8

ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ
им. Г.И. Будкера СО РАН

А.Д. Букин, Н.А. Грозина, М.С. Дубровин,
В.Н. Иванченко, В.А. Таюрский, С.И. Эйдельман

UNIMOD 2—УНИВЕРСАЛЬНАЯ ПРОГРАММА МОДЕЛИРОВАНИЯ ЭКСПЕРИМЕНТОВ НА ВСТРЕЧНЫХ e^+e^- ПУЧКАХ

5. Руководство пользователя версии 2.0

ИЯФ 94-20

НОВОСИБИРСК

UNIMOD2 — универсальная программа моделирования экспериментов на встречных e^+e^- пучках. [1]

5. Руководство пользователя версии 2.0

А.Д. Букин, Н.А. Грозина, М.С. Дубровин, В.Н. Иванченко, В.А. Таюрский, С.И. Эйдельман

Институт ядерной физики
Новосибирск, 630090

Аннотация

Описана программа UNIMOD2 моделирования прохождения излучения через вещество. Точность моделирования основных процессов взаимодействия элементарных частиц с веществом около 1% при энергии до 10 ГэВ. Область применения программы — моделирование экспериментов по физике высоких энергий. При переносе предыдущей версии UNIMOD2 на VAX-3600 и SPARC station существенно модифицированы правила оформления задания.

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

[1]Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 93-02-3295).

1 Введение

В планировании и обработке экспериментов по физике высоких энергий широко используются программы моделирования прохождения элементарных частиц через вещество. В ведущих физических центрах популярны программы EGS, GEANT, FLUKA, разработанные в середине 70-х — 80-х годах, которые применяются для расчётов электромагнитных и ядерных ливней, моделирования экспериментов.

В Институте ядерной физики СО РАН в 1978–1982 была разработана программа UNIMOD1 [1]. Эта программа в течение длительного времени использовалась для моделирования экспериментов на детекторах НД (Нейтральный Детектор) на установке ВЭПП-2М [2] и МД-1 (Магнитный Детектор) на установке ВЭПП-4 [3], а также для методических расчетов.

В 1985 году была начата разработка новой версии программы моделирования UNIMOD2, которая включила в себя целый ряд назревших усовершенствований. Для этой версии был заново разработан ввод данных, который позволяет оперативно переходить от сравнительно простых методических расчётов к моделированию полномасштабных детекторов. Кроме чисто структурных улучшений была проведена большая работа по повышению точности моделирования взаимодействия частиц с веществом. В основном, это касается ядерных взаимодействий адронов. В 1990 году программа была реализована на машинах ЕС-1061 в рамках операционной системы СВМ (аналог VM/CMS) [5]. Тогда же началась разработка на основе UNIMOD2 программы моделирования детектора СНД (Сферический Нейтральный Детектор) [6], который будет работать на ВЭПП-2М.

Следует отметить, что программы UNIMOD1 и UNIMOD2 разрабатывались параллельно с черновской программой GEANT [7], которая, безусловно, является наиболее широко используемой в мире для моделирования экспериментов. Проведённое нами сравнение моделирования по UNIMOD2 и GEANT [8] показало совпадение с высокой точностью результатов расчётов. Это указывает на то, что точность расчётов определяется формулами для вычисления сечений и потерь энергии основными частицами каскада, которые практически идентичны в обеих программах, а отличие в некоторых алгоритах моделирования дает пренебрежимый вклад. Существенное различие между программами состоит в организации ввода-вывода, кроме того они содержат несовпадающие

наборы подпрограмм моделирования геометрических блоков. Преимуществом UNIMOD2 является более удобный, в особенности для моделирования больших детекторов, ввод данных для моделирования, более аккуратное моделирование ядерных взаимодействий при низких энергиях (P < 1 ГэВ/с), а также доступность разработчиков программы для пользователей Института. Но даже пользователям GEANT, на наш взгляд, является полезным использование UNIMOD2 в методических расчётах — это позволит лучше понять возможные систематические ошибки.

В 1991 – 1991 годах в нашем Институте был образован кластер из трёх ЭВМ VAX-3600 с операционной системой VAX/VMS. Было принято решение сделать версию программы UNIMOD2 также и для VAX/VMS. Несмотря на большое сходство диалектов Фортрана-77 на VAX и EC-1061, работа по переносу программ на VAX заняла несколько месяцев и была закончена в конце 1991 года.

К началу 1993 года версия программы UNIMOD2 на VAX была вполне отлажена, а программа моделирования детектора СНД на её основе начала активно использоваться для разработки и отладки программ реконструкции событий. Кроме того, с использованием UNIMOD2 было разработано моделирование экспериментов на установке ДЕЙТРОН на накопителе ВЭПП-3 [9].

В 1993 году в ИЯФ СО РАН появились несколько станций SPARC/IPC под операционной системой UNIX, на которых, в частности, планировалось вести моделирование. Поэтому было затрачено необходимое количество усилий для запуска программы UNIMOD2 на SPARC/IPC, чтобы иметь возможность проводить и на этих машинах моделирование СНД и ДЕЙТРОНа, а также другие расчёты.

Все изменения, внесённые в UNIMOD2, настолько сильно отразились на правилах оформления задания и процедуре запуска на счёт, что появилась настоятельная необходимость в новом издании руководства пользователя.

2 Основные принципы UNIMOD2

Основные идеи, заложенные в структуру UNIMOD2, можно сформулировать следующим образом.

а) Работа программы разбита на два этапа: сначала интерпретатор ввода анализирует входной поток данных и в соответствии с ним генерирует головную программу на языке Фортран-77 в отдельный

файл на диске, затем программа транслируется стандартным компилятором и запускается на счёт, причём данные для счёта в виде содержимого общих областей передаются из шага обработки входного потока на шаг счёта через файлы в рабочей директории.

б) Структура головной программы определяется интерпретатором ввода на основе входных данных. Если требуется добавить какие-то программы пользователя, то они, в основном, должны осуществлять анализ и запись результатов и не влиять на развитие каскада частиц.

в) Различия в структуре задания на счёт при моделировании сложного детектора для эксперимента по физике высоких энергий и простых методических расчётов должны быть скорее количественные, чем качественные. В то же время простые расчёты должны достаточно просто оформляться.

г) Не исключая возможности написания своих программ для пользователя, ставилась задача минимизировать объём обязательного программирования, предоставив возможность использовать UNIMOD2 и тем пользователям, которые имеют небольшой опыт программирования или вообще его не имеют.

д) Структура программы должна максимально способствовать объединению усилий группы, ведущей эксперимент на сложном детекторе. Это значит, что программы, разработанные каким-то членом группы, должны легко включаться в работу других сотрудников.

Как и любая программа моделирования экспериментов по физике элементарных частиц, UNIMOD2 состоит из блока ввода и обработки входного потока, цикла набора статистики и вывода результатов.

Исходные данные для ввода требуется приготовить текстовыми редакторами в файлах с расширением **".DAT"** по специальным правилам. Эти правила введены для удобства ввода, а также для более жёсткого контроля за ошибками. Интерпретатор ввода, разработанный для этой задачи, проверяет входные данные на синтаксические ошибки, а также на правильность ссылок на элементы детектора.

Программы ввода и обработки заказа зафиксированы и не могут быть оперативно изменены. Программа для цикла набора статистики и вывода результатов формируется на основании заказа на счёт. Это достигается включением в заказ на счёт списков программ, которые

используются для решения формализованных задач, например, поиск пересечения траектории с поверхностью некоторой геометрической фигуры, или вычисление сечения какого-либо процесса взаимодействия частиц с веществом, или генерация параметров частиц — продуктов столкновения электрона и позитрона и т.д. По указанию пользователя могут использоваться как готовые программы из библиотеки UNIMOD2, так и оригинальные программы пользователя, подготовленные в соответствии с описанием.

Для повышения удобства при работе с объёмными описаниями сложных детекторов введена возможность составлять входной поток данных из отдельных файлов, которым можно придать функциональное назначение. При запуске на счёт процедуре UNIMOD2 указывается имя входного файла данных, в тексте которого, при необходимости, используются команды включения других файлов данных.

Входной поток может компоноваться из отдельных своих числовых массивов. Кроме того, в комплекте UNIMOD2 имеются также стандартные массивы описания свойств веществ, элементарных частиц, а также список программ моделирования взаимодействий.

Названия программ, входящих в стандартный комплект UNIMOD2, имеют однотипный вид: префикс UM, номер версии (одна цифра) программы UNIMOD2 (сейчас это всегда 1) и трёхзначный номер. По тому, в какой из нижеприведённых интервалов попадает этот номер, можно определить, к кому обращаться с проблемами, возникшими в соответствующей программе:

000 – 099 Букин А.Д.
100 – 199 Эйдельман С.И.
200 – 299 Таюрский В.А.
300 – 399 Иванченко В.Н.
400 – 499 Букин А.Д.
600 – 699 Грозина Н.А.
700 – 799 Дубровин М.С.
800 – 899 Букин А.Д.

Любая диагностика должна начинаться с имени подпрограммы, которая выдаёт сообщение, для ускорения поиска автора сообщения в непонятных случаях. Это не относится к печати, которая появляется при нормальной работе программы, хотя, как правило, нормальная печать тоже сопровождается идентификацией.

3 Процедура запуска программы

В описании работы программы неизбежно приходится использовать некоторые термины, связанные с организацией работы в конкретной операционной системе: VAX/VMS или UNIX. Описание этих терминов можно найти в соответствующих книгах, например, [10, 11].

Запуск программы UNIMOD2 на счёт осуществляется процедурой: в операционной системе VAX/VMS файл UNIMOD2.COM, в системе UNIX — файл UNIMOD2. Несмотря на значительное перекрытие набора параметров и системных переменных, мы решили, для удобства пользователя, описать правила запуска программы раздельно для VAX/VMS и UNIX. В приводимых образцах команд **жирным шрифтом** будут выделены те части, которые надо воспроизводить без изменения, а *курсивом* будут набраны слова, которые требуют замены конкретными значениями.

3.1 Система рабочих файлов UNIMOD2

Файлы с данными для входного потока могут находиться в нескольких разных директориях. Для указания программе ввода, какие директории и в каком порядке просматривать при поиске файлов данных, необходимо создать специальный файл, в котором каждая строка содержит в первой позиции какую-либо букву латинского алфавита, по которой на данную директорию будет производиться ссылка при вводе данных, затем двоеточие и следом полное имя директории. Полное имя файла со списком директорий должно быть сообщено программе UNIMOD2 через логическое имя DIRLIST на VAX/VMS и через *environment variable* DIRLIST в системе UNIX (детали в разделах 3.2,3.3). В этот файл обязательно следует включить основную директорию UNIMOD, где имеются файлы с описанием свойств частиц, программ моделирования процессов, параметров веществ. Список директорий, заданных таким образом, позволяет заметно упростить входной поток.

В любом месте файла данных можно дать указание приостановить ввод из данного файла и начать ввод из другого командой, размещённой на отдельной строке:
=(*S:FNAME*)

где *S* — латинская буква, приписанная директории в файле DIRLIST, *FNAME* — имя включаемого файла не более 8 символов (расширение ".DAT" указывать не надо, оно подразумевается). Символ директории и двоеточие можно опустить, в этом случае поиск файла с указанным названием будет проводиться в следующем порядке:

- рабочая директория,
- директории по порядку из файла DIRLIST,
- директория, содержащая программу UNIMOD2.

Глубина вложенности при вызове новых файлов не должна превышать 2. Нет никаких других ограничений, кроме глубины вложенности, на разбиение всего входного потока на отдельные файлы, поэтому разбиение можно осуществлять только из соображений удобства работы. Интерпретатор ввода обрабатывает весь входной поток, как один большой файл, где текст каждого вложенного файла вставлен на место команды вставки.

При просмотре входного потока производится поиск синтаксических ошибок, перенос данных в соответствующие общие области программы, генерация файла с головной программой UM1000.FOR, в которой учтены реальные размеры общих блоков (что предотвращает выход за пределы общих областей) и обеспечивается обращение в соответствующих местах головной программы к программам, указанным в заказе. Сгенерированная программа транслируется и запускается на счёт.

Взаимные ссылки разных элементов входного потока производятся по именам, назначенным этим элементам в заказе на счёт. Это облегчает контроль за правильностью ссылок и модификацию описания детектора при добавлении или вычеркивании элементов описания.

По ходу работы программа создаёт ряд рабочих файлов, файл результата (протокол работы, сообщения, статистика счёта, гистограммы) и, возможно, какие-то файлы пользователя, записанные из подпрограмм пользователя. Все файлы записываются в рабочую директорию, переход в которую осуществляется в самом начале работы процедуры. По окончании счёта текущей директорией (*default directory*) остаётся рабочая директория. Способы определения рабочей директории см. в разделах 3.2,3.3.

Запуск разных вариантов счёта одновременно в разных рабочих директориях рекомендуется осуществлять для избежания коллизий при записи-чтении стандартных промежуточных файлов с одинаковыми именами. Особенно важно это делать при пакетной обработке заданий, когда управлять временем прохождения заданий нет возможности.

3.2 Работа на VAX/VMS

Перед началом работы программы должны быть определены логические имена UNIMOD и DIRLIST (см. раздел 3.1), а также символы LIB$UM — список библиотек для сборки программы, и UNI_DEF_DIR — рабочая директория по умолчанию. Приведём возможный набор команд:

$ assign disk$d1:[unimod] unimod
$ assign *full_name_of_dir_list* **dirlist**
$ LIB$UM == "unimod:unimod2/lib"
$ UNI_DEF_DIR == "*working_directory***"**

Эти команды могут быть включены в командный файл LOGIN.COM или специальный командный файл, который надо запускать перед работой программы.

Переменная LIB$UM используется для передачи списка библиотек редактору связей LINK. В указанном примере список состоит только из одной библиотеки с программами UNIMOD2 (эту библиотеку обязательно надо указывать). При необходимости увеличить этот список надо добавлять через запятую полные имена других библиотек с ключом "/lib" и объектных файлов подпрограмм.

Отметим, что программы пользователя, введённые во входном потоке (см. 4.20), включаются в сборку автоматически и не требуют отдельного описания.

Если переменная UNI_DEF_DIR не определена, то в качестве рабочей директории используется директория с логическим именем DISK$DAY:[*user_name*], которым для VAX-кластера ИЯФ СО РАН обозначена личная директория каждого пользователя на рабочем диске с возможностью хранения файлов не более суток.

Вызов процедуры в интерактивном режиме осуществляется командой:

$ unimod:unimod2 ***s:file1 file2 options***

где

file1 — имя начального файла данных (расширение имени не указывается, должно быть .DAT). Перед именем может быть указан символ директории *s* из списка DIRLIST с двоеточием, в которой находится данный файл (в противном случае действует порядок поиска файлов по умолчанию — см. раздел 3.1).

file2 — имя выходного файла (без расширения, подразумевается расширение ".LIS"). В этот файл выводятся распечатки исходных данных, идентификация программ моделирования процессов, диагностика сбоев, гистограммы. Файл записывается в текущую директо-

рию, точнее, в директорию, которая будет текущей на шаге счёта. Приписывание буквенных меток с двоеточием к выходному файлу не допускается, также как указание полного имени файла (с директорией и именем диска).

options — несколько (или ни одного) дополнительных параметров процедуры в произвольном порядке, разделённых пробелами.

Список дополнительных параметров:

SOURCE — вывести протокол трансляции головной программы вместе с исходным текстом в файл UM1000.LIS.

MAP — вывод протокола сборки головной программы в файл UM1000.MAP.

SYNTAX — счёт не запускается, делается только раскодировка заказа, генерация головной программы и её трансляция.

DEBUG — для файла UM1000.FOR выполняется отладочная трансляция (опция компилятора DEBUG). Сборка тоже выполняется с ключом DEBUG. Для счёта в этом режиме надо пользоваться инструкцией по работе с интерактивным отладчиком VAX/VMS.

CONT — в конце счёта не удаляются объектный модуль головной программы UM1000 и файл с содержимым общих блоков COMMONS.WORK, передающиеся из шага обработки заказа на шаг счёта. Эти массивы нужны для запуска счёта с тем же самым заказом (см. описание параметра FAST).

FAST — параметр быстрого запуска счёта с использованием ранее полученной головной программы и содержимого общих блоков с помощью параметра CONT в предыдущем счёте. Такой счёт имеет смысл, если в целях отладки меняется текст какой-либо подпрограммы, использующейся на шаге счёта, или в том случае, когда в какой-либо подпрограмме пользователя меняется содержимое общих блоков (например, исходное случайное число).

PILE — указание делать сборку программы обработки заказа. Если этот параметр не указан, то берётся стандартная сборка, что подходит для большинства расчётов.

MODE=*s* — указание в начале счёта сменить текущую директорию командой SET DEFAULT на директорию с меткой "*s*" из списка DIRLIST (см. раздел 3.1). Если этот параметр не указан, то производится переключение на директорию, определённую в системной переменной UNI_DEF_DIR, если она не определена, то на директорию DISK$DAY:[*username*] (временный диск на VAX-кластере в ИЯФ СО РАН).

PUT=*filename* — после ввода заказа на счёт и проверки на синтаксис содержимое общих блоков будет переписано в файл с именем *filename*.COMMONS в текущую директорию (имя файла *filename* не более 8 символов).

GET=*s:filename* — перед вводом нового заказа на счёт сначала будет восстановлено содержимое общих блоков от предыдущего счёта из файла *filename*.COMMONS. Таким образом можно экономить на вводе, раскодировке и проверке на правильность ссылок в очень больших описаниях. Метку директории *s* вместе с двоеточием можно не указывать.

NOOPT — нет оптимизации при трансляции головной программы. При отсутствии этого параметра и параметра DEBUG производится оптимизация.

Примеры обращения к процедуре UNIMOD2:

```
@unimod:unimod2 tsuni1 res1
@unimod:unimod2 tsuni2 res2 mode=a
```

Команды постановки этих же заданий в пакетную очередь:

```
submit unimod:unimod2/par=(tsuni1,res1)
submit unimod:unimod2/par=(tsuni2,res2,mode=a)
```

3.3 Работа в системе UNIX

Основным отличием системы UNIX от VAX/VMS, которое проявляется при попытке написать процедуру, является отсутствие в UNIX (C-shell) тех богатых возможностей, какие предоставляет язык командных процедур DCL в VAX/VMS. Как следствие, набор параметров процедуры не удалось сделать тождественным на SPARC-station и на VAX.

Другим отличием UNIX от VAX/VMS является то, что в UNIX большие и маленькие буквы считаются разными символами. В компиляторе Фортрана "SPARCompiler FORTRAN from SunPro (release 2.0.1)" эта

особенность устранена и его синтаксис полностью совпадает с компилятором VAX/VMS "VAX FORTRAN v5.1-10". В интерпретаторе входных данных программы UNIMOD2 тоже сделана попытка не различать большие и маленькие буквы в системе UNIX, однако, не везде это удалось сделать достаточно последовательно. Поэтому настоятельно рекомендуется не использовать в названии одного файла смесь маленьких и больших букв. Кроме того, в списке DIRLIST (см. раздел 4) названия директорий следует записывать в точности так, как они высвечиваются по команде *pwd*.

В файловой системе UNIX нет понятия *версии файла*, что с разных точек зрения может представляться достоинством или недостатком файловой системы. С одной стороны, это снимает проблему накопления разных версий рабочих файлов с одним названием, с другой стороны, конфликты одновременно выполняющихся заданий становятся более острыми.

Так как при работе программы UNIMOD2 запускается компилятор Фортрана для трансляции головной программы, то должны быть определены соответствующие системные переменные, позволяющие транслировать фортрановские программы командой f77.

Команды настройки могут быть примерно такими:

```
set path ($path /disk2/g/lang /disk2/g/lang/SC2.0.1)
setenv MANPATH /disk2/g/man:disk2/g/lang/man
```

Приведённые команды соответствуют файловой структуре SPARCstation с названием узла *vepp3sun* в ИЯФ СО РАН. Эти и другие необходимые команды могут быть добавлены в командные файлы, которые автоматически запускаются при входе в машину: .login или .cshrc.

Перед началом работы программы должны быть определены *environment variables* UNIMOD и DIRLIST (см. раздел 3.1), а также LIB_UM — список библиотек для сборки программы, и UNI_DEF_DIR — рабочая директория (в начале работы процедуры производится переход командой cd в эту директорию, а если переменная не определена, то работа продолжается в текущей директории).

Приведём возможный набор команд, в котором производятся соответствующие присвоения:

setenv UNIMOD /disk2/g/unimod
setenv DIRLIST *full_dirlist_filename*
setenv UNI_DEF_DIR *working_directory_name*

Директории, в которых при сборке производится поиск библиотек, перечисляются через ":" в переменной LD_LIBRARY_PATH, например:
setenv LD_LIBRARY_PATH *directory1:directory2:* ...

а список переменных частей имён библиотек приводится в переменной LIB_UM, например:
setenv LIB_UM "*nam1 nam2*"

Выражение "переменных частей имён" означает, что приведённый список библиотек соответствует именам файлов libnam1.a и libnam2.a.

Наконец, для удобства можно ввести альтернативное имя для процедуры unimod2, например:

```
alias uni2 '"$UNIMOD"unimod2'
```

Запуск программы в интерактивном режиме после этого может осуществляться командой uni2 и списком параметров в командной строке, перечисляемых через пробел:
uni2 *s:file1 file2 options*

Первые два параметра процедуры — обязательные.

file1 — первым параметром является имя начального файла данных (расширение имени не указывается, должно быть ".dat"). Перед именем может быть указана метка директории *s* из списка DIRLIST с двоеточием, в которой находится данный файл.

file2 — имя выходного файла (без расширения, подразумевается расширение .lis). В этот файл выводятся распечатки исходных данных, идентификация программ моделирования процессов, диагностика сбоев, гистограммы. Файл записывается в текущую директорию. Приписывание буквенных меток с двоеточием к выходному файлу не допускается.

source — не удаляется с диска в конце работы файл UM1000.f с текстом головной программы.

syntax — счёт не запускается, делается только раскодировка заказа, генерация головной программы и её трансляция.

cont — в конце счёта не удаляются объектный модуль головной программы UM1000 и файл с содержимым общих блоков COMMONS.WORK, передающиеся из шага обработки заказа на шаг счёта. Эти массивы нужны для запуска счёта с тем же самым заказом (см. описание параметра fast).

fast — параметр быстрого запуска счёта с использованием ранее полученной головной программы и содержимого общих блоков с помощью параметра cont в предыдущем счёте. Такой счёт имеет

смысл, если в целях отладки меняется текст какой-либо подпрограммы, использующейся на шаге счёта, или в том случае, когда в какой-либо подпрограмме пользователя меняется содержимое общих блоков (например, исходное случайное число).

pile — указание делать сборку программы обработки заказа. Если этот параметр не указан, то берётся стандартная сборка, что подходит для большинства расчётов.

put *filename* — после ввода заказа на счёт и проверки на синтаксис содержимое общих блоков будет переписано в файл с именем *filename*.COMMONS (имя файла *filename* не более 8 символов).

get *s:filename* — перед вводом нового заказа на счёт сначала будет восстановлено содержимое общих блоков от предыдущего счёта из файла *filename*.COMMONS. Можно указывать метку директории *s*. Таким образом можно экономить на вводе, раскодировке и проверке на правильность ссылок в очень больших описаниях.

opt *n* — уровень оптимизации при трансляции головной программы задаётся равным *n*. *n* должно быть числом в интервале от 1 до 4. При отсутствии этого параметра задаётся уровень оптимизации 2.

