

Министерство образования и науки Российской Федерации

Национальный исследовательский ядерный  
университет «МИФИ»

**М.И. Гуревич, Д.А. Шкаровский**

**Расчет переноса нейтронов  
методом Монте-Карло  
по программе MCSU**

*Рекомендовано УМО «Ядерные физика и технологии»  
в качестве учебного пособия  
для студентов высших учебных заведений*

Москва 2012

УДК 539.18:519.245(075)+621.039(075)

ББК 22.383я7+31.4я7

Г 95

*Гуревич М.И., Шкаровский Д.А.* **Расчет переноса нейтронов методом Монте-Карло по программе MSU: Учебное пособие.** М.: НИЯУ МИФИ, 2012. – 156 с.

Пособие дает необходимый минимум информации для расчёта переноса нейтральных частиц методом Монте-Карло по универсальной отечественной программе MSU, предназначенной для расчёта любых ядерно-энергетических установок. Существуют свободно распространяемые версии, программа снабжена полной документацией.

Позволяет начать работать с программой и может служить введением в документацию. Кратко изложены физические основы и принцип моделирования. Кроме того, поскольку подобные программы обладают универсальной структурой, ими можно воспользоваться и как введением в работу с любой другой универсальной программой расчёта переноса частиц, основанной на методе Монте-Карло.

Основано на курсе лекций, которые читались в НИЯУ МИФИ и МФТИ, предназначено для студентов, аспирантов и специалистов, осваивающих расчёты методом Монте-Карло.

Подготовлено в рамках Программы создания и развития НИЯУ МИФИ.

*Рецензенты: проф. МАИ, канд.техн.наук Е.С.Глушков;  
проф. МФТИ, д-р физ.-мат.наук Р.Т.Исламов*

ISBN 978-5-7262-1762-8

© Национальный исследовательский ядерный университет «МИФИ», 2012

## СОДЕРЖАНИЕ

Введение .....	4
Глава 1. Оцененные ядерные данные .....	7
1.1. Экспериментальные данные .....	7
1.2. Оцененные данные .....	10
1.3. Использование оцененных данных .....	13
1.4. Бенчмарк-эксперименты .....	14
Глава 2. Методы расчетов переноса излучения .....	15
Глава 3. Метод Монте-Карло .....	19
Глава 4. Общая схема моделирования методом Монте-Карло .....	24
Глава 5. Программы, основанные на методе Монте-Карло .....	31
Глава 6. Проект MCU .....	32
Глава 7. Установка, генерация и запуск на счет .....	34
Глава 8. Общая схема расчета .....	39
8.1. Шаг INPUT .....	39
8.2. Шаг CALCULATION .....	40
8.3. Шаг OUTPUT .....	40
8.4. Шаг BURNUP .....	41
Глава 9. Режим многопроцессорных вычислений .....	41
Глава 10. MCU Office .....	43
Глава 11. Исходные данные для физического модуля .....	51
11.1. Метод суммарного изотопа .....	66
Глава 12. Исходные данные для геометрического модуля .....	71
Глава 13. Задание многоуровневой геометрии .....	89
Глава 14. Исходные данные для модуля источника .....	111
14.1. Точечный источник .....	112
14.2. Сложный источник без смещений распределений .....	114
14.3. Смещение распределений .....	127
Глава 15. Исходные данные для модуля регистрации .....	129
Глава 16. Исходные данные для транспортного модуля .....	138
Глава 17. Исходные данные для управляющего модуля .....	140
Глава 18. Пример исходных данных для расчета .....	141
Список использованной литературы .....	153

## ВВЕДЕНИЕ

Под расчётом переноса излучения понимают расчёты потоков нейтральных частиц (нейтронов и фотонов) или, что почти то же самое, расчёты распределения плотности этих частиц. Область энергии нейтронов и фотонов такова, что в любой момент времени частицы имеют точно определенные положение в пространстве, энергию и направление полета, что полностью их характеризует. Волновые свойства частиц не рассматриваются.

Поток или плотность определены в фазовом пространстве, т.е. существует зависимость от точки в пространстве, энергии частицы и направления полета. Для ряда задач достаточно упрощенных распределений, например, усреднений по направлению полета, которые зависят лишь от геометрических координат и энергии.

Будут использоваться два класса объектов: вещества или материальные среды и свободно летящие частицы.

Материальные среды или вещества описываются лишь в той мере, которая нужна для учёта их взаимодействия с нейтральными частицами. Для большого класса задач вещество можно рассматривать как совокупность ядер. Для сравнительно низких энергий такой параметр, как температура, является существенным. Для энергий, сравнимых с тепловой энергией молекул, существенна также химическая структура вещества.

Описываемые методы расчёта касаются только стационарных процессов, то есть таких процессов, для которых распределение частиц, а также характеристики веществ не зависят от времени.

Наиболее важное свойство применяемых моделей – то, что частицы между собой не взаимодействуют, это следует из отсутствия электрического заряда и малого числа частиц по сравнению с числом ядер в веществе.

Кроме потоков, большой интерес представляют скорости поглощения нейтронов ядрами определенного нуклида (конкретного изотопа определенного химического элемента). Из общей физики известно, что скорость ядерной реакции в элементе фазового объема выражается формулой  $\Sigma_n(\mathbf{r}, E)\psi(\mathbf{r}, \Omega, E)$ , где  $\psi$  – поток, зависящий от всех координат фазового пространства. Коэффициент пропорциональности для нейтронов и  $\gamma$ -квантов  $\Sigma_n$  не зависит от направ-

ления полета. Это коренным образом отличается от взаимодействия вещества с видимым светом, где очень существенна анизотропия оптических свойств. Данный коэффициент называется *макроскопическим сечением данной реакции*.

Особенно важную роль играет полное макроскопическое сечение – сумма сечений всех возможных взаимодействий с ядрами всех нуклидов в веществе. Оно обозначается  $\Sigma_t$ . Выражение  $\Sigma_t \nu$  имеет ясный физический смысл – число любых взаимодействий (столкновений) частиц с ядрами в элементе фазового объема в секунду.

Важную роль может играть макроскопическое сечение поглощения  $\Sigma_a$  – сумма сечений всех реакций, в которых частица исчезает.

Таким образом, если значения потока и сечений известны с хорошей точностью, то можно определить и скорости реакций. В ряде методов расчета, например в методе Монте-Карло, удобнее сразу вычислять скорости реакций одновременно с расчётом потоков.

Следует также сказать, что при анализе результатов расчётов человек не использует огромную массу чисел – значения потоков в большом наборе точек, для разных энергий и т.п. Он оперирует относительно небольшим набором величин, получающихся усреднением потоков или скоростей реакций по тем или иным пространственно-энергетическим областям, например по отдельным деталям конструкции. Все эти величины теоретически являются интегралами потока с различными весами и поэтому называются функционалами потока или просто функционалами.

Программы расчёта не могут вычислять потоки в бесконечном числе точек, многие из них выдают лишь заранее запрошенные функционалы.

При расчётах переноса излучения нейтральных частиц возникают две задачи, имеющие разный физический смысл и свои особенности при решении.

В так называемых неоднородных задачах существует независимый от потока частиц «внешний» источник. Например, имеется источник  $\gamma$ -излучения и надо определить потоки фотонов и уровень их безопасности для людей.

Другой пример – термоядерная установка, представляющая собой источник не только  $\gamma$ -квантов, но и нейтронов. При наличии

бланкета, кроме этих первичных нейтронов, возникают новые нейтроны в бланкете. Однако нейтроны в бланкете зависят от существующего потока нейтронов, так как возникают в результате их поглощения и последующего деления ядер. Генерация термоядерных нейтронов, напротив, не зависит от потока, значит, это и есть источник.

В однородных задачах, называемых также задачами на  $k_{eff}$ , никакого внешнего источника не существует. Поток нейтронов должен поддерживать себя сам. Типичной задачей этого типа является расчёт параметров работы ядерного реактора, в котором происходит самоподдерживающаяся цепная реакция.

К тому же классу относятся задачи, возникающие при оценке безопасности хранилищ ядерного топлива или контейнеров для его перевозки.

## **ГЛАВА 1. ОЦЕНЕННЫЕ ЯДЕРНЫЕ ДАННЫЕ**

Расчёт реальных объектов любым методом опирается на константы, характеризующие свойства веществ, из которых состоит объект.

Для расчёта потоков нейтронов – это характеристики взаимодействия нейтронов с ядрами различных нуклидов. Такие взаимодействия зависят от энергии нейтрона, а в ряде случаев и от температуры веществ. Для получения числовых значений соответствующих констант необходимо сочетание обработки экспериментальных данных и методов теоретической ядерной физики.

При проведении расчётов обычно пользуются готовыми библиотеками констант, и необходимо иметь некоторое представление об этих библиотеках.

### **1.1. ЭКСПЕРИМЕНТАЛЬНЫЕ ДАННЫЕ**

Основная масса данных о взаимодействии нейтронов с ядрами получена экспериментально в середине прошлого века, а точнее, в период с 1950 по 1970 г., когда было проведено большое количество разнообразных экспериментов. Это были эксперименты на ускорителях заряженных частиц, позволяющие получать нейтроны в результате ядерных реакций на мишени ускорителя, нейтронных пучках, выведенных из активных зон реакторов. Кроме того, практиковались облучение материалов в самих реакторах и даже использование нейтронов ядерных взрывов.

По мере увеличения количества выполненных экспериментов росло и количество полученных данных, что вызвало необходимость их упорядочения и централизованного хранения. С этой целью в 1951 году в Брукхейвенской национальной лаборатории (Brookhaven National Laboratory, BNL) начался проект по созданию базы данных по нейтронным сечениям [1]. Хранение данных осуществлялось на перфокартах, и к 1962 году уже было собрано полмиллиона IBM перфокарт и дополнительно ожидалось поступление по 200 000 перфокарт ежегодно.

С развитием компьютерных технологий появилась возможность отказаться от использования перфокарт и перенести все накопленные данные на магнитную ленту. Для этого в 1964 году в рамках этого проекта была создана система «Sigma Center Information Stor-

age and Retrieval System» (SCISRS). Эта система была предназначена для хранения данных на магнитной ленте. Для обеспечения работы с данными использовалась программа для IBM 7094. Тот факт, что программа была написана на машинном языке, оказался серьезным недостатком системы, поскольку в результате появления новых более мощных машин, а также дата-центров в других организациях, которые хотели использовать SCISRS на своих компьютерах, выяснилось, что простого переноса данных с одних лент на другие недостаточно. Необходимо было полностью переписывать программу. Кроме того, в процессе эксплуатации выявились проблемы, связанные с нерациональностью хранения данных (некоторые данные повторялись, что увеличивало расход ленты) и недостаточным количеством информации, пригодной для обеспечения программной обработки данных. Поэтому уже в середине 1960-х годов начинаются работы по модернизации системы, и в 1971 году создается новая система «Cross Section Information Storage and Retrieval System or» (CSISRS) с новой программой и руководством пользователя. Эта система обслуживала базу данных более чем из миллиона точек. При этом ожидалось, что скорость пополнения базы составит по 120 000 точек ежегодно.

В СССР аналогичные работы велись в Физико-энергетическом институте, где в 1963 году был создан Центр ядерных данных. Основной задачей центра был сбор информации по советским лабораториям.

Для сбора информации по лабораториям Европы и Японии в 1964 году было создано Европейское агентство по ядерной энергии. Сейчас это NEA Data Bank, agency of the Organization for Economic Development (OECD). Для обработки и хранения данных агентство использовало собственную программу – Neutron Data Direct Access System (NEUDADA).

Для упрощения процедуры обмена данными в общемировом масштабе в том же 1964 году при МАГАТЭ создается комиссия по ядерным данным. Помимо инициации и обеспечения систематического международного обмена данными, комиссия должна была также способствовать созданию центров по сбору информации в тех странах, которые не принимают участия в таком обмене. Для работы с поступающими данными использовалась программа Data Storage And Retrieval System (DASTAR).



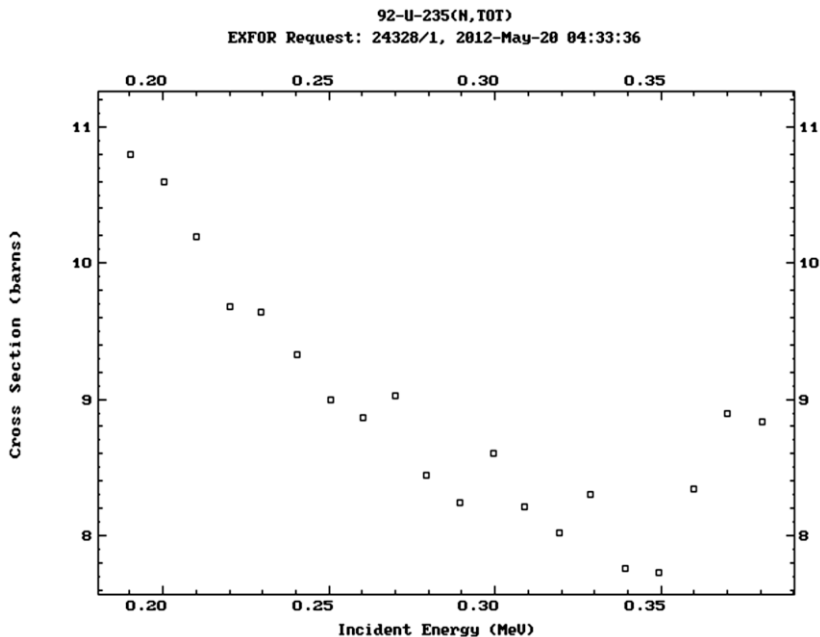


Рис. 1. Пример из он-лайн библиотеки EXFOR

В 1966 году была достигнута договоренность о ежегодных встречах всех четырех центров в МАГАТЭ. Один из обсуждавшихся вопросов – формат представления данных для стандартизации и упрощения обмена данными между участниками. В результате в 1969 году были зафиксированы основные требования к формату, получившему название EXFOR – сокращение от EXchange FORmat. В 1970 году начался обмен данными в новом формате. Этот процесс состоял из двух частей. Новые экспериментальные данные сразу кодировались в соответствии с требованиями EXFOR, а для перевода данных из других форматов создавались специальные программы. Через несколько лет объединенные усилия участников проекта привели к созданию международной библиотеки экспериментальных данных по нейтронным реакциям EXFOR. В 1988 году к библиотеке открыт доступ по сети Интернет.

## 1.2. ОЦЕНЕННЫЕ ДАННЫЕ

Вначале для пользователей было достаточно просто наличия экспериментальных данных. Однако постепенно, с накоплением всё большего количества информации, возник интерес к оцененным данным, вызванный тем, что в библиотеку EXFOR поступали данные экспериментов различного качества. М.Н. Николаев, один из создателей всемирно известной системы констант БНАБ [2] и руководитель проекта по созданию библиотеки РОСФОНД писал: «Экспериментальные данные достаточно разбросаны, нередко противоречивы и всегда неполны» [3]. Таким образом, появилась необходимость в сортировке, а в ряде случаев и некоторой коррекции данных. Эти специально отобранные, проверенные, а зачастую скорректированные данные и стали называть *оцененными*.

В сущности, для практического использования через специально отобранные экспериментальные точки необходимо провести плавные кривые. М.Н. Николаев считал: чтобы оценить нейтронные данные для одного нуклида, требуется от нескольких недель до 2–3 человеколет высококвалифицированного труда.

Такие значительные сроки обусловлены необходимостью подробного исследования не только собственно экспериментальных данных, но и истории проведения эксперимента, поскольку до того, как приступить к анализу собственно данных, необходимо выяснить, насколько этим данным можно доверять, что в свою очередь означает необходимость поиска и анализа соответствующих отчетов, статей и любых других источников информации об экспериментах. В настоящее время эта задача усложняется еще и тем, что со времени получения основной массы экспериментальных данных прошло уже более сорока лет.

После отбора экспериментальных данных необходимо подобрать кривые, которые бы наиболее достоверно описывали зависимость сечения реакции от энергии. Для этого используются различные формализмы расчёта, однако основными являются формализмы Брейта–Вигнера, Адлер–Адлера, и Рейха–Мура, причем в последнее время наибольшее распространение получил формализм Рейха–Мура. При использовании формализмов их параметры подбирают таким образом, чтобы в рамках соответствующего способа расчёта сечений обеспечивалась возможность правильно восстановить экспериментальную зависимость.

Для формализмов Брейта–Вигнера и Адлер–Адлера можно достаточно быстро рассчитывать сечения нуклидов для произвольной температуры непосредственно по резонансным параметрам, в то время как формализм Райха–Мура позволяет «продлить» область разрешенных резонансов.

Часто в процессе этой работы оценщик сталкивается с полным отсутствием требуемой информации. В таком случае он принимает решения, исходя из собственного опыта и интуиции, поэтому оцененные данные, вообще говоря, всегда несут в себе неточность, связанную с личностью оценщика.

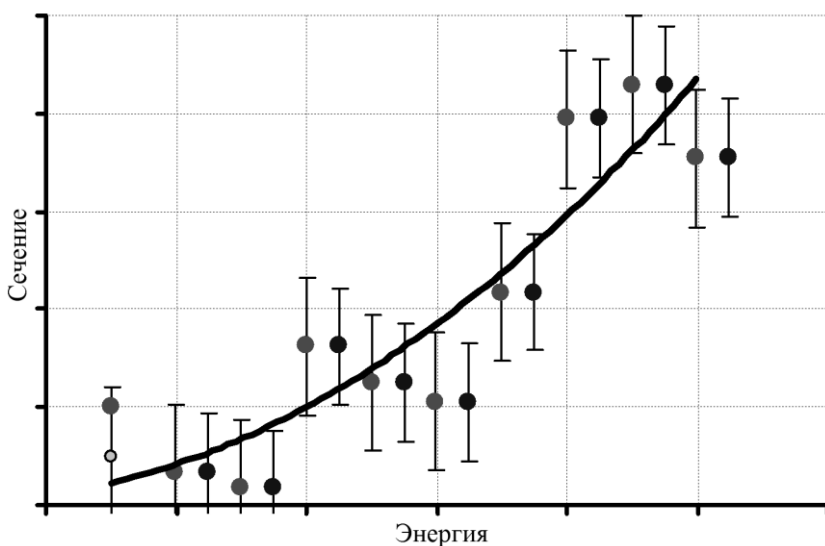


Рис. 2. Оцененные нейтронные данные – пример аппроксимации

С момента появления экспериментальной базы работа по оценке ядерных данных велась в различных лабораториях по всему миру. Естественно, тут же возникла необходимость в некотором формате представления оцененных данных, поскольку без этого сравнить две разные оценки одних и тех же данных, выполненные различными лабораториями было невозможно.

В 1964 году Генри Хонек (Henry Honeck) работавший в Брукхейвенской национальной лаборатории начал разработку системы

хранения оцененных нейтронных данных, которая бы не зависела от какой-либо конкретной программы обработки данных и была бы полностью документирована. В результате он создал библиотеку Evaluated Nuclear Data File (ENDF), которая связала оцененные ядерные данные и расчётные программы.

Всего было три версии библиотеки ENDF: ENDF/A, ENDF/B и ENDF/C. Первая версия предназначалась для хранения частично оцененных данных. В ней одновременно могли присутствовать несколько различных наборов оцененных и экспериментальных данных, касающихся одного и того же процесса, часть данных могла отсутствовать и т.д. Вторая версия использовала тот же формат, что и первая, но предназначалась для хранения одного полного набора данных для проведения нейтронных расчётов. В этой библиотеке информация упорядочивалась по номеру материала, типу данных и типу реакции. Предполагалось, что эта библиотека будет обновляться на регулярной основе. Третья версия оказалась неудачной и распространения не получила. При её создании была предпринята попытка создать такой формат, который сможет обеспечить хранение и удобный доступ к огромному количеству дополнительной информации, которая может понадобиться в расчётах. В дальнейшем эта идея была до некоторой степени реализована при последующем расширении возможностей ENDF/B. По мере наполнения библиотеки ENDF/B интерес к ENDF/A постепенно угас.

Первая версия ENDF/B-I появилась в 1968 году и была составлена из доступных на тот момент оценок, выполненных в различных лабораториях США, простой конвертацией в формат ENDF. В 1970 году вышла вторая версия библиотеки (ENDF/B-II), содержащая как переоцененные данные из предыдущей версии, так и новые оценки. Последней на момент написания этого текста версией библиотеки является ENDF/B-VII, вышедшая в 2006 году. Эта библиотека базируется на шестой версии формата ENDF (ENDF-6).

Вообще в мире не так уж и много различных библиотек оцененных ядерных данных. Помимо ENDF/B (США), существуют JENDL (Япония), JEFF (Западная Европа), CENDL (Китай), ФОНД, БРОНД и РОСФОНД (Россия).

### 1.3. ИСПОЛЬЗОВАНИЕ ОЦЕНЕННЫХ ДАННЫХ

На практике оцененные ядерные данные могут использоваться в расчётах как непосредственно, так и после некоторого предварительного преобразования.

Несмотря на то, что существуют программы, работающие с данными, представленными непосредственно в формате ENDF, они, скорее, исключение. На практике, большинство программ реализует второй способ – предварительное преобразование данных.

В основном это связано непосредственно с оптимизацией скорости счёта. Дело в том, что непосредственное использование данных в формате ENDF во время счёта ведет к огромным затратам машинного времени на выполнение преобразований, многие из которых, вообще говоря, можно выполнить заранее. Так, например, если программа при расчёте использует разбиение энергетической шкалы на группы, (т.е. сечение нуклида постоянно в заданном интервале по энергии), то имеет смысл заранее подготовить такую групповую библиотеку. В случаях, когда программа использует точную зависимость сечений от энергии (в этом случае говорят о поточечном представлении данных, поскольку кривая зависимости сечения от энергии в программах часто представлена в виде довольно большого набора точек), опять же имеет смысл заранее перевести данные в формат, который удобен для работы расчётной программы.

При этом надо понимать, что такие преобразования могут не влиять на точность данных или влиять незначительно, как в случае с переходом к поточечному представлению, при котором, как правило, задается достаточно высокая точность преобразования, либо вносить заметную погрешность. Её необходимо учитывать при проведении расчёта. Последнее верно, например, при создании групповых библиотек. В этом случае при проведении расчёта необходимо понимать, что такие библиотеки могут использоваться только для расчёта тех систем, для которых данная библиотека предназначена, в противном случае нужно быть готовым к получению совершенно неверного результата.

Для преобразования данных из формата ENDF в необходимый для расчётной программы формат в мировой практике получила распространение американская программа NJOY, позволяющая готовить данные как в групповом, так и поточечном приближении.

Тем не менее часто производители программного обеспечения при создании библиотек для своих программ применяют собственные разработки, используя их как вместо NJOY, так и вместе с ней.

#### **1.4. БЕНЧМАРК-ЭКСПЕРИМЕНТЫ**

Лет 40 назад нейтронные данные применялись только для прикидочных расчётов, а окончательные значения проектных характеристик определяли на основе сравнения результатов расчётов с результатами, так называемых, макроскопических экспериментов на установках, моделирующих проектируемый объект.

Однако сейчас мир изменился не только физически, но и экономически. Почти все критические стенды, использовавшиеся для моделирования, давно демонтированы. Проведение новых экспериментов на имеющихся установках требует значительных финансовых затрат. Поэтому важную роль играет планирование эксперимента.

При постановке любого эксперимента необходимо решать вопрос о том, каким образом лучше организовать эксперимент, чтобы с минимальными затратами получить максимальное количество достоверных и пригодных к практическому использованию данных. Для этого необходимо найти наилучшие в этом смысле параметры установки, характеристики образцов, размещаемых в установке, а часто и место их размещения, поскольку размещение образцов в удобном для экспериментатора месте может вызвать полную непригодность полученных данных на практике. Планирование эксперимента выполняется с помощью расчётов.

Современное состояние компьютерной техники и программного обеспечения позволяет проводить расчёты настолько точные, что ими можно заменить значительное количество экспериментов. Подтверждение точности расчётов программ, используемых для выполнения таких численных экспериментов, обычно проводится следующим образом. Находятся реально выполненные и достаточно хорошо описанные эксперименты, сходные с планируемым. Для выполнения такого рода расчётов обычно используется программа прецизионного класса. Как правило, в таких программах реализован метод Монте-Карло, который позволяет задавать рассчитываемую систему практически без упрощений как в части нейтронных

данных, так и в части её геометрии. Средствами такой программы создаются трёхмерные модели выбранных экспериментальных установок. Эти модели обсчитываются, и выполняется сравнение полученных результатов с имеющимися экспериментальными данными. После выполнения такого сравнения делается вывод о точности, с которой данная программа рассчитывает установки рассмотренного типа.

Определение точности расчёта – необходимый шаг при проведении численного эксперимента в поддержку планируемого реального.

Для упрощения процесса поиска и отбора экспериментов для этой работы в мире идет сбор информации обо всех критических экспериментах. Данный проект был начат в 1996 году США и Россией, а затем к нему присоединились и другие страны. Сегодня Международный Справочник по критическим экспериментам в обоснование ядерной безопасности [4] содержит описания около четырёх тысяч критических экспериментов. Кроме экспериментов на стендах и лабораторных установках существует ряд экспериментов, выполненных на реакторах. Эти эксперименты собираются в рамках Международного проекта по физике реакторов [5].

Главной характеристикой эксперимента, пригодного для верификации прецизионных программ, является точное описание экспериментального оборудования и условий проведения эксперимента, позволяющее быть уверенным, что рассчитывается точно тот же объект, что и реально существующий. Такие эксперименты называют *бенчмарк-экспериментами*.

## **ГЛАВА 2. МЕТОДЫ РАСЧЁТОВ ПЕРЕНОСА ИЗЛУЧЕНИЯ**

Все методы расчётов переноса излучения, применяемые при проектировании и сопровождении различных установок, можно условно разделить на традиционные и компьютерные (рис. 3).



Рис. 3. Методы расчёта переноса излучения

При традиционном подходе используют данные из таблиц, справочников и руководств. Эти данные могут быть получены с помощью аналитических вычислений, из результатов экспериментов или с помощью расчётов модельных задач. Чаще всего применяют не слишком громоздкие формулы и таблицы подставляемых в них параметров, подобранных на основе экспериментальных данных. Как правило, такой подход позволяет быстро получить результат, однако с большой погрешностью. Часто для решения поставленной задачи оказывается достаточно результата с 50-процентной точностью. Например, в экспериментах 1950-х годов по оценке прохождения излучения через различные материалы погрешность в 40–50 % вполне удовлетворяла экспериментаторов. В качестве другого примера, можно привести использование таблиц для определения области радиоактивного заражения при чрезвычайной ситуации в случае военных действий. В этом случае грубой, но быстрой оценки, полученной в полевых условиях без примене-



ния вычислительной техники, будет достаточно для общего понимания ситуации, что необходимо для принятия решений.

Компьютерные методы позволяют получить результат с большей точностью. В зависимости от используемого в расчётах метода, затрачиваемое на получение результата время может оказаться значительным. Вообще говоря, расчёт может длиться секунды, недели и даже месяцы.

Компьютерные методы можно условно разделить на инженерные и прецизионные (детерминистические и стохастические).

Инженерные методы в основном используют для решения задачи малогрупповые диффузионные уравнения. При расчёте ядерно-энергетических установок термином «группа» обычно обозначают отрезок оси энергии; малогрупповой метод использует разбиение на небольшое число отрезков, т.е. зависимость от энергии огрубляется. Для них характерно приближение пространственной гомогенизации, когда реальная установка заменяется набором областей, содержащих некоторые обобщенные по своим свойствам материалы. Этот подход позволяет быстро получить результат, однако необходимость использования многочисленных приближений делает инженерные программы узкоспециализированными.

В последнее время всё чаще применяют новые методы инженерных расчётов – так называемые нодальные. В этом случае реактор тоже разбивают на ячейки и используют малогрупповое приближение энергетических зависимостей, потоки и все функционалы считают определенными относительно малым набором характеристик состояния ячеек (например, средним потоком на гранях, скоростью генерации нейтронов в ячейке и т.п.). Но внутреннюю структуру ячейки не заменяют на гомогенную среду, а с помощью более точных методов расчета получают линейные зависимости между характеристиками состояния ячеек. Затем с помощью данных зависимостей и состыковки значений на гранях соседних ячеек получают уравнения для реактора. Это можно рассматривать как обобщение диффузионного подхода, в котором скаляры заменяются тензорами.

Однако и эти классы методов являются специализированными, хотя бы по набору существующих рассчитываемых ячеек.

Детерминистические методы основаны, как правило, на решении интегродифференциальных уравнений переноса излучения. Для них характерно использование техники многогрупповых методов (SN-методы, методы дискретных ординат и т.д.). Система, как правило, задается с помощью растрового описания геометрии, т.е. разбиения пространства на регулярную сеть ячеек одинаковой формы с одним материалом (например, параллелепипедов) для учета 2D- и 3D-геометрии. Пример задания геометрии установки с использованием цилиндрических координат дан на рис. 4.

