа: введём модифицированную функцию правдоподобия

$$\tilde{\mathcal{L}} = \mathcal{L} + \sum_{j=1}^k w_j^2 F_j^2(\alpha_1, \alpha_2, \dots, \alpha_n). \quad (6.47)$$

Если коэффициенты w_j достаточно велики, то эту модифицированную функцию можно минимизировать уже без ограничений. У этого метода, кроме неопределённого условия « w_j достаточно велики», есть серьёзный недостаток – профиль минимизируемой функции становится очень сложным и программы численной минимизации не смогут находить минимум.

6.5.1 Итерационный метод решения

В работе [62] предложен алгоритм итерационного решения задачи нахождения минимума функции, основанный на методе множителей Лагранжа (см., например, [12]). Вводится модифицированная функция правдоподобия

$$\hat{\mathcal{L}} = \mathcal{L} + \sum_{j=1}^k w_j F_j(\alpha_1, \alpha_2, \dots, \alpha_n). \quad (6.48)$$

Среди экстремумов функции $\hat{\mathcal{L}}$ в расширенном пространстве переменных $\alpha_1, \alpha_2, \dots, \alpha_n, w_1, w_2, \dots, w_k$ будет находиться и искомый минимум \mathcal{L} с ограничениями (6.46). Естественно, нельзя искать решение методами численной минимизации, потому что искомая точка в расширенном пространстве является именно экстремумом (ни максимумом, ни минимумом) функции $\hat{\mathcal{L}}$, и это осложняет нахождение решения этим методом, так как аналитическое решение задачи возможно в крайне редких случаях. Используя запись функции правдоподобия в виде квадратичной формы по отклонениям измеренных параметров от истинных и близость к линейной зависимости уравнений связи, в работе [62] выведена сходящаяся итерационная процедура нахождения решения. Правда, сходимость этой процедуры доказывается только при линейных уравнениях связи. Кроме того, наличие ошибок округления при вычислениях может приводить к смещению относительно истинной точки минимума. К тому же уравнения связи в этой процедуре на любом шаге итерации выполняются, вообще говоря, с каким-то приближением.

6.5.2 Использование численных методов условной минимизации

Как уже указывалось в главе 5, имеется обширная литература по методам условной оптимизации без предположения линейности уравнений связи [41, 42, 43, 44, 45, 45, 47]. Выбрав какой-либо из этих алгоритмов, можно достаточно эффективно решить задачу поиска минимума при наличии уравнений связи. Основной проблемой в этом направлении является скорость сходимости выбранного алгоритма и степень приближённости выполнения уравнений связи.

В то же время физические задачи кинематической реконструкции всегда допускают исключение переменных и сведение задачи условной минимизации к минимизации функции меньшего числа переменных без уравнений связи между ними.

6.5.3 Исключение «лишних» переменных

Физические задачи кинематической реконструкции всегда допускают уменьшение числа переменных с целью избавления от уравнений связи и сведение задачи условной минимизации к минимизации функции меньшего числа переменных без уравнений связи между ними. Основным недостатком этого подхода является то, что он строго индивидуален для

каждой задачи. Рассмотрим его на примере реконструкции события, рождённого на установке со встречными электрон-позитронными пучками:

$$\begin{aligned} e^+e^- &\rightarrow \eta'\gamma, \\ \eta' &\rightarrow \eta\pi^0\pi^0, \\ \eta &\rightarrow \gamma\gamma, \\ \pi^0 &\rightarrow \gamma\gamma. \end{aligned}$$

В конечном состоянии регистрируются 7 фотонов (энергии и углы) – 21 параметр. В качестве уравнений связи здесь выступают законы сохранения энергии-импульса для всей системы в целом, а также инвариантные массы пар фотонов, полученных от распада частиц с известной массой, например, $m_\eta \approx 550$ МэВ и $m_\pi \approx 135$ МэВ. Если не считать массу η' -мезона известной, а использовать получающуюся в результате кинематической реконструкции массу системы $m(\eta\pi\pi)$ для анализа распределения по этому параметру для оценки уровня фона, то всего уравнений связи набирается 7. Это значит, что после исключения «лишних» переменных должно остаться $21 - 7 = 14$ параметров. Рассмотрим схему параметризации события, обеспечивающую исключение «лишних» переменных.

Первым параметром можно выбрать массу системы $\eta\pi\pi$, обозначив её как M_x . Законы сохранения энергии-импульса позволяют вычислить энергию фотона отдачи $E_{\gamma 1}$ и импульс P_{s1} системы $\eta\pi\pi$ в системе центра масс сталкивающихся электрона и позитрона:

$$E_{\gamma 1} = P_{s1} = \frac{W^2 - M_x^2}{2W}. \quad (6.49)$$

Формула написана в «релятивистской» системе единиц, где принято, что скорость света $c = 1$. Здесь также введено обозначение W — суммарная энергия электрона и позитрона в системе центра масс.

Вторым оптимизируемым параметром можно выбрать массу $m_{\pi\pi}$ системы $\pi^0\pi^0$, которая меняется в пределах от $2m_\pi$ до $(M_x - m_\eta)$. Для того, чтобы упростить задачу для программы численной минимизации и избавиться от пределов изменения параметра, можно ввести параметр λ_1 , который однозначно определяет массу системы и не имеет ограничений. Масса $m_{\pi\pi}$ выражается через него следующим образом:

$$m_{\pi\pi} = \frac{M_x - m_\eta}{2} (1 + \sin \lambda_1) + m_\pi \cdot (1 - \sin \lambda_1). \quad (6.50)$$

Теперь можно вычислить импульс η -мезона и $\pi\pi$ -системы в системе покоя η' -мезона. Выражение для этого импульса p_η через параметр λ_1 выглядит

следующим образом:

$$\begin{aligned}
 p_\eta &= \frac{1}{2M_x} \sqrt{(M_x^2 + m_\eta^2 - m_{\pi\pi}^2)^2 - 4M_x^2 m_\eta^2} = \\
 &= \frac{|\sin(\frac{\lambda_1}{2} - \frac{\pi}{4})|}{2M_x} \sqrt{(M_x - m_\eta - 2m_\pi) \times} \\
 &\quad \times \sqrt{(M_x + m_\eta - m_{\pi\pi})(M_x + m_\eta + m_{\pi\pi})(M_x - m_\eta + m_{\pi\pi})}.
 \end{aligned} \tag{6.51}$$

Выберем направление движения η' -мезона вдоль оси z . Тогда фотон отдачи будет двигаться в отрицательном направлении вдоль оси z .

Пусть распад $M_x \rightarrow m_\eta + m_{\pi\pi}$ происходит в плоскости XZ . Теперь можно выбрать третий параметр – угол вылета θ_η в системе покоя η' -мезона, так что импульс η -мезона в этой системе записывается в виде вектора

$$\mathbf{p}_\eta = p_\eta \cdot (\sin \theta_\eta, 0, \cos \theta_\eta).$$

Четвёртый и пятый параметр – углы вылета $\theta_{\gamma 2}$ и $\varphi_{\gamma 2}$ вылета одного из фотонов распада η -мезона в сферической системе координат в системе покоя η -мезона. Другой фотон от распада η вылетает в противоположном направлении с тем же самым импульсом

$$p_{\gamma 2} = p_{\gamma 3} = \frac{m_\eta}{2}.$$

Шестой и седьмой параметр – углы вылета θ_π и φ_π одного из пионов в системе покоя $\pi^0 \pi^0$. Оба пиона в этой системе имеют одинаковые по величине импульсы

$$\begin{aligned}
 p_\pi &= \sqrt{\left(\frac{m_{\pi\pi}}{2}\right)^2 - m_\pi^2} = \\
 &= \frac{|\cos(\frac{\lambda_1}{2} - \frac{\pi}{4})|}{2} \sqrt{(M_x - m_\eta - 2m_\pi)(m_{\pi\pi} + 2m_\pi)}.
 \end{aligned} \tag{6.52}$$

Восьмой и девятый параметр – углы вылета $\theta_{\gamma 4}$ и $\varphi_{\gamma 4}$ одного из фотонов распада первого пиона в системе покоя этого пиона. Импульсы всех фотонов от распада пионов в системе покоя пиона равны $m_\pi/2$.

Десятый и одиннадцатый параметр – углы вылета $\theta_{\gamma 6}$ и $\varphi_{\gamma 6}$ одного из фотонов распада второго пиона в системе покоя этого пиона.

Набор этих одиннадцати параметров полностью определил вектора импульсов всех семи фотонов, правда, в разных системах отсчёта. Во всех случаях мы имеем такие характеристики системы отсчёта и распадной

частицы: масса «родительской» частицы M и её вектор импульса \mathbf{P} , масса «дочерней» частицы m и её вектор импульса \mathbf{p} в системе покоя «родительской» частицы. Используя эти данные, можно провести преобразование Лоренца из системы покоя «родительской» частицы в систему, в которой задан её импульс, по следующей формуле:

$$\mathbf{p}' = \mathbf{p} + \frac{\mathbf{P}}{M} \cdot \left[\sqrt{p^2 + m^2} + \frac{(\mathbf{p} \cdot \mathbf{P})}{M + \sqrt{P^2 + M^2}} \right]. \quad (6.53)$$

Поэтапно применяя это преобразование ко всем фотонам, мы получим вектора импульсов всех семи фотонов в системе центра масс e^+e^- .

Наконец, введя ещё три параметра – углы Эйлера поворота всей системы, как целого, мы получим энергии и углы всех фотонов в лабораторной системе координат, зависящие от 14 свободных параметров без уравнений связи, в то же время сами полученные импульсы и углы фотонов строго удовлетворяют уравнениям связи (6.46).

Примечание. Если система центра масс e^+e^- не совпадает с лабораторной системой, то все импульсы надо преобразованием Лоренца перевести в эту систему.

Мы успешно свели задачу минимизации функции от 21 параметра с семью уравнениями связи к задаче минимизации функции 14 независимых параметров. Задача стала намного проще, однако, численная минимизация функции 14 переменных — тоже непростая задача. Тем более, что предполагаемая кинематическая реконструкция проводится в автоматическом режиме, событие за событием, и у экспериментатора нет возможности изучать особенности минимизации в каждом событии. В таких условиях сокращение числа независимых переменных может привести не только к сокращению общего компьютерного времени, необходимого на кинематическую реконструкцию сотен или более событий, но и привести к качественному эффекту – улучшению разрешения по инвариантным массам и т.п., связанному с более качественной минимизацией. Такое сокращение возможно, если по каким-либо переменным можно найти минимум аналитически (при всех остальных переменных – фиксированных).

В данной задаче такая возможность возникает, если эксперимент проводится в системе центра масс e^+e^- , тогда энергии и импульсы конечных частиц не зависят от трёх углов поворота всей системы как целого. Кроме того, угловое разрешение детектора в разных проекциях

должно быть одинаково: $\sigma_\theta = \sigma_\varphi = \sigma_\alpha/\sqrt{2}$. Если к тому же это разрешение достаточно хорошее, чтобы можно было вклад в функцию правдоподобия от отклонения по углу каждой частицы от измеренного направления приближённо заменить на следующее выражение

$$\frac{\Delta\alpha_i^2}{2\sigma_{\alpha,i}^2} \approx \frac{1}{\sigma_{\alpha,i}^2} [1 - (\mathbf{e}_i \cdot \mathbf{n}_i)], \quad (6.54)$$

где введено обозначение для единичных векторов измеренного направления движения частицы \mathbf{e}_i и предполагаемого \mathbf{n}_i в нашей модели реконструкции, то минимум по углам поворота находится аналитически [63, 64].

Рассмотрим отдельно угловую часть функции правдоподобия:

$$L_A \approx \sum_i \frac{1 - \cos \Delta\alpha_i}{\sigma_i^2} = \sum_i \frac{1 - \mathbf{e}_i \cdot \mathbf{n}_i}{\sigma_i^2} = \sum_i \frac{1}{\sigma_i^2} - \sum_i \frac{\mathbf{e}_i \cdot \mathbf{n}_i}{\sigma_i^2}. \quad (6.55)$$

Здесь в виде σ_i обозначено разрешение в измерении пространственного угла i -ой частицы. Первая сумма в этом выражении не зависит от искомым углов поворота, а вторая сумма имеет экстремумы в тех же точках, что и L_A . Обозначим вторую сумму как

$$L_M = \sum_i \frac{\mathbf{e}_i \cdot \mathbf{n}_i}{\sigma_i^2}. \quad (6.56)$$

Когда эта сумма достигнет максимума, тогда функция L_A достигнет минимума. Пусть единичные вектора \mathbf{s}_i представляют направления движения частиц в нашей модели до общего поворота системы, тогда вектора \mathbf{n}_i могут быть представлены в виде

$$\mathbf{n}_i = \mathbf{T} \cdot \mathbf{s}_i \quad (6.57)$$

Теперь мы можем сформулировать задачу по-другому: необходимо найти матрицу поворота \mathbf{T} такую, что

$$L_M = \sum_i \frac{\mathbf{e}_i^T \mathbf{T} \mathbf{s}_i}{\sigma_i^2} = Tr \left(\mathbf{T} \sum_i \frac{\mathbf{s}_i \mathbf{e}_i^T}{\sigma_i^2} \right) = Tr(\mathbf{T} \mathbf{V}) \quad (6.58)$$

достигает максимального значения. Здесь посредством верхнего индекса « \mathbf{A}^T » обозначается транспонированная матрица, а $Tr(\mathbf{A})$ — функция квадратной матрицы, известной под названием «след матрицы», — сумма диагональных элементов. Здесь также введена матрица

$$\mathbf{V} = \sum_i \frac{\mathbf{s}_i \mathbf{e}_i^T}{\sigma_i^2}. \quad (6.59)$$

Пусть трёхмерные вектора $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ — столбцы матрицы \mathbf{V} , а единичные вектора $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ — строки матрицы поворота \mathbf{T} . Тогда функция, максимум которой надо найти, запишется в виде

$$L_M = \sum_{k=1}^3 \mathbf{u}_k \cdot \mathbf{v}_k \quad (6.60)$$

Для строк матрицы поворота справедливы следующие соотношения:

$$\mathbf{u}_1^2 = \mathbf{u}_2^2 = 1, \quad \mathbf{u}_1 \cdot \mathbf{u}_2 = 0, \quad \mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2. \quad (6.61)$$

Теперь задачу можно переформулировать следующим образом: для данных векторов $\mathbf{v}_k, k = 1, 2, 3$ найти вектора \mathbf{u}_1 и \mathbf{u}_2 , максимизирующие функцию L_M (6.60) при условиях (6.61).

Далее приведём решение из работы [64] для трёх вариантов, в зависимости от ранга матрицы \mathbf{V} . Для удобства введём обозначения:

$$V_{ij} = \mathbf{v}_i \cdot \mathbf{v}_j, \quad V_{123} = \mathbf{v}_1 \cdot [\mathbf{v}_2 \times \mathbf{v}_3] \quad (6.62)$$

Ненулевой детерминант.

Для случая ненулевого детерминанта $V_{123} \neq 0$ искомые вектора $\mathbf{u}_{1,2}$ могут быть записаны в следующем виде

$$\mathbf{u}_1 = x_1 \mathbf{v}_1 + x_2 \mathbf{v}_2 + x_3 \mathbf{v}_3, \quad \mathbf{u}_2 = x_7 \mathbf{v}_1 + x_8 \mathbf{v}_2 + x_9 \mathbf{v}_3, \quad (6.63)$$

где

$$\begin{aligned} x_1 &= \frac{2\lambda_3 V_{123} - V_{22} V_{33} + V_{23}^2}{z_1 V_{123}}, \\ x_2 &= \frac{V_{33} V_{12} - V_{13} V_{23} - \lambda_1 V_{123}}{z_1 V_{123}}, \\ x_3 &= \frac{V_{123} \cdot (2V_{13} \lambda_3 - V_{23} \lambda_1) + (V_{22} V_{13} - V_{12} V_{23})(z_1 - V_{33})}{z_1 V_{123} \cdot (z_1 - V_{33})}, \\ x_7 &= \frac{-\lambda_1 V_{123} + V_{12} V_{33} - V_{13} V_{23}}{z_1 V_{123}}, \\ x_8 &= \frac{2\lambda_2 V_{123} - V_{11} V_{33} + V_{13}^2}{z_1 V_{123}}, \\ x_9 &= \frac{V_{123} \cdot (2V_{23} \lambda_2 - V_{13} \lambda_1) + (V_{23} V_{11} - V_{12} V_{13})(z_1 - V_{33})}{z_1 V_{123} \cdot (z_1 - V_{33})}, \\ \lambda_1 &= \frac{2\lambda_2 [V_{12} \cdot (z_1 - V_{33}) + V_{13} V_{23}]}{V_{13}^2 + (V_{11} - z_1)(z_1 - V_{33})}, \quad \lambda_2 = \frac{\lambda_3 [(V_{11} - z_1)(z_1 - V_{33}) + V_{13}^2]}{V_{23}^2 + (V_{22} - z_1)(z_1 - V_{33})}, \\ \lambda_3 &= \frac{V_{123} \cdot [z_1^2 - (V_{22} + V_{33})z_1 + V_{22} V_{33} - V_{23}^2]}{z_1^3 + b \cdot z_1 / 2 + 2(V_{11} V_{22} V_{33} - V_{11} V_{23}^2 - V_{22} V_{13}^2 - V_{33} V_{12}^2) + 4V_{12} V_{13} V_{23}} \end{aligned} \quad (6.64)$$

и z_1 — корень уравнения

$$z_1^4 + bz_1^2 + cz_1 + d = 0, \quad (6.65)$$

где

$$\begin{aligned} b &= 2(V_{12}^2 + V_{13}^2 + V_{23}^2 - V_{11}V_{22} - V_{11}V_{33} - V_{22}V_{33}), \\ c &= 8V_{123}^2, \\ d &= (V_{22}^2 + V_{33}^2 + 2V_{23}^2)V_{11}^2 - 2(V_{11} + V_{22} + V_{33})V_{123}^2 - \\ &\quad - 2(V_{22}V_{12}^2 + V_{33}V_{13}^2 + 2V_{12}V_{13}V_{23})V_{11} + \\ &\quad + (V_{33}^2 + 2V_{13}^2)V_{22}^2 - 2(V_{33}V_{23} + 2V_{12}V_{13})V_{22}V_{23} + 2(V_{13}^2 + V_{23}^2)V_{12}^2 + \\ &\quad + 2V_{33}^2V_{12}^2 - 4V_{33}V_{12}V_{13}V_{23} + V_{12}^4 + V_{13}^4 + V_{23}^4 + 2V_{13}^2V_{23}^2. \end{aligned}$$

Обычно уравнение (6.65) имеет четыре действительных корня (их можно найти, используя формулы Феррари, например, из справочника [12]). Для всех действительных значений корня надо вычислять соответствующее значение функции L_M и выбрать вариант с максимальным L_M .

Нулевой детерминант: матрица \mathbf{V} ранга 2.

Может быть так, что $V_{123} \approx 0$, но имеются по крайней мере два неколлинеарных вектора \mathbf{v}_k . Для простоты будем считать, что это вектора \mathbf{v}_1 и \mathbf{v}_2 .

Введём вспомогательные переменные

$$\begin{aligned} D &= V_{11}V_{22} - V_{12}^2 > 0, \\ r_1 &= \frac{V_{22}V_{13} - V_{12}V_{23}}{D}, \\ r_2 &= \frac{V_{11}V_{23} - V_{12}V_{13}}{D}. \end{aligned} \quad (6.66)$$

Теперь решение можно представить в виде:

$$\begin{aligned} \mathbf{u}_1 &= x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + x_4[\mathbf{v}_1 \times \mathbf{v}_2], \\ \mathbf{u}_2 &= x_7\mathbf{v}_1 + x_8\mathbf{v}_2 + x_{10}[\mathbf{v}_1 \times \mathbf{v}_2], \\ \mathbf{u}_3 &= r_1\mathbf{v}_1 + r_2\mathbf{v}_2, \end{aligned} \quad (6.67)$$

где

$$\begin{aligned}
x_1 &= \frac{(V_{22}+2V_{12}\lambda_3)r_1r_2+r_1^2V_{12}+2(r_2^2V_{22}-z_1)\lambda_3}{(4\lambda_2\lambda_3-1)\lambda_1z_1}, \\
x_2 &= -\frac{(V_{12}+2V_{11}\lambda_3)r_1r_2+r_1^2V_{11}+2r_2^2V_{12}\lambda_3-z_1}{(4\lambda_2\lambda_3-1)\lambda_1z_1}, \\
x_7 &= -\frac{(V_{12}+2V_{22}\lambda_2)r_1r_2+r_2^2V_{22}+2r_1^2V_{12}\lambda_2-z_1}{(4\lambda_2\lambda_3-1)\lambda_1z_1}, \\
x_8 &= \frac{(V_{11}+2V_{12}\lambda_2)r_1r_2+r_2^2V_{12}+2(r_1^2V_{11}-z_1)\lambda_2}{(4\lambda_2\lambda_3-1)\lambda_1z_1}, \\
x_4 &= \frac{r_1}{z_1}, \\
x_{10} &= \frac{r_2}{z_1}, \\
z_1 &= \pm\sqrt{D\cdot(1+r_1^2+r_2^2)}, \\
\lambda_2 &= \frac{(V_{11}z_1-D)(1+r_1^2)+2r_1r_2V_{12}z_1+r_2^2(V_{22}z_1-2D)}{2(r_1r_2D+V_{12}z_1)}, \\
\lambda_1 &= \pm\sqrt{\frac{z_1-V_{33}}{1-4\lambda_2\lambda_3}}, \\
\lambda_3 &= \frac{(V_{22}z_1-D)(1+r_2^2)+2r_1r_2V_{12}z_1+r_1^2(V_{11}z_1-2D)}{2(r_1r_2D+V_{12}z_1)}.
\end{aligned} \tag{6.68}$$

Нулевой детерминант: матрица \mathbf{V} ранга 1.

В этом случае все вектора \mathbf{v}_k пропорциональны одному вектору \mathbf{p} :

$$\mathbf{v}_1 = a \cdot \mathbf{p}, \quad \mathbf{v}_2 = b \cdot \mathbf{p}, \quad \mathbf{v}_3 = c \cdot \mathbf{p}, \quad \mathbf{p}^2 = 1. \tag{6.69}$$

Максимальное значение L_M в этом случае равно

$$f_M = \max L_M = \max \text{Tr}(\mathbf{T}\mathbf{V}) = \sqrt{a^2 + b^2 + c^2} \tag{6.70}$$

Матрица поворота \mathbf{T} при этом неоднозначна. Одно из возможных решений следующее:

$$\mathbf{T} = \mathbf{T}_2\mathbf{T}_1, \quad \text{где } \mathbf{T}_2 = \begin{pmatrix} 0 & \frac{\sqrt{b^2+c^2}}{f_M} & \frac{a}{f_M} \\ -\frac{c}{\sqrt{b^2+c^2}} & -\frac{ab}{\sqrt{b^2+c^2}f_M} & \frac{b}{f_M} \\ \frac{b}{\sqrt{b^2+c^2}} & -\frac{ac}{\sqrt{b^2+c^2}f_M} & \frac{c}{f_M} \end{pmatrix}, \tag{6.71}$$

третья строка матрицы \mathbf{T}_1 равна вектору \mathbf{p} . Первые две строки матрицы \mathbf{T}_1 — два единичных вектора, которые образуют с \mathbf{p} правую тройку векторов.

Матрица \mathbf{V} не может быть точно равна нулю, поэтому приведённые три варианта решения охватывают все возможные случаи.

Рассмотренные способы кинематической реконструкции применимы к событиям с любым количеством частиц и к любому набору измеренных экспериментальных величин. Это касается и случаев с «потерянными» частицами при наличии в детекторе нечувствительных областей, и случаев с неполной регистрацией характеристик частиц, например, когда углы вылета измеряются, а энергии — нет, или измеряется только сумма энергий для какой-нибудь пары частиц и т.п. Наличие уравнений связи позволяет в этих случаях уточнить или восстановить недостающие характеристики частиц. Для немагнитных детекторов типичным примером кинематической реконструкции может служить вычисление энергий частиц в трёх- и четырёх-частичных событиях по измеренным углам этих частиц [65].

6.5.4 Реконструкция трёх-частичных событий

Рассмотрим случай, когда в детекторе зарегистрированы только углы частиц, и известны суммарная энергия и импульс системы. Кроме того, ограничим исследование случаем, когда полный импульс системы равен нулю (симметричные установки со встречными пучками). Для асимметричных установок со встречными пучками качественно все выводы остаются в силе, но все формулы усложняются, так как Лоренц-преобразования импульсов частиц из системы центра масс в лабораторную систему зависят от самих импульсов, которые в нашем случае не измерены и являются свободными переменными.