В отличие от версии UNIMOD2 на VAX/VMS, в системе UNIX нет возможности заказать переключение текущей директории (отсутствует параметр mode=s). Если не определена переменная UNI_DEF_DIR, то директория, из которой запускается задание (как в интерактивном, так и в пакетном режиме), остаётся текущей директорией на всё время исполнения задания.

Примеры команд для запуска UNIMOD2:

```
uni2 tsuni1 res1
uni2 tsuni2 res2 opt 4
```

Те же задания, что указаны выше, можно запустить в фоновом режиме:

```
uni2 tsuni1 res1 &
uni2 tsuni2 res2 opt 4 &
```

При этом терминал свободен для других работ, которые не мешают выполнению данных заданий (не удаляют и не модифицируют содержимое файлов, необходимых для работы UNIMOD2). Однако, прекращать сессию командой logout нельзя — дочерние процессы прервутся аварийно. Кроме того, некоторые сообщения выводятся на терминал и могут нервировать пользователя.

Более удобный способ запустить независимое задание — это пакетный режим:

```
batch
uni2 tsuni1 res1
d-ctrl
```

Последняя строка — это нажатие клавиши d в управляющем режиме (при нажатой клавише ctrl).

После этого терминал освобождается полностью (можно даже прерывать сессию). По окончании задания по "почте" будет переслан протокол работы задания.

4 Входной поток данных UNIMOD2

Как уже сказано, заказ на счёт может компоноваться из отдельных массивов данных. Однако, интерпретатор ввода расматривает все входные данные, как единый входной поток.

Весь входной поток данных разбивается на разделы, в каждом из которых свои правила описания соответствующих данных. Признаком начала раздела является отдельная строка, в первых двух позициях которой записаны два символа "**", а затем сразу кодовое слово — название раздела. Порядок следования разделов не фиксирован, описания одного и того же раздела могут встречаться несколько раз, при этом новая информация добавляется к ранее введённой. Начало и конец файла описания, вызываемого внутри другого файла, не обязательно должны совпадать с границей раздела. В кодовых словах названия раздела существенны только первые шесть символов, остальные безразличны.

Имеются некоторые общие правила для всех разделов.

Необходимым условием при оформлении задания является предварительное описание всех элементов системы, на которые имеются ссылки в описании очередного элемента (иными словами, все имена, на которые есть ссылки при описании очередного элемента, должны быть описаны ранее).

При вводе обрабатываются только первые 72 позиции строки, оставшиеся позиции строки игнорируются.

Ввод и обработка заказа ведётся до обнаружения первой синтаксической ошибки, затем выдаётся диагностика и счёт аварийно прекращается. При этом на печать выдаётся только одна строка заказа, во время обработки которой обнаружена ошибка (хотя иногда ошибка может быть в предыдущих строках), и диагностическое сообщение.

Список возможных кодовых слов:

MATERIals	список веществ с их характеристиками
REFEREnce	локальные системы координат
PARTICles	описание свойств частиц
CUTS	управляющие числа для развития каскада
ROUTINes	список подпрограм, включаемых в различные точки UNIMOD
GATES	интервалы времени в *нс* от начала события
RESPONces	"амплитуды" со счётчиков
BLOCKS	геометрическое описание блоков вещества
CHAMBErs	описание камер
HITS	"амплитуды" с камер
DISCRIminators	описание типов дискриминаторов
LOGICModules	описание типов логических блоков электроники
TRIGGEr	описание схемы запуска детектора
CONSTRaints	условия на гистограммные параметры события
FORMULas	логические формулы для отбора событий в гистограммы
HISTOGrams	описание гистограмм
MFIELD	описание способов вычисления магнитных полей
MVOLUMes	описание отсеков магнитного поля
REGIME	набор общих управляющих чисел
PRESS	отбрасывание некоторых ранее введённых блоков вещества по именам блоков
PRIVATe	тексты подпрограмм, которые присоединяются к файлу UM1000.FOR
CLEAN	удаление "лишних" программ вычисления гистограммных параметров.

Символ & (амперсанд), встретившийся в начале строки или в любом месте строки после запятой, превращает оставшуюся часть строки в комментарий.

Пробелы игнорируются везде, кроме текстовых последовательностей, заключённых в апострофы.

Большинство элементов детектора снабжается именами. Имена не могут начинаться с цифры или служебных символов, не могут также содержать скобки, запятые, двоеточия, &. Первоначально вводимые имена заканчиваются символом ":" (двоеточие), затем следует обычно информация, относящаяся к этому имени. Ссылки на элементы детектора и заказа производятся, как правило, по ранее введённым именам. Ссылка на неопределённое к этому моменту имя приведёт к аварийному

завершению.

Различие между большими и малыми буквами игнорируется.

Запись числовой информации бесформатная, разделителем является запятая или конец строки, а также служебные символы.

При записи дробных чисел можно использовать десятичную точку и признак порядка E. Порядок задаётся целым числом со знаком (плюс можно не писать). Абсолютная величина порядка не должна превышать 20. Примеры:

```
-5, 3.3, 1.5e-3, 1E10, 2e+10
```

> Стандартные единицы измерения энергии — МэВ, расстояния — см, времени — наносекунды. Отклонения от стандарта оговариваются в описании сразу же.

Как правило, элементы описания можно произвольно переносить на следующие строки. Формальные ограничения на перенос можно сформулировать так: элементы (числа, имена) переносятся только целиком и служебные символы (двоеточие, скобки) должны быть на одной строке с предыдущим элементом.

Может случиться так, что при вводе нового элемента заказа обнаружилось совпадение имён с ранее введённым однотипным элементом. Стандартной реакцией программы на это является обязательная печать сообщения вида:

XXXXXX: TTTTTT, line NNNNNN (MMMMMMMM), DDDDDDDD - duplicate name
где

XXXXXX название программы, из которой выдано сообщение,

TTTTTT тип вводимой информации,

NNNNNN номер строки в MMMMMMMM, в которой замечено повторное имя,

MMMMMMMM имя массива, в котором попалась такая строка,

DDDDDDDD имя, которое совпало.

После этого возможны два вида реакции интерпретатора на такое совпадение. В первом варианте прежние характеристики элемента заменяются на новые (в описании соответствующего раздела при этом есть примечание — возможна замена по имени), во втором варианте происходит аварийное завершение.

Для некоторых типов элементов заказа предоставляется возможность удалить введённый ранее элемент, указав после его имени и двоеточия ключевое слово DELETE, например:
UM1620: DELETE

В этом случае в описании раздела в данном руководстве есть примечание “возможно удаление по имени”.

Далее следует описание ввода в некоторой стандартной последовательности, которой, вообще говоря, можно не следовать при написании заказа. В заголовке каждого подраздела записывается кодовое слово типа информации, как оно было бы написано в заказе на счёт.

Напомним ещё раз, что произвол в порядке следования разделов ограничен условием: если в описании какого-либо элемента есть ссылка на другой элемент по имени, то это имя должно быть описано ранее.

Обратим внимание, что примеры в описании интерпретатора ввода иллюстрируют синтаксические правила и не могут заменить описания конкретных подпрограмм. Правила описания стандартных подпрограмм приводятся в разделе 9.

Тем пользователям, которым надо моделировать какую-либо простую систему и изучать данное подробное описание ввода нерационально, можно начать с раздела 12 “Простые примеры заданий”, уточняя затем отдельные места по конкретным разделам.

4.1 **MATERIALS — описание веществ

Описание простого вещества начинается с имени (до 8 символов), затем записывается двоеточие и через запятую перечисляются атомный номер, атомный вес, плотность в $г/см^3$, средний ионизационный потенциал (кэВ).

Для сложного вещества, то есть вещества, состоящего из различных атомов, после имени вводятся последовательно описания всех компонентов. При этом для каждого компонента указывается его парциальная плотность в данном веществе. Розыгрыш пробега до взаимодействия в сложном веществе в основном варианте производится путем раздельного вычисления сечения взаимодействия на атомах каждого компонента. Программа может использовать для вычисления сечений усреднённые характеристики (усреднение она сделает сама), если записать после характеристик последнего компонента ключевое слово AVRG. Генерация продуктов взаимодействия для сложного вещества всегда выполняется на атомах одного из компонентов в соответствии с вероятностями взаимодействия.

Пример:

```
Fe:   26, 55.85, 7.87, 0.3
NaI:  11, 22.99, 0.56, 0.08
      53,126.9 , 3.11, 0.5, AVRG
```

Возможна замена по имени.

4.2 **REFERENCE — локальные системы координат

В этом разделе определяются декартовы системы координат, отличные от основной. Каждая система обязательно снабжается именем до восьми символов. Система задаётся тремя векторами в основной системе координат: вектором смещения начала координат (см), вектором направления оси X и вектором направления оси Y. Вектора направлений не обязаны быть нормированными. Ортогональность векторов направлений проверяется, и если угол между ними отклоняется от 90 градусов не больше, чем на 5 градусов, то направление Y, если надо, подправляется в плоскости XY так, чтобы вектора стали ортогональными. Если же отклонение слишком большое, то выдаётся сообщение и работа аварийно завершается.

Пример:

```
SYS1: 0,0,0, 1,1,0,-1,1,0
SYS2: 100,50,0, 1,0,0, 0,1,0
```

Возможна замена по имени.

4.3 **PARTICLES — свойства частиц

Таблица свойств частиц является самым консервативным разделом входного потока: частицы нежелательно менять местами, изменять массы или заряды, т.к. многие программы моделирования взаимодействий настроены на определённые порядковые номера частиц в таблице частиц (для ускорения счёта). Особенно это относится к фотону, электрону и позитрону. Тем не менее, возможность замены характеристик ранее введённых частиц оставлена, что даёт возможность модификации свойств частиц и расширения таблицы за счёт включения новых частиц.

Возможность DELETE для свойств частиц заблокирована.

Описание каждой частицы начинается с имени (до восьми символов), затем двоеточие и через запятую: масса, заряд в единицах позитронного

заряда, спин в единицах $\hbar$, полный магнитный момент частицы в единицах ядерного магнетона.

На этом описание частицы можно закончить, если она стабильная. Для нестабильных частиц дальше можно написать произведение времени жизни частицы на скорость света (см) и описание каналов распада.

Каждый канал распада описывается его вероятностью и в круглых скобках списком продуктов распада. Сумма вероятностей распада по всем каналам не обязана быть равной единице, программа ввода отнормирует их после ввода на единицу. Пример:

```
ELECTRON: 0.511, -1, 0.5, -1836.3,
MU- :     105.66,-1, 0.5, -8.88, 6.58E+4,
          100 (ELECTRON), 2.2E-5 (ELECTRON, POSITRON,
          ELECTRON)
```

Возможна замена по имени.

4.4 **CUTS — константы, управляющие развитием каскада частиц и задающие правила выбора очередного перемещения

Каждый комплект чисел, управляющих развитием каскада, начинается с имени комплекта (до 4 символов) и двоеточия, затем записывается:

- максимальный шаг (см) частиц, на движение которых влияет магнитное поле,
- максимальный шаг заряженных частиц в веществе (г/$см^2$),
- максимально допустимое изменение энергии частицы на элементарном шаге (МэВ),
- максимальная доля кинетической энергии частицы, которую она может потерять на элементарном шаге,
- пороговая энергия γ-квантов в процессе тормозного излучения,
- пороговая кинетическая энергия рождённых электронов в процессе рассеяния на электронах атомов,
- ограничение на средний угол рассеяния частиц в градусах.

Примечание. В применении ограничений есть особенность: из двух ограничений на потерю энергии на каждом перемещении выбирается более мягкое (т.е. позволяющее сделать бóльший шаг); то же самое относится к ограничению на перемещение в г/$см^2$ и ограничению на средний угол многократного рассеяния — из двух выбирается более мягкое.

Пороговые энергии используются не для "выбраковки" частиц из каскада, а задают минимальную энергию рождающихся частиц. Если надо ввести пороговые кинетические энергии для выбывания частиц из каскада (в том числе для фотонов, электронов, позитронов), то их следует далее перечислять через запятую в форме: название частицы (кин. энергия, МэВ).

Например:

```
CUT1:10, 1, 10, 0.7, 2, 2, 10, PHOTON(0.5),ELECTRON(2)
CUT2:10,10,100, 0.2, 10, 0.5,2, NEUTRON(100)
```

Частицы, достигшие порога, выбывают из каскада, кинетическая энергия заряженных частиц и фотонов переносится в энергию возбуждения среды EXCIT в общем блоке /UMB014/. В случае особого поведения частицы в остановке включается специальная программа моделирования остановки частицы данного сорта (см. описание точки SPECEND в разделе 4.5). Эта программа может добавлять энерговыделение в переменные DENER или EXCIT в общем блоке /UMB014/. Если такая программа отсутствует или же она выдаёт флаг "ложного" взаимодействия UFALS=1 в общем блоке /UMB029/, то для нестабильных частиц моделируется распад.

Возможна замена по имени.

4.5 **ROUTINES — регистрация подпрограмм

В этой части описания перечисляются программы, которые необходимо включить в сборку. Программы разбиты на несколько комплектов, которые отвечают различным функциональным требованиям к этим программам. Каждый такой комплект имеет свое фиксированное имя, описание каждого комплекта начинается с этого имени. Для краткости назовём *поименованной точкой программы UNIMOD2* место в головной программе UM1000.FOR, куда могут быть включены программы одного комплекта.

Список программ в комплекте может быть пустой (в этом случае имя комплекта можно не писать), порядок перечисления комплектов также не фиксирован.

В некоторых точках программы должны выполнять определённые действия и их формальная структура фиксирована (используемые общие блоки и список формальных параметров). В описании такой точки есть пометка символом (*). Требования к этим программам будут описаны в разделе 8. В остальных точках работа программы определяется автором программы.

Ниже следует описание каждого комплекта с пояснением, на какой стадии моделирования эти программы вызываются и иногда с краткими примерами.

INPUT: программы для начального ввода, подготовки констант и распечаток, например:

INPUT: SUBR1, SUBR2, SUBR3(UPAR1)

EVSTART: перед началом моделирования очередного события.

PRIMARY*: программы первичного моделирования.

NEXTPART: перед началом моделирования прохождения очередной частицы через систему.

NEXTSURV: перед началом второго просмотра буфера ожидания частиц при наличии выделенного блока (см. раздел 4.19).

FINISH: конец события, гистограммные параметры и формулы отбора ещё не вычислены.

EVEND: конец события, гистограммные параметры события и формулы отбора уже вычислены, событие занесено в гистограммы.

NEXTEN: конец очередного временно́го интервала счёта (но после завершения моделирования очередного события).

RESULT: конец счёта, распечатка результатов.

PARAMETR*: программы вычисления гистограммных параметров. Этот комплект программ имеет такую особенность, что не все программы, написанные в задании, войдут в сборку. Оставлены в сборке будут только те программы, выходные параметры которых

используются в формулах, условиях на параметры, в гистограммах или как входные параметры других гистограммных программ, которые оставлены в сборке.

В заказе на счёт надо писать имя программы, затем открывающую скобку, перечислить имена входных параметров, поставить точку с запятой, затем перечислить имена выходных параметров (не более 4 символов в имени) и закрыть скобку.

Если входных параметров нет, тем не менее после открывающей скобки надо писать точку с запятой.

Выходные параметры должны быть обязательно!

Пример:

```
SUBR1(AIN1,AIN2,AIN3;AUT1,AUT2,AUT3)
```

MOVE: очередное перемещение частицы. Характеристики частицы до перемещения и после перемещения находятся в общих областях /UMB012/ и /UMB013/, соответственно.

GEOMETRY*: геометрические программы. Вслед за названием программы в круглых скобках следует писать целое число или пробел. Целое число означает, сколько параметров требуется программе для описания геометрической фигуры (нуль здесь используется на равных правах). Пробел означает, что количество параметров может меняться от блока к блоку, и программа сама будет контролировать количество параметров.

CHAMBER*: программы генерации сработавших проволочек на проволочных камерах. Так же, как для геометрических программ, в скобках надо указывать количество геометрических параметров.

COUNAMPL*: генерация отклика детектора в счетчиках. В скобках после имени программы записывается число — сколько аналоговых величин вырабатывает программа.

PROCESS*: программы моделирования взаимодействия частиц с веществом.

В описании за каждой программой в круглых скобках надо перечислять имена частиц, которые могут испытывать это взаимодействие при движении в детекторе, например:

```
PROCESS: BREMSS(ELECTRON, POSITRON), PAIRPR(PHOTON)
```

Если вместо названия частицы написано Q/=0, то программа моделирует взаимодействие, в котором участвуют все заряженные частицы. Если вместо названия частицы написано ##, то это означает, что все частицы участвуют в этом взаимодействии, например:

```
CHARG(Q/=0),  NEWINT(##)
```

Последнее практически означает, что программа NEWINT должна сама анализировать тип частицы.

Для удаления ранее введённой программы можно использовать DELETE

IONIZLS*: программа генерации ионизационных потерь заряженными частицами. Здесь можно ввести только одну программу. Если ввести ещё одну программу, то старая программа автоматически удаляется.

Для удаления ранее введённой программы можно использовать DELETE, например:

```
IONIZLS: UM1630(DELETE),
```

при этом удалится ранее введённая программа UM1630 и заряженные частицы не будут терять энергию при перемещении в веществе.

MAGFIELD*: программы, вычисляющие вектор магнитного поля в точке, где находится частица, без учёта общего множителя ко всем полям.

ENDLIFE: конец жизни частицы. Характеристики частицы в конце пробега и продукты взаимодействия уже занесены в общую область /UMB015/ с историей события.

SPECEND*: программы моделирования остановки частицы. В скобках за названием программы необходимо указать имя частицы, при остановке которой в веществе (или достижении пороговой кинетической энергии) необходимо вызвать данную программу. Перед

обращением к такой программе кинетическая энергия заряженных частиц и фотонов заносится в переменную EXCIT в /UMB014/ и их полная энергия полагается равной массе. Энергия нейтральных частиц не меняется. Программа моделирования взаимодействия в остановке может модифицировать переменные DENER и EXCIT, что может впоследствии быть учтено программой моделирования отклика детектора. Если программа моделирования остановки выставит флаг "ложного взаимодействия" UFALS=1 в /UMB029/, то для нестабильных частиц генерируется распад.

Имена программ могут повторяться, но с разными типами частиц.

MULTIPL*: программа генерации угла многократного рассеяния заряженных частиц на атомах. Можно ввести только одну программу многократного рассеяния. При вводе ещё одной программы старая программа заменяется на новую. Для удаления ранее введённой программы можно использовать DELETE

DECAYS*: список программ распадов. Для каждой нестабильной частицы и каждого её канала распада регистрируется программа распада (программы могут повторяться!) в следующей форме: название программы и в круглых скобках имя частицы, которая распадается, затем имена частиц — продуктов распада (все через запятую). Например:

```
DECAYS: UM1065(K0/S,PI+,PI-), UM1065(K0/S,PI0,PI0)
```

Возможен особый случай, когда задана ненулевая вероятность распада по некоторому каналу, но не описана программа распада. Тогда при распаде по этому каналу частица будет бесследно исчезать и для первых нескольких случаев будет напечатано сообщение.

CHAMPL*: программы генерации аналоговых сигналов с проволочек камер. После каждой программы в скобках — количество аналоговых сигналов с каждой проволочки.

ENTRANCE: программы обработки входа-выхода заряженных частиц в геометрические объёмы.

4.6 **GATES — временны́е ворота

Вводятся временны́е ворота электронных устройств в нсек. Каждому комплекту (от-до) присваивается свое имя (до четырёх символов).

Пример:

```
GAT1: 50,100,  GAT2:  0,1000
```

Возможна замена по имени.

4.7 **RESPONCES — отклик детектора в счётчиках

Вводится описание "амплитуд" со счётчиков. Описание каждой амплитуды начинается с имени (до шести символов), затем двоеточие и имя программы преобразования характеристик последнего шага частицы в отклик детектора (в простейшем случае — добавка энерговыделения в амплитуду). Если имя опущено, то берётся имя программы для амплитуды, введённой перед этим. После имени программы можно написать имя временны́х ворот из раздела 4.6. В этом случае программа добавки амплитуды вызывается только при попадании текущего времени частицы в интервал ворот.

```
AMP1:  PRGAM1,   AMP2:  ,   AMP3:  ,  AMP4:  ,
AMP5:            AMP6: ,GAT1,  AMP7: ,   AMP8: PRGRM2,GAT2
```

4.8 **BLOCKS — описание блоков вещества

Здесь вводятся описания геометрических объёмов — блоков. Это описание, как правило, самое трудоёмкое, поэтому при описании можно многие величины определять по умолчанию.

Определение блока начинается с имени (до восьми символов) и двоеточия. Затем через запятую вводятся остальные характеристики. Так как ввод позиционный, то в случае пропуска какого-либо параметра его место всё же необходимо отмечать запятой. После двоеточия перечисляются:

- Имя блока, в который вложен данный. Если имя опущено, то принимается, что блок не вложен ни в какой другой.
- Имя вещества. Если имя опущено, то полагается, что блок пустой (вакуум).
- Имя локальной системы координат, к которой будут относиться геометрические характеристики объёма. Если имя опущено, блок задаётся в основной системе.

- Имя комплекта чисел, управляющих развитием каскада в данном блоке. Если имя опущено, берётся имя из последнего введённого блока.
- Имя амплитуды, которая регистрируется при прохождении заряженных частиц через этот счётчик. Если имя опущено, амплитуда отсутствует (поглотитель). Допускаются ссылки от нескольких блоков на одну и ту же амплитуду, при этом правильность учёта амплитуд зависит от программы моделирования амплитуды.
- Способ розыгрыша пробегов заряженных частиц (1 или 2). По умолчанию полагается равным 1 — это значит, что коэффициент поглощения при переходе из блока в блок вычисляется заново (подробнее об этом см. в 5). Любой способ розыгрыша пробега приводит к правильному результату, однако от способа розыгрыша может зависеть время счёта.
- Способ розыгрыша пробегов нейтральных частиц (1 или 2). По умолчанию полагается равным 1.
- Имя геометрической программы, которая обслуживает этот тип геометрии. Если имя опущено, то принимается, что тип геометрии такой же, как у последнего введённого блока. Затем сразу же, без запятой, после имени в круглых скобках перечисляются геометрические числовые параметры блока. Подразумевается, что параметры блока являются числами с плавающей точкой.

