

Institute for Nuclear Research
of the Russian Academy of Sciences

Федеральное государственное бюджетное учреждение науки
Институт ядерных исследований
Российской академии наук



Н.М. Соболевский

Алгоритм запоминания дерева адронного каскада
в транспортном коде SHIELD

препринт
preprint

ПРЕПРИНТ ИЯИ РАН
1398/2015
ФЕВРАЛЬ 2015

МОСКВА 2015 MOSCOW

Institute for Nuclear Research
of the Russian Academy of Sciences

Федеральное государственное бюджетное учреждение науки
Институт ядерных исследований
Российской академии наук



Н.М. Соболевский

Алгоритм запоминания дерева адронного каскада в транспортном коде SHIELD

ПРЕПРИНТ ИЯИ РАН
1398/2015

ФЕВРАЛЬ 2015
МОСКВА
2015

Institute for Nuclear Research
of the Russian Academy of Sciences
60-th October Anniversary prospect 7a,
Moscow 117312, Russia

Федеральное государственное бюджетное учреждение науки
Институт ядерных исследований Российской академии наук
проспект 60-летия Октября, 7а
Москва, 117312

Н.М.Соболевский

Алгоритм запоминания дерева адронного каскада
в транспортном коде SHIELD

Описан алгоритм моделирования и запоминания, для последующей обработки, дерева адронного каскада в сложной макроскопической мишени под действием адронов и ядер. Алгоритм реализован в транспортном коде SHIELD

N.M.Sobolevsky

Algorithm for storing of the hadron cascade tree in
the SHIELD transport code

An algorithm of modeling and storing for subsequent processing of the tree of hadron cascade in complex macroscopic targets under irradiation by hadrons and nuclei is described. The algorithm is implemented in the transport code SHIELD.

Н.М.Соболевский

Алгоритм запоминания дерева
адронного каскада
в транспортном коде SHIELD

Препринт 1398/2015

Февраль 2015

Подписано в печать 02.02.2015

Ф-т 60x84/8. Уч.-изд.л. 0,7. Зак. 22345 Тираж 50 экз.
БесплатноПечать цифровая
Издательский отделФедеральное государственное бюджетное учреждение науки
Институт ядерных исследований Российской академии наук

117312, Москва, проспект 60-летия Октября, 7а

ISBN 978-5-94274-285-0

© Федеральное государственное
бюджетное учреждение науки
Институт ядерных исследований
Российской академии наук, 2015
Institute for Nuclear Research
of the Russian Academy of Sciences, 2015

1. Введение

В процессе моделирования адронного каскада транспортным кодом SHIELD (<http://www.inr.ru/shield/>), каскад запоминается в специальных массивах. При этом запоминаются все акты упругого и неупругого взаимодействия, распада частиц*, их вылета из мишени или поглощения, и другая информация, о чем подробнее ниже. Таким образом, дерево каскада запоминается без потери физической информации, включая имеющиеся корреляции.

По окончании моделирования очередного каскада производится обработка дерева для регистрации результатов (построения спектров частиц, энерговыделения в зонах мишени, активации и др.). Возможна визуализация дерева. При необходимости деревья могут быть записаны на внешний носитель и сохранены для последующей обработки всего ансамбля деревьев. Перед началом моделирования следующего межъядерного каскада массивы дерева обнуляются.

Дерево адронного каскада состоит из ветвей. Каждая ветвь включает инициирующую частицу, в состоянии перед пробегом и в конечной точке пробега, ядро мишень (если имеется) и ядро/ядра-продукты, а также вторичные частицы ядерной реакции, которые могут инициировать новую ветвь. Ветвь может состоять только из частицы в состоянии перед пробегом и в конце пробега (например, при вылете из мишени). Вторичные частицы, которые не могут инициировать новую ветвь, записываются в массивы источников частиц на стадии пересылки продуктов взаимодействия/распада в массивы дерева.

Массивы источников содержат продукты распада мезонов (гамма-кванты, электроны, позитроны, нейтрино/антинейтрино), а также нейтроны низких энергий, с энергиями ниже 14.5 МэВ. В отличие от массивов дерева, массивы источников организованы не по ветвям, а в виде стека FIFO (см. раздел 7). Перенос нейтронов низких энергий моделируется далее с помощью программы LOENT на основе нейтронных констант БНАБ. Для моделирования переноса частиц из других источников необходимо привлекать внешние программы.

Массивы дерева – это двумерные массивы. Всего существует 4 массива с именами TREE, ITREE, TRAN, ITRAN для хранения вещественных и целых переменных. Например, строки массива TREE хранят координаты частицы, направление движения, энергию и др., а также параметры ядер-мишеней и ядер-продуктов. Строки массивов короткие, от 5 до 15 ячеек. Длина массивов может достигать десятков тысяч строк и более. При необходимости длина массивов легко меняется средствами языка Фортран 77. Структура массивов дерева и источников описана подробнее ниже, в разделах 5-8.