Пусть массы частиц известны в рамках теоретической модели и равны m_1 , m_2 , m_3 (если массы неизвестны, то надо перебирать несколько теоретических моделей процесса). Полную энергию системы обозначим как W . В силу равенства нулю импульса системы, импульсы всех трёх частиц p_1 , p_2 , p_3 должны лежать в одной плоскости. Пусть в этой плоскости углы между частицами равны, соответственно, α_{12} , α_{13} и α_{23} . Естественно, эти три угла удовлетворяют соотношению

$$\alpha_{12} + \alpha_{13} + \alpha_{23} = 2\pi. \quad (6.72)$$

Применяя закон сохранения импульса в плоскости рождения частиц, получим

$$\begin{cases} p_1 + p_2 \cos \alpha_{12} + p_3 \cos \alpha_{13} = 0, \\ p_2 \sin \alpha_{12} = p_3 \sin \alpha_{13}. \end{cases} \quad (6.73)$$

Отсюда нетрудно получить соотношения:

$$\frac{p_1}{\sin \alpha_{23}} = \frac{p_2}{\sin \alpha_{13}} = \frac{p_3}{\sin \alpha_{12}} = x, \quad (6.74)$$

где введена вспомогательная переменная x . С учётом этих соотношений уравнение сохранения энергии записывается в виде

$$\sqrt{m_1^2 + x^2 \cdot \sin^2 \alpha_{23}} + \sqrt{m_2^2 + x^2 \cdot \sin^2 \alpha_{13}} + \sqrt{m_3^2 + x^2 \cdot \sin^2 \alpha_{12}} = W. \quad (6.75)$$

Очевидно, что это уравнение имеет ровно два решения на действительной оси, различающиеся знаком. В принципе, зная, что корень обязательно есть и единственный, уравнение можно решать любым численным методом, например, методом деления интервала пополам. С другой стороны, формально это уравнение является алгебраическим уравнением восьмой степени, если последовательно избавиться от всех квадратных корней:

$$a_4 x^8 + a_3 x^6 + a_2 x^4 + a_1 x + a_0 = 0, \quad (6.76)$$

где

$$a_0 = \left[W^2 - (m_1 + m_2 + m_3)^2 \right] \cdot \left[W^2 - (m_1 + m_2 - m_3)^2 \right] \times \\ \times \left[W^2 - (m_1 - m_2 + m_3)^2 \right] \cdot \left[W^2 - (m_1 - m_2 - m_3)^2 \right],$$

$$\begin{aligned}
a_1 = & -8 \cdot [(((2(m_2^2 + m_3^2) \cdot m_1^2 - 3m_1^4 + m_2^4 - 10m_2^2m_3^2 + m_3^4) W^2 + \\
& + (3m_1^2 + m_2^2 + m_3^2) W^4 + (3m_2^4 + 2m_2^2m_3^2 + 3m_3^4) m_1^2 - \\
& - 3(m_2^2 + m_3^2) m_1^4 - W^6 + m_1^6 - m_2^6 + m_2^4m_3^2 + m_2^2m_3^4 - m_3^6) \sin^2 \alpha_{12} - \\
& - (2(m_2^2 - 2m_3^2) m_1^2 - m_1^4 - m_2^4 - 4m_2^2m_3^2 + m_3^4) W^2 - \\
& - (2m_1^2 + 2m_2^2 + m_3^2) W^4 - 2(m_2^2 + m_3^2) m_1^2m_3^2 + W^6 + (m_1^4 + m_2^4) m_3^2 - \\
& - 2m_2^2m_3^4 + m_3^6) \sin^2 \alpha_{13} + \\
& + ((2(2m_2^2 - m_3^2) m_1^2 + m_1^4 - m_2^4 + 4m_2^2m_3^2 + m_3^4) W^2 - \\
& - (2m_1^2 + m_2^2 + 2m_3^2) W^4 - 2(m_2^2 + m_3^2) m_1^2m_2^2 + W^6 + m_1^4m_2^2 + \\
& + m_2^6 - 2m_2^4m_3^2 + m_2^2m_3^4) \sin^2 \alpha_{12}) - \\
& - (((m_2^2 + m_3^2) m_1^2 - 3m_1^4 + m_2^4 - 10m_2^2m_3^2 + m_3^4) W^2 + \\
& + (3m_1^2 + m_2^2 + m_3^2) W^4 + (3m_2^4 + 2m_2^2m_3^2 + 3m_3^4) m_1^2 - \\
& - 3(m_2^2 + m_3^2) m_1^4 - W^6 + m_1^6 - m_2^6 + m_2^4m_3^2 + m_2^2m_3^4 - m_3^6) \times \\
& \times \sin \alpha_{13} \sin \alpha_{12} \cos \alpha_{13} \cos \alpha_{12}],
\end{aligned}$$

$$\begin{aligned}
a_2 = & -8 \cdot \left[(2 (2 (3m_1^2 - m_2^2 - m_3^2) W^2 + 6 (m_2^2 + m_3^2) m_1^2 - 3W^4 - 3m_1^4 - 3m_2^4 - \right. \\
& - 2m_2^2 m_3^2 - 3m_3^4) \sin^4 \alpha_{12} + (7W^4 - 10W^2 m_1^2 + 6W^2 m_2^2 - 6W^2 m_3^2 + 3m_1^4 - \\
& - 6m_1^2 m_2^2 - 10m_1^2 m_3^2 + 3m_2^4 + 6m_2^2 m_3^2 + 7m_3^4) \sin^2 \alpha_{12} - 2W^4 + 4W^2 m_3^2 - \\
& - 2m_3^4) \sin^4 \alpha_{13} - 2 ((2 (3m_1^2 - m_2^2 - m_3^2) W^2 + 6 (m_2^2 + m_3^2) m_1^2 - 3W^4 - \\
& - 3m_1^4 - 3m_2^4 - 2m_2^2 m_3^2 - 3m_3^4) \sin^2 \alpha_{12} - 2 ((m_1^2 - m_2^2 + 2m_3^2) W^2 - W^4 + \\
& + m_1^2 m_3^2 - m_2^2 m_3^2 - m_3^4)) \sin^3 \alpha_{13} \sin \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} + (2 (2 (m_2^2 + m_3^2) m_1^2 - \\
& - 3W^4 + 6W^2 m_1^2 - m_1^4 - m_2^4 - 4m_2^2 m_3^2 - m_3^4) + (7W^4 - 10W^2 m_1^2 - 6W^2 m_2^2 + \\
& + 6W^2 m_3^2 + 3m_1^4 - 10m_1^2 m_2^2 - 6m_1^2 m_3^2 + 7m_2^4 + 6m_2^2 m_3^2 + 3m_3^4) \sin^2 \alpha_{12}) \times \\
& \times \sin^2 \alpha_{13} \sin^2 \alpha_{12} + 4 ((m_1^2 + 2m_2^2 - m_3^2) W^2 - W^4 + m_1^2 m_2^2 - m_2^4 - m_2^2 m_3^2) \times \\
& \left. \times \sin \alpha_{13} \sin^3 \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} - 2 (W + m_2)^2 (W - m_2)^2 \sin^4 \alpha_{12} \right],
\end{aligned}$$

$$\begin{aligned}
a_3 = & 32 (((8W^2 - 2m_1^2 + 3m_2^2 + 3m_3^2) \sin^2 \alpha_{12} - \\
& - (5W^2 - 3m_1^2 + 5m_2^2 + 3m_3^2) \sin^4 \alpha_{12} - 2W^2) \sin^2 \alpha_{13} - \\
& - ((5W^2 - 3m_1^2 + 3m_2^2 + 5m_3^2) \sin^2 \alpha_{12} - 4 (W^2 - m_1^2 + m_2^2 + m_3^2) \sin^4 \alpha_{12} - \\
& - (W^2 + m_3^2)) \sin^4 \alpha_{13} + ((3W^2 - m_1^2 + 3m_2^2 + m_3^2) \sin^2 \alpha_{12} - 4W^2) \times \\
& \times \sin \alpha_{13} \sin \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} + \\
& + ((3W^2 - m_1^2 + m_2^2 + 3m_3^2) - 4 (W^2 - m_1^2 + m_2^2 + m_3^2) \sin^2 \alpha_{12}) \times \\
& \times \sin^3 \alpha_{13} \sin \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} + \\
& + ((W^2 + m_2^2) \sin^2 \alpha_{12} - 2W^2) \sin^2 \alpha_{12}) \sin^2 \alpha_{13} \sin^2 \alpha_{12},
\end{aligned}$$

$$\begin{aligned}
a_4 = & 16 ((8 \sin^4 \alpha_{12} - 8 \sin^2 \alpha_{12} + 1) \sin^4 \alpha_{13} - 2 (4 \sin^2 \alpha_{12} - 3) \sin^2 \alpha_{13} \sin^2 \alpha_{12} - \\
& - 4 (2 \sin^2 \alpha_{12} - 1) \sin^3 \alpha_{13} \sin \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} + \\
& + 4 \sin \alpha_{13} \sin^3 \alpha_{12} \cos \alpha_{13} \cos \alpha_{12} \sin^4 \alpha_{12}) \sin^4 \alpha_{13} \sin^4 \alpha_{12}.
\end{aligned}$$

Однако, коэффициенты при нечётных степенях x все равны нулю, поэтому уравнение имеет степень 4 по переменной x^2 . Используя схему решения Феррари [12], уравнение четвёртой степени можно решить, используя

аналитические формулы, что должно несколько быстрее приводить к успеху. Учитывая громоздкость формул для коэффициентов полинома a_i и большую вероятность опечатки при кодировании программ, приведём контрольный пример, который можно использовать при отладке.

Возьмём

$$\begin{aligned} m_2 &= 2m_1, & m_3 &= 3m_1, & W &= 10m_1, \\ \alpha_{12} &= \frac{2\pi}{3}, & \alpha_{13} &= \frac{3\pi}{4}, & \alpha_{23} &= \frac{7\pi}{12}. \end{aligned} \quad (6.77)$$

Тогда коэффициенты полинома должны быть равны

$$\begin{aligned} a_0 &= 51609600m_1^8, \\ a_1 &= -96000(56 + 9\sqrt{3})m_1^6 \approx -6872492m_1^6, \\ a_2 &= 4(25813 + 5786\sqrt{3})m_1^4 \approx 143338.6m_1^4, \\ a_3 &= -\frac{3}{2}(354 + 163\sqrt{3})m_1^2 \approx -954.4864m_1^2, \\ a_4 &= \frac{9}{64}(7 + 4\sqrt{3}) \approx 1.958654. \end{aligned} \quad (6.78)$$

В соответствии с алгоритмом Феррари надо решить вспомогательное кубическое уравнение (нужен один его корень):

$$y^3 + b_2y^2 + b_1y + b_0 = 0, \quad (6.79)$$

где

$$\begin{aligned} b_2 &= -\frac{a_2}{a_4} \approx -73182.20m_1^4, \\ b_1 &= \frac{a_3a_1}{a_4^2} - \frac{4a_0}{a_4} \approx 1604494079m_1^8, \\ b_0 &= \frac{4a_2a_0 - a_1^2}{a_4^2} - \frac{a_3^2a_0}{a_4^3} \approx -10855743156294m_1^{12}. \end{aligned}$$

У этого уравнения в нашем случае все коэффициенты действительные, поэтому должен быть хотя бы один действительный корень. В соответствии с алгоритмом Кардано [12] приводим уравнение к неполному виду заменой $y = z - b_2/3$:

$$z^3 + p \cdot z + q = 0, \quad (6.80)$$

где

$$p = b_1 - \frac{b_2^2}{3} \approx -180717534m_1^8,$$

$$q = 2 \left(\frac{b_2}{3} \right)^3 - \frac{b_2 b_1}{3} + b_0 \approx -747987935368 m_1^{12}.$$

Теперь три корня могут быть вычислены (в комплексном виде) по следующей формуле:

$$y_k = A_k - \frac{p}{3A_k} + \frac{b_2}{3}, \quad (6.81)$$

где

$$A_k = e^{\frac{2\pi k i}{3}} \cdot \sqrt[3]{\sqrt{\left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2} - \frac{q}{2}}, \quad k = 0, 1, 2.$$

В нашем примере корни вспомогательного уравнения третьей степени равны:

$$y_0 \approx 39560.96 m_1^4, \quad y_1 \approx 13948.54 m_1^4, \quad y_2 \approx 19672.70 m_1^4. \quad (6.82)$$

Для наших целей подходит любой из этих корней. Возьмём, например, $y = y_1$. Теперь с помощью этого корня можно исходное уравнение четвёртой степени разложить на два квадратных уравнения:

$$x^4 + \frac{a_3}{2a_4} x^2 + \frac{y_1}{2} = \pm \sqrt{\rho_2 x^4 + \rho_1 x^2 + \rho_0}, \quad (6.83)$$

где

$$\begin{aligned} \rho_2 &= \frac{a_3^2}{4a_4^2} - \frac{a_2}{a_4} + y_1 \approx (11.65985 m_1^2)^2, \\ \rho_1 &= \frac{a_3}{2a_4} y_1 - \frac{a_1}{a_4} \approx 2 \cdot (11.65985 m_1^2) \cdot (4721.323 m_1^4), \\ \rho_0 &= \frac{y_1^2}{4} - \frac{a_0}{a_4} \approx (4721.323 m_1^4)^2. \end{aligned}$$

Такой и есть замысел Феррари — чтобы под корнем квадратным в (6.83) оказался полный квадрат, тогда корень извлекается и уравнение превращается в

$$x^4 + (-243.6588 \pm 11.65985) m_1^2 x^2 + (6974.268 \pm 4721.323) m_1^4 = 0. \quad (6.84)$$

Отсюда получаем полный набор из четырёх корней:

$$\begin{aligned} x_1 &\approx 12.56803 m_1, & x_2 &\approx 8.60486 m_1, \\ x_3 &\approx 15.6897 m_1, & x_4 &\approx 3.025247 m_1. \end{aligned} \quad (6.85)$$

Проверка показала, что все эти корни удовлетворяют уравнению (6.76), но для уравнения (6.75) первые три из них оказались ложные, а корень x_4 хорошо подошёл. Этот пример показывает, что получая решения алгебраического уравнения, мы должны все действительные корни проверять, удовлетворяют ли они уравнению (6.75).

Все эти сложности с решением уравнения исчезают, если все три массы нулевые. В этом случае решение очень простое:

$$x = \frac{W}{\sin \alpha_{12} + \sin \alpha_{13} + \sin \alpha_{23}}. \quad (6.86)$$

Теперь вернёмся в начало данного раздела и рассмотрим условие, что импульсы всех частиц должны лежать в одной плоскости. Если углы частиц измеряются в эксперименте, то такая ситуация никогда не реализуется. В рамках общего подхода с минимизацией логарифмической функции правдоподобия в тех случаях, когда справедливо приближение (6.54), можно выписать результат проецирования экспериментальных треков на плоскость. Пусть \mathbf{n}_i — искомые единичные вектора, \mathbf{e}_i — экспериментально зарегистрированные направления движения трёх частиц. Оптимальный вариант соответствует случаю, когда достигается максимум функции

$$\sum_{i=1}^3 \frac{(\mathbf{n}_i \cdot \mathbf{e}_i)}{\sigma_{\alpha,i}^2}.$$

Степень компланарности векторов \mathbf{e}_i можно характеризовать величиной

$$\Delta_0 = (\mathbf{e}_1 \cdot [\mathbf{e}_2 \times \mathbf{e}_3]),$$

т.е. смешанным произведением трёх векторов. Решение в случае $\Delta_0 = 0$ тривиально: $\mathbf{n}_i = \mathbf{e}_i$. Если ввести обозначения $w_i = 1/\sigma_{\alpha,i}^2$, $\cos \alpha_{12} = \mathbf{e}_1 \cdot \mathbf{e}_2$, $\cos \alpha_{13} = \mathbf{e}_1 \cdot \mathbf{e}_3$, $\cos \alpha_{23} = \mathbf{e}_2 \cdot \mathbf{e}_3$, то в приближении $\Delta_0^2 \ll 1$ можно записать решение в следующем виде. Пусть нормаль к искомой плоскости векторов \mathbf{n}_i представлена в виде

$$\mathbf{N} = z_1 \cdot [\mathbf{e}_2 \times \mathbf{e}_3] + z_2 \cdot [\mathbf{e}_3 \times \mathbf{e}_1] + z_3 \cdot [\mathbf{e}_1 \times \mathbf{e}_2]. \quad (6.87)$$

Тогда коэффициенты z_i могут быть представлены в виде

$$\begin{aligned}
z_1 &= \frac{w_2 w_3}{2A_0} \cdot [2w_1 w_2 \cdot (1 - \cos^2 \alpha_{12}) (\cos \alpha_{12} \cos \alpha_{23} - \cos \alpha_{13}) + \\
&\quad + w_1 w_3 (2 \cos \alpha_{12} \cos \alpha_{23} - \cos^2 \alpha_{12} \cos \alpha_{13} + \cos^3 \alpha_{13} - \\
&\quad \quad \quad - \cos \alpha_{13} \cos^2 \alpha_{23} - \cos \alpha_{13}) + \\
&\quad + 2w_2 w_3 (1 - \cos^2 \alpha_{23}) (\cos \alpha_{12} \cos \alpha_{23} - \cos \alpha_{13})], \\
z_2 &= \frac{w_1 w_3}{2A_0} \cdot [2w_1 w_2 \cdot (1 - \cos^2 \alpha_{12}) (\cos \alpha_{12} \cos \alpha_{13} - \cos \alpha_{23}) + \\
&\quad + 2w_1 w_3 \cdot (1 - \cos^2 \alpha_{13}) (\cos \alpha_{12} \cos \alpha_{13} - \cos \alpha_{23}) + \\
&\quad + w_2 w_3 (2 \cos \alpha_{12} \cos \alpha_{13} - \cos^2 \alpha_{12} \cos \alpha_{23} + \cos^3 \alpha_{23} - \\
&\quad \quad \quad - \cos \alpha_{23} \cos^2 \alpha_{13} - \cos \alpha_{23})], \\
z_3 &= \frac{w_1 w_2}{A_0} \cdot [w_1 w_2 \cdot (1 - \cos^2 \alpha_{12}) + \\
&\quad + w_1 w_3 \cdot (1 - \cos^2 \alpha_{12}) (1 - \cos^2 \alpha_{13}) + \\
&\quad + w_2 w_3 \cdot (1 - \cos^2 \alpha_{12}) (1 - \cos^2 \alpha_{23})].
\end{aligned} \tag{6.88}$$

Нормировочная константа A_0 может быть найдена из условия равенства единице модуля вектора нормали:

$$\begin{aligned}
|\mathbf{N}|^2 = 1 = & z_1^2 \cdot (1 - \cos^2 \alpha_{23}) + z_2^2 \cdot (1 - \cos^2 \alpha_{13}) + z_3^2 \cdot (1 - \cos^2 \alpha_{12}) + \\
& + 2z_1 z_2 \cdot (\cos \alpha_{13} \cos \alpha_{23} - \cos \alpha_{12}) + \\
& + 2z_1 z_3 \cdot (\cos \alpha_{12} \cos \alpha_{23} - \cos \alpha_{13}) + \\
& + 2z_2 z_3 \cdot (\cos \alpha_{12} \cos \alpha_{13} - \cos \alpha_{23}).
\end{aligned} \tag{6.89}$$

Единичные вектора \mathbf{n}_i можно теперь записать в виде

$$\mathbf{n}_i = \frac{\mathbf{e}_i - z_i \Delta_0 \cdot \mathbf{N}}{\sqrt{1 - z_i^2 \Delta_0^2}}. \tag{6.90}$$

Соответственно, теперь вместо экспериментально измеренных углов α_{ij} можно оценить проекции этих углов α_{ij}^* на предполагаемую плоскость разлёта всех трёх частиц:

$$\begin{aligned}
\cos \alpha_{12}^* &= \left(1 + \frac{z_1^2 + z_2^2}{2} \Delta_0^2\right) \cos \alpha_{12} - z_1 z_2 \Delta_0^2, \\
\cos \alpha_{13}^* &= \left(1 + \frac{z_1^2 + z_3^2}{2} \Delta_0^2\right) \cos \alpha_{13} - z_1 z_3 \Delta_0^2, \\
\cos \alpha_{23}^* &= \left(1 + \frac{z_2^2 + z_3^2}{2} \Delta_0^2\right) \cos \alpha_{23} - z_2 z_3 \Delta_0^2.
\end{aligned} \tag{6.91}$$

Естественно, получив улучшенные оценки углов между частицами, мы должны использовать их в законах сохранения энергии-импульса (6.73–6.75).

6.5.5 Реконструкция четырёх-частичных событий

В том случае, если измерены углы вылета четырёх частиц и известны полная энергия системы и импульс, а также массы частиц, то можно определить импульсы частиц p_i аналогичными вычислениями. Полный набор уравнений для переменных p_i следующий:

$$\begin{aligned} \sum_{i=1}^4 p_i \sin \theta_i \cos \varphi_i &= 0, \\ \sum_{i=1}^4 p_i \sin \theta_i \sin \varphi_i &= 0, \\ \sum_{i=1}^4 p_i \cos \theta_i &= 0, \\ \sum_{i=1}^4 \sqrt{p_i^2 + m_i^2} &= W. \end{aligned} \tag{6.92}$$

Если обозначить единичные вектора вдоль направления импульсов через

$$\mathbf{e}_i = (\sin \theta_i \cos \varphi_i, \sin \theta_i \sin \varphi_i, \cos \theta_i),$$

и ввести обозначения для смешанных произведений векторов

$$\begin{aligned} \Delta_1 &= (\mathbf{e}_2 \cdot [\mathbf{e}_3 \times \mathbf{e}_4]), & \Delta_2 &= -(\mathbf{e}_1 \cdot [\mathbf{e}_3 \times \mathbf{e}_4]), \\ \Delta_3 &= (\mathbf{e}_1 \cdot [\mathbf{e}_2 \times \mathbf{e}_4]), & \Delta_4 &= -(\mathbf{e}_1 \cdot [\mathbf{e}_2 \times \mathbf{e}_3]), \end{aligned} \tag{6.93}$$

то система из первых трёх уравнений имеет общее решение следующего вида:

$$p_1 = z \cdot \Delta_1, \quad p_2 = z \cdot \Delta_2, \quad p_3 = z \cdot \Delta_3, \quad p_4 = z \cdot \Delta_4. \tag{6.94}$$

Величина параметра z выбирается такой, чтобы удовлетворялось уравнение сохранения энергии, а его знак выбирается таким образом, чтобы все импульсы имели положительный знак. Если последнее невозможно, то решение не существует (например, когда все импульсы направлены в одну полусферу).

Решать уравнение

$$\sum_{i=1}^4 \sqrt{m_i^2 + z^2 \Delta_i^2} = W \tag{6.95}$$

можно численными методами, например, методом деления пополам.

Если избавиться от квадратных корней, то получается уравнение восьмой степени относительно переменной $\mu = z^2$. Прямых формул для решения таких уравнений не существует, поэтому самым перспективным

направлением ускорения решения представляется ускорение процедуры сходимости при решении уравнения (6.95).

Chapter 7

Доверительные интервалы. Установление верхнего предела

В отличие от теории вероятностей, статистика является более сложной и неоднозначной наукой. Проиллюстрируем это обсуждением некоторых проблем с доверительными интервалами и верхними пределами.

Причины сложности построения доверительных интервалов в том, что распределение вероятностей по оценкам параметра зависит от истинного значения параметра, которого мы не знаем. Один из способов приближённой оценки доверительных интервалов – методом максимального правдоподобия, – был подробно рассмотрен в разд. 6.1. Однако существует точный метод построения доверительных интервалов, предложенный Нейманом [61]. Алгоритм Неймана рассмотрим позже на примере оценки времени жизни в разд. 7.2. Далее на нескольких примерах рассмотрим, к каким ошибкам в уровне доверия может приводить метод максимального правдоподобия и насколько отличаются доверительные интервалы, полученные по функции правдоподобия и классическим методом Неймана. Более детальное обсуждение проблем, возникающих при определении доверительных интервалов, можно найти в работах [66, 67].

7.1 Определение среднего для распределения Гаусса

Это пример «хорошего» распределения вероятностей по оценке параметра. Пусть мы определяем неизвестное среднее значение a Гауссова распределения с известной дисперсией σ^2 по ограниченной выборке n событий. Логарифмическая функция правдоподобия записывается в виде

$$L = n \cdot \ln \left(\sigma \sqrt{2\pi} \right) + \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - a)^2. \quad (7.1)$$

Легко показать, что зависимость от параметра a функции правдоподобия имеет вид параболы с минимумом в точке

$$\hat{a} = \frac{1}{n} \sum_{i=1}^n x_i \quad (7.2)$$

и расстоянием от точки минимума до точек пересечения параболы с горизонтальной линией на уровне $(L_{min} + 0,5)$:

$$\sigma_a = \pm \frac{\sigma}{\sqrt{n}}. \quad (7.3)$$

С другой стороны, легко убедиться, что распределение вероятностей по \hat{a} является гауссовым с центром в точке a и среднеквадратичным отклонением σ_a . В данном случае 68%-й доверительный интервал

$$\hat{a} = \frac{1}{n} \sum_{i=1}^n x_i \pm \frac{\sigma}{\sqrt{n}} \quad (7.4)$$

в точности соответствует утверждению, что истинное значение параметра в 68% случаев попадает внутрь доверительного интервала, полученного по функции правдоподобия. Рассмотрим примеры, когда это утверждение нарушается.