Вместо списка чисел может быть ссылка на тождественный список чисел ранее введённого блока в виде: *=имяблока* (это полезно для экономии оперативной памяти и упрощения входного файла).

Пример:

```
BLOCK1:,        AIR,    ,   CUT1,,    .   1,1, UM1814(0,12,0,0,5)
BLOCK2:BLOCK1,Fe,    SYS1,,    AMPL1,1,1,      (-50, 0,60)
BLOCK3:BLOCK1,Fe,    SYS2,,    AMPL1,1,1,      (=BLOCK2)
```

4.9 **CHAMBERS — описание проволочных камер

Описание каждой проволочной камеры начинается с имени (до шести символов) и двоеточия, затем:

- имя системы координат, в которой заданы геометрические параметры камеры, или пробел, если в основной системе координат;
- имя блока, в который вложена камера;
- имя временны́х ворот или пробел;
- количество проволочек (не больше 32767);
- название программы, обслуживающей данный тип камер, и сразу за названием в круглых скобках, через запятую, перечисляются геометрические параметры камеры в локальной системе координат (числа с плавающей точкой, сколько чисел — зависит от используемой программы).

Если проволочки при срабатывании выдают амплитуды, которые надо регистрировать индивидуально, то описание на этом не кончается. Далее через запятую записывается название программы, генерирующей амплитуду с проволочки. Пример:

```
CHAMB1:   , BLOK1, GAT1,500, CHAMBR(100,355.5,0.1), AMPWIR
CHAMB2:SYS5, BLOK3,GAT2,1000, CHAMBR(100,355.5,0.1),
```

4.10 **HITS — амплитуды с проволочных камер

Вводятся описания суммарных амплитуд с камер. В одну амплитуду суммируются амплитуды с отдельных проволочек, а с каких проволочек суммировать амплитуды, задаётся интервалом номеров проволочек (нумерация проволочек в каждой камере своя, начиная с 1). Края интервала включаются в суммирование. Сколько аналоговых сигналов снимается с одной проволочки и что происходит, если одна проволочка срабатывает несколько раз — определяется программой генерации сигнала.

Описание каждой амплитуды начинается с имени амплитуды (до восьми символов), затем двоеточие, название программы, которая используется для генерации амплитуды с каждой проволочки, имя камеры, и через запятую два целых числа, определяющих крайние номера проволочек, например:

AMPWIR1: AMPWIR, KAMP1, 100, 200

Числа не должны превышать 32767.

4.11 **DISCRIMINATORS — типы дискриминаторов амплитуд

Типы дискриминаторов описываются следующим образом: записывается название программы, обслуживающей данный тип, затем в скобках указывается, сколько входов должно быть обеспечено для данного типа. Если это количество не указано (скобки в этом случае тоже опускаются), то дискриминатор может работать с переменным количеством входов. Пример:

DISCR1, DISCR2, DISCR3(2), DISCR4(4)

4.12 **LOGICMODULES — типы блоков логической электроники

Описываются типы блоков логической электроники. Каждый блок может иметь несколько входов (или ни одного, если программа, обслуживающая этот тип блока, такое допускает) и обязательно один выход. На выходе блока генерируется в соответствии с информацией на входах нуль или единица. Собственное имя каждого конкретного блока отождествляется с именем выхода с этого блока. Описанием типа блока является имя программы, которая моделирует работу блока. После имени в скобках можно указать, сколько входов должно быть обеспечено. Если эта информация вместе со скобками опущена, то блок может работать с любым количеством входов.

Пример:

PRBL1, PRBL2, BLCOIN, BLBLOR, BLOCK1(0)

4.13 **TRIGGER — схема электроники детектора

Описание каждого блока электроники начинается с присвоения ему какого-либо имени (до четырёх символов), затем двоеточие, название типа блока (имя программы, обслуживающей такой блок) и в скобках список дополнительной информации о входах в этот блок (имена блоков, выходы с которых поданы на вход данного блока, или список амплитуд для дискриминаторов). Если блок является дискриминатором, то после списка входов и закрывающей скобки записываются два числа — пороги. Использование этих порогов определяется программой моделирования дискриминатора. Пример:

D1: DISCR1(AMP1,AMP2,AMP3),0.5,200,

D2: DISCR2(WAM1,WAM2),1,20,

CC1: PRBL2(D1,D2)

При указании амплитуд на входе в дискриминатор можно в дополнительных круглых скобках после имени амплитуды записывать уточняющую информацию — номер единицы аналоговой информации в "амплитудном" комплекте, если программа генерации амплитуд генерирует больше одной величины одновременно. Если это не указано, то подразумевается 1. Пример:

D1: DISCR1(AMP1(2),AMP2(2),AMP3),10,20

Именем амплитуды на входе в дискриминатор может быть амплитуда со счётчика или суммарная амплитуда с камер.

4.14 **CONSTRAINTS — список условий на параметры событий

Вводятся условия на параметры. Результаты проверки этих условий ("да"=1, "нет"=0) могут использоваться при вычислении логических формул или непосредственно для отбора событий в гистограммы. Для лучшего понимания той части описания, которая связана с гистограммами, можно прочитать препринт [12].

Описание условия начинается с имени условия (до 4 символов), затем двоеточие, имя параметра и два граничных значения. Если имя условия опущено вместе с двоеточием, то принимается, что имя условия совпадает с именем параметра. Пример:

SEL1: DTET, -0.5,10, SEL2: DTET,-0.5,15, DFI ,-5,5,

4.15 **FORMULAS — логические формулы отбора событий

Логические формулы отбора могут содержать имена ранее определённых формул или условий на параметры, круглые скобки любой вложенности, символы операций логического сложения (+), умножения (*), отрицания (/). При вычислении формулы наивысший приоритет имеют скобки, меньший приоритет — отрицание, затем умножение, затем сложение. Текст формулы записывается сразу после имени, назначенного этой формуле, (до 4 символов) и двоеточия, например:

FOR1: SEL1+SEL2*DFI, FOR2: FOR1+/FOR1

4.16 **HISTOGRAMS — заказ на построение гистограмм

Описание каждой гистограммы состоит из имени (до восьми символов), двоеточия, имени условия отбора в гистограмму, сразу за ним в круглых скобках размерность гистограммы, затем параметр, центр, шаг гистограммы. Если условие отбора не указано, то в гистограмму будут занесены все события.

Размерность гистограммы нельзя указывать произвольно, в настоящее время возможны такие варианты: (0) — "центр, шаг" при этом не следует писать; (30), (80) — одномерные гистограммы с числом каналов 30 и 80, соответственно; (24*24) — двумерная гистограмма 24 на 24 канала, при этом данные о параметре, центре, шаге следует писать в двух экземплярах, для каждой оси свой комплект.

Имя гистограммы вместе с двоеточием можно не писать, в этом случае имя состоит из пробелов. Имена гистограмм не обязаны быть уникальными, в случае совпадения выдаётся сообщение и работа продолжается.

Если первым или третьим символом имени 80-канальной гистограммы является "@", то гистограмма будет напечатана в виде, удобном для просмотра на экране дисплея. Если первые два символа имени "h/", то такая гистограмма будет записана в файл прямого доступа в формате PAW [13].

Примеры:

HIST1: SEL1(0) ,PAR1,
HIST2: SEL2(30),PAR2,0,0.1
HIST2: (30),PAR2,0,0.1
HIST3: SEL3(80),PAR3,100,1
@HIS3: SEL3(80),PAR3,100,1
HIST4: SEL4(24*24),PAR1,0,0.1,PAR2,10,2,
(24*24),PAR3,100,1,PAR2,10,2

4.17 **MFIELD — описание способа вычисления магнитного поля

Начинается описание каждого способа вычисления с имени (до четырёх символов), затем двоеточие, тип описания (кодовое сочетание букв, возможны три типа: MP — нестандартная программа вычисления магнитного поля, UF — однородное поле, SP — кубический сплайн

дефекта 2 специального вида). Для всех трёх типов представление последующей информации разное:

MP — после MP записывается запятая, затем имя программы вычисления магнитного поля и сразу за ним в круглых скобках через запятую перечисляются вспомогательные числа для этой программы,

UF — через запятую перечисляются три проекции магнитного поля на óси основной системы координат,

SP — после запятой записывается N1 — количество узлов сплайна по оси X и в скобках перечисляются X-координаты этих узлов, затем N2 — количество узлов по координате Y и в скобках Y-координаты этих узлов, N3 — количество узлов по координате Z и в скобках Z-координаты этих узлов и, наконец, перечисляются через запятую все $12 * N1 * N2 * N3$ коэффициентов сплайна в узлах (по 4 коэффициента в каждом узле для каждой из трёх компонент поля). Сначала перечисляются все коэффициенты для X-компоненты, затем все для Y-компоненты, затем — для Z-компоненты. Перебор узлов идет таким образом: сначала для первого узла по Y и Z координатам перебираются узлы по X-координате, затем номер узла по Y-координате увеличивается на единицу и снова перебираются все узлы по X-координате и т.д. Четыре коэффициента сплайна в каждом узле имеют следующий смысл: значение сплайна (величины соответствующей компоненты поля) в этом узле, частная производная по переменной X, затем по Y и по Z.

На основе этих данных значение поля будет вычисляться программой Н.А.Грозиной SPLINE.

4.18 **MVOLUMES — описание отсеков магнитного поля

Описываются "отсеки" магнитного поля — геометрические области пространства, в которых по-разному задан способ вычисления магнитного поля. Описание начинается с имени отсека (до 4 символов), затем после двоеточия: имя отсека, в который вложен данный отсек (если это имя опущено, предполагается, что отсек не вложен ни в какой другой);

имя матрицы преобразования к локальной системе координат (если отсек задан в основной системе, то это имя опускается); имя способа вычисления магнитного поля в данном отсеке; имя геометрической программы, "обслуживающей" данный отсек и сразу же за этим именем в круглых скобках перечисляются геометрические параметры этого отсека (в формате, определяемом обслуживающей программой). Пример:

OTC1: ,,SPL1,UM1700(100,20)
OTC2: OTC1, MAT1, FIL1, UM1007(-100, 100, -10, 10, -1, 1)

4.19 **REGIME — общие константы

Под разными фиксированными именами вводятся числа (одному имени соответствует одно или несколько чисел). В задании эти числа должны следовать за двоеточием, в данном описании за двоеточием следует характеристика этих чисел.

RAND: исходное случайное число (нормальное представление — десятичное, для обозначения 16-ричного числа в начале числа следует поставить символ Z, как в фортрановском формате, и затем не более шестнадцати 16-ричных цифр. Случайное число должно быть нечётное. В случае необходимости число дополняется слева нулями). Если число вводится в десятичном виде, оно должно быть меньше чем 2^{32}.

ENERGY: энергия одного пучка, разброс энергий в одном пучке (МэВ).

INTREGION: координаты центра области встречи (3 числа, см), среднеквадратичные отклонения места встречи от среднего по каждой координате.

MARKEDBLOCK: имя выделенного блока. Если таким образом выделить какой-либо блок, то каскад частиц развивается в два прохода: сначала рассматриваются только те частицы, начало траектории которых лежит внутри указанного блока, затем есть возможность вызвать какие-либо программы пользователя (см. раздел 4.5, группа программ NEXTSURV) и только потом рассматриваются все остальные частицы. Функцией таких программ пользователя может быть принятие решения, стоит ли рассматривать это событие до конца или уже очевидно, что это событие не удовлетворит каким-либо критериям.

MAXLIFE: предел по собственному времени частиц от начала события (наносекунды). По достижении этого времени частицы выбывают из каскада. По умолчанию 10^{10}.

ENDTASK: максимальная статистика, запас по времени — целое число секунд процессорного времени.

FIELDFAC: общий множитель к магнитным полям в системе (если он равен нулю, то считается, что полей вообще нет). По умолчанию полагается равным нулю.

TIMESTEP: интервал в секундах процессорного времени между выходами на точку программы NEXTEN (см. раздел 4.5). По умолчанию полагается равным (-1). В этом случае выхода на точку NEXTEN не делается.

PRINT: целое число — уровень печати о развитии события,

= 0 - нет печати,
= 1 - начало-конец события, исх.случайное число,
= 2 - начало и конец жизни каждой частицы,
= 3 - параметры каждого перемещения всех частиц.

Более высокий уровень всегда включает в себя предыдущие.

Пример:

ENERGY: 4730, 4.5

ENDTASK: 5000, 10,

PRINT:1

4.20 **PRIVATE — ввод текстов подпрограмм пользователя

Вводятся тексты программ на Фортране-77 в файл UM1000.FOR. Тексты никуда не записываются на хранение, транслируются вместе с головной программой и автоматически включаются в сборку.

Если эти программы ниоткуда не вызываются, то они просто будут занимать оперативную память. Таким образом, в нормальных случаях

текст программы должен быть введён в данном разделе, а её вызов должен быть организован из какой-нибудь точки программы (см. раздел 4.5).

Такими программами могут быть, например, программы вычисления гистограммных параметров, программы нестандартной обработки событий и т.д.

4.21 **PRESS — условия на блоки вещества

В данном разделе вводятся маски на имена блоков, в соответствии с которыми часть блоков может быть удалена. Маска состоит из символов, которые должны совпадать с соответствующими символами имени, и символов "*", которые означают "любой символ" (один). Маска, как и обычное имя блока, при необходимости дополняется справа пробелами.

Запись списка масок начинается с обозначения одного из двух режимов:

REJECT: блок выбрасывается, если он удовлетворяет хотя бы одной из последующих масок,

REMAIN: блок оставляется, если он удовлетворяет хотя бы одной из последующих масок, все остальные блоки выбрасываются.

Удаление блоков осуществляется сразу же после ввода списка масок после REMAIN или REJECT и этот список забывается. Так что при повторном использовании такой возможности чистка осуществляется независимо, а не путем объединения списков масок.

После операции чистки выдается сообщение только о количестве удалённых блоков.

При удалении блока, в который вложен другой блок или камера, происходит "переподчинение" другому блоку без всякого сообщения. Эффект такого переподчинения должен быть учтён самим автором задания. Необходимо также учитывать, что если блок является счётчиком, то при его удалении амплитуда со счётчика всегда будет нулевой. Количество масок в списке ограничено числом 100.

Примеры:

```
REJECT: DP**V, ABC*****, ******OO,
REMAIN: VACU****, COOR****,
```

4.22 **CLEAN — удаление "лишних" программ вычисления гистограммных параметров события

Лишними признаются программы, выходные параметры которых не упоминаются в условиях на параметры, заказе на гистограммы и не используются в качестве входных параметров других программ, признанных нужными в данном счёте. Эта операция в любом случае проделывается после обработки всего заказа на счёт (нет возможности её отключить), и её специальный заказ может иметь смысл только для того, чтобы по команде $ROUTINES распечатался не полный список программ вычисления параметров, а только те программы, которые будут "работать".

Процедура удаления лишних программ вычисления параметров события позволяет иметь отдельный полный файл описания всех программ вычисления параметров, из которых в сборку войдут только те, которые необходимы для построения гистограмм в данном счёте.

4.23 Распечатка введённых данных

Заказ на распечатку данных из любого раздела производится на отдельной строке: в первой позиции должен быть "$", за ним должно следовать ключевое слово из вышеприведённого списка разделов, например

```
$REGIME
```

вызовет распечатку данных из раздела REGIME (см. 4.19). Появление команды распечатки в каком-либо месте входного потока вызывает немедленную печать информации из соответствующего раздела данных, поэтому запрос на распечатку каких-либо данных должен располагаться после ввода соответствующих разделов. Два названия разделов нельзя использовать для заказа на распечатку: PRESS и CLEAN. Кроме названий разделов заказ на распечатку может состоять из ключевого слова START (начать распечатку всех строк заказа без преобразования формата), STOP (прекратить распечатку всех строк, начатую командой $START), LIMITS (распечатка таблицы максимально возможного количества блоков, камер, гистограмм и т.д.)

5 Моделирование взаимодействия частиц с веществом

В этом разделе кратко описывается текущий набор программ моделирования процессов взаимодействия частиц с веществом детектора. В программе UNIMOD2 используются три вида подпрограмм, моделирующих процессы, изменяющие состояние частиц.

Первый тип подпрограмм — это моделирование *непрерывных* взаимодействий: ионизационных потерь и многократного рассеяния заряженных частиц на атомах вещества. Непрерывными эти взаимодействия названы потому, что вероятность каждого акта взаимодействия велика, вследствие этого рационально учитывать суммарный эффект этих взаимодействий на некотором отрезке траектории. Программы моделирования ионизационных потерь и многократного рассеяния должны быть написаны по определённым правилам и ссылки на них должны быть в заказе на счёт.

Следующий тип подпрограмм — моделирование распадов нестабильных частиц. Случайный выбор момента распада частицы полностью определяется временем жизни частицы и не требует написания специальных программ для каждой частицы (осуществляется головной программой). Напротив, моделирование продуктов распада, как правило, специфично для каждого типа исходной частицы. В описании канала распада каждой нестабильной частицы должно быть указано имя программы, моделирующей характеристики продуктов распада.

Еще один тип подпрограмм моделирует взаимодействия, которые, в отличие от ранее введённых *непрерывных*, можно назвать *точечными*. Эти программы имеют два режима работы: в первом режиме они должны выдавать головной программе вероятность данного типа взаимодействия на единицу длины траектории частицы, во втором режиме — генерировать характеристики продуктов взаимодействия. Точка взаимодействия выбирается случайно в соответствии с суммой вероятностей взаимодействия по всем процессам, затем в точке взаимодействия выбирается случайно процесс взаимодействия с вероятностью, пропорциональной вероятности взаимодействия на единицу длины траектории. Когда процесс выбран, происходит обращение к соответствующей программе во втором режиме, где она должна генерировать продукты взаимодействия.

Траектория заряженных частиц представляет собой ломаную линию, отрезки которой при отсутствии магнитного поля — прямолинейные,

а при наличии магнитного поля представляют собой отрезки спирали. В точках излома производится учёт суммарного эффекта непрерывных взаимодействий: уменьшение энергии за счёт ионизационных потерь и изменение направления движения за счёт многократного рассеяния.

В точке остановки заряженных частиц возможен ещё один тип взаимодействия — в покое. Для каждого типа частицы можно в заказе на счёт указать имя программы моделирования взаимодействия, которая будет вызвана в точке остановки частицы для генерации продуктов взаимодействия.

Для повышения точности моделирования процессов при меняющейся непрерывно энергии рассматриваемой частицы для всех процессов введена возможность выставить флаг "ложное взаимодействие" в режиме генерации характеристик продуктов взаимодействия. Ненулевое значение этого флага при выходе из программы моделирования взаимодействия означает для головной программы, что исходная частица осталась неизменной и может продолжать движение, если моделировалось взаимодействие "на лету". Если такой флаг установлен программой взаимодействия в остановке, то в этом случае анализируется возможность распада (для нестабильной частицы).

Такой приём с использованием флага "ложное взаимодействие" для повышения точности моделирования при условии меняющейся энергии частицы является реализацией метода выравнивания сечения.

Быстро-переменное поведение сечения взаимодействия может быть не только из-за потерь энергии частицей, но и за счёт слоистого вещества, например, когда тонкие слои разных веществ следуют друг за другом. Для моделирования прохождения частиц через такие системы в UNIMOD2 можно пользоваться стандартным подходом (первый способ розыгрыша пробега), при котором частицы последовательно проводятся через каждое вещество.

При этом вероятности взаимодействий вычисляются только для того вещества, в котором находится частица в данный момент. Затем происходит смещение частицы в точку взаимодействия, но если по дороге встречается граница блока раньше точки взаимодействия, то на границе делается остановка, вычисляется вероятность взаимодействия для вещества другого блока, и розыгрыш пробега производится заново.

Кроме того, существует возможность использования метода выравнивания сечения (второй способ розыгрыша пробега). При таком способе розыгрыша взаимодействия в качестве вероятности взаимодействия выбирается максимальная вероятность по всем блокам. В соответствии с этой завышенной вероятностью генерируется пробег, и теперь розыгрыш пробега на каждой границе блока не производится вплоть до точки взаимодействия. А в точке взаимодействия принимается (случайно) решение, что взаимодействие действительно произошло, с вероятностью, равной отношению действительной вероятности взаимодействия в этой точке при текущих параметрах частицы в данном веществе к вероятности, использованной для розыгрыша пробега. Если принимается решение, что взаимодействие не произошло, то движение частицы продолжается.

Пользователь в заказе для каждого блока вещества отмечает, какой способ розыгрыша пробега (1 или 2, по умолчанию 1) должен использоваться для нейтральных и заряженных частиц. Отметим, что вопрос выбора способа розыгрыша пробега не имеет отношения к точности моделирования — и в том, и в другом случае результаты получаются одинаковыми (с точностью до статистики). От выбора способа розыгрыша пробега может зависеть только время моделирования. В большинстве случаев предпочтителен стандартный способ (номер 1), однако, в особых случаях способ номер 2 может дать существенный выигрыш во времени (такой случай подробно исследовался в [14]).

Ниже следует краткое описание процессов взаимодействия, моделируемых в UNIMOD2 в настоящее время.

5.1 Рождение e^+e^- пар фотоном в поле ядра

Моделирующая подпрограмма — UM1210. Полные сечения процесса взяты из [15]. Для генерации вторичных частиц используются дифференциальные сечения [16]. Процесс моделируется для атомного номера в интервале 1 – 94, энергии фотона от $2 \cdot m\,c^2$ до 10^6 МэВ. Подробно моделирование рождения пар описано в [17].

5.2 Тормозное излучение электронов и позитронов

Моделирующая подпрограмма — UM1200. Для моделирования процесса используются дифференциальные спектры, полученные в работах [18, 19]. Атомный номер вещества в интервале 1–92, энергия начальной частицы 1 кэВ – 10000 МэВ. Моделирование тормозного излучения подробно описано в [17].

5.3 Комптон-эффект и рэлеевское рассеяние

Моделирующие подпрограммы — `UM1230` (Комптон-эффект) и `UM1250` (рэлеевское рассеяние). Для моделирования неупругого рассеяния фотонов используются полные сечения [15, 20, 23] и дифференциальное сечение для свободных электронов [25]. Для моделирования упругого рассеяния используются полные сечения [15] и дифференциальные сечения [4]. Атомный номер в интервале 1–94. Алгоритмы моделирования обоих процессов изложены в [17].