Моделирование и запоминание каскадного дерева управляется подпрограммой SUBROUTINE GENTRE (эта управляющая программа, вместе с подпрограммами, которые с ней непосредственно связаны, находится в файле **gentree.f**). Блок-схема подпрограммы GENTRE представлена в разделе 3. Чтобы не загромождать рисунок, на схеме не показан контроль переполнения массивов и диагностика.

* Под «частицами» понимаются адроны и транспортируемые ядерные фрагменты.

Отметим, что программа GENTRE вызывает не только программы из файла **gentree.f**. Программа GENTRE вызывает также программы задания/розыгрыша первичной частицы INSPAR, моделирования длины пробега частиц RANGE, моделирования распада мезонов DECAY, упругого ядерного рассеяния ELSCAT и моделирования ядерных реакций REACT (которая, в свою очередь, вызывает генератор ядерных реакций MSDM). Эти пять отдельных частей кода SHIELD здесь не рассматриваются.

2. Подпрограммы управления генерацией дерева

Прежде чем переходить к рассмотрению блок-схемы Рис. 1, опишем кратко функции подпрограмм управления генерацией дерева, которые находятся в файле **gentree.f**. Всего имеется 12 таких подпрограмм.

1. SUBROUTINE GENTRE(JSTAT)

Головная подпрограмма, управляющая моделированием и запоминанием дерева межъядерного каскада. Вход: JSTAT – номер дерева.

2. SUBROUTINE CLEAR

Очистка массивов дерева и источников перед началом моделирования очередного дерева.

3. SUBROUTINE ESTABL

Установка начальных значений управляющих переменных и счетчиков перед началом моделирования очередного дерева.

4. SUBROUTINE TAKPAR(ICOMPL)

Проверка, завершено ли моделирование очередного дерева. Если нет, извлечение из массива дерева частицы, инициирующей новую ветвь дерева. Перемещение извлеченной частицы вниз, в исходную позицию для создания новой ветви. Смещение части предыдущей ветви на одну линию вверх, чтобы занять освободившееся место в массиве дерева.

Выход: ICOMPL=+1 моделирование дерева закончено, иначе ICOMPL=0.

5. SUBROUTINE SENS(KOVER)

После моделирования пробега частицы пересылка шагов пробега частицы (через зоны мишени) из локального массива STEPR (для данного пробега) в общие массивы пробегов данного дерева TRAN и ITRAN.

Выход: KOVER=+1, если массивы TRAN и ITRAN переполнились. Иначе KOVER=0.

6. SUBROUTINE CLOSE(JABNOR)

Закрытие массивов дерева и массивов источников, если моделирование дерева завершено.

Вход: JABNOR=0 для нормального закрытия массивов дерева.

JABNOR=1 при аварийном закрытии (переполнение массивов).

7. SUBROUTINE CLOSAC

Аварийное закрытие массивов дерева и массивов источников частиц.

8. SUBROUTINE PREPAR

Если происходит ядерное взаимодействие, разыгрывается индекс ядра-мишени NUCLID и тип взаимодействия KSTATE (упругое или неупругое) для налетающей частицы в конечной точке ее пробега.

9. SUBROUTINE SEND(IOVER)

Пересылка вторичных частиц и ядер из массивов SPT и SNU MSDM-генератора или DECAY-генератора в массивы дерева и источников частиц. При этом быстрые ядра-продукты преобразуются в транспортируемые частицы. Программа SEND решает, куда пересылается данный продукт: в массивы дерева или источников.

Выход: IOVER=+1 если массивы дерева и/или источников переполнились,

Иначе IOVER=0.

10. SUBROUTINE NUCPAR(NUCFIR,NUCLAS,NPRFIR,NPRLAS,LINADD)

Вызывается программой SEND при преобразовании быстрых ядер-продуктов в транспортируемые частицы. При этом массивы дерева TREE и ITREE преобразуются соответствующим образом (see comments inside).

11. SUBROUTINE SENDNU(J,K)

Пересылка строки № J массива SNU(10,J) в строку № K массивов дерева TREE(15,K) и ITREE(7,K). Программа SENDNU вызывается программой SEND.

12. SUBROUTINE SENDPR(J,K,LLL)

Пересылка строки № J массива SPT(6,J) или в строку № K массивов дерева TREE(15,K) и ITREE(7,K), или в строку № K массивов источников. LLL – управляющий параметр. Программа SENDPR вызывается программой SEND.

3. Описание блок-схемы программы GENTRE

После входа в подпрограмму GENTRE (см. Рис. 1) фиксируется номер дерева NTREE и состояние датчика случайных чисел (seed) IXINIT. Используется датчик RANLUX из библиотеки CERN.

Далее производится очистка массивов дерева (CALL CLEAR) и установка исходных значений глобальных управляющих переменных и счетчиков (CALL ESTABL, см. следующий раздел).

После этого задаётся или разыгрывается первичная частица (CALL INSPAR), т.е. тип снаряда, точка входа в мишень, направление, энергия. Первичная частица сразу заносится в первую строку массивов дерева TREE, ITREE.