7.2 Среднее время жизни

Требуется определить среднее время τ жизни системы (например, нестабильной частицы), используя n измерений t_i . Распределение

вероятностей по времени распада даётся экспоненциальным распределением:

$$\frac{dW}{dt} = \frac{1}{\tau} \exp\left(-\frac{t}{\tau}\right). \quad (7.5)$$

Логарифмическая функция правдоподобия в этом случае равна

$$L = n \cdot \ln \tau + \frac{1}{\tau} \sum_{i=1}^n t_i. \quad (7.6)$$

Минимум функции L_0 достигается при

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^n t_i; \quad \min L = L_0 = n + n \cdot \ln \hat{\tau}. \quad (7.7)$$

Эта оценка, очевидно, является несмещённой при любом числе измерений n . Легко заметить, что статистические свойства оценки параметра и доверительных интервалов здесь не зависят от самой величины τ (можно все времена измерять в единицах τ), но зависят от числа измерений n . Границы доверительного интервала могут быть определены только численными методами. В табл. 7.1 приведены границы доверительных интервалов для нескольких значений n и для уровня доверия 68% ($\Delta L = 1/2$) и 95% ($\Delta L = 2$). В пределе больших n доверительный интервал становится симметричным с полуразмером

$$\Delta\tau = \hat{\tau} \cdot \sqrt{\frac{2\Delta L}{n}}. \quad (7.8)$$

Из таблицы видно, что даже при $n = 50$ доверительные интервалы ещё не становятся симметричными. На рис. 7.1 приведены графики функции правдоподобия для нескольких значений n . Для малых n график функции правдоподобия заметно несимметричен. Посмотрим, насколько соответствует реальная вероятность «покрытия» истинного значения доверительным интервалом ожидаемой вероятности (рис. 7.2). Здесь наглядно проявляется справедливость утверждения, что метод максимального правдоподобия, строго говоря, даёт правильные результаты только в пределе большой статистики. Рассматриваемая задача с измерением времени жизни является хорошим объектом для демонстрации классического подхода для точного определения доверительных интервалов [19], развитого Нейманом [61].

Table 7.1: Границы $\Delta\tau^{(\pm)} = |\tau - \hat{\tau}|$ доверительных интервалов для уровня доверия 68% ($\Delta L = L - L_0 = 1/2$) и 95% ($\Delta L = 2$), определённых по методу максимального правдоподобия.

n	$\Delta L = \frac{1}{2}$			$\Delta L = 2$		
	$\frac{\Delta\tau^{(-)}}{\hat{\tau}}$	$\frac{\Delta\tau^{(+)}}{\hat{\tau}}$	$\sqrt{\frac{2\Delta L}{n}}$	$\frac{\Delta\tau^{(-)}}{\hat{\tau}}$	$\frac{\Delta\tau^{(+)}}{\hat{\tau}}$	$\sqrt{\frac{2\Delta L}{n}}$
1	0,5759	2,3144	1,0000	0,7780	18,0588	2,0000
2	0,4689	1,2283	0,7071	0,6822	5,3054	1,4142
3	0,4095	0,8940	0,5774	0,6206	3,1642	1,1547
4	0,3697	0,7253	0,5000	0,5759	2,3144	1,0000
5	0,3405	0,6212	0,4472	0,5410	1,8579	0,8944
6	0,3177	0,5496	0,4082	0,5127	1,5714	0,8165
7	0,2992	0,4968	0,3780	0,4891	1,3736	0,7559
8	0,2839	0,4560	0,3536	0,4689	1,2283	0,7071
9	0,2708	0,4232	0,3333	0,4513	1,1164	0,6667
10	0,2595	0,3963	0,3162	0,4357	1,0274	0,6325
20	0,1939	0,2614	0,2236	0,3405	0,6212	0,4472
50	0,1290	0,1558	0,1414	0,2367	0,3456	0,2828

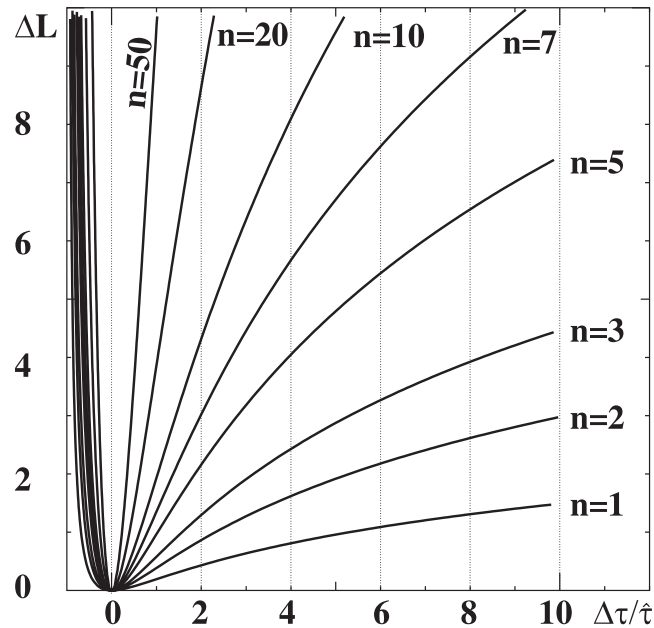


Figure 7.1: График функции $\Delta L = L - L_0$ в зависимости от $\Delta\tau/\hat{\tau} = (\tau - \hat{\tau})/\hat{\tau}$ при нескольких значениях n

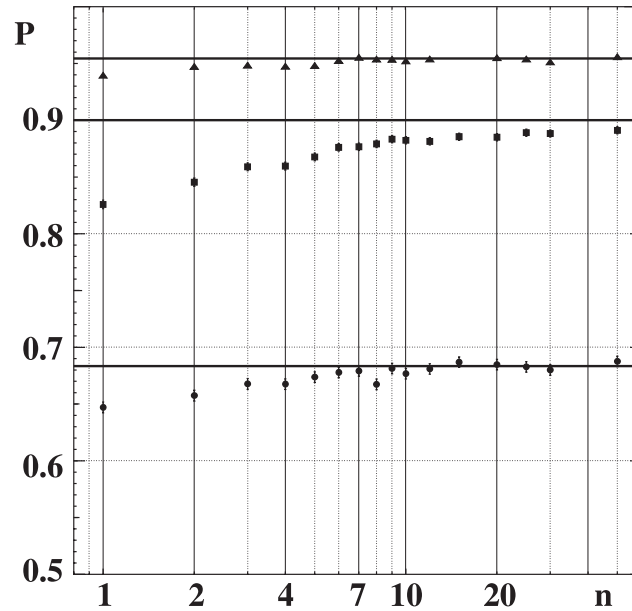


Figure 7.2: Вероятность попадания истинного значения времени жизни внутрь доверительного интервала. Обозначения: ● – $\Delta L = 1/2$, ▲ – $\Delta L = 2$, ■ – $\Delta L = 0,8221$ (полубесконечный интервал, 90%-й верхний предел). Горизонтальные сплошные линии показывают «правильное» значение вероятности (0,6827, 0,9545 и 0,9)

Классический подход для точного определения доверительных интервалов. Описываемый далее метод справедлив для любых оценок параметров, не только для тех, что получены методом максимального правдоподобия (даже для смещённых оценок – в некоторых случаях может оказаться, что доверительный интервал не включает полученную оценку параметра). Возьмём оценку времени жизни $\hat{\tau}$ из выражения (7.7). Распределение вероятностей по значениям $\hat{\tau}$ при некотором истинном значении времени жизни τ равно

$$\frac{dW}{d\hat{\tau}} = f_n(\hat{\tau}) = \frac{1}{(n-1)!} \left(\frac{n\hat{\tau}}{\tau}\right)^{n-1} \frac{n}{\tau} \exp\left(-\frac{n\hat{\tau}}{\tau}\right). \quad (7.9)$$

Как уже указывалось, эта оценка параметра является несмещённой, т. е. среднее значение $\hat{\tau}$ равно τ . Наиболее вероятным значением $\hat{\tau}$ является

$$\hat{\tau}_{\text{н.в.}} = \frac{n-1}{n}\tau.$$

Используя распределение вероятностей (7.9), можно получить доверительный интервал для $\hat{\tau}$ при любом истинном значении τ с уровнем доверия $\beta < 1$. Центральный доверительный интервал можно определить двумя функциями $\hat{\tau}_1(\tau)$ и $\hat{\tau}_2(\tau)$ таким образом:

$$\int_0^{\hat{\tau}_1} f_n(\hat{\tau}) d\hat{\tau} = \frac{1-\beta}{2}; \quad \int_{\hat{\tau}_2}^{\infty} f_n(\hat{\tau}) d\hat{\tau} = \frac{1-\beta}{2}. \quad (7.10)$$

Для данной функции распределения вероятностей интеграл вычисляется в виде

$$\int_a^{\infty} f_n(\hat{\tau}) d\hat{\tau} = \left\{ 1 + \frac{na}{\tau} + \frac{1}{2} \left(\frac{na}{\tau}\right)^2 + \dots + \frac{1}{k!} \left(\frac{na}{\tau}\right)^k + \dots + \frac{1}{(n-1)!} \left(\frac{na}{\tau}\right)^{n-1} \right\} \exp\left(-\frac{na}{\tau}\right). \quad (7.11)$$

К сожалению, функции $\hat{\tau}_1(\tau)$ и $\hat{\tau}_2(\tau)$, найденные из уравнения (7.10), сложным образом зависят от n . Тем не менее, если решить уравнение для каждого τ и нанести на график, где по одной оси откладывается $\hat{\tau}$, а по другой τ , то получится доверительная зона. На рис.7.3 приведён пример такой доверительной зоны для $n = 5$, $\beta = 0,6827$. Теперь очевидно, что если мы для каждой оценки $\hat{\tau}$, полученной в соответствии с формулой (7.7), проведём вертикальную линию до пересечения с

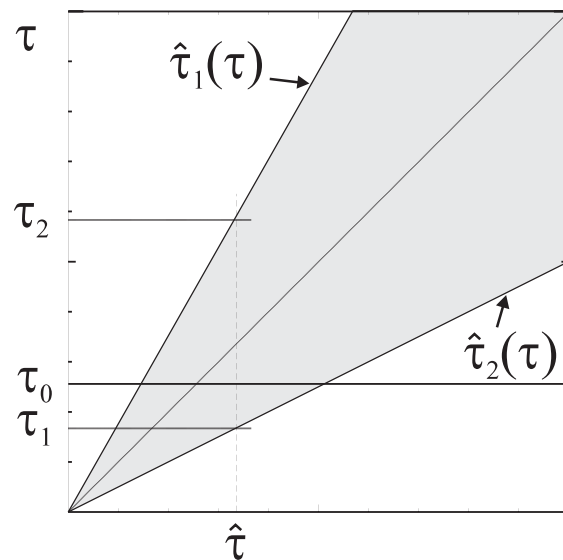


Figure 7.3: Пример построения доверительной зоны и доверительных интервалов для эксперимента по измерению времени жизни с $n = 5$, $\beta = 0,6827$

доверительной зоной (как показано на рисунке), то получим доверительный интервал (τ_1, τ_2) , который ровно в 68,27% случаев покрывает истинное значение τ_0 (каково бы ни было истинное значение!). Стоит ещё раз подчеркнуть специфику этого подхода: построение доверительной зоны идёт «по горизонтали», а считается доверительный интервал для каждой полученной в эксперименте оценки параметра «по вертикали». Нетрудно заметить, что несмотря на необходимость решения уравнения численным способом, решение для доверительного интервала в данном случае может быть представлено в удобном, не зависящем от $\hat{\tau}$ виде: $\Delta\tau^{(\pm)}/\hat{\tau}$ (как в табл. 7.1). В табл. 7.2 приведены доверительные интервалы, полученные при классическом подходе. Совпадения доверительных интервалов, полученных приближённо с помощью функции правдоподобия, с теми, которые получены более точным классическим методом Неймана, нет даже при статистике $n = 50$ измерений. Однако легко заметить, что длина интервала при больших n совпадает лучше, чем его положение. При $n = 50$ длины этих интервалов совпадают с точностью 1% (при смещении около 2% от длины интервала). Из сравнения результатов точного метода оценки доверительного интервала и приближённого по функции правдоподобия можно сделать вывод, что определение доверительного интервала по функции правдоподобия при анализе одного

Table 7.2: Границы $\Delta\tau^{(\pm)} = |\tau_{2,1} - \hat{\tau}|$ доверительных интервалов для уровня доверия 68% и 95%, определённых по классическому методу Неймана

n	$\beta = 0,6827$		$\beta = 0,9545$	
	$\frac{\Delta\tau^{(-)}}{\hat{\tau}}$	$\frac{\Delta\tau^{(+)}}{\hat{\tau}}$	$\frac{\Delta\tau^{(-)}}{\hat{\tau}}$	$\frac{\Delta\tau^{(+)}}{\hat{\tau}}$
1	0,457 (0,577*)	4,789 (2,314)	0,736 (0,778)	42,45 (18,06)
2	0,394 (0,469)	1,824 (1,228)	0,648 (0,682)	7,690 (5,305)
3	0,353 (0,410)	1,194 (0,894)	0,592 (0,621)	4,031 (3,164)
4	0,324 (0,370)	0,918 (0,725)	0,551 (0,576)	2,781 (2,314)
5	0,302 (0,341)	0,760 (0,621)	0,519 (0,541)	2,159 (1,858)
6	0,284 (0,318)	0,657 (0,550)	0,492 (0,513)	1,786 (1,571)
7	0,270 (0,299)	0,584 (0,497)	0,470 (0,489)	1,538 (1,374)
8	0,257 (0,284)	0,529 (0,456)	0,452 (0,469)	1,359 (1,228)
9	0,247 (0,271)	0,486 (0,423)	0,435 (0,451)	1,225 (1,116)
10	0,237 (0,260)	0,451 (0,396)	0,421 (0,436)	1,119 (1,027)
20	0,182 (0,194)	0,285 (0,261)	0,331 (0,341)	0,654 (0,621)
50	0,124 (0,129)	0,164 (0,156)	0,232 (0,237)	0,356 (0,346)

*В скобках указаны границы доверительных интервалов из табл. 7.1, приближённо оцененные по функции правдоподобия

измерения времени жизни (или любого другого параметра, который связан в эксперименте с экспоненциальным распределением вероятности) приводит к значительной ошибке, но с повышением статистики измерений использование приближённого метода всё более обоснованно.

7.3 Измерение параметров Гауссова распределения

Рассмотрим задачу определения параметров a и σ Гауссова распределения

$$\frac{dW}{dx} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-a)^2}{2\sigma^2}\right] \quad (7.12)$$

по выборке n измерений случайной величины x_i , $i = 1, \dots, n$.

Логарифмическая функция правдоподобия зависит от двух параметров

$$L = \sum_{i=1}^n \left[\frac{(x_i - a)^2}{2\sigma^2} + \ln(\sqrt{2\pi}\sigma) \right] \quad (7.13)$$

и достигает минимума при

$$\hat{a} = \frac{1}{n} \sum_{i=1}^n x_i; \quad \hat{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{a})^2}. \quad (7.14)$$

Границы доверительных интервалов для ΔL , соответствующего уровню доверия β , получаются из уравнений

$$\left(\frac{a - \hat{a}}{\hat{\sigma}} \right)^2 = e^{\frac{2\Delta L}{n}} - 1; \quad 2 \ln \left(\frac{\sigma}{\hat{\sigma}} \right) + \left(\frac{\hat{\sigma}}{\sigma} \right)^2 - 1 = \frac{2\Delta L}{n}. \quad (7.15)$$

Первое из этих уравнений определяет симметричный доверительный интервал

$$\Delta a^{(\pm)} = \hat{\sigma} \cdot \sqrt{e^{\frac{2\Delta L}{n}} - 1}. \quad (7.16)$$

Второе уравнение определяет несимметричный доверительный интервал $\Delta\sigma^{(\pm)}$ и может быть решено только численно для конкретных ΔL и n .

Обобщение классического метода Неймана построения доверительных интервалов на случай зависимости теоретической модели от нескольких параметров легче сделать в терминах «эллипсов ошибок», когда доверительные интервалы указываются не для каждого параметра в отдельности, а в виде области в многомерном пространстве (хотя технические сложности могут сделать это практически невозможным). Все же попробуем определить доверительные интервалы отдельно для a и σ .

Известно, что переменная

$$t = \sqrt{n-1} \cdot \frac{\hat{a} - a}{\hat{\sigma}} \quad (7.17)$$

имеет стандартное распределение Стьюдента (см., например, [48]), не зависящее от значений a и σ . Плотность вероятности распределения Стьюдента с $(n-1)$ степенью свободы равна

$$\frac{dW}{dt} = \frac{A}{\left(1 + \frac{t^2}{n-1}\right)^{n/2}}, \quad -\infty < t < +\infty. \quad (7.18)$$

Здесь A — нормировочная константа. Пользуясь этим распределением, для каждого n и уровня доверия β можно посчитать границы симметричного доверительного интервала для переменной t : $-\Delta t < t < \Delta t$. Отсюда можно определить симметричный доверительный интервал для параметра a :

$$\Delta a = |a - \hat{a}| < \hat{\sigma} \cdot \frac{\Delta t}{\sqrt{n-1}} = \frac{\Delta t}{\sqrt{n(n-1)}} \sqrt{\sum_{i=1}^n (x_i - \hat{a})^2}, \quad (7.19)$$

который независимо от истинных значений a и σ в β доле случаев будет включать истинное значение a .

Распределение вероятностей для величины $\mu = n \cdot \hat{\sigma}^2 / \sigma^2$ является стандартным распределением χ^2 с $(n-1)$ степенью свободы. Для него можно найти границы доверительного интервала $\mu_1 < \mu < \mu_2$ с уровнем доверия β . Тогда границы доверительного интервала для параметра σ получаются

$$\sigma_1 = \sqrt{\frac{n}{\mu_2}} \cdot \hat{\sigma} < \sigma < \sqrt{\frac{n}{\mu_1}} \cdot \hat{\sigma} = \sigma_2. \quad (7.20)$$

В таблице 7.3 приведены границы доверительных интервалов для a и σ при нескольких значениях n . Вместо оценки максимального правдоподобия $\hat{\sigma}$ на практике чаще используется так называемая «несмещенная» оценка среднеквадратичного отклонения

$$\tilde{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{a})^2} \quad (7.21)$$

Как видно из таблицы 7.4, обе эти оценки являются смещенными, несмещенной оценкой является только оценка дисперсии $\tilde{\sigma}^2$. Так как границы доверительных интервалов не зависят от вспомогательных обозначений, то при вычислении границ вместо $\hat{\sigma}$ можно подставлять ее выражение через $\tilde{\sigma}$:

$$\hat{\sigma} = \sqrt{\frac{n-1}{n}} \cdot \tilde{\sigma}.$$

Отличие приближенной оценки границ доверительных интервалов от точных в таблице 7.3 заметны, но какая при этом допускается ошибка в вероятности покрытия истинного значения из этой таблицы не видно. Для определения этой вероятности были проведены расчеты методом Монте-Карло, и результаты представлены на рис.7.4 для среднего и среднеквадратичного отклонения отдельно.

Table 7.3: Границы доверительных интервалов для параметров Гауссова распределения, определяемых по n измерениям переменной x . В скобках указаны оценки границ доверительных интервалов по функции правдоподобия по формуле (7.15).

n	$\beta = 0.6827$ ($\Delta L = 0.5$)		
	$\frac{\Delta a}{\hat{\sigma}}$	$\frac{\sigma_1}{\hat{\sigma}}$	$\frac{\sigma_2}{\hat{\sigma}}$
2	1.837 (0.805)	1.003 (0.651)	7.065 (1.821)
3	0.934 (0.629)	0.903 (0.698)	2.947 (1.604)
4	0.691 (0.533)	0.878 (0.729)	2.190 (1.493)
5	0.571 (0.471)	0.870 (0.751)	1.879 (1.424)
6	0.497 (0.426)	0.868 (0.768)	1.708 (1.376)
7	0.445 (0.392)	0.869 (0.782)	1.600 (1.341)
8	0.407 (0.365)	0.870 (0.794)	1.525 (1.313)
9	0.377 (0.343)	0.872 (0.804)	1.469 (1.291)
10	0.353 (0.324)	0.874 (0.812)	1.426 (1.273)
12	0.316 (0.295)	0.878 (0.826)	1.363 (1.245)
15	0.277 (0.263)	0.885 (0.842)	1.303 (1.215)
20	0.236 (0.226)	0.893 (0.861)	1.243 (1.182)
30	0.189 (0.184)	0.906 (0.884)	1.182 (1.144)
50	0.144 (0.142)	0.922 (0.908)	1.130 (1.109)
n	$\beta = 0.9545$ ($\Delta L = 2$)		
	$\frac{\Delta a}{\hat{\sigma}}$	$\frac{\sigma_1}{\hat{\sigma}}$	$\frac{\sigma_2}{\hat{\sigma}}$
2	13.968 (2.528)	0.621 (0.471)	49.59 (4.366)
3	3.201 (1.671)	0.630 (0.526)	8.074 (3.042)
4	1.909 (1.311)	0.647 (0.564)	4.449 (2.511)
5	1.435 (1.107)	0.663 (0.593)	3.296 (2.223)
6	1.185 (0.974)	0.678 (0.616)	2.744 (2.041)
7	1.027 (0.878)	0.690 (0.635)	2.423 (1.914)
8	0.918 (0.805)	0.701 (0.651)	2.212 (1.821)
9	0.837 (0.748)	0.711 (0.665)	2.062 (1.748)
10	0.773 (0.701)	0.720 (0.677)	1.951 (1.691)
12	0.680 (0.629)	0.735 (0.698)	1.795 (1.604)
15	0.587 (0.553)	0.753 (0.722)	1.649 (1.515)
20	0.491 (0.471)	0.776 (0.751)	1.511 (1.424)
30	0.388 (0.378)	0.806 (0.788)	1.376 (1.326)
50	0.293 (0.289)	0.841 (0.829)	1.265 (1.239)

Table 7.4: Наиболее вероятные и средние значения некоторых оценок параметра σ Гауссова распределения по n измерениям x_i .

Величина	$\hat{\sigma}$	$\tilde{\sigma}$	$(\hat{\sigma})^2$	$(\tilde{\sigma})^2$
Наиболее вероятное	$\sqrt{\frac{n-2}{n}}$	$\sqrt{\frac{n-2}{n-1}}$	$\frac{n-3}{n}$	$\frac{n-3}{n-1}$
Среднее значение	$\frac{(n-2)!!}{(n-3)!!} \cdot \sqrt{\frac{R_n}{n}} a$	$\frac{(n-2)!!}{(n-3)!!} \cdot \sqrt{\frac{R_n}{n-1}}$	$\frac{n-1}{n}$	1

^aЗдесь использованы обозначения:

$$n!! = n \cdot (n-2) \cdot (n-4) \cdots, \quad R_{2k} = 2/\pi, \quad R_{2k+1} = \pi/2.$$

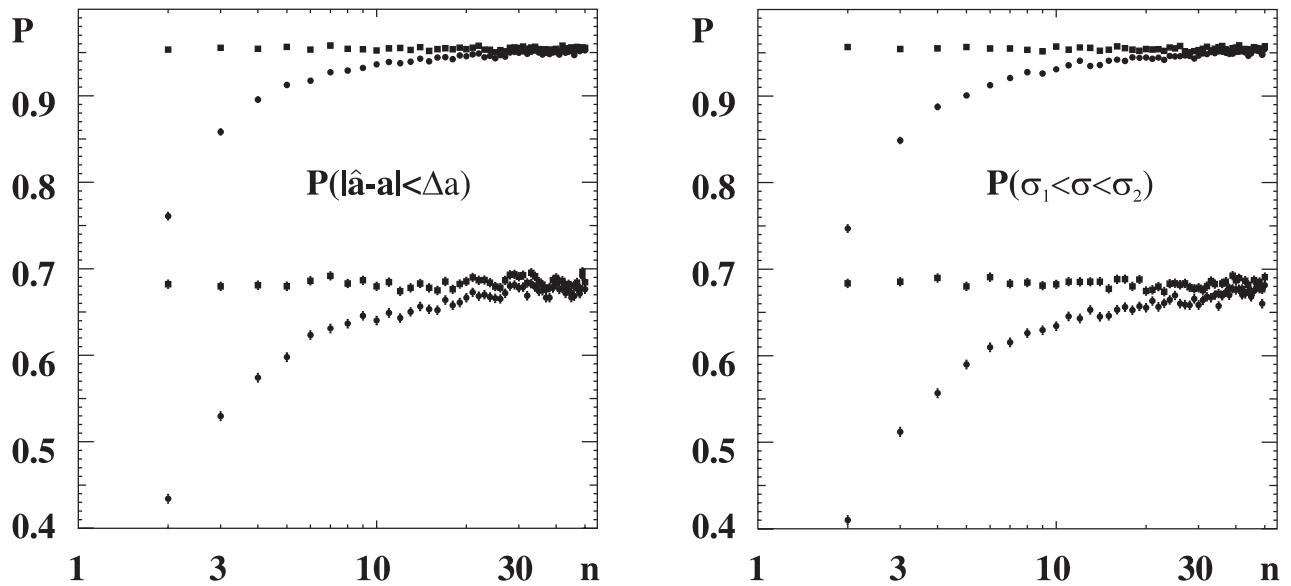


Figure 7.4: Вероятность P покрыть истинное значение параметров a (слева) и σ (справа) Гауссова распределения доверительным интервалом для двух уровней доверия: 68.27% и 95.45%. \bullet — доверительный интервал определяется по функции правдоподобия, \blacksquare — доверительный интервал определяется методом Неймана. Статистика расчета в каждой точке равна 10^4 .