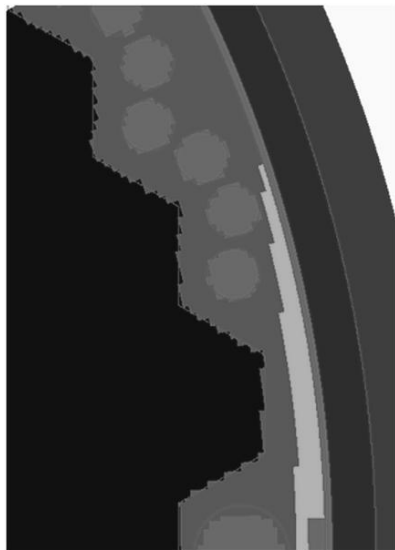


Рис. 4. Растровая геометрия

Программы, реализующие детерминистический подход, используют значительно меньше приближений, чем инженерные, и поэтому с их помощью можно рассчитывать системы более точно. К недостаткам таких программ можно отнести в первую очередь сложность точного описания геометрии. Уточнение описания уменьшает размеры ячеек и, соответственно, увеличивает их количество, что тут же приводит к расходу оперативной памяти, поскольку для каждой ячейки надо определить характеризующие её многогрупповые константы. В области энергий, где имеется резонансный захват нейтронов ядрами и сечения при малых изменениях энергий могут меняться на несколько десятичных порядков, число энергетических групп резко возрастает, также требуя увеличения необходимых ресурсов памяти. Это составляет второй значительный недостаток метода - необходимость больших объёмов оперативной памяти.

Зависимость от направления полёта тоже приходится представлять в конечномерном виде. Увеличение точности данного представления также увеличивает требования к объёму памяти.

Метод Монте-Карло – наиболее универсальный метод, применяемый для расчёта переноса излучений. Как правило, программы, реализующие метод Монте-Карло, позволяют моделировать трёхмерные системы с произвольной геометрией, используя комбинаторный подход, основанный на описании сложных пространственных форм комбинациями простых тел или поверхностей с помощью теоретико-множественных операций пересечения, дополнения и объединения. При расчёте такие программы применяют константы непосредственно для нуклида, то есть память затрачивается только на хранение информации для присутствующих в материалах нуклидов. При этом константы используются не групповые, а поточечные, что обеспечивает возможность моделирования с непрерывным слежением за энергией частицы. Для описания резонансов часто встречающихся нуклидов возможно применение их теоретического описания в виде формул.

Основное преимущество метода Монте-Карло над другими методами – возможность точного описания любой геометрии и использования не групповых, а поточечных констант. Это позволяет сократить до минимума количество применяемых при расчёте приближений, что позволяет говорить о методе Монте-Карло как о численном эксперименте, способном заменить эксперимент реальный. Направление полёта частиц также моделируется без какой-либо дискретизации.

Основной недостаток метода – время, затрачиваемое на получение результата. Однако метод Монте-Карло хорошо работает на многопроцессорных кластерах и даже сетях ЭВМ. Он получает всё большее распространение благодаря своей универсальности и увеличению мощностей вычислительной техники.

### **ГЛАВА 3. МЕТОД МОНТЕ-КАРЛО**

Родоначальником метода Монте-Карло считается Энрико Ферми, который занимался его развитием в 1930 гг. в Италии. В 1940 г. Джон фон Нейман и Станислав Улам в Лос-Аламосе применили метод в компьютерном моделировании.

Метод Монте-Карло основан на получении большого числа реализаций стохастического (случайного, вероятностного) процесса, который формируется таким образом, чтобы его вероятностные

характеристики совпадали с аналогичными величинами решаемой задачи.

При расчёте моделируется поведение индивидуальных частиц от момента рождения до поглощения (с делением ядра или без деления) либо вылета частицы из системы. Основанием для такой возможности служит практически полное отсутствие взаимодействия нейтральных частиц друг с другом, а значит, и независимость их траекторий. Для расчёта переноса заряженных частиц данный метод применяется реже и сталкивается со значительными трудностями. Совокупность всех событий, произошедших с частицей от момента рождения до «смерти», называется *траекторией* или *историей частицы*.

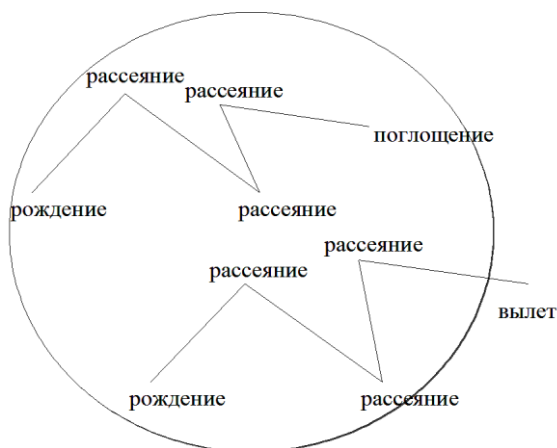


Рис. 5. Моделирование поведения индивидуальных частиц

По смыслу метод Монте-Карло очень похож на игру в рулетку, когда всё дело решает случай. Кстати, название методу дал работавший с Уламом в одной группе в Лос-Аламосе Николас Константин Метрополис, дядя которого был азартным игроком и постоянно занимал у него денег.

Моделирование полёта частицы ведётся на основании выборки псевдослучайных чисел и расчёта вероятностей взаимодействия частиц этого типа с веществом. Последовательности псевдослучай-

ных чисел генерируются с помощью специальных алгоритмов. Эти числа, несмотря на детерминированность процесса их генерации, обладают многими свойствами аналогичными свойствам независимых случайных чисел, равномерно распределённых на отрезке  $[0,1]$ .

Параметры расчёта вероятностей взаимодействий нейтральных частиц с ядрами измеряют в экспериментах и хранят в специальных файлах оцененных ядерных данных.

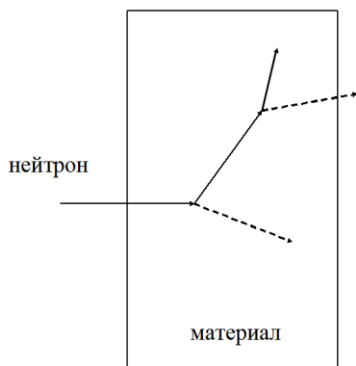


Рис. 6. Моделирование пробега

Пробег частицы до столкновения в бесконечной однородной среде с полным макроскопическим сечением  $\Sigma_t$  – случайная величина с плотностью вероятности  $p(\ell) = \Sigma_t \exp(-\Sigma_t \ell)$ .

Её значения в интервале  $(0, \infty)$  можно получить с помощью случайных чисел  $\gamma$ , равномерно распределённых в интервале  $(0,1)$ :

$$\ell = -\frac{1}{\Sigma_t} \ln \gamma$$

При наступлении события столкновения необходимо определить нуклид, с которым столкнулся нейтрон. Для этого разыгрывается случайное число  $\xi$ , равномерно распределённое на интервале  $[0,1]$ . Считается, что взаимодействие происходит с нуклидом номер  $k$ , если

$$\sum_{j=1}^{k-1} \Sigma_{t,i} \leq \xi \sum_{j=1}^K \Sigma_{t,i} < \sum_{j=1}^k \Sigma_{t,i},$$

где  $K$  - число нуклидов в материале, а  $\Sigma_{t,i}$  – парциальное полное сечение нуклида  $i$ .

После определения нуклида, на котором произошло столкновение, необходимо установить реакцию, то, что произойдет в результате столкновения. Непосредственными результатами столкновения могут быть рассеяние или поглощение. Однако это не конечная реакция, поскольку, например, в результате поглощения нейтрона ядром могут произойти либо деление ядра, либо захват нейтрона ядром, что в свою очередь порождает дальнейший выбор. Опреде-

ление типа реакции производится следующим образом. Считается, что происходит реакция типа  $n$ , если

$$\sum_{j=1}^{n-1} \Sigma_j \leq \xi' < \sum_{j=1}^n \Sigma_j,$$

где  $N$  - полное число реакций, определённое для заданного нуклида.

В процессе моделирования траектории частицы выделяются так называемые события. Для регистрации используются следующие события: пробег, столкновение или поглощение частицы в интересующей нас геометрической области.

Метод Монте-Карло состоит в генерации последовательности случайных чисел  $x_j$ , математическое ожидание которых и даёт искомую величину. Обычно одновременно генерируют много последовательностей случайных чисел, каждая из которых даёт свой целевой функционал, например интеграл потока какой-либо области.

Как оценка математического ожидания последовательности  $\{x_j\}$  берётся выборочное среднее

$$Mx \approx \bar{x} = (x_1 + x_2 + \dots + x_N)/N, \quad (1)$$

где  $N$  – число членов в последовательности. Очевидно, что оценка несмещённая, т.е.  $M(x) = M(\bar{x})$ .

Точность полученного результата определяется его дисперсией

$$D(\bar{x}) = M\bar{x}^2 - M(\bar{x}^2) = \frac{D(x)}{N},$$

где  $D(x) = D(x_i)$  – дисперсия каждого слагаемого  $x_i$ . Часто используется и среднееквадратичное отклонение  $\sigma = \sqrt{D(\bar{x})} = \frac{\sqrt{D(x)}}{\sqrt{N}}$ .

Таким образом, точность определения распределения частиц в рассчитываемой системе (и, соответственно, рассчитываемых характеристик) зависит от количества промоделированных частиц. Чем больше моделируется частиц, тем точнее ответ. Из определения среднееквадратичного отклонения непосредственно следует, что с увеличением количества промоделированных частиц статистическая погрешность убывает как  $\frac{1}{\sqrt{N}}$ , поэтому для уменьшения погрешности в два раза требуется промоделировать в четыре раза больше частиц.

Заранее значение  $D(x)$  редко бывает известным. Если члены последовательности взаимно независимые случайные величины, то допустима оценка

$$D\bar{x} = \frac{1}{N^2} (x_1^2 + \dots + x_N^2 - N\bar{x}^2). \quad (2)$$

Эта смещённая оценка, но относительное смещение есть  $(N-1)/N$ , что при больших  $N$  не существенно.

Основными событиями являются:

- столкновение частицы с ядром;
- поглощение частицы;
- свободный пробег частицы внутри выделенной области пространства.

По любому из них можно строить способ оценки или регистрации.

Для каждого целевого функционала (т.е. величины, которую надо получить по каждому событию) рассчитывается величина, называемая вкладом одного события в данный функционал. Очень часто вклад этот нулевой или, как говорят, вклад отсутствует. Например, если целевой функционал скорость поглощения нейтронов в топливе, то событие поглощения нейтрона в воде никакого вклада не дает.

Так как подавляющая часть рассчитываемых величин выражается через интегралы потока с различными весами, то основными функционалами можно считать интегралы потока по объемам  $V_m$  и энергетическим группам  $\Delta E_n$

$$(\bar{\Phi}V)_{mn} = \iint_{V_m \Delta E_n} \Phi(\mathbf{r}, E) dV dE.$$

При регистрации по столкновениям вклад есть  $1/\Sigma_t$ , если столкновение произошло в объеме  $V_m$ , а энергия лежала в интервале  $\Delta E_n$ , и ноль в противном случае. Для регистрации по поглощениям вклад есть  $1/\Sigma_a$ . При регистрации по пробегам вклад равен  $\ell$ , где  $\ell$  есть длина регистрируемого пробега внутри  $V_m$ , а энергия лежит в  $\Delta E_n$ .

События на независимых траекториях взаимно независимы. События на одной траектории не являются независимыми. Поэтому вклады по событиям одного типа по одной траектории объединяются и получают вклады траекторий по столкновениям, поглощениям и пробегам. Например, для интеграла потока по  $V_m$  и  $\Delta E_n$ , оценка по столкновениям для траектории будет

$\langle \text{число столкновений в } V_m \text{ и } \Delta E_n \text{ данной частицы} \rangle / \Sigma_t$ .

Для получения окончательной оценки суммируют все вклады данного типа всех траекторий и делят сумму на общее число траекторий. Для любой величины можно получить три оценки: по столкновениям, поглощениям и пробегам.

Необходимо учитывать, что, чем меньше выбранная для регистрации геометрическая область, тем меньше частиц в неё попадет. Это значит, для получения статистически приемлемого результата нам придется моделировать большее количество частиц. То же верно и для ширины энергетического интервала, в котором ведется регистрация. Чем меньше интервал, тем больше частиц необходимо моделировать.

#### **ГЛАВА 4. ОБЩАЯ СХЕМА МОДЕЛИРОВАНИЯ МЕТОДОМ МОНТЕ-КАРЛО**

Как уже говорилось, основой расчёта переноса нейтронов и фотонов методом Монте-Карло является моделирование траектории одной частицы от рождения до поглощения или утечки. Для этого служит специальный программный блок. На его вход необходимо подать стартовое расположение частицы в фазовом пространстве, то есть начальные геометрические координаты, энергию, направление полёта и, наконец, тип частицы: нейтрон или фотон. Для всего расчёта необходимо моделировать очень большой набор траекторий.

Важно иметь в виду, что в процессе моделирования траектории могут возникать новые частицы, что соответствует реальным физическим процессам.

При поглощении нейтрона ядро может разделиться и породить несколько нейтронов деления. Это происходит лишь в конце траектории.

Существует реакция  $n \rightarrow 2n$ : нейтрон поглощается ядром, а затем оно испускает два нейтрона. Эту реакцию можно было бы объявить одним из типов деления, и считать, что старый нейтрон погиб, и родилось два новых нейтрона. Однако по традиции считается, что траектория при этом не кончается, а просто родился один новый нейтрон. Точно так же трактуют более редкую реакцию  $n \rightarrow 3n$ : один нейтрон остался, и родилось два новых. Реакций вида  $n \rightarrow 4n$  и



$n \rightarrow 5n$  при энергиях частиц, характерных для ядерных и термоядерных установок, не встречается. Таким образом, траектория ещё не кончена, а новые частицы уже образовались.

Тем более что это относится к генерации фотонов, происходящей при неупругом рассеянии и ряде других реакций. При этом нейтрон траектории остается, и траектория ещё не окончена.

Частицы, появившиеся в процессе моделирования траектории данного нейтрона, обычно называют потомками этого нейтрона. При моделировании траекторий этих частиц возникают новые нейтроны и фотоны, которые можно также назвать потомками первого нейтрона. То есть некоторый исходный нейтрон порождает дерево траекторий, которые мы будем называть расширенной траекторией. Фотон также порождает новые фотоны. В реакторных вычислениях обычно не учитывают возможность порождения фотонами новых нейтронов.

Существуют две основные задачи переноса частиц: задача с источником (неоднородная) и задача на  $k_{eff}$  (однородная). Схемы моделирования набора траекторий для этих задач несколько различаются.

Стационарную задачу с источником (а мы рассматриваем лишь стационарный случай) осмысленно решать только для подкритичной системы. Это означает, что расширенная траектория всегда конечна, хотя число отдельных частиц в ней может быть очень велико. Траектории, принадлежащие расширенной траектории, никак нельзя считать независимыми. Например, нейтрон деления появляется в точке трёхмерного пространства, где был поглощен породивший его нейтрон.

Всё это позволяет организовать вычисления следующим образом.

Существует блок, моделирующий внешний источник. Он генерирует частицы, то есть начальные точки траекторий, которые взаимно независимы.

По каждой частице из источника генерируется расширенная траектория, пока все потомки нейтрона источника не исчезнут. Две разные расширенные траектории независимы. Для каждого рассчитываемого функционала регистрируются суммарные вклады по каждой расширенной траектории. Эти случайные величины оказываются независимыми. Их усреднение и есть оценка функционала, а

их независимость позволяет оценивать дисперсию по обычной формуле (2), где  $x_i$  – суммарный вклад по расширенным траекториям, а  $N$  – их общее число.

Отметим, что усреднение дает оценку функционала, отнесённую к одному нейтрону источника. Зная мощность источника, то есть число нейтронов испускаемых в секунду, можно легко получить абсолютное значение функционала.

Часто, по чисто программистским соображениям, из источника выбирают не по одной частице, а пакетами частиц, которые образуют очередь на моделирование траекторий. Общая схема моделирования для задачи с источником представлена на рис. 7.

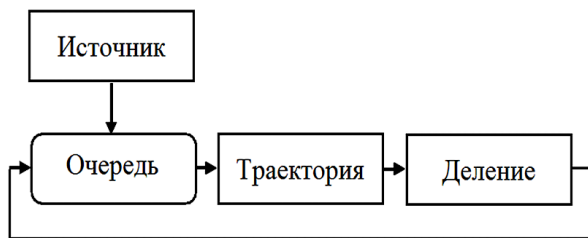


Рис. 7. Задача с источником

Для расчёта на  $k_{eff}$  такой подход невозможен – никакого внешнего источника нет, и любой нейтрон в системе является потомком, поэтому для расчёта однородных задач используется метод поколений.

Фактически метод поколений оперирует условно-критическими системами, смысл которых надо пояснить.

В реальном реакторе все нейтроны рождены в результате деления. Каждый из них имеет в среднем одного потомка, то есть баланс нейтронов можно представить соотношением

$$g = K_T g,$$

где  $g$  – скорость появления нейтронов деления в секунду в единице фазового объема, а символом  $K_T$  обозначен оператор генерации потомков.

Такая система называется критической. В расчёте всегда используется не совсем точная модель, хотя бы потому, что все измерения геометрических размеров, химических составов компонентов

установки и т.п. имеют погрешности. В модели вместо оператора  $K_T$  используется несколько измененный оператор  $K_M$ , а для него может и не существовать решения вида  $g = K_M g$ . Таким образом, физически осмысленного однородного стационарного решения, когда каждый нейтрон в среднем имеет одного потомка, может не быть. Однако можно рассмотреть задачу

$$g = (K_M/k_{eff}) g ,$$

где  $k_{eff}$  – некоторое заранее неизвестное положительное число, причём максимальное из всех возможных.

Физически это означает, что если бы среднее число вторичных нейтронов на каждый акт деления уменьшилось в  $k_{eff}$  раз, то модельная система стала бы критичной, в ней была бы возможность самоподдерживающейся реакции. Для распределения  $g$  решение такой задачи  $k_{eff}$  означает среднее число вторичных нейтронов, рожденных каждым нейтроном в системе.

В силу критичности модельной системы после деления на  $k_{eff}$  решение подобной задачи называется условно критичным. При этом неизвестными являются как число  $k_{eff}$ , так и функция  $g$ , которая определена с точностью до множителя.

В методе поколений заранее фиксируется число нейтронов в поколении, обозначенное в программе MCU как NTOT. Первое поколение вырабатывается блоком, моделирующим источник. Его распределение по пространству, естественно, отличается от ещё неизвестного распределения  $g$ .

Происходит моделирование траекторий всех нейтронов поколения. При этом нейтроны, появившиеся в результате  $n \rightarrow 2n$  и сходных реакций, также моделируются, как и возникающие фотоны. В отличие от этого нейтроны деления откладываются в особую очередь и на моделирование не выбираются. Так происходит, пока текущее поколение не исчерпается.

В результате в текущем поколении нет частиц, а в отложенной очереди имеется некоторое число  $M$ , не обязательно совпадающее с NTOT. Даже если бы модельная система была строго критичной и текущее поколение строго соблюдало распределение  $g$ ,  $M$  было бы отлично от NTOT в силу вероятностного характера моделирования. Строгое соответствие  $g$  возможно лишь для системы с конечным числом состояний, а не для реактора.

Далее выполняется процесс нормализации, в результате которого образуется NTOT частиц, каждая из которых является одной из частиц отложенной очереди. Если  $M < NTOT$ , то некоторые частицы придется выбрать два или более раз, а если  $M > NTOT$ , некоторые частицы не будут выбраны вообще. Существует много алгоритмов нормализации. Во всех из них соблюдается «равноправие» отложенных нейтронов по отношению к их выборке. Алгоритмы, используемые в программе MSU, описаны в документации.

После окончания нормализации очередь отложенных нейтронов очищается.

Отношение  $M/NTOT$  есть вклад в оценку  $k_{eff}$ .

Выбранные NTOT частиц образуют новое текущее поколение. Вновь начинается этап моделирования их траекторий. Общий процесс состоит из сменяющих друг друга этапов моделирования траекторий текущего поколения и образованием очереди отложенных частиц, а затем нормализации, создающей новое поколение. Схема этого процесса представлена на рис. 8.



Рис. 8. Блок-схема для неоднородной задачи

Блок источника работает только один раз, в то время как число поколений всегда не меньше нескольких тысяч, а часто и существенно больше. При расчёте небольших систем, например ячеек реактора с зеркальными граничными условиями, уже второе поколение оказывается распределённым способом, близким к решению, независимо от распределения источника. В таких системах можно использовать источник, в котором все частицы рождаются в одной точке. Однако при рассмотрении больших систем, например реактора целиком, использование точечного источника приведет к тому, что только после тысяч поколений распределение примет разумную форму, т.е. при усреднении по всем поколениям нужно

про моделировать много тысяч поколений, чтобы ошибка в начальной стадии мало сказывалась. Если нет никаких сведений о правильном распределении, лучше взять для источника равномерное распределение по активной зоне, а еще лучше по топливу.

При расчёте задачи на  $k_{eff}$  нет расширенных траекторий, и при расчёте оценок функционалов накопленные суммы откликов делятся на общее число промоделированных траекторий. Таким образом, оценки отнесены к одному нейтрону деления.

При расчёте абсолютных значений функционалов полученные оценки следует умножить на мощность реактора, выраженную в числе рождений нейтронов в секунду.

Два последовательных поколения и их вклады в оценки функционалов нельзя считать независимыми – нейтроны второго поколения рождены в точках поглощения первого. Поэтому применение формулы (2) для определения дисперсии оценок не корректно. Для точной оценки дисперсии нужно было бы рассчитывать автокорреляционные функции вкладов, но это вызывает дополнительные трудности. Поэтому обычно используют более упрощённый метод, который и считается стандартным в программе MSU. Этот метод можно назвать *методом серий*.

Всё множество поколений разбивают на серии – наборы по Nbatch последовательных поколений, где константа Nbatch задаётся пользователем или берётся по умолчанию. Таким образом, серия номер  $m$  состоит из поколений с номерами  $[(m-1)*Nbatch+1, m*Nbatch]$ . Зависимость между поколением  $m$  и  $m+k$  убывает с ростом  $k$ , поколения «забывают» свою предысторию. Для некоторых значений  $k$  поколения  $m$  и  $m+k$  можно считать уже независимыми. Считается, что число Nbatch в несколько раз превышает такое  $k$ . Тогда в двух соседних сериях зависимости сказываются лишь на стыках, для относительно малого числа поколений. Поэтому суммарные вклады по всем траекториям серий можно считать почти независимыми даже для двух последовательных серий. Поэтому накопление статистики идет в два этапа.

Для каждой серии и для каждого функционала рассчитываются средний вклад по серии

$$y_s = \frac{\sum_{i=1}^{Nbatch} \sum_{j=1}^{Ntot} y_{sij}}{(Nbatch * )},$$

где  $s$  – номер серии,  $i$  – номер поколения в серии, а  $j$  – номер траектории в поколении. Символом  $y_{sij}$  обозначена соответствующая сумма вкладов по траектории для соответствующего функционала. Никаких оценок дисперсии внутри серии не производится.

Окончательная оценка идет по формуле

$$y = (y_1 + y_2 + \dots + y_{N_s}) / N_s,$$

где  $N_s$  означает число промоделированных серий. Эта оценка тождественна усреднению вкладов по траекториям по общему числу траекторий.

Оценка дисперсии оценки  $y$  делается по формуле

$$\left( \frac{y_1^2 + \dots + y_{N_s}^2}{N_s} - y^2 \right) / N_s.$$

Ясно, что это есть формула (2), применённая к средним вкладам по сериям.

В программе MCU реализованы несколько другие формулы, отличающиеся от приведённых простыми алгебраическими преобразованиями, но их вид менее нагляден.

Определение допустимых для данного типа расчёта значений чисел  $N_{tot}$  и  $N_{batch}$  далеко не простая задача. При этом следует помнить, что в задачах на  $k_{eff}$  имеется систематическая погрешность оценок, обратно пропорциональная  $N_{tot}$ . Для каждого типа реакторов были проведены исследования для выбора этих величин. Например, для реактора ВВЭР допустимо проводить расчёты с  $N_{tot}=2000$ ,  $N_{batch}=40$ .

При расчёте задачи с источником программа MCU также формально работает с сериями, но в этом случае значения  $N_{tot}$  и  $N_{batch}$  малозначимы. Можно брать значения, заданные по умолчанию.

В ряде случаев при решении задач методом Монте-Карло используется метод статистических весов [6, 7]. При моделировании переноса в этом случае каждая частица снабжается дополнительным атрибутом – статистическим весом  $w > 0$ .

Этот вес в процессе моделирования траектории может меняться. Например, после столкновения частица, имеющая вес  $w$ , с вероятностью  $p$  должна поглотиться, иначе она рассеивается. Вместо моделирования возможного поглощения вес частицы становится равным  $w \cdot (1 - p)$ , и она обязательно рассеивается. Во всех реализациях метода статистических весов обеспечена невозможность неограни-

ченного убывания статистического веса частицы – если он становится слишком маленьким, частица уничтожается.

Статистический вес частицы входит множителем во вклад события в оценку функционала. Например, вклад в оценку скорости реакции от события поглощения частицы с весом 2.0 равен сумме вкладов от поглощения в том же элементе фазового пространства *двух частиц* со статистическим весом 1.0.

Аналогично, при моделировании реакции  $n \rightarrow 2n$  вместо создания дополнительного нейтрона, вес частицы после рассеяния удваивается.

Можно показать, что использование статистических весов, согласованных с истинными вероятностями событий, оставляет неизменными математические ожидания всех моделируемых случайных величин.

Обычный, аналоговый метод моделирования можно интерпретировать как метод, когда все статистические веса тождественно равны 1.

## ГЛАВА 5. ПРОГРАММЫ, ОСНОВАННЫЕ НА МЕТОДЕ МОНТЕ-КАРЛО

Первые программы, реализующие метод Монте-Карло, были очень простыми. Они создавались на машинном языке и предназначались для решения только узкоспециальных задач. В них не было универсальной геометрии и поточечного представления сечений.

Сейчас такие программы пишутся на языках высокого уровня. Основные используемые в настоящее время пакеты программ, предназначенные для моделирования переноса излучений, написаны на Фортране. За более чем полувековой период на этом языке была создана огромная библиотека подпрограмм и функций. Причем все они были не только написаны, но также проверены и отлажены в течение этого длительного периода времени.

Современные программы имеют достаточно сложную внутреннюю структуру, которая может сильно различаться в разных пакетах.

Практически все программы включают универсальный геометрический модуль, позволяющий описывать без упрощений произвольную трёхмерную геометрию с точным заданием всех материалов моделируемой системы.

Из программ, реализующих метод Монте-Карло для расчёта переноса нейтронов и фотонов, можно выделить: MCNP, KENO (США), TRIPOLI (Франция), MONK (Великобритания), GEANT4 (Европа), MCU (Россия).

## **ГЛАВА 6. ПРОЕКТ MCU**

Проект MCU, развиваемый в России с 1982 года, направлен на разработку и практическое использование универсальной компьютерной программы для численного моделирования процессов переноса различного вида излучений (нейтронов,  $\gamma$ -квантов, электронов, позитронов) в трёхмерных системах методом Монте-Карло. Для электронов и позитронов предполагается, что их плотность настолько мала, что самосогласованным полем можно пренебречь.

*Основные направления работы:* развитие алгоритмов метода Монте-Карло; создание программных единиц (модулей) для решения однородных и неоднородных уравнений переноса излучений; создание константного обеспечения (библиотек) для этих программ.

Всего к 2012 году в рамках проекта было выпущено пять пакетов MCU. Каждый пакет имеет свое собственное название и состоит из модулей, библиотек и некоторого набора вспомогательных программ.

Все программы, созданные на базе одного пакета, различаются в основном набором модулей и, как следствие, областью применения, однако все они имеют однотипный ввод исходных данных. При этом нужно понимать, что если конкретная программа не предназначена для расчёта, например фотонов, то и исходные данные, относящиеся к моделированию фотонов, никакого влияния на ход расчёта не окажут либо вызовут сообщение об ошибке.

Далее мы будем рассматривать исходные данные для программы, собранной из модулей пакета MCU-5. Любая такая программа обязательно содержит следующие модули: физический, геометрический, источников, регистрации, транспортный, управляющий и оборудования. Назначение каждого из модулей непосредственно отражено в его названии.

Физический модуль вырабатывает необходимые сечения взаимодействия частиц со средой и моделирует их столкновения с ядрами. Геометрический – вычисляет все необходимые функции, ар-



гументами которых служат пространственные координаты, а также моделирует полёт частиц от одного столкновения до другого. Модуль источников моделирует частицы источника или начальный пакет нейтронов при решении однородной задачи. Модуль регистрации накапливает статистику событий, необходимую для оценки функционалов, а потом обрабатывает эту статистику. Транспортный модуль, иногда называемый модулем траекторий, осуществляет организацию моделирования отдельных траекторий частиц. Управляющий модуль организует совместную работу всех модулей. Модуль оборудования содержит такие общие для всех других модулей подпрограммы, как датчик псевдослучайных чисел, программы открытия файлов, программы ввода-вывода и др. Свое название этот модуль получил из-за того, что ранее он содержал все подпрограммы, которые зависели от используемого компилятора или собственно компьютера, то есть оборудования. Кроме того, в программе может присутствовать модуль выгорания, который обеспечивает расчёт изменения изотопного состава материалов в процессе кампании реактора, а также некоторые другие модули.

Выбранный порядок перечисления модулей не случаен. Он соответствует тому порядку, в котором программа вводит исходные данные. Единственный модуль, для которого не требуется никаких исходных данных, это модуль оборудования.