Обращение CALL TAKPAR(ICOMPL) извлекает из дерева и перемещает в ближайшую свободную строку дерева очередную частицу, которая открывает следующую ветвь (если это первичная частица, то она остается на месте). Все строки дерева, расположенные ниже извлеченной частицы, сдвигаются назад на одну строку. Если такой частицы в дереве не оказалось (ICOMPL=+1), то это означает, что моделирование дерева закончено, массивы закрываются (CALL CLOSE) и происходит выход из GENTRE.

Далее вызывается программа моделирования пробега (CALL RANGE), которая транспортирует частицу через зоны мишени до конечной точки пробега. В результате работы программы RANGE выдается флаг KSTATE состояния частицы в конце пробега (который на данном этапе может принимать значения KSTATE=0, 3-7, см. ниже), а также в массиве STEPR записывается информация о шагах пробега через зоны.

Обращение CALL SENS пересылает информацию о шагах данного пробега в массивы TRAN, ITRAN, в которых сохраняется информация о всех шагах всех пробегов в данном дереве.

Затем анализируется состояние частицы в конце пробега (флаг KSTATE):

KSTATE=7 – частица попала под обрезание ($E < E_{\text{cutoff}}$);

KSTATE=3 – вылет частицы из мишени;

KSTATE=4 – поглощение частицы из-за ионизационных потерь.

В указанных случаях генерация данной ветви завершена. В соответствующие счетчики событий добавляется по единице, модифицируется глобальная управляющая переменная $LINE2=LINE2+2$, и управление передается на начало новой ветви (CALL TAKPAR).

KSTATE=5 или 6 – распад на лету или в покое соответственно. Вызывается программа моделирования распадов (CALL DECAY), продукты распада пересылаются из массива SPT в массивы источников (CALL SEND); управление передается на начало новой ветви (CALL TAKPAR).

KSTATE=0 – упругое рассеяние или неупругое взаимодействие определяется при обращении CALL PREPAR. В результате флаг KSTATE приобретает значение 1 (упругое) или 2 (неупругое), а также разыгрывается ядро-мишень NUCLID. Кроме того возможно значение KSTATE=6 при захвате π^- и K^- на водороде, что ведет к обращению CALL DECAY.

При упругом рассеянии (KSTATE=1) вызывается соответствующая программа (CALL ELSCAT) и результаты рассеяния пересылаются в дерево (CALL SEND).

При неупругом взаимодействии (KSTATE=2) вызывается генератор ядерных реакций MSDM (CALL REACT) и результаты взаимодействия пересылаются в дерево (CALL SEND).

После пересылки продуктов распадов, упругого рассеяния или ядерных реакций в дерево межъядерного каскада (после обращения CALL SEND) управление передается на начало новой ветви (CALL TAKPAR).

Чтобы не загромождать изложение и блок схему, здесь опущены проверки на переполнение массивов и другие ненормальные ситуации. Обычно в таких случаях управление передается на метку 1 и данное каскадное дерево моделируется заново. При этом выдается соответствующая диагностика.

4. Управляющие переменные

При моделировании очередного дерева используются глобальные управляющие переменные, которые находятся в COMMON блоке:

```
COMMON /MANALL/ NTREE,LINE1,LINE2,NBRMAX,NPRMAX,NNEUTR,NGAMMA,
& MELSC,MINTER,MCUT,MOU,MABSOR,MDECAY,NSTRAN,
& NEMIN,NEPLS,NNUEL,NANUE,NNUMU,NANUM
```

NTREE – уникальный номер дерева;

LINE1 – номер строки массивов TREE и ITREE начиная с которой необходимо просматривать эти массивы чтобы взять очередную частицу, которая инициирует следующую ветвь дерева;

LINE2 – номер строки массивов TREE и ITREE начиная с которой записывается следующая ветвь дерева;