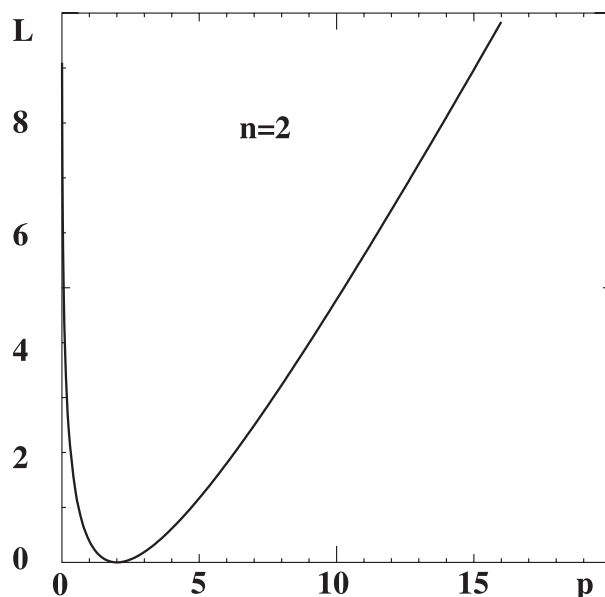


Figure 7.5: Функция правдоподобия для Пуассонова распределения вероятностей

7.4 Среднее значение Пуассонова распределения

Рассмотрим простейший случай, когда по единственному измерению n мы пытаемся определить среднее значение p Пуассонова распределения (3.33). Функция правдоподобия в этом случае может быть записана в виде

$$L = p - n + n \cdot \ln \frac{n}{p} \quad (7.22)$$

(здесь использовано то обстоятельство, что к логарифмической функции правдоподобия можно добавить любое слагаемое, не зависящее от параметра p). График этой функции всегда несимметричен (пример графика при $n = 2$ приведён на рис. 7.5). Легко получить оценку параметра p по методу максимального правдоподобия:

$$\hat{p} = n. \quad (7.23)$$

68%-й доверительный интервал получается решением уравнения

$$\sigma_p - n \cdot \ln \left(1 + \frac{\sigma_p}{n} \right) = \frac{1}{2} \quad (7.24)$$

относительно σ_p . Очевидно, что при любом $n > 0$ существует ровно два решения в области $p > 0$. В пределе больших n величина $\frac{\sigma_p}{n}$ становится малой, и логарифм можно разложить в ряд до второго члена, тогда решение получается в следующем виде:

$$\frac{\sigma_p^2}{2n} = \frac{1}{2} \implies \sigma_p = \pm\sqrt{n}. \quad (7.25)$$

Уравнение (7.24) можно решать любым численным способом, например, методом деления пополам. Можно также использовать метод последовательных приближений с рекуррентными формулами. Например, можно использовать такой подход. Если обозначить за величину $x = \sigma_p/n$, то для положительной ветви решения можно использовать формулу

$$x_{k+1} = \frac{[\ln(1+x_k) + \frac{1}{2n}] \cdot (1+x_k) - x_k}{x_k}; \quad x_1 = 1 \quad (7.26)$$

и для отрицательной ветви

$$x_{k+1} = \exp\left(x_k - \frac{1}{2n}\right) - 1; \quad x_1 = -1. \quad (7.27)$$

В случае $n = 0$ при прямолинейном использовании рекомендаций метода максимального правдоподобия, оптимальное значение $\hat{p} = 0$ и 68%-й доверительный интервал получается $(0; 0,5)$.

Полученный таким образом доверительный интервал можно проверить методом Монте–Карло, вычислив вероятность P – в какой доле случаев доверительный интервал захватывает истинное значение p . Зависимость этой величины $P(p)$ представлена на рис. 7.6 (там же нарисован аналогичный график для 95%-го интервала, где $\Delta L = 2$). Обращает на себя внимание медленная сходимость результата к асимптотическому при увеличении статистики вопреки традиционному мнению, что для Пуассонова распределения при статистике 10 или более событий все характеристики совпадают с характеристиками Гауссова распределения. Причиной негладкости представленной функции является неограниченная статистика расчёта методом Монте–Карло, а дискретность распределения Пуассона.

Доверительный интервал по методу Неймана. Дискретность распределения вероятностей не позволяет определять границы зоны аналогично формуле (7.10). Традиционное решение в таком случае – это минимальным образом расширить доверительную зону так, чтобы

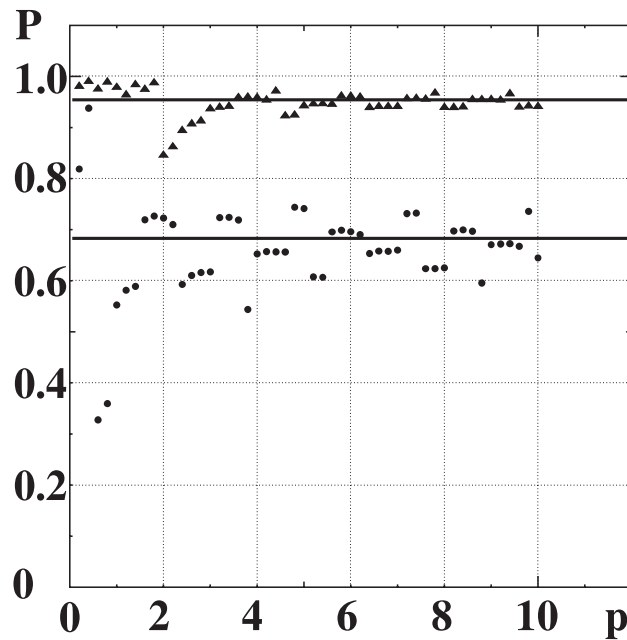


Figure 7.6: Вероятность P попадания истинного значения параметра p Пуассона распределения в доверительный интервал, полученный методом максимального правдоподобия. Обозначения: ● – 68%-й доверительный интервал, ▲ – 95%-й доверительный интервал. Горизонтальные сплошные линии показывают «правильное» значение вероятности (0,6827 и 0,9545)

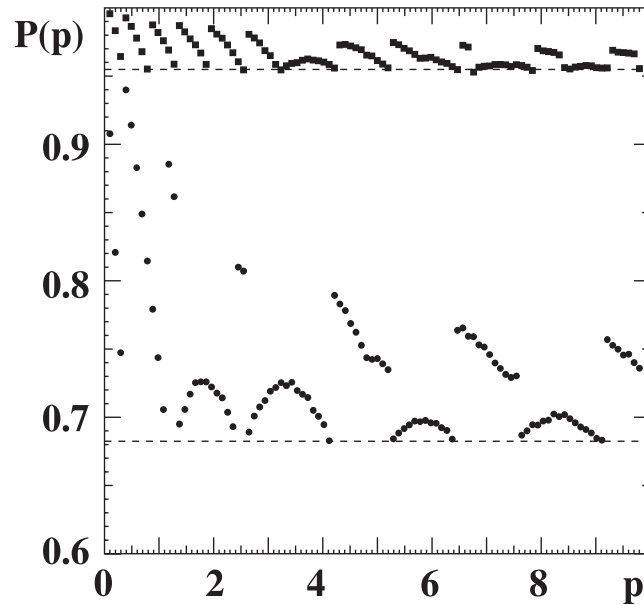


Figure 7.7: Вероятность покрытия P истинного значения параметра p системой доверительных интервалов для $\beta = 0,6827$ (●) и $0,9545$ (■) из табл. 7.5

всегда выполнялось неравенство $P(p) \geq \beta$. Такой подход называется консервативным. В нашем случае оценка параметра $\hat{p} = n$ имеет то же самое распределение вероятностей Пуассона. В рамках консервативного подхода при построении доверительной зоны для любого значения p мы должны определить интервал суммирования (k_1, k_2) такой, чтобы

$$\sum_{k=k_1}^{k_2} \frac{p^k}{k!} e^{-p} \geq \beta. \quad (7.28)$$

Доверительная зона для случая $\beta = 0,6827$ изображена на рис. 7.8. Так как результаты эксперимента здесь – всегда целые числа, то можно для первых нескольких значений n привести вычисленные доверительные интервалы (табл. 7.5). На рис. 7.7 показана вероятность покрытия истинного значения p системой доверительных интервалов с уровнем доверия $\beta = 0,6827$ и $0,9545$. Сравнивая вероятности покрыть истинное значение доверительным интервалом с 68%-м уровнем доверия, полученным по функции правдоподобия (рис. 7.6) и методом Неймана (рис. 7.7), можно заметить, что при $p > 2$ разброс вероятности для этих методов одинаков (определяется дискретностью распределения вероятностей), однако для метода Неймана строго выполняется условие

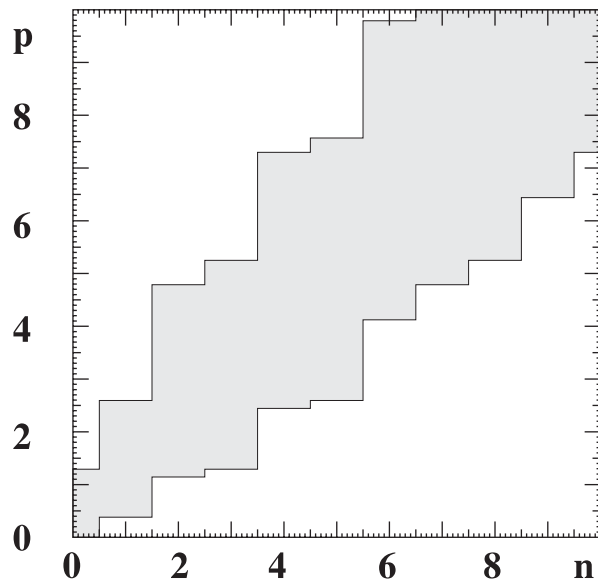


Figure 7.8: Доверительная зона для оценки параметра p распределения Пуассона с уровнем доверия $\beta = 0,6827$

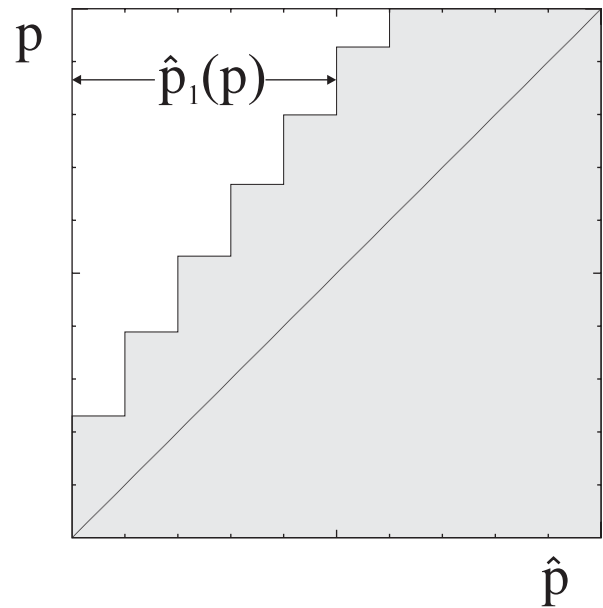


Figure 7.9: Доверительная зона с уровнем доверия 90% для оценки параметра распределения Пуассона \hat{p}

Table 7.5: Границы доверительных интервалов с уровнем доверия β для оценки параметра p распределения Пуассона по единственному измерению n

n	$\beta = 0,6827$		$\beta = 0,9$		$\beta = 0,9545$	
	p_1	p_2	p_1	p_2	p_1	p_2
0	0,000	1,290	0,000	2,436	0,000	3,234
1	0,382	2,593	0,105	4,951	0,047	5,752
2	1,145	4,787	0,532	6,470	0,337	6,742
3	1,290	5,250	1,102	7,969	0,787	8,332
4	2,444	7,296	1,745	8,597	1,324	9,769
5	2,593	7,569	2,433	9,716	1,917	11,100
6	4,122	9,792	2,436	11,993	2,551	13,280
7	4,787	10,811	3,589	12,531	3,214	14,764
8	5,250	11,242	4,532	14,508	3,234	15,360
9	6,438	13,299	4,951	15,298	4,309	16,401
10	7,296	14,312	5,976	15,986	5,208	18,768

«консервативности» $P(p) \geq 68,27\%$. Кроме того, с увеличением p разброс уменьшается, и в асимптотике оба метода будут эквивалентны.

7.5 Верхний предел на параметр распределения Пуассона

Под верхним пределом понимается граница одностороннего интервала, вероятность попадания в который измеряемого параметра называется уровнем доверия. Обычно в физике элементарных частиц используется верхний предел с 90%-м уровнем доверия. Пример использования приближённой оценки верхнего предела времени жизни по функции правдоподобия приведён на рис.7.2 (результаты нанесены символами ■). Видно, что при малом числе измерений, когда этот результат и представляет интерес, ошибка в уровне доверия значительна и составляет около 10%.

Для параметров, имеющих непрерывное распределение вероятностей, как в случае с измерением времени жизни, классический подход применим и для точного установления верхнего предела. Практическое отличие от нахождения двустороннего доверительного интервала состоит в следующем. При расчёте доверительной зоны надо при каждом предполагаемом значении времени жизни τ вычислять границу интервала $\hat{\tau}_1(\tau)$, для которой вероятность $P(\hat{\tau} \geq \hat{\tau}_1)$ равна заданному уровню доверия (например, 0,9). Проведя соответствующие вычисления, получим в случае однократного измерения времени жизни $\hat{\tau}$ 90%-й верхний предел, равный $9,49 \cdot \hat{\tau}$, а для среднего по двум измерениям – $3,76 \cdot \hat{\tau}$, что несколько превышает приближённые оценки по методу максимального правдоподобия.

Как мы уже видели, в случае дискретного распределения оценок параметра ситуация осложняется. Рассмотрим в качестве примера эксперимент, оценка параметра $\hat{p} = n$ в котором пропорциональна числу отсчётов n , распределённому по закону Пуассона:

$$W_n = \frac{p^n}{n!} e^{-p}, \quad n = 0, 1, 2, \dots \quad (7.29)$$

с некоторым неизвестным параметром p . При попытке построить доверительную зону классическим методом, из-за дискретности распределения мы вынуждены будем заменить правило вычисления

Table 7.6: Верхний предел на параметр p распределения Пуассона на 90, 95 и 99%-м уровне доверия

Число зарегистрированных событий n	0	1	2	3	4	5
Верхний предел на параметр p ($\beta = 90\%$)	2,30	3,89	5,32	6,68	7,99	9,27
Верхний предел на параметр p ($\beta = 95\%$)	3,00	4,74	6,30	7,75	9,15	10,51
Верхний предел на параметр p ($\beta = 99\%$)	4,61	6,64	8,41	10,05	11,60	13,11

левой границы доверительной зоны (7.10) на

$$\sum_{n=\hat{p}_1(p)}^{\infty} W_n \geq \beta > \sum_{n=1+\hat{p}_1(p)}^{\infty} W_n, \quad (7.30)$$

где β — заданный уровень доверия. Построенная по этому правилу доверительная зона для 90%-го уровня доверия изображена на рис. 7.9. В табл. 7.6 приведена величина верхнего предела на параметр Пуассонова распределения p для нескольких значений числа зарегистрированных событий. Верхний предел, установленный в рамках этого подхода, очевидно, обладает следующим свойством: каково бы ни было истинное значение параметра распределения Пуассона p , при многократном повторении этого эксперимента верхний предел в $100 \cdot \beta\%$ случаев (или немного больше) превысит истинное значение параметра p . При этом обязательно подразумевается, что *независимо от результата эксперимента* устанавливается верхний предел, что в реальном эксперименте не выполняется, ибо при результате больше некоторого, как правило, приводится двусторонний доверительный интервал.

В работе [66] предлагается способ борьбы с этим и некоторыми другими недостатками прямолинейного применения классического метода

Table 7.7: Таблица доверительных интервалов для Пуассонова распределения*

n	0	1	2	3	4	5
$\beta = 0,9$	2,44	4,36	5,91	7,42	8,60	9,99
	0,00	0,11	0,53	1,10	1,47	1,84
$\beta = 0,95$	3,09	5,14	6,72	8,25	9,76	11,26
	0,00	0,05	0,36	0,82	1,37	1,84
n	6	7	8	9	10	11
$\beta = 0,9$	11,47	12,53	13,99	15,30	16,50	
	2,21	3,56	3,96	4,36	5,50	
$\beta = 0,95$	12,75	13,81	15,29	16,77	17,82	
	2,21	2,58	2,94	4,36	4,75	

*Таблица приводится из работы [66]

построения доверительных зон (в частности, избавление от доверительных интервалов нулевой длины). Одно из предложений в этой работе – это формировать (по горизонтали) доверительную зону с учётом особенностей обработки получаемых результатов. В применении к нашей задаче предлагается построить доверительную зону так, чтобы при числе зарегистрированных событий 0 (и, может быть, 1) получался верхний предел на параметр p , а при больших значениях n – двусторонний доверительный интервал. Теперь можно методом Монте–Карло проверить, насколько соответствует вероятность «покрытия» истинного значения параметра объявленному уровню доверия, если при многократном повторении эксперимента с некоторым истинным значением параметра p пользоваться для представления результата табл. 7.7 (график зависимости $P(p)$ приведён на рис. 7.10). Построенная доверительная зона выглядит недостаточно оптимальной. Посмотрим, нельзя ли для этого конкретного случая построить доверительную зону, для которой превышение вероятности покрытия истинного значения параметра над заявленным уровнем доверия минимально? Для заданного уровня доверия β эта задача математически соответствует следующей формулировке: для любого значения параметра p подобрать интервал суммирования (k_1, k_2)

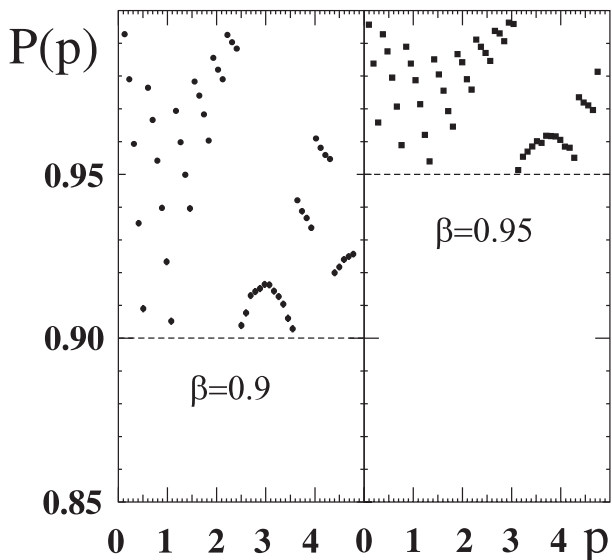


Figure 7.10: Вероятность $P(p)$ «покрытия» истинного значения параметра p доверительным интервалом с уровнем доверия β из табл. 7.7. Обозначения: \bullet – $\beta = 0,9$, \blacksquare – $\beta = 0,95$

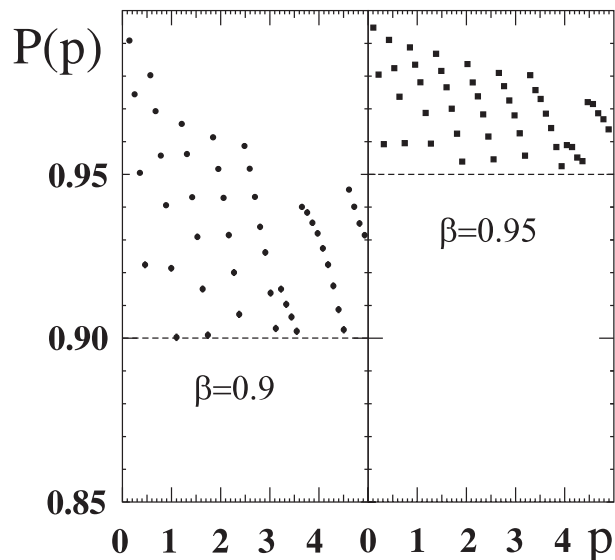


Figure 7.11: Вероятность $P(p)$ «покрытия» истинного значения параметра p доверительным интервалом с уровнем доверия β . Обозначения: \bullet – $\beta = 0,9$, \blacksquare – $\beta = 0,95$. Доверительная зона построена численной минимизацией превышения вероятности «покрытия» над уровнем доверия (табл. 7.8)

Table 7.8: Таблица доверительных интервалов для пуассонова распределения, полученная численной минимизацией превышения над уровнем доверия β вероятности покрытия истинного значения параметра p доверительным интервалом

n	0	1	2	3	4	5
$\beta = 0,9$	3,15	5,38	6,91	8,50	9,13	10,81
	0,00	0,11	0,53	1,10	1,75	2,43
$\beta = 0,95$	3,98	6,12	7,58	9,08	10,63	12,23
	0,00	0,05	0,36	0,82	1,37	1,97
n	6	7	8	9	10	11
$\beta = 0,9$	12,53	13,16	14,97	15,58	17,47	18,07
	3,15	3,59	4,53	5,38	5,98	6,91
$\beta = 0,95$	13,86	14,50	16,20	17,91	18,57	20,35
	2,61	3,29	3,98	4,46	5,32	6,12

для функции

$$P_{k_1 k_2}(p) = \sum_{k=k_1}^{k_2} \frac{p^k}{k!} e^{-p}, \quad 0 \leq k_1 \leq k_2 < \infty \quad (7.31)$$

так, чтобы всегда выполнялось неравенство $P_{k_1 k_2}(p) - \beta \geq 0$, но величина превышения была минимальной. В области малых n такую минимизацию можно провести численно и составить таблицу доверительных интервалов (табл.7.8). Вероятность «покрытия» истинного значения параметра p доверительными интервалами из этой таблицы представлена на рис. 7.11.

Сравнение всех этих вариантов показывает, что задача постановки верхнего предела на параметр распределения Пуассона, сильно осложнённая дискретностью распределения, не имеет однозначного решения. Очевидно, что для единообразия должно быть соглашение между экспериментаторами относительно таблиц доверительных интервалов при малом числе зарегистрированных событий. Результаты такого соглашения отражаются в статистическом разделе обзора физики частиц [68].

7.6 Подгонка резонансной кривой

В экспериментах на коллайдерах, когда зависимость числа событий от

энергии в системе центра масс подгоняется кривой возбуждения резонанса с целью получить параметры изучаемого резонанса (массу, ширину и т.п.), распределение вероятностей по полученным оценкам является дискретным, хотя это распределение часто бывает достаточно плотным. Построение доверительной зоны для таких случаев представляется достаточно трудоемкой задачей, и результат будет зависеть от многих обстоятельств: распределения интеграла светимости по энергии, наличия фона, теоретической модели и т.п. Очевидно, что этим надо заниматься в тех случаях, когда есть уверенность, что построение доверительных интервалов по функции правдоподобия дает неверный результат. Но даже эта более простая задача проверки доверительных интервалов может потребовать большой вычислительной мощности, так как при обработке экспериментальных данных обычно проводится численная минимизация логарифмической функции правдоподобия, и численное моделирование многократного повторения обработки эксперимента при разных истинных значениях параметров теоретической модели может представлять сложную техническую задачу.

Здесь мы рассмотрим совсем упрощенную модель эксперимента по определению параметров резонанса.

Пусть в десяти точках по энергии получено число событий какого-либо канала распада резонанса, пусть также нет никакого фона (что редко бывает) и свободных параметров модели всего два: масса резонанса и интенсивность распада. То есть видимая форма резонанса фиксирована и определяется, например, известным аппаратурным разрешением σ , а истинная ширина резонанса много меньше этой величины. Для простоты выберем эту форму в виде Гауссова распределения, хотя обычно даже в таких простых случаях она имеет несимметричный вид.

Итак, проведен эксперимент, в котором в точках с энергией

$$E_i = E_0 + (i - 5.5) \cdot \sigma/2, \quad i = 1, 2, \dots, 10$$

с равным интегралом светимости измерено число событий n_i распада резонанса по изучаемому каналу. Ожидаемое число событий в точке i равно

$$p_i = B \cdot \exp \left[-\frac{(m - E_i)^2}{2\sigma^2} \right]. \quad (7.32)$$

Функция правдоподобия, зависящая от параметров m и B , может быть

записана в виде

$$L = \sum_{i=1}^{10} \left[p_i - n_i + n_i \cdot \ln \frac{n_i}{p_i} \right]. \quad (7.33)$$

В случае $n_i = 0$ последнее слагаемое в скобках полагается равным нулю. По параметру B минимизация может быть проведена аналитически:

$$B_0 = \frac{\sum_{i=1}^{10} n_i}{\sum_{i=1}^{10} \exp \left[-\frac{(m-E_i)^2}{2\sigma^2} \right]}, \quad (7.34)$$

а минимизация по параметру m может быть быстро выполнена, например, методом «золотого сечения» (см. разд. 5.1).

Возьмем теперь истинное значение $m = E_0$, и в зависимости от истинного значения B проверим вероятность покрытия истинных значений m и B 68%-ми доверительными интервалами, полученными по функции правдоподобия. Легко заметить, что среднее значение

$$\left\langle \sum_{i=1}^{10} n_i \right\rangle = B \cdot \sum_{i=1}^{10} \exp \left[-\frac{(i-5.5)^2}{8} \right] \approx 5 \cdot B. \quad (7.35)$$

Поэтому интервал B от 1 до 100 соответствует средней статистике эксперимента от 5 до 500. На рис.7.12 приведена вероятность покрытия истинных значений B и m доверительными интервалами, полученными по функции правдоподобия на уровне $\Delta L = 1/2$ (68%-ый доверительный интервал).