5.4 Фотоэффект

Моделирующая подпрограмма — UM1240. Для моделирования используются полные сечения [15]. Описание алгоритма моделирования процесса содержится в [17]. Атомный номер вещества в интервале 1–94.

5.5 Ионизационные потери

Моделирующая подпрограмма — UM1630. Подробное описание алгоритма моделирования средней величины потерь и флуктуаций приведено в [5], алгоритм розыгрыша флуктуаций отдельно описан в [21, 22]. В ионизационных потерях учитываются только возбуждения атомных электронов с передачами энергии меньше установленного порога рождения δ-электронов (см. раздел 4.4). Поправка на эффект плотности даётся формулой из работ Штернхаймера [24].

Кроме того, для лёгких частиц к непрерывным потерям отнесены потери электронов и позитронов на излучение мягких фотонов (с учётом флуктуаций) с энергией ниже установленного порога на излучение тормозных γ-квантов (см. раздел 4.4). Подробно об этом написано в [5].

5.6 Аннигиляция позитронов

Моделирующая программа — UM1220. Для моделирования используется полное сечение [20] и дифференциальное сечение [25]. Алгоритм моделирования изложен в [17].

5.7 Рассеяние заряженных частиц на электронах атомов

Моделирующая программа — UM1221. Рассеяние заряженных частиц на электронах атомов моделирутся с помощью алгоритма, описанного в [17]. Имеется несколько модифицированная версия UM1363.

5.8 Тормозное излучение мюонов

Моделирующая программа — UM1635. Формулы для описания процесса взяты из [26].

5.9 Многократное рассеяние

Моделирующая программа — UM1611. Процесс многократного рассеяния заряженных частиц на атомах вещества моделируется способом, описанным в [1], в соответствии с теорией Мольер и Бете [27] (первые три члена разложения в функциональный ряд). Единственное техническое отличие состоит в том, что функции $F^{(i)}$ в интегральном распределении (см. [1]) раньше представлялись быстросходящимися рядами, а теперь эти ряды интерполируются сплайном из [21], что экономит время счёта.

5.10 Распады нестабильных частиц

Имена программ, отвечающих за распады соответствующих частиц, приведены в разделе 9.11. Их также можно узнать в распечатке свойств частиц, написав в конце задания:

`$PARTICLES`

При моделировании распадов нестабильных частиц применяется тот же подход, что и в [1]. Феноменологические константы в матричных элементах взяты из [28].

5.11 Распады и взаимодействия остановившихся частиц

В результате ионизационных и радиационных потерь в веществе заряженные частицы теряют свою кинетическую энергию и могут практически остановиться. При этом отрицательно заряженные частицы

могут быть захвачены атомными ядрами вещества, позитроны аннигилируют с электронами, а нестабильные положительно заряженные частицы распадаются.

В программе UNIMOD2 предусмотрена возможность подключения программ моделирования процесса остановки частицы в тех случаях, когда кинетическая энергия частицы оказалась ниже установленного порога (см. раздел 4.5, группа программ SPECEND). Если пользователь не включает такую программу для стабильной частицы, то она просто выбывает из каскада, для нестабильной частицы моделируется распад. В библиотеке UNIMOD2 имеются программы моделирования остановок следующих частиц: $e^+, \mu^-, \pi^-, K^-, \bar{p}$, детально описанные в [29].

Остановки этих частиц моделируются следующими программами: e^+ — UM1320, μ^- — UM1323, π^- — UM1650, K^- — UM1373, $\bar{p}$ — UM1645 [32].

5.12 Ядерные взаимодействия адронов

В библиотеке программ имеется пакет программ моделирования взаимодействия адронов с ядрами атомов NUC92 [29], который включает в себя программы упругого и неупругого ядерного взаимодействия. Неупругое взаимодействие моделируется по программе NUCRIN [30, 31] (программный интерфейс — UM1371) для следующих адронов: $\pi^{\pm}, p, \bar{p}, n, \bar{n}, K^{\pm}, K_L, K_S, \Lambda^0, \Sigma^{\pm}, \Sigma^0, \Xi^0, \Xi^-$. Следует отметить, что программа NUCRIN допускает произвольные значения импульсов налетающих адронов, однако её рекомендуется использовать для адронов с импульсом 0.1 – 5 ГэВ/с. В результате моделирования генерируются вторичные частицы, вылетающие из ядра, часть энергии выделяется в виде возбуждения среды. Упругое взаимодействие моделируется для тех же адронов по программе NUCREL (программный интерфейс — UM1316), интервал допустимых импульсов в этом случае значительно шире. Частица, испытавшая упругое рассеяние, записывается в буфер вторичных частиц, её энергия не меняется.

Числа, уточняющие параметры сечения взаимодействия, считываются из файлов REACCHAN.DAT и PARTICLE.DAT. Файлы находятся в той же директории, что и программа UNIMOD2.

5.13 Взаимодействие частиц со сложными веществами

В UNIMOD2 предусмотрена возможность моделирования прохождения частицы через среду, состоящую из различных атомов. Если пользователь при вводе данных объявил вещество данного блока сложным, он должен задать атомные параметры компонент вещества и плотность каждой компоненты. Программа вычисляет эффективные параметры смеси согласно алгоритму, подробно описанному в [1]. Три эффективных параметра смеси Z, A, I (средний ионизационный потенциал) позволяют правильно усреднить характерные комбинации Z/A, Z^2/A, $(Z/A)\ln(I)$. Далее, по желанию пользователя, возможны два варианта моделирования.

В первом варианте для моделирования пробега частицы используются эффективные параметры смеси (указывается AVRG в конце списка компонентов в 4.1), во втором — эффективные параметры используются только для моделирования ионизационных потерь и многократного рассеяния заряженных частиц, а все остальные процессы взаимодействия частиц с веществом моделируются отдельно для каждой из компонент. Продукты взаимодействия моделируются всегда на атомах одного из компонентов.

6 Общие области программы

В этом разделе приведено краткое описание тех общих областей программы, которые могут представлять интерес для пользователя при написании своих подпрограмм.

При описании общих областей подразумевается, что в программе имеется неявное описание типа переменных:

IMPLICIT INTEGER *2 (U-W), REAL *8 (Q-S)

которое определяет тип переменной, если нет явного описания.

Все длины в общих областях измеряются в см, времена — в наносекундах, энергии — в МэВ.

Реальные длины буферов, размещённых в общих областях программы, зависят от входных данных; они определяются интерпретатором ввода и устанавливаются при генерации текста программы UM1000. В данном описании длина таких буферов показана либо символом ".", либо переменной, определённой в этой же общей области. В любом случае пользователю в своей подпрограмме можно указать произвольную длину

буфера, например, равную 1.

/UMB000/ UBLOCK, UCENTR, WBL(20,UBLOCK)

UBLOCK — количество геометрических объёмов (блоков),

UCENTR — номер выделенного объёма или нуль,

WBL — описание геометрических объёмов (блоков):

WBL(1-4,.) — имя блока (до восьми символов) или пробелы, если имя не назначено,

WBL(5,.) — тип блока (номер программы в списке геометрических программ в /UMB048/),

WBL(6,.) — индекс начала описания размеров и положения блока в массиве QBLOCK (/UMB001/),

WBL(7,.) — количество двойных слов в геометрическом описании,

WBL(8,.) — номер блока, в который вложен данный, или 0, если нет охватывающего блока,

WBL(9,.) — количество блоков, вложенных в данный, или 0, если нет вложенных блоков,

WBL(10,.) — индекс в UCLIST (/UMB002/) - начало списка номеров блоков, вложенных в данный,

WBL(11,.) — номер вещества или 0, если вакуум,

WBL(12,.) — номер матрицы преобразования вектора к системе блока или 0, если основная система координат,

WBL(13,.) — количество вложенных проволочных камер или нуль,

WBL(14,.) — индекс начала списка номеров вложенных проволочных камер в массиве UCHAML (/UMB005/),

WBL(15,.) — номер комплекта чисел в /UMB033/, управляющих развитием каскада в данном блоке (пороговые энергии электронов, квантов, максимальный шаг и т.д.),

WBL(16,.) — номер способа розыгрыша пробега заряженных частиц (1 или 2),

WBL(17,.) — номер способа розыгрыша пробега нейтральных частиц (1 или 2),

WBL(18,.) — индекс в UMATLS (/UMB032/) начала списка номеров веществ, содержащихся внутри данного блока (с учётом вложенных),

WBL(19,.) — количество номеров в предыдущем списке. Если оба указанных выше способа розыгрыша пробегов равны 1, то количество номеров равно нулю. То же самое, если охватывающий блок имеет способы розыгрыша пробегов 2.

WBL(20,.) — номер амплитуды в /UMB016/, которая регистрируется при прохождении заряженных частиц через данный счётчик, или 0, если блок представляет собой поглотитель.

/UMB001/ QBLOCK(.) — характеристики блоков и отсеков магнитного поля, определяющие положение, ориентацию, размеры. Количество чисел, описывающих блок, зависит от типа блока.

/UMB002/ UCLOS0, UCLIST(.)

UCLOS0 — количество номеров в начале UCLIST, представляющих список ни во что не вложенных блоков (нулевой уровень),

UCLIST — массив номеров блоков, описывающих вложенность блоков друг в друга (используется в /UMB000/).

/UMB003/ NMAT, AMAT(7,NMAT) — характеристики веществ

DIMENSION UAMAT(14,.)

EQUIVALENCE (UAMAT(1,1),AMAT(1,1))

UAMAT(1,.) — количество компонентов вещества. Для простых веществ равно 1.

UAMAT(2,.) — для сложных веществ, если не равно нулю, указывает на то, что пробег до точки взаимодействия надо моделировать по характеристикам эффективного вещества, которые располагаются после описания последнего компонента данного вещества. Индивидуальные характеристики компонентов учитываются при этом для розыгрыша продуктов взаимодействия.

AMAT(2,.) — ат.номер,

AMAT(3,.) — ат.вес,

AMAT(4,.) — плотность (г/$см^3$),

AMAT(5,.) — ср. иониз. потенциал (кэВ),

AMAT(6-7,.) — название вещества (до восьми символов). Если имя отсутствует (пробелы), то данное вещество входит в ближайшее из предыдущих веществ с именем в качестве компонента. Для всех компонентов сложного вещества плотность понимается как парциальная плотность, т.е. реальная плотность сложного вещества получается суммированием плотностей компонентов. При ссылках на сложное вещество его номер всегда равен номеру первого компонента (это значит, что в номерах веществ могут быть "просветы", например, 1, 2, 5, 6, 8, 15, 16,...).

/UMB004/ NCHAMB, WCHAM(13,NCHAMB)

NCHAMB — количество проволочных камер,

WCHAM(1-3,.) — имя камеры (до шести символов),

WCHAM(4,.) — номер блока, в который вложена камера. Реакция камеры на движение заряженной частицы моделируется только при нахождении частицы внутри указанного блока.

WCHAM(5,.) — тип камеры (номер программы в /UMB049/, "обслуживающей" эту камеру),

WCHAM(6,.) = 0, если амплитуды не генерируются,

> 0 — количество амплитуд с групп проволочек, при этом в WCHAM(7,.) — индекс первой амплитуды в /UMB019/,

< 0 — номер программы в /UMB022/ генерации амплитуд с обратным знаком, при этом в WCHAM(7,.) индекс амплитуды с первой проволочки в /UMB020/ (амплитуды генерируются с каждой проволочки и записываются в /UMB020/ подряд).

WCHAM(7,.) — вспомогательный индекс (см.WCHAM(6,.)),

WCHAM(8,.) - - индекс начала геометрического описания поверхности, образуемой проволочками, и ориентации проволочек в массиве QCHAM в /UMB006/,

WCHAM(9,.) — количество двойных слов в этом описании,

WCHAM(10,.) — номер матрицы в /UMB026/ преобразования вектора к системе камеры или нуль,

WCHAM(11,.) — количество проволочек,

WCHAM(12,.) — индекс начала "да-нет"-ной информации о срабатывании проволочек в общей области /UMB018/. Левая граница поля всегда выровнена по границе полуслова.

WCHAM(13,.) — номер временных ворот в /UMB058/.

/UMB005/ UCHAML(.) — Списки номеров вложенных в блоки камер.

/UMB006/ QCHAM(.) — Описание поверхностей, образуемых проволочками камер, и ориентации проволочек.

/UMB007/ UMFIEL, UMF(10,UMFIEL)

UMFIEL — количество отсеков магнитного поля,

UMF(1-2,.) — имя отсека,

UMF(3,.) — тип геометрии отсека (номер программы в /UMB048/),

UMF(4,.) — номер охватывающего отсека или 0,

UMF(5,.) — индекс начала списка отсеков, вложенных в данный (в массиве UFLIST в /UMB008/),

UMF(6,.) — количество вложенных отсеков или 0,

UMF(7,.) — индекс начала геометрического описания отсека в QBLOCK в /UMB001/,

UMF(8,.) — количество двойных слов в этом описании,

UMF(9,.) — номер матрицы преобразования вектора к системе отсека из /UMB026/ или нуль,

UMF(10,.) — номер описания магнитного поля в данном отсеке (индекс в UMDES в /UMB010/).

/UMB008/ UMUNCL, UFLIST(.)

UMUNCL — количество отсеков поля, не вложенных в другие отсеки (список номеров этих отсеков приводится в самом начале UFLIST).

Далее в UFLIST следуют списки номеров отсеков, вложенных в другие отсеки (пользоваться через UMF(2-3,.)).

/UMB009/ NRECO, WORDS(NRECO) — список параметров процедуры UNIMOD2 в данном счёте

CHARACTER *8 WORDS

/UMB010/ FIELD0, NFDESC, UMDES(12, NFDESC)

DIMENSION NUMDES(6,.)

EQUIVALENCE (UMDES(1,1),NUMDES(1,1))

FIELD0 — общий множитель ко всем полям в детекторе,

NFDESC — количество описаний магнитного поля,

UMDES(1-2,.) — имя описания,

UMDES(3,.) — тип описания (целое число):

0 — нестандартное описание (структуру "знает" только подпрограмма пользователя),

1 — однородное поле,

2 — кубический сплайн дефекта 2.

UMDES(4,.) — индекс в массиве FSPLI в /UMB011/ — начало списка узлов сплайна,

UMDES(5,.) — индекс в FSPLI — начало списка коэффициентов сплайна,

UMDES(6,.) — количество коэффициентов сплайна для одной компоненты магнитного поля,

NUMDES(4,.) — количество узлов сплайна по оси X,

NUMDES(5,.) — количество узлов сплайна по оси Y,

NUMDES(6,.) — количество узлов сплайна по оси Z,

Последние числа имеют указанный смысл только при описании сплайном (тип описания 2).
При типе 0 в UMDES(5,.) записывается номер подпрограммы пользователя в /UMB051/, в UMDES(6,.) — индекс первого параметра для этой программы в FSPLI, в UMDES(7,.) — количество этих параметров.
При типе 1 в UMDES(4,.) заносится индекс в FSPLI первой компоненты поля.

/UMB011/ FSPLI(.) — параметры для описания поля:

1. Для программы пользователя (тип 0) структура описания не фиксирована (программа пользователя сама должна понимать занесённые числа).
2. В случае однородного поля всё описание состоит из трёх чисел — проекций поля на оси X, Y, Z.
3. В случае сплайна сначала приводятся координаты узлов сплайна по оси X, затем — по оси Y, затем — по оси Z. Следом идут коэффициенты сплайна в узлах для X-проекции поля, потом — для Y-ой и, наконец, для Z-ой проекции поля. В каждом узле для каждой проекции приводится четыре коэффициента: значение функции в узле и три частные производные функции по X, Y, Z. Порядок перечисления узлов следующий: сначала для первого узла по Y и Z перечисляются коэффициенты для всех узлов по X, затем для второго узла по Y и первого узла по Z снова перебираются узлы по оси X и т.д.

/UMB012/ UTYP, UCOUN, UBOX, UMAT, ENERG, TIME, QR(3), QV(3), QSPIN(3), BET, GAM

Описание текущей частицы перед очередным перемещением:

UTYP — тип частицы (целое число),

UCOUN — номер блока, в котором находится частица или 0 (0, 1, 2, ...),

UBOX — номер отсека магн. поля, в котором находится частица или 0 (0, 1, 2, ...),

UMAT — номер вещества, в котором находится частица (0, 1, ...).

Для UCOUN, UBOX, UMAT номер 0 означает отсутствие номера блока, отсека, вещества.

ENERG — полная энергия частицы в МэВ,

TIME — время от начала розыгрыша события в нсек,

QR — координаты радиус-вектора частицы,

QV — единичный вектор направления скорости в пространстве,

QSPIN — направление вектора поляризации частицы в пространстве (модуль не больше 1 или $QSPIN(1) = 2$, если поляризация не задана).

BET = $\sqrt{1 - (\text{масса}/ENERG)^2}$,

GAM = *ENERG*/масса,

/UMB013/ UTYPF, UCOUNF, UBOXF, UMATF, ENERGF, TIMEF, QRF(3), QVF(3), QSPINF(3), BETF, GAMF

Общий блок характеристик частицы в конце очередного перемещения. Смысл всех чисел точно такой же, как в /UMB012/.

/UMB014/ QDL, QDRINT, DENER, DENRS, EXCIT, DTIME, WINT, WSTOP, WRANG, WSPARK, WAMPL

Параметры перемещения:

QDL — длина очередного перемещения частицы вдоль траектории,

QDRINT — остаток расстояния до точки взаимодействия,

DENER — уменьшение энергии частицы при перемещении (за счёт ионизационных потерь и приравненных к ним)

DENRS — ограничение на изменение энергии на очередном перемещении частицы,

EXCIT — энергия взаимодействия, пошедшая на возбуждение среды,

DTIME — добавка к времени жизни частицы,

WINT — номер типа "точечного" взаимодействия, случившегося в конце перемещения, или 0, если взаимодействия нет,

WSTOP — целое число, характеризующее тип "сработавшего" ограничения на величину элементарного перемещения:

= 1 максимальное перемещение заряженной частицы в магнитном поле,

= 2 максимальное перемещение заряженной частицы в веществе,

= 3 максимальное изменение энергии заряженной частицы,

= 4 граница блока,

= 5 предел по времени чувствительности детектора,

= 6 граница отсека магнитного поля,

= 7 "точечное" взаимодействие,

= 8 ионизационный пробег,

WRANG — способ розыгрыша пробега на очередном перемещении частицы (1 или 2),

WSPARK — количество номеров проволочек, добавленных в /UMB018/ на очередном перемещении,

WAMPL — количество амплитуд с проволочек, добавленных в /UMB020/ на очередном перемещении,

/UMB015/ NPAR1, NPAR2, NZAP1, NZAP2, PARTC1(28, 64), PARTC2(28, 64) — буфер частиц

```
DIMENSION QPART1(14, 64), QPART2(14, 64)
DIMENSION WPART1(56, 64), WPART2(56, 64)
EQUIVALENCE (QPART1(1,1), PARTC1(1,1))
EQUIVALENCE (WPART1(1,1), PARTC1(1,1))
EQUIVALENCE (QPART2(1,1), PARTC2(1,1))
EQUIVALENCE (WPART2(1,1), PARTC2(1,1))
```

NPAR1 — номер очередной частицы, которую надо "проводить" через детектор,

NPAR2 — количество частиц в истории события к текущему моменту,

NZAP1 — номер буфера в массиве PARTC1 (1, 2, ...),

NZAP2 — номер буфера в массиве PARTC2 (1, 2, ...),

WPART1(1,.) — номер поколения (целое число 1, 2, 3, ...),

WPART1(2,.) — номер частицы-родителя. Для первого поколения этот номер равен 0,

WPART1(3,.) — причина "смерти" частицы. Положительное число означает номер "точечного" процесса взаимодействия (в порядке описания их во входном потоке), завершившего путь частицы. 0 — частица ещё не рассматривалась, (-1) — конец ионизационного пробега (в том числе достижение пороговой энергии), (-2) — выход за пределы разрешённого временно́го интервала, (-3) — слишком много перемещений, (-4) — принято решение не моделировать оставшиеся частицы события, (-5) — распад нестабильной частицы,

WPART1(4,.) — тип частицы,

PARTC1(3,.) — энергия частицы в начале траектории,

PARTC1(4,.) — время старта частицы (нсек),

QPART1(3-5,.) — начальная точка траектории,

QPART1(6-8,.) — начальное направление вектора скорости (единичный вектор),

QPART1(9-11,.) — начальное направление вектора поляризации или первое число = 2,

QPART1(12-14,.) — конечная точка траектории,

В PARTC2, QPART2, WPART2 — такое же расположение информации, как в вышеописанных массивах. Вся история развития события группируется в блоки по 64 частицы ($112 * 64 = 7168$ байт), которые в случае превышения 128 частиц в событии записываются на магнитный диск (временно). NZAP1 и NZAP2 — порядковые номера таких блоков (буферов), находящиеся в оперативной памяти в описываемом общем блоке. Перемещение блоков частиц на диск и обратно возможно с помощью стандартной программы.
Внутреннее представленние целых чисел длиной в 2 байта, используемых для номера частицы-родителя, ограничивает число частиц в каждом событии величиной 32767, и, следовательно, максимальный потребный объём диска под историю события $= 7K \times 512 = 3584K$. Если NPAR2 > 32000, то история события после 128-ой частицы выбрасывается и полагается WPART1(2,.)= -1 для всех частиц с номером больше 128. При этом меняется порядок просмотра частиц (рассматриваются последние частицы и информация о них после моделирования их "жизни" затирается).

/UMB016/ NAMPL, UAMPL(6, NAMPL)

UAMPL(1-3,.) — имя амплитуды в счётчиках (до шести символов),

UAMPL(4,.) — номер программы преобразования энерговыделения в амплитуду. Список программ в общем блоке UMB017,

UAMPL(5,.) — номер временных ворот в /UMB058/ или нуль,

UAMPL(6,.) — индекс первой аналоговой величины в "амплитуде" в /UMB020/.

/UMB017/ UCONV, UNAMC(4, UCONV)

UNAMC(1-3,.) — имена программ преобразования энерговыделения в амплитуду,

UNAMC(4,.) — количество аналоговых величин, генерируемых программой (сюда могут входить также времена).

/UMB018/ WIRES(.) - "да-нет"-ная информация о сработавших проволочках в камерах. Единичное значение бита соответствует срабатыванию проволочки, нулевое — отсутствию срабатывания. Информация, относящаяся к любой камере, начинается с левой границы полуслова, т.е. описание сработавших проволочек в камере всегда занимает целое количество полуслов.

/UMB019/ NWAMP, UAMPWR(8, NWAMP) — амплитуды с камер.