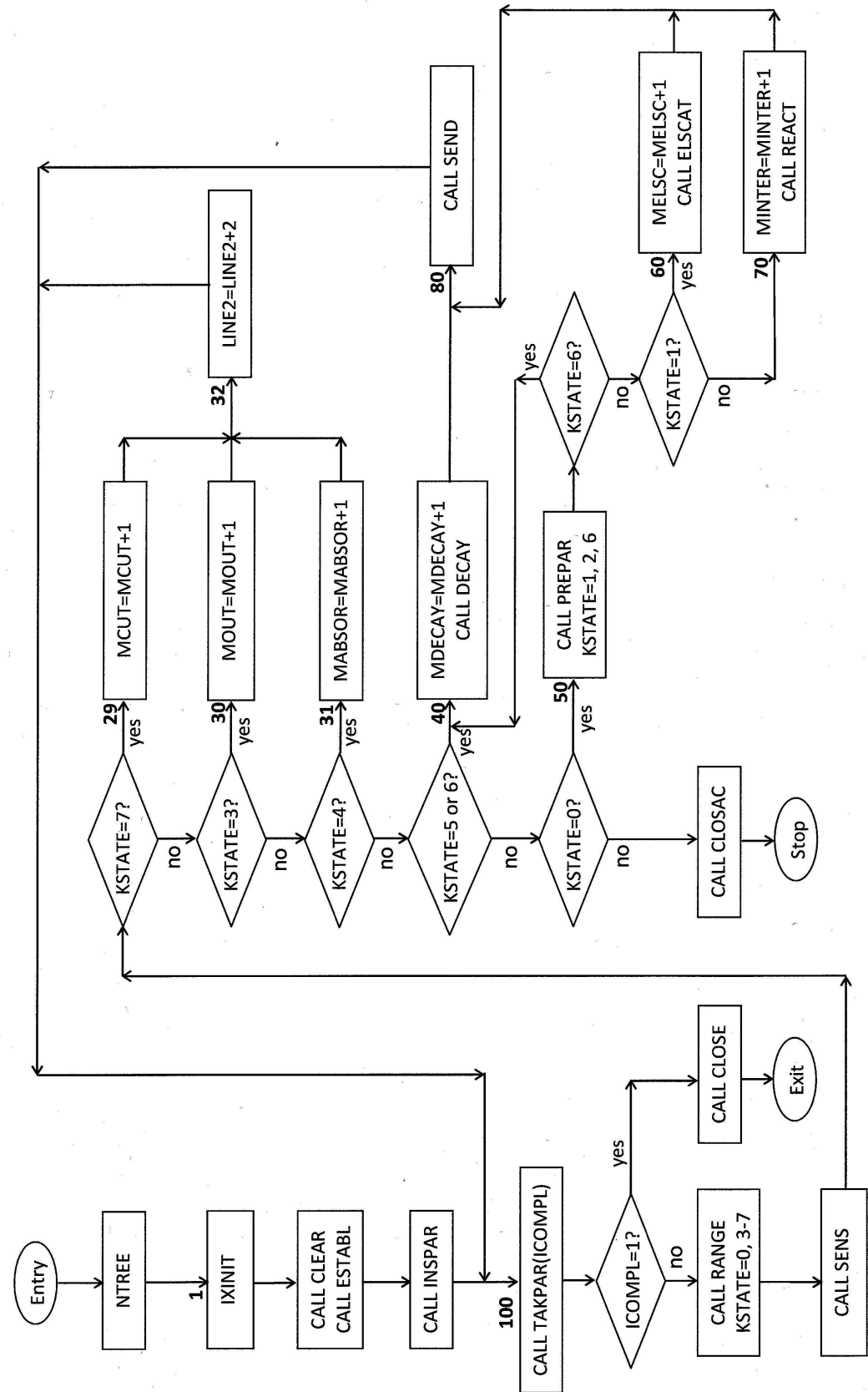


Fig. 1. Flowchart of SUBROUTINE CENTRE

NBRMAX – максимальный номер ветви, существующей в дереве на данный момент;

NPRMAX – максимальное число частиц в дереве на данный момент (не используется, резервировано);

NNEUTR – номер строки в массиве источника нейтронов в которую должен быть записан очередной нейтрон;

NGAMMA – номер строки в массиве источника гамма-квантов в которую должен быть записан очередной гамма-квант;

MELSC – счетчик событий упругого рассеяния в дереве;

MINTER – счетчик событий неупругого взаимодействия в дереве;

MCUT – счетчик событий, когда частица попала под обрезание;

MOUT – счетчик событий, когда частица вылетела из мишени;

MABSOR – счетчик событий, когда частица остановилась из за ионизационных потерь;

MDECAY – счетчик событий распада в дереве;

NSTRAN – номер первой свободной строки в массивах TRAN и ITRAN;

NEMIN – номер строки в массиве источника электронов, в которую должен быть записан очередной электрон;

NEPLS – номер строки в массиве источника позитронов, в которую должен быть записан очередной позитрон;

NNUEL – номер строки в массиве источника электронных нейтрино, в которую должно быть записано очередное электронное нейтрино;

NANUE – номер строки в массиве источника электронных антинейтрино, в которую должно быть записано очередное электронное антинейтрино;

NNUMU – номер строки в массиве источника мюонных нейтрино, в которую должно быть записано очередное мюонное нейтрино;

NANUM – номер строки в массиве источника мюонных антинейтрино, в которую должно быть записано очередное мюонное антинейтрино.

При пересылке программой SEND продуктов неупругого взаимодействия, а также продуктов упругого рассеяния и распада, из массивов SPT и SNU в массивы дерева и источников, используются локальные управляющие переменные, которые находятся в COMMON блоке:

```
COMMON/MANLOC/ KPRINU,KSECNU,KCASC,KNEUTR,KGAMMA,
&              KEMIN,KEPLS,KNUEL,KANUE,KNUMU,KANUM
```

Подчеркнем, что локальные переменные относятся к данному взаимодействию или распаду, а не ко всему дереву в целом, но используются программой SEND при модификации глобальных управляющих переменных в COMMON блоке /MANALL/.