Все модули написаны на языке Фортран 90/95, однако с использованием набора макрокоманд, которые позволяют внутри одного текста хранить несколько различных версий модуля. Получившаяся таким образом модификация Фортрана получила название Гуртран. Перевод с Гуртрана на Фортран осуществляется с помощью программы MCUREP, которая входит в состав программы MCU.

Программа MCU поставляется пользователю без исполняемого файла, только с набором текстов на Гуртране. Однако процесс генерации исполняемого файла, включая и перевод с Гуртрана на Фортран, автоматизирован. Эта автоматизация осуществляется с помощью файла пакетной обработки и заключается в получении текста программы в виде одного файла на языке Фортран с помощью программы MCUREP, которая поставляется в виде исполняемого модуля и текста на языке Фортран, и отправки его на компиляцию с использованием того компилятора, который выбран пользователем. В настоящее время автоматизированный процесс полу-

чения текста на языке Фортран возможен в операционных системах Windows, а по запросу и в Linux и других Unix-подобных системах. На компиляцию же текста на Фортране вообще нет никаких ограничений, кроме, конечно же, наличия работающего компилятора Фортрана 90/95, что позволяет использовать, вообще говоря, любой такой компилятор на любой операционной системе.

Программа распараллелена на основе стандарта Message Passing Interface (MPI) (как MPI-1, так и MPI-2), который является наиболее распространенным стандартом интерфейса обмена данными в параллельном программировании.

Помимо собственно средств для выполнения расчётов, программа MCU всегда содержит графический интерфейс пользователя MCU Office для визуального контроля исходных данных и анализа результатов расчётов.

## **ГЛАВА 7. УСТАНОВКА, ГЕНЕРАЦИЯ И ЗАПУСК НА СЧЁТ**

Установка программы на компьютер пользователя состоит в простом копировании директории с программой и директории с ее библиотекой. Рассмотрим на примере программы MCU-FREE процесс установки на компьютер с операционной средой Windows. После установки программы MCU-FREE на компьютер, например, на диск C в директорию MCU-FREE, содержимое этой директории будет выглядеть следующим образом:

```
C:\MCU-FREE\MCUFREE\..
C:\MCU-FREE\MCUFREE\EXE
C:\MCU-FREE\MCUFREE\MAKEMCU
C:\MCU-FREE\MCUFREE\MCUDOC
C:\MCU-FREE\MCUFREE\MCUOFF3
C:\MCU-FREE\MCUFREE\MEMORY
C:\MCU-FREE\MCUFREE\RESTEST
C:\MCU-FREE\MCUFREE\RUNTEST
C:\MCU-FREE\MCUFREE\TEXTGUR
C:\MCU-FREE\MDBFREE50\..
C:\MCU-FREE\MDBFREE50\ABBNL
C:\MCU-FREE\MDBFREE50\BNABMCU
C:\MCU-FREE\MDBFREE50\BURN5
C:\MCU-FREE\MDBFREE50\DOSIM
C:\MCU-FREE\MDBFREE50\LI PAR
C:\MCU-FREE\MDBFREE50\VESTA
```

Директория MCUFREE содержит средства для расчёта задач и состоит из набора поддиректорий:

- поддиректория EXE, предназначенная для хранения исполняемых файлов программы;
- поддиректория MAKEMCU, содержащая средства генерации программы;
- поддиректория MCUDOC, содержащая документацию по программе;
- поддиректория MCUOFF3, содержащая файлы программы MCU OFFICE;
- поддиректория MEMORY с файлами, определяющими параметры генерации программы;
- поддиректория RESTEST с результатами расчётов тестовых вариантов из директории RESTEST;
- поддиректория RUNTEST, содержащая исходные данные тестовых задач и средства запуска тестовых расчётов;
- поддиректория TEXTGUR с исходными текстами программы.

Директория MDBFREE50 содержит файлы банка ядерных данных. Названия ее поддиректорий соответствуют названиям библиотек, входящих в состав банка ядерных данных.

После того как все файлы скопированы, необходимо скорректировать с учётом имени директории, в которую мы установили программу, пути в переменных pathexe и pathdb файла mcu5.bat, находящегося в директории EXE.

Поскольку мы установили программу в директорию C:\MCU-FREE, то необходимо изменить две строки в файле C:\MCU-FREE\MCUFREE\EXE\mcu5.bat следующим образом:

```
set pathexe= C:\MCU-FREE\MCUFREE\EXE\  
set pathdb= C:\MCU-FREE\MDBFREE50\  

```

При этом важно соблюдать несколько правил. Первое правило состоит в том, что после завершающего обратного слеша ‘\’ не должно быть пробелов, табуляций или каких-либо других символов. Второе и третье правила относятся к использованию Unix-подобных операционных систем, включая Linux. В Linux, в отличие от Windows, в именах файлов используется прямой слеш ‘/’. Кроме того, в Linux в имени файла важно учитывать регистр. Например,

MCU-FREE и MCU-Free в Linux, в отличие от Windows, – разные имена.

Если планируются многопроцессорные расчёты, то такую же операцию нужно проделать и в файлах `mcu5mpi.bat` и `mcu5mpibynet.bat`.

После установки необходимо выполнить генерацию программы, т.е. получить исполняемый файл. Для этого в директории МАКЕМСУ следует выполнить команду `makemcu.bat <М или F> <имя файла параметров> <компилятор> <опция>`

<М или F> – определяет, какой подмодуль физического модуля будет использоваться в области энергии нейтронов ниже 5 эВ. Значение М (или m) используется для генерации программы с подмодулем MOFITTG, а F (или f) – для генерации программы с подмодулем FIMTOEN. Различие между подмодулями будет обсуждаться далее, а здесь достаточно отметить, что в программе MCU-FREE имеется только подмодуль FIMTOEN, поэтому использование параметра М (или m) вызовет сообщение об ошибке.

<имя файла параметров> – имя файла параметров из папки MEMORY без расширения MEM. Начинающему пользователю нет необходимости изменять поставляемый с программой файл. Заданных в нем значений управляющих параметров, как правило, достаточно для начала работы с программой. В программе MCU-FREE имеется два файла параметров: `free.mem` и `freempi.mem`. Их единственное различие состоит в том, что в первом отключены функции, обеспечивающие работу в многопроцессорном режиме.

<компилятор> – параметр, отвечающий за выбор компилятора с языка Фортран 90/95. Вызовы поддерживаемых компиляторов задаются в файле `setcomp.bat`, находящемся в директории МАКЕМСУ, и пользователь, вообще говоря, может добавлять компиляторы по своему усмотрению. Однако на практике это не требуется. По умолчанию, если не указывать в команде параметр, отвечающий за выбор компилятора, будет выбран Intel Fortran. При указании в качестве этого параметра G или g при компиляции будет вызываться компилятор G95, который доступен по лицензии GNU на сайте проекта <http://g95.org>.

<опция> – если этот параметр задан как g, то будет выполнен только перевод текстов программы с языка Гуртран на язык Фор-

тран. Если данный параметр задан как *cl*, то будет выполнена только компиляция полученного ранее текста программы на языке Фортран. Для доступа к этому параметру необходимо ввести предыдущий параметр, то есть символ компилятора. При отсутствии этого параметра выполняется полная генерация программы. Начиная с этого пользователю данный параметр обычно не требуется.

Команда для генерации установленной нами программы MCU-FREE компилятором G95 для использования в однопроцессорном режиме будет выглядеть следующим образом:

```
makemcu.bat f free g
```

Поскольку данная команда должна выполняться непосредственно из папки МАКЕМКУ, то нам придется воспользоваться каким-либо файловым менеджером (например, программой FAR) или сделать это с помощью MCU Office.

Теперь, когда программа сгенерирована, можно приступить непосредственно к расчётам. Рассмотрим общую схему работы программы. Она устроена следующим образом. При запуске исполняемого файла программа ищет в папке, из которой был выполнен запуск, файл с фиксированным именем MCU5.INI. В этом файле в первой строке должно быть записано имя файла исходных данных, во второй – путь к банку ядерных данных, третья строка может отсутствовать, быть пустой либо содержать указание, какой шаг выполнять (I – только ввод, C – только счёт, F – только финальная обработка, B – только шаг выгорания).

Таким образом, запустить программу на счёт можно множеством различных способов, в том числе и средствами MCU Office. Ниже описан один из наиболее часто используемых способов.

Каждый вариант, описанный в файле исходных данных с соответствующим именем, рассчитывается в своей собственной директории, в которую копируется упоминавшийся выше файл *mcu5.bat*. Основной функцией этого файла пакетной обработки является создание файла MCU5.INI и вызова требуемого пользователю исполняемого файла программы (*mcu\_f.exe*, при расчёте с FIMTOEN, или *mcu\_m.exe*, при расчёте с MOFITTG).

Файл *mcu5.bat* запускается из директории с вариантом с помощью командной строки или файлового менеджера. Эта команда имеет вид

```
mcu5.bat <M или F> <имя варианта> <опция>
```

<M или F> – определяет, какой подмодуль физического модуля будет использоваться в области в области энергии нейтронов ниже 5 эВ. M (или m) задается для использования подмодуля MOFITTG, F (или f) для использования подмодуля FIMTOEN.

<имя варианта> – имя файла исходных данных, содержащего описание рассчитываемого варианта.

<опция> – при отсутствии этого параметра последовательно выполняются все шаги, возможны следующие значения:

I – только ввод;

C – только счёт;

F – только финальная обработка;

B – только шаг выгорания;

D – удалить файлы, созданные при предыдущем запуске.

Пример:

```
mcu5 f inputdata.
```

Все программы MCU поставляются с исходными данными для тестирования правильности установки и генерации программы. В MCU-FREE в директории RUNTEST содержится директория BURNUP, содержащая вариант burnup, предназначенный для тестирования моделирования переноса нейтронов и выгорания ядерного топлива в ячейке реактора типа ВВЭР. Для запуска тестового расчёта необходимо запустить файл runtest.bat, находящийся в той же директории.

После запуска файла на счёт на экран выводится некоторая информация о расчёте, а в директории с вариантом образуются файлы. Тестирование можно признать успешным, если после окончания расчёта в этой папке имеется файл с именем файла исходных данных, к которому добавлено расширение FIN. В нашем случае наличие файла burnup.FIN указывает на то, что мы всё правильно установили и сгенерировали. Если программа закончила свою работу, а такой файл не образовался, то необходимо проверить правильность выполнения процедуры установки.

Наиболее распространенная ошибка при установке состоит в том, что неверно указаны пути в файле mcu5.bat. В этом случае программа не находит исполняемый файл или библиотеку. В первом случае в начале счёта на экран выводится сообщение о том, что не найден файл mcu\_f.exe (или mcu\_m.exe, в зависимости от вы-

бранного подмодуля физического модуля), а во втором – в файл burnup.LST поступает сообщение о том, что не найден файл DEFAULT.PHY.

После успешного завершения тестирования программа готова к использованию.

## **ГЛАВА 8. ОБЩАЯ СХЕМА РАСЧЁТА**

Для проведения расчёта следует подготовить файл с исходными данными варианта. Рекомендуется задавать имя файла длиной от 1 до 8 символов без расширения. Полные имена всех рабочих файлов задачи, создаваемых программой в папке, из которой она была запущена, образуются из данного имени с помощью специальных расширений.

Рекомендуется для каждого расчёта создавать отдельную папку.

Во всех рассматриваемых ниже примерах файл исходных данных будет называться NAME.

Расчёт одного состояния методом Монте-Карло осуществляется как последовательное выполнение трёх шагов с условными названиями INPUT, CALCULATION и OUTPUT. При расчётах с изменением изотопного состава материалов реактора к трём шагам расчёта одного состояния добавляется шаг BURNUP. (Описание этого шага дается для ознакомления и подробно рассматриваться не будет.)

### **8.1. ШАГ INPUT**

На шаге INPUT производится ввод исходных данных, подготовка и запись в рабочие файлы информации для проведения расчёта. С запуска этого шага начинается собственно расчёт. Нормальное завершение работы шага является необходимым условием для запуска и работы остальных шагов. Поэтому после окончания шага рекомендуется просмотреть файл NAME.LST, в который заносится вся диагностика, предупреждения (WARNING) и сообщения об ошибках (ERROR) в исходных данных. При обнаружении ошибок их необходимо исправить и произвести запуск шага заново. Если шаг INPUT отработал без ошибок, то могут быть последовательно выполнены два следующих шага: CALCULATION и OUTPUT.

После окончания работы шага INPUT в директории, содержащей файл исходных данных для варианта, образуются следующие файлы:

NAME.LST – текстовый файл, который содержит контрольные сообщения о прохождении задачи, диагностическую печать и прочую информацию;

NAME.DAT – текстовый файл, содержащий только те строки из файла NAME, которые могут быть изменены на шагах CALCULATION и OUTPUT;

NAME.PMC – бинарный файл, содержимое которого в дальнейшем не меняется;

NAME.MCU – бинарный файл, предназначенный для последующей записи результатов работы программы на этапе CALCULATION;

NAME.SYS – служебный файл, содержащий информацию о ходе выполнения расчёта.

## **8.2. ШАГ CALCULATION**

На шаге CALCULATION осуществляется непосредственное проведение расчёта варианта. Режим работы шага определяется содержанием файла NAME.DAT, который создается автоматически из исходного файла NAME на шаге INPUT. Перед началом выполнения шага пользователь может переопределить значения управляющих параметров, редактируя файл NAME.DAT.

Во время работы на шаге CALCULATION в файл NAME.MCU происходит периодическая запись информации, накопленной в результате моделирования. Это позволяет многократно прерывать и возобновлять расчёт без потери накопленных за время счёта результатов. Продолжение расчёта осуществляется просто повтором команды на запуск расчёта.

Нельзя прерывать счёт во время сохранения промежуточной информации в файл NAME.MCU, о чем свидетельствует выводимая на экран надпись: «Attention! Saving to disk. Break is forbidden until finished». Обязательно необходимо дождаться появления надписи «Finished». В противном случае все накопленные результаты будут потеряны, а продолжение расчёта невозможно.

В файл NAME.LST поступают сообщения о прохождении задачи.

## **8.3. ШАГ OUTPUT**

На шаге OUTPUT происходят финальная обработка и печать результатов расчёта в файл NAME.FIN. Режим работы шага опреде-



ляется содержанием файла NAME.DAT. Перед выполнением шага OUTPUT пользователь может переопределить значения некоторых параметров редактированием файла NAME.DAT.

Шаг может быть выполнен после первой записи результатов в файл NAME.MCU на шаге CALCULATION. Как правило, это рекомендуется делать только после того, как будет накоплена значимая статистика.

Если после анализа результатов, оказывается, что расчёт необходимо продолжить, то нужно увеличить количество заказанных серий в файле NAME.DAT и повторить команду на запуск расчёта.

#### **8.4. ШАГ BURNUP**

На шаге BURNUP производится расчёт новых составов материалов. Новые составы записываются в формате, пригодном для использования на шаге INPUT.

Если после окончания шага достигнуто заданное в исходных данных время работы реактора, программа выходит из цикла, в противном случае осуществляется переход к шагу INPUT.

В процессе работы после окончания шага OUTPUT часть полученных к данному моменту времени файлов копируется в файлы, имеющие такое же имя, но к расширению добавляется окончание вида ‘\_Vnnn’, где nnn – это количество шагов BURNUP, завершившихся к данному моменту времени.

### **ГЛАВА 9. РЕЖИМ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛЕНИЙ**

Распараллеливание программы осуществлено на базе программного интерфейса Message Passing Interface (MPI). Он является наиболее распространенным стандартом интерфейса обмена данными в параллельном программировании, и его реализации существуют для большого числа компьютерных платформ. Для установки программы на многопроцессорную ЭВМ достаточно наличия на ней любой реализации MPI с любым компилятором Fortran 90/95, при условии, что с их помощью можно скомпилировать работающее в многопроцессорном режиме приложение.

При работе в режиме многопроцессорных вычислений программа задействует для расчёта все доступные ей процессоры (точнее, ядра; далее в тексте процессоры предполагаются одноядерными).

Коэффициент распараллеливания программы в отсутствие промежуточных записей равен 1.

Общая схема расчёта при этом остается такой же, как и при расчёте на одном процессоре.

Процессоры, доступные задаче, получают при запуске логические номера: 0,1, ... Они имеют смысл лишь для этой задачи.

Основным процессором является нулевой процессор. Помимо собственно счёта он контролирует прохождение задачи. Таким образом, однопроцессорный расчёт с точки зрения программы это расчёт, выполняемый только на нулевом процессоре.

Все файлы, создаваемые ненулевыми процессорами, получают дополнение к расширению вида ‘\_Pm’, где m – номер процессора.

При запуске программы на счёт пользователь указывает количество доступных программе процессоров. Шаги CALCULATION и BURNUP выполняются на заданном количестве процессоров. При продолжении расчёта (CALCULATION) необходимо использовать то же количество процессоров, что и при первичном счёте. При задании другого количества процессоров программа не выйдет на счёт. Шаги INPUT, OUTPUT могут выполняться на одном процессоре при этом счёт можно осуществить на любом количестве процессоров. Это может быть полезно при расчётах на кластерах с высокой загруженностью очереди, поскольку позволяет выполнять отдельные этапы на одном процессоре, доступ к которому получить, как правило, легче.

Количество серий для моделирования, заданное в исходных данных, относится к *одному* процессору. Общее количество серий, которое будет промоделировано, получается умножением заданного количества серий на заказанное количество процессоров.

Продолжение счёта выполняется так же, как и в случае нераспараллеленных вычислений.

На этапе CALCULATION каждый процессор получает свою собственную последовательность случайных чисел, не пересекающуюся с последовательностями других процессоров. Иными словами, на каждом процессоре выполняется независимый расчёт. Это накладывает на пользователя необходимость разумного выбора количества серий для моделирования на каждом процессоре, исходя из необходимости получения разумного ответа на одном процессоре. В качестве примера неправильного задания можно привести

ситуацию, когда при расчёте на  $k_{eff}$  для каждого процессора задается расчёт только одной серии, а общее количество процессоров задается большим. В этом случае суммарное число просчитанных историй будет велико, однако результат на каждом процессоре и, следовательно, на их совокупности будет определяться распределением нейтронов источника, работающего в первом поколении, а не нейтронами деления. Таким образом, на каждом процессоре должна быть накоплена значимая статистика.

Сбор информации, накопленной процессорами, осуществляется в моменты промежуточной и финальной записи на диск. В момент промежуточной записи происходит наибольшая потеря скорости вычислений, поскольку возникают затраты времени на синхронизацию процессоров (ожидание самого медленного) и на собственно запись на диск. Однако поскольку это событие регулируется пользователем и является редким вплоть до полного отсутствия, то программу можно эффективно использовать и в системах с медленными дисками, и межпроцессорными обменами памятью.

Пользователю рекомендуется выбирать максимально возможный, однако приемлемый с точки зрения аварийного прерывания счёта интервал для сохранения промежуточных результатов на диск. Данная рекомендация может быть отнесена и к однопроцессорным вычислениям.

Результаты расчётов транспортного модуля всегда относятся к *одному* процессору. Эти данные получены нулевым процессором.

## **ГЛАВА 10. MCU OFFICE**

Основными функциями программного комплекса MCU Office являются:

- визуализация исходных данных для геометрического модуля и генерация сообщений об ошибках, если таковые имеются;
- обеспечение возможности редактирования текста файла исходных данных для MCU;
- просмотр файлов, создаваемых программой MCU в процессе расчёта.

Кроме того, пользователю предоставляется ряд дополнительных возможностей, таких как:

- запуск расчёта в отдельном окне;

- генерация исполняемого модуля программы MCU;
- сравнение двух рисунков;
- графическое отображение результатов расчёта;
- отображение в текстовом виде на фоне рисунка геометрической модели результатов расчёта, а также другой информации о модели;
- визуализация источника, заданного в исходных данных для программы MCU и др.

MCU Office ориентирован на работу в ОС Windows. В Linux и других Unix-подобных системах для запуска этого комплекса можно воспользоваться свободным программным обеспечением Wine, что создаст условия для корректной работы основных функций, связанных с визуализацией исходных данных для геометрического модуля.

Для удобства запуска MCU Office рекомендуется создать на рабочем столе ярлык к программе

C:\MCU-FREE\MCUFREE\MCUOFF3\mcuoffice.exe.

Для этого можно щёлкнуть по правой кнопке «мыши» на рабочем столе, выбрать Создать→Ярлык и следовать указаниям системы.

Рассмотрим сначала генерацию исполняемого модуля программы MCU и запуск расчёта, который осуществляются из MCU Office с помощью диалоговых окон, привычных для пользователя Windows.

На рис. 9 приводится диалоговое окно (предназначенное для генерации программы), которое можно вызвать с помощью пункта меню Starter→Make MCU.

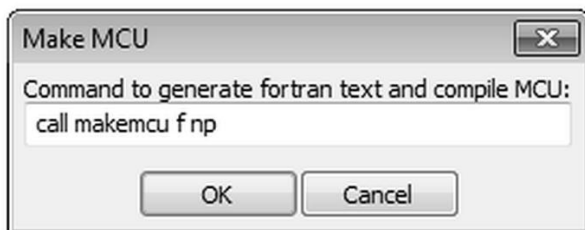


Рис. 9. Диалоговое окно для генерации программы

Единственная строка этого диалога содержит вызов команды генерации исполняемого модуля программы MCU. Использование команды call перед собственно командой генерации необязательно, однако может быть полезным при возникновении ошибок в процессе генерации.

Запуск расчёта осуществляется аналогичным образом с помощью пункта меню Starter→Run MCU, который вызывает на экран диалоговое окно (рис. 10).

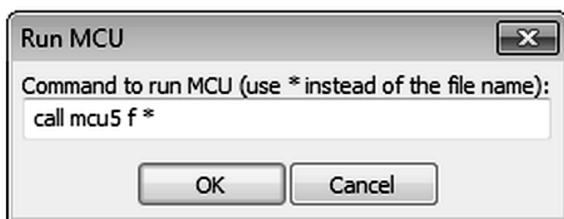


Рис. 10. Диалоговое окно для запуска расчёта

Обратите внимание на то, что вместо имени файла, содержащего исходные данные для расчёта, в команде указан символ \*. Программа автоматически подставит вместо этого символа имя открытого в данный момент в MCU Office файла.

Основной и наиболее частой функцией программного комплекса MCU Office является визуализация исходных данных для геометрического модуля как для их проверки, так и получения рисунков в отчётах, статьях и докладах.

Окно программного комплекса MCU Office (рис. 11) состоит из следующих областей:

- 1) заголовка, в котором отображается имя открытого в программе файла исходных данных для MCU;
- 2) меню (описание команд меню приводится ниже);
- 3) строки состояния для отображения информации;
- 4) первой области строки состояния, в которой отображаются трёхмерные координаты точки, соответствующие текущему положению указателя мыши;
- 5) вторая область строки состояния, в которой отображаются номера материала, зоны и объекта, соответствующие текущему положению указателя «мыши»;

- 6) третьей области строки состояния, в которой указывается тип областей, отображаемых на рисунке (материалы, зоны или объекты);
- 7) горизонтальной полосы прокрутки;
- 8) вертикальной полосы прокрутки.

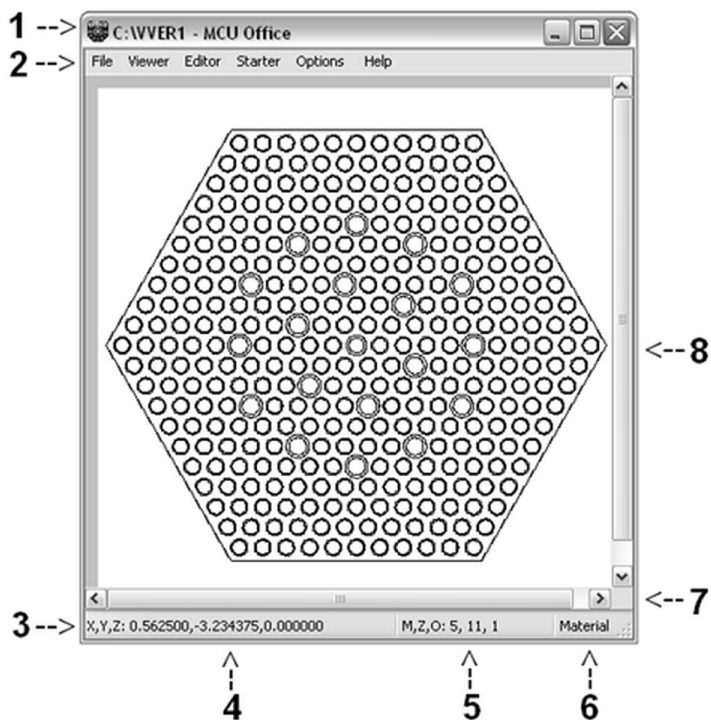


Рис. 11. Окно программного комплекса MCU Office

Описание всех команд меню можно найти в документации по MCU Office, которая поставляется вместе с программой и вызывается с помощью пункта меню Help→MCU Help. Здесь мы рассмотрим лишь базовые понятия, используемые в MCU Office, и основные действия.

Получаемый с помощью MCU Office рисунок – это сечение трёхмерной конструкции произвольным прямоугольником произвольно направленной секущей плоскости.

Параметры такого прямоугольника задаются пользователем либо с помощью указателя «мыши», либо непосредственно численно с помощью диалогового окна Viewer→Data (рис. 12). Это диалоговое окно автоматически появляется на экране сразу после открытия файла исходных данных, которое можно осуществить с помощью пункта меню File→Open, и содержит значения, использованные в последний раз при работе с программой либо принятые по умолчанию.

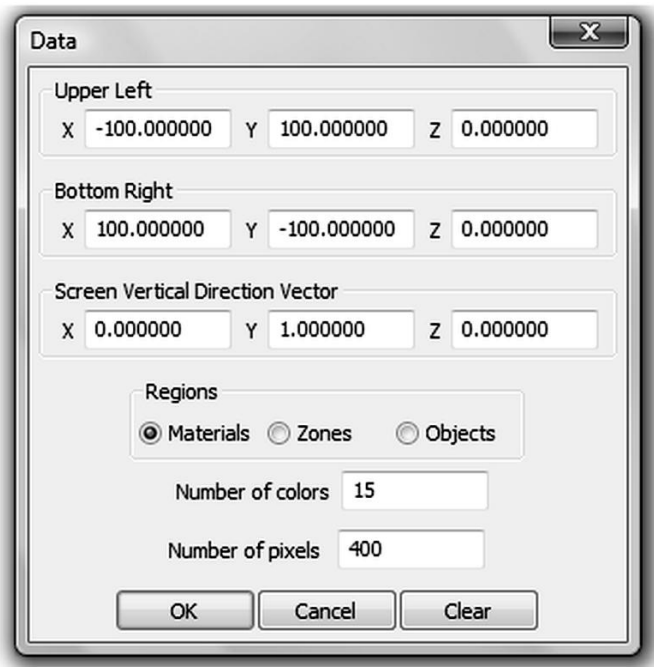


Рис. 12. Диалоговое окно Data

Рассмотрим это диалоговое окно и его параметры более подробно.

Произвольный прямоугольник в трёхмерном пространстве однозначно задается координатами левого верхнего угла (*Upper Left X, Y, Z*), координатами правого нижнего угла (*Bottom Right X, Y, Z*) и координатами вектора, соответствующего направлению сверху вниз на экране (*Screen Vertical Direction Vector X, Y, Z*). Например,

вектор с координатами  $(0,0,1)$  определяет направление сверху вниз на экране, соответствующее оси  $OZ$  в трёхмерном пространстве.

Кнопки *Materials*, *Zones*, и *Objects* служат для отображения на рисунке материальных зон, регистрационных зон или регистрационных объектов соответственно. Подробно такие понятия рассматриваются в главе, посвященной описанию геометрии. Здесь будет достаточно сказать, что материальная зона – это некоторая область пространства, заполненная одним материалом.

Поле *Number of Colors* определяет количество цветов, используемых при создании рисунка. Максимальное число цветов – 15. Цвета перебираются последовательно, начиная с 1 до указанного. Например, если используемое число цветов определено как 3, то соответствие номеров зон номерам цветов будет следующим: зона 1 – цвет 1, 2 – 2, 3 – 3, 4 – 1, 5 – 2... Таким образом, каждая 3-я зона будет закрашена 3-м цветом.

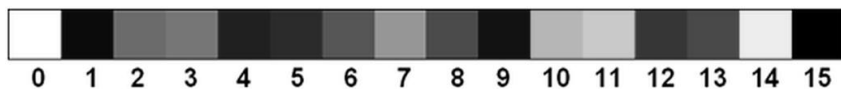


Рис. 13. Цветовая палитра MCU Office (приводится в черно-белом виде)

Число цветов, определенное как 0 или 1, означает, что рисунок будет черно-белым. В этом случае выделяются не области, а границы областей. Отрицательные значения применяются для создания черно-белых рисунков со штриховкой. Абсолютное значение числа цветов в этом случае означает количество используемых штриховок. Отрицательное значение этого поля можно заменить на 0 или положительное, не рекомендуется заменять его на другое отрицательное значение.

Поле *Number of Pixels* определяет длину большей стороны прямоугольника в пикселях.

Для вывода на экран выбранного прямоугольника используется кнопка *Ok*. Если создание рисунка не требуется, то используется кнопка *Cancel*. Для возврата к значениям, принятым по умолчанию (сечение в плоскости  $XY$  при  $Z=0$ , прямоугольник  $400 \times 400$  см), используется кнопка *Clear*.