UAMPWR(1,.) — номер программы (в /UMB022/), используемой для генерации амплитуды с отдельной проволочки,

UAMPWR(2,.) — количество проволочек с последовательными номерами, которые дают вклад в данную амплитуду,

UAMPWR(3,.) — номер первой проволочки в камере из этого интервала (нумерация проволочек в каждой камере начинается с единицы),

UAMPWR(4,.) — индекс первого слова в /UMB020/, где записывается аналоговая информация,

UAMPWR(5-8,.) — имя амплитуды (до восьми символов).

/UMB020/ AMPVA(.) — аналоговые сигналы с камер и счётчиков. До начала события всё заполнено нулями.

/UMB021/ NSPARK, USPARW(2,NSPARK) — список сработавших на очередном перемещении частицы проволочек (номер камеры, номер проволочки). NSPARK — количество сработавших проволочек.

/UMB022/ UTYPAM(4,.) — имена программ, генерирующих амплитуды на отдельной проволочке при прохождении одной частицы.

UTYPAM(1-3,.) — имя программы,

UTYPAM(4,.) — количество слов аналоговой информации, генерируемое при срабатывании одной проволочки.

/UMB023/ NTYPAR, UTYPES(16, NTYPAR) — свойства частиц

DIMENSION TYPES(8,NTYPAR)

EQUIVALENCE (UTYPES(1,1), TYPES(1,1))

NTYPAR — количество типов частиц,

TYPES(1-2,.) — название частицы,

TYPES(3,.) — масса частицы в МэВ,

TYPES(4,.) — заряд частицы в единицах заряда позитрона,

TYPES(5,.) — спин частицы в единицах $\hbar$,

TYPES(6,.) — полный магнитный момент частицы в единицах ядерного магнетона,

TYPES(7,.) — произведение скорости света на время жизни частицы (см) или 10^{30}, если частица стабильная,

UTYPES(15,.) — количество мод распада,

UTYPES(16,.) — начало списка мод распада в /UMB024/.

/UMB024/ UMODES(4,.)

DIMENSION PMODES(2,.)

EQUIVALENCE (PMODES(1,1), UMODES(1,1))

PMODES(1,.) — вероятность распада по данному каналу (сумма вероятностей равна 1),

UMODES(3,.) — количество вторичных частиц в данном канале распада,

UMODES(4,.) — индекс начала списка типов вторичных частиц в /UMB025/.

/UMB025/ UTPLST (.) — списки типов частиц для разных каналов распада.

/UMB026/ NMATRX. NMBUF1, QMATRX(13,NMATRX) — преобразования координат из общей системы в локальную.

NMATRX — количество локальных систем координат,

NMBUF1 — не используется,

QMATRX(1,.) — имя локальной системы,

QMATRX(2-4,.) — вектор смещения (см),

QMATRX(5-13,.) — матрица поворота (по столбцам). Если вектор в общей системе $RO(i)$, то вектор в локальной системе $R(j)$ находится суммированием по k величины $QMATRX(3 \cdot j+k+1,.) \cdot (RO(k)-QMATRX(k+1,.)$. Матрица поворота всегда ортогональная, поэтому обратное преобразование соответствует транспонированной матрице. Вектор $RO(i)$ в общей системе находится суммированием по k величины $QMATRX(3 \cdot k+i+1,.) \cdot R(k)$ и добавлением $QMATRX(i+1,.)$.

/UMB027/ QVECTO(6,.) — вектора положения частицы и направления скорости (попарно) для всех систем координат. Применено или нет преобразование к данной системе — указывается в общем блоке /UMB028/.

/UMB028/ USIGNM(.) — указатели, применено(=1) или нет (=0) преобразование к данной системе координат векторов QR и QV. Если преобразование к текущим координатам и скорости частицы применено, то в /UMB027/ находятся их величины в соответствующей локальной системе координат.

/UMB029/ UINTR, UFALS, UNMBR, UREGPR, UBUF1, UMATIN, PTOT, PINTR(UINTR), ADECA

UINTR — количество "точечных" взаимодействий,

UFALS — признак "ложного" взаимодействия, который программа генерации продуктов реакции может выставить. Частица продолжает движение,

UNMBR — номер точечного взаимодействия, которому выпало произойти (в блоке /UMB030/),

UREGPR — режим работы программы взаимодействия (1 — следует вычислять коэффициент поглощения, 2 — генерировать продукты взаимодействия),

UBUF1 — резерв,

UMATIN — номер вещества, для которого ведётся расчёт коэффицента поглощения или генерация продуктов взаимодействия. Используется программами обслуживания взаимодействий без анализа, простое это вещество или компонент сложного,

PTOT — сумма коэффициентов поглощения PINTR,

PINTR — коэффициенты поглощения для каждого типа взаимодействия (в ед. 1/см),

ADECA — коэффициент поглощения (1/см) за счёт распада.

/UMB030/ UNAMIN(3,UINTR) — имена программ моделирования процессов взаимодействия.

/UMB031/ ULSINT(NTYPAR, UINTR)

ULSINT(j,i) — указывает для i-ого взаимодействия и частицы типа j, подвержена ли эта частица данному взаимодействию (=1) или нет (=0).

/UMB032/ UMATLS(.) — списки номеров веществ, вложенных в блоки, для которых возможна генерация пробега типа 2.

/UMB033/ NCUTS, CUTS(10,NCUTS) — управляющие числа для развития каскада

DIMENSION LCUTS(10,.)

EQUIVALENCE (CUTS(1,1), LCUTS(1,1))

NCUTS — количество комплектов чисел,

CUTS(1,.) — имя комплекта чисел,

CUTS(2,.) — максимальный шаг частиц, на которые влияет магнитное поле,

CUTS(3,.) — максимальный шаг заряженных частиц в веществе в г/$см^2$,

CUTS(4,.) — максимально допустимое изменение энергии частицы на пробеге до точки взаимодействия (МэВ),

CUTS(5,.) — максимальная доля энергии, которую может потерять частица на элементарном перемещении до точки взаимодействия (величина от 0 до 1),

CUTS(6,.) — пороговая энергия γ-квантов (МэВ) в процессе тормозного излучения,

CUTS(7,.) — пороговая кинетическая энергия электронов в процессе рассеяния на электронах атомов,

CUTS(8,.) — ограничение на средний угол многократного рассеяния (градусы),

LCUTS(9,.) — количество пороговых кинетических энергий для разных частиц. При понижении энергии частицы ниже пороговой она выбывает из каскада.

LCUTS(10,.) — индекс описания первой пороговой энергии в /UMB060/. Частица с энергией ниже пороговой не рассматривается. Если частица заряженная или фотон, то её кинетическая энергия добавляется в энергию возбуждения среды EXCIT в /UMB014/ и далее может использоваться в регистрирующих приборах. Энергия заряженной частицы уменьшается до энергии покоя. Если для данного типа частиц имеется программа моделирования взаимодействия в остановке, то управление передаётся ей, в противном случае для нестабильных частиц вызывается программа моделирования распада.

Примечание. Из двух ограничений на изменение энергии частицы на каждом шаге "работает" более слабое — то, которое позволяет сделать бóльшее перемещение. То же самое касается пары ограничений на максимальный шаг заряженных частиц и на угол многократного рассеяния.

/UMB034/ BFIELD, B0V0, QFAZE, RFIELD(3), RV0XB(3), RBLOC(6,.)

BFIELD — абсолютная величина магнитного поля (кГс),

B0V0 — косинус угла между полем и скоростью частицы,

QFAZE — множитель перед пройденным расстоянием для получения приращения фазы винтовой линии,

RFIELD — единичный вектор направления магнитного поля,

RV0XB — вектор, равный векторному произведению единичного вектора скорости частицы на единичный вектор RFIELD,

RBLOC — вектора RFIELD и RV0XB, преобразованные в локальные системы координат. Проведены эти преобразования или нет, можно узнать по указателям в USIGNM в /UMB028/.

/UMB035/ NMAMAX, NMUSED, ALFI(UINTR,.)

NMAMAX — максимальное количество веществ в списке "вложенных" веществ,

NMUSED — количество веществ, для которых в ALFI есть вычисленные коэффициенты поглощения,

ALFI — коэффициенты поглощения для каждого процесса и каждого компонента вещества.

/UMB036/ NMATM, ALFASM, ALFIM(UINTR,.)

NMATM — номер вещества, коэффициент поглощения которого используется для розыгрыша пробега,

ALFASM — суммарный коэффициент поглощения, который был использован для розыгрыша пробега,

ALFIM — парциальные коэффициенты поглощения по процессам и разным веществам (необходимы при розыгрыше пробега типа 2).

/UMB037/ NEVENT, NCHARG, NNEUT, NPTOT, NTSTAR, NTIMEV, QRANDS, QPULST(3), QE2TOT, JSURV

NEVENT — номер события,

NCHARG — количество заряженных частиц в событии,

NNEUT — количество нейтральных частиц в событии,

NPTOT — полное количество частиц в событии,

NTSTAR — остаток времени счёта в момент, когда событие стартовало (мсек),

NTIMEV — полное время CPU на это событие (мсек),

QRANDS — случайное число, с которого стартовало событие,

QPULST — вектор импульса системы начальных электрона и позитрона (МэВ/с),

QE2TOT — полная энергия системы начальных электрона и позитрона (МэВ),

JSURV — номер просмотра буфера частиц (1 или 2).

/UMB038/ SLUM, QSRAND, RINTRG(3), SINTRG(3), QNAMCT, VTAPE(3), VTAPE2(3), NRUN, ENERGY, E1SPR, IREND, NEVMAX, NTSAVE, TIMAX, NTIMAX, NSLONG, NRANLG(2), NTISUM, NPARMX, NPARSM, JSINT, INTOLD, LPRINT

SLUM — суммарное значение величины, используемой для вычисления интеграла сечения,

QSRAND — стартовое случайное число для всего счёта,

RINTRG — координаты центра области взаимодействия (см),

SINTRG — размеры центра области взаимодействия (ср.-кв.,см),

QNAMCT — имя каталога файлов на магнитных лентах для записи результатов (сейчас не используется),

VTAPE1 — логическое имя магнитной ленты (сейчас не используется),

VTAPE2 — логическое имя запасной магнитной ленты (сейчас не используется),

NRUN — цифровая часть имени файла с результатами моделирования для записи на маг. ленту. Если имеет нулевое значение — запись не заказана (сейчас не используется),

ENERGY — энергия одного пучка (МэВ),

E1SPR — разброс энергий в одном пучке (МэВ),

IREND — необязательное средство для остановки программы. При единичном значении этого флага счёт заканчивается. Нормальное значение переменной IREND — нуль,

NEVMAX — максимальное потребное количество событий. По достижении этой статистики счёт прекращается,

NTSAVE — запас по времени для записи результатов на маг. ленту и распечатки (сек. процессорного времени). Генерация нового события не начинается, если оставшееся время до конца счёта меньше суммы максимального времени на событие, зарегистрированного на предыдущих событиях, и NTSAVE,

TIMAX — ограничение по времени в нсек на время чувствительности детектора от начала события,

NTIMAX — время генерации в мсек самого долгосчётного события,

NSLONG — номер самого долгосчётного события,

NRANLG — случайное число, с которого начался розыгрыш самого долгосчётного события,

NTISUM — суммарное время в мсек, затраченное на генерацию всех событий,

NPARMX — максимальное количество частиц, достигнутое в каком-либо событии,

NPARSM — суммарное количество всех частиц во всех событиях,

JSINT — величина интервала в секундах процессорного времени между выходами на точку программы NEXTEN. Если JSINT≤ 0, то выхода на точку NEXTEN не делается,

INTOLD — остаток времени счёта в момент последнего выхода на точку NEXTEN (сек),

LPRINT — целое число 0, 1, 2, . . . — уровень печати о развитии события.

/UMB039/ NDISCR, UDISCR(10,NDISCR)

DIMENSION PDISCR (5,.)

EQUIVALENCE (UDISCR(1,1), PDISCR(1,1))

NDISCR — количество дискриминаторов,

UDISCR(1-2,.) — собственное имя дискриминатора,

UDISCR(3,.) — выход дискриминатора (0 или 1),

UDISCR(4,.) — тип дискриминатора, целое число. Является одновременно номером программы в /UMB040/, обслуживающей этот тип дискриминаторов,

UDISCR(5,.) — количество входных амплитуд,

UDISCR(6,.) — индекс описания первой амплитуды в /UMB041/,

PDISCR(4-5,.) — два порога дискриминатора. Использование зависит от программы, моделирующей дискриминатор.

/UMB040/ NPROGD, UNAMDS(4, NPROGD)

NPROGD — количество типов дискриминаторов,

UNAMDS(1-3,.) — название программы,

UNAMDS(4,.) — целое число, количество входов в дискриминатор (0, 1, 2, . . .). Если это число = -1, то программа может работать с любым числом входных импульсов.

/UMB041/ ULISD(.) — списки номеров амплитуд в /UMB020/ для дискриминаторов.

/UMB042/ NLOG, ULOGS(6,NLOG)

NLOG — количество логических блоков быстрой электроники,

ULOGS(1-2,.) — название блока,

ULOGS(3,.) — тип блока (номер программы),

ULOGS(4,.) — выход блока. Если блок сработал, то равен единице, в противном случае — нуль,

ULOGS(5-6,.) — количество входов и индекс первого входа в ULISBL в /UMB044/.

/UMB044/ ULISBL(2,.) — списки входов для логических блоков электроники. Первое число — признак общего блока: 1 — общий блок дискриминаторов /UMB039/, 2 — общий блок логической электроники /UMB042/. Второе число — индекс в этом общем блоке.

/UMB045/ NPGIST, VNAMPR(7,NPGIST)

NPGIST — количество программ вычисления параметров,

VNAMPR(1-3,.) — название программы,

VNAMPR(4,.) — количество входных параметров,

VNAMPR(5,.) — количество выходных параметров,

VNAMPR(6,.) — индекс первого номера входного параметра в ULPIN в /UMB047/,

VNAMPR(7,.) — индекс первого вых. параметра в /GIST18/.

/UMB048/ UGEOM, UNAMG(4,UGEOM)

UGEOM — количество геометрических программ,

UNAMG(1-3,.) — название программы,

UNAMG(4,.) — количество параметров, описывающих геометрическую структуру, или число меньше нуля, если количество параметров может быть переменным и интерпретатор ввода не должен контролировать количество параметров.

/UMB049/ NCHPR, UNCHPR(4, NCHPR)

NCHPR — количество программ моделирования проволочных камер,

UNCHPR(1-3,.) — название программы,

UNCHPR(4,.) — количество параметров, характеризующих данный тип камер, или число меньше нуля, если это количество может быть переменным.

/UMB050/ USION, USMUL, UNION(3), UNMUL(3)

USION =0 нет ионизационных потерь
=1 ионизационные потери моделируются
USMUL =0 нет многократного рассеяния
=1 многократное рассеяние моделируется
UNION — название программы ионизационных потерь
UNMUL— название программы многократного рассеяния

/UMB055/ NLENTH(120) — размеры общих блоков в байтах.

/UMB057/ UINTAB(6,.)

```
DIMENSION ALFASK(3,.)
EQUIVALENCE (UINTAB(1,1),ALFASK(1,1))
```

UINTAB(1,.) — индекс в ALFI (/UMB035/) начала массива парциальных коэффициентов поглощения,

UINTAB(2,.) — количество компонент в UMB035 для этого вещества,

UINTAB(3,.) — номер вещества, которому они соответствуют,

UINTAB(4,.) — номер первой компоненты вещества в /UMB003/,

ALFASK(3,.) — суммарный коэффициент поглощения по компонентам и процессам.

/UMB058/ NGATE, GATCH(3,NGATE)

NGATE — количество временны́х ворот,

GATCH(1,.) — имя ворот,

GATCH(2-3,.) — временны́е ворота в нсек (интервал).

/UMB059/ NDPMAT, DOPMAT(.)

NDPMAT — количество вспомогательных величин на каждое вещество (расширение /UMB003/).

Если обозначить за NSH = (UMAT-1)*NDPMAT, то:

```
DOPMAT(NSH+1)= AMAT(4,UMAT)*AMAT(2,UMAT)/AMAT(3,UMAT)
DOPMAT(NSH+2)= 0.3066 * DOPMAT(NSH+1)
DOPMAT(NSH+3)= (11.9185 + 16.7*AMAT(5,UMAT)) * AMAT(5,UMAT)**2
DOPMAT(NSH+4)= 0.978467E-3 * AMAT(5,UMAT) * EXP(0.666667 +
DOPMAT(NSH+3) / AMAT(2,UMAT))
DOPMAT(NSH+5)= 0.666667 / DOPMAT(NSH+4)**1.5
DOPMAT(NSH+6)= AMAT(2,UMAT)**2 * AMAT(4,UMAT)* (5. +
AMAT(2,UMAT)**2) / AMAT(3,UMAT) / (1. + AMAT(2,UMAT)**2)
DOPMAT(NSH+7)= 0.978467E-3 * AMAT(5,UMAT)
```

/UMB060/ CUTKIN(2,.) — списки пороговых энергий

```
DIMENSION NPCUT(2,.)
EQUIVALENCE (CUTKIN(1,1),NPCUT(1,1))
```

NPCUT(1,.) — номер типа частицы в /UMB023/,

CUTKIN(2,.) — пороговая кинетическая энергия (МэВ).

/UMB061/ ALLCUT(NTYPAR,NCUTS)

ALLCUT(i,j) — пороговая кинетическая энергия (МэВ) для i-ой частицы в j-ом комплекте управляющих чисел каскада (см. /UMB033/)

/UMB062/ NDECPR, UDECDS(5, NDECPR) — список программ распадов

NDECPR — количество программ во всем списке,

UDECDS(1-3,.) — имя программы,

UDECDS(4,.) — номер типа нестабильной частицы,

UDECDS(5,.) — номер её канала распада

/UMB068/ NAMS, COM, VER, FMAG — комментарии к программе первичного моделирования (формируются самой программой при первом обращении к ней)

character *4 NAMS — символическое название процесса,

character *15 COM — уточняющие параметры в текстовом виде,

character *5 VER — обозначение версии,

real *4 FMAG — отношение значения функции плотности вероятности к мажоранте, если для генерации используется метод Неймана.

В виде исключения из ранее декларированных правил именования программ и общих блоков целая серия программ и общих блоков включена со своими прежними именами — некоторая выборка из существующего гистограммного комплекса. Ниже перечисляются общие блоки, представляющие интерес для пользователей программы UNIMOD2.

/GIS001/ NUSL, KUSL(5, NUSL) — условия на параметры.

NUSL — количество условий,

KUSL(1,.) — имя условия,

KUSL(2,.) — номер параметра в блоке /GIS002/,

KUSL(3,.) — первая граница,

KUSL(4,.) — вторая граница,

KUSL(5,.) — результат проверки условия для очередного события (0 или 1).

/GIS002/ NPAR, NAMPAR(NPAR) — имена параметров.

/GIS003/ FORM(.) — тексты логических формул. Следуют непрерывно друг за другом.

CHARACTER *1 FORM

На шаге ввода тексты формул не меняются, а на шаге исполнения имена условий заменены на номера условий в /GIS001/, имена формул заменены на «10000 + номер формулы» в /GIS004/.

/GIS004/ NFORM, NFORML(4, NFORM).

NFORM — количество формул,

NFORML(1,.) — имя формулы,

NFORML(2,.) — индекс первого символа формулы в /GIS003/ FORM

NFORML(3,.) — количество байтов в формуле в FORM

NFORML(4,.) — результат вычисления формулы (0 или 1)

/GIS009/ IGPAR(.) — список номеров параметров, которые используются для занесения в гистограммы.

/GIS010/ NGIS0, NGIST0(6, NGIS0)

NGIS0 — количество нулевых гистограмм,

NGIST0(1-2,.) — имя гистограммы или пробелы,

NGIST0(3-4,.) — не используется (занесено $\texttt{FEDCBA9876543210}_{16}$),

NGIST0(5,.) — номер условия отбора событий. Является номером условия на параметр, если не превышает 10000, в противном случае является «номером формулы + 10000». Если 0, то условие не задано.

NGIST0(6,.) — характеристика списка параметров гистограммы. В первом полуслове — начальный индекс списка номеров в /GIS009/, второе полуслово — количество номеров в списке.

/GIS011/ NGIS1, NGIST1(8,NGIS1)

Первые 6 описателей гистограммы на 30 каналов такие же, как в предыдущем блоке.

NGIST1(7,.) — центр гистограммы,

NGIST1(8,.) — шаг гистограммы.

/GIS012/ NGIS2, NGIST2(8, NGIS2) — описание одномерных гистограмм с числом каналов 80. Структура описания такая же, как для 30-канальных.

/GIS013/ NGIS3,NGIST3(11,.) — описание двумерных гистограмм 24 × 24.

Первые 8 слов описания имеют такой же смысл, как в описании одномерных гистограмм, включая “параметр-центр-шаг” для первой оси. NGIST(9-11,.) описывают “параметр-центр-шаг” по второй оси.

/GIS014/ NBUHG0(6,.) — бухгалтерия по нулевым гистограммам.

DIMENSION QBUHG0(3,.)

EQUIVALENCE (NBUHG0(1,1), QBUHG0(1,1))

NBUHG0(1,.) — количество событий, попавших в гистограмму,

NBUHG0(2,.) — количество забракованных событий,

QBUHG0(2,.) — сумма значений параметров,

QBUHG0(3,.) — сумма квадратов значений параметров.

/GIS015/ NBUHG1(36,.) — бухгалтерия по 30-канальным гистограммам + 30 накопителей.

/GIS016/ NBUHG2(86,.) — бухгалтерия по 80-канальным гистограммам + 80 накопителей.

/GIS017/ NBUHG3(586,.) — бухгалтерия по двумерным гистограммам + 24 × 24 = 576 накопителей.

/GIST18/ PAR(.) — значения параметров.

7 Выходные файлы и номера логических устройств

В процессе работы программы создаётся несколько файлов в рабочей директории. Бóльшая часть этих файлов — временные, такие файлы по окончании работы процедуры удаляются, если нет параметра процедуры, который управляет процессом удаления. Приведём список файлов, создаваемых программой (в скобках указывается имя этого файла в системе UNIX, если оно отлично от имени в VAX/VMS):

UM1000.FOR — текст головной программы для цикла генерации событий (UM1000.f).

file2.**LIS** — файл результатов. В этот файл выводится стандартная выходная информация: протокол ввода, сообщения об ошибках, гистограммы распределений, статистика работы программы.

BLOK_NUM.WORK — файл для записи информации о номере блока вещества, в котором родилась частица.

COMMONS.WORK — в этот файл выводится содержимое общих блоков, заполненных на шаге ввода и обработки заказа, для последующего ввода в одноимённые общие блоки на шаге счёта.

HISTORY.WORK — записывается информация о частицах в каскаде, если число частиц в событии превышает 128.

MYPROG.FOR — промежуточный файл с текстами личных программ, введённых в разделе входного потока **PRIVATE. В конце шага обработки заказа этот файл целиком включается в файл головной программы счёта.