KPRINU:
 KPRINU=1 – имеет место взаимодействие, т.е. присутствует ядро мишень;
 KPRINU=0 – ядро мишень отсутствует (в случае распада);

KSECNU – число ядер продуктов;

KCASC – число каскадных частиц;

KNEUTR – число нейтронов с энергией ниже 14.5 МэВ;

KGAMMA – число гамма квантов;

KEMIN – число электронов

KEPLS – число позитронов

KNUEL – число электронных нейтрино

KANUE – число электронных антинейтрино

KNUMU – число мюонных нейтрино

KANUM – число мюонных антинейтрино

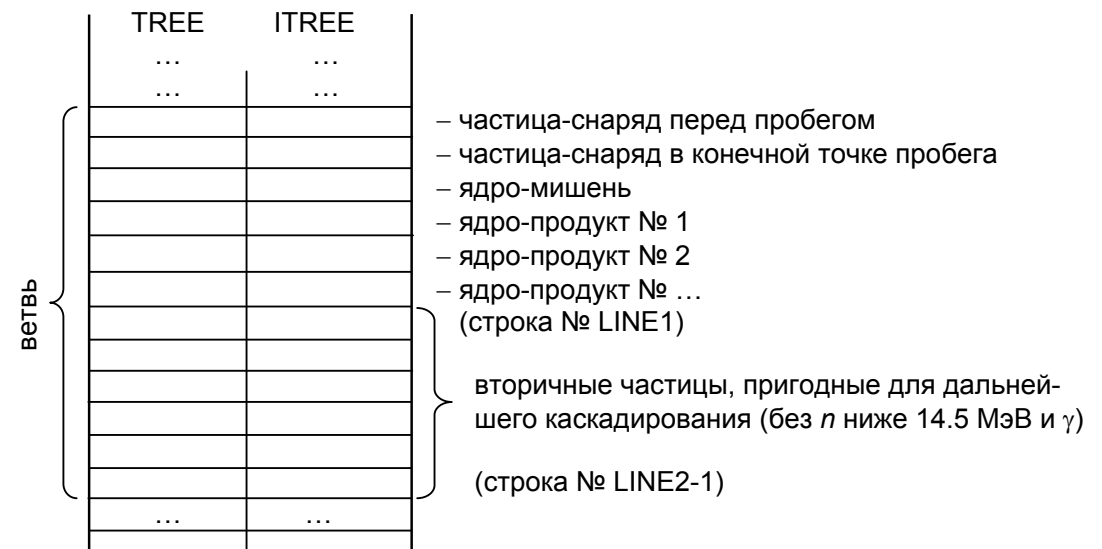
5. Структура массивов дерева TREE и ITREE

Дерево межъядерного каскада запоминается в двух двумерных массивах TREE и ITREE в COMMON блоке

```
COMMON /TREE/ TREE(15,30000),ITREE(7,30000),LTR13,LTR7,LTR300
```

(целые переменные LTR13, LTR7, LTR300 служат для изменения размерностей массивов при необходимости, например при повышении энергии снаряда, см. SUBROUTINE DIMAR).

Каждая строка этих массивов содержит вещественные и целые переменные, как частиц, так и ядер. Дерево запоминается в виде последовательности ветвей, в порядке возрастания номеров ветвей:



Первоначально, после пересылки продуктов реакции в дерево, ветвь содержит снаряд, ядро-мишень и ядра продукты, а также вторичные частицы, как показано на схеме. Однако вторичные частицы не остаются в своих строках по мере развития каскада. Каждая ветвь просматривается, начиная со строки № LINE1, и частица из этой строки перемещается в строку № LINE2, чтобы инициировать новую ветвь. Затем все строки данной ветви, начиная со строки № LINE1+1, сдвигаются вверх на одну строку и управляющие переменные LINE1 и LINE2 переопределяются.

Извлеченная частица в строке № LINE2 инициирует новую ветвь, в которой все вторичные частицы опять записаны. Между тем сканирование предыдущей ветви продолжается до тех пор, пока все оставшиеся в этой ветви вторичные частицы не будут перемещены и не инициируют новые ветви. Затем сканированию подвергается следующая ветвь.

Извлечение частиц из ветви и сдвиг в массивах дерева осуществляется программой SUBROUTINE TAKPAR(ICOMPL).

Таким образом, когда дерево завершено, каждая ветвь состоит из следующих строк:

- частица перед пробегом;
- частица в конце пробега;
- ядро-мишень;
- ядра-продукты в покое (не преобразованные в транспортируемые частицы).

Если частица вылетела из мишени, поглотилась, или попала под обрезание, то ветвь состоит из двух строк:

- частица перед пробегом;
- частица в конце пробега, в точке вылета, поглощения или обрезания.

Тем не менее, каждая частица в дереве имеет индикаторы, которые позволяют определить, к какой ветви принадлежит данная частица, кто инициировал эту ветвь, какой тип взаимодействия и с каким ядром имел место, номер поколения, и так далее, т.е. дерево записывается без потери физической информации.

Массивы дерева содержат две дополнительные строки – первую и последнюю. Первые строки массивов TREE, ITREE содержат счетчики и управляющие переменные. Последняя строка массива ITREE содержит указатель “end of TREE”. Последняя строка массива TREE не используется (нулевая). Первая и последняя строки заполняются программой SUBROUTINE CLOSE(JABNOR).