После того как рисунок был создан хотя бы один раз, следующий прямоугольник можно выбрать с помощью «мыши».



Для выбора прямоугольника на текущей секущей плоскости необходимо двойным нажатием левой клавиши «мыши» выбрать левый верхний угол, после чего также отметить правый нижний. В появляющемся после этого диалоге Data выбрать кнопку Ok. При необходимости до выбора кнопки Ok можно произвести коррекцию данных в диалоге Data. Для отказа от рисования следует выбрать кнопку *Cancel*.

В сущности, таким образом можно выбрать только прямоугольник внутри рисунка на экране. Для того чтобы увидеть, что находится вне этого рисунка, можно воспользоваться пунктом меню Viewer→Zoom→Out или воспользоваться «горячей клавишей» *o*. Отметим команду, противоположную этой, Viewer→Zoom→In, доступную также по «горячей клавише» *i*.

Для поворота секущей плоскости вокруг произвольной оси нужно, удерживая на клавиатуре нажатой клавишу Ctrl, двойным нажатием на левую клавишу «мыши» начать выбор оси поворота, а затем двойным нажатием на ту же клавишу завершить выбор. В появившемся диалоге указывается угол поворота секущей плоскости. На основании точек начала и конца выбранного отрезка оси поворота автоматически определяются размеры и положение нового прямоугольника на секущей плоскости. Поворот производится по правилу «правого винта».

При перемещении указателя «мыши» по рисунку в первой области строки состояния указываются координаты указателя «мыши» в трёхмерном пространстве, а во второй области строки состояния – соответствующие этой точке номера материала, зоны и объекта.

Нажатие правой клавиши «мыши» предоставляет расширенную информацию о выбранной точке: имя файла исходных данных, значение текущих координат курсора в трёхмерном пространстве, номер материала, его состав и др.

При изучении получаемых описанным выше способом сечений часто бывает необходимым измерить расстояние между точками на рисунке. Для этого необходимо установить курсор в начальную точку и, удерживая нажатой правую клавишу «мыши», вести курсор в конечную точку. Во второй области строки состояния будет указываться текущая длина отрезка.

Нажатие клавиши Enter на клавиатуре вызывает перерисовку текущего прямоугольника. При этом если данные были изменены и измененный файл сохранен на диск под тем же именем, то новый рисунок будет содержать внесенные изменения.

Перемещение по истории рисунков назад и вперед осуществляется с помощью пунктов меню Viewer→Backwards и Viewer→Forwards соответственно.

Для практического знакомства с MCU Office можно воспользоваться примерами исходных данных для геометрического модуля MCU, которые приведены в директории

C:\MCU-FREE\MCUFREE\MCUOFF3\VAR.

Помимо генерации двумерных рисунков, MCU Office обладает возможностями для экспорта данных в пакеты, работающие с трёхмерной графикой, а именно: Voxler, ParaView, VisIt, Amira, 3ds Max и его бесплатный аналог gmax.

При подготовке отчётов о выполненных расчётах полезными оказываются функции, позволяющие отобразить полученные результаты на рисунке как графически, так и в текстовом виде. Для этого можно использовать пункты меню Starter→Insert Data (Color) и Starter→Insert Data (Text).

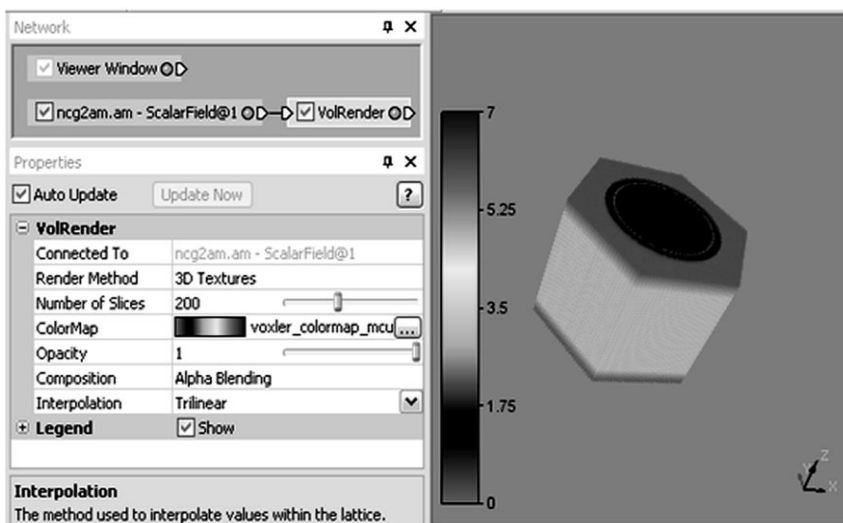


Рис. 14. Пример настроек программы Voxler

## ГЛАВА 11. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ФИЗИЧЕСКОГО МОДУЛЯ

Физический модуль программ, реализующих метод Монте-Карло, служит для ввода, обработки и использования всей информации, связанной с материальными составами и сечениями изотопов, входящих в конструкцию рассчитываемой системы. Основное назначение физического модуля - моделирование взаимодействия частицы с веществом и определение параметров вторичных частиц.

Взаимодействие нейтронов с веществом имеет разные существенные особенности для нейтронов разных энергий. При описании таких взаимодействий принято выделять три энергетические области: быструю, резонансную и тепловую.

Быстрая область характеризуется тем, что в ней имеются неупругие процессы, а именно: неупругое рассеяние, реакции ( $n,2n$ ) и ( $n,3n$ ), а также необходим учёт анизотропии упругого рассеяния в системе центра масс.

Для резонансной области весьма важным является корректное описание сложного вида зависимости сечений от энергии, а именно: разрешённых и неразрешённых резонансов в сечениях захвата, деления и упругого рассеяния.

В тепловой области, которую часто также называют областью термализации при моделировании рассеяния нейтронов, важно учитывать тепловое движение ядер среды, химические связи ядер в кристаллических решетках, жидкостях и молекулах, а иногда и эффекты интерференции нейтронных волн, рассеянных на ядрах.

Физический модуль SOFIZM программы MCU состоит из нескольких подмодулей. Каждый подмодуль осуществляет розыгрыш и обработку столкновений в соответствии с типом частиц и их энергией. Далее будем говорить только о подмодулях, предназначенных для использования при моделировании траекторий нейтронов. Такие подмодули используют различные библиотеки констант и модели описания взаимодействия нейтрона с веществом в разных энергетических областях.

Каждая конкретная программа, собранная из модулей пакета MCU-5, может иметь свой собственный, отличный от других аналогичных программ, набор подмодулей и, следовательно, библиотек. При этом всегда имеется однозначная связь между подмодулем и библиотекой. В определенном смысле характеристики подмодуля

во многом определяются той библиотекой, которой он пользуется. Далее мы последовательно рассмотрим основные библиотеки и использующие их подмодули.

Все библиотеки программы составляют её банк данных и размещаются в собственной поддиректории директории банка программы. Название поддиректории обычно совпадает с названием библиотеки. Имена файлов совпадают с именами нуклидов в системе наименований, принятой в MCU. Согласно этой системе имя нуклида начинается с его обозначения в таблице Менделеева, записанного прописными буквами, к которому добавляется значение числа нуклонов в ядре. Причем если составленное таким образом имя оказывается длиннее четырёх символов, то от значения заряда оставляют только две последних цифры. Например,  $^{249}\text{Pu}$  в этой системе получит имя PU49.

В первую очередь рассмотрим библиотеку ACE/MCU, отсутствующую в программе MCU-FREE, однако важную для описания пакета MCU-5. Библиотека получена из файлов ENDF/B-VII.0 и других источников с помощью программы NJOY [8] с незначительным изменением формата, в котором выдаются данные о сечениях. Основное отличие формата, используемого библиотекой ACE/MCU, состоит в переходе с двойной точности представления значений в файле на одинарную, что практически не влияет на результаты моделирования.

В библиотеке ACE/MCU содержится полная информация о взаимодействии нейтронов с ядрами в поточечном представлении в энергетической области от 0 до 20 МэВ, за исключением законов рассеяния нейтронов в области термализации, когда необходимо учитывать химические связи или кристаллические структуры, например, при расчёте движения нейтронов в графите. Это дает возможность моделировать взаимодействие нейтронов со средой с максимально возможной степенью детализации, используя непосредственно информацию, содержащуюся в файлах оценённых нейтронных данных.

Библиотека состоит из файлов вида <имя нуклида>.<расширение>. В каждом таком файле содержится вся информация об одном нуклиде (изотопе или элементе). Имя файла совпадает с именем нуклида. В качестве расширения файла используют закодированные названия источников первичной информации.

Например, файл U235.E70 содержит данные для 92-U-235, полученные из ENDF/B-VII Release 0. Эта информация подтверждается и комментарием во второй строке файла. Все файлы, относящиеся к одной температуре материала, располагаются в одной поддиректории библиотеки, название которой и определяет температуру, для которой эти файлы были получены.

Обычно в программе MCU библиотеку ACE/MCU рекомендуется использовать для моделирования переноса нейтронов в области энергий выше 100 кэВ. Использование границы в 100 кэВ обусловлено тем, что в стандартной компоновке программ на базе пакета MCU-5 файлы библиотеки рассчитаны только для температуры 300 К и, кроме того, имеют незаблокированные сечения в области неразрешённых резонансов. При расчётах установок, в которых преобладают медленные нейтроны, такая граница позволяет избежать погрешностей, вызванных указанными ограничениями. При расчётах систем с другим спектром нейтронов либо для понижения указанной границы необходимо использовать библиотеку с расширенным набором температур и более точным описанием области неразрешённых резонансов, которое достигается добавлением в файлы библиотеки так называемых вероятностных таблиц Левита. Такие возможности в настоящее время уже имеются в некоторых коммерческих программах на базе пакета MCU-5.

С библиотекой ACE/MCU работает подмодуль FARION. Подмодуль самостоятельно выбирает из имеющихся в библиотеке значений температуры наиболее близкое к заданному в исходных данных.

Далее рассмотрим библиотеку BNAB/MCU. Константы этой библиотеки описывают взаимодействие нейтронов с ядрами в идеологии системы 26-групповых констант БНАБ [9] в энергетической области от нуля до 10,5 МэВ с нижней группой от нуля до 0,215 эВ. Области как разрешённых, так и неразрешённых резонансов в БНАБ описываются с помощью подгрупп. Важно отметить, что библиотека BNAB/MCU не содержит зависимости от температуры.

При наличии нескольких близко расположенных резонансов поглощения внутри одной энергетической группы невозможно точно описать зависимость сечения от энергии. Однако возможно оценить суммарную длину энергетических отрезков, на которых сече-

ние находится в данном диапазоне ( $S_i, S_{i+1}$ ). Если считать плотность вероятности для энергии внутри группы постоянной, то для нейтрона внутри группы можно с вероятностью отношения этой суммарной длины к длине всей группы выбрать среднюю точку отрезка ( $S_i, S_{i+1}$ ) в качестве сечения. Такие значения и составят подгруппу с номером  $i$ . В действительности плотность вероятности не совсем постоянна, но схема остается той же – для каждой подгруппы определена вероятность и отрезок сечений.

В программе MCU библиотека BNAB/MCU используется подмодулем FIMBROEN при моделировании взаимодействия нейтронов с веществом в энергетической области от 10,5 МэВ до границы области термализации, обычно задаваемой в исходных данных для MCU как 4,65 эВ. Сечения для элемента рассчитываются подмодулем как сумма сечений составляющих его изотопов. Содержание изотопов в их естественной смеси приводится в текстовом файле mixtures.bna.

Для моделирования взаимодействия нейтронов с веществом в области термализации при расчёте рассеяния важно учитывать тепловое движение и химические связи ядер вещества рассеивателя, а также эффекты интерференции нейтронных волн, рассеянных на разных ядрах. Тепловые сечения рассеяния кристаллических замедлителей в форме вероятностных таблиц хранятся в библиотеке VESTA. Данные этой библиотеки представлены в энергетической области от  $1,0 \times 10^{-5}$  до 4,65 эВ. Эта информация позволяет рассматривать непрерывное изменение энергии медленных нейтронов и моделировать их рассеяние методом вероятностных таблиц, а также учитывать корреляции между изменением энергии и углом рассеяния. Вероятностные таблицы составлены с применением аналитической процедуры, использующей информацию о законе рассеяния нейтронов  $S(\alpha, \beta)$  в области термализации.

Во многих программах, созданных на базе пакета MCU-5, реализована возможность автоматической подготовки данных библиотеки VESTA (с помощью обращения к подмодулю STEN по информации из библиотеки фононных спектров BOFS) для любой температуры. Эта возможность реализована следующим образом. При отсутствии в VESTA в пределах заданного допуска информации по требуемым температурам для того или иного изотопа программа

автоматически подготавливает необходимую информацию. Данная информация сохраняется в файлах в разделе TMPDAT банка данных. При повторной необходимости в этой информации (например, в следующем расчёте) программа получит её из ранее подготовленных файлов, хранящихся в разделе TMPDAT. Иными словами, при отсутствии информации в основной библиотеке VESTA программа пытается получить информацию из раздела TMPDAT, а в случае отсутствия её и в этом разделе автоматически осуществляет её подготовку, сохранение для дальнейшего использования и ввод.

Библиотека VESTA используется подмодулем FIMTOEN.

Для наиболее точного описания области разрешённых резонансов в банке данных программы MCU имеется библиотека LIPAR. Она даёт возможность по аналитическим формулам вычислить зависимость сечений от энергии нейтрона при любой температуре. Библиотека содержит файлы в двух форматах.

Файлы первого формата имеют расширение LIP. В таких файлах для каждого нуклида записаны параметры его резонансов: энергия, спин, парциальные ширины и другая информация, необходимая для расчёта сечений в формализме Брейта–Вигнера или Адлер–Адлера. Каждый нуклид имеет свою индивидуальную структуру резонансов в области энергий от  $\sim 0,01$  эВ до сотен кэВ. Для описания сечений большинства нуклидов используется многоуровневый формализм Брейта–Вигнера, в котором, кроме интерференции потенциального и резонансного рассеяния, учитывается также интерференция между резонансами. Файлы этого формата представляют собой просто соответствующие разделы файлов оценённых ядерных данных.

Файлы второго формата имеют расширение LPW. В этих файлах зависимость сечений от энергии задана в поточечном представлении. Имеются данные о полном сечении, сечениях рассеяния, радиационного захвата и, если нуклид делящийся, - деления. В отдельной секции файла содержится информация о полном числе вторичных нейтронов деления в зависимости от энергии. Информация для файлов этого формата получена с помощью программы NJOY непосредственно из файлов оценённых ядерных данных, в том числе и использующих формализм Рейха–Мура.

С библиотекой LIPAR работает подмодуль RAPAN. Так же, как и в случае с библиотекой BNAB/MCU и подмодулем FIMBROEN, сечения для элемента рассчитываются как сумма сечений состав-

ляющих его изотопов согласно информации в текстовом файле mixtures.rap.

Пользователь имеет возможность в определённой степени выбирать компоновку подмодулей для каждого конкретного расчёта. Ниже рассмотрим схему, обеспечивающую наибольшую точность расчёта большинства программ, собранных на базе пакета MCU-5.

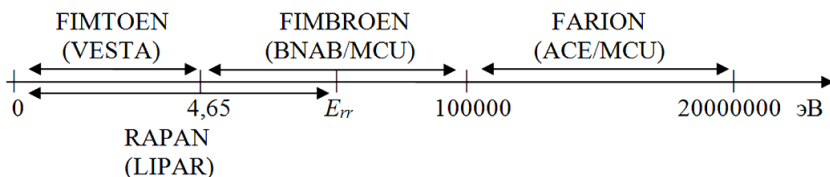


Рис. 15. Схема сшивки подмодулей физического модуля

Программа MCU устроена таким образом, что пользователь может только сдвигать границы сшивки подмодулей FIMTOEN, FIMBROEN и FARION. Управление работой подмодуля RAPAN также возможно, однако это рекомендуется только продвинутым пользователям и практически никогда не используется. Вызов подмодуля RAPAN осуществляется автономно из подмодулей FIMTOEN и FIMBROEN в том случае, если нуклид, на котором происходит взаимодействие, имеет в своих сечениях резонансы и энергия нейтрона ниже заданного значения верхней границы области разрешённых резонансов  $E_{rr}$ .

Схема, приведённая на рис. 15, обеспечивает следующую модель расчёта. Замедлители рассчитываются с учётом всех эффектов до энергии 4,65 эВ с использованием библиотеки VESTA. Взаимодействие нейтронов с нуклидами до границы  $E_{rr}$  моделируется по формулам с использованием данных библиотеки LIPAR или «закона  $1/v$ » ( $\sigma(v) = \sigma_0 v_0 / v$ , где  $v$  – скорость нейтрона,  $v_0 = 2200$  м/с,  $\sigma_0$  – сечение при  $v = v_0$ ) для нейтронов с энергией менее 4,65 эВ, либо в групповом и подгрупповом приближениях BNAB/MCU для нейтронов с энергией более 4,65 эВ. Область неразрешённых резонансов рассчитывается с помощью подгруппового приближения библиотеки BNAB/MCU до 100 кэВ. Далее до 20 МэВ используется поточечное представление библиотеки ACE/MCU. При отсутствии в компоновке программы библиотеки ACE/MCU всё моделирова-



ние траекторий нейтронов с энергиями выше границы сшивки подмодулей FIMTOEN и FIMBROEN определяется библиотекой BNAB/MCU. В этом случае будет использоваться групповое приближение, а максимальная моделируемая энергия нейтрона составит 10,5 МэВ. Именно такая ситуация имеет место при использовании программы MCU-FREE.

Исходные данные любого модуля программы MCU состоят из отдельных предложений, каждое из которых включает одну или нескольких строк. Первый символ первой строки всегда отличен от пробела. С него начинается идентификатор из не более чем шести букв и цифр, называемый меткой предложения. Он оканчивается пробелом или концом строки. Далее идут параметры предложения, состоящие из отдельных фрагментов, разделённых пробелами или запятыми. Если предложение не помещается в одной строке, то можно использовать дополнительные строки. Первая позиция дополнительной строки должна быть пробелом. Символ ; означает, что дальше идет комментарий. Символ \* в первой позиции означает, что вся строка является комментарием. Данные для каждого модуля MCU заканчиваются предложением с меткой FINISH.

Основным понятием, используемым в физическом модуле, является понятие материала. Считается, что вся система состоит из задаваемого пользователем набора материалов. Материал характеризуется номером, температурой и нуклидным составом, то есть всеми нуклидами, входящими в состав данного материала с указанием их концентраций.

Фрагмент исходных данных для физического модуля должен начинаться с обязательного предложения PIN и заканчиваться обязательным предложением FINISH.

Предложение PIN имеет два численных параметра. Первый отвечает за вывод в файл NAME.LST строк исходных данных по мере их ввода, а второй – за вывод в этот файл дополнительной информации. При проведении сложных расчётов это предложение часто задается следующим образом:

```
PIN 1 1
```

При таком использовании в NAME.LST удобно читать диагностику при наличии ошибок в исходных данных, а также контролировать нуклидный состав материалов и заданные для каждого нуклида параметры расчёта, в том числе и принятые по умолчанию.

При отсутствии необходимости в таком контроле выдачу можно отключить, обнулив соответствующий параметр.

Единственным обязательным разделом во фрагменте исходных данных для физического модуля является раздел с описанием материалов. Остальные разделы могут отсутствовать. В этом случае определяемые на уровне ввода значения соответствующих параметров будут приняты программой по умолчанию.

Общий вид задания раздела следующий:

```
<описание первого материала>
```

```
...
```

```
<описание последнего материала>
```

Каждый материал характеризуется своим индивидуальным номером, температурой, нуклидным составом.

Полное описание материала можно найти в Руководстве пользователя, здесь мы ограничимся только основными, наиболее часто используемыми параметрами. При таком подходе общий вид формата описания материала будет выглядеть следующим образом:

```
MATR number T=t GROUP=group NAME=type DENSAA=densaa  
DENSWA=denswa DENSAAW=densaw DENSWW=densww  
name dens MODS=mods DTEM=dtem  
END
```

*MATR* – имя заголовка описания материала.

*number* – номер материала, равный порядковому номеру следования материала в разделе, начиная с 1; **пропуск номеров не допускается**; константа целого типа.

*t* – температура материала в градусах Кельвина; константа вещественного типа. Значение по умолчанию равно 300 К.

*group* – имя группы, к которой относится материал; константа символьного типа. При наличии этого параметра указанный номер материала (*number*) считается внутренним для указанной группы. Материал будет перенумерован автоматически в соответствии с его порядковым номером в списке материалов. Имя группы и указанный в этом предложении номер материала могут быть использованы при описании геометрии системы. По умолчанию материал не относится ни к какой группе.

*type* – тип имен нуклидов в материале; константа символьного типа:

MCU – стандартные 4-символьные имена, принятые в MCU (значение по умолчанию);

ZA – имена вида ZZZAAA, перекодировка в стандартное 4-символьное имя осуществляется по информации из файла AW.LIB из банка данных программы (например, для  $^{12}_6\text{C}$  задание ZZZAAA как 6012 или 006012 соответствует заданию стандартного для MCU имени C12, ZZZ означает заряд ядра, AAA – число нуклонов в нем).

*densaa* – ядерная плотность материала, ядер/см<sup>3</sup> (указывается без множителя 10<sup>24</sup>); предполагается ввод атомных долей нуклидов в материале; константа вещественного типа.

*denswa* – плотность материала, г/см<sup>3</sup>; предполагается ввод атомных долей нуклидов в материале; константа вещественного типа.

*densaw* – ядерная плотность материала, ядер/см<sup>3</sup> (указывается без множителя 10<sup>24</sup>); предполагается ввод весовых долей нуклидов в материале; константа вещественного типа.

*densww* – плотность материала, г/см<sup>3</sup>; предполагает ввод весовых долей нуклидов в материале константа вещественного типа.

Параметры DENSAA, DENSWA, DENSAW, DENSWW являются необязательными и могут не задаваться. Если они используются, то задается только один из этих четырех параметров. В том случае, если их несколько, то программа будет использовать последний.

*name* – имя нуклида в составе материала, определенное в файле DEFAULT.PHY; константа символьного типа.

*dens* – при отсутствии параметров DENSAA, DENSWW, DENSAW или DENSWA определяет число ядер нуклида в одном кубическом сантиметре (**указывается без множителя 10<sup>24</sup>**); константа вещественного типа. При наличии параметра DENSAA или DENSWA определяет атомную долю нуклида в материале, при наличии параметра DENSAW или DENSWW – весовую долю нуклида в материале. В этих случаях автоматически выполняется нормировка долей на их сумму по всем нуклидам материала и с помощью информации из файла AW.LIB по заданной для этого материала плотности определяется число ядер каждого нуклида в одном кубическом сантиметре.

*mods* – определяет модель расчёта сечений рассеяния в области термализации; используется подмодулями FIMTOEN и MOFITG; константа символьного типа:

G – модель идеального одноатомного газа;  
T – модель тяжелого газа;  
CONR – расчёт сечений для бериллия и графита с учётом когерентных эффектов при упругом рассеянии  
H2OK – расчёт сечений для атома H с учётом химической формулы соединения H<sub>2</sub>O;  
CH2K – расчёт сечений для атома H с учётом химической формулы соединения полиэтилен;  
ZRHK – расчёт сечений для атома H с учётом химической формулы соединения ZrH;  
HYH – расчёт сечений для атома H с учётом химической формулы соединения YH (только для подмодуля MOFITTG);  
D2OK – расчёт сечений для атома D с учётом химической формулы соединения D<sub>2</sub>O;  
BEOK – расчёт сечений для атома Be и атома O с учётом химической формулы соединения BeO.

*dtem* – допуск по температуре (градусы Кельвина); определяет интервал [ $T_K - dtem, T_K + dtem$ ] поиска констант нуклида для температуры  $T_K$  в библиотеках TERCON и VESTA; константа вещественного типа.

END – признак окончания описания материала. Это предложение может быть пропущено, если следующим предложением является предложение MATR или предложение FINISH.

Отметим ряд важных особенностей записи указанных предложений.

Имя нуклида *name* записывается с первой позиции, а за ним (через пробел) – его ядерная плотность *dens*. Если на одной строке расположено больше одного нуклида, то перед описанием последующего нуклида ставится знак «слеш» ( $'/'$ ). Начинающему пользователю рекомендуется каждый нуклид помещать в отдельной строке.

При вводе исходных данных обязательным является задание имени нуклида *name* и его ядерной плотности *dens*. Остальные параметры могут не задаваться, и тогда им будет присвоено стандартное значение (значение по умолчанию) из текстового файла DEFAULT.PHY, который **не подлежит редактированию** пользователем и расположен в банке данных программы. Описание этого

файла, как и описание файла AW.LIB можно найти в Руководстве пользователя.

Вне разделов с описаниями материалов могут находиться предложения, необходимые для изменения принятых по умолчанию значений. Здесь рассмотрим основные: DEF, TEMPR, DELN и EGRC.

При необходимости изменить в конкретном расчёте значение по умолчанию, заданное в файле DEFAULT.PHY и не подлежащее редактированию пользователем, следует пользоваться предложением DEF.

```
DEF name MODS=mods DTEM=dtem
```

DEF – имя предложения;

*name* – имя нуклида в составе материала, определённое в файле DEFAULT.PHY; константа символьного типа;

*mods*, *dtem* – новые значения по умолчанию соответствующих параметров для нуклида *name*, полный список которых можно найти в Руководстве пользователя.

Все предложения DEF должны располагаться в разделе данных для физического модуля *до описания первого материала*. Можно применять любое количество таких предложений. Если для какого-либо параметра одного нуклида было задано несколько различных значений, то используется последнее заданное значение.

С помощью предложения TEMPR можно изменить значение температуры в градусах Кельвина, приписываемое всем материалам и равное 300 К. Для материалов, содержащих в описании конструкцию 'T=...', значение, заданное предложением TEMPR, не используется. Отметим, что при наличии в файле исходных данных более одного предложения TEMPR используется значение, заданное в последнем по счёту предложении.

```
TEMPR value
```

TEMPR – имя предложения;

*value* – значение температуры, константа вещественного типа.

Предложение DELN служит для определения режима расчёта энергий нейтронов деления. Общий вид предложения следующий:

```
DELN valdeln
```

При *valdeln*=0 не производится деления нейтронов на мгновенные и запаздывающие. При *valdeln*=1 такое деление производится, и физическим модулем дополнительно вырабатывается

энергия в соответствии со спектром запаздывающих нейтронов. По умолчанию значение равно 0.

Предложение EGRC служит для определения энергетических границ областей работы нейтронных подмодулей составного физического модуля. Иными словами, с его помощью задаются границы сшивки подмодулей FIMTOEN (MOFITTG), FIMBROEN и FARION:

`EGRC valegrc1,valegrc2,valegrc3`

*valegrc1* – нижняя энергетическая граница (эВ) области работы нейтронного подмодуля FARION; константа вещественного типа;

*valegrc2* – нижняя энергетическая граница (эВ) области работы нейтронного подмодуля FIMBROEN; константа вещественного типа;

*valegrc3* – нижняя энергетическая граница (эВ) области работы нейтронного подмодуля FIMTOEN (MOFITTG); константа вещественного типа.

По умолчанию: *valegrc1*=20 МэВ, *valegrc2*=1 эВ, *valegrc3*=0. Эти значения указывают границы работы для подмодуля FIMBROEN от 20 МэВ (а фактически от 10,5 МэВ в связи с используемой версией библиотеки BNAB/MCU) до 1 эВ, а для FIMTOEN – от 1 до 0 эВ.

При параметрах массива EGRC, задаваемых **по умолчанию, подмодуль FARION не используется** в расчётах. Для включения этого подмодуля в расчёты необходимо задать *valegrc1*<20 МэВ.

Значения по умолчанию ведут свою историю из тех времен, когда модуль FARION в расчётах использовался крайне редко, а основным нейтронным подмодулем, применяемым в области термализации, был MOFITTG, использующий 40-групповое транспортное приближение и работающий с библиотекой TEPCON. В компоновках с подмодулем MOFITTG граница сшивки *valegrc2* не должна превосходить 1 эВ.

В компоновках с подмодулем FIMTOEN рекомендуемая граница сшивки *valegrc2* составляет 4,65 эВ.

Важно отметить, что значения параметров предложения EGRC должны совпадать с границами энергетических групп библиотеки BNAB/MCU, поскольку в противном случае подмодуль FIMBROEN не сможет работать.

На самом деле начинающему пользователю можно не задумываться о тонкостях настройки предложения EGRC, а выбрать один из двух вариантов её использования.

Для расчётов без подмодуля FARION использовать

EGRC 20.0E+06, 4.65, 0.0,

а для расчётов с подмодулем FARION

EGRC 0.1E+06, 4.65, 0.0

Рассмотрим пример задания исходных данных для физического модуля:

```
PIN 1 0
TEMPR 297.
DEF H DTEM=3.
MATR 1
U235 0.0008255 / U238 0.022105
O 0.045861
MATR 2
ZR 0.04273
NB 0.000432
HF 6.6E-6
MATR 3
H 0.0667222
O 0.0333611
MATR 4
AL 1.E-6
END
EGRC 20.0E+6, 4.65, 0.
FINISH
```

Этот пример основан на реальной расчётной модели экспериментального стенда ZR-6. В данном примере заданы материалы топлива, оболочки, теплоносителя и зазора между оболочкой и теплоносителем.