Результат получился несколько неожиданный — даже при статистике порядка 5 событий оценка доверительного интервала по функции правдоподобия дает достаточно точную вероятность покрытия истинного значения. Для оценки массы резонанса даже не видно ухудшения при низкой статистике порядка 5 событий. Возможно, это объясняется отсутствием сильной корреляции между параметрами B и m (коэффициент линейной корреляции в нашем случае равен нулю). По-видимому, если бы свободными параметрами модели были полная ширина на полувысоте Γ и интенсивность распада B , то точность оценки обоих параметров ухудшалась с уменьшением статистики.

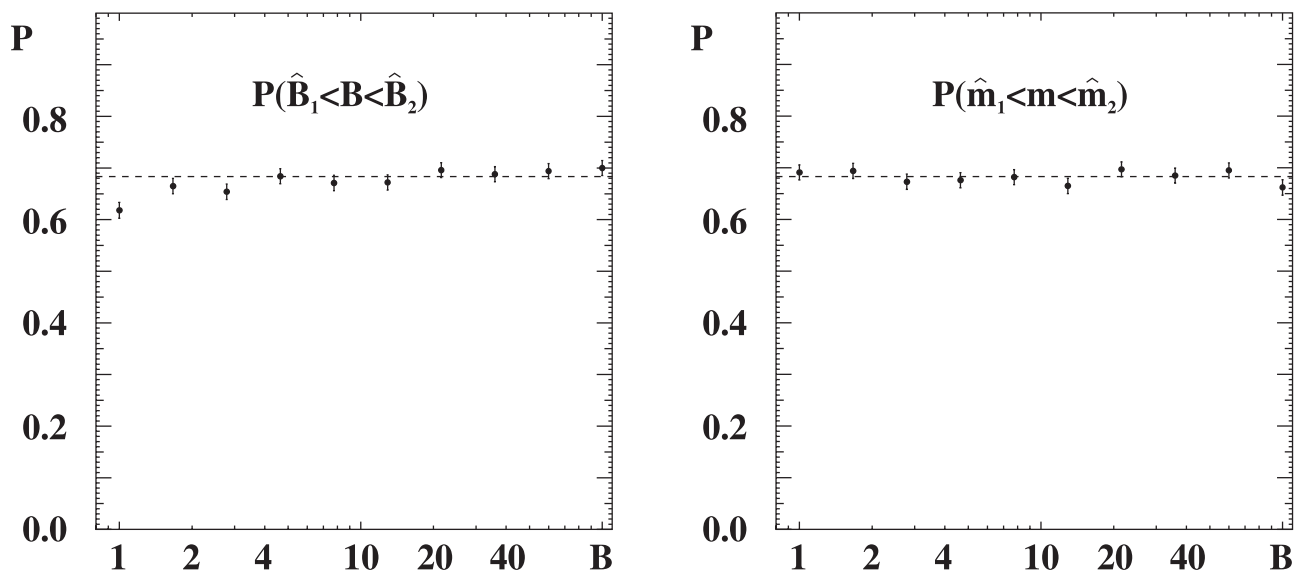


Figure 7.12: Вероятность покрытия истинных значений параметров B и m резонансной кривой доверительными интервалами (\hat{B}_1, \hat{B}_2) и (\hat{m}_1, \hat{m}_2) , соответственно, полученными по функции правдоподобия на уровне $\Delta L = 1/2$ (ожидаемый уровень доверия 68%). Статистика расчетов методом Монте-Карло в каждой точке — 10^3 «экспериментов»

Chapter 8

Разделение событий

В практике обработки экспериментальных данных часто требуется поток экспериментальных событий «рассортировать» на несколько типов событий в соответствии с теоретическими описаниями этих событий.

8.1 Разделение событий двух классов

Типичной задачей обработки эксперимента является задача оптимального разделения событий двух классов. Условия можно сформулировать следующим образом. Пусть каждое событие характеризуется набором параметров \mathbf{x} и в потоке событий имеются события двух классов. События первого класса распределены в соответствии с известным распределением вероятностей $f_1(\mathbf{x})$, события второго класса – $f_2(\mathbf{x})$. Требуется подобрать такую функцию параметров $\Psi(\mathbf{x})$, чтобы по значению этой функции оптимальным образом разделить события на две «чистые» группы, т.е. если $\Psi(\mathbf{x}) > \Psi_0$, то событие относится к первой группе, в противном случае – ко второй. Требование оптимальности можно конкретизировать следующим образом. Пусть выбор порогового значения Ψ_0 определяется тем, чтобы в первую группу попадала заданная часть ϵ_0 событий первого класса. Оптимальным параметром отбора назовём такую функцию $\Psi(\mathbf{x})$, которая минимизирует примесь ϵ_1 в первую группу событий второго класса.

Похожие определения оптимальности отбора можно формулировать по-разному, но решение задачи существует, причём всегда одно и то же: наилучшим параметром разделения событий является отношение функций

правдоподобия $\Psi(\mathbf{x}) = f_1(\mathbf{x})/f_2(\mathbf{x})$ (критерий Неймана – Пирсона [60, 12]) или любая взаимно однозначная функция этого отношения (часто используют логарифм отношения функций правдоподобия). Однако на практике этот теоретический результат можно использовать только тогда, когда достаточно хорошо известны функции распределения $f_1(\mathbf{x}), f_2(\mathbf{x})$. В качестве иллюстрации рассмотрим пример разделения событий двух классов, где каждое событие описывается двумя параметрами.

Пусть события первого класса имеют функцию распределения вероятностей

$$f_1(x_1, x_2) = \frac{1}{2\pi\sqrt{1-q^2}} \exp \left\{ -\frac{x_1^2 + x_2^2 - 2qx_1x_2}{2(1-q^2)} + \frac{x_1 + x_2 - 1}{1+q} \right\}, \quad (8.1)$$

а события второго класса – функцию

$$f_2(x_1, x_2) = \frac{1}{8\pi\sqrt{1-q^2}} \exp \left\{ -\frac{x_1^2 + x_2^2 - 2qx_1x_2}{8(1-q^2)} - \frac{x_1 + x_2 + 1}{4(1+q)} \right\}. \quad (8.2)$$

Легко убедиться, что $\langle x_1^{(1)} \rangle = \langle x_2^{(1)} \rangle = 1$, $\langle x_1^{(2)} \rangle = \langle x_2^{(2)} \rangle = -1$, $D(x_1^{(1)}) = D(x_2^{(1)}) = 1$, $D(x_1^{(2)}) = D(x_2^{(2)}) = 2$ и q – коэффициент корреляции (одинаковый для событий обоих классов). В качестве параметров разделения событий рассмотрим три варианта: «научный» параметр с использованием точных функций распределения

$$\Psi_1 = \ln \frac{f_1(x_1, x_2)}{f_2(x_1, x_2)} = \ln 4 + 3 \cdot \frac{2qx_1x_2 - x_1^2 - x_2^2}{8(1-q^2)} + \frac{5(x_1 + x_2) - 3}{4(1+q)}, \quad (8.3)$$

«научный» параметр с неправильной функцией распределения (не замечена корреляция)

$$\Psi_2 = \ln \frac{f_1^*}{f_2^*} = \ln 4 - \frac{3}{8}(x_1^2 + x_2^2) + \frac{5(x_1 + x_2) - 3}{4} \quad (8.4)$$

и «ненаучный» параметр

$$\Psi_3 = x_1 + x_2. \quad (8.5)$$

Теперь, пользуясь этими параметрами отбора, вычислим величину «примеси» событий второго класса в первую группу

$$\epsilon_i = \int_{\Psi_i(x_1, x_2) > \Psi_0} f_2(x_1, x_2) dx_1 dx_2, \quad (8.6)$$

при условии $\int_{\Psi_i(x_1, x_2) > \Psi_0} f_1(x_1, x_2) dx_1 dx_2 = \epsilon_0 = 0,5.$

На рис. 8.1 приведены графики зависимости $\epsilon_1, \epsilon_2, \epsilon_3$ от коэффициента корреляции q . Как и должно быть, параметр отбора Ψ_1 оказался самым лучшим. Интересно отметить, что для $-1 < q < -0,72$ «ненаучный» параметр Ψ_3 лучше разделяет события, чем параметр Ψ_2 , построенный на неправильной функции распределения, не учитывающей корреляцию параметров.

8.2 Использование корреляционной матрицы для разделения событий

В случае разделения двухчастичных событий бывает возможным применить специальный приём, предложенный Сидоровым В.А. [69] в 1967 году, и использованный для разделения частиц в первых экспериментах на встречных пучках в Новосибирске [70, 71].

Формально для применения метода корреляционных матриц должны выполняться следующие условия: в двухчастичных событиях частицы всегда должны быть одинакового типа (например, если первая частица – пион, то и вторая частица должна быть пионом) и распределения вероятностей по параметру разделения должны быть одинаковы для первой и второй частицы. Тогда, даже не зная этих распределений вероятности, можно оценить количество пар частиц первого и второго типа, а также восстановить распределения вероятностей. Точнее этот метод можно было бы назвать не методом разделения частиц, а методом оценки общего числа частиц первого и второго типа по двумерному распределению событий. Рассмотрим идею этого метода, введя следующие обозначения.

Пусть математическое ожидание числа событий первого типа равно N_x , событий второго типа – N_y . И пусть по выбранному параметру разделения для экспериментальных событий построена двумерная гистограмма n_{ij} ; $i = 1, 2, \dots, k$; $j = 1, 2, \dots, k$. Предположим, что распределение вероятностей событий первого типа по каналам гистограммы равно x_i , $i = 1, \dots, k$, а для событий второго типа – y_i , $i = 1, \dots, k$. Из условия нормировки должны

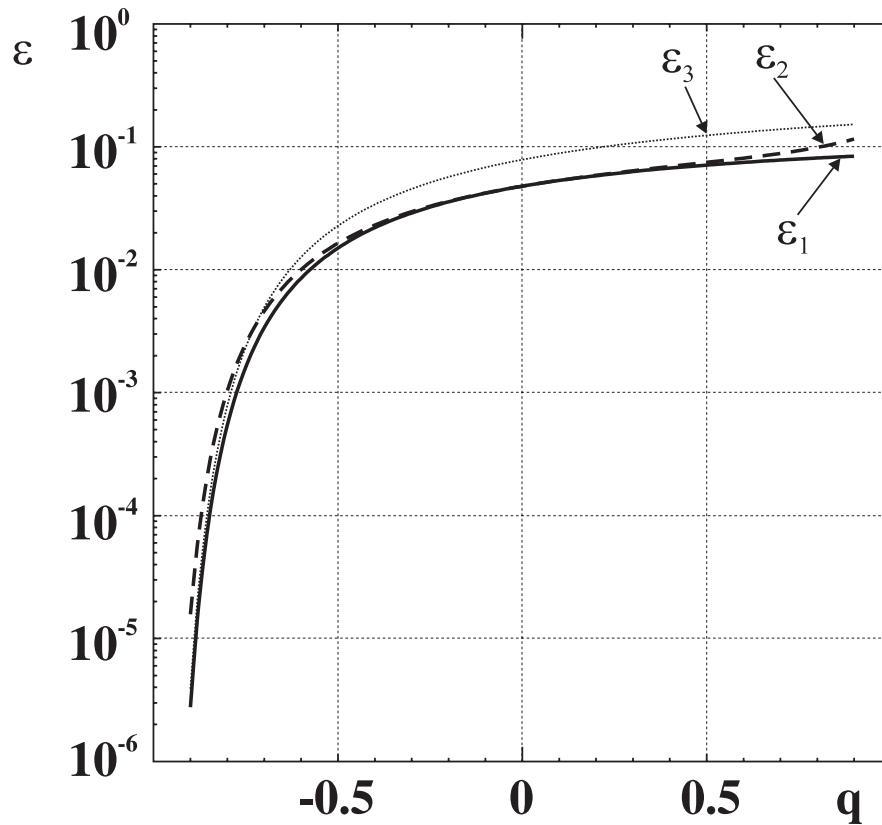


Figure 8.1: Величина примеси событий второго класса к событиям первого класса при использовании разных параметров отбора: ϵ_1 – критерий Неймана – Пирсона с правильной функцией правдоподобия; ϵ_2 – критерий Неймана – Пирсона с неправильной функцией правдоподобия (не учтена корреляция); ϵ_3 – параметр отбора $\Psi_3 = x_1 + x_2$

8.2. ИСПОЛЬЗОВАНИЕ КОРРЕЛЯЦИОННОЙ МАТРИЦЫ ДЛЯ РАЗДЕЛЕНИЯ С

быть справедливы тождества:

$$\sum_{i=1}^k x_i = 1; \quad \sum_{i=1}^k y_i = 1. \quad (8.7)$$

Математическое ожидание числа событий в каждой ячейке гистограммы теперь можно написать в виде

$$p_{ij} = N_x \cdot x_i \cdot x_j + N_y \cdot y_i \cdot y_j. \quad (8.8)$$

Если бы не было статистических флуктуаций при регистрации событий, то мы могли бы составить $k \cdot (k + 1) / 2$ уравнений

$$p_{ij} = n_{ij}, \quad i = 1, 2, \dots, k, \quad j = i, i + 1, \dots, k \quad (8.9)$$

(учтена симметрия гистограммы относительно диагонали). Всего независимых переменных у нас $2k$, а уравнений – $k \cdot (k + 1) / 2$, поэтому при достаточно большом k число уравнений превысит число неизвестных и система уравнений может иметь единственное решение. Легко получить это необходимое условие:

$$\frac{k \cdot (k + 1)}{2} \geq 2k \implies k \geq 3. \quad (8.10)$$

Однако, это рассуждение только позволяет проиллюстрировать идею метода. В действительности, всегда есть статистические флуктуации (в данном случае соответствуют распределению вероятностей Пуассона), кроме того, при значении $k > 3$ уравнений уже будет больше, чем неизвестных, и система уравнений станет несовместной. Метод максимального правдоподобия позволяет обойти эти трудности.

Сначала сделаем замену переменных, которая избавит нас от условий нормировки:

$$\hat{x}_i = \sqrt{N_x} \cdot x_i; \quad \hat{y}_i = \sqrt{N_y} \cdot y_i; \quad \hat{x}_i, \hat{y}_i \geq 0. \quad (8.11)$$

Обратное преобразование очевидно:

$$N_x = \left(\sum_{i=1}^k \hat{x}_i \right)^2; \quad x_i = \frac{\hat{x}_i}{\sum_{j=1}^k \hat{x}_j}; \quad N_y = \left(\sum_{i=1}^k \hat{y}_i \right)^2; \quad y_i = \frac{\hat{y}_i}{\sum_{j=1}^k \hat{y}_j}. \quad (8.12)$$

В этих переменных математическое ожидание числа событий в каждой ячейке гистограммы записывается немного проще:

$$p_{ij} = \hat{x}_i \cdot \hat{x}_j + \hat{y}_i \cdot \hat{y}_j. \quad (8.13)$$

Теперь можно составить логарифмическую функцию правдоподобия:

$$L = \sum_{i=1}^k \sum_{j=1}^k (p_{ij} - n_{ij} \cdot \ln p_{ij}). \quad (8.14)$$

Условия минимума этой функции составят систему из $2k$ уравнений:

$$\begin{aligned} \frac{\partial L}{\partial \hat{x}_i} &= \sum_{j=1}^k \hat{x}_j \cdot \left(2 - \frac{n_{ij} + n_{ji}}{p_{ij}}\right) = 0, \quad i = 1, 2, \dots, k, \\ \frac{\partial L}{\partial \hat{y}_i} &= \sum_{j=1}^k \hat{y}_j \cdot \left(2 - \frac{n_{ij} + n_{ji}}{p_{ij}}\right) = 0, \quad i = 1, 2, \dots, k. \end{aligned} \quad (8.15)$$

Однако, записав такие условия минимума, мы «забыли» ограничения на переменные – они не могут быть отрицательными. Если же формально минимум попадает по какой-то переменной в запрещённую область, то в силу непрерывности функции правдоподобия её минимальное значение по данной переменной будет достигаться на границе разрешённой области, т. е. при значении соответствующей переменной, равном нулю. Тогда можно модифицировать условия минимума так, чтобы они были справедливы для любого случая:

$$\begin{aligned} \sum_{j=1}^k \hat{x}_i \hat{x}_j \cdot \left(2 - \frac{n_{ij} + n_{ji}}{p_{ij}}\right) &= 0, \quad i = 1, 2, \dots, k, \\ \sum_{j=1}^k \hat{y}_i \hat{y}_j \cdot \left(2 - \frac{n_{ij} + n_{ji}}{p_{ij}}\right) &= 0, \quad i = 1, 2, \dots, k. \end{aligned} \quad (8.16)$$

Если сложить оба уравнения при фиксированном « i », то получим следующее условие:

$$\sum_{j=1}^k p_{ij} = \frac{1}{2} \sum_{j=1}^k (n_{ij} + n_{ji}). \quad (8.17)$$

Введём обозначение для полной экспериментальной статистики

$$N = \sum_{i=1}^k \sum_{j=1}^k n_{ij}$$

8.2. ИСПОЛЬЗОВАНИЕ КОРРЕЛЯЦИОННОЙ МАТРИЦЫ ДЛЯ РАЗДЕЛЕНИЯ С

и просуммируем последнее равенство по « i ». Тогда получим ещё одно соотношение:

$$N_x + N_y = N. \quad (8.18)$$

Допустим, мы получили одно решение $\hat{x}_i^{(0)}$, $\hat{y}_i^{(0)}$ системы уравнений (8.16), такое, что все $\hat{x}_i^{(0)}$, $\hat{y}_i^{(0)}$ больше нуля. Тогда легко убедиться, что имеется бесконечно много решений, зависящих от параметра θ :

$$\hat{x}_i = \hat{x}_i^{(0)} \cos \theta + \hat{y}_i^{(0)} \sin \theta; \quad \hat{y}_i = \hat{y}_i^{(0)} \cos \theta - \hat{x}_i^{(0)} \sin \theta, \quad (8.19)$$

а интервал возможных значений параметра θ , на котором функция правдоподобия равна одной и той же константе, определяется требованием неотрицательности всех переменных:

$$-\arctan \left(\min_i \frac{\hat{x}_i^{(0)}}{\hat{y}_i^{(0)}} \right) \leq \theta \leq \arctan \left(\min_i \frac{\hat{y}_i^{(0)}}{\hat{x}_i^{(0)}} \right). \quad (8.20)$$

Соответствующая оценка количества событий первого типа равна

$$N_x = N - N_y = \left(\sqrt{N_x^{(0)}} \cos \theta + \sqrt{N_y^{(0)}} \sin \theta \right)^2. \quad (8.21)$$

Если в соответствии с методом максимального правдоподобия для оценки количества событий первого типа мы будем строить график логарифмической функции правдоподобия в зависимости от N_x , то этот эффект проявится в виде «столика». Здесь нас ожидает ещё один сюрприз в проблеме раздельной оценки N_x и N_y . Очевидно, что при оптимизации всех переменных решение инвариантно относительно одновременной замены $N_x \leftrightarrow N_y$ и $x_i \leftrightarrow y_i$, а это означает, что графики функции правдоподобия по переменным N_x и N_y совпадают. При этом могут возникнуть два «столика» на одном уровне или один общий. Проиллюстрируем указанные эффекты несколькими численными экспериментами.

Пусть истинное число событий первого типа равно $\bar{N}_x = 40$, число событий второго типа $\bar{N}_y = 100$, и распределения вероятностей по $k = 4$ каналам гистограммы соответственно равны:

$$\bar{x}_i = \frac{1-q}{1-q^4} q^{i-1}; \quad \bar{y}_i = \frac{1-q}{1-q^4} q^{4-i}; \quad i = 1, 2, 3, 4. \quad (8.22)$$

Тогда используя генератор случайных чисел для распределения Пуассона, мы можем получить при $q = \frac{1}{2}$ результат «эксперимента», приведённый в табл. 8.1.

Table 8.1: Случайная выборка двумерного распределения двухчастичных событий (8.22) при $q = 1/2$.

$i \backslash j$	1	2	3	4
1	5	4	2	6
2	5	5	4	5
3	7	5	9	15
4	3	8	8	30

Table 8.2: Случайная выборка двумерного распределения двухчастичных событий (8.22) при $q = 0.3$.

$i \backslash j$	1	2	3	4
1	24	5	3	2
2	7	5	1	3
3	3	1	9	15
4	2	3	13	58

На рис.8.2 приведены графики логарифмической функции правдоподобия и оценки x_1, y_1 в зависимости от N_x .

На рис. 8.3 приведён также график зависимости оптимального N_y от N_x . Хорошо видно, что в пределах «столика» сумма $N_x + N_y$ постоянна.

В этом примере корреляционная матрица оказалась не очень эффективным средством для оценки числа частиц каждого сорта. Очевидно, что качество разделения этим методом зависит от степени «перекрывания» распределений по параметру разделения. «Улучшим» наш параметр разделения, выбрав меньшее значение $q = 0.3$, и повторим всю процедуру.

Получившееся двумерное распределение представлено в табл.8.2.

На рис.8.4 приведены графики логарифмической функции правдоподобия и оценки x_1, y_1 в зависимости от N_x .

График зависимости оптимального N_y от N_x приведён на рис.8.5. На этот раз в пределах «столиков» сумма $N_x + N_y$ также постоянна.

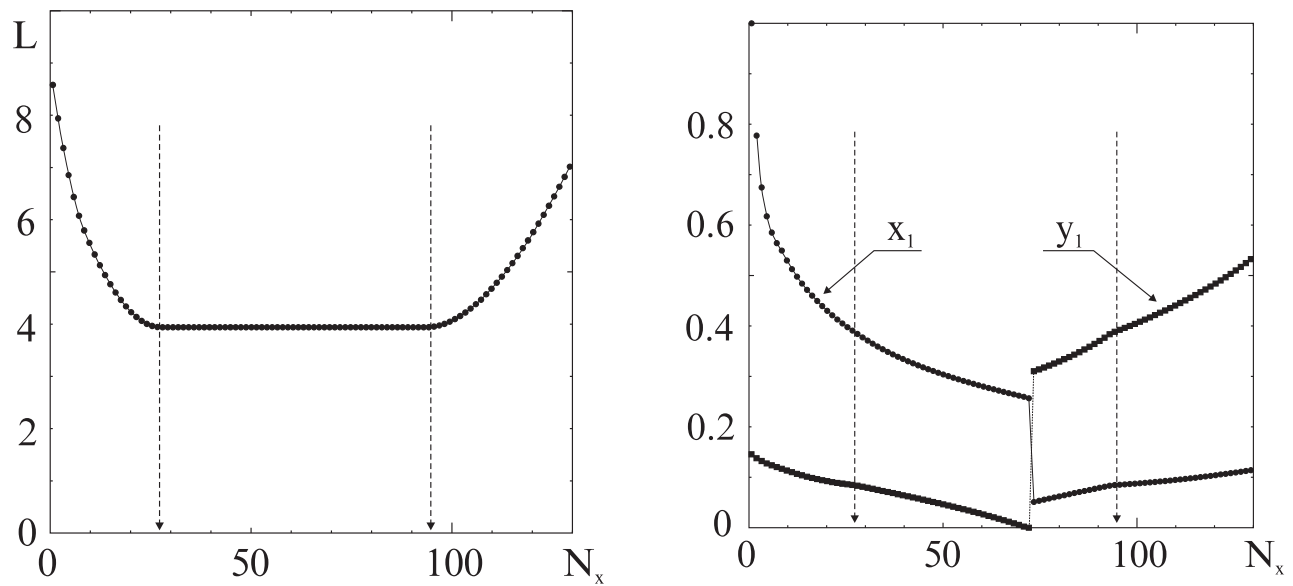


Figure 8.2: Графики логарифмической функции правдоподобия (слева) и оценок вероятности x_1, y_1 для «экспериментальных» данных из табл. 8.1.

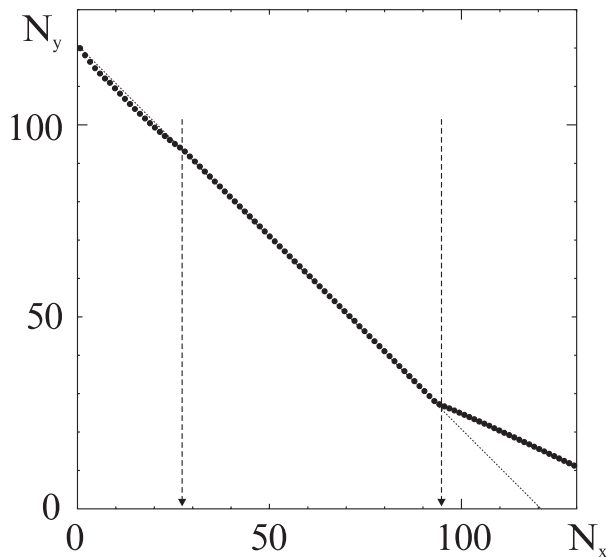


Figure 8.3: График зависимости оптимального значения N_y от N_x для «экспериментальных» данных из табл. 8.1. Хорошо видно, что в пределах «столика» сумма $N_x + N_y$ постоянна и равна $N = 121$.