Первая строка массива TREE содержит счетчики:

- TREE(1,1)=FLOAT(MELSC) число упругих рассеяний в дереве;
- TREE(2,1)=FLOAT(MINTER) число неупругих взаимодействий в дереве;
- TREE(3,1)=FLOAT(MCUT) число случаев «cut off» в дереве;
- TREE(4,1)=FLOAT(MOUT) число вылетов из мишени в дереве; \
- TREE(5,1)=FLOAT(MABSOR) число поглощений в дереве;
- TREE(6,1)=FLOAT(MDECAY) число распадов в дереве;
- TREE(7,1)+TREE(1,15) 0.0

Первая строка массива ITREE содержит управляющие переменные:

- ITREE(1,1)=0
- ITREE(2,1)=NTREE номер дерева;

- ITREE(3,1)=LINE2-2 число строк в массивах TREE, ITREE;
- ITREE(4,1)=NBRMAX число ветвей в дереве;
- ITREE(5,1)=NPRMAX не используется;
- ITREE(6,1)=NNEUTR-2 число нейтронов в массиве источника нейтронов;
- ITREE(7,1)=NGAMMA-2 0, число γ -квантов в массиве источника γ -квантов;

Последняя строка массива ITREE содержит указатель “end of TREE”:

- ITREE(2,LINE2)=-NTREE
- ITREE(3,LINE2)=JABNOR (0 – нормальное закрытие массивов дерева, 1 – аварийное закрытие)

Остальные пять ячеек в строке LINE2 не используются (нулевые).

6. Параметры частиц и ядер в массивах дерева TREE и ITREE

COMMON /TREE/ TREE(15,30000),ITREE(7,30000),LTR13,LTR7,LTR300

6.1. Параметры частиц:

- TREE(1,*) x(см)
- TREE(2,*) y(см)
- TREE(3,*) z(см)
- TREE(4,*) cos θ
- TREE(5,*) sin φ
- TREE(6,*) cos φ
- TREE(7,*) T (кинетическая энергия: МэВ для адронов, МэВ/А для ядер)
- TREE(8,*) W (статистический вес)
- TREE(9,*) ZONE (номер зоны в мишени)
- TREE(10,*) KOL (флаг состояния K^0 мезона: KOL=1 если $K^0 = K^0_L$; KOL=0 если $K^0 = K^0_S$)
- TREE(11,*) KOS (флаг состояния K^0 мезона: KOS=1 если $K^0 = K^0_S$; KOS=0 если $K^0 = K^0_L$)
- TREE(12,*) t_0 (время старта адронного каскада, не используется)
- TREE(13,*) t (текущее время адронного каскада, не используется)
- TREE(14,*) A иона (если транспортируется ядерный фрагмент)
- TREE(15,*) Z иона (если транспортируется ядерный фрагмент)
- ITREE(1,*) JPART – тип частицы: 1 - нейтрон, 2 - протон, ... 25 - ион (A,Z)
- ITREE(2,*) NTREE – уникальный номер дерева, которому принадлежит частица
- ITREE(3,*) NBRAN – уникальный номер ветви дерева, которой принадлежит инициирующая частица
- ITREE(4,*) NUNPAR – номер ветви, из которой была извлечена частица, инициирующая ветвь № NBRAN
- ITREE(5,*) KGENER – номер поколения частицы (NOTE: упругое рассеяние не увеличивает номер поколения частицы)
- ITREE(6,*) KPASS – флаг частицы относительно пробега:
 - KPASS=-1 частица еще не извлекалась для транспортировки
 - KPASS=0 частица перед розыгрышем пробега
 - KPASS=+1 частица в конечной точке пробега
- ITREE(7,*) KSTATE – состояние частицы в конце пробега:
 - 0 – упругое рассеяние или неупругое взаимодействие

- 1 – упругое рассеяние
- 2 – неупругое взаимодействие
- 3 – вылет за пределы мишени
- 4 – поглощение из-за ионизационных потерь
- 5 – распад на лету
- 6 – распад в покое
- 7 – энергия частицы попала под обрезание

6.2. Параметры ядер-мишеней и ядер-продуктов

- TREE(1,*) x(см)
- TREE(2,*) y(см)
- TREE(3,*) z(см)
- TREE(4,*) A
- TREE(5,*) Z
- TREE(6,*) U* - энергия возбуждения (МэВ)
- TREE(7,*) не используется
- TREE(8,*) px
- TREE(9,*) py импульс (в дереве не используется; внутренняя переменная)
- TREE(10,*) pz
- TREE(11,*) Mx
- TREE(12,*) My момент импульса (резервирован, но не используется)
- TREE(13,*) Mz
- TREE(14,*) не используется
- TREE(15,*) не используется
- ITREE(1,*) JPART=0 – для ядра всегда ноль, в отличие от частиц
- ITREE(2,*) NTREE – уникальный номер дерева, которому принадлежит ядро
- ITREE(3,*) NBRAN – уникальный номер ветви дерева, которой принадлежит ядро
- ITREE(4,*) не используется
- ITREE(5,*) KGENUC – номер поколения ядра-продукта (NOTE: этот номер на 1 больше, чем номер поколения налетающей частицы, кроме актов упругого рассеяния)
- ITREE(6,*) KBEAFT – флаг ядра (ядро-мишень или ядро-продукт):
 - KBEAFT=0 для ядра-мишени
 - KBEAFT=1 для ядра-продукта
- ITREE(7,*) KINTER – тип взаимодействия, приведшее к данному ядру-продукту:
 - 1 – упругое рассеяние
 - 2 – неупругое взаимодействие
 - 3 – деление
 - 4 – Ферми-развал (резервировано, но не используется)
 - 5 – мультифрагментация (резервировано, но не используется)
 - 6 – полный развал ядра