В целях упрощения переноса программы на другие ЭВМ было принято решение зафиксировать логические номера устройств ввода-вывода для стандартных программ, а не пользоваться возможностью в VAX/VMS запроса незанятого логического устройства. Программы пользователя для своих операций ввода-вывода должны использовать незанятые номера логических устройств. Приводим полный список стандартных логических номеров устройств ввода-вывода с краткими комментариями:

9 — ввод-вывод блоков истории события в файл HISTORY.WORK.

11 — ввод файла REACCHAN.DAT.

12 — ввод файла PARTICLE.DAT на шаге счёта; ввод файла DIRLIST при обработке заказа.

13 — ввод файла DIRLIST в программе UM1097.

14 — ввод-вывод информации о номере блока вещества, в котором родилась частица (если количество частиц больше 128).

20 — ввод-вывод текстов личных программ.

21 — печать текстов личных программ.

22 — программа чтения файлов UM1077.

23 — чтение характеристик частицы из файла истории события.

43 — запись-чтение содержимого общих блоков.

45, 46, 47 — входной поток для файлов трёх уровней вложенности.

48 — вспомогательный канал ввода в программе обработки заказа.

49 — вывод текста программы UM1000.FOR.

50 — вывод гистограмм в формате PAW.

8 Правила написания подпрограмм

8.1 Общие замечания

Особенности организации программы UNIMOD2 заключаются в том, что стандартные программы моделирования процесов взаимодействия, распадов, анализа взаимного расположения частицы и геометрических фигур и т.д. написаны по определённым правилам, единым для каждой группы программ. При этом список программ, которые вызываются в данном счёте, создаётся в процессе обработки заказа, поэтому его можно легко модифицировать, дополняя список именами своих программ, написанных по этим же правилам, или исключая из списка стандартные программы по каким-либо соображениям.

Программы следует писать на языках программирования Фортран-77 или других языках с соблюдением фортрановских соглашений о вызовах подпрограмм и передаче параметров.

8.1.1 Типы переменных

В дополнение к фортрановским правилам описания типов переменных по умолчанию ниже подразумевается следующее определение:

IMPLICIT INTEGER *2 (U-W), REAL *8 (Q-S)

Это рекомендуется делать во всех программах для уменьшения вероятности неправильного описания типа переменной из общих блоков программы UNIMOD2, так как во всех инструкциях подразумевается именно такое неявное описание типа.

Общие блоки, необходимые для работы программ, не выписываются в этой главе. Описание общих блоков надо смотреть в разделе 6. Здесь же названия переменных из этих общих блоков используются без пояснений.

8.1.2 Диагностические сообщения

Любая диагностика о ненормальном ходе моделирования должна начинаться с имени подпрограммы, которая её выдает. Если требуется после диагностики аварийно завершить работу, то в фортрановских программах следует пользоваться оператором STOP 8.

Программам, предназначенным для включения в программу UNIMOD2 для общего пользования должны назначаться имена в соответствии со следующим правилом: имя начинается с букв UM, затем следует номер версии (для начального комплекта — 1) и число, указывающее на

принадлежность тому или другому автору (распределение номеров на данный момент см. в разделе 4). Пример имени: UM1024.

Для таких программ не рекомендуется вводить без веских оснований общие блоки — лучше пользоваться параметрами подпрограмм. Если же необходимо ввести общий блок, то название для него следует выбирать UMB*nnn*, где *nnn* трехзначный номер в соответствующих пределах для каждого автора (см. раздел 4).

8.1.3 Программная компонента BLOCK DATA

Необходимо придерживаться следующих правил при написании BLOCK DATA:

- создавать только поименованные программные компоненты BLOCK DATA. Имена лучше образовывать на основе имён программ, в которых используется информация из общих блоков, заполняемых в BLOCK DATA. Например, если для программы UM1017 создаётся программная компонента BLOCK DATA, то её удобно назвать следующим образом:

  BLOCK DATA UM1017_BD

- в подпрограммах, использующих информацию из BLOCK DATA, перечислять в операторе EXTERNAL имена соответствующих программных компонент BLOCK DATA, например:

  subroutine UM1017
  implicit real *8 (Q-S), integer *2 (U-W)
  external UM1017_BD
  ...

8.1.4 Вспомогательные подпрограммы

Для моделирования случайных процессов следует использовать генератор случайных чисел DRNDM и другие генераторы, написанные на его основе: DDCOSI, DDGAUS и т.д.

Для поворота единичного вектора на угол THETA со случайным азимутальным углом можно пользоваться программой UM1026:

DOUBLE PRECISION RUNIT(3), RCT, RST
CALL UM1026(RUNIT,RCT,RST)

Здесь перед обращением к программе должен быть определен вектор RUNIT, который после вызова программы UM1026 окажется повернутым. Нормированность вектора существенным образом используется при преобразовании, поэтому проводится проверка неравенства ABS(RUNIT**2-1.) < 1.E-10, и если оно не выполняется, то делается аварийный останов. Угол поворота задается своими косинусом RCT и синусом RST.

Однако, для экономии времени при поворотах одного и того же вектора начальной частицы для получения векторов направления скорости вторичных частиц можно пользоваться следующей возможностью в программе UM1026. Если первая компонента вектора RUNIT(1) на входе в программу UM1026 окажется больше 1.5, то это не рассматривается как нарушение нормировки, но меняет алгоритм вычисления компонент повернутого вектора. Вектор RUNIT, дополнительные к нему орты и азимутальный угол поворота берётся от предыдущего обращения, из параметров данного обращения к программе используется только косинус и синус полярного угла (переменные RCT и RST).

8.2 Генерация параметров первичных частиц

Используемые общие блоки: /UMB013/, /UMB037/, /UMB038/.

Программа первичного моделирования должна поочерёдно моделировать характеристики частиц, рождённых в столкновении электрона и позитрона, заносить их в общий блок /UMB013/ (UTYPF — тип частицы, целое число, ENERGF — полная энергия (МэВ), TIMEF — начальное время события (нсек), QRF(3) — вектор положения частицы в общей системе координат (см), QVF(3) — единичный вектор направления движения частицы, QSPINF(3) — вектор направления поляризации) и записывать в буфер частиц путём обращения к программе UM1024 (программа без параметров).

Перед обращением к программам первичного моделирования в /UMB013/ определены величины TIMEF = 0 — стартовое время события в нсек, QRF — радиус-вектор точки взаимодействия в соответствии с положением центра области взаимодействия встречных пучков и среднеквадратичными размерами по трём осям (см), QSPINF(1) — первая компонента вектора поляризации положена равной 2, что соответствует отсутствию поляризации. Вообще говоря, программы первичного моделирования могут менять эти величины по своему усмотрению, однако, это должно быть оговорено в описании этой программы для общего пользования.

Возможным источником информации для программ первичного моделирования являются переменные: RINTRG(3) — координаты центра области взаимодействия, SINTRG(3) — среднеквадратичные размеры области взаимодействия по каждой оси, ENERGY — энергия одного пучка и E1SPR разброс энергий в одном пучке, МэВ. Эти переменные определяются данными во входном потоке и их не следует менять.

При первом обращении к программе первичного моделирования она должна выдать сообщение, содержащее название этой программы, имя автора, дату фиксации версии и, желательно, краткую характеристику процесса. Это необходимо для повышения надёжности моделирования, т.к. головная программа никак не контролирует работу программ первичного моделирования.

Не накладывается ограничений на вызов нескольких программ первичного моделирования, этим можно разумно пользоваться.

В общем блоке /UMB037/ имеются переменные QE2TOT и QPULST, которые формируются головной программой на основе данных о разбросе энергий в каждом пучке и имеют смысл, соответственно, полной энергии и вектора полного импульса (МэВ) системы начального электрона и позитрона (в предположении, что пучки движутся в направлении оси X). Эти величины могут использоваться или игнорироваться программами первичного моделирования, а также модифицироваться для использования последующими программами первичного моделирования. Например, под видом программ первичного моделирования можно включать программы, модифицирующие энергию и импульс системы для других программ первичного моделирования. Так можно моделировать радиационные поправки в некоторых случаях.

На программы первичного моделирования возлагается работа по формированию сечения регистрации. В переменной SLUM надо накапливать такую сумму, чтобы отношение числа зарегистрированных событий к SLUM давало сечение регистрации этих событий в нанобарнах. Если известен интеграл сечения по фазовому объёму SIGMA, то на каждом событии программа первичного моделирования должна в SLUM добавлять 1./SIGMA. Если программа первичного моделирования не производит такого суммирования, то сечение регистрации придётся определять вне рамок программы UNIMOD.

8.3 Программы моделирования процессов взаимодействия частиц с веществом

В программе используются общие блоки: /UMB029/, /UMB003/, /UMB012/, /UMB013/, /UMB014/. Переменная UMATIN в /UMB029/ — номер вещества, в котором движется частица. Вакууму соответствует UMATIN=0. Программа моделирования взаимодействия не должна анализировать, простое это вещество или сложное. Для неё все вещества — простые. Многокомпонентность веществ учитывается головной программой.

Необходимо реализовать два режима работы программы в зависимости от значения переменной UREGPR:

UREGPR=1 программа моделирования процесса должна вычислить коэффициент поглощения частицы в указанном веществе в единицах 1/см и занести его в переменную PINTR(UNMBR). Программа может использовать метод выравнивания сечения и "сообщать" головной программе вместо истинного коэффициента поглощения завышенный коэффициент, соответствующий сумме истинного и "ложного" взаимодействия. Затем при генерации продуктов взаимодействия иногда (с соответствующей вероятностью) надо отказываться от моделирования взаимодействия (см. ниже). Такой прием рекомендуется использовать, когда сечение взаимодействия известно в дифференциальном виде и не интегрируется в элементарных функциях или сечение зависит от меняющейся энергии (для заряженной частицы). В последнем случае можно пользоваться информацией из /UMB012/ и /UMB014/: ENERG — полная энергия частицы в начале очередного перемещения (МэВ), DENRS — максимально возможное уменьшение энергии (МэВ) на очередном перемещении.

UREGPR=2 программа должна моделировать продукты взаимодействия. Со времени последнего обращения к программе с UREGPR=1 могут измениться такие характеристики, как полная энергия ENERG, номер вещества UMATIN, коэффициент поглощения PINTR(UNMBR).

Если программа приняла решение, что произошло "ложное" взаимодействие, то она должна "выставить флаг" UFALS=1 и вернуть управление головной программе. Для правильного принятия решения необходимо использовать также значение PINTR(UNMBR).

Если же принято решение, что взаимодействие действительно произошло, то моделировать его необходимо для текущих параметров частицы и вещества из /UMB012/.

Если в результате взаимодействия возникает энергия возбуждения среды, которая может регистрироваться, то её следует добавить в переменную EXCIT. Характеристики продуктов реакции: UTYPF — номер типа частицы, ENERGF — полная энергия в МэВ, QVF(3) — единичный вектор направления движения, QSPINF(3) — вектор поляризации частицы, формируются по очереди для всех продуктов реакции в общем блоке /UMB013/ с записью характеристик каждой частицы в буфер ожидания путем обращения к программе UM1024 (программа без параметров). В настоящее время движение частиц со спином не моделируется, поэтому следует полагать QSPINF(1)=2.

8.4 Распады нестабильных частиц

На каждый канал распада каждой нестабильной частицы регистрируется имя программы, моделирующей этот распад. Программа должна иметь столько параметров, сколько зарегистрировано частиц — продуктов распада. Обращение из головной программы производится в следующей форме:

CALL XXXXXX(IT1,IT2,IT3,...)

где IT1, IT2, ...— номера типов частиц-продуктов распада в том порядке, в каком они зарегистрированы в описании частиц.

При работе программ следует использовать следующие блоки: /UMB012/, /UMB013/, /UMB023/.

Для каждой частицы, получающейся при распаде, надо в общем блоке /UMB013/ сформировать переменные UTYPF (тип частицы), ENERGF (полная энергия в МэВ), QVF — единичный вектор направления движения. Поляризация не моделируется, поэтому QSPINF(1) = 2. После этого обращением к программе UM1024 (без параметров) записать характеристики новой частицы в буфер ожидания и перейти к моделированию следующей частицы.

Преобразование Лоренца можно осуществлять программой

UM1064(QV, GAM, QVF, ENERGF, EMAS)

где QV — единичный вектор направления движения распадающейся частицы, GAM — её Лоренц-фактор, EMAS — масса новой частицы (МэВ). Параметры QVF и ENERGF перед обращением должны содер-

жать единичный вектор направления движения и полную энергию в МэВ новой частицы в системе покоя распадающейся. После обращения в этих параметрах будут аналогичные величины в лабораторной системе.

8.5 Программы моделирования геометрических блоков

Программа должна иметь следующий список параметров: (UGPAR, QPAR, QR0, QV0, RMAX, RMIN, BFIELD, B0V0, QFAZE, RBUNIT, RV0B, JR, R), где массивы имеют следующую размерность:

DIMENSION QPAR(.), QR0(3), QV0(3), RBUNIT(3), RV0B(3)

Все параметры являются входными, кроме JR и R.

Транслятор может не принимать запись QPAR(UGPAR), поэтому в программе можно писать реальную размерность массива, а UGPAR можно использовать для контроля, что массив описания сформирован правильно.

Такие параметры частицы, как тип, энергия, заряд, масса, следует выбирать из общих блоков /UMB012/ и /UMB023/ (см. раздел 6).

Входные параметры:

QPAR(.) — массив геометрических параметров блока (в программе в операторе DIMENSION надо подставлять реальную размерность),

QR0(3) — вектор положения частицы в локальной системе координат,

QV0(3) — единичный вектор направления движения частицы в локальной системе координат блока,

RMAX — максимальное расстояние вдоль траектории, на которое следует анализировать пересечение с поверхностью блока,

RMIN — минимальное расстояние вдоль траектории, которое считается отличным от нуля. Геометрическая программа никогда не должна выдавать в качестве результата расстояние, меньшее RMIN. Если расстояние до какой-то грани меньше RMIN, надо считать, что эту грань уже прошли.

BFIELD — абсолютное значение индукции магнитного поля (кГс) в точке, где находится частица. Если BFIELD=0 или заряд=0, то остальные входные параметры не определены.

B0V0 — косинус угла между напряженностью поля и скоростью частицы,

QFAZE — множитель перед пройденным расстоянием для получения приращения фазы винтовой линии,

RBUNIT(3) — единичный вектор направления магнитного поля в локальной системе координат,

RV0B(3) — вектор, равный векторному произведению вектора направления скорости на единичный вектор вдоль вектора магнитной индукции (всё в локальной системе координат).

Параметр JR является и входным, и выходным. Параметр R — расстояние вдоль траектории до ближайшей грани блока, — является выходным.

Режим работы программы	JR	
	Вход	Выход
Определить, частица внутри или снаружи блока. R вычислять не требуется, параметры магнитного поля можно не использовать.	0	0 — снаружи 1 — внутри
Вычислить расстояние до ближайшей грани (предполагая, что частица внутри блока).	1	0 — ошибка, частица не внутри 1 — вычислено R 2 — R не вычислено, R≥RMAX
Вычислить расстояние до ближайшей грани (предполагая, что частица снаружи блока).	2	0 — ошибка, частица не снаружи 1 — вычислено R 2 — R не вычислено, R≥RMAX или нет пересечения

В тех случаях, когда расстояние до грани превышает RMAX, то его можно не вычислять точно и R не определять.

Если программа обнаружит, что статус частицы ошибочный, т.е. головная программа "думает", что частица внутри, а она — снаружи блока, или наоборот, то следует выдавать сообщение с подробной информацией для анализа ситуации.

Предполагается, что все геометрические программы могут работать с магнитным полем. По крайней мере, все должны проверять BFIELD и если есть поле, а программа не рассчитана на магнитный вариант, то выдавать сообщение и аварийно останавливаться (STOP 8).

8.6 Моделирование ионизационных потерь

Эта программа должна учитывать любые непрерывные потери энергии, то есть те потери, при которых не моделируется рождение вторичных частиц.

Программа, генерирующая ионизационные потери энергии на участке пути с учётом флуктуаций, используется для всех заряженных частиц. Вызов:

CALL XXXXXX(E, PM, CH, EP, D, RX, MAT, *)

E — начальная энергия летящей частицы в МэВ,

PM — масса частицы в МэВ/c^2,

CH — заряд частицы в единицах заряда позитрона,

EP — порог рождения δ-электронов в МэВ,

D — выходной параметр, значение потерянной энергии (МэВ) на перемещении RX (см); в случае недостаточной энергии для прохождения расстояния RX — выход по метке (RETURN 1) и в D должно быть расстояние до точки остановки,

RX — перемещение,

MAT — номер вещества, в котором находится частица,

* — выход по метке, если частица не может пройти заданное перемещение RX; в этом случае выдаётся расстояние до точки остановки.

8.7 Генерация угла многократного рассеяния

Вызов:

CALL XXXXXX(E, PM, CH, RX, MAT, TETA)

где

E — энергия летящей частицы в МэВ,

PM — масса частицы в МэВ/c^2,

CH — заряд частицы в единицах заряда позитрона,

RX — перемещение в см,

MAT — номер вещества, в котором находится частица,

TETA — угол поворота вектора скорости от первоначального направления (радиан).

Первые пять аргументов — входные, последний аргумент — выходной.

8.8 Вычисление магнитного поля

Эти программы необходимо подготавливать пользователям, если поле не однородное и не представляется кубическим сплайном.

Обращение к программе:

CALL XXXXXX(BFIL, FSP, NP)

где

BFIL(3) — компоненты поля (кГс), которые программа должна вычислить (без учёта общего множителя ко всем полям в /UMB010/ FIELD0),

FSP(NP) — параметры поля (задаются в исходных данных при запуске программы UNIMOD2).

Координаты точки, в которой надо вычислить магнитное поле (радиус-вектор частицы), находятся в общем блоке /UMB012/..., QR(3).

8.9 Моделирование проволочной камеры

Вызов программы:

CALL XXXXXX(IFL, ICH, QCH, UCH, QR, QV, QDL, BFIELD, B0V0, QFAZE, RFIELD, RV0XB, NWIRE) где

IFL — управляющая переменная. Перед обращением к программе моделирования камеры эта переменная содержит нуль. Если пересечения траектории с камерой нет, то эту переменную следует оставить без изменения. В зависимости от ситуации можно выдать в этой переменной другие два значения: 1 — есть номер проволочки, и на этом перемещении другие проволочки в этой камере не сработали, 2 — есть номер проволочки, но требуется ещё обратиться к этой же программе с теми же параметрами и она выдаст ещё номер сработавшей проволочки. В последнем случае повторное обращение производится с IFL=2 и не предполагается, что сработавших проволочек всего две — цикл обращений может продолжиться под управлением IFL по тем же правилам:

0 — нет проволочки

1 — есть одна проволочка и больше не будет

2 — есть проволочка и надо ещё обратиться к программе за номером следующей сработавшей проволочки на этом же элементарном перемещении.

ICH — номер камеры (использовать необязательно).

QCH(UCH) — описание камеры (в подпрограмме следует писать реальную размерность массива, например, QCH(10), если камера описывается 10 параметрами),

Далее все вектора заданы в локальной системе координат.

QR — вектор положения частицы.

QV — вектор направления скорости частицы (единичный).

QDL — величина перемещения частицы (см).

BFIELD — абсолютное значение индукции магнитного поля (кГс) в точке, где находится частица. Если BFIELD=0 или заряд частицы равен нулю, то остальные входные параметры не определены.

B0V0 — косинус угла между напряженностью поля и скоростью частицы.

QFAZE — множитель перед пройденным расстоянием для получения приращения фазы винтовой линии.

RFIELD(3) — единичный вектор направления магнитного поля.

RV0XB(3) — вектор, равный векторному произведению вектора направления скорости на единичный вектор вдоль вектора магнитной индукции.

NWIRE — порядковый номер в камере (1, 2,...) сработавшей проволочки.

Информация о положении частицы в /UMB012/UTYP,..., QR(3), QV(3),..., BET, GAM; о точке, в которую перемещается частица, в /UMB013/..., QRF(3), QVF(3),..., BETF, GAMF и о величине перемещения в /UMB014/QDL.

8.10 Генерация амплитуды с проволочки

Программа моделирования амплитуды с проволочки может генерировать сразу несколько аналоговых характеристик срабатывания проволочки, причём эти характеристики не обязательно описывают величину сигнала, а например, время срабатывания, координату вдоль проволочки и т.д. Весь этот комплект аналоговых сигналов называется здесь “амплитудой”. Перед началом события все амплитуды зануляются и далее программы генерации амплитуд должны модифицировать эти величины при каждом срабатывании данной проволочки от последовательно пролетающих около неё частиц.

Хронологический порядок моделирования частиц не гарантируется головной программой, но каждая частица в блоке /UMB012/ в переменной TIME характеризуется временем (нсек) от начала события, что можно использовать при моделировании амплитуды.

Как правило, алгоритм генерации амплитуды тесно связан с алгоритмом определения пересечения траектории с камерой (программа моделирования камеры). Поэтому во многих случаях представляется удобным писать единую программу для моделирования пересечения траектории с камерой и амплитуды, обеспечивая второй вход в программу оператором ENTRY, или вводить дополнительные общие блоки для связи между этими двумя подпрограммами.

Обращение к программе:

CALL XXXXXX(AMP, NA, ICH, NWIRE)

где

AMP(NA) — комплект аналоговых величин, входящих в одну “амплитуду”,

ICH — номер камеры (использовать не обязательно),

NWIRE — номер сработавшей проволочки в ней, выданный программой моделирования камеры.

Характеристики частицы в начале и конце перемещения следует брать из общих блоков /UMB012/ и /UMB013/. Описание камеры, если требуется, находится в /UMB004/.

8.11 Генерация отклика детектора в счётчике

Вызов программы:

CALL XXXXXX(AMP, UNAM)

где AMP(UNAM) — массив аналоговых величин, составляющих один амплитудный комплект.

Перед обращением элементы массива определены: в начале события весь массив зануляется, затем при каждом обращении к программе генерации амплитуды она должна модифицировать элементы массива, используя их прежние значения. При модификации можно использовать характеристики очередного перемещения частицы в /UMB012/, /UMB013/, /UMB014/.

8.12 Программы вычисления параметров события

Все параметры программы представляют собой целые числа — индексы в массиве PAR в общем блоке

/GIST18/ PAR(2)

Индексы формируются головной программой и менять их не следует — испортятся константы в головной программе.