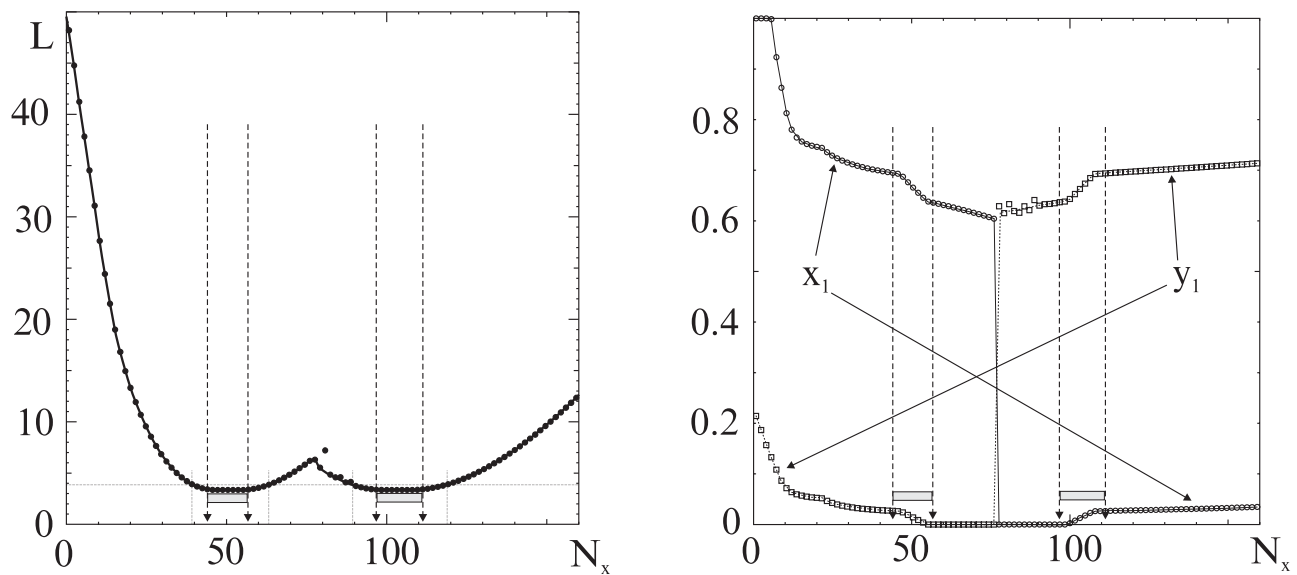


Figure 8.4: Графики логарифмической функции правдоподобия (слева) и оценок вероятности x_1, y_1 для «экспериментальных» данных из табл. 8.2.

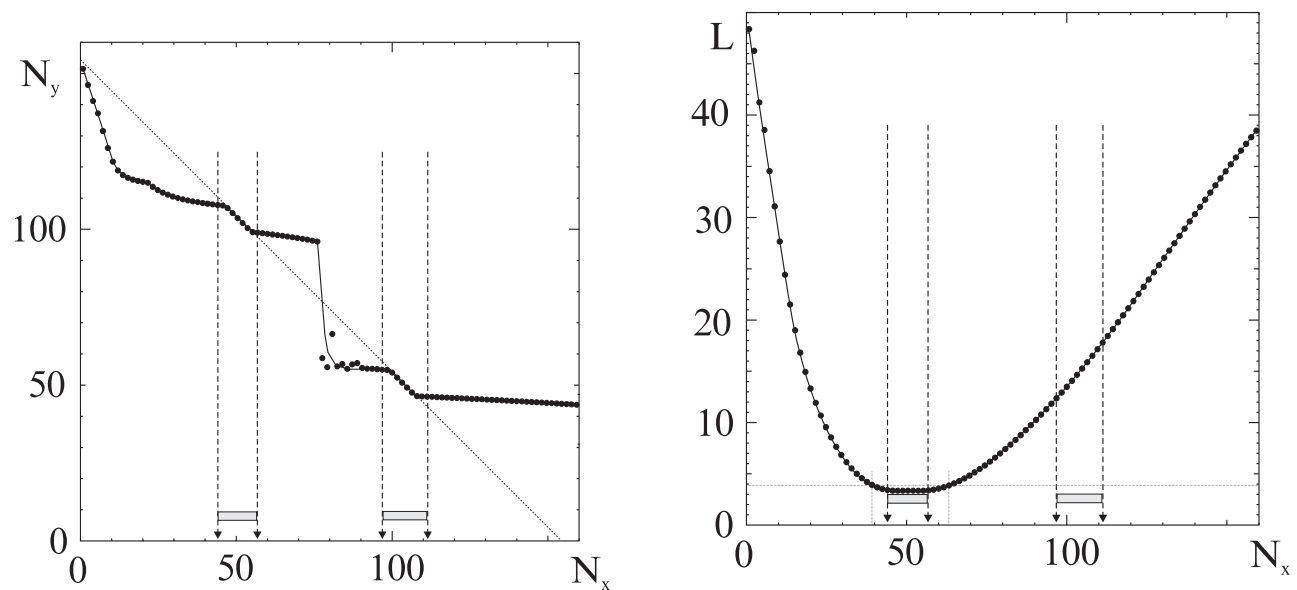


Figure 8.5: График зависимости оптимального значения N_y от N_x для «экспериментальных» данных из табл. 8.2. Хорошо видно, что в пределах «столиков» сумма $N_x + N_y$ постоянна и равна $N = 154$.

Figure 8.6: График логарифмической функции правдоподобия для «экспериментальных» данных из табл. 8.2. Для определения частиц первого сорта использовано условие $x_1 + x_2 > y_1 + y_2$.

8.2. ИСПОЛЬЗОВАНИЕ КОРРЕЛЯЦИОННОЙ МАТРИЦЫ ДЛЯ РАЗДЕЛЕНИЯ С

По графику функции правдоподобия на рис. 8.4 можно получить оценки числа событий:

$$N_x = 51 \pm 12; \quad N_y = 104 \pm 14. \quad (8.23)$$

Полученные оценки ошибок существенно больше статистических и автоматически включают систематическую ошибку метода. Определение систематической ошибки работает так, что если параметр разделения имеет высокую эффективность разделения, то систематическая добавка в виде «столика» может совсем исчезнуть. В меньшей степени эффект «столика» зависит от количества каналов в двумерной гистограмме — увеличение числа каналов немного улучшает разделение, но при этом увеличивается и число переменных, по которым надо минимизировать функцию правдоподобия. Здесь надо искать компромисс.

На первый взгляд неожиданное явление — тождественность графиков функции правдоподобия для первого сорта событий и для второго, — имеет простое объяснение. Мы не формализовали, что называть событиями первого сорта. Любое математическое условие, определяющее, что называть событиями первого сорта, сразу снимает эту проблему, и графики функций правдоподобия по N_x и N_y становятся разными. На рис. 8.6 представлен график $L(N_x)$ при условии

$$x_1 + x_2 > y_1 + y_2. \quad (8.24)$$

Второй «столик» при этом исчез. На аналогичном графике для N_y исчезнет другой «столик».

Естественно, в этом методе может быть несколько источников систематической ошибки, которые надо специально исследовать: присутствие третьего типа событий, различие распределений по параметру разделения для первой и второй частицы в событии, фон от многочастичных событий, в которых зарегистрировано только две из них, и т. п.

Chapter 9

Проверка гипотез. Критерии согласия теории с экспериментальными данными

В процессе обработки эксперимента может возникнуть вопрос, насколько согласуются полученные экспериментальные данные с теоретическими предсказаниями. Однозначного ответа на этот вопрос не существует, проверять согласие можно различными статистическими критериями. Наиболее популярным в экспериментальной физике является критерий χ^2 .

9.1 Критерий χ^2

Пусть экспериментальные измерения обозначены $y_i, i = 1, \dots, n$, ошибки измерений σ_i и теоретические предсказания – $y_i^{(th)}$. величиной χ^2 называется

$$\chi^2 = \sum_{i=1}^n \frac{(y_i - y_i^{(th)})^2}{\sigma_i^2}. \quad (9.1)$$

Число n принято называть числом степеней свободы. Предположим, что величины y_i распределены по закону Гаусса вокруг своего среднего значения $y_i^{(th)}$ со среднеквадратичным отклонением σ_i . Тогда оказывается, что величина χ^2 имеет стандартное распределение, вид которого зависит только от числа слагаемых n . Функция распределения вероятностей по величине χ^2 легко получается:

$$F_n(\chi^2) = A_n \cdot (\chi^2)^{\frac{n}{2}-1} e^{-\frac{\chi^2}{2}}, \quad (9.2)$$

где

$$A_n = \frac{1}{2^{n/2} \Gamma(n/2)} = \begin{cases} \frac{1}{2^k \cdot (k-1)!}, & n = 2k, \\ \frac{k! \cdot 2^{k-\frac{1}{2}}}{(2k)! \sqrt{\pi}}, & n = 2k + 1. \end{cases}$$

Среднее значение χ^2 и дисперсия равны

$$\langle \chi^2 \rangle = n, \quad D(\chi^2) = 2n. \quad (9.3)$$

Характеристическая функция (см. разл. 3.1) тоже легко вычисляется:

$$\varphi(t) = \frac{1}{\sqrt{1 - 2it}}^n. \quad (9.4)$$

Наиболее вероятное значение χ^2 всегда меньше среднего и равно $(n - 2)$ для $n > 2$ (для одной и двух степеней свободы производная функции распределения нигде не обращается в нуль, поэтому поиск наиболее вероятного требует специального анализа и в обоих случаях приводит к значению 0).

Так как вероятность получить какое-нибудь точное значение при непрерывном распределении вероятностей строго равна нулю, то для анализа согласия эксперимента с теорией используется интегральное распределение $P_n(\chi^2)$ – вероятность зарегистрировать значение χ^2 , превышающее то, которое зарегистрировано в данном эксперименте (ещё её кратко называют уровнем доверия, по-английски – *Confidence Level*). Если эта вероятность очень мала, то согласие плохое. Однако не надо думать, что если уровень доверия в точности равен 1 или очень близок к 1, то согласие хорошее. Это скорее всего свидетельствует о неправильной оценке ошибок σ_i . Учитывая последнее замечание, можно считать уровнем доверия $\min\{P_n(\chi^2), 1 - P_n(\chi^2)\}$ или вводить «доверительный интервал» (χ_1^2, χ_2^2) для уровня доверия $P < 0,5$ такой, что $P_n(\chi_1^2) = 1 - P$ и $P_n(\chi_2^2) = P$.

Вычисление вероятностей $P_n(\chi^2)$ – не очень простое занятие, поэтому обычно пользуются таблицами значений или графиками. На рис. 9.1 приведены графики $P_n(\chi^2)$ для некоторых значений n .

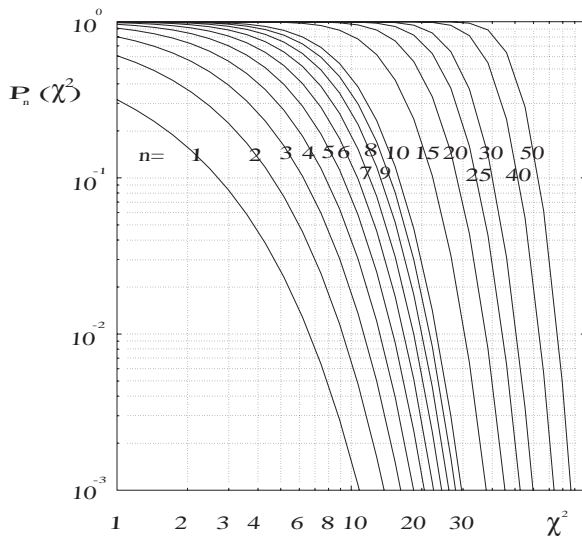


Figure 9.1: Графики функции $P_n(\chi^2)$ для некоторых значений n

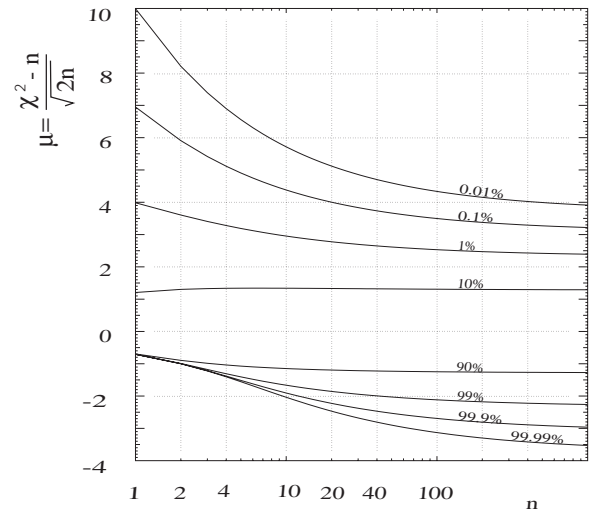


Figure 9.2: Доверительные интервалы для $\mu = (\chi^2 - n)/\sqrt{2n}$ для нескольких уровней доверия в зависимости от числа степеней свободы n

Может возникнуть необходимость получить значение $P_n(\chi^2)$ для значения n , которого нет на графике. Тогда можно воспользоваться рекуррентной формулой

$$P_n(\chi^2) = P_{n-2}(\chi^2) + \frac{B_n}{(n-2)!!} e^{-\chi^2/2} (\chi^2)^{\frac{n-2}{2}}, \quad (9.5)$$

где

$$B_n = \begin{cases} 1, & \text{если } n \text{ — чётное,} \\ \sqrt{\frac{2}{\pi}}, & \text{если } n \text{ — нечётное.} \end{cases}$$

Используя это соотношение, можно представить $P_n(\chi^2)$ в виде суммы.

$$P_n(\chi^2) = \begin{cases} e^{-\frac{\chi^2}{2}} \times \left(\frac{(\chi^2)^{\frac{n-2}{2}}}{(n-2)!!} + \frac{(\chi^2)^{\frac{n-4}{2}}}{(n-4)!!} + \dots + \frac{\chi^2}{2} + 1 \right), \\ \text{при } n \text{ — чётном,} \\ \sqrt{\frac{2}{\pi}} \times e^{-\frac{\chi^2}{2}} \times \left(\frac{(\chi^2)^{\frac{n-2}{2}}}{(n-2)!!} + \frac{(\chi^2)^{\frac{n-4}{2}}}{(n-4)!!} + \dots + \frac{(\chi^2)^{\frac{1}{2}}}{1} \right) + P_1(\chi^2), \\ \text{при } n \text{ — нечётном,} \end{cases} \quad (9.6)$$

$$P_1(\chi^2) = \sqrt{\frac{2}{\pi}} \int_{\sqrt{\chi^2}}^{\infty} e^{-\frac{\lambda^2}{2}} d\lambda.$$

Наконец, при больших n можно использовать тот факт, что отдельные слагаемые в χ^2 статистически независимы и имеют одинаковое распределение вероятностей. Тогда в соответствии с центральной предельной теоремой при больших n распределение вероятностей по χ^2 должно стремиться к гауссову распределению. Учитывая, что $\langle \chi^2 \rangle = n$, $D(\chi^2) = 2n$, получаем, что переменная $\mu = \frac{\chi^2 - n}{\sqrt{2n}}$ при больших n распределена по закону Гаусса. На рис. 9.2 представлены доверительные интервалы по μ в зависимости от n для некоторых уровней доверия.

Если при вычислении теоретических предсказаний какое-то количество параметров теоретической модели подбиралось так, чтобы минимизировать χ^2 (это эквивалентно применению метода максимального правдоподобия оценки параметров модели для Гауссового распределения измеряемых величин), то очевидно, что результаты проверки на согласие теории с экспериментом будут неправильными. В первом приближении положение можно исправить, уменьшив число степеней свободы на количество оптимизируемых параметров.

Этот рецепт строго справедлив в случае линейной зависимости ожидаемого значения от параметров. Действительно, пусть в формуле (9.1) теоретические значения $y^{(th)}$ следующим образом зависят от оптимизируемых параметров:

$$y_i^{(th)} = a_i + \sum_{j=1}^k r_{i,j} \sigma_i \alpha_j, \quad i = 1, 2, \dots, n, \quad (9.7)$$

где $k < n$, a_i — математическое ожидание y_i , α_k — оптимизируемые параметры. Безразмерные случайные величины x_i , распределённые по

нормальному закону, остались теми же самыми, но вместо χ^2 теперь используется результат минимизации

$$\lambda = \min_{\alpha_j} \chi^2 = \min_{\alpha_j} \sum_{i=1}^n \left(x_i - \sum_{j=1}^k r_{i,j} \alpha_j \right)^2. \quad (9.8)$$

Если ранг прямоугольной матрицы, составленной из коэффициентов $r_{i,j}$, равен k , то это означает, что столбцы матрицы в n -мерном пространстве образуют собой k линейно независимых векторов \mathbf{R}_j . Тогда в пространстве переменных x_i можно найти такое ортогональное преобразование, в базисе которого первые $(n - k)$ векторов ортогональны ко всем векторам \mathbf{R}_j . Если преобразованием поворота перейти в этот базис, то новые случайные переменные ξ_i останутся статистически независимыми и распределёнными по нормальному закону, однако, величина λ теперь будет записана как

$$\lambda = \min_{\alpha_j} \left[\sum_{i=1}^{n-k} \xi_i^2 + \sum_{i=n-k+1}^n \left(\xi_i - \sum_{j=1}^k \tilde{r}_{i,j} \alpha_j \right)^2 \right] = \sum_{i=1}^{n-k} \xi_i^2. \quad (9.9)$$

Последняя запись означает, что случайная величина λ является суммой квадратов $(n - k)$ случайных величин, распределённых по нормальному закону, и, значит, имеет распределение χ^2 для $(n - k)$ степеней свободы.

Чтобы убедиться, что это не всегда так, рассмотрим простой пример. Пусть зависимость теоретической модели от параметра α такова, что при двух измеряемых величинах x_1, x_2 величина λ равна

$$\lambda = \min_{\alpha} \chi^2 = \min_{\alpha} \left[(x_1 - \alpha)^2 + (x_2 - |\alpha|)^2 \right]. \quad (9.10)$$

Если в пространстве случайных величин x_1, x_2 сделать преобразование поворота

$$x_1 = \frac{\xi_1 + \xi_2}{\sqrt{2}}; \quad x_2 = \frac{\xi_1 - \xi_2}{\sqrt{2}} \iff \xi_1 = \frac{x_1 + x_2}{\sqrt{2}}; \quad \xi_2 = \frac{x_1 - x_2}{\sqrt{2}}, \quad (9.11)$$

то переменная λ при различных значениях ξ_1, ξ_2 равна

$$\lambda = \begin{cases} \xi_1^2, & \text{если } \xi_1 < 0, \xi_2 < 0, \\ \xi_2^2, & \text{если } \xi_1 > 0, \xi_2 > 0, \\ \xi_1^2 + \xi_2^2, & \text{если } \xi_1 < 0, \xi_2 > 0, \\ \min\{\xi_1^2; \xi_2^2\}, & \text{если } \xi_1 > 0, \xi_2 < 0, \end{cases} \quad (9.12)$$

Теперь мы можем найти интегральную функцию распределения вероятностей по λ :

$$P(\lambda < \lambda_0) = \frac{1}{2\pi} \int_{\lambda < \lambda_0} e^{-\frac{\xi_1^2 + \xi_2^2}{2}} d\xi_1 d\xi_2 \quad (9.13)$$

Используя обозначение для интеграла вероятности Гаусса

$$\Phi(x) = \sqrt{\frac{2}{\pi}} \int_0^x e^{-\frac{t^2}{2}} dt, \quad (9.14)$$

можно записать ответ в таком виде:

$$P(\lambda < \lambda_0) = \Phi(\sqrt{\lambda_0}) + \frac{1}{4} \left[1 - e^{-\frac{\lambda_0}{2}} - \Phi^2(\sqrt{\lambda_0}) \right]. \quad (9.15)$$

Отсюда легко получается дифференциальная функция распределения вероятностей по λ :

$$F(\lambda) = \frac{1}{\sqrt{2\pi\lambda}} e^{-\frac{\lambda}{2}} \cdot \left[1 - \frac{1}{2} \Phi(\sqrt{\lambda}) \right] + \frac{1}{8} e^{-\frac{\lambda}{2}}, \quad (9.16)$$

что не совпадает с распределением χ^2 ни с каким числом степеней свободы. И хотя наиболее вероятное значение равно нулю, так же как для χ^2 с одной степенью свободы (интегрируемая расходимость при $\lambda = 0$), но уже среднее значение и дисперсия отличаются от соответствующих значений:

$$\langle \lambda \rangle = 1 + \frac{1}{4} - \frac{1}{2\pi} \approx 1,091; \quad D(\lambda) = 2 + \frac{11}{16} - \frac{3}{4\pi} - \frac{1}{4\pi^2} \approx 2,423. \quad (9.17)$$

Всё же это распределение ближе к χ^2 с одной степенью свободы, чем с двумя.

9.2 Критерий знаков

В качестве дополнительного критерия согласия к критерию χ^2 можно предложить критерий знаков [12]. Дело в том, что критерий χ^2 чувствителен только к абсолютной величине отклонения измеренного значения от теории. В то же время естественно ожидать примерно равного числа положительных и отрицательных разностей $y_i - y_i^{(th)}$.

Пусть в эксперименте с n измерениями зарегистрировано m положительных разностей $y_i - y_i^{(th)}$. Если вероятности положительной и отрицательной разности равны и разные экспериментальные точки статистически независимы, то распределение вероятностей по числу m является биномиальным (см. разд. 3.2.4). По аналогии с критерием χ^2 назовём доверительной вероятностью $Z_n(m)$ (или уровнем доверия) вероятность зарегистрировать большее, чем m , число положительных разностей

$$Z_n(m) = \sum_{i=m+1}^n \frac{n!}{2^i i! (n-i)!}. \quad (9.18)$$

Очевидно, что о плохом согласии свидетельствует не только слишком большое, но и слишком малое количество положительных разностей. Учитывая симметрию распределения по m и тождественность распределений по количеству положительных и отрицательных разностей, можно объединить проверки на «слишком большое» и «слишком малое» количество положительных разностей, выбирая в качестве проверяемого числа m максимальное из двух чисел – количество положительных или отрицательных разностей $y_i - y_i^{(th)}$. Представляется более удобным для изображения на графиках измерять отклонение m от его среднего значения $n/2$ в единицах стандартного отклонения: $\sigma_m = \sqrt{D(m)} = \sqrt{n/4}$. На рис. 9.3 приведены графики уровня доверия для нескольких значений n .

9.3 Распределение Стьюдента

Распределением Стьюдента называется распределение с функцией плотности вероятности [12, 58]:

$$f_\nu(t) = \frac{C_\nu}{\left(1 + \frac{t^2}{\nu}\right)^{\frac{\nu+1}{2}}}, \quad C_\nu = \frac{(\nu-1)!!}{\sqrt{\nu}(\nu-2)!!} \times \begin{cases} \frac{1}{2}, & \nu - \text{чётное,} \\ \frac{1}{\pi}, & \nu - \text{нечётное,} \end{cases} \quad (9.19)$$

где ν называется количеством степеней свободы. Это распределение симметрично относительно нуля, поэтому все нечётные моменты

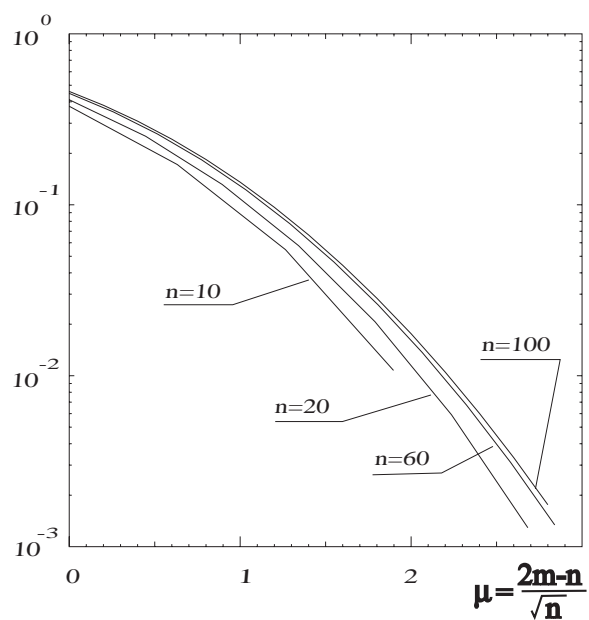


Figure 9.3: Уровень доверия по критерию знаков в зависимости от нормализованного числа $\mu = (2m - n)/\sqrt{n}$

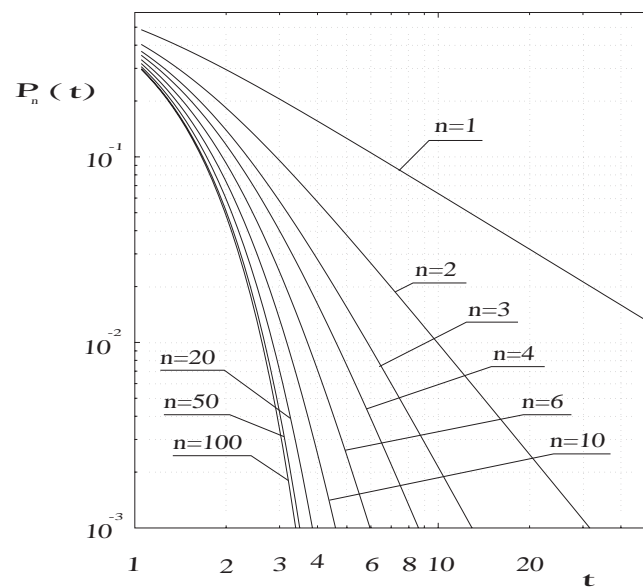


Figure 9.4: Уровень доверия для переменной t , имеющей распределение Стьюдента с n степенями свободы. Приведены графики для нескольких степеней свободы

распределения равны нулю, чётные моменты

$$M_{2k} = \frac{(2k)! \nu^k (\nu - 2k - 2)!!}{2^k \cdot k! (\nu - 2)!!}$$

существуют только для $2k < \nu$. Если устремить к бесконечности количество степеней свободы $\nu \rightarrow \infty$, то распределение асимптотически переходит в гауссово распределение со средним значением 0 и дисперсией 1, при этом все моменты тоже стремятся к соответствующим моментам гауссова распределения:

$$\lim_{\nu \rightarrow \infty} M_{2k} = \frac{(2k)!}{k! 2^k}.$$

Приведём несколько примеров случайных величин, имеющих распределение Стьюдента.