Топливо  $UO_2$  задано как материал номер один. Оно состоит из  $^{235}U$  с концентрацией  $0,0008255 \cdot 10^{24}$  ядер/ $cm^3$ ,  $^{238}U$  с концентрацией  $0,022105 \cdot 10^{24}$  ядер/ $cm^3$  и кислорода с концентрацией  $0,045861 \cdot 10^{24}$  ядер/ $cm^3$ . Поскольку описание нуклидов  $^{235}U$  и  $^{238}U$  выполнено в одной строке, используется разделитель – знак «слеш» ('/').

Второй материал содержит нуклидный состав оболочки тепло-выделяющего элемента, выполненной из сплава циркония, ниобия и гафния.

Третий материал определяет состав теплоносителя, являющегося одновременно и замедлителем. В качестве такого материала применяют воду. Для водорода в воде важно использовать соответствующую модель замедлителя, а именно MODS=H2OK. В нашем примере мы не указываем этого специально, поскольку для водорода такая модель установлена по умолчанию во всех программах, собранных на базе пакета MCU-5. Однако для других замедлителей умолчания могут различаться в различных программах, поэтому настоятельно рекомендуется проверять умалчиваемую модель по файлу DEFAULT.PHY. При необходимости изменения значения по умолчанию следует пользоваться предложением DEF. Кроме того, если планируется использовать для расчёта несколько различных программ, то имеет смысл указать модель для замедлителя с помощью предложения DEF.

В приведённом примере мы переопределили заданное для водорода по умолчанию значение допуска по температуре с 1 до 3 К. Сделано это потому, что программа MCU-FREE не позволяет досчитывать отсутствующие в библиотеке VESTA данные. Посмотреть, какие температуры доступны для расчётов водорода по модели H2OK, можно непосредственно в файле HQQQH2OK.VST из библиотеки VESTA. Для этого нужно просмотреть раздел температур в конце файла. Он выглядит следующим образом:

```
TEMPERATURES :  
NTEMPR= 7  
0.30000E+03 0.35000E+03 0.40000E+03 0.53300E+03  
0.55500E+03 0.57900E+03 0.60000E+03
```

Видно, что всего файл содержит семь температур, и ближайшая к требуемой температура составляет 300 К. Если не изменять допуск по умолчанию, то программа остановится по ошибке, связанной с отсутствием требуемой температуры в библиотеке.

Четвёртый материал описывает вакуум газового зазора между топливом и оболочкой. В MCU отсутствует материал, соответствующий вакууму, а задание материала без нуклидов вызывает сообщение об ошибке. Традиционно для задания вакуума используют алюминий в незначительной концентрации. Выбор алюминия в ка-



честве заменяющего вакуум нуклида обусловлен тем, что расчёт сечений для алюминия не требует особых затрат.

Предложение EGRC задает схему расчёта без использования модуля FARION и поточечной библиотеки ACE/MCU. Расчёт кристаллических замедлителей, а в данном случае водорода в воде, будет вестись до максимально возможного для библиотеки VESTA значения энергии 4,65 эВ подмодулем FIMTOEN. Моделирование нуклидов, сечения которых имеют резонансы, будет осуществляться от нуля электрон-вольт до верхней границы области разрешённых резонансов с помощью подмодуля RAPAN по информации из библиотеки LIPAR, в области неразрешённых резонансов – по методу подгрупп, а выше границы области неразрешённых резонансов – по групповому подходу подмодулем FIMBROEN с использованием информации из библиотеки BNAB/MCU. Нуклиды, сечения которых не имеют резонансов, моделируются в области энергий от нуля до 4,65 эВ подмодулем FIMTOEN с использованием «закона  $1/v$ », а выше 4,65 эВ – подмодулем FIMBROEN по групповому подходу с использованием информации из библиотеки BNAB/MCU.

Поскольку ни для какого из материалов не указана собственная температура, то для каждого из них будет использоваться температура, принятая по умолчанию, которую мы изменили с помощью предложения TEMPR с 300 на 297 К. Таким образом, температура всех материалов в моделируемой системе задана равной 297 К.

Заметим, что материал теплоносителя можно было задать с использованием значения плотности воды в  $\text{г/см}^3$  и атомных долей водорода и кислорода в молекуле  $\text{H}_2\text{O}$  следующим образом:

```
MATR 3 DENSWA=0.998
```

```
H 2
```

```
O 1
```

Для проверки можно произвести ввод исходных данных примера с новым заданием третьего материала, изменив предложение PIN:

```
PIN 1 1
```

Тогда в файл NAME.LST поступит дополнительная информация и в том числе следующие строки (приводится в сокращённом виде):

```
Debug print for material: 3
```

```
denswa: 0.998
```

```
H 6.67222E-02 E70 H2OK 0 0.00000E+00 3.00000E+00 ...  
O 3.33611E-02 E70 G 2 0.00000E+00 1.00000E+00 ...  
End of debug print for material 3
```

Из данной печати видно, что концентрации водорода и кислорода, которые записаны непосредственно после имен нуклидов и будут использованы программой в расчёте, совпадают с приведёнными в примере. Отметим также, что приводимые далее параметры соответствуют параметрам файла DEFAULT.PHY и печатаются в том же порядке. Второй параметр – это модель, используемая для замедлителя, а именно: установленная по умолчанию для водорода H2OK и газовая для кислорода. Кроме того, обратим внимание на пятый параметр, который определяет допуск по температуре для поиска информации в библиотеке VESTA. Для водорода он был изменён с помощью предложения DEF и поэтому равен 3, в то время как для кислорода он не менялся и равен определённому в DEFAULT.PHY значению по умолчанию 1. Здесь важно отметить, что поскольку расчёт будет выполняться с помощью подмодуля FIMTOEN, допуск по температуре для кислорода не имеет никакого значения, так как его сечения будут рассчитываться подмодулем непосредственно в процессе моделирования по формулам с точным учётом температуры без обращения к библиотеке VESTA.

### **11.1. МЕТОД СУММАРНОГО ИЗОТОПА**

При изменении изотопного состава материалов в результате выгорания образуется значительное количество новых нуклидов. Соответственно при моделировании программа начинает затрачивать на расчёт такого выгоревшего состояния значительно большее количество оперативной памяти и, что более существенно, времени. В MCU существует возможность «спрятать» ряд нуклидов в так называемый суммарный изотоп. Для этого задается список нуклидов, входящих в суммарный изотоп. Обычно это нуклиды, не оказывающие значительного влияния на результаты моделирования конкретной системы. Входящие в суммарный изотоп нуклиды обрабатываются программой иначе, чем нуклиды, не входящие в него. Достаточно подробное описание метода приведено в Руководстве пользователя.

Суммарный изотоп задается предложениями SI, SINOT или SIDEN:

SI list

*SI* – имя предложения;

*list* – список нуклидов, входящих в суммарный изотоп.

SINOT list

*SINOT* – имя предложения;

*list* – список нуклидов, не входящих в суммарный изотоп.

SIDEN value

*SIDEN* – имя предложения;

*value* – все нуклиды с плотностью меньше данного значения входят в суммарный изотоп.

Любой список действует с момента его объявления. Можно использовать любое количество предложений описания суммарного изотопа. Это позволяет задавать состав суммарного изотопа независимо для каждого материала. Можно использовать предложение без указания списка. В этом случае, считается, что с этого момента суммарный изотоп не содержит нуклидов.

При одновременном наличии перед описанием материала списков SI и SINOT активным является тот, который был объявлен последним.

Предложение SIDEN активно всегда, независимо от наличия или отсутствия предложений SI и SINOT.

Рассмотрим применение суммарного изотопа на практике. Предположим, что в результате выгорания топлива в нашем примере его состав стал следующим (в реальности количество нуклидов будет значительно больше):

MATR 1

U235 8.23004E-04

U238 2.21039E-02

O 4.58610E-02

U236 4.32436E-07

NP39 3.03987E-07

PU39 6.38359E-07

SR90 1.22803E-07

Y91 1.17981E-07

ZR92 1.29299E-07

ZR93 1.37112E-07

ZR94 1.39429E-07

ZR95 1.32641E-07

ZR96 1.37782E-07

```

MO97 1.29905E-07
MO98 1.26028E-07
MO00 1.37343E-07
RU01 1.13975E-07
XE34 1.71279E-07
XE36 1.90869E-07
CS37 1.34569E-07
BA38 1.45073E-07
BA40 1.03816E-07
LA39 1.37833E-07
CE41 1.13826E-07
CE42 1.25722E-07
CE44 1.17229E-07
PR43 1.00242E-07

```

Отправим ряд нуклидов в суммарный изотоп. На практике для этого чаще используют предложение SINOT, поскольку гораздо легче сказать, что нужно оставить, чем угадать, что придется выкинуть. Добавим перед предложением MATR 1 предложение

```

SINOT O U235 U236 U238 NP237 NP238 PU38 PU39 PU40
PU41 PU42 AM2M CM45

```

После описания первого материала добавим пустое предложение SINOT, очищая список нуклидов суммарного изотопа для того, чтобы не портить составы остальных материалов.

Для контроля ввода изменим предложение PIN следующим образом:

```
PIN 1 1
```

В результате наш пример будет выглядеть так:

```

PIN 1 1
TEMPR 297.
DEF H DTEM=3.
SINOT O U235 U236 U238 NP237 NP238 PU38 PU39 PU40
PU41 PU42 AM2M CM45
MATR 1
U235 8.23004E-04
U238 2.21039E-02
O 4.58610E-02
U236 4.32436E-07
NP39 3.03987E-07
PU39 6.38359E-07
SR90 1.22803E-07
Y91 1.17981E-07

```

ZR92 1.29299E-07  
ZR93 1.37112E-07  
ZR94 1.39429E-07  
ZR95 1.32641E-07  
ZR96 1.37782E-07  
MO97 1.29905E-07  
MO98 1.26028E-07  
MO00 1.37343E-07  
RU01 1.13975E-07  
XE34 1.71279E-07  
XE36 1.90869E-07  
CS37 1.34569E-07  
BA38 1.45073E-07  
BA40 1.03816E-07  
LA39 1.37833E-07  
CE41 1.13826E-07  
CE42 1.25722E-07  
CE44 1.17229E-07  
PR43 1.00242E-07  
END  
SINOT  
MATR 2  
ZR 0.04273  
NB 0.000432  
HF 6.6E-6  
MATR 3  
H 0.066714  
O 0.033357  
MATR 4  
AL 1.E-6  
MATR 5  
U235 0.0008255  
U238 0.022105  
O 0.045861  
MATR 6  
U235 0.0008255  
U238 0.022105  
O 0.045861  
MATR 7  
U235 0.0008255  
U238 0.022105  
O 0.045861

```
MATR 8
U235 0.0008255
U238 0.022105
O 0.045861
END
EGRC 20.0E+06,4.65,0.0
FINISH
```

Рассмотрим печать в файле NAME.LST (приводится в сокращенном виде):

```
Debug print for material: 1
U235 8.23004E-04 E70 T 4 1.00000E+02 1.00000E+00 ...
U238 2.21039E-02 E70 T 4 4.65000E+03 1.00000E+00 ...
O 4.58610E-02 E70 G 2 0.00000E+00 1.00000E+00 ...
U236 4.32436E-07 E70 T 4 1.00000E+03 1.00000E+00 ...
PU39 6.38359E-07 E70 T 4 4.65000E+02 1.00000E+00 ...
End of debug print for material 1
```

Видно, что в первом материале остались только те нуклиды, которые мы и просили оставить с помощью предложения SINOT, а остальные были свернуты в суммарный изотоп.

Проверим второй материал:

```
Debug print for material: 2
ZR 4.27300E-02 J32 T 2 0.00000E+00 1.00000E+00 ...
NB 4.32000E-04 E70 T 4 4.65000E+03 1.00000E+00 ...
HF 6.60000E-06 J32 T 4 2.15000E+02 1.00000E+00 ...
End of debug print for material 2
```

Видим, что список суммарного изотопа действительно пуст и не влияет на дальнейшие материалы.

В заключение отметим, что при работе с исходными данными для физического модуля можно выполнить коррекцию некоторых данных с помощью внешних текстовых файлов, имена которых совпадают с именем файла исходных данных, к которому они относятся, а расширения имеют вид PDCn (PIN Data Correction), где  $n$  может принимать значения от 1 до 9 или отсутствовать, что соответствует  $n=0$ . Обработка таких файлов производится в порядке увеличения  $n$ , т.е. в первую очередь будет обработан файл с расширением PDC, а в последнюю – PDC9. Информация из этих файлов замещает или дополняет данные, полученные из файла исходных данных. Сами файлы должны располагаться в той же директории, что и файл исходных данных. Следует иметь в виду, что файл с

расширением PDC используется при расчётах выгорания для изменения составов материалов.

Правила записи файлов вида PDCn приводятся в Руководстве пользователя.

## ГЛАВА 12. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ГЕОМЕТРИЧЕСКОГО МОДУЛЯ

Расчёт переноса излучения в программе MCU всегда осуществляется в конечной области пространства. Даже если расчёт производится в бесконечной системе, то вычисления ведутся в конечной области пространства, а бесконечность описывается с помощью задания специальных условий на границе этого пространства.

Конечная область пространства, в которой производится расчёт, называется *контейнером системы*. Она считается разбитой на конечное число областей, называемых *геометрическими зонами*. Внутри контейнера системы не должно быть областей, не входящих ни в одну из геометрических зон.

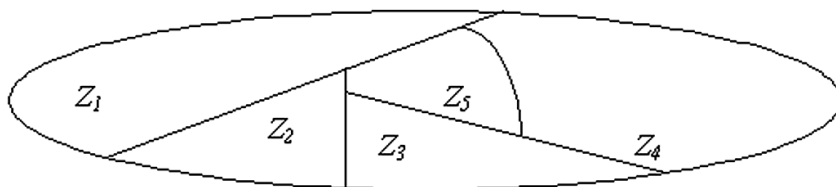


Рис. 16. Пять геометрических зон

Геометрическая зона считается гомогенной, т.е. однородной по своим свойствам. В частности, это означает, что любая геометрическая зона состоит только из одного материала. Каждой геометрической зоне сопоставлен номер соответствующего материала, называемый *материальным номером*. Под *материальной зоной* понимается совокупность всех точек с одним материалом, т.е. материальная зона составляется из набора геометрических зон, взятых целиком. Моделирование траекторий частиц ведется именно с учётом материальных номеров геометрических зон, поскольку именно для них в физическом модуле заданы нуклидный состав и другие характеристики материалов.

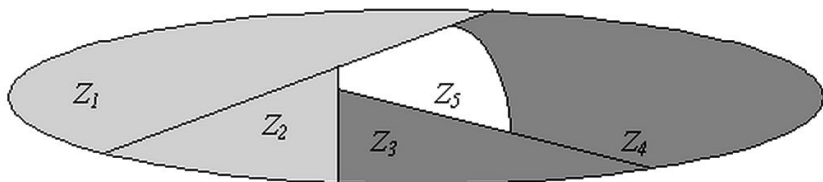


Рис. 17. Три материальные зоны

Для работы регистрационного модуля контейнер системы разбивается на два независимых набора областей, **называемых регистрационными зонами** и **регистрационными объектами**. Номер регистрационной зоны, к которой принадлежит данная геометрическая зона, называется её **регистрационным номером**, номер соответствующего регистрационного объекта – **объектным номером**.

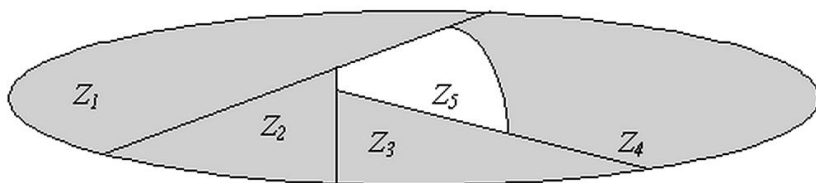


Рис. 18. Две регистрационные зоны

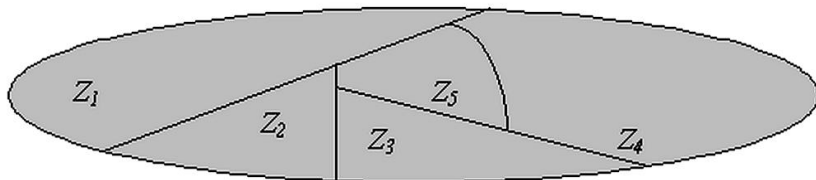


Рис. 19. Один регистрационный объект

Любая регистрационная зона или регистрационный объект может состоять из одной или нескольких геометрических зон.

Каждая геометрическая зона принадлежит только одной регистрационной зоне и одному регистрационному объекту.

Важно понимать, что основным разбиением системы является разбиение на геометрические зоны, а уже для каждой геометриче-



ской зоны задается три числа: номер материальной зоны (материала), номер регистрационной зоны и номер регистрационного объекта. При этом если номера материальных зон непосредственно связаны с устройством моделируемой системы, поскольку несут информацию о её материальном составе, то номера регистрационных зон и регистрационных объектов связаны с тем, в каких областях пользователь хочет получить расчётные значения. Так, на рис. 16 используется пять геометрических зон для задания геометрии системы, состоящей в реальности из трёх материалов и имеющей форму, показанную на рис. 17. Судя по рис. 18, пользователю нужно получить два результата: в центральной зоне, совпадающей с геометрической зоной  $Z_5$ , и в её окружении, то есть во всех остальных геометрических зонах. Кроме того, судя по рис. 19, ему требуется результат для всей системы целиком, поэтому он объединяет в один регистрационный объект все геометрические зоны. Забегая немного вперед, отметим, что в MCU-5 можно получать результат не только для регистрационных зон и регистрационных объектов, но и для материальной зоны, а некоторые характеристики также и для системы в целом, независимо от её разбиения на регистрационные зоны и регистрационные объекты.

Подытоживая, можно сказать, что частицы «летают» по материальным зонам, а результат получается в системе, материальных зонах, регистрационных зонах и регистрационных объектах.

Кроме описания зон должны быть заданы *граничные условия*, то есть каждой точке внешней границы системы должно быть сопоставлено правило, что делать с частицей, покидающей систему через эту точку. Считается, что вся внешняя граница разбита на несколько поверхностей, и в пределах одной поверхности граничное условие одно. Типичными граничными условиями являются утечка, трансляция, зеркальное отражение, белое отражение и так далее.

Таким образом, описать геометрию системы, значит:

- задать все геометрические зоны, в совокупности составляющие всю систему;
- для каждой геометрической зоны задать соответствующий ей номер материальной зоны, регистрационной зоны и регистрационного объекта;

- задать граничные условия на внешних поверхностях системы.

Основная задача геометрического модуля состоит в моделировании прямолинейных отрезков траектории частицы.

В процессе работы геометрический модуль производит следующие расчёты:

- определение номера зоны, в которой находится частица;
- определение точки выхода частицы из текущей зоны;
- определение следующей зоны, в которую попадет частица, или выход частицы за пределы контейнера;
- расчёт оптического пути частицы, то есть определение точки следующего столкновения.

Средства описания геометрии делятся на универсальные и специализированные.

Специализированные средства ориентированы на описание узкого класса систем, например реакторов типа ВВЭР или РБМК. Они имеют следующие преимущества:

- сокращение, по сравнению с универсальными средствами, необходимого количества входных данных за счёт использования специфики данного класса систем;
- возможность более жёсткого контроля правильности вводимых данных;
- меньшее время, затрачиваемое пользователем на освоение;
- простая возможность создания графического ввода информации.

Число разных классов инженерных конструкций, которые требуется описывать, велико, что порождает целый набор таких средств.

При использовании современной техники скорость универсального геометрического вычислителя мало уступает скорости работы специализированного, и поэтому в МСУ-5 применяется универсальный геометрический модуль NCG – геометрический вычислитель универсального типа. Он использует компилятор с универсального языка комбинаторного описания геометрии NCGSIM.

Следует различать программу геометрического модуля, которая производит расчёт, и язык описания геометрии. Несколько разных

программ геометрических модулей могут использовать один и тот же язык описания геометрии. И наоборот, один вычислительный модуль может иметь несколько входных языков.

В модуле NCG нет какой-либо заранее заданной и встроенной в него геометрии той или иной системы. Систему полностью описывает пользователь на языке NCGSIM. Универсальность модуля NCG объясняется именно тем, что с его помощью можно рассчитывать любую заданную систему. Универсальность же языка описания геометрии NCGSIM объясняется использованием для описания произвольной трёхмерной системы средств комбинаторной геометрии. Именно NCGSIM мы и будем рассматривать далее.

Фактически все универсальные геометрические модули, применяемые в программах, предназначенных для моделирования переноса излучения методом Монте-Карло, используют тот или иной комбинаторный способ построения геометрии системы. Обычно различают два основных способа построения комбинаторной геометрии – метод поверхностей и метод тел. Оба способа математически эквивалентны. В NCGSIM используется комбинаторный способ задания с помощью тел.

При таком подходе простейшими областями в комбинаторной геометрии являются так называемые тела: шары, параллелепипеды, цилиндры и так далее. Каждое тело кроме типа имеет параметры, описывающие конкретную форму тела и его положение в пространстве.

Простейший пример тела – шар, который в NCGSIM называется SPH (рис. 20).

Его параметрами являются  $C$  и  $R$ , где  $C$  – радиус-вектор центра, а  $R$  – радиус шара. Вся граница шара представляет одну поверхность.

Другое простое, но часто используемое тело – прямой круговой цилиндр с образующей, параллельной оси  $OZ$ , имеющий имя RCZ (рис. 21).

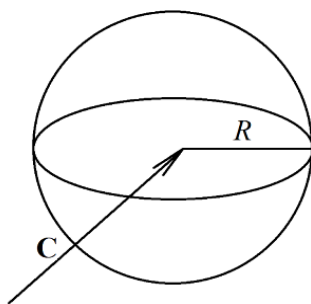


Рис. 20. Шар SPH

Он имеет параметры  $C$ ,  $H$ ,  $R$ , где  $C$  – радиус-вектор центра нижнего основания,  $H > 0$  – высота, а  $R$  – радиус цилиндра. Поверхности цилиндра при необходимости (например, при описании граничных условий) перечисляются в следующем порядке: нижнее основание, верхнее основание, боковая поверхность.

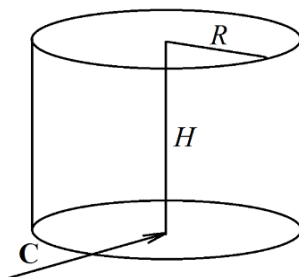


Рис. 21. Цилиндр RCZ

На самом деле RCC это упрощенная форма правильного кругового цилиндра RCC (рис. 22).

Параметры RCC –  $C$ ,  $H$  и  $R$ , где  $C$  – радиус-вектор центра одного нижнего основания,  $H$  – вектор, соединяющий центр нижнего основания с центром другого основания, а  $R$  – радиус цилиндра. Поверхности RCC перечисляются в том же порядке, что и у RCZ.

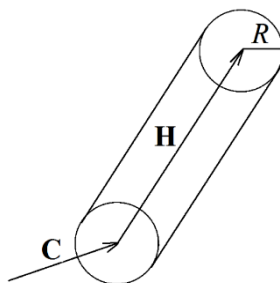


Рис. 22. Цилиндр RCC

Всего в NCGSIM имеется более 20 типов тел. Полное описание тел имеется в Руководстве пользователя.

При описании геометрии возможно использование имён констант. Константы задаются предложениями вида

EQU <имя константы> = <выражение>

<имя константы> начинается с латинской буквы и состоит из не более шести латинских букв или цифр.

Выражение состоит из чисел, имён ранее определённых констант, знаков арифметических операций и обозначений некоторых функций. Например,

EQU A=24/SQRT(9) ; A есть 8

EQU SP=A\*COS(30) ; SP есть 4 корней из 3

Полное описание возможностей можно найти в Руководстве пользователя.

Для построения геометрических зон из тел используют три операции теории множеств: дополнение, пересечение и объединение.

Аргументами этих операций служат тела, которые являются трёхмерными областями.

Дополнение области определено как совокупность точек, не принадлежащих этой области. Считается, что все рассматриваемые области замкнуты, то есть содержат свои граничные точки, поэтому под дополнением понимается совокупность точек, не принадлежащих области, и её граничные точки. Таким образом, граничные точки относятся как к области, так и к её дополнению. Дополнение обозначается знаком «минус» без пробела после него.

Пересечение двух и более областей определено как совокупность точек одновременно принадлежащих этим областям. Нет знака пересечения, символы областей просто разделяются пробелами. Знак пересечения отсутствует.

Объединение двух и более областей определено как совокупность точек, любая из которых принадлежит хотя бы одной из этих областей. Объединение обозначается знаком  $\cup$  (заглавная латинская буква), окруженным пробелами.

Операция дополнения считается старше операции пересечения, то есть выражения  $A - B$  и  $-B A$  понимаются одинаково: как пересечение  $A$  с дополнением области  $B$ . Операция пересечения старше операции объединения, то есть  $A \cup C D$  понимается как объединение  $A$  с пересечением  $C$  и  $D$ .

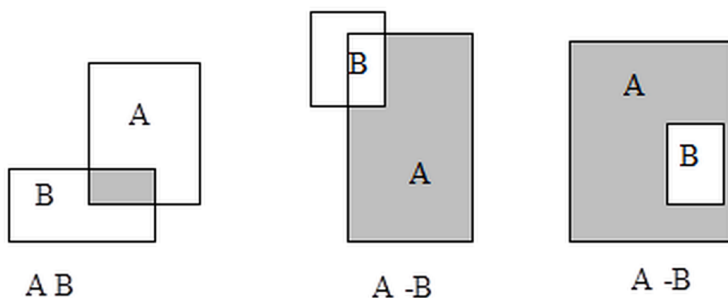


Рис. 23. Использование комбинаторной геометрии

На рис. 23 заштрихованы  $A B$  и  $A - B$  соответственно. Выражение  $A - B$  можно трактовать как разность: совокупность точек  $A$ , не входящих в  $B$ .

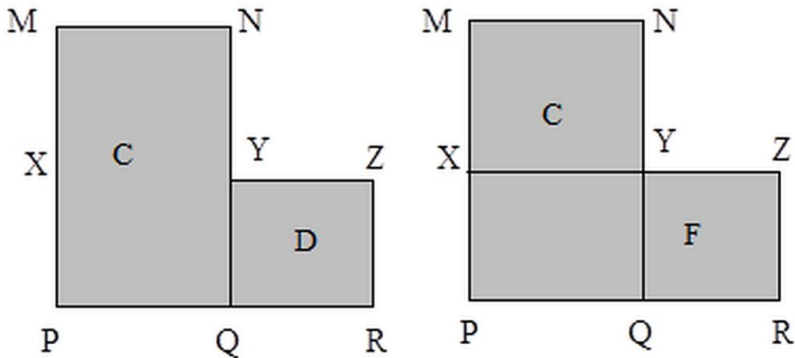


Рис. 24. Использование комбинаторной геометрии

Заштрихованы  $CUD$  и  $CUF$ .

Отметим:  $CUD = CUF$ , хотя  $D$  и  $F$  различны:  $D$  есть прямоугольник  $YZRQ$ , а  $F$  -  $XZRP$ . Вариант  $CUD$  более предпочтителен, так как объединение областей, имеющих общие внутренние точки, приводит к потере быстродействия модуля.

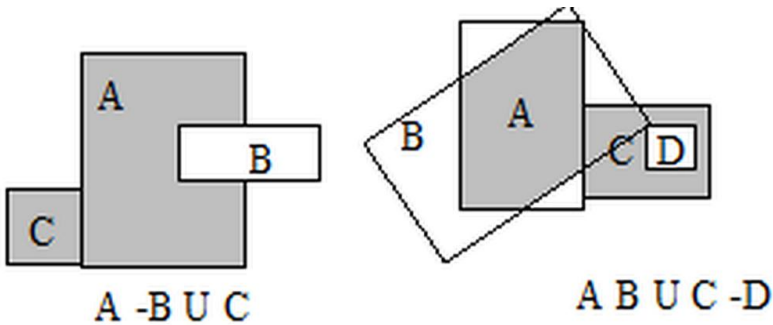


Рис. 25. Использование комбинаторной геометрии

На рис. 25 приведены примеры использования всех трех операций. Заштрихованы результаты.

Часто оказывается удобным объединить в одну зону несколько областей, не имеющих общих точек, даже учитывая граничные точки.

Все геометрические зоны строятся из тел с помощью трёх вышеописанных операций. Это позволяет создавать достаточно слож-

ные формы. При этом некоторые распространенные геометрические объекты приходится задавать не совсем привычным способом (например, труба задаётся как пересечение цилиндра с внешним радиусом трубы и дополнения цилиндра с внутренним радиусом трубы).

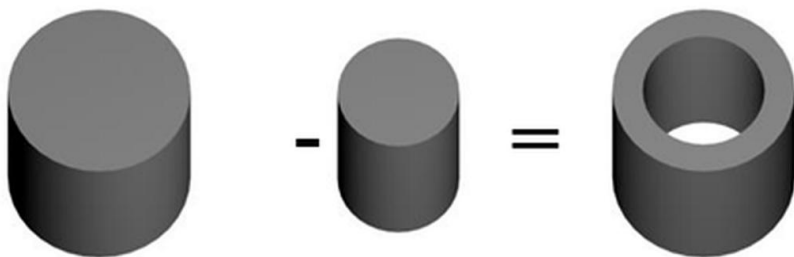


Рис. 26. Описание трубы средствами комбинаторной геометрии

Перейдем непосредственно к описанию моделируемой системы с помощью средств языка NCGSIM.