7. Массивы источников

В массивы источников частиц пересылаются частицы, перенос которых не моделируется в адронном каскаде кода SHIELD. Пересылка производится программой SEND. К таким частицам относятся:

- (1) нейтроны с энергией ниже 14.5 МэВ, рожденные в адронном каскаде, а также продукты распада мезонов: (2) гамма-кванты, (3) электроны, (4) позитроны, (5) электронное нейтрино, (6) электронное антинейтрино, (7) мюонное нейтрино, (8) мюонное антинейтрино.

Эти источники находятся, соответственно, в COMMON блоках:

```
COMMON /SOURN/ SOURN(13,5000),ISOURN(7,5000),LSN13,LSN7,LSN100
COMMON /SOURG/ SOURG(13,1000),ISOURG(7,1000),LSG13,LSG7,LSG100
COMMON /SEMIN/ SEMIN(13,100),ISEMIN(7,100),L1313,L137,L13100
COMMON /SEPLS/ SEPLS(13,100),ISEPLS(7,100),L1413,L147,L14100
COMMON /SNUEL/ SNUEL(13,100),ISNUEL(7,100),L1713,L177,L17100
COMMON /SANUE/ SANUE(13,100),ISANUE(7,100),L1813,L187,L18100
COMMON /SNUMU/ SNUMU(13,100),ISNUMU(7,100),L1913,L197,L19100
COMMON /SANUM/ SANUM(13,100),ISANUM(7,100),L2013,L207,L20100
```

Перенос нейтронов с энергией ниже 14.5 МэВ далее моделируется программой LOENT, которая является частью кода SHIELD. Для переноса остальных частиц надо привлекать внешние программы.

В отличие от массивов дерева TREE и ITREE, массива источников устроены просто, как стеки FIFO. Ветви не используются. Каждая рожденная частица занимает одну строку в массиве источника. Например, все «испарительные» нейтроны, образовавшиеся в межъядерном каскаде, заносятся, в порядке их появления, в массивы SOURN и ISOURN. Затем программа LOENT извлекает их по одному и транспортирует через мишень.

Размещение параметров частиц в массивах источников аналогично размещению параметров частиц в массиве дерева TREE. Например, в строках массива SOURN(13,5000), параметры нейтронов с энергией ниже 14.5 МэВ расположены так:

```
SOURN(1,*)÷SOURN(3,*) – x,y,z (см)
SOURN(4,*)÷SOURN(6,*) – cos θ, sin φ, cos φ
SOURN(7,*) – энергия нейтрона (МэВ)
SOURN(8,*) – W (статистический вес)
SOURN(9,*) – номер зоны мишени
SOURN(10,*) – момент времени рождения нейтрона t0
SOURN(11,*) – текущее время t (мкс)
SOURN(12,*), SOURN(13,*) – не используются
```

Строки массива ISOURN заполняются аналогично строкам массива ITREE для частиц. Этот массив позволяет определить, в какой ветви адронного каскада был рожден данный нейтрон, и отсюда извлечь, если надо, информацию поколения, родителе и др.

Массивы источников, так же, как и массивы дерева, имеют две дополнительные строки, первую и последнюю. При закрытии массивов источников эти строки заполняются так же, как и аналогичные строки в массивах дерева.

8. Запоминание пробогов через зоны мишени по шагам

При розыгрыше пробега частицы программой RANGE, информация о шагах пробега через зоны мишени запоминается в массиве STEPR, в COMMON блоке

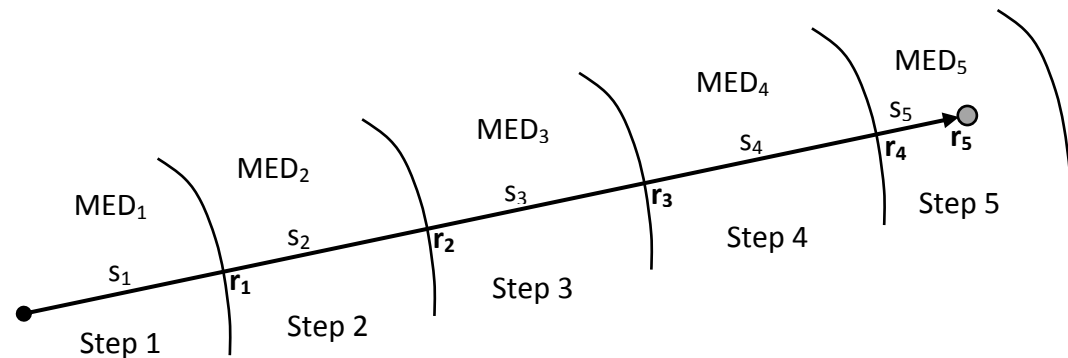
```
COMMON /STERAN/ STEPR(9,500),LST9,LST50.
```