Пусть имеется несколько измерений одной и той же случайной величины $x_i, i = 1, \dots, n$, распределённой по гауссову закону со средним значением μ и дисперсией σ^2 . Выборочное среднее $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, выборочная дисперсия

$S^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$. Тогда величина $t = \sqrt{n} \frac{\bar{x} - \mu}{S}$ имеет распределение Стьюдента с $\nu = n - 1$ степенями свободы. Также распределение Стьюдента с $\nu = n - 1$ степенями свободы имеет и каждая из величин $t_i = \frac{x_i - \mu}{S}$.

Эти примеры, однако, имеют невысокую практическую ценность, так как часто истинное значение среднего μ неизвестно. Можно построить случайную величину из наблюдаемых x_i такую, что она имеет распределение Стьюдента с $\nu = n - 2$ степенями свободы [12]:

$$t_i = \frac{(x_i - \bar{x}) \sqrt{n(n-2)}}{\sqrt{S^2(n-1)^2 - (x_i - \bar{x})^2 n}}, \quad \nu = n - 2 \quad (9.20)$$

Рассмотрим ещё один пример. Пусть проведено n измерений одной и той же величины $x_i, i = 1, \dots, n$ с разными точностями σ_i . Общую степень согласия всех экспериментальных данных можно проверить по критерию χ^2 (см. разд. 9.1), используя величину

$$\chi_\nu^2 = \sum_{i=1}^n \frac{(x_i - \bar{x}^{(n)})^2}{\sigma_i^2}, \quad \text{где } \nu = n - 1, \quad \bar{x}^{(n)} = \frac{1}{s_n} \sum_{i=1}^n \frac{x_i}{\sigma_i^2}, \quad s_n = \sum_{i=1}^n \frac{1}{\sigma_i^2}. \quad (9.21)$$

Для проверки степени согласия одного измерения со всеми остальными можно воспользоваться величиной, имеющей распределение Стьюдента с числом степеней свободы $\nu = n - 2$ (для простоты будем считать, что последнее измерение x_n мы сравниваем с остальными):

$$t = \frac{(x_n - \bar{x}^{(n-1)})}{\sigma_n} \sqrt{\frac{(n-2)s_{n-1}}{s_n \sum_{i=1}^{n-1} \frac{(x_i - \bar{x}^{(n-1)})^2}{\sigma_i^2}}}. \quad (9.22)$$

Степень согласия по распределению Стьюдента $f_\nu(t)$ можно определить по аналогии с другими критериями согласия как вероятность переменной t по абсолютной величине превысить наблюдаемое значение:

$$P(t) = \int_{-\infty}^{-|t|} f_\nu(\xi) d\xi + \int_{|t|}^{\infty} f_\nu(\xi) d\xi. \quad (9.23)$$

Графики зависимости уровня доверия от t для нескольких степеней свободы показаны на рис. 9.4.

9.4 Критерий Колмогорова

Мощным критерием согласия произвольного экспериментального распределения с теоретическим распределением вероятностей является критерий Колмогорова [19]. Этот критерий применим к любому непрерывному распределению вероятностей (пример так называемой *непараметрической статистики*). Для использования критерия Колмогорова для сравнения экспериментального распределения x_i , $i = 1, \dots, n$ и теоретического $f(x)$ вычисляются две интегральные функции распределения: теоретическая

$$F(x) = \int_{-\infty}^x f(\xi) d\xi \quad (9.24)$$

и выборочная

$$F_n(x) = \begin{cases} 0, & x < x_1, \\ \frac{k}{n}, & x_k < x < x_{k+1}, \quad k = 1, 2, \dots, n-1, \\ 1, & x > x_n \end{cases} \quad (9.25)$$

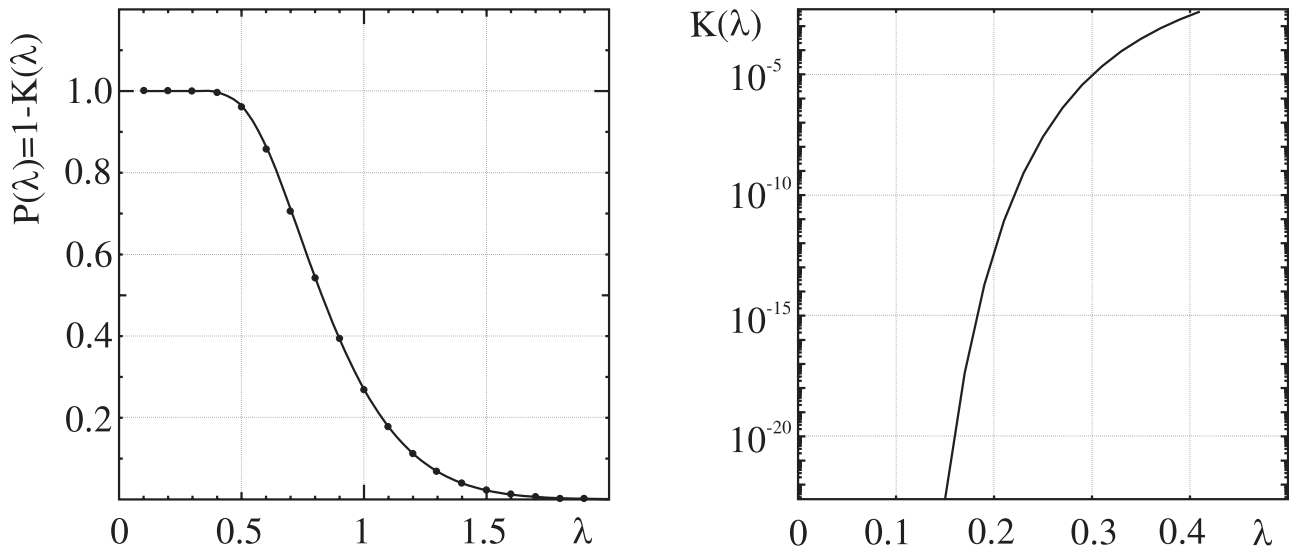


Figure 9.5: Графики функций $P(\lambda)$ (слева) и $K(\lambda)$ (справа) для критерия согласия Колмогорова

Максимальное отклонение выборочной функции распределения от теоретической можно использовать в качестве критерия согласия. Распределение вероятностей по такому отклонению не зависит от вида теоретической функции, а только от числа событий n в выборке. К сожалению, точное распределение вероятностей для каждого конечного значения n может быть рассчитано только численными методами, поэтому обычно используют асимптотическую функцию распределения (это, собственно, и называется критерием Колмогорова). Оказывается, что распределение вероятностей по величине

$$\lambda = \sqrt{n} \times \max_x |F_n(x) - F(x)| \quad (9.26)$$

в пределе $n \rightarrow \infty$ описывается формулой

$$\begin{aligned} \lim_{n \rightarrow \infty} W \left(\sqrt{n} \times \max_x |F_n(x) - F(x)| < \lambda \right) = \\ = K(\lambda) = \sum_{k=-\infty}^{\infty} (-1)^k \exp(-2k^2 \lambda^2) \end{aligned} \quad (9.27)$$

Величина вероятности $P(\lambda) = 1 - K(\lambda)$ тем меньше, чем больше зарегистрировано отклонение выборочной функции распределения от теоретической, и может быть названа уровнем доверия по критерию Колмогорова. Графики функций $K(\lambda)$ и $P(\lambda)$ приведены на рис.9.5. Во

многих сборниках таблиц математической статистики приводятся таблицы значений функций $P(\lambda)$ или $K(\lambda)$ (см., например, [73]).

Критерий Колмогорова можно использовать и для проверки предположения, что две выборочные функции $F_{n_1}(x)$ и $F_{n_2}(x)$ соответствуют одному и тому же распределению вероятностей. Для двух выборок n_1 и n_2 событий одного и того же распределения величина

$$\lambda = \sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}} \times \max_x |F_{n_1}(x) - F_{n_2}(x)| \quad (9.28)$$

в пределе $n_1, n_2 \rightarrow \infty$ имеет распределение вероятностей Колмогорова, и для степени согласия двух выборок можно пользоваться функцией $P(\lambda)$.

В работе [72] на нескольких примерах проведено сравнение мощности критериев Колмогорова и χ^2 . Как и ожидалось, невозможно сделать вывод, что какой-то из этих критериев во всех ситуациях более мощный, чем другой, естественно, не считая тех случаев, когда критерий χ^2 просто нельзя применить.

Проиллюстрируем это утверждение двумя примерами. В каждом примере проверяется «ложная» теория, предполагающая, что измерения в 10 «экспериментальных» точках имеют Гауссово распределение со средним значением 0 и ср.-кв. отклонением σ . По каждому критерию установим пороговое значение степени доверия 5%, и если степень доверия по данному критерию становится меньше этого порога, гипотеза признаётся «ложной». Вероятность отвергнуть «ложную» гипотезу будем называть мощностью критерия.

Первый пример. Пусть истинное распределение вероятностей действительно Гауссово со ср.-кв. отклонением σ , но истинное среднее значение для измерения номер i равно

$$\bar{x}_i = \frac{i - 5.5}{4.5} \cdot \sigma \cdot \mu, \quad i = 1, 2, \dots, 10, \quad (9.29)$$

где μ — параметр отклонения истинной теории от ложной. При $\mu = 0$ гипотеза становится правильной, и мощность любого критерия в соответствии с выбранным порогом отбраковки должна равняться 0,05. На рис. 9.6 представлена зависимость мощности критериев χ^2 и Колмогорова от параметра μ . Критерий χ^2 имеет в этом случае заметно более высокую мощность.

Второй пример. Пусть «ложная» гипотеза та же самая — 10 независимых измерений с нулевым средним и ср.-кв. отклонением σ . Истинные значения

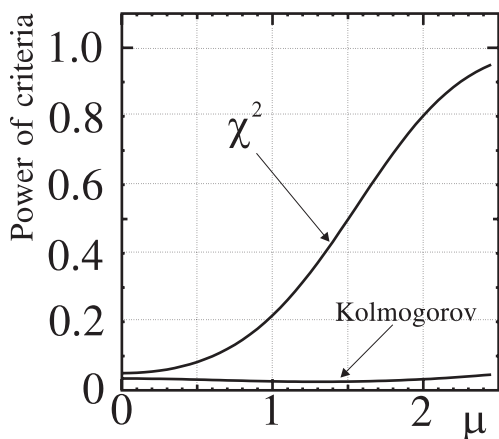


Figure 9.6: Мощность критериев Колмогорова и χ^2 для первой модели отличия истинной теории от ложной (9.29).

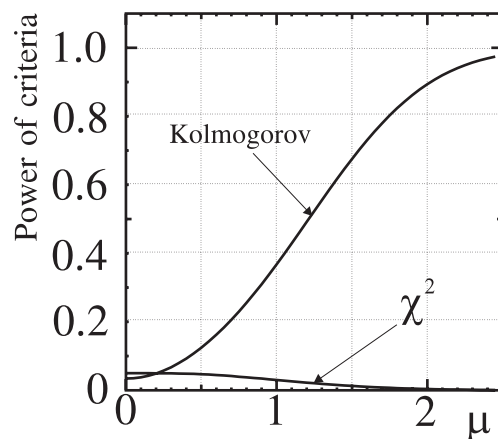


Figure 9.7: Мощность критериев Колмогорова и χ^2 для второго примера отклонения истинной теории от ложной (9.30).

этих величин равны

$$\bar{x}_i = \begin{cases} -\frac{\sigma \cdot \mu}{\sqrt{1+\mu^2}}, & i = 1, \\ \frac{\sigma \cdot \mu}{\sqrt{1+\mu^2}}, & i = 2, 3, \dots, 10, \end{cases} \quad \sigma_i = \frac{\sigma}{\sqrt{1+\mu^2}}, \quad i = 1, 2, \dots, 10. \quad (9.30)$$

В этом случае преимущество критерия Колмогорова бесспорно.

9.5 Критерий Мизеса – Смирнова

Другим примером непараметрической статистики является критерий согласия Мизеса – Смирнова [74, 75]. Здесь в качестве меры отличия выборочной функции от теоретической используется среднеквадратичное отклонение

$$M = n \times \int_0^1 [F(\xi) - F_n(\xi)]^2 dF(\xi). \quad (9.31)$$

Как и в случае с критерием Колмогорова, распределение вероятностей по величине M не зависит от $F(\xi)$, а только от числа отсчетов n . Интегральная функция распределения вероятностей

$$\Phi_n(\mu) = W(M > \mu) \quad (9.32)$$

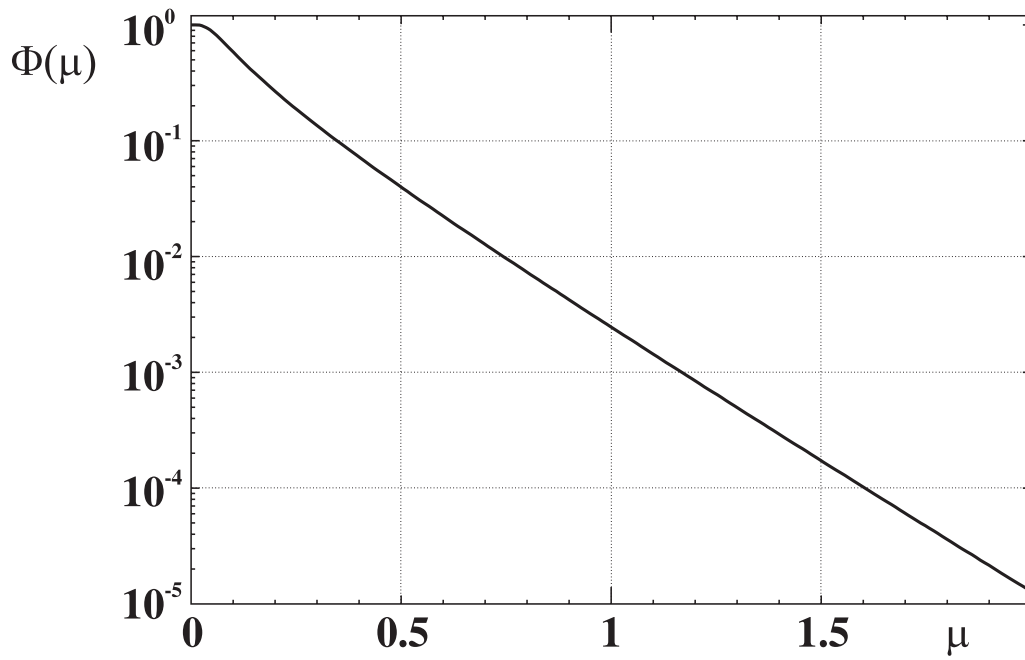


Figure 9.8: Интегральная функция распределения вероятностей $\Phi(\mu)$

может использоваться в качестве степени согласия экспериментального распределения с теорией. Однако, как и в случае с критерием Колмогорова, вычисление этой функции распределения представляет значительные трудности, и на практике используется предельная функция распределения

$$\Phi(\mu) = \lim_{n \rightarrow \infty} \Phi_n(\mu). \quad (9.33)$$

Вычисление этой функции тоже непростая задача, однако, во многих сборниках статистических таблиц приводятся таблицы значений $\Phi(\mu)$ или $1 - \Phi(\mu)$ (см., например, [73]). Кроме того, можно воспользоваться характеристической функцией [75], полученной Смирновым для этого распределения,

$$\varphi(t) = \int \exp(it\mu) d\Phi(\mu) = \sqrt{\frac{\sqrt{2it}}{\sin \sqrt{2it}}}, \quad (9.34)$$

и с помощью формулы обращения вычислять $\Phi(\mu)$. Полученный таким образом график функции $\Phi(\mu)$ приведен на рис. 9.8. Скорость сходимости $\Phi_n(\mu)$ к $\Phi(\mu)$ демонстрирует рис. 9.9.

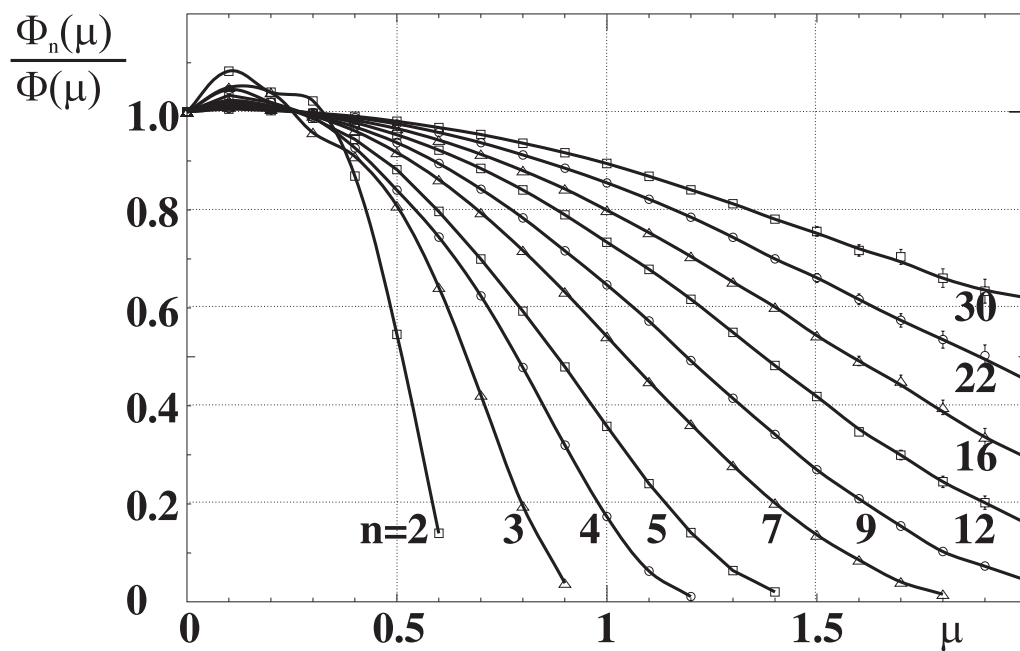


Figure 9.9: Отношение интегральной функции распределения $\Phi_n(\mu)$ к $\Phi(\mu)$ при разных n . Значения функции $\Phi_n(\mu)$ вычислялись методом Монте–Карло (стат. ошибка на графике не приведена)

Chapter 10

Система аналитических вычислений REDUCE

Физику-экспериментатору иногда приходится выполнять не только численные расчёты, но и проводить трудоёмкие аналитические вычисления. Существенную помощь при этом может оказать система аналитических вычислений REDUCE [76, 77]. Мы не будем в данном разделе стремиться к полному описанию возможностей REDUCE ввиду ограниченности места, рассмотрим только самые простые команды REDUCE, позволяющие использовать его в разных областях. И совсем не будем касаться мощных средств для вычислений в физике элементарных частиц. Желающие более подробно ознакомиться с этими возможностями REDUCE кроме изучения полного руководства пользователя могут прочитать препринты [78, 79].

Система REDUCE является интерпретатором, а не компилятором. Это значит, что команды вводятся по очереди, анализируются на наличие синтаксических ошибок и, если нет ошибок, сразу же выполняются. Если при этом получается какой-то результат, то он может быть сразу напечатан. При введении новых определений и подстановок прежние вычисления автоматически не повторяются, для повторения надо выдать команды заново.

10.1 Числа, переменные, структура команд

Система REDUCE использует следующие символы:

- Буквы английского алфавита (в большинстве реализаций большие и маленькие буквы не различаются);
- Цифры от 0 до 9;
- Специальные символы:
! " " \$ % ' () * + , - . / : ; < > = { }.

Текст команд вводится в первых 72 позициях строки, запись команд бесформатная (не привязана к каким-либо позициям), между командами и операндами может быть произвольное количество пробелов. Конец команды отмечается символом «;» или «\$».

Целые числа вводятся как последовательность цифр, которым может предшествовать символ знака «+» или «-». В дробных числах можно использовать десятичную точку и символ десятичного порядка «E», за которым записывается порядок в виде целого числа со знаком или без знака. Первым символом числа (если есть знак, то после знака) должна быть цифра. В стандартном режиме дробные числа превращаются в обыкновенные дроби и запоминаются как отношение двух целых чисел. Нет ограничений на величину целого числа, кроме естественного ограничения по оперативной памяти ЭВМ, необходимой для хранения информации.

Идентификатором переменной может быть любая последовательность букв и цифр, начинающаяся с буквы. Максимальная длина может быть разной в разных реализациях, но длина в 24 символа всегда возможна. Есть целый ряд идентификаторов, зарезервированных системой, большинство из них являются операторами языка REDUCE или математическими функциями и не должны использоваться в качестве идентификаторов переменных:

*	**	+	.	/	:=	<	<=	=	>	>=
ABS	ACOS	ACOSH	ALGEBRAIC	AND	ANTISYMMETRIC					
ARGLLENGTH	ARRAY	ASIN	ASINH	ATAN	ATANH	BEGIN				
BYE	CLEAR	COEFF	COMMENT	CONS	CONT	COS				
COSH	COT	DEFINE	DEG	DEN	DEPEND	DET	DF			
DIFFERENCE	DILOG	DISPLAY	DO	E	ED	EDITDEF				
END	EPS	EQ	EQUAL	ERF	EXP	EXPINT	EXPR			
EXPT	FACTOR	FACTORIZE	FASLOUT	FEXPR	FLAGOP	FOR				
FORALL	FOREACH	G	GCD	GEQ	GO	GOTO	GREATERP			
I	IF	IN	INDEX	INFIX	INPUT	INT	INTEGER			
K!*	KORDER	LAMBDA	LCOF	LEQ	LESSP	LET	LINEAR			
LINELENGTH	LISP	LOAD	LOG	LTERM	MACRO					
MAINVAR	MASS	MAT	MATCH	MATRIX	MAX	MEMBER	MEMQ			
MIN	MINUS	MSHELL	NEQ	NIL	NODEPEND	NONCOM				
NOSPUR	NOT	NUM	OFF	ON	OPERATOR	OR				
ORDER	OUT	PART	PAUSE	PFACTORIZE	PI	PLUS				
PRECEDENCE	PRECISION	PROCEDURE	PRODUCT	QUIT	QUOTIENT					
REAL	REDERR	REDUCT	REMAINDER	REMFAC	REMIND	REPEAT				
RESULTANT	RETRY	RETURN	SAVEAS	SCALAR	SETMOD	SETQ				
SHARE	SHOWTIME	SHUT	SIN	SINH	SMACRO	SOLVE				

SPUR	SQRT	STRUCTR	SUB	SUM	SYMBOLIC	SYMMETRIC
T	TAN	TANH	TIMES	TP	TRACE	VARNAME
VECTOR	WEIGHT	WHERE	WHILE	WRITE	WS	WTLEVEL

Некоторые из зарезервированных идентификаторов обозначают особые числа, которые можно использовать в своих заданиях для REDUCE:

E – основание натуральных логарифмов $e \approx 2,718281828$

I – мнимая единица $i = \sqrt{-1}$, квадрат её автоматически заменяется на -1

NIL – синоним нуля

PI – константа $\pi \approx 3,141592654$.

Ввиду такого большого количества зарезервированных идентификаторов надо быть особенно осторожным при выборе имён переменных, так как вероятность случайного совпадения велика, а диагностика ошибок может быть не вполне ясной.

По умолчанию идентификаторам присваивается тип **SCALAR**, значением такой переменной может быть любое алгебраическое выражение. Но возможны и другие типы переменных, которые требуют специального описания. Такие типы переменных будут рассмотрены в соответствующих местах.

Строка символов, заключённая в двойные апострофы, не анализируется и не преобразуется, например, "This is a string". Такие строки, в основном, используются в сообщениях, выдаваемых в текст результатов.

Целые строки можно сделать комментариями, которые REDUCE будет игнорировать. Для этого есть две возможности:

- Оператор комментария **COMMENT** должен быть помещён перед комментарием и его действие кончается символом конца оператора «;» или «\$».
- Символ «%» превращает оставшуюся часть строки в комментарий (он может быть помещён и в начало строки, тогда вся строка является комментарием).

10.2 Выражения

Часто в качестве операндов используются выражения. Основным видом выражений являются алгебраические выражения, где наряду с числами и скалярными переменными используются знаки операций (запись

выражений очень близка к математической записи, а также к записи выражений в программах на Фортране):

+ – сложение

– – вычитание

* – умножение

/ – деление

** – возведение в степень.

Среди этих операций есть приоритетность в порядке исполнения: сложение и вычитание имеют наименьший приоритет, следом идут умножение и деление; наивысший приоритет имеет возведение в степень. Для изменения порядка выполнения операций и однозначного задания этого порядка в сомнительных случаях следует использовать пары круглых скобок.

Пример выражения:

$$A*B+C/(A+C-D/(A+C))**2+(D-C)**(A+B),$$

что эквивалентно математической записи

$$a \cdot b + \frac{c}{\left(a + c - \frac{d}{a+c}\right)^2} + (d - c)^{a+b}.$$

Логическое выражение имеет своим результатом ответ «да» или «нет». Логическое выражение может быть составлено следующими способами:

- Два алгебраических выражения, соединённых оператором отношения. Операторы отношения могут быть следующие:

> левое выражение больше правого

>= больше или равно

< меньше

<= меньше или равно

= равно.