Первое обязательное предложение данных для геометрического модуля – HEAD <параметры>. Начинаящий пользователь может в качестве параметров использовать 1 0 0. Описание смысла параметров дано в Руководстве.

```
HEAD 1 0 0
```

С помощью второго предложения задаются граничные условия для контейнера системы. Вообще говоря, существует два способа описания контейнера системы. Здесь рассмотрим более простой и наиболее часто используемый второй способ, когда контейнером служит первое тело, описанное далее в секции описания тел. В качестве контейнера можно использовать тела SPH, RCZ и некоторые другие (см. Руководство пользователя). Контейнер задаётся одним предложением (необязательные параметры опущены):

```
CONT <граничные условия>
```

<список граничных условий> состоит из столькох фрагментов, сколько есть граничных поверхностей у тела, служащего контейнером. Если контейнером системы служит шар, то такая поверхность одна, а если это цилиндр, – три. Для каждой поверхности задаётся свое граничное условие. Граничные условия задаются в порядке перечисления поверхностей тела-контейнера.

Рассмотрим три наиболее часто используемых граничных условия:

- В – чёрная граница (все вылетающие частицы поглощаются);
- М – зеркальное отражение;
- Т – трансляционная симметрия.

Трансляционная симметрия должна быть задана согласованно на противоположных гранях. Смысл такой симметрии заключается в том, что частица, вылетающая из системы через, например, верхнее основание цилиндра, влетает в него в противоположной точке нижнего основания с теми же энергией и направлением полета. В определённом смысле происходит как бы сдвиг цилиндра вверх по направлению полёта частицы, поэтому трансляционную симметрию называют также сдвиговой симметрией.

В MCU-5, начиная с версий 2010 года, в отличие от предыдущих пакетов контейнер обрезает все тела, которые выходят за его пределы. За это отвечает предложение CNTAND <value>, которое при его наличии всегда располагается непосредственно перед предложением CONT. Значение <value> может быть единицей, означающей, что принудительное пересечение всех тел с контейнером включено, или нулем, означающим, что оно выключено. При отсутствии предложения или употреблении его без параметров принудительное пересечение всех тел с контейнером включено.

После того как определены граничные условия, необходимо задать все тела, которые будут использованы при описании так называемой глобальной геометрии. При описании геометрии в модуле NCG применяется декартова система координат. Расположение начала системы координат и ориентация осей в пространстве может быть выбрана произвольно, наиболее удобно для пользователя. Эту систему координат далее будем называть *глобальной*. Отдельные элементы при задании многоуровневой (сложной) геометрии могут быть описаны в *локальных* системах координат, отличающихся от глобальной системы точкой начала координат, а в некоторых случаях еще и поворотом.

Часть исходных данных для модуля NCG, в которой описываются тела, называется секцией тел. Первым телом в этой секции должно быть тело, используемое в качестве контейнера.

Каждое тело описывается одним предложением вида  
<метка> <имя тела> <параметры>



<метка> определяет тип тела в виде трёхсимвольного ключа, например, SPH для шара.

<имя тела> служит для ссылок на него. Это - идентификатор из не более шести латинских букв и цифр, начинающийся с буквы. В качестве имени тела нельзя использовать идентификаторы U и T, которые имеют специальный смысл.

<параметры> – константы, включающие числа или выражения, аналогичные используемым при определении констант, по количеству параметров для данного типа тела. Разделителями служат пробелы или запятые (возможно, вместе с пробелами), поэтому в выражениях не должно быть внутренних пробелов.

Например, предложение

```
SPH SPHERE 10 20 30 40
```

задает расположенную в точке (10, 20, 30) сферу с именем SPHERE радиусом 40 см.

Секция тел заканчивается предложением END.

После секции тел следует секция зон. Каждая геометрическая зона описывается одним предложением следующего вида (здесь и далее в квадратных скобках приводятся параметры, которые можно не задавать):

```
<имя зоны> <имя тела> [<знак операции> <имя тела>  
 [<знак операции><имя тела >[...]]]<конечный фрагмент>
```

Как уже говорилось, для построения геометрических зон используются три известные операции теории множеств: дополнение, пересечение и объединение. Аргументами этих операций служат заданные в предыдущей секции тела, которые являются трёхмерными областями.

<имя зоны> - метка предложения, которая служит для ссылки на данную зону. Это любой идентификатор, являющийся последовательностью из не более чем шести латинских букв и цифр, начинающийся с буквы. Все имена зон должны быть различными.

<имя тела> - имя-идентификатор тела, описанного в секции тел, или дополнение к этому телу, если <имя тела> содержит в первой позиции знак минус.

<знак операции> - определяет операцию объединения, которая обозначается буквой U, окружённой пробелами, или операцию пересечения, которая обозначается пробелом или серией пробелов.

<конечный фрагмент> - определяет для данной геометрической зоны номер материала, номер регистрационной зоны и номер регистрационного объекта. Он может определяться двумя альтернативными способами, при этом разные зоны можно описывать разными способами.

Первый способ является наследием и некоторым развитием подхода, принятого в пакете MCU-4. Он обеспечивает возможность использования исходных данных для геометрического модуля, написанных для программ, созданных из модулей пакета MCU-4. При использовании этого способа <конечный фрагмент> имеет несколько форм записи. Рассмотрим две из них. Первая и наиболее часто употребляемая форма выглядит следующим образом:

/<зона>:<материал> [/<объект>]

Здесь <зона> означает номер регистрационной зоны, <материал> - номер материала, <объект> - номер регистрационного объекта для данной геометрической зоны. При отсутствии задания номера регистрационного объекта используется его значение, принятое по умолчанию и равное 1.

Вторая форма, которую мы рассмотрим, удобна в том случае, если нас не интересует разбиение моделируемой системы на регистрационные зоны и регистрационные объекты. Такая ситуация возможна в том случае, если расчёт ведется для получения каких-либо характеристик системы в целом. В этом случае достаточно только указать, какой материал находится в данной геометрической зоне. Для этого используется следующая форма записи:

:<материал>

Здесь <материал> означает номер соответствующего данной геометрической зоне материала. В этом случае значения регистрационной зоны и регистрационного объекта для данной геометрической зоны будут приняты равными 1.

Второй способ разработан в рамках пакета MCU-5 и обеспечивает больше возможностей для моделирования. При использовании этого способа <конечный фрагмент> начинается символом #. Далее, для задания материального, регистрационного и объектного номеров данной зоны, а также других атрибутов используются поименованные параметры. Общий вид задания значения параметра:

параметр = значение

Возможно использование следующих параметров:

- M или m – для задания номера материала;
- G или g – для задания символьного имени группы, к которой относится заданный с помощью переменной M материальный номер. При наличии этого параметра материальный номер будет присвоен соответствующей геометрической зоне следующим образом. В исходных данных для физического модуля выполняется поиск материала с таким же символьным именем группы и номером, после чего определяется его порядковый номер в списке материалов, который и присваивается соответствующей геометрической зоне. Значением параметра является константа символьного типа. При отсутствии параметра G номер материала присваивается геометрической зоне обычным образом без дополнительной обработки.

- Z или z – для задания номера регистрационной зоны;
- или o – для задания номера регистрационного объекта.

Материальный номер должен быть задан обязательно, остальные могут отсутствовать. В случае отсутствия номера регистрационной зоны или регистрационного объекта его значение принимается равным 1.

Секция зон заканчивается предложением END.

На этом задание глобальной геометрии заканчивается. Если модель достаточно проста и сложная геометрия не применяется, то никаких дополнительных данных для геометрического модуля не требуется.

Раздел данных для геометрического модуля заканчивается обязательным предложением FINISH.

Рассмотрим в качестве примера описание заполненной воздухом бесконечной вертикальной трубы с внешним радиусом 7 см и толщиной 2 см:

```

HEAD
CONT T T B
RCZ CylOut 0,0,0 10 7
RCZ CylIn 0,0,0 10 5
END
Tube CylOut -CylIn #m=1 z=1 o=1
Inside CylIn #m=2 z=2 o=1
END
FINISH

```

В качестве начала координат мы выбрали точку (0,0,0). Труба задаётся двумя телами: цилиндром `CylOut` с внешним радиусом трубы 7 см и цилиндром `CylIn` с внутренним радиусом трубы 2 см. По условиям задачи система должна быть бесконечной по высоте, а программа требует выполнять расчёт в конечной системе. Для разрешения этого противоречия используются граничные условия, задаваемые на границах контейнера системы. В данном примере в качестве контейнера системы выбран цилиндр `CylOut`, поэтому он и описан первым, а бесконечность задаётся с помощью указания трансляционной симметрии на верхнем и нижнем его основаниях. При задании граничных условий в предложении `CONT` они перечисляются в том порядке, в котором определены соответствующие поверхности тела, выбранного в качестве контейнера. Для цилиндра `CylOut`, заданного с помощью тела `RCZ`, эти поверхности перечисляются в следующем порядке: нижнее основание, верхнее основание, боковая поверхность. Таким образом, в наших исходных данных для нижнего и верхнего оснований определено граничное условие трансляции, а для боковой поверхности – чёрная граница. Такое описание граничных условий задаёт бесконечную по высоте трубу, причем все вылетающие через боковую поверхность частицы поглощаются.

Обратим внимание на то, что центр нижнего основания у каждого цилиндра расположен в начале координат – точке (0,0,0). Это однозначно определяет положение нашей трубы в пространстве.

В секции описания геометрических зон мы отдельно описываем собственно трубу и её пустую внутреннюю часть. Важно отметить: при этом в модели не остается неописанных областей, поскольку их наличие внутри контейнера будет вызывать потери траекторий при расчёте их.

Геометрическая зона `Tube`, собственно являющаяся трубой, определяется как пересечение `CylOut` и дополнения к `CylIn`, что можно рассматривать как разность `CylOut` и `CylIn`. Этой зоне мы поставили в соответствие материал номер 1, который должен содержать нуклидный состав материала трубы. Мы также указали, что данная геометрическая зона принадлежит регистрационной зоне 1 и регистрационному объекту 1.

Внутренняя пустая часть трубы `Inside` состоит только из цилиндра `CylIn`. Мы указали, что эта геометрическая зона заполнена материа-

лом 2, который содержит нуклидный состав воздуха, а также принадлежит регистрационной зоне 2 и регистрационному объекту 1.

Заметим, что для данного варианта при регистрации, например потока, ответ, полученный в регистрационном объекте 1, будет с точностью до статистической погрешности совпадать с суммой ответов, полученных для регистрационных зон 1 и 2. Это объясняется тем фактом, что в нашем задании регистрационный объект номер 1 содержит в себе всю моделируемую систему, а две регистрационные зоны также покрывают её полностью. При этом, однако, следует помнить, что при суммировании ответов, полученных для разных регистрационных зон или регистрационных объектов, мы не можем сказать ничего определенного о статистической погрешности результата такого суммирования. Использование возможности регистрации по материалам, зонам и объектам в значительной степени решает эту проблему, позволяя заранее задавать разбиение геометрии системы по регистрационным зонам и регистрационным объектам с учётом геометрии тех областей, в которых требуется получить ответы.

Рассмотрим еще два простых примера задания геометрии.

Первый пример – описание шляпы типа цилиндр. Ниже приведен текст описания этой шляпы:

```
HEAD 5 0 100
CONT В В В
RCZ C1 0,0,-80 160 80
EQU DP = 40 ; диаметр полей шляпы
EQU DO = 20 ; диаметр тульи
EQU DH = 0.5 ; толщина материала шляпы
EQU HH = 24 ; высота шляпы
RCZ CO 0,0,0 DH DP/2
RCZ CM 0,0,0 HH DO/2
RCZ CI 0,0,0 HH-DH DO/2-DH
END
Z0 C1 -CO -CM # Z=7 M=8 ; воздух вне шляпы
Z1 CO -CM # Z=2 M=2 ; поля
Z2 CM -CI # Z=3 M=2 ; тулья с верхним дном
Z4 CI # Z=8 M=8 ; воздух внутри шляпы
END
FINISH
```

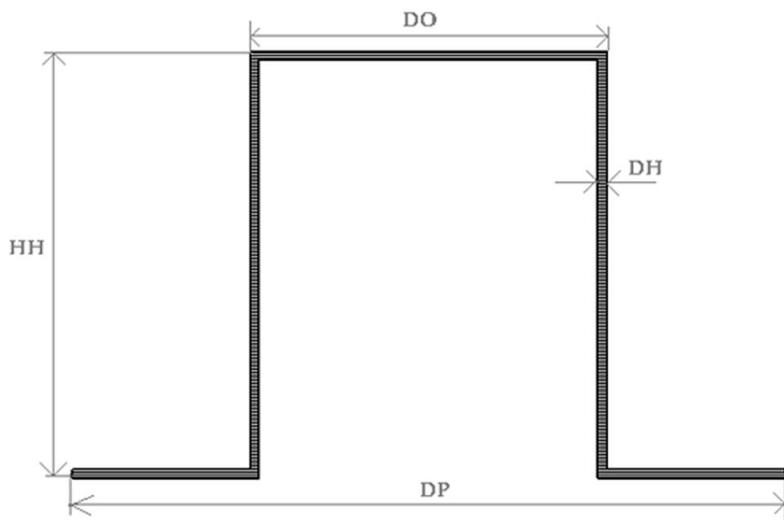


Рис. 27. Вертикальный разрез шляпы типа цилиндр с указанием размеров

Размеры на рис. 27 обозначены теми же именами, что и в файле в конструкциях EQU. Так как при описании цилиндра используется радиус, а не диаметр, в первых двух цилиндрах в качестве радиуса применяется выражение <диаметр>/2. В третьем цилиндре в качестве радиуса – ещё более сложное выражение: из половины диаметра тульи вычитается толщина шляпы. Использование выражений прямо в описании тел позволяет ограничиться только теми четырьмя константами, без которых нельзя точно определить размеры и форму шляпы.

Также следует обратить внимание на то, что в задании все геометрические зоны имеют различные номера регистрационных зон. В практических случаях это редко бывает оправдано, но здесь сделано для иллюстрации того, что и сама шляпа и пространство вне её состоят из 2-х зон: Z1, Z2 и Z0, Z4. На рис. 27 этого не видно, так как изображались не регистрационные зоны, а материалы, которые у обеих зон шляпы есть 2, а у обеих зон пространства – 8. Рис. 28 демонстрирует разбиение на регистрационные зоны.

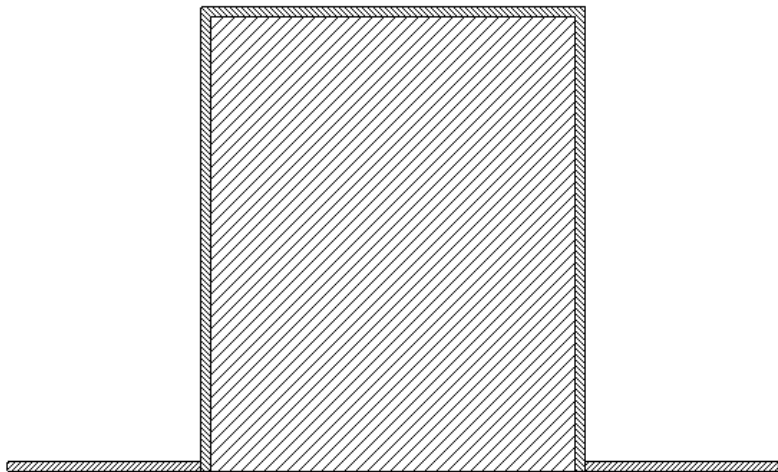


Рис. 28. Разбиение шляпы типа цилиндр на регистрационные зоны

Следующий пример – это описание шляпы типа котелок. У нее те же размеры полей, диаметра тульи и толщины шляпы. Тулья имеет цилиндрическую часть высотой НТ и верхнюю часть в виде полусферы.

```

HEAD 5 0 100
CONT В В В
RCZ C1 0,0, -80 160 80
EQU DP = 40 ; диаметр полей шляпы
EQU DO = 20 ; диаметр тульи
EQU DH = 0.5 ; толщина материала шляпы
EQU НТ = 7 ; высота цилиндрической части
RCZ CO 0,0,0 DH DP/2
RCZ CM 0,0,0 НТ DO/2
RCZ CI 0,0,0 НТ DO/2-DH
SPH SM 0,0,НТ DO/2
SPH SI 0,0,НТ DO/2-DH
PLZ Z0 НТ
END
ZI C1 -CO -CM -SM # Z=7 M=8
Z1 CO -CM # Z=2 M=2
Z2 CM -CI # Z=3 M=2
Z3 SM -SI Z0 # Z=4 M=2

```

```

Z4 CI # Z=8 M=8
Z5 SI Z0 # Z=9 M=8
Z6 SM -Z0 -CM # Z=11 M=8
END
FINISH

```

Помимо двух очевидных новых тел (внешнего шара SM и внутреннего шара SI) появляется ещё тело Z0. Это полупространство выше горизонтальной плоскости на высоте верхней границы цилиндрической части тульи, которое необходимо для того, чтобы «обрезать» нижние части шаров. Если бы высота цилиндрической части превышала диаметры шаров, то можно было бы обойтись без этого тела, вычитая из шаров внешний цилиндр тульи. В нашем случае шары имеют точки под нижней границей шляпы, и нужно дополнительное тело. Шляпа теперь состоит из трёх зон, а окружающее пространство – из четырёх.

Ниже приведены рисунки, аналогичные рисункам для шляпы типа цилиндр.

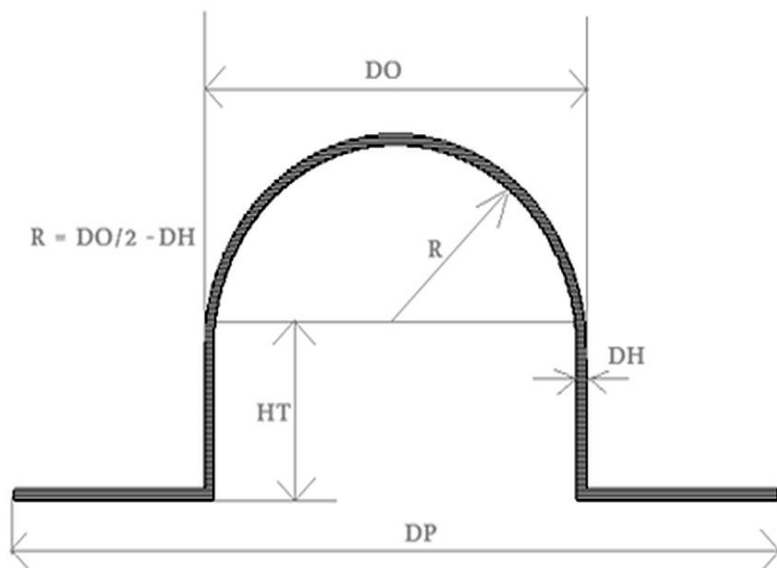


Рис. 29. Вертикальный разрез шляпы типа котелок с указанием размеров



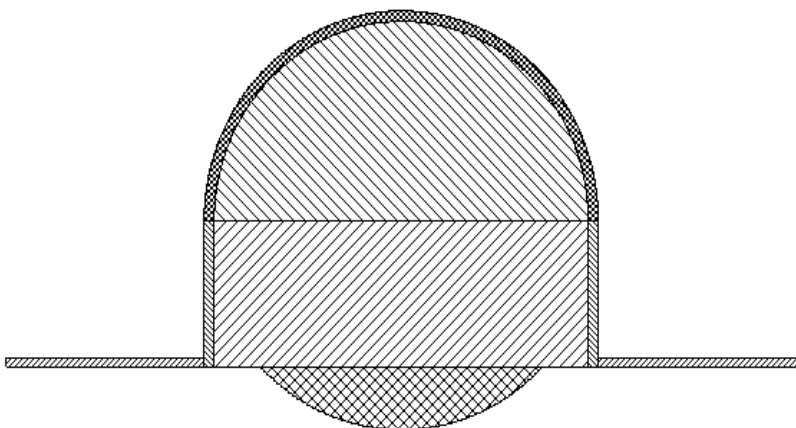


Рис. 30. Разбиение шляпы типа котелок на регистрационные зоны

### ГЛАВА 13. ЗАДАНИЕ МНОГОУРОВНЕВОЙ ГЕОМЕТРИИ

Существует возможность упростить задание повторяющихся геометрических объектов. Такие объекты в NCGSIM можно задавать с помощью сетей и решёток.

Сначала рассмотрим задание повторяющихся геометрических объектов с помощью сетей.

*Сеть* является набором геометрических объектов, называемых ячейками сети и размещённых в одной или нескольких геометрических зонах, описанных в глобальной системе координат. Эти геометрические зоны называются *носителем сети*.

В MSU возможно задание дву- или трёхмерных сетей. В сущности, сети – это дву- или трёхмерные массивы плотно примыкающих друг к другу ячеек одинаковой формы. Они позволяют плотно заполнить часть пространства ячейками, имеющими одинаковую внешнюю форму, при этом их внутреннее содержание может быть различным. Требование одинаковости внешней формы ячеек сети является ключевым. Это непосредственно связано с тем, что при моделировании траекторий частиц внутри сети используется не глобальная система координат, как при описании глобальной геометрии, а локальные системы координат ячеек сети.

Носитель сети отмечается в исходных данных следующим образом. Непосредственно после имени геометрической зоны в скобках

указывается имя сети, которая будет вложена в эту геометрическую зону.

Например, предложение

Z1 (LTH) A1 -A2 #m=5

означает, что геометрическая зона Z1, образованная вычитанием тела A2 из тела A1 и состоящая из материала 5, является носителем сети LTH. При этом всё содержимое сети LTH, выходящее за пределы геометрической зоны Z1 будет автоматически обрезано. Это означает, что частица всегда покидает сеть при пересечении границы носителя сети, то есть происходит переход в глобальную систему координат. Если сеть не заполняет весь носитель сети, то она автоматически дополняется фиктивными ячейками, заполненными гомогенной средой с атрибутами зоны-носителя. Для двумерной сети все верхние и нижние грани ячеек (торцы) должны совпадать с границей или выходить за пределы носителя сети, поскольку по этой оси не происходит дополнение сети фиктивными ячейками. **Запрещается использовать** одну зону в качестве носителя нескольких сетей.

Хотя это и редко встречается на практике, можно использовать несколько геометрических зон в качестве носителя одной сети. При этом исходные данные будут выглядеть, например, таким образом:

Z1 (LTH) A1 -A2 #m=5

Z2 (LTH) A3 -A4 #m=6

Ячейка сети, а точнее, её прототип, называемый **прототипом ячейки сети**, описывается в единой для неё локальной системе координат. Это описание напоминает описание простой геометрической системы. Сначала описываются все тела, используемые для определения геометрических зон, а затем (с помощью комбинаторной геометрии) и собственно геометрические зоны.

**Первое тело прототипа ячейки – его контейнер.** Контейнером может служить либо специальный параллелепипед SBOX, либо параллелепипед RPP с рёбрами, параллельными осям координат, либо шестигранная призма SHEX с осью, параллельной оси OZ. Важным является отсутствие граничных условий, так как при вылете из ячейки частица не покидает систему в целом.

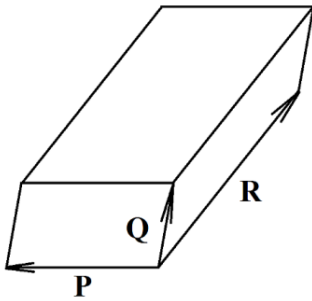


Рис. 31. Тело SBOX

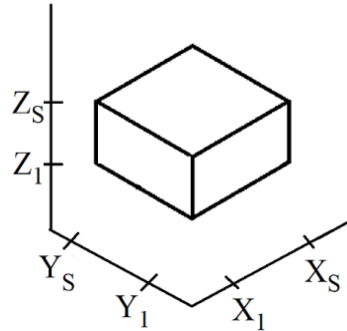


Рис. 32. Тело RPP

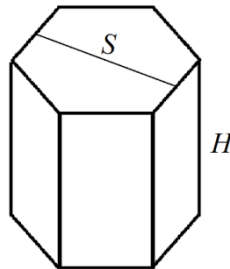


Рис. 33. Тело SHEX

Тело SBOX всегда имеет вершину в начале координат и задаётся тремя векторами  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  – рёбрами, исходящими из начала координат (рис. 31).

Тело RPP (рис. 32) задается с помощью параметров  $X_1$ ,  $X_s$ ,  $Y_1$ ,  $Y_s$ ,  $Z_1$ ,  $Z_s$ , где  $X_1 < X_s$ ,  $Y_1 < Y_s$ , а  $Z_1 < Z_s$ .

Тело SHEX (рис. 33) задается параметрами  $S$ ,  $H$  и  $f$ , где  $S$  - размер под ключ,  $H$  - высота, а  $f$  - угол поворота относительно оси  $Ox$  в градусах. При использовании тела в качестве контейнера прототипа ячейки двумерной сети параметр  $f$  можно не указывать, в этом случае две боковые грани призмы параллельны плоскости  $OYZ$ . При использовании в трёхмерной сети этот параметр нужно задавать обязательно.

Отметим, что, в отличие от тела SBOX, начало координат у тела SHEX находится в центре шестигранника нижнего основания, а не в углу параллелепипеда, как у тела SBOX.

Двумерная сеть образуется из  $N \times M$  ячеек с помощью сдвигов вида:  $C + P(i-1) + Q(j-1)$ , где  $i, j$  - целые числа, причем  $i=1, 2, \dots, N$ , а  $j=1, 2, \dots, M$ .

Вектор  $C$ , называемый корневым вектором, задает положение в пространстве ячейки с индексами (1,1).

Для параллелепипеда RPP векторы  $P$  и  $Q$  являются векторами его первых двух рёбер. Вектор  $P$  направлен вдоль оси  $OX$  и имеет длину  $X_s - X_1$ , а вектор  $Q$  - вдоль  $OY$  и имеет длину  $Y_s - Y_1$ .

Для шестиугольной призмы SHEX векторы  $P$  и  $Q$  лежат в плоскости  $OXY$ , вектор  $P$  имеет длину размера под ключ, а поворот этого вектора соответствует ориентации призмы; вектор  $Q$  получается из вектора  $P$  поворотом на  $120^\circ$  против часовой стрелки (рис. 34).

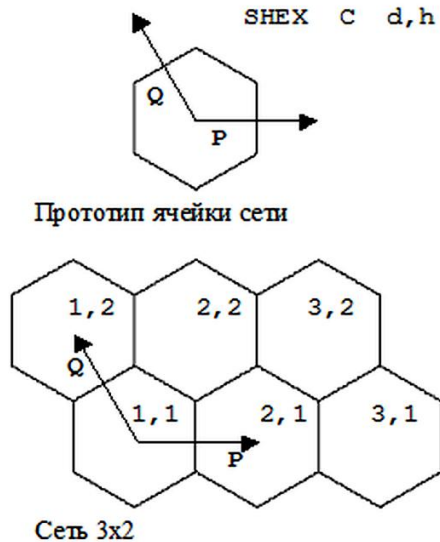


Рис. 34. Двумерная сеть

Отметим, что, хотя строки  
SBOX A1 2,0,0 0,1,0 0,0,3

и

SBOX A2 0,1,0 2,0,0 0,0,3

задают одно и то же тело, при выполнении сдвигов сеть будет генерироваться различным образом.

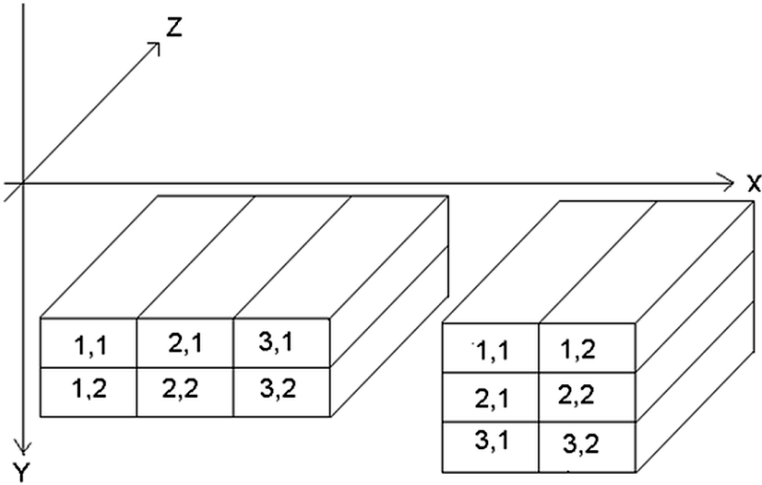


Рис. 35. Пример задания разных сетей с помощью одинакового тела

Трёхмерная сеть в определённом смысле получается из набора двумерных сетей, служащих её слоями. Она образуется из  $N \times M \times L$  ячеек с помощью сдвигов вида  $C + P(i-1) + Q(j-1) + R(k-1)$ , где  $i, j, k$  - целые числа, принимающие значения:  $1 \leq i \leq N$ ,  $1 \leq j \leq M$ ,  $1 \leq k \leq L$ . Векторы  $C$ ,  $P$  и  $Q$  определяются так же, как в двумерном случае. Для параллелепипеда (как RPP, так и SBOX) вектор  $R$  соответствует его третьему ребру. Для шестиугольной призмы вектор  $R$  определён как  $h\ell_z$ , где  $h$  - высота призмы, а  $\ell_z$  - единичный вектор оси OZ.