Пример оформления программы OMEG, описанной во входном потоке следующим образом:

PARAMETR: OMEG(DTET, DFI; OM, AM)

Возможный текст программы:

```
      SUBROUTINE OMEG(IN1,IN2,IOUT)
      COMMON/GIST18/PAR(2)
      PAR(IOUT)=SQRT(PAR(IN1)**2 + PAR(IN2)**2)
      PAR(IOUT+1)=ABS(PAR(IN1))
      IF(PAR(IOUT+1).LT.ABS(PAR(IN2)))
     *          PAR(IOUT+1)=ABS(PAR(IN2))
      RETURN
      END
```

Подробное описание гистограммной программы GIST см. в работе [12]. Однако, в программе UNIMOD2 используются не все подпрограммы пакета GIST (нет гистограмм HBOOK и NTUPLES).

8.13 Программы, анализирующие класс события по информации из выделенного блока

Эти программы вызываются в точке NEXTSURV, если во входном потоке указан выделенный блок. Необходимость анализа события в выделенном блоке может возникнуть, если в программе первичного моделирования с большой вероятностью генерируются события с кинематикой,

не представляющей интереса для пользователя. Частицы такого события можно не проводить через детектор. В UNIMOD2 это реализуется благодаря тому, что в первую очередь анализируются частицы, рождённые внутри выделенного блока, затем вызываются подпрограммы в точке NEXTSURV. Эти подпрограммы могут принимать решение — осуществлять ли моделирование дальше или перейти к генерации следующего события.

Во входном потоке данных программу, анализирующую класс события (например, название программы SUBTST) следует описать следующим образом:

```
**ПРОГРАММЫ
NEXTSURV: SUBTST(*200)
```

Сама программа SUBTST на языке FORTRAN-77 должна выглядеть следующим образом:

```
      SUBROUTINE SUBTST(*)
      ...
      RETURN
      ...
      RETURN 1
      END
```

Оператор RETURN соответствует нормальному развитию события, RETURN 1 — переход на начало следующего события.

8.14 Моделирование дискриминаторов амплитуд

Программа моделирования дискриминатора должна иметь следующие параметры:

SUBROUTINE XXXXXX(URES, UINTOT, UINDEX,PTHR1,PTHR2)

и использовать общие блоки:

/UMB020/ AMPVA(2)

/UMB041/ ULISD(2)

UINTOT равно полному количеству входных амплитуд, UINDEX — номер первого индекса входной амплитуды в общем блоке UMB041. Таким образом, входными амплитудами являются:

AMPVA(ULISD(UINDEX)), AMPVA(ULISD(UINDEX + 1)), ...,

AMPVA(ULISD(UINDEX + UINTOT - 1))

Входными параметрами также являются два порога PTHR1 и PTHR2, несмотря на то, что программа моделирования может их не использовать или использовать один порог.

Единственным выходным параметром является URES, который может принимать значения 0 или 1. 1 означает, что дискриминатор сработал.

8.15 Моделирование блоков логической электроники

Программа моделирования логического блока электроники должна иметь следующие параметры:

SUBROUTINE XXXXXX(URES, UINTOT, UINDEX)

и использовать общие блоки:

/UMB039/NDISCR, UDISCR(10,2)

/UMB042/NLOG, ULOGS(6,2)

/UMB044/ULISBL(2,2)

На вход логического блока подаётся UINTOT логических импульсов со значениями 0 или 1. UINDEX — индекс описания первого входа в блоке /UMB044/. Следующие входы описаны сразу же вслед за первым. На примере первого входа покажем, как определяется значение 0 или 1 на входе.

ULISBL(1,UINDEX) = 1. Это означает, что входное значение импульса равно UDISCR(3, ULISBL(2, UINDEX))

ULISBL(1,UINDEX) = 2. Это означает, что входное значение импульса равно ULOGS(4, ULISBL(2, UINDEX))

Выходная переменная URES может принимать значения 0 или 1 в зависимости от значений входных импульсов и функции логического блока.

9 Описание подпрограмм

В этом разделе приведены описания программ, сгруппированных по функциям, которые они выполняют.

9.1 Генерация параметров первичных частиц

Моделирование взаимодействия сталкивающихся во встречных пучках электрона и позитрона является, наряду с моделированием отклика

регистрирующих устройств, наименее стандартизованной частью программы UNIMOD2. Предполагается, что для каждого изучаемого процесса взаимодействия должна быть написана подпрограмма, учитывающая современное состояние теории этого процесса. Тем не менее, имеется несколько стандартных программ, которые можно использовать для проверки правильности моделирования или программ реконструкции событий.

UM1054 — генерируется одна начальная частица с энергией и разбросом энергий в соответствии с исходными данными (одного пучка). Направление скорости фиксировано в положительном направлении оси X, точка вылета генерируется в соответствии с описанием области взаимодействия. Тип частицы задается целым числом в параметре при обращении к программе. Пример описания программы UM1054 во входных числах для генерации начального фотона:

PRIMARY: UM1054(1)

UM1086 — моделирование процесса упругого рассеяния e^+e^- с радиационными поправками [33]. Программа требует задания четырёх параметров:

1. Целое число 1, 2 или 3 — номер оси в основной декартовой системе координат, вдоль которой летят начальные электроны. Если это число задано со знаком минус, то электроны движутся в отрицательном направлении вдоль этой оси.
2. Число от 0.01 до 80 — минимальный угол любой из конечных заряженных частиц с осью пучков в градусах (рассеяние на близкий к нулевому угол не моделируется).
3. Граничная энергия γ-кванта между мягкой и жёсткой областью (МэВ). При меньшей энергии квант моделируется приближенно по формулам сопутствующего излучения, при большей энергии — точно.
4. Уточняющий множитель к мажоранте. Точное значение мажоранты не удалось вычислить, поэтому при каких-то условиях может потребоваться ускорить счёт, уменьшая мажоранту, в других случаях может быть необходимым повысить надёжность за счёт затрат времени, увеличивая мажоранту.

Пример описания программы:

PRIMARY: UM1086(3, 20., 10., 1.)

UM1318 — моделирование пучка частиц равномерной плотности, вылетающих из прямоугольника

$RINTRG(2) \times RINTRG(3)$, расположенного перпендикулярно оси X на координате RINTRG(1). Направление пучка задаётся вектором SINTRG. Тип частицы задаётся именем в параметре программы, например:

PRIMARY: UM1318('MU- ')

UM1319 — генерация одной частицы, сферически симметрично вылетающей из начала координат. Тип частицы задаётся текстовой константой во входном параметре программы:

PRIMARY: UM1319('*name*␣␣␣␣').

UM1360 — рождение псевдоскаляра (π^0, η, η') и фотона в e^+e^- аннигиляции. Имя псевдоскалярной частицы (PI0, ETA0 или ETA0P) записывается в параметре программы, например:

PRIMARY: UM1360('ETA0 ')

UM1361 — рождение двух псевдоскалярных мезонов (K^+K^-, K_SK_L, $\pi^+\pi^-$) в e^+e^- аннигиляции. В двух параметрах необходимо ввести имена двух рождающихся частиц, например:

PRIMARY: UM1361('K+ ','K- ')

UM1368 — моделирование рождения двух частиц с заданным углом между ними. Первые два параметра — имена частиц, третий параметр — угол (градусы), четвёртый — отношение энергий частиц. Пример:

PRIMARY: UM1368('PI+ ','PI- ', 30., 0.9)

UM1376 — моделирование процесса трёхквантовой аннигиляции. Первый параметр — порог на энергию фотона (МэВ), второй — минимальный угол фотона с осью пучков Z (градусы).

UM1377 — моделирование процесса двухквантовой аннигиляции с учётом излучения мягких фотонов. Первый параметр — порог на энергию фотона (МэВ), второй — минимальный угол фотона с осью пучков Z (градусы), третий — управляющее слово типа *character* $*$ 4: если '2GAM' — моделируется только часть сечения, когда энергия

мягких фотонов меньше порога; '3GAM' — моделируется только часть сечения с энергией фотонов выше порога.

9.2 Программы моделирования геометрических блоков

9.2.1 Немагнитный вариант

UM1300 — тип геометрии — симметричная "арбузная" долька. Блок задаётся 6-ю числами: радиусы сфер, ограничивающих дольку; косинус и синус угла φ боковых плоскостей (ось X — ось симметрии дольки, боковые плоскости $\pm\varphi$); косинус и синус угла θ торцевых плоскостей, ограничивающих дольку. $\varphi < \pi, 0 < \theta < \pi$. Если значение третьего параметра больше 2, то это означает, что вместо косинусов и синусов в четвёртом и шестом параметре заданы соответствующие углы в градусах.

UM1301 — тип геометрии — усечённая "арбузная" долька. Блок задаётся 8-ю числами: радиусы сфер, ограничивающих дольку; косинус и синус угла φ боковых плоскостей; косинус и синус угла θ 1-й плоскости, ограничивающей дольку по θ; косинус и синус угла θ 2-й плоскости. $\pi > \varphi > 0, \pi > \theta_2 > \theta_1 > 0$. Если значение третьего параметра больше 2, то это означает, что вместо косинусов и синусов в четвёртом, шестом и восьмом параметре заданы соответствующие углы в градусах.

UM1302 — тип геометрии — шар. Блок задаётся одним числом — радиусом шара.

UM1303 — тип геометрии — усечённая 4-гранная пирамида. Блок задаётся 7-ю числами: высота большего основания над плоскостью XY; высота вершины пирамиды над большим основанием; высота пирамиды; размеры большего основания (это симметричная трапеция с основаниями, параллельными оси X): расстояния от проекции вершины пирамиды до большего и меньшего оснований трапеции; полудлина большего основания трапеции, полудлина меньшего основания.

UM1304 — тип геометрии — прямоугольный параллелепипед. Блок задаётся 6-ю числами: полуразмеры параллелепипеда; координаты его центра.

UM1745 — тип геометрии — цилиндр, смещённый по Z. 6 параметров: внутренний и внешний радиусы (со знаком), Z-координата и направление (±1) по каждому из торцов. Положительное направление означает, что внутренность блока расположена по вектору нормали к торцу, проведённому из начала системы координат.

UM1747 — тип геометрии — конус, ограниченный плоскостью. Ось конуса совпадает с Z, плоскость перпендикулярна Z. 5 параметров: Z-координата плоскости, напрвление внутренности блока (±1), Z-координата вершины конуса, тангенс угла раствора конуса, направление внутренностей относительно вершины конуса (±1, "+" — вдоль Z, "-" — против Z).

9.2.2 Магнитный вариант

UM1007 — тип геометрии — прямоугольный параллелепипед. Задаётся шестью параметрами X1, X2, Y1, Y2, Z1, Z2 (координаты граней параллелепипеда по осям X, Y и Z, соответственно).

UM1814 — тип геометрии — цилиндр. Основание в плоскости (Y,Z). Задаётся пятью параметрами: X1, X2, Y, Z, R (где X1 < X2, (Y,Z) — координаты центра основания, R — радиус).

UM1844 — тип геометрии — прямая призма, в основании которой лежит равнобедренная трапеция. Основание в плоскости (X,Y). Задаётся семью параметрами X, Y, Z, HX, HZ, YA, YB, где (X,Y,Z) — точка центра грани призмы, которая включает меньшее основание трапеции; HX — высота трапеции; HZ — высота призмы; YA и YB основания трапеции, где YA < YB.

9.3 Моделирование проволочной камеры

UM1059 — плоская проволочная камера, геометрия которой задаётся тремя числами: X — координата пересечения плоскости камеры с осью X, шаг между проволочками, Y — координата первой проволочки (всё в см). Плоскость камеры перпендикулярна оси X, проволочки направлены по оси Z, длина проволочки бесконечна (ограничивается габаритами охватывающего блока). Срабатывает ближайшая проволочка к точке пересечения траектории частицы с плоскостью камеры.

9.4 Генерация амплитуды с проволочки

UM1060 — генерация амплитуд — условная. Реальная цель программы — записать информацию о срабатывании проволочек камеры для последующего использования в программах вычисления параметров. "Амплитудный" комплект состоит из трёх величин, в которых при срабатывании очередной проволочки производятся следующие изменения: в первую компоненту прибавляется единица, во вторую — номер проволочки, в третью — квадрат номера проволочки.

9.5 Генерация амплитуды в счётчике

UM1056 — моделирование "сцинтилляционного счётчика". В амплитудном комплекте две величины: выделившаяся энергия в счётчике (МэВ) за счёт ионизационных потерь и время первой добавки в амплитуду (нсек). Выделившаяся энергия поправляется на эффект подавления световыхода при большой плотности ионизации (см. [1], стр.38).

UM1095 — копия программы UM1056, в которой удалён учёт насыщения световыхода при большой плотности ионизации.

9.6 Программы вычисления параметров события

UM1057 — входных параметров нет, выходных параметров — переменное число (сколько будет описано). Программа рассчитана на информацию, генерируемую программой UM1056. Первый выходной параметр — сумма всех амплитуд счётчиков, второй параметр — первая амплитуда, третий параметр — вторая амплитуда и т.д.

UM1061 — входных параметров нет, выходных параметров — переменное число, кратное четырём. Каждая такая группа соответствует очередной проволочной камере: количество сработавших проволочек, количество срабатываний проволочек (с учётом кратности), средняя координата, ср.-кв. разброс. Последние два параметра тоже с учётом кратности. Программа рассчитана на информацию от программ UM1059, UM1060. Координата прохождения частицы равна Y-координате ближайшей к ней проволочки в соответствующей локальной системе координат.

9.7 Моделирование дискриминаторов амплитуд

UM1864 — Может работать с переменным количеством входов. Имеет два входных порога. Если сумма входных импульсов попадает в интервал между порогами, то на выходе "1".

9.8 Моделирование блоков логической электроники

UM1865 — логическая сумма входов. Если хотя бы на один из входов подана логическая "1", то на выходе "1". Может работать с переменным количеством входов.

UM1866 — совпадение входных импульсов. Если на всех входах "1", то на выходе тоже "1". Может работать с переменным количеством входов.

UM1867 — инверсия суммы. Если на всех входах "0", то на выходе "1". Может работать с переменным количеством входов.

9.9 Моделирование взаимодействия частиц с веществом

UM1200 — процесс тормозного излучения (e^+ или e^- на ядре атома).

UM1210 — процесс рождения e^+e^- - пар фотоном на ядре атома.

UM1220 — аннигиляция позитрона.

UM1221 — рассеяние заряженных частиц на электронах атома с большими передачами энергии (рождение δ-электронов).

UM1230 — комптоновское рассеяние.

UM1240 — фотоэффект.

UM1250 — рэлеевское (когерентное) рассеяние фотона.

UM1316 — упругое ядерное взаимодействие $\pi^+, \pi^-, K_S^0, K_L^0, K^+, K^-, p, n, \bar{p}, \bar{n}$.

UM1363 — рождение δ - электронов.

UM1371 — неупругое ядерное взаимодействие π^+, π^-, K_S^0, K_L^0, K^+, $K^-, p, n, \bar{p}, \bar{n}$.

UM1611 — многократное рассеяние заряженных частиц.

UM1630 — ионизационные потери заряженных частиц.

UM1635 — тормозное излучение мюонов.

9.10 Взаимодействие частиц в остановке

UM1320 — аннигиляция позитрона в остановке.

UM1323 — захват μ^- ядром.

UM1373 — ядерное взаимодействие K^- в остановке.

UM1378 — ядерное взаимодействие нейтронов в остановке.

UM1645 — ядерное взаимодействие $\bar{p}$ в остановке.

UM1650 — ядерное взаимодействие π^- в остановке.

9.11 Распады нестабильных частиц

UM1065 — изотропный двухчастичный распад.

UM1102 — распады $K_L^0 \to \pi^0\pi^0\pi^0$, $\eta^0 \to \pi^0\pi^0\pi^0$, $\eta^0 \to \pi^+\pi^-\pi^0$.

UM1104 — распады $K_L^0 \to \pi^+\pi^-\pi^0$, $K^+ \to \pi^+\pi^0\pi^0$, $K^+ \to \pi^+\pi^-\pi^+$, $K^- \to \pi^-\pi^0\pi^0$, $K^- \to \pi^-\pi^-\pi^+$.

UM1103 — распады $\mu^- \to e^-\nu_\mu\bar{\nu}_e$, $\mu^+ \to e^+\bar{\nu}_\mu\nu_e$, $\tau^- \to e^-\bar{\nu}_e\nu_\tau$, $\tau^+ \to e^+\nu_e\bar{\nu}_\tau$, $\tau^- \to \mu^-\bar{\nu}_\mu\nu_\tau$, $\tau^+ \to \mu^+\nu_\mu\bar{\nu}_\tau$.

UM1105 — распады $K_L^0 \to \pi^+\mu^-\bar{\nu}_\mu$, $K_L^0 \to \pi^-\mu^+\nu_\mu$, $K_L^0 \to \pi^+e^-\bar{\nu}_e$, $K_L^0 \to \pi^-e^+\nu_e$, $K^- \to \pi^0\mu^-\bar{\nu}_\mu$, $K^- \to \pi^0e^-\bar{\nu}_e$, $K^+ \to \pi^0\mu^+\nu_\mu$, $K^+ \to \pi^0e^+\nu_e$.

UM1106 — распады $\pi^0 \to e^-e^+\gamma$, $\eta \to^0 e^+e^-\gamma$.

UM1107 — распады $n \to pe^-\bar{\nu}_e$, $\bar{n} \to \bar{p}e^+\nu_e$.

UM1108 — распады $\eta^0 \to \pi^+\pi^-\gamma$, $\eta^{0\prime} \to \pi^+\pi^-\gamma$.

UM1109 — распады $\eta^{0\prime} \to \eta^0\pi^+\pi^-$, $\eta^{0\prime} \to \eta^0\pi^0\pi^0$.

9.12 Программы распечатки вспомогательной информации

UM1322 — распечатка каждого перемещения. Имеет смысл помещать в точки MOVE, ENTRANCE, NEXTPART.

UM1379 — печать о конце жизни частицы (точка ENDLIFE).

10 Стандартные файлы данных

Кроме программ, в комплекс UNIMOD2 входит набор некоторых стандартных файлов, включение которых в заказ существенно облегчает работу по оформлению счёта. При необходимости эти файлы можно использовать как основу для создания своей версии, скопировав файл в свою директорию с изменением имени. Расширение имени у всех файлов ".dat".

Большинство стандартных программ описано в нижеперечисленных массивах данных, так что обычно достаточно указать в заказе на счёт, например, включение массива ALLGEOM2:

=(ALLGEOM2)

чтобы все стандартные геометрические программы были описаны, и тогда их имена можно использовать в описании своих геометрических блоков.

Ниже перечисляются стандартные файлы с кратким комментарием. Так как эти файлы написаны по общим правилам интерпретатора ввода (раздел 4), то детали их устройства легко понимаются по их текстам.

PROPERT2 — описание свойств частиц, составленное по данным [28].

ALLGEOM2 — описание имеющихся геометрических программ.

STNINTR2 — описание стандартного набора программ взаимодействия частиц с веществом.

STANDEC2 — описание программ распадов.

STNMATR2 — описание стандартного набора веществ. Описано большинство простых веществ, названия выбраны в соответствии с таблицей Менделеева, например, для железа — FE, для меди — CU и т.д.

STANJOB2 — вызов стандартного набора массивов:

PROPERT2 — свойства частиц,

STNINTR2 — стандартный набор процессов взаимодействия,

ALLGEOM2 — геометрические программы,

STANDEC2 — программы распадов,

STNMATR2 — описание веществ.

TSUNI1 — пример заказа на счёт UNIMOD2 без магнитного поля (TSUNI1.LIS — результат счёта).

TSUNI2 — пример заказа на счёт UNIMOD2 с магнитным полем (TSUNI2.LIS — результат счёта).

11 Таблица свойств частиц

В таблице приведено текущее состояние таблицы свойств частиц, составленное по данным [28].