- Два логических выражения, соединённых знаками логических операций (в виде слов):

OR – логическое «или»

AND – логическое «и»

NOT – логическое отрицание (используется в сочетании только с одним логическим выражением, следующим за ним).

- Логическая функция с аргументами, например:

NUMBERP(U) – определяет, является ли выражение U числом

FIXP(U) – определяет, является ли выражение U целым числом

EVENP(U) – определяет, является ли выражение U чётным числом.

Используются логические выражения в условных операторах и операторах цикла.

10.3 Операторы присвоения и подстановки

Одним из основных операторов языка REDUCE является оператор присвоения. Обозначается он двумя символами подряд: двоеточие и знак равенства «:=». Слева от кода оператора должно быть название переменной, справа – выражение. Переменная и выражение должны быть одного типа, в простейшем варианте – скалярного типа. Это значит, что выражение должно быть алгебраическим выражением с использованием скалярных переменных, а слева может быть или скалярная переменная, или элемент массива, или элемент матрицы, или имя оператора со списком параметров. Примеры:

$A1 := B+C;$

$BB(K,L,M) := X-2*Y;$

$K(3,5) := X**2-Y;$

Если оператор заканчивается символом «;» и не является элементом составного оператора, то значение, присваиваемое переменной, распечатывается. К сожалению, наиболее распространённая ошибка при записи этого оператора не диагностируется, так как является допустимой записью в других конструкциях. Это ошибочное использование знака равенства вместо оператора присвоения.

Одним из наиболее сложных операторов является оператор подстановки. В простейшем варианте он состоит из кода оператора LET и списка правил подстановки, перечисленных через запятую. Примеры:

$LET A=C, C=B+D, MU=A*C-LAMBDA;$

$LET A*B**2=0; LET A+B+C=D;$

Оператор подстановки, в отличие от оператора присвоения, не является исполняемым. Правила подстановки запоминаются непосредственно в том виде, как они были определены, для дальнейшего использования при вычислении выражений. Вычисление выражений начинается с того, что вместо переменных, которые имеют какие-либо значения, подставляются эти значения, и в соответствии с текущими правилами подстановки производятся замещения. После упрощения выражений (приведения подобных членов, сокращения общих множителей и т. д.) снова делается проверка, не появилось ли новых комбинаций переменных, которые надо заменять, и так до тех пор пока новые замены станут невозможными. Типичная ошибка при этом возникает, когда какая-либо переменная прямо или косвенно заменяется на неё же. Например, `LET A = A+B$`.

Отменить действие оператора подстановки можно оператором `CLEAR`, например:

```
CLEAR A,C,A*B**2;
```

Существует много других модификаций оператора подстановки, одна из наиболее полезных может быть продемонстрирована на примере:

```
FOR ALL X LET SIN(X)**2 = 1 - COS(X)**2;
```

Такая запись означает, что при упрощении выражения вторая степень синуса, какой бы ни был у него аргумент, заменяется на его выражение через косинус. Дело в том, что такие тригонометрические преобразования не определены по умолчанию, поэтому, удачно выбрав соответствующие подстановки, можно существенно упростить выражение.

10.4 Групповые и составные операторы

Несколько операторов, записанных подряд, можно объединить в групповой или составной оператор, заключив их в соответствующие «скобки». Для группового оператора скобками являются символы «<<» и «>>», для составного оператора – кодовые слова `BEGIN` и `END`. Такие операторы употребляются там, где по определению должен стоять один оператор: в условных операторах, операторах цикла и т. д. (примеры использования в разд. 10.5). Несмотря на большое сходство групповых и составных операторов, между ними есть и существенная разница. Вместо группового оператора всегда можно употребить составной, а наоборот не всегда возможно. Значением группового оператора является значение последнего оператора в группе, значением составного оператора является нуль, если последним оператором не будет оператор `RE-`

TURN, сопровождаемый выражением, значение которого надо считать значением составного оператора. Оператором RETURN внутри составного оператора можно прервать нормальный порядок исполнения операторов в составном операторе. Оператор RETURN может также находиться внутри оператора IF ... THEN ... ELSE ... в составном операторе. В составном операторе сразу после слова BEGIN можно описать локальные переменные, которые будут определены только внутри составного оператора, с помощью операторов описания типа SCALAR, INTEGER или REAL. Перед словом END знак конца оператора не надо ставить, а после него – обязательно. Для группового оператора – наоборот.

10.5 Условные операторы, операторы цикла и массивы

Основная форма условного оператора начинается со слова IF, затем должно быть логическое выражение, затем слово THEN и оператор, который надо выполнить, если истинно логическое выражение, например:

```
IF A=B THEN C:=A**2;
```

Условный оператор может быть продолжен после выражения записью слова ELSE и другого выражения, которое надо выполнить, если логическое выражение ложно. Например:

```
IF X=5 AND Y=0 THEN <<FF:=Q1; A:=B+C >> ELSE <<FF:=0; A:=-1;>>
```

Основная форма оператора цикла в REDUCE:

FOR *переменная* := *число* STEP *число* UNTIL *число* DO *оператор*,

например:

```
FOR I1:=1 STEP 3 UNTIL 300 DO A:=A*(I1+2);
```

Сочетание «STEP 1 UNTIL» может быть заменено на двоеточие, например,

```
FOR I1:=1:10 DO A:=A*I1;
```

В тех случаях, когда количество повторений неизвестно заранее, можно использовать следующую форму цикла:

WHILE *логическое выражение* DO *оператор*

Например, фрагмент задания для REDUCE

```
K1:=1$ TERM:=1$ SINR:=0$ WHILE ABS(TERM)>1/1000 DO <<SINR:=SINR+TERM; K1:= K1+2; TERM:=-TERM/K1;>>
```

вычислит синус угла в один радиан с точностью в 0,001. Логическое выражение оценивается заново каждый раз перед выполнением оператора, и цикл продолжается до тех пор, пока выражение истинно.

Противоположное использование логического выражения применяется в цикле следующего вида:

`REPEAT оператор UNTIL логическое выражение`

Здесь оператор выполняется, затем оценивается логическое выражение, и, если оно *ложно*, цикл повторяется.

Естественное применение в операторах цикла находят массивы. Переменная может быть объявлена массивом с помощью оператора `ARRAY`, например:

`ARRAY A1(2, 3), BB(3, 3, 3);`

В этой записи числа в скобках обозначают максимально возможное значение каждого индекса, минимальным значением подразумевается нуль. В момент объявления переменной массивом все его элементы полагаются равными нулю. Элементы массива могут использоваться во всех выражениях, а также находиться в левой части оператора присвоения.

10.6 Матрицы

Очень похожим на массивы объектом являются матрицы — двумерные массивы с некоторыми дополнительными возможностями. Переменная объявляется матрицей с помощью оператора `MATRIX`, например,

`MATRIX MT1(3, 3), A1(1, 3), ZZ;`

Каждый индекс изменяется от 1 до указанного значения. Если пределы изменения индексов не указаны, как в случае с матрицей `ZZ`, то они будут определены в последующих операторах присвоения. Используя матрицы, можно образовывать матричные выражения, как в общепринятой матричной алгебре. При этом можно использовать операции сложения «+», вычитания «-», умножения «*», деления «/», возведения в целую степень «**». При этом операции деления и возведения в целую степень (в том числе и в отрицательную) возможны только для квадратных матриц.

В матричных выражениях иногда удобно использовать запись матрицы без привлечения собственного идентификатора этой матрицы. Для этого используется оператор `MAT`, за которым в круглых скобках построчно перечисляются элементы матрицы. Например, первоначальное

определение матрицы из двух строк и трёх столбцов может быть следующим:

`MATRIX A1(2,3); A1:=MAT((1-L,0,3),(2,3,A+B));`

Имеется несколько удобных функций, использующих матричные выражения в качестве аргументов.

Детерминант квадратной матрицы может быть найден с помощью оператора `DET`, например, `DD:=DET(MS);`. След квадратной матрицы (сумма диагональных элементов) находится оператором `TRACE`, транспонированная матрица находится оператором `TR`.

10.7 Дифференцирование и интегрирование

Оператор дифференцирования `DF` позволяет находить частные производные алгебраических выражений. В круглых скобках после названия оператора вначале записывается выражение, которое надо дифференцировать, затем название переменной, по которой надо дифференцировать, затем можно указать число – порядок производной. Пример:

`DF(FUN, X1, 2, X2, X3, 3)` означает $\frac{\partial^6}{\partial^2 X_1 \partial X_2 \partial^3 X_3}$.

Неопределённый интеграл от алгебраического выражения можно найти с помощью оператора `INT`, имеющего два аргумента: подынтегральное выражение и переменную, по которой надо интегрировать.

Операторы дифференцирования и интегрирования правильно понимают и используют большинство алгебраических и тригонометрических функций, таких как `EXP`, `LOG`, `SIN`, `COS` и др.

10.8 Режимы работы

`REDUCE` имеет множество режимов работы, включаемых и выключаемых с помощью «флагов». Для включения используется оператор `ON`, за которым следует список флагов через запятую, для выключения – оператор `OFF` со списком флагов, например: `ON GCD, FORT; OFF NAT, FACTOR;`. В начале работы все они имеют некоторое состояние по умолчанию. Приведём некоторые из них, указывая в скобках в конце описания состояние по умолчанию.

- DIV** – при наличии рациональных дробей числитель делится почленно на знаменатель (OFF).
- EXP** – при вычислении выражений раскрываются скобки (ON).
- FACTOR** – производится попытка разложить выражение на множители из полиномов с целочисленными коэффициентами (OFF).
- FLOAT** – обыкновенные дроби заменяются представлением десятичными дробями, при этом точность определяется конкретной реализацией REDUCE. Можно повысить точность представления, включив флаг ON BIGFLOAT (по умолчанию при этом будет точность 10 десятичных цифр, но и это можно изменить, выдав команду PRECISION *число_знаков*). Первоначальное состояние флага FLOAT – OFF.
- FORT** – результаты вычисления выражений печатаются в виде, пригодном для вставки в фортрановскую программу (OFF).
- GCD** – делается попытка сократить общие множители в дробях (OFF).
- LCM** – при приведении дробей к общему знаменателю производится поиск наименьшего общего кратного знаменателей (OFF).
- MCD** – сумма дробей приводится к общему знаменателю (ON).
- NAT** – выдача результатов производится в «естественном» виде (показатели степеней печатаются на отдельных строках и т.п.), непригодном для последующего ввода в REDUCE (ON).
- NUMVAL** – приводит к численной оценке выражений, что касается, в основном, математических функций от численных аргументов (OFF).

10.9 Некоторые дополнительные команды

Окончить сеанс работы с REDUCE можно оператором QUIT. Если по ошибке использован оператор END, то система продолжит общение на языке LISP, на котором, собственно, и написана система REDUCE. Это может оказаться неприятным сюрпризом, так как большинству пользователей этот язык совершенно незнаком. Вернуться к работе на языке REDUCE можно командой

```
(BEGIN) RETURN
```

При работе с REDUCE можно ввести группу команд, записанных в заранее подготовленном файле, оператором IN:

```
IN "file_name";
```

Ввод из этого файла заканчивается, когда встретится комбинация «;END;».

Можно также направить весь последующий вывод результатов в файл на диске командой

```
OUT "file_name";
```

По окончании вывода можно закрыть файл командой

```
SHUT "file_name";
```

после чего дальнейший вывод снова будет производиться на экран.

Bibliography

- [1] *Катцан Г.* Язык Фортран 77 / Пер. с англ. М.: Мир, 1982.
- [2] *Соловьёв П.В.* Fortran для персонального компьютера. Справочное пособие. М.: Арист, 1991.
- [3] *Дайамендстоун Дж. и компания “И-Ар-Ай” Трейнинг.* Использование ОС VAX/VMS. М.: Мир, 1992.
- [4] *Kernigan B.W., Pike R.* The UNIX programming environment. Prentice-Hall, 1984.
Керниган Б.В., Пайк Р. UNIX – универсальная среда программирования / Пер. с англ. А.М.Березко, В.А.Иващенко, под ред. и с предисл. М.И.Белякова. М.: Финансы и статистика, 1992.
- [5] *Немет Э., Снайдер Г., Субасс С., Хейн Т.Р.* UNIX: руководство системного администратора / Пер. с англ. С.М.Тимачева. Под ред. М.В.Коломьщева. Киев: ВНУ, 1999.
- [6] *Bourne Ph.E.* UNIX for VMS Users. Digital Press, 1990.
- [7] *Столмен Р.* Руководство по GNU Emacs. АНО «ИЛК и РЛ», 1999.
- [8] *Соболь И.М.* Численные методы Монте-Карло. М.: Наука, 1973.
- [9] *Ермаков С.М., Михайлов Г.А.* Курс статистического моделирования. М.: Наука, 1976.
- [10] *Ермаков С.М.* Метод Монте-Карло и смежные вопросы. М.: Наука, 1971.

- [11] *Елепов Б.С., Кронберг А.А., Михайлов Г.А., Сабельфельд К.К.* Решение краевых задач методом Монте-Карло. Новосибирск: Наука, 1980.
- [12] *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. М.: Наука, 1973.
- [13] *Стечкин С.Б., Субботин Ю.Н.* Сплайны в вычислительной математике. М.: Наука, 1976.
- [14] *Буккин А.Д.* Интерполяция экспериментальных данных при отсутствии теоретической модели. ЖЭТФ 116, вып. 3(9), стр. 760–776, 1999.
- [15] *Anikeev V.B., Popov A.V., Zhigunov V.P.* An approach to a B-spline description of experimental distributions. Nuclear Instruments and Methods in Physics Research A 372 (1996) 482–488.
- [16] *De Boor C.* On calculating with B-splines. J. Approximation Theory 6, №1 (1972), 50–62.
- [17] *Лукач Е.* Характеристические функции / Пер. с английского В.М.Золотарёва. М.: Наука, 1979.
- [18] *Ландау Л.Д.* On the energy loss of fast particles by ionization. J. Phys. USSR, 8 (1944) 201–205.
О потерях энергии быстрыми частицами на ионизацию. В кн.: “Л.Д.Ландау. Собрание трудов”, М. Наука, 1969, стр. 482–490.
- [19] *Идьё В., Драйард Д., Джеймс Ф., Рус М., Садуле Б.* Статистические методы в экспериментальной физике / Пер. с англ. В.С.Курбатова. Под ред. А.А.Тяпкина. М.: Атомиздат, 1976.
- [20] *Jennewein T., Achleitner U., Weihs G., Weinfurter H. and Zeilinger A.* Fast and Compact Quantum Random Number Generator. *Subm. to: Rev. Sci. Instrum.*
e-Print archive: quant-ph/9912118, 1999, 29 дек. 23 с.
- [21] *Кнут Д.* Искусство программирования. Т. 2. Получисленные алгоритмы. М.: Мир, 1977.
- [22] *Буккин А.Д.* О мультипликативных генераторах псевдослучайных чисел. Препринт ИЯФ СО АН СССР 86–37. Новосибирск, 1986.

- [23] *Акопов Н.З., Саввиди Г.К., Тер-Арутюнян-Саввиди Н.Г.* Матричный генератор псевдослучайных чисел. Препринт ЕФИ–867(18)–86. Ереван, 1986.
- [24] *l'Escuyer P.* Efficient and Portable Combined Random Number Generators // Communications of ACM. Vol. 31, N^o 6 (1988, June). P. 742–749, 774.
- [25] *James F.* A review of pseudorandom number generators // Computer Physics Communications 60 (1990). P. 329–344.
- [26] *Marsaglia G.* Random Numbers Fall Mainly in the Planes. Proc. Nat. Acad. Sci. 61 (1968). P. 25–28.
- [27] *Букин А.Д.* Корреляции псевдослучайных чисел мультипликативной последовательности. Препринт ИЯФ 89–86, Новосибирск, 1989.
- [28] *Dyadkin I.G., Hamilton K.G.* A study of 64-bit multipliers for Lehmer pseudorandom number generators. Computer Physics Communications 103 (1997). P. 103–130.
- [29] *Salicio J.* Monte Carlo methods // 1984 CERN school of computing. Preprint CERN 85–09. 1985, 9 July.
- [30] *Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., Keller E.* Equation of state calculations for fast computing machines / J. Chem. Phys. 6 (1953). P. 1087.
- [31] *Гулд Х., Тобочник Я.* Компьютерное моделирование в физике. М.: Мир, 1990. Ч. II.
- [32] *Weinzierl S.* Introduction to Monte Carlo methods. Preprint NIKHEF-00-012, Amsterdam. 2000, 23 June. Pres. at: Research School on Subatomic Physics, Amsterdam, The Netherlands. 2000, June. e-Print archive: hep-ph/0006269.
- [33] *Brun R., Bruyant F., Maire M., McPherson A.C., Zanarini P.* GEANT3. CERN preprint DD/EE/84–1. Genève, 1987.
- [34] *Букин А.Д., Грозина Н.А., Дубровин М.С., Иванченко В.Н., Таярский В.А., Эйдельман С.И.* UNIMOD2 — универсальная программа моделирования экспериментов на встречных e^+e^- пучках. 5. Руководство пользователя версии 2.0. Препринт ИЯФ СО РАН 94–20. Новосибирск, 1994.

- [35] “Stopping Powers for Electrons and Positrons”, ICRU Report No. 37 (1984).
- [36] *Sternheimer R.M.* Phys. Rev. 88 (1952) 851.
- [37] *Sternheimer R.M., Peierls R.F.* “General expression for the density effect for the ionization loss of charged particles”. Phys. Rev. B3 (1971) 3681–3692.
- [38] *Вавилов П.В.* ЖЭТФ 32, №4 (1957) 920
- [39] *Буккин А.Д., Грозина Н.А.* Моделирование флуктуаций ионизационных потерь тяжёлых заряженных частиц. Препринт ИЯФ СО АН СССР 87–9, 1987.
Monte Carlo simulation of fluctuations of the ionization losses of heavy charged particles. Computer Physics Communications 78 (1994) 287–290.
- [40] *Аоки М.* Введение в методы оптимизации. М.: Наука, 1977.
- [41] *Гилл Ф., Мюррей У.* Численные методы условной оптимизации. М.: Мир, 1977.
- [42] *Gill Ph.E., Murray W.* Numerically stable methods for quadratic programming. Math. Prog. 14 (1978). P. 349–372.
- [43] *Chamberlain R.M., Powell M.D.J., Lemarechal C.* The watchdog technique for forcing convergence in algorithms for constrained optimization. Math. Prog. Study 16 (1982). P. 1–17.
- [44] *Goldfarb D., Idnani A.* A numerically stable dual method for solving strictly convex quadratic programs. Math. Prog. 27 (1983). P. 1–33.
- [45] *Burke J.V., Han S.P.* A robust sequential quadratic programming method. Math. Prog. 43 (1989). P. 277–303.
- [46] *Coleman Th.F. and Fenyés P.A.* Partitioned quasi-Newton methods for nonlinear equality constrained optimization. Math. Programming 53 (1992). P. 17–44.
- [47] *Heinz J., Spellucci P.* A successful implementation of the Pantoja-Mayne SQP method. Optimization Methods and Software 4 (1994). P. 1–28.

- [48] *James F.* Function minimization / Proceedings of the 1972 CERN computing and data processing school, Pertisau, Austria, 10–24 September, 1972. Preprint CERN 72–21, 1972, 1 Dec. P. 1–52.
- [49] *James F., Roos M.* MINUIT – a system for function minimization and analysis of the parameter errors and correlations. Computer Phys. Communications 10 (1975). P. 343–367.
Preprint CERN-DD-75-20, Geneva, 1975, 38 p.
CERN program library (D506).
- [50] *Rosenbrock H.H.* An automatic method for finding the greatest or least value of a function / Comput. J. **3** (1960). P. 175.
- [51] *Nelder J.A., Mead R.* A simplex method for function minimization / Comput. J. **7** (1965). P. 308.
- [52] *Manevich A.I., Boudinov E.* An efficient conjugate directions method without linear minimization / Nuclear Instruments and Methods in Physics Research A 455 (2000). P. 698–705.
- [53] *Fletcher R., Reeves C.M.* Function minimization by conjugate gradients / Comput. J. **7** (1964). P. 149.
- [54] *Fletcher R., Powell M.J.D.* A rapidly converging descent method for minimization / Comput.J. **6** (1963). P. 163.
- [55] *Fletcher R.* A new approach to variable metric algorithms / Comput.J. **13** (1970). P. 317.
- [56] *Gelfand L.M., Tsetlin M.L.* The principle of non-local search in automatic optimization systems // Soviet Phys. Dokl. **6** (1961). P. 192.
- [57] *Goldstein A.A., Price J.F.* On descent from local minima // Math. Comput. **25** (1971). P. 569.
- [58] *Худсон Д.* Статистика для физиков. М.: Мир, 1970.
- [59] *Яноши Л.* Теория и практика обработки результатов измерений. М.: Мир, 1965.
- [60] *Neyman J., Pearson E.S.* On the problem of the most efficient tests of statistical hypotheses // Phil. Trans. Roy. Soc., Ser. A, **231**(1933). P. 289–337.

- [61] *Neyman J.* Trans. Royal Soc. London, Ser. A, **236** (1937). P. 333–380. Reprinted in: A Selection of Early Statistical Papers on J.Neyman. Berkeley: University of California Press, 1967. P. 250–289.
- [62] *Berge J.P., Solmitz F.T., Taft H.D.* Kinematical analysis of interaction vertices from bubble chamber data. Rev. Sci. Instr. **32** (1961). P. 538–548.
- [63] *Bukin A.D.* On the kinematic reconstruction of multiparticle events. Talk given at Computing in High-energy Physics (CHEP 97), Berlin, Germany, 7–11 Apr 1997.
- [64] *Bukin A.D.* Optimal rotation procedure. Preprint Budker INP 97–50, Novosibirsk, 1997.
- [65] *Курдадзе Л.М., Панин В.С., Эйдельман С.И.* Кинематическая реконструкция трёх- и четырёх-частичных событий в e^+e^- аннигиляции. Препринт ИЯФ СО РАН, Новосибирск, 1979.
- [66] *Feldman G.J., Cousins R.D.* A Unified Approach to the Classical Analysis of Small Signals. Phys. Rev. D**57** (1998). P. 3873–3889. e-Print archive: physics/9711021.
- [67] *Букин А.Д.* Сравнение метода построения доверительного интервала по функции правдоподобия с классическим методом Неймана. Препринт ИЯФ 2001-12. Новосибирск, 2001.
- [68] *Particle Data Group.* Review of particle physics. Eur. Phys. J. C **15** (2000), P. 1.
- [69] *Букин А.Д., Курдадзе Л.М., Пахтусова Е.В. и др.* Метод разделения частиц с использованием функций правдоподобия и корреляционных матриц. Препринт ИЯФ 77–92, Новосибирск, 1977.
- [70] *Auslender V.L., Budker G.I., Pestov Yu.N. et al.* Investigation of the ρ -meson resonance with electron-positron colliding beams. Phys. Lett. **25B** (1967) 433.
- [71] *Ауслендер В.Л., Будкер Г.И., Пахтусова Е.В. и др.* Исследование ρ -мезонного резонанса на встречных электрон-позитронных пучках. Ядерная физика **9** (1969) 114.
- [72] *Букин А.Д.* Критерий согласия Колмогорова. Препринт ИЯФ 98–81. Новосибирск, 1998.

- [73] *Большев Л.Н., Смирнов Н.В.* Таблицы математической статистики. М.: Наука, 1965.
- [74] *Смирнов Н.В.* О распределении ω^2 -критерия Мизеса // *Смирнов Н.В.* Теория вероятностей и математическая статистика. Избранные труды. М.: Наука, 1970.
- [75] *Кендалл М., Стьюарт А.* Статистические выводы и связи / Пер. с англ. Л.И.Гальчука, А.Т.Терехина. Под ред. А.Н.Колмогорова. М.: Наука, 1973.
- [76] *Hearn A.C.* REDUCE User's Manual / RAND Pub. CP78 (rev. 7/87).
- [77] *Еднерал В.Ф. и др.* Язык аналитических вычислений REDUCE. М.: Изд-во МГУ, 1983.
- [78] *Грозин А.Г.* Решение физических задач на языке REDUCE.
Ч. 1. Язык REDUCE. Ч. 2. Классический нелинейный осциллятор. Препринт ИЯФ СО АН СССР 88–115. Новосибирск, 1988.
Ч. 3. Нелинейные волны на воде. Ч. 4. Вычисление тензора кривизны. Ч. 5. Сложение моментов в квантовой механике. Препринт ИЯФ СО АН СССР 88–136. Новосибирск, 1988.
Ч. 6. Квантовый нелинейный осциллятор. Ч. 7. Ротатор в слабом поле. Ч. 8. Ротационные переходы в чармонии. Препринт ИЯФ СО АН СССР 88–140. Новосибирск, 1988.
- [79] *Грозин А.Г.* Система REDUCE в физике элементарных частиц. Квантовая хромодинамика. Препринт ИЯФ СО АН СССР 90–62. Новосибирск, 1990.

Букин Александр Дмитриевич, д-р физ.-мат. наук
Эйдельман Семен Исаакович, д-р физ.-мат. наук

ЭВМ в физическом эксперименте

Подписано в печать

Формат 60 × 84/16