В исходных данных прототипы ячеек сети и собственно сети описываются следующим образом.

После описания тел и геометрических зон глобальной геометрии в первую очередь описываются все прототипы ячеек сети. Это связано с программной реализацией, которая требует, чтобы при описании сети все используемые в ней прототипы уже были заданы.

Описание прототипа ячейки сети имеет следующую структуру:

```
CELL <имя прототипа ячейки сети>
<раздел тел>
<раздел зон>
```

SBOX В  $p_x, p_y, p_z$   $q_x, q_y, q_z$   $r_x, r_y, r_z$

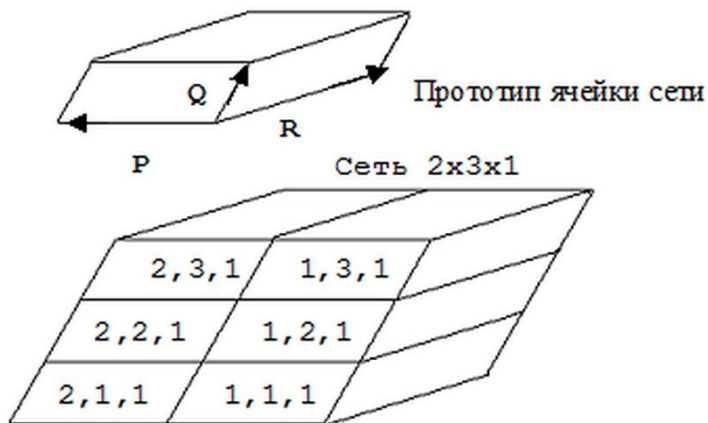


Рис. 36. Трёхмерная сеть

*<имя прототипа ячейки сети>* - произвольный идентификатор, являющийся последовательностью из не более шести латинских букв и цифр, начинающийся с буквы. Это имя служит для ссылки на данный прототип при задании картограммы сети. Оно должно быть уникальным среди имен прототипов ячеек сетей и имён прототипов элементов решёток. Имена констант или имена тел могут совпадать с именами прототипов ячеек сетей и прототипов элементов решёток.

При описании прототипа ячеек сети могут использоваться константы, определенные ранее. Могут быть определены новые константы, но их имена будут доступны лишь в пределах описания данного прототипа. Если определяемая константа имеет имя, уже использованное при описании глобальной геометрии, то новое значение будет сохраняться только для описания данного прототипа ячейки сети.

Раздел тел ничем не отличается от секции тел в описании глобальной геометрии.

Первым телом, которое является контейнером, обязано быть либо тело SBOX, либо тело SHEX, либо тело RPP.

В разделе зон необходимо сослаться только на тела, описанные в разделе тел данного прототипа ячейки сети. Единственное отли-

чие этого раздела от секции зон глобальной геометрии состоит в дополнительных возможностях, появляющихся при определении номеров материалов, регистрационных зон и регистрационных объектов. Эти возможности непосредственно связаны с тем, что номера материалов, регистрационных зон и регистрационных объектов зависят от положения ячейки в сети. В самом деле, предположим, что мы устанавливаем прототип в несколько различных ячеек сети, как это и происходит на практике. Каким образом должны при этом обрабатываться номера (например, регистрационных зон), приписанные геометрическим зонам нашего прототипа? Должны ли они оставаться одинаковыми или же стать различными? Для решения этого вопроса при описании прототипа, помимо обычных, могут использоваться условные номера материалов, регистрационных зон и регистрационных объектов.

Условный номер при использовании первого способа задания номеров материала, регистрационной зоны и регистрационного объекта кодируется отрицательным целым числом, абсолютное значение которого и является значением условного номера. При использовании второго способа к имени параметра добавляется буква I или i, а само число остается положительным.

Например, использование условных номеров для задания материала, регистрационной зоны и регистрационного объекта может выглядеть следующим образом:

```
Inside CylIn #im=1 IZ=1 io=1
```

Перекодировка условных номеров в реальные для разных сетей и разных прототипов ячеек происходит независимо, поэтому *рекомендуется* для каждого прототипа ячейки сети кодировать условные номера, начиная с абсолютного значения, равного 1. При выполнении перекодировки используются задаваемые пользователем таблицы. Они задаются при описании конкретной сети, поскольку один прототип ячейки сети может использоваться для задания нескольких сетей.

Описание сети имеет следующую структуру:

```
NET <имя> <вектор> <столбцов> <строк> <слоев>  
<картограмма имен прототипов ячеек сети>  
<картограммы номеров регистрационных зон>  
<картограммы номеров регистрационных объектов>  
<картограммы номеров материалов>
```

<картограммы символьных имен групп>  
END

<имя> - имя сети, заданное последовательностью из не более шести латинских букв и цифр, начинающейся с буквы. Оно служит для ссылки на данную сеть при описании носителя сети.

<вектор> - корневой вектор сети, определённый тремя координатами  $C_x$ ,  $C_y$ ,  $C_z$ , каждая из которых может быть задана действительным числом, ранее определённым именем константы или выражением, содержащим числа и константы. Координаты должны быть разделены запятыми или пробелами.

<столбцов> - натуральное число, которое равно числу сдвигов по первому вектору прототипа ячейки ( $N$ ).

<строк> - то же по второму вектору ( $M$ ).

<слов> - то же по третьему вектору ( $L$ ). Этот параметр не задётся, если описывается двумерная сеть.

Векторы **C**, **P** и **Q** определяются телом-контейнером ячеек сети. Это тело должно быть одним и тем же для всех прототипов ячеек одной сети.

Задаваемые далее картограммы служат для расстановки прототипов в сети и определения реальных номеров регистрационных зон, регистрационных объектов, материалов. Обязательной картограммой является только картограмма имён прототипов ячеек, остальные могут отсутствовать, если при описании применяемых в сети прототипов не использовались соответствующие условные номера.

*Картограмма имён прототипов ячеек сети* для двумерного случая задается как  $M$  предложений, содержащих по  $N$  фрагментов каждое, где  $N$  – число сдвигов по первому вектору,  $M$  – число сдвигов по второму вектору.

Предложения начинаются с идентификатора, состоящего из латинской буквы T и двухсимвольного номера строки, т.е. T01, T02, T03,..., T09, T10, T11,... и т.д. Каждое предложение отвечает одной строке, т.е. набору индексов  $(1,j), \dots, (N,j)$ , и содержит  $N$  фрагментов. Каждый фрагмент – это имя прототипа, соответствующего данной ячейке.

Гомогенная ячейка с атрибутами зоны-носителя, которой при необходимости дополняется сеть, кодируется в картограмме символом ноль.



Порядок предложений в картограмме имен может быть произвольным, поскольку метка предложения содержит номер строки.

Рекомендуется задавать предложения либо строго по возрастанию значений  $j$ , либо строго по убыванию.

Для трёхмерной сети картограмма задается послойно. Задается  $L$  двумерных картограмм размерности  $N \times M$ , каждая из которых начинается картой вида

TLAYER <номер слоя>

<номер слоя> - натуральные числа от 1 до  $L$  по порядку.

После этого предложения следует  $M$  предложений, начинающихся с идентификаторов T01, T02, T03, ..., T09, T10, T11, ... Идентификаторы повторяются для каждого слоя.

*Картограммы номеров регистрационных зон* задаются, если хотя бы в одном из используемых в сети прототипов имеется хотя бы один условный номер регистрационной зоны. Для каждого такого номера должна вводиться своя картограмма.

Картограмма номеров регистрационных зон состоит из  $M$  предложений, содержащих  $N$  номеров. Предложения начинаются с идентификатора, состоящего из латинской буквы Р и двух натуральных чисел  $k$  и  $j$ . Число  $k$  записывается первым всегда с использованием двух цифр (для чисел меньше 10 – левый 0) и определяет значение условного номера регистрационной зоны. Число  $j$  записывается вторым и также с использованием двух цифр. Оно определяет номер строки ячеек сети и меняется от 1 до  $M$ . Эти идентификаторы выглядят следующим образом: P0101, P0102 и т.д. После идентификатора в предложении записываются натуральные числа, имеющие значение номера регистрационной зоны, который будет присвоен геометрической зоне с условным номером регистрационной зоны  $k$  в прототипе ячейки, который будет размещён в сети путем сдвига на  $j$  позиций по второму вектору и на число позиций, соответствующих положению в предложении, по первому вектору.

Значения условных номеров локальны в каждом прототипе ячейки, и в разных ячейках в зависимости от их положения в сети одному и тому же условному номеру регистрационной зоны могут соответствовать разные реальные номера регистрационных зон.

Все номера регистрационных зон, заданные в прототипах ячеек без использования условных указателей, сохраняются во всех ячей-

ках независимо от того, что указано в картограммах. Их перекодировка никогда не производится.

Общее количество картограмм равно максимальному значению  $K$  условного номера в использованных прототипах. Это, в частности, означает, что необходимо задавать картограммы и для пропущенных условных номеров. Иными словами, если в используемых в сети прототипах определены условные номера 1 и 3, а условный номер 2 не используется, то для него всё равно придется задавать картограмму.

Использование имён констант в картограмме номеров регистрационных зон *не разрешается*.

Начинающему пользователю рекомендуется не использовать ноль в картограммах регистрационных номеров. Общее же правило по использованию ноля в картограммах выглядит следующим образом. Если в картограмме имён прототипов ячеек сети в какой-либо позиции задан ноль, то в картограмме регистрационных номеров в этой позиции также может быть указан ноль. В противном случае ноль можно указывать только для тех прототипов, в которых не используется ни одного условного номера. Если же в прототипе присутствует хотя бы один условный номер, то в картограмме регистрационных номеров в соответствующей позиции должно быть указано отличное от нуля значение, если  $k$  меньше или равно максимальному абсолютному значению, использованному в этом прототипе при задании условных номеров, даже если условный номер с абсолютным значением равным собственно  $k$  в прототипе отсутствует. Если вся  $k$ -я картограмма состоит из одного и того же значения, то вместо  $M$  предложений можно написать одно предложение вида  $P\langle k \rangle ALL$ . Например, предложение

`P02ALL 23`

означает, что условный номер регистрационной зоны с абсолютным значением 2 во всех ячейках получит реальное значение 23.

Если сеть трёхмерная, то картограммы регистрационных номеров задаются послойно. Задаётся  $L \times K$  картограмм размерности  $M \times N$ . Перед каждой картограммой вводится предложение вида

`P\langle k \rangle LAY <номер слоя>`

$\langle k \rangle$  - натуральное число, имеющее два символа (для чисел меньше 10 левый 0 записывается), значение условного регистрационного указателя;

<номер слоя> - натуральное число, имеющее значение номера слоя. Слои перечисляются по порядку от 1 до L.

Затем вводятся M предложений, построенных аналогично двумерному случаю.

*Картограммы номеров регистрационных объектов* устроены полностью аналогично картограммам номеров регистрационных зон, но в метках предложений латинская буква P должна быть заменена латинской буквой O.

*Картограммы номеров материалов* устроены полностью аналогично картограммам номеров регистрационных зон, но в метках предложений латинская буква P должна быть заменена латинской буквой M.

*Картограммы символьных имен групп* вводятся после картограмм материальных номеров. Эти картограммы аналогичны картограммам материальных номеров, но в метках предложений латинская буква M должна быть заменена латинской буквой G, а вместо номера материала вводится символьное имя группы. Задание этих картограмм требуется, если в разделе PIN для описания материала используется символьное имя группы. В тех местах картограммы, где символьное имя группы не определено, ставится цифра ноль.

Рассмотрим пример исходных данных, использующий сети:

```
HEAD 1 0 0
CONT M M M M M M
RPP CA -1 7 -1 5 0 8
RPP CN -0.5 6.5 -0.5 4.5 0 8
END
BHE CA -CN # M=7 Z=7
ENV (NT) CN # M=1 Z=1
END
*
CELL P
RPP K 0,2 0,2 0,8
RCZ B 1,1,0 8 0.7
END
PM K -B # M=1 Z=2
PS B # M=9 Z=3
END
*
```

```

CELL Q
RPP K 0,2 0,2 0,8
RCZ B 1,1,0 8 0.5
END
PM K -B # M=1 Z=6
PS B # M=9 iZ=1
END
*
NET NT 0,0,0 3,2
T01 P P Q
T02 Q P P
*
P0101 0 0 4
P0102 5 0 0
END
FINISH

```

В этом примере в контейнер CA помещён второй параллелепипед CN, который заполняется двумерной сетью NT. Сеть состоит из ячеек с формой параллелепипеда с квадратным основанием  $2 \times 2$  см.

Размерность сети указана как  $3 \times 2$ , и такой набор из шести ячеек не заполняет носитель полностью. Поэтому с каждой стороны сеть автоматически дополняется рядом из пустых, т.е. гомогенных ячеек с атрибутами геометрической зоны-носителя сети. Каждая из этих новых ячеек обрезается границами носителя. Это можно увидеть, построив с помощью MCU Office сечение для отображения регистрационных зон со следующими параметрами диалога Data:

```

Upper Left X=-2, Y=6, Z=1
Bottom Right X=8, Y=-2, Z=1
Screen Vertical Direction Vector X=0, Y=1, Z=0

```

Сеть состоит из двух типов ячеек, их прототипы названы P и Q. Прототип Q содержит условный номер регистрационной зоны с номером 1. Поэтому, кроме основной картограммы, имеется одна картограмма номеров регистрационных зон. В позициях, соответствующих позициям прототипа P, стоят нули, так как этот прототип ячейки не содержит условных номеров регистрационных зон. В этих позициях могли бы быть и другие цифры, что никак не сказалось бы на результатах моделирования.

Первый и, возможно, главный аргумент за использование сетей следующий. После описания повторяющегося геометрического объекта, который затем устанавливается в сеть необходимое коли-

чество раз, с точки зрения программы он остается описанным только один раз. Иными словами, его размножения не происходит. Это позволяет существенно снизить нагрузку на оперативную память компьютера, затрачиваемую на проведение расчёта.

Вторым аргументом в пользу использования сетей служит то, что при расчёте траекторий частиц внутри сети переход от глобальных координат к локальным и обратно производится только дважды: один раз, когда частица входит в сеть, и второй раз, когда она её покидает. Во время нахождения частицы внутри сети используются локальные системы координат ячеек, составляющих эту сеть. Тот факт, что ячейки составляют строго упорядоченный массив, позволяет избежать процесса поиска геометрической зоны, в которую попадает частица при выходе из ячейки, поскольку она однозначно определяется по индексу ячейки в массиве. В некоторых случаях этот факт позволяет сократить общее время счёта.

В отличие от сетей, решётки присутствуют только на этапе ввода исходных данных, а при выполнении непосредственно расчёта их уже не существует. Вся локальная геометрия, описанная с помощью решёток, на этапе ввода исходных данных интегрируется в глобальную геометрию.

Решётка состоит из набора элементов. *Элемент решётки* – это некоторая область пространства, разбитая на геометрические зоны с присвоенными им номерами материалов, регистрационных зон и регистрационных объектов. Элементы решётки размещаются в геометрических зонах глобальной геометрии. Элемент решётки, а точнее, его прототип, описывается в локальной системе координат.

Прототип элемента решётки, аналогично прототипу ячейки сети, имеет контейнер, которым, в отличие от прототипа ячейки сети, может быть тело любого типа. Контейнер прототипа решётки может быть расположен произвольно относительно начала локальной системы координат прототипа. Следует иметь в виду, что по умолчанию тела, используемые в прототипе, не пересекаются автоматически с его телом-контейнером. Пользователю необходимо следить за этим самостоятельно.

Прототип элемента решётки может содержать в себе геометрические зоны-носители сетей и описания этих сетей. Геометрические зоны прототипа элемента решётки могут быть носителями только тех сетей, которые принадлежат данному прототипу элемента ре-

шётки. Корневые векторы этих сетей даются в локальной системе координат прототипа. Иными словами, прототип элемента решётки может содержать только те сети, которые описаны внутри него. Прототипы ячеек этих сетей описываются отдельно от прототипа элемента решётки и от самой решётки. Один и тот же прототип ячейки может использоваться в сетях нескольких прототипов элементов решёток и одновременно в сетях глобальной геометрии. Несмотря на то, что сеть описывается внутри прототипа элемента решётки, имя такой сети должно быть уникальным во всем описании геометрии.

Прототип элемента решётки размещается в зонах основной геометрии с помощью движений. Под движением понимается совокупность вращения и сдвига или зеркальной симметрии и сдвига. Как правило, используется только сдвиг. Движение можно трактовать как перенос прототипа на конкретное место. Движения вырабатываются специальными подпрограммами генерации решёток. Какую именно подпрограмму использовать, определяет пользователь в исходных данных? При размещении элемента в геометрических зонах глобальной геометрии каждая такая геометрическая зона модифицируется путем пересечения с дополнением

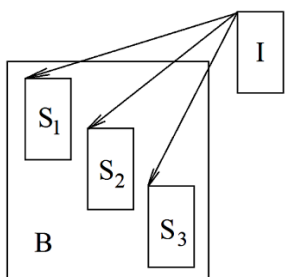


Рис. 37. Решётка из трёх элементов

контейнера размещаемого элемента. Иначе говоря, из геометрической зоны, в которой размещается элемент решётки, вычитается тело, являющееся контейнером прототипа этого элемента и установленное с помощью движения в нужную точку пространства.

На рис. 37 изображена решётка из трёх элементов.  $I$  - прототип элемента решётки,  $S_1, S_2, S_3$  - векторы сдвигов для элементов решётки. Геометрическая зона  $B$  преобразуется в  $B - S_1(I) - S_2(I) - S_3(I)$ , где  $S_i(I)$  -  $i$ -й элемент решётки.

В отличие от сетей, в решётках отсутствует требование одинаковости контейнеров прототипов элементов и, как следствие, регулярности размещения элементов в пространстве.

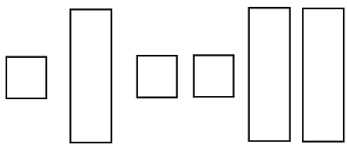


Рис. 38. Пример решётки

На рис. 38 изображена решётка, образованная с помощью двух прототипов с различной формой контейнеров.

Одна решётка может быть наложена на несколько зон, а на одну зону – несколько решёток. Элементы всех наложенных решёток не должны иметь общих точек, за исключением граничных, которые могут быть общими.

Допускается наложение решётки на зону-носитель сети. Если точка одновременно лежит в ячейке сети и элементе решётки, приоритет отдается решётке. Таким образом, решётка как бы выдавливает из глобальной геометрии всё, на что она накладывается.

При описании прототипов элементов решёток, кроме обычных номеров материалов, регистрационных зон и регистрационных объектов, можно использовать условные номера. Условные номера можно использовать и в картограммах сетей, описанных внутри прототипа элемента решётки. Геометрической зоне с условным номером реальный номер будет присвоен автоматически только после генерации решётки. Рассмотрим используемый при этом алгоритм на примере определения реального номера регистрационной зоны.

Обозначим через  $N$  максимальный реальный номер регистрационной зоны, использованный к моменту генерации решётки. Пусть в первом элементе решётки имеются условные номера регистрационных зон со значениями до  $k$  включительно. Тогда зоны с условным номером один получают реальный номер равный  $N+1$ , с условным номером два -  $N+2$  и так далее до  $N+k$ . После этого  $N$  увеличивается на  $k$ , и начинается размещение второго элемента решётки. Если набор используемых условных номеров имеет пропуски (например, 1, 2, 5, 6), то и набор реальных номеров регистрационных зон также будет иметь пропуски.

Определение реальных номеров для материалов и регистрационных объектов осуществляется аналогично. Результаты такого автоматического преобразования обычно контролируются с помощью MCU Office.

Описание решётки имеет следующий вид:

```
LATT <генератор движений> <имя зоны> ... <имя зоны>
LISTEL <имя ПЭР> (<список ограничения>) ...
```

<имя ПЭР> (<список ограничения>)

PARM <строка параметров>

<генератор движений> - название подпрограммы генерации, которая должна быть использована для создания решётки. Имеется три таких подпрограммы: GLTL, G2MP и G2AR.

<имя зоны> - имя геометрической зоны, на которую накладываются элементы решётки. Должны быть приведены имена всех геометрических зон, на которые накладывается данная решётка. Имена могут быть разделены пробелами или запятыми. Все эти геометрические зоны должны быть описаны ранее в разделе глобальной геометрии.

<имя ПЭР> - имя прототипа элемента решётки, который будет использован при создании решётки. Все используемые прототипы должны быть уже описаны. В процессе ввода каждому имени прототипа присваивается номер по порядку его размещения в строках: 1, 2 и т.д. Этот номер используется при задании параметров подпрограммы генератора. Прототип из списка может быть не использован в расчёте, однако нумерация всё равно ведётся по списку.

<список ограничения> - имя или несколько имён тел, описанных в глобальной геометрии. Имена могут быть разделены пробелами или запятыми. Если имя какого-либо тела приведено со знаком минус, то оно означает дополнение этого тела. В списке ограничения могут использоваться только дополнительные имена тел, приведённые в описании тел после основного имени в скобках. Наличие списка ограничения позволяет описывать элементы решётки, обрезанные за счёт пересечения с одной или несколькими граничными поверхностями тел, образующих зоны глобальной геометрии. Данный список может отсутствовать. При отсутствии списка скобки не используются.

<строка параметров> - определяет параметры для подпрограмм генерации. Набор параметров определяется в зависимости от выбранной подпрограммы и будет описан далее при рассмотрении каждой подпрограммы.

Как уже говорилось выше, имеется три подпрограммы генерации: GLTL, G2MP и G2AR.

GLTL задает радиус-векторы положения экземпляров элементов решётки путем прямого перечисления их координат. В определённом смысле это непосредственная реализация идеи расстановки



элементов в решётке. Очередная копия прототипа элемента решётки устанавливается непосредственно в указанную точку пространства в соответствии с параметрами тела, используемого в этом прототипе в качестве контейнера.

Если решётка задаётся с помощью генератора GLTL, то *<строка параметров>* состоит из последовательности фрагментов следующего вида:

*/<целое число> <движения>*

*<целое число>* - это порядковый номер прототипа элемента в строке LISTEL. По умолчанию он равен 1.

Здесь мы будем рассматривать только простые сдвиги. Поэтому в качестве параметра *<движения>* рассматривается только его простой вариант, параметр *<радиус-вектор положения>* - параметр, определённый тремя координатами радиуса-вектора, каждая из которых может быть задана действительным числом, ранее определённым именем константы или выражением, содержащим числа и константы. Параметры должны быть разделены запятыми или пробелами. Указание номера прототипа элемента распространяется только на один следующий за ним фрагмент.

В качестве примера рассмотрим задание решётки из шести элементов:

```
LATT GLTL ZN1 , FUT
LISTEL TPIC TP2D
PARM X1,Y1,0.5 X2,Y2,0.5 X3,Y3,0.5
      /2 X4,Y4,0.5 X5,Y5,0.5 X6,Y6,0.5
```

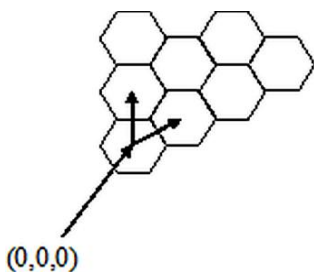


Рис. 39. Задание решётки

В описании решётки – два прототипа элементов, поскольку наибольший номер, использованный в строке параметров, равен 2. В качестве первого прототипа указывается TPIC, а в качестве второго – TP2D. Второй прототип применяется лишь один раз. Его копия устанавливается в точку с координатами X4,Y4,0.5. Поскольку для остальных точек номер

прототипа не указан, то по умолчанию используется прототип номер 1, копии которого и размещаются в пять других точек.

Если бы в строке параметров вместо цифры два использовалась цифра три, то при описании решётки в строке LISTEL пришлось бы указывать три имени, несмотря на то, что реально указывались бы только два прототипа: первый и третий.

Генератор G2MP используется для задания решётки в виде регулярного двумерного массива элементов. Для этого G2MP применяет картограмму, аналогичную той, при помощи которой производится описание сетей.

Решётка получается из регулярного двумерного массива, образуемого как набор сдвигов вида

$$\{A+(i-1)B+(j-1)C: 1 \leq i \leq I, 1 \leq j \leq J\}.$$

В случае применения этого генератора *<строка параметров>* имеет следующую структуру:

*<столбцов I>*, *<строк J>*, *<вектор A>*, *<вектор B>*, *<вектор C>*,  
*<картограмма имен прототипов элементов>*

*<столбцов I>*, *<строк J>* – параметры, определяющие размерность двумерной решётки, а именно: число столбцов и строк в картограмме.

*<вектор A>* – задается тремя координатами корневого вектора **A**, определяющего положение элемента из первого столбца и первой строки в пространстве.

*<вектор B>* – задается тремя координатами вектора сдвига **B**, определяющего сдвиг по столбцам.

*<вектор C>* – задается тремя координатами вектора сдвига **C**, определяющего сдвиг по строкам.

Каждая из координат векторов **A**, **B**, **C** может быть задана действительным числом, ранее определённым именем константы или выражением, содержащим числа и константы. Параметры должны быть разделены запятыми или пробелами.

*<картограмма имен прототипов элементов>* – картограмма имен прототипов элементов состоит из *J* предложений, содержащих по *I* фрагментов каждый.

Метками *J* предложений картограммы должны быть идентификаторы, состоящие из буквы L и двузначного номера строки, т.е. L01, L02, L03, ..., L09, L10, L11, ... и т.д.

Каждое предложение отвечает одной строке и содержит *I* фрагментов. Один фрагмент имеет вид

*<имя прототипа элемента решётки>*

<имя прототипа элемента решётки> – имя прототипа элемента решётки, определенное в списке прототипов LISTEL. Если в какой-либо позиции решётки элемент должен отсутствовать, вместо имени прототипа в соответствующую позицию картограммы записывается ноль.

Ниже приведен пример задания гексагональной решётки кассет реактора ВВЭР–1000 с помощью генератора сдвигов G2MP:

```

EQU TVSP = 23.6 ;TVS lattice pitch
EQU NETLX=-3.5*TVSP
EQU NETLY=-7*TVSP*SQRT(3)/2
LATT G2MP ZZZ
LISTEL K1 K12 K13 K14 K15 K17 K21 K26 K28
      K29 K20 K2A K2B K2C K3 K3B K1Z K2Z
PARAM 15,15;columns(I), rows (J)
      NETLX, NETLY,0. ;vector A
      TVSP,0.,0. ;vector B
      -TVSP*SIN(30),TVSP*COS(30),0. ;vector C
L15 0 0 0 0 0 0 0 0 K3 K3 K3 K3 K3 0
L14 0 0 0 0 0 0 0 K3 K3B K28 K2C K26 K2C K29 K3B K3
L13 0 0 0 0 0 K3 K29 K1 K14 K12 K13 K15 K1 K28 K3
L12 0 0 0 0 K3 K2C K15 K2B K2A K20 K2A K2B K14 K2C K3
L11 0 0 0 K3 K26 K13 K2A K17 K1 K1 K17 K2A K12 K26 K3
L10 0 0 K3 K2C K12 K20 K1 K2A K28 K2A K1 K20 K13 K2C K3
L09 0 K3 K28 K14 K2A K1 K21 K1 K1 K21 K1 K2A K15 K29 K3
L08 0 K3B K1 K2B K17 K2A K1 K29 K1 K2A K17 K2B K1 K3B 0
L07 K3 K29 K15 K2A K1 K28 K1 K1 K28 K1 K2A K14 K28 K3 0
L06 K3 K2C K13 K20 K1 K2A K21 K2A K1 K20 K12 K2C K3 0 0
L05 K3 K26 K12 K2A K17 K1 K1 K17 K2A K13 K26 K3 0 0 0
L04 K3 K2C K14 K2B K2A K20 K2A K2B K15 K2C K3 0 0 0 0
L03 K3 K28 K1 K15 K13 K12 K14 K1 K29 K3 0 0 0 0 0
L02 K3 K3B K29 K2C K26 K2C K28 K3B K3 0 0 0 0 0 0
L01 0 K3 K3 K3 K3 K3 K3 0 0 0 0 0 0 0 0

```

В этом примере гексагональные кассеты расставляются в гексагональную же решётку размерностью 15×15. Первый элемент решётки, соответствующий первому столбцу и первой строке картограммы, размещается в точке NETLX, NETLY,0. В данном примере решётка начинается с нуля, соответствующего первому столбцу строки L01 и означающего, что никакой прототип в эту позицию не устанавливается. Сдвиги от столбца к столбцу определяются направлением и длиной вектора (TVSP,0,0). Сдвиги ме-

жду строками определяются направлением и длиной вектора  $(-TVSP*\sin(30), TVSP*\cos(30), 0)$ .

Этот генератор исторически является развитием идеи, реализованной в генераторе G2AR. В отличие от G2MP генератор G2AR использует несколько более сложный и значительно менее наглядный формат задания упорядоченной решётки. Вследствие этого G2AR в рамках данного курса рассматриваться не будет и рекомендуется для самостоятельного изучения.