№ пп	Название частицы	Масса (МэВ)	Заряд в ед. e^+	Спин в ед. $\hbar$	Полн.маг. момент в ед.яд. магнетона	Произв. скор.св. на время жизни, см	Вероятность распада, % (продукты распада)
1	PHOTON	0	0	1	0	Стабильная	
2	ELECTRON	0.511	-1	1/2	-1838.28	Стабильная	
3	POSITRON	0.511	1	1/2	1838.28	Стабильная	
4	NU-E	0	0	1/2	0	Стабильная	
5	A/NU-E	0	0	1/2	0	Стабильная	
6	NU-MU	0	0	1/2	0	Стабильная	
7	A/NU-MU	0	0	1/2	0	Стабильная	
8	NU-TAU	0	0	1/2	0	Стабильная	
9	A/NU-TAU	0	0	1/2	0	Стабильная	
10	MU-	105.66	-1	1/2	-8.89	$6.587 \cdot 10^4$	98.6($e^-, \nu_\mu, \bar{\nu}_e$)
11	MU+	105.66	1	1/2	8.89	$6.587 \cdot 10^4$	98.6($e^+, \bar{\nu}_\mu, \nu_e$)
12	PI0	134.96	0	0	0	$2.6 \cdot 10^{-6}$	98.85($\gamma\gamma$) 1.15($e^-e^+\gamma$)
13	PI-	139.57	-1	0	0	780.4	100($\mu^-\bar{\nu}_\mu$)
14	PI+	139.57	1	0	0	780.4	100($\mu^+\nu_\mu$)

№ пп	Название частицы	Масса (МэВ)	Заряд в ед. e^+	Спин в ед. $\hbar$	Полн.маг. момент в ед.яд. магнетона	Произв. скор.св. на время жизни, см	Вероятность распада, % (продукты распада)
15	K0/S	497.67	0	0	0	2.675	68.61($\pi^+\pi_-$)
							31.39($\pi^0\pi^0$)
16	K0/L	497.67	0	0	0	1554	21.5($\pi^0\pi^0\pi^0$)
							12.4($\pi^+\pi^-\pi^0$)
							13.55($\pi^+\mu^-\bar{\nu}_\mu$)
							13.55($\pi^-\mu^+\nu_\mu$)
							19.35($\pi^-e^+\nu_e$)
							19.35($\pi^+e^-\bar{\nu}_e$)
							0.2($\pi^+\pi^-$)
							0.1($\pi^0\pi^0$)
17	K-	493.67	-1	0	0	370.9	63.5($\mu^-\bar{\nu}_\mu$)
							21.17($\pi^-\pi^0$)
							5.6($\pi^-\pi^-\pi^+$)
							1.73($\pi^-\pi^0\pi^0$)
							3.18($\pi^0\mu^-\bar{\nu}_\mu$)
							4.82($\pi^0e^-\bar{\nu}_e$)
18	K+	493.67	1	0	0	370.9	63.5($\mu^+\nu_\mu$)
							21.17($\pi^+\pi^0$)
							5.6($\pi^+\pi^-\pi^+$)
							1.73($\pi^+\pi^0\pi^0$)
							3.18($\pi^0\mu^+\nu_\mu$)
							4.82($\pi^0e^+\nu_e$)
19	ETA0	548.8	0	0	0	$1.9\cdot10^{-8}$	38.9($\gamma\gamma$)
							31.9($\pi^0\pi^0\pi^0$)
							23.7($\pi^+\pi^-\pi^0$)
							4.9($\pi^+\pi^-\gamma$)
							0.5($e^+e^-\gamma$)
20	PROTON	938.28	1	1/2	2.7928	Стабильная	
21	A/PROTON	938.28	-1	1/2	-2.7928	Стабильная	
22	NEUTRON	939.57	0	1/2	-1.913	$2.7\cdot10^{13}$	100($pe^-\bar{\nu}_e$)
23	A/NEUTRN	939.57	0	1/2	1.913	$2.7\cdot10^{13}$	100($\bar{p}e^+\nu_e$)
24	ETA0P	957.57	0	0	0	$8.2\cdot10^{-11}$	44.14($\eta^0\pi^+\pi^-$)
							22.06($\eta^0\pi^0\pi^0$)
							30($\pi^+\pi^-\gamma$)
							2.4($\pi^+\pi^-\pi^0\gamma$)
							1.9($\gamma\gamma$)
25	LAMBDA	1115.60	0	1/2	-0.61	7.89	64.2($p\pi^-$)
							35.8($n\pi^0$)
26	A/LAMBDA	1115.60	0	1/2	0.61	7.89	64.2($\bar{p}\pi^+$)
							35.8($\bar{n}\pi^0$)
27	SIGMA+	1189.37	1	1/2	2.38	2.40	51.64($p\pi^0$)
							48.36($n\pi^+$)
28	A/SIGMA+	1189.37	-1	1/2	-2.38	2.40	51.64($\bar{p}\pi^0$)
							48.36($\bar{n}\pi^-$)

№ пп	Название частицы	Масса (МэВ)	Заряд в ед. e^+	Спин в ед. $\hbar$	Полн.маг. момент в ед.яд. магнетона	Произв. скор.св. на время жизни, см	Вероятность распада, % (продукты распада)
29	SIGMA0	1192.47	0	1/2	0	$1.7\cdot10^{-9}$	100($\Lambda\gamma$)
30	A/SIGMA0	1192.47	0	1/2	0	$1.7\cdot10^{-9}$	100($\bar{\Lambda}\gamma$)
31	SIGMA-	1197.35	-1	1/2	-1.14	4.44	100($n\pi^-$)
32	A/SIGMA-	1197.35	1	1/2	1.14	4.44	100($\bar{n}\pi^+$)
33	KSI0	1314.9	0	1/2	-1.25	8.69	100($\Lambda\pi^0$)
34	A/KSI0	1314.9	0	1/2	1.25	8.69	100($\bar{\Lambda}\pi^0$)
35	KSI-	1321.32	-1	1/2	-0.69	4.92	100($\Lambda\pi^-$)
36	A/KSI-	1321.32	1	1/2	0.69	4.92	100($\bar{\Lambda}\pi^+$)
37	OMEGA-	1672.45	-1	1.5	0	2.46	67.8(ΛK^-)
							23.6($\Xi^0\pi^-$)
							8.6($\Xi^-\pi^0$)
38	A/OMEGA-	1672.45	1	3/2	0	2.46	67.8($\bar{\Lambda}K^+$)
							23.6($\bar{\Xi}^0\pi^+$)
							8.6($\bar{\Xi}^-\pi^0$)
39	TAU-	1784.0	-1	1/2	0.526	0.01	17.6($\mu^-\bar{\nu}_\mu\nu_\tau$)
							17.4($e^-\bar{\nu}_e\nu_\tau$)
							10.1($\pi^-\nu_\tau$)
							0.67($K^-\nu_\tau$)
							21.6($\pi^-\pi^0\nu_\tau$)
							7.0($\pi^-\pi^-\pi^+\nu_\tau$)
							25.63
40	TAU+	1784.0	1	1/2	-0.526	0.01	17.6($\mu^+\nu_\mu\bar{\nu}_\tau$)
							17.4($e^+\nu_e\bar{\nu}_\tau$)
							10.1($\pi^+\bar{\nu}_\tau$)
							0.67($K^+\bar{\nu}_\tau$)
							21.6($\pi^+\pi^0\bar{\nu}_\tau$)
							7.0($\pi^+\pi^+\pi^-\bar{\nu}_\tau$)
							25.63
41	DEUTERON	1875.613	1	1	0	Стабильная	

12 Простые примеры заданий

Приведём два простых примера: один без магнитного поля (файл TSUNI1.DAT), другой — с магнитным полем (файл TSUNI2.DAT).

Файл TSUNI1.DAT

```
=(STANJOB2)
**CUTS
CUT1:1000,5,1000,0.5,1.,2.,90
PHOTON(1),ELECTRON(2),POSITRON(2)
**MATERIALS
CBIHEC:82,208,11.4,0.8
**ROUTINE
INPUT:
COUNAMPL:UM1095(2)
PRIMARY:UM1054(1)
PARAMETR: UM1057(;SUM,A1,A2,A3,A4)
          UM1061(;NT1,NW1,X1,SIG1,NT2,NW2,X2,SIG2)
CHAMBER:UM1059(3)
CHAMPL: UM1060(3)
&GEOMETRY:TEST07(6)
RESULT: UM1091
**RESPONCES
AMP1: UM1095, AMP2: , AMP3: , AMP4:  ,
**BLOCKS
KB1 : ,,,CUT1,,1,1,UM1007(0,45,-100,100,-100,100)
KUB1:KB1,CBIHEC,,CUT1,AMP1,1,1,UM1007(9.8,10.2,-100,100,-100,100)
KUB2:KB1,CBIHEC,,CUT1,AMP2,1,1,UM1007(19.8,20.2,-100,100,-100,100)
KUB3:KB1,CBIHEC,,CUT1,AMP3,1,1,UM1007(29.8,30.2,-100,100,-100,100)
KUB4:KB1,CBIHEC,,CUT1,AMP4,1,1,UM1007(39.8,40.2,-100,100,-100,100)
**CHAMBERS
CHAM1: ,KB1,,1000,UM1059(19,0.1,-50)
CHAM2: ,KB1,,1000,UM1059(42,0.1,-50)
**HITS
AMPK1:UM1060,CHAM1,1,1000
AMPK2:UM1060,CHAM2,1,1000
**CONSTRAINT
a1,-1,0.01
a2,-1,0.01
**FORMULA
f1:a1*a2
**HISTOGRAMS
 (30),SUM, 60, 4,
```



```
a1(30),SUM, 60, 4,
f1(30),SUM, 60, 4,
 (30),A1 , 15, 1,
 (30),A2 , 30, 2,
 (30),A3 , 30, 2,
 (30),A4 , 30, 2,
 (30),NT1, 14.5, 1,
 (30),NW1, 14.5, 1,
 (30),X1 , 0 ,0.5,
 (30),SIG1,2,0.5,
 (30),NW2, 14.5, 1,
 (30),X2 , 0 ,1,
 (30),SIG2,4,1,
TEST: (24*24), NW1,10.5,1, NW2,10.5,1
**REGIME
ENDTASK:100,010
enERGY:1000.0,0
rand:z1f28f862f172179b
```

Файл TSUNI2.DAT

```
=(STANJOB2)
**CUTS
CUT1:1000,5,1000,0.9,1.,2.,60
PHOTON(1),ELECTRON(2),POSITRON(2)
**MATERIALS
CBIHEC:82,208,11.4,0.8
**ROUTINES
INPUT:
COUNAMPL:UM1095(2)
PRIMARY:UM1054( 1)
PARAMETR: UM1057(;SUM,A1,A2,A3,A4)
          UM1061(;NT1,NW1,X1,SIG1,NT2,NW2,X2,SIG2)
CHAMBER:UM1059(3)
CHAMPL: UM1060(3)
RESULT:
**RESPONCES
AMP1: UM1095, AMP2: , AMP3: , AMP4:  ,
**BLOCKS
KB1 : ,,,CUT1,,1,1,UM1007(0,45,-100,100,-100,100)
KUB1:KB1,CBIHEC,,CUT1,AMP1,1,1,UM1007(9.8,10.2,-100,100,-100,100)
KUB2:KB1,CBIHEC,,CUT1,AMP2,1,1,UM1007(19.8,20.2,-100,100,-100,100)
KUB3:KB1,CBIHEC,,CUT1,AMP3,1,1,UM1007(29.8,30.2,-100,100,-100,100)
KUB4:KB1,CBIHEC,,CUT1,AMP4,1,1,UM1007(39.8,40.2,-100,100,-100,100)
```

```
**CHAMBERS
CHAM1: ,KB1,,1000,UM1059(19,0.1,-50)
CHAM2: ,KB1,,1000,UM1059(42,0.1,-50)
**HITS
AMPK1:UM1060,CHAM1,1,1000
AMPK2:UM1060,CHAM2,1,1000
**HISTOGRAMS
 (30),SUM, 60, 4,
 (30),A1 , 15, 1,
 (30),A2 , 30, 2,
 (30),A3 , 30, 2,
 (30),A4 , 30, 2,
 (30),NT1, 14.5, 1,
 (30),NW1, 14.5, 1,
 (30),X1 , 0 ,0.5,
 (30),SIG1,2,0.5,
 (30),NW2, 14.5, 1,
 (30),X2 , 0 ,  1,
 (30),SIG2,4,  1,
**MFIELD
UNIF:UF,0,5,0
**MVOLUMES
OTC1:,,UNIF,UM1007(-100,100,-100,100,-100,100)
**REGIME
FIELDFAC:1
& RAND:Z453D544104AD31F5,
ENDTASK:100,10
ENERGY: 100.0,0
PRINT:0
```


Список литературы

[1] *А.Д. Букин, В.Н. Иванченко, М.Ю. Лельчук, В.А. Таюрский, С.И. Эйдельман, В.И. Юрченко.* UNIMOD — универсальная программа моделирования экспериментов на встречных e^+e^- - пучках. Препринт ИЯФ СО АН СССР 84-33, Новосибирск, 1984.

[2] *S.I. Dolinsky, V.P. Druzhinin, M.S. Dubrovin et al.* Summary of Experiments with the Neutral Detector at the e^+e^- Storage Ring VEPP-2M. Physics Reports 202(1991)99–170.

[3] *A.S. Artamonov, S.E. Baru, A.E. Blinov et al.* High precision measurement of the Υ-meson Mass, Phys. Letters 118B (1982) 225;
S.E. Baru et al. Total cross section of two-photon production of hadrons. Z. Phys. C — Particles and Fields 53 (1992) 219–224;
A.E. Blinov et al. Pion pair production in photon-photon collisions. Z. Phys. C — Particles and Fields 53 (1992) 33–39;
S.E. Baru, A.E. Blinov, V.E. Blinov et al. Measurement of the two-photon width of the a_2, η', η. Z. Phys. C — Particles and Fields 48 (1990) 581–585.

[4] *J.H. Hubbel, I.Overbo.* J. Chem. Ref. Data, vol.8, №1, p.1979.

[5] *А.Д. Букин, Н.А. Грозина, М.С. Дубровин, И.Л. Кац, В.Н. Иванченко, В.А. Таюрский, С.И. Эйдельман.* UNIMOD-2 — универсальная программа моделирования экспериментов на встречных e^+e^- - пучках.
Часть 1. Общее описание. Препринт ИЯФ СО АН СССР 90-93, Новосибирск, 1990.
Часть 2. Руководство пользователя. Препринт ИЯФ СО АН СССР 90-95, Новосибирск, 1990.
Часть 3. Руководство пользователя-программиста. Препринт ИЯФ СО АН СССР 90-96, Новосибирск, 1990.

[6] *В.М. Аульченко, В.А. Аксёнов, П.М. Бесчастнов и др.* СНД — сферический нейтральный детектор для ВЭПП-2М. Препринт ИЯФ СО АН СССР 87-36, Новосибирск, 1987;
V.M. Aulchenko, B.O. Baibusinov, T.V. Baier et al. SND — detector for VEPP-2M and Φ-factory. Proceedings of the Workshop on Physics and Detectors for DAΦNE. Frascati, April 9–12, 1991, pp. 605–613.

[7] *R.Brun, F.Bruyant, M.Maire, A.C.McPherson, P.Zanarini.* GEANT3. CERN preprint DD/EE/84-1, Genève, 1987.

[8] *A.D. Bukin, M.S. Dubrovin, N.A. Grozina, V.N. Ivanchenko, V.A. Tayurski, S.I. Eidelman.* UNIMOD2 — Universal Monte Carlo code for simulation of e^+e^--experiments. Proceedings of Workshop on Detector and Event Simulation in High Energy Physics, 8 - 12 April 1991, NIKHEF, Amsterdam, The Netherlands, pp. 79–85.

[9] *L.G. Isaeva, B.A. Lazarenko, D.M. Nikolenko, S.G. Popov, I.A. Rachek, Yu.G. Ukraintsev, E.P. Tsentalovich, B.B. Wojtsekhowski and V.V. Nelyubin.* Detector system for $e-d$ scattering experiments on the VEPP-3 storage ring. Nuclear Instruments and Methods in Physics Research, A325(1993), pp.16-22;
E.P. Tsentalovich. Results of the Novosibirsk T20 experiment. Presented at the 13th International Conference on Particles and Nuclei, Perugia, Italy, June 28 - July 2, 1993.

[10] *Дж. Дайамендстоун и компания "И-Ар-Ай Трейнинг".* Использование ОС VAX/VMS. М: "Мир", 1992.

[11] *Б.В. Керниган, Р. Пайк.* UNIX — универсальная среда программирования. М: "Финансы и статистика", 1992.

[12] *А.Д. Букин, В.Н. Иванченко.* Гистограммная программа GIST. Препринт ИЯФ СО РАН 93-81, Новосибирск, 1993.

[13] *R.Brun, O.Couet, C.Vandoni, P.Zanarini, M.Goossens.* PAW — Physics Analysis Workstation, the complete reference. Version 1.07, CERN program library entry Q121, CERN Geneva, Switzerland, 1989.

[14] *A.D. Bukin.* UNIMOD2: Monte Carlo code for simulation of high energy physics experiments. Some special features. Submitted poster to the Workshop on Detector and Event Simulation in High Energy Physics, February 22 – 26, 1993, Tallahassee, Florida, USA.

[15] *Э.Сторм, Х.Израэль.* Сечения взаимодействия гамма-излучения. Москва. Атомиздат 1973.

[16] *Y.S.Tsai.* R.M.Ph. 46(1974)813.

[17] *В.А. Таюрский.* EMSH — программа расчёта прохождения через вещество электронов и фотонов при энергии 10 кэВ – 1 ТэВ. Препринт ИЯФ СО АН СССР 89–16, Новосибирск, 1989.

[18] *S.M. Seltzer, M.J. Berger,* NIM B12 (1985) 95.

[19] *L. Kissel et al.* At. Data and Nucl. Data Tables 28 (1983) 381.

[20] *В. Гайтлер.* Квантовая теория излучения. Москва. ИЛ 1956.

[21] *Букин А.Д., Грозина Н.А.* Моделирование флуктуаций ионизационных потерь тяжёлых заряженных частиц.— Препринт ИЯФ 87-9. Новосибирск 1987.

[22] *A.D. Bukin, N.A. Grozina.* Monte Carlo simulation of fluctuations of the ionization losses of heavy charged particles. Computer Physics Communications 78, №3 (1994) 287.

[23] *E. Massaro.* NIM A251(1986)546.

[24] *Sternheimer R.M.* Phys.Rev. B3 (1971), 3681.

[25] *В.Б. Берестецкий и др.* Релятивистская квантовая теория. Часть 1. — М: Наука, 1968.

[26] *Бугаев и др.* Космические мюоны и нейтрино. Атомиздат, 1970.

[27] *H.A. Bethe.* Phys.Rev. 89 (1953) 1256.

[28] Review of Particle Properties, Phys. Letters 204B (1988).

[29] *A.D. Bukin, N.A. Grozina, V.N. Ivanchenko, K. Hänßgen.* UNIMOD2 — the Universal Monte Carlo Code. 4. Simulation of Hadron-Nucleus Interactions with the NUC92 Code. Preprint BUDKERINP 92-93, Novosibirsk, 1992.

[30] *K. Hänßgen, J. Ranft.* Comput. Phys. Commun. 39(1986)37; ibid 39(1986)53.

[31] *А.Д. Букин, В.П. Дружинин, В.Н. Иванченко и др.* Моделирование взаимодействия адронов с ядрами. Сравнение расчётов по программе NUCRIN с экспериментальными данными. Препринт ИЯФ 86-18, Новосибирск, 1986.

[32] *Н.А. Грозина.* Моделирование аннигиляции антипротонов на ядрах вещества в точке остановки. Препринт ИЯФ СО АН СССР 90-145, Новосибирск, 1990.

[33] *A.D. Bukin.* Simulation of the Elastic Scattering Process $e^+e^- \to e^+e^-$ with Radiative Corrections. Preprint INP 85-124, Novosibirsk, 1985.

Содержание

1 Введение 3

2 Основные принципы UNIMOD2 4

3 Процедура запуска программы 7
- 3.1 Система рабочих файлов UNIMOD2 . . . 7
- 3.2 Работа на VAX/VMS . . . 9
- 3.3 Работа в системе UNIX . . . 11

4 Входной поток данных UNIMOD2 15
- 4.1 **MATERIALS — описание веществ . . . 18
- 4.2 **REFERENCE — локальные системы координат . . . 19
- 4.3 **PARTICLES — свойства частиц . . . 19
- 4.4 **CUTS — константы, управляющие развитием каскада частиц и задающие правила выбора очередного перемещения . . . 20
- 4.5 **ROUTINES — регистрация подпрограмм . . . 21
- 4.6 **GATES — временны́е ворота . . . 25
- 4.7 **RESPONCES — отклик детектора в счётчиках . . . 26
- 4.8 **BLOCKS — описание блоков вещества . . . 26
- 4.9 **CHAMBERS — описание проволочных камер . . . 27
- 4.10 **HITS — амплитуды с проволочных камер . . . 28
- 4.11 **DISCRIMINATORS — типы дискриминаторов амплитуд 29
- 4.12 **LOGICMODULES — типы блоков логической электроники . . . 29
- 4.13 **TRIGGER — схема электроники детектора . . . 29
- 4.14 **CONSTRAINTS — список условий на параметры событий . . . 30
- 4.15 **FORMULAS — логические формулы отбора событий . 30
- 4.16 **HISTOGRAMS — заказ на построение гистограмм . . . 31
- 4.17 **MFIELD — описание способа вычисления магнитного поля . . . 31
- 4.18 **MVOLUMES — описание отсеков магнитного поля . . . 32
- 4.19 **REGIME — общие константы . . . 33
- 4.20 **PRIVATE — ввод текстов подпрограмм пользователя . 34
- 4.21 **PRESS — условия на блоки вещества . . . 35
- 4.22 **CLEAN — удаление “лишних” программ вычисления гистограммных параметров события . . . 36

- 4.23 Распечатка введённых данных . . . 36

5 Моделирование взаимодействия частиц с веществом 37
- 5.1 Рождение e^+e^- пар фотоном в поле ядра . . . 39
- 5.2 Тормозное излучение электронов и позитронов . . . 39
- 5.3 Комптон-эффект и рэлеевское рассеяние . . . 40
- 5.4 Фотоэффект . . . 40
- 5.5 Ионизационные потери . . . 40
- 5.6 Аннигиляция позитронов . . . 40
- 5.7 Рассеяние заряженных частиц на электронах атомов . . . 41
- 5.8 Тормозное излучение мюонов . . . 41
- 5.9 Многократное рассеяние . . . 41
- 5.10 Распады нестабильных частиц . . . 41
- 5.11 Распады и взаимодействия остановившихся частиц . . . 41
- 5.12 Ядерные взаимодействия адронов . . . 42
- 5.13 Взаимодействие частиц со сложными веществами . . . 43

6 Общие области программы 43

7 Выходные файлы и номера логических устройств 66

8 Правила написания подпрограмм 68
- 8.1 Общие замечания . . . 68
  - 8.1.1 Типы переменных . . . 68
  - 8.1.2 Диагностические сообщения . . . 68
  - 8.1.3 Программная компонента BLOCK DATA . . . 69
  - 8.1.4 Вспомогательные подпрограммы . . . 69
- 8.2 Генерация параметров первичных частиц . . . 70
- 8.3 Программы моделирования процессов взаимодействия частиц с веществом . . . 72
- 8.4 Распады нестабильных частиц . . . 73
- 8.5 Программы моделирования геометрических блоков . . . 74
- 8.6 Моделирование ионизационных потерь . . . 76
- 8.7 Генерация угла многократного рассеяния . . . 76
- 8.8 Вычисление магнитного поля . . . 77
- 8.9 Моделирование проволочной камеры . . . 77
- 8.10 Генерация амплитуды с проволочки . . . 79
- 8.11 Генерация отклика детектора в счётчике . . . 79
- 8.12 Программы вычисления параметров события . . . 80

8.13 Программы, анализирующие класс события по информации из выделенного блока 80
8.14 Моделирование дискриминаторов амплитуд 81
8.15 Моделирование блоков логической электроники 82

9 Описание подпрограмм **82**
9.1 Генерация параметров первичных частиц 82
9.2 Программы моделирования геометрических блоков 85
9.2.1 Немагнитный вариант 85
9.2.2 Магнитный вариант 88
9.3 Моделирование проволочной камеры 88
9.4 Генерация амплитуды с проволочки 89
9.5 Генерация амплитуды в счётчике 89
9.6 Программы вычисления параметров события 89
9.7 Моделирование дискриминаторов амплитуд 90
9.8 Моделирование блоков логической электроники 90
9.9 Моделирование взаимодействия частиц с веществом 90
9.10 Взаимодействие частиц в остановке 91
9.11 Распады нестабильных частиц 91
9.12 Программы распечатки вспомогательной информации . . . 92

10 Стандартные файлы данных **92**

11 Таблица свойств частиц **93**

12 Простые примеры заданий **96**

А.Д. Букин, Н.А. Грозина, М.С. Дубровин,
В.Н. Иванченко, В.А. Таюрский, С.И. Эйдельман

UNIMOD 2—универсальная программа моделирования экспериментов на встречных e^+e^- пучках

5. Руководство пользователя версии 2.0

Ответственный за выпуск С.Г. Попов
Работа поступила 16 февраля 1994 г.

Сдано в набор 17 февраля 1994 г.
Подписано в печать 18 февраля 1994 г.
Формат бумаги 60×90 1/16 Объем 7.0 печ.л., 5.5 уч.-изд.л.
Тираж 120 экз. Бесплатно. Заказ N° 20

Обработано на IBM PC и отпечатано на
ротапринте ИЯФ им. Г.И. Будкера СО РАН,
Новосибирск, 630090, пр. академика Лаврентьева, 11.