По завершении моделирования пробега данной частицы, информация из массива STEPR пересылается в массивы TRAN и ITRAN COMMON блока

```
COMMON /TRAN/ TRAN(5,20000),ITRAN(6,20000),LSR5,LSR6,LSR500.
```

Таким образом, в массивах TRAN и ITRAN содержится информация о всех шагах пробегов всех частиц в данном дереве. Это позволяет вычислять энергвыделение во всех зонах мишени, а также проводить оценку по пробегам (Track Length Estimation) потоков частиц в зонах мишени. Пересылка данных из массива STEPR в массивы TRAN и ITRAN осуществляется программой SUBROUTINE SENS.

Размещение информации в массиве STEPR поясняется рисунком и таблицей ниже. Заметим, что запоминаются только точки входа в зону и выхода из зоны. Траектория внутри зоны может немного отличаться от прямой линии из-за многократного рассеяния.



| Array STEPR(cell,line) | | | | | | |
|------------------------|---------------|-------------|--------|--------|-------------------|-------------------|
| cell 1 | cell 2 | cell 3 | cell 4 | cell 5 | cell 6÷8 | cell 9 |
| No. of step | No. of media | No. of zone | s_k | JPART | r_k | T_k |
| 0 | $ISTEP_{max}$ | - | - | JPART | $r_0 = r_{init}$ | $T_0 = T_{init}$ |
| 1 | MED_1 | $ZONE_1$ | s_1 | JPART | r_1 | T_1 |
| ... | ... | ... | ... | ... | ... | ... |
| 5 | MED_5 | $ZONE_5$ | s_5 | JPART | $r_5 = r_{final}$ | $T_5 = T_{final}$ |

Строки массива STEPR, включая строку шага № 0, пересылаются в строки массивов TRAN и ITRAN, начиная со строки № NSTRAN этих массивов, как показано ниже. После пересылки управляющая переменная NSTRAN модифицируется программой SENS.

```
TRAN(1,*)=STEPR(4,**)
TRAN(2,*)=STEPR(6,**)
TRAN(3,*)=STEPR(7,**)
TRAN(4,*)=STEPR(8,**)
TRAN(5,*)=STEPR(9,**)
```

```
ITRAN(1,*)=ITREE(1,LINE2)
ITRAN(2,*)=ITREE(2,LINE2)
ITRAN(3,*)=ITREE(3,LINE2)
ITRAN(4,*)=INT(STEPR(1,**))
ITRAN(5,*)=INT(STEPR(2,**))
ITRAN(6,*)=INT(STEPR(3,**))
```

В результате, шаги пробега в массивах TRAN и ITRAN размещены следующим образом:

| Array TRAN(cell,line) | | | Array ITRAN(cell,line) | | | | | |
|-----------------------|-----------|--------|------------------------|-------------|---------------|-------------|---------------|-------------|
| cell 1 | cells 2÷4 | cell 5 | cell 1 | cell 2 | cell 3 | cell 4 | cell 5 | cell 6 |
| s_k | r_k | T_k | JPART | No. of TREE | No. of branch | No. of step | No. of medium | No. of zone |

Рассмотрим подробнее структуру массива STEPR. В первой строке массива STEPR (шаг № 0) находится частица в состоянии перед пробегом. Далее идут шаги до пересечения поверхностей, разграничивающих зоны мишени и последний шаг до точки взаимодействия (вылета/поглощения/распада).

Формат шага № 0 отличается от остальных шагов. Эта строка является управляющей.

```
STEPR(1,1) - 0.0 (№ шага, real*8)
STEPR(2,1) - число шагов в пробеге, включая шаг № 0
STEPR(3,1) - № зоны в которой находится начальная точка пробега
STEPR(4,1) - 0.0 всегда (длина нулевого шага,  $s_0=0.0$ )
STEPR(5,1) - JPART (идентификатор частицы)
STEPR(6,1)÷STEPR(8,1) - x, y, z координаты начальной точки (см)
STEPR(9,1) - энергия частицы в начальной точке (МэВ или МеВ/А)
```

Массивы TRAN и ITRAN, подобно массивам дерева TREE и ITREE, имеют две дополнительные строки, первую и последнюю, для размещения управляющих переменных. В массиве TRAN эти строки не используются (нулевые). В массиве ITRAN эти строки содержат следующие данные:

Первая строка:

```
ITRAN(1,1) - 0
ITRAN(2,1)=NNTREE           номер дерева
ITRAN(3,1)=NSTRAN-2        полное число шагов во всех пробегах в дереве
ITRAN(4,1)=NBRMAX          полное число ветвей в дереве
ITRAN(5,1) - 0
ITRAN(6,1) - 0
```

Последняя строка:

```
ITRAN(2,NSTRAN)=-NNTREE
ITRAN(3,NSTRAN)=JABNOR (0-нормальное закрытие, 1-аварийное закрытие)
Остальные четыре ячейки в строке NSTRAN не используются (нулевые).
```