Пример задания данных для геометрического модуля, содержащий сеть, решётку и сеть в элементе решётки:

```
HEAD 7 0 100
CONT W
EQU d=6.
EQU r1 = 12
SPH CNTR 0,0,0 r1
RPP HOCL -d,d -d,d 2.0,6.0
RPP HOCN -d,d -d,d -2.0,2.0
END
ZB CNTR -HOCL -HOCN /1:3
ZL HOCL /2:2 ; Носитель решётки
ZN (NT) HOCN /4:2 ; Носитель сети
END
* Первый прототип ячейки для сети NT
CELL CN1
EQU DR=4
SBOX B1 3,0,0 0,DR,0 0,0,DR
RCZ CI 1.5 DR/2. 0. DR DR*0.3
END
ZC1 B1 -CI /3:12
ZC2 CI /4:2
END
* Второй прототип ячейки для сети NT,
* условные регистрационные зоны в старом формате
CELL CN2
SBOX B1 3,0,0 0,4,0 0,0,4
RPP P2 0.6,2.4 0.5,3.5 0,4
END
ZC1 B1 -P2 /-1:11
ZP2 P2 /-2:10
END
* Сеть NT, используется в глобальной геометрии
```

```

NET NT -6, -6, -2 4,3
* Картограмма ячеек - основная картограмма
T01 CN1 CN1 CN1 CN1
T02 CN2 CN1 CN2 CN1
T03 CN1 CN2 CN1 CN1
* Картограмма номеров регистрационных зон
P0101 0 0 0 0
P0102 5 0 6 0
P0103 0 7 0 0
P0201 0 0 0 0
P0202 9 0 9 0
P0203 0 10 0 0
END
* Прототип ячейки для сети внутри элемента решётки,
* условные регистрационные зоны в новом формате
CELL CLN
EQU DP=0.8
EQU DQ=0.7
EQU DR=3.0
SBOX BXL DP,0,0 0,DQ,0 0,0,DR
END
CD2 BXL # iZ=1 M=14
END
* Первый прототип элемента решётки
LCELL BUU
EQU R1=1
EQU R2=0.2
EQU X=0.1
EQU Y=0.
EQU Z=0
RCZ BL 0,0,0 3 R1
RCZ C1 -X,Y,Z 3 R2
RCZ C2 X,Y,Z 3 R2
END
LZZ BL -C1 -C2 # iZ=1 M=1
C12 C1 U C2 # iZ=2 M=5
END
ENDL
* Второй прототип элемента решётки, содержащий сеть
* с условными указателями в картограмме рег.номеров
LCELL BVV
RPP BL -0.8,0.8 -0.7,0.7 0,3

```

```

END
CC3 (VNT) BL /2:2
END
* Внутренняя сеть в элементе решётки
NET VNT -0.8, -0.7, 0 2, 2
T01 CLN CLN
T02 CLN CLN
P0101 -1 3
P0102 4 -1
END
ENDL
* Решётка
EQU POXY = 2
LATT GLTL ZL
LISTEL BUU BVV
PARM -POXY, -POXY, 2.5
/2 -POXY, POXY, 2.5 POXY, POXY, 2.5
/2 POXY, 1.5-POXY, 2.5
*
FINISH

```

На рис. 40 приведены последовательно:

- a) разрез сети в основной геометрии, изображающий материалы;
- b) разрез сети в основной геометрии, изображающий регистрационные зоны;
- c) разрез решётки в основной геометрии, изображающий материалы;
- d) разрез решётки в основной геометрии, изображающий регистрационные зоны;

Стоит обратить внимание на две особенности задания геометрии, использованные в данном примере.

В описании сети NT имеются две картограммы номеров регистрационных зон: для первого и второго условного номера. В позициях, соответствующих прототипу ячейки CN1, в обеих картограммах стоит ноль. Это связано с тем, что в прототипе ячейки CN1 условные номера регистрационных зон не используются, и, следовательно, из картограмм номеров регистрационных зон никакие значения не выбираются. Число в этой позиции не играет роли.

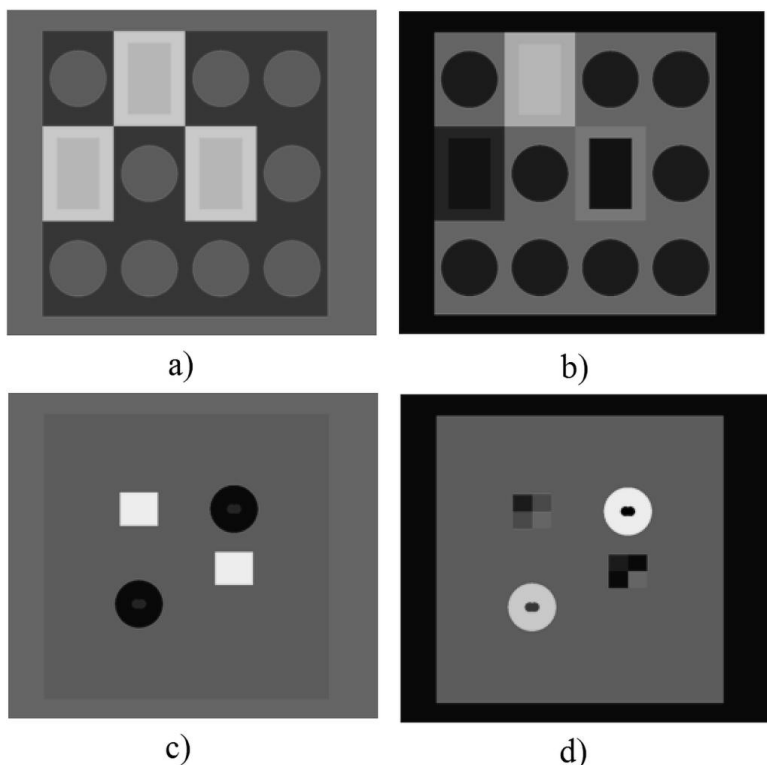


Рис. 40. Пример с сетью, решёткой и сетью в элементе решётки

## ГЛАВА 14. ИСХОДНЫЕ ДАННЫЕ ДЛЯ МОДУЛЯ ИСТОЧНИКА

Модуль источника предоставляет возможность выполнять расчёты с одним из двух типов источников: простым или сложным. При решении задач на собственное значение часто достаточно возможностей простого источника. Это связано с тем, что в этом случае источник используется только для генерации нейтронов нулевого поколения. При моделировании последующих поколений распределение сходится к старшей собственной функции со степенной скоростью, то есть предел не зависит от начального распределения. При усреднении оценок по поколениям зависимость от начального распределения имеет порядок  $1/N$  при достаточно больших  $N$ , где  $N$  - число промоделированных поколений. Эта величина меньше ста-

тистической ошибки, которая уменьшается как  $1/N^{0.5}$ . Более того, в программе имеется возможность не учитывать вклад нескольких начальных поколений.

При описании источника можно использовать константы и выражения, так же как при описании геометрии. Константы, определённые при описании геометрии, доступны при описании источника.

#### 14.1. ТОЧЕЧНЫЙ ИСТОЧНИК

Наиболее простой источник – это точечный источник с изотропным распределением по углу. Его спектр по энергиям может быть либо  $\delta$ -функцией, либо ступенчатой функцией.

Признаком использования простого источника является предложение SPNT, которое должно быть первым предложением фрагмента исходных данных для модуля источника. Все остальные предложения могут отсутствовать. Описание фрагмента обязательно заканчивается предложением FINISH:

```
SPNT X, Y, Z
ESET <массив из 27 чисел>
SPEC <массив из 26 чисел>
ENSO <число>
FINISH
```

$X, Y, Z$  - координаты точки источника, задаваемые в сантиметрах. За начало координат принимается начало отсчёта, определённое в данных к геометрическому модулю.

<массив из 27 чисел> - границы энергетических интервалов, задаваемых в убывающем порядке. Энергия задаётся в электрон-вольтах. В результате работы источника будут вырабатываться энергии, равные серединам интервалов. Границы интервалов, используемые по умолчанию, приведены в табл. 1.

<массив из 26 чисел> - вероятности энергетических интервалов в виде ненормированных весов. Используемые по умолчанию вероятности соответствуют спектру деления  $^{235}\text{U}$  тепловым нейтроном (см. табл. 1).

<число> - энергия в электрон-вольтах.

При наличии предложения ENSO спектр состоит из одной  $\delta$ -функции, то есть все частицы вырабатываются с одинаковой энер-



гией. Предложение ENSO нельзя использовать в сочетании с предложениями ESET и SPEC.

Отметим, что единственное формальное ограничение на значение энергии в предложениях SPEC и ENSO состоит в запрете на использование отрицательных величин, однако оно должно согласовываться с возможностями физического модуля. Так, например, если применяется такая конфигурация физического модуля, при которой энергетический диапазон моделирования определён от 0 до 10,5 МэВ, то не следует задавать в источнике ненулевую вероятность рождения нейтронов с энергиями выше 10,5 МэВ.

Как уже упоминалось, при решении задач на собственное значение источник используется только для формирования первого поколения нейтронов, поэтому нет необходимости применять сложные пространственные и энергетические распределения. Тем не менее для быстрого выхода на распределение частиц, близкое к собственной функции, целесообразно сразу разбросать нейтроны по некоторой пространственной области. Для этого используются возможности, предоставляемые сложным источником.

Таблица 1

Значения, используемые по умолчанию,  
при задании точечного источника

Границы энергетических интервалов, эВ	Вероятности энергетических интервалов
1,0E+07	0,016
6,5E+06	0,088
4,0E+06	0,184
2,5E+06	0,270
1,4E+06	0,202
8,0E+05	0,141
4,0E+05	0,061
2,0E+05	0,024
1,0E+05	0,010
4,65E+04	0,003

2,15E+04	0,001
1,0E+04	0
4,65E+03	0
2,15E+03	0
1,0E+03	0
465	0
215	0
100	0
46,5	0
21,5	0
10	0
4,65	0
2,15	0
1	0
0,465	0
0,215	0
0	

В числе возможностей сложного источника есть работа со смещёнными распределениями. В большинстве случаев нет необходимости в применении этой опции, и для формирования нулевого поколения она никогда не используется, поэтому описание работы со смещёнными распределениями вынесено в отдельный раздел.

## **14.2. СЛОЖНЫЙ ИСТОЧНИК БЕЗ СМЕЩЕНИЙ РАСПРЕДЕЛЕНИЙ**

В определённом смысле сложный источник представляет собой набор примитивных источников, каждый из которых определяется собственной формой, вероятностью срабатывания и весом генерируемых частиц. Порядок секций фрагмента исходных данных для сложного источника строго фиксирован и задается следующим образом:

```
NPS <n>
PROB <массив>
<секции описания спектра>
<секции примитивных источников>
FINISH
```

Предложение NPS является признаком использования сложного источника и определяет количество примитивных источников  $\langle n \rangle$ , которые будут использованы для формирования сложного источника.

Предложение PROB задает  $n$  интенсивностей примитивных источников:  $f_1, f_2, \dots, f_n$ . При генерации очередной частицы примитивный источник с номером  $j$  будет использован с вероятностью  $f_j / (f_1 + f_2 + \dots + f_n)$ .

Таким образом, параметры предложения PROB можно интерпретировать как ненормированные вероятности, их нормировка не влияет на работу программы, то есть предложение

```
PROB 4 8 6 2
```

совершенно равносильно предложению с нормированными вероятностями

```
PROB 0.2 0.4 0.3 0.1
```

Источник должен определять все фазовые координаты генерируемых частиц, в том числе направление полёта. В программе MCU (так же, как и в программе MCNP) отсутствует возможность задания произвольного закона распределения направлений. Считается, что при генерации пространственной точки рождения частицы одновременно оказывается определённым «основное направление», характерное для данной точки. Единичный вектор этого направления будет в дальнейшем называться *реперным вектором*. Распределение направлений полета задаётся в виде распределения величины косинуса угла, между вектором скорости частицы и реперным вектором, этот косинус будет обозначаться буквой  $\mu$ . Азимутальное распределение всегда равномерно по окружности.

Описание спектра, задаваемое в *<секции описания спектра>*, определяет правило генерации величин энергии, обозначаемой  $E$ , и величин упомянутого косинуса  $\mu$ .

Секция описания спектра обязательно начинается с предложения

```
ANGLEN <имя спектра> <зависимость>
```

<имя спектра> – это некоторое произвольное символьное имя, которое в дальнейшем будет использоваться для ссылок на этот спектр. Оно не должно содержать более шести латинских букв и цифр.

Параметр <зависимость> может принимать следующие значения:

*NO* - независимость  $E$  и  $\mu$ ;

*MTOE* - зависимость  $E$  от  $\mu$ ;

*ETOM* - зависимость  $\mu$  от  $E$ ;

*FUNC* - функциональная зависимость.

Если параметр <зависимость> отсутствует, то это означает независимость  $E$  и  $\mu$ , т.е. эквивалентно значению *NO*. В большинстве случаев бывает достаточно независимых распределений  $E$  и  $\mu$ , поэтому разъяснение смысла зависимостей *MTOE*, *ETOM*, *FUNC* в настоящем пособии не приводится.

После указания вида зависимости энергии и направления полёта генерируемых источником частиц необходимо определить эту зависимость более конкретно. Этой цели служат предложения MDIS и EDIS, которые задаются непосредственно после предложения ANGLN:

MDIS <тип распределения  $\mu$ > <число>

EDIS <тип распределения  $E$ > <число>

В модуле источника, кроме нескольких стандартных распределений, применяются распределения, задаваемые таблицами. В последнем случае можно использовать дискретные и линейные распределения.

Для дискретного распределения задают конечный набор возможных значений случайной величины и вероятности для каждого из них.

При использовании линейного распределения необходимо задать конечный набор примыкающих друг к другу отрезков

$[w_0, w_1], [w_1, w_2], \dots, [w_{n-1}, w_n]$ ,

а также вероятности для каждого из этих отрезков. Внутри каждого отрезка случайная величина  $w$  распределена равномерно.

MDIS определяет распределение косинуса угла между вектором скорости частицы  $\mu$  и реперным вектором, а EDIS – распределение энергии генерируемых частиц.

<тип распределения  $\mu$ > принимает одно из следующих значений:

D - дискретное распределение;

L - линейное распределение;

S - изотропное распределение (допустимо при независимости величин  $E$  и  $\mu$ ).

<тип распределения  $E$ > принимает одно из следующих значений:

D - дискретное распределение;

L - линейное распределение;

S - спектр деления  $^{235}\text{U}$  тепловым нейтроном;

F <имя нуклида> – спектр деления нуклида с параметрами, взятыми из библиотеки BNAB/MCU, где <имя нуклида> – идентификатор, содержащий не более четырёх символов и обозначающий нуклид так же, как и в данных для физического модуля, т.е. в соответствии с именами, приведенными в файле DEFAULT.PHY из банка данных программы.

Параметры D и L требуют обязательного наличия параметра <число>.

В случае дискретного распределения параметр <число> равен числу возможных значений  $E$  или  $\mu$ , а в случае линейного распределения – числу отрезков  $\{\mu_i\}$  или  $\{E_i\}$ .

Если для  $E$  или  $\mu$  используется линейное или дискретное распределение (параметры L или D в предложениях EDIS и MDIS, соответственно), то далее для каждого использованного параметра L или D необходимо задать ещё два массива: массив для узлов возможных значений этой величины (предложения EMES и MMES соответственно) и массив вероятностей выборки значений этой величины (предложения EPRO и MPRO соответственно). В случае дискретного распределения узлы есть возможные значения случайной величины, в случае линейного – границы отрезков в возрастающем порядке, в этом случае их на 1 больше, чем вероятностей. Вероятности, как всюду в источнике, можно задавать ненормированными.

После окончания описания спектров можно переходить к описанию примитивных источников.

Описание каждого примитивного источника начинается с предложения TYPE <тип частицы>

Возможны следующие значения параметра <тип частицы>, определяющего тип генерируемых этим примитивным источником частиц:

- N* - нейтроны;
- PH* – фотоны;
- EL* - электроны;
- PO* - позитроны.

Далее задаются характеристики примитивного источника, причём именно в том порядке, в котором они рассматриваются ниже.

В первую очередь необходимо определить контейнер примитивного источника, то есть область пространства, в которой для данного примитивного источника разыгрываются точки рождения. Контейнер определяется типом и параметрами, задающими предложение вида <контейнер> <массив параметров>

<контейнер> задает тело контейнера. В качестве контейнера можно использовать тела RPP, RCZ, SPH, HEXU либо RCZD. Во всех телах за исключением RCZD вероятность рождения частиц распределена равномерно по объёму. Для RCZD можно задать независимое по высоте и радиусу азимутально-симметричное распределение.

<массив параметров> задает координаты контейнера в глобальной системе координат (той же, которая использовалась при задании геометрических данных для геометрического модуля).

Несмотря на то, что тела, используемые в модуле источников в качестве контейнеров, похожи на аналогичные тела, используемые для задания геометрии системы, имеются некоторые различия. Рассмотрим каждое из них подробно.

*RPP* - прямоугольный параллелепипед, ориентированный по осям координат,  $\{(x_i): a_i \leq x_i \leq b_i\}$ . Равенство  $a_i = b_i$  возможно только для одного индекса  $i$ , и означает вырождение параллелепи-

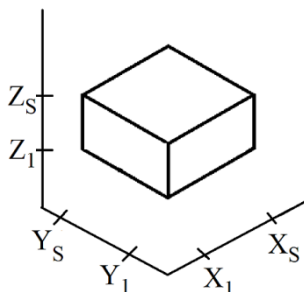


Рис. 41. Тело RPP

педа в прямоугольник, перпендикулярный  $i$ -й координатной оси. Это полезно для моделирования поверхностных источников.

*<массив параметров>* есть  $x_0$ ,

$x_1, y_0, y_1, z_0, z_1$ .

**RCZ** – сектор прямого цилиндрического слоя правильного кругового цилиндра с образующей, параллельной оси  $OZ$ ,  $\{(r, \varphi, z): r \leq r \leq R, \varphi_H \leq \varphi \leq \varphi_K, 0 \leq z \leq H\} + (x_0, y_0, z_0)$ . Точка  $(x_0, y_0, z_0)$  – середина нижнего основания;  $-\pi \leq \varphi_H < \varphi_K \leq \pi$ ;  $0 \leq H$ ;  $r \leq R$ , кроме того, не допускается одновременно  $H=0$  и  $r=R$ , то есть допускается либо трёхмерная область, либо сектор круга (в частности, полный круг), либо кусок цилиндрической поверхности.

*<массив параметров>* есть  $x_0$ ,

$y_0, z_0, H, R, r, \varphi_H, \varphi_K$

$x_0, y_0, z_0$  – координаты центра нижнего основания,  $H$  – высота цилиндра,  $R$  – внешний радиус цилиндра,  $r$  – внутренний радиус цилиндра. Параметры  $\varphi_H, \varphi_K$  задаются в градусах и определяют необходимый сектор. Отсутствие параметра  $r$  означает  $r=0$ , т.е. полный цилиндр, включая ось. Отсутствие параметров  $\varphi_H, \varphi_K$  означает полный круг:  $\varphi_H = -180^\circ, \varphi_K = 180^\circ$ . Равенство  $R=r$  позволяет определить поверхностный источник на цилиндрической поверхности. Равенство  $H=0$  позволяет определить поверхностный источник на круге.

**SPH** – сектор шара,  $\{(r, \varphi, \theta): r \leq r \leq R, \varphi_H \leq \varphi \leq \varphi_K, 0 \leq \theta \leq \pi\} + (x_0, y_0, z_0)$ . Точка  $(x_0, y_0, z_0)$  – центр шара;  $\pi \leq \varphi_H < \varphi_K \leq \pi$ ;  $r \leq R$  – часть шара или сферы.

*<массив параметров>* есть  $x_0, y_0, z_0, R, r, \varphi_H, \varphi_K$

Правила умолчания те же, что и для RCZ. Углы также задаются в градусах.

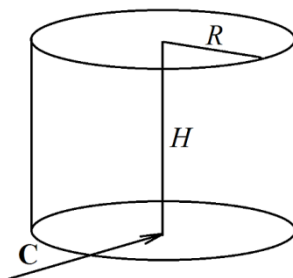


Рис. 42. Тело RCZ

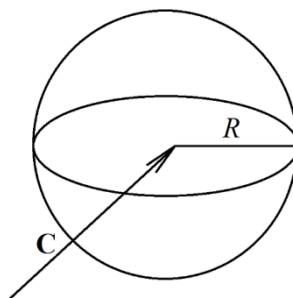


Рис. 43. Тело SPH

$HEXY$  - сектор правильной шестигранной вертикально ориентированной призмы.

$\langle \text{массив параметров} \rangle$  есть  $x_0, y_0, z_0, H, S, \varphi, Sb, Se$

$x_0, y_0, z_0$  - координаты центра нижнего основания;  $H$  - высота;  $S$  - размер под ключ.

Если прочие параметры опущены, то полная шестигранная призма расположена так, что две её грани параллельны плоскости  $Oxz$ . Параметр  $\varphi$  означает поворот данной призмы вокруг оси  $Z$  на  $\varphi$  градусов против часовой стрелки.

Номера треугольников  $Sb, Se$  служат для выделения секторов и задаются в соответствии с нумерацией, приведённой на рис. 45. Выделение секторов производится по направлению против часовой стрелки от меньшего номера ( $Sb$ ) к большему ( $Se$ ), поэтому используется как положительная, так и отрицательная нумерация.

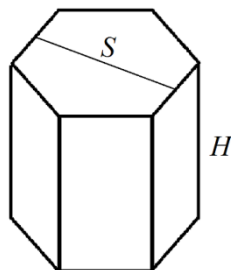


Рис. 44. Тело  $HEXY$

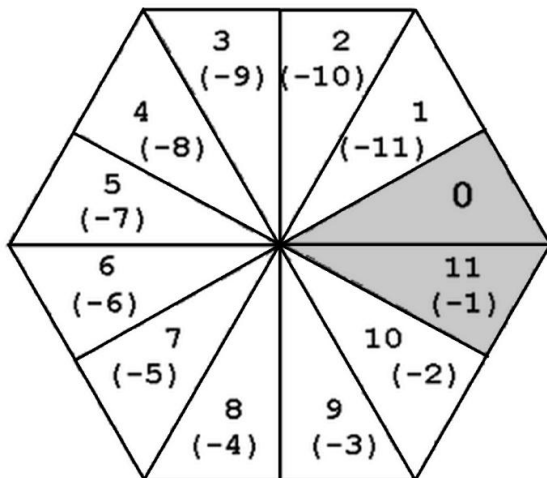


Рис. 45. Нумерация секторов шестиугольника

Например, для закрашенной на рис. 45 части шестигранной призмы можно использовать описание



HEXY X, Y, Z H D 0 -1, 0

Для оставшейся незакрашенной части –

HEXY X, Y, Z H D 0 1, 10

Описание

HEXY X, Y, Z H D 60 9, 10

выделяет часть шестигранной призмы, отличную от закрашенной, однако за счёт поворота на  $60^\circ$  пространственное распределение генерируемых частиц будет таким же.

*RCZD* - прямой круговой цилиндр с образующей, параллельной оси  $OZ$ , с внутренним распределением вероятностей рождения. Это тело приводится здесь лишь для полноты описания. Его использование требует понимания смещения распределений, и начинающему пользователю не стоит им пользоваться. Для такого цилиндра реализовано независимое по высоте и радиусу азимутально-симметричное распределение. Оба независимых распределения могут быть дискретными и кусочно-линейными.

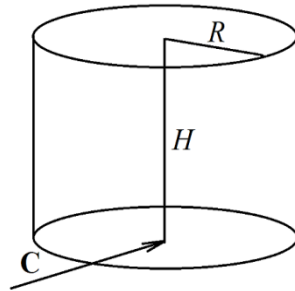


Рис. 46. Тело RCZD

<массив параметров> есть  $x_0, y_0, z_0, N_h, N_r$ .

Первые три параметра означают координаты центра нижнего основания.

При генерации вырабатываются случайные числа  $h$  и  $r$ , являющиеся внутренними координатами цилиндра, а также равномерно распределённый по окружности угол  $\psi$ . Координаты точек рождения определяются следующим образом:

$$x_0 + r \cos \psi, y_0 + r \sin \psi, z_0 + h.$$

$N_h$  – целое, отличное от нуля число, означающее тип распределения  $h$ .

Если  $N_h < 0$ , то распределение дискретно, и  $|N_h|$  равно числу возможных значений величины  $h$ .

Если  $N_h > 0$ , то распределение линейно, и  $|N_h|$  равно числу отрезков для  $h$ .

$N_r$  - целое отличное от нуля число, означающее тип распределения  $r$ .

Если  $N_r < 0$ , то распределение дискретно, и  $|N_r|$  равно числу возможных значений величины  $r$ .

Если  $N_r > 0$ , то распределение линейно, и  $|N_r|$  равно числу отрезков для  $r$ .

Случайная величина  $h$  может принимать при желании и отрицательные значения, в этом случае  $z_0$  не является центром нижнего основания. Иными словами, реальная высота и положение нижнего основания определяются задаваемым распределением.

Величина  $r$  всегда неотрицательна.

Далее необходимо задать собственно распределения. Для этого используются четыре предложения, задаваемые в указанном порядке:

HGRI <массив>

HPC <массив>

RGRI <массив>

RPC <массив>

Предложение *HGRI* служит для ввода массива действительных чисел, задающего положения  $h_i$ . В зависимости от определённого в предложении *RCZD* типа распределения вводятся либо  $|N_h|$  возможных значений высоты, либо границы  $|N_h|$  отрезков. В том случае, когда используется линейное разбиение, внутри каждого из отрезков случайная величина распределена равномерно.

Предложение *HPC* служит для ввода массива из пар чисел  $(v_i, f_i)$ , соответствующих значениям высоты или границам отрезков, заданных в предложении *HGRI*. Числа  $v_i$  означают реальные интенсивности моделируемого источника. Числа  $f_i$  означают с точностью до множителя вероятности, с которыми будут генерироваться частицы в программе.

Предложения *RGRI* и *RPC* позволяют аналогичным образом задать распределение вероятностей рождения частиц по радиусу.

Поскольку модуль источника помимо энергии частицы для определения направления её полёта генерирует ещё и косинус угла между вектором скорости частицы и реперным вектором, то после задания контейнера примитивного источника необходимо задать правило выработки реперного вектора. Это правило определяется строкой вида

REPER <массив параметров>

<массив параметров> может принимать следующие значения:

три действительных числа, задающие координаты постоянного реперного вектора, который нормируется на свою длину;

$R$  и три действительных числа, которые задают координаты характерной точки системы  $\mathbf{a}$ . В этом случае реперный вектор в точке рождения частицы  $\mathbf{x}$  определяется по формуле  $\mathbf{v}_p = (\mathbf{a}-\mathbf{x})/(\mathbf{a}-\mathbf{x}, \mathbf{a}-\mathbf{x})^{-0.5}$ ;

число 1 либо  $-1$ , которое определяет знак нормали ( $sgn$ ). При каждой генерации точки рождения примитивный источник вырабатывает зависящий от этой точки вектор нормали. В случае вырождения тела контейнера к плоской фигуре это перпендикуляр к плоскости, направленный по соответствующей оси. В случае цилиндрической или сферической поверхности – внешняя нормаль. В цилиндрическом и сферическом случаях вектор нормали вырабатывается и без вырождения тела к поверхности так, как будто внешний радиус уменьшен, и точка рождения лежит на цилиндрической или сферической поверхности. В случае тела RPP без вырождения конструкция 1 или  $-1$  недопустима.

После задания реперного вектора необходимо указать спектр, который будет использоваться при генерации частиц данным примитивным источником. Это осуществляется с помощью предложения SNAM <имя спектра>

Параметр <имя спектра> является именем одного из описанных ранее спектров. Указываемое здесь имя должно совпадать с именем, заданным для спектра с помощью его заглавного предложения ANGLEN.

Заканчивается описание примитивного источника указанием регистрационных объектов, заданных при описании геометрии рассчитываемой системы, которые должны ограничивать данный примитивный источник геометрически. Регистрационные объекты используются для задания геометрически сложных источников, поскольку модуль источников не имеет средств комбинаторного описания геометрии. В этом случае определяется список регистрационных объектов, описанных в геометрическом модуле, в которых разрешено рождение частицы. Если координаты частицы, выбранные в контейнере примитивного источника, не попадают ни в один регистрационный объект из списка, то они отбрасываются и генерируется новая точка рождения. Использование регистрационных объектов позволяет одновременно ввести пространственную зависимость весовых множителей. Вместе со списком объектов вводит-

ся список весовых множителей той же длины. Если точка рождения оказывается в  $i$ -м регистрационном объекте списка регистрационных объектов, то для нее используется  $i$ -й весовой множитель из списка множителей.

При описании сначала указывается количество регистрационных объектов. Для этого используется предложение

NOBJ  $\langle n \rangle$

Параметр  $\langle n \rangle$  – неотрицательное целое число, определяющее количество используемых регистрационных объектов. Если значение параметра равно нулю, то это означает, что данный примитивный источник не ограничивается никакими регистрационными объектами. Если значение параметра отлично от нуля, необходимо задать два дополнительных предложения:

LOBJ  $\langle$ массив чисел $\rangle$

WOBJ  $\langle$ массив чисел $\rangle$

В обоих случаях  $\langle$ массив чисел $\rangle$  – это массив из  $n$  целых чисел, где  $n$  – количество регистрационных объектов, заданное в предложении NOBJ. Для предложения LOBJ в этот массив записываются номера регистрационных объектов, используемых для ограничения геометрической области рождения. Для предложения WOBJ в этот массив записываются веса соответствующих объектов. Так как в предложении WOBJ указываются веса, а не вероятности, то никаких перенормировок в дальнейшем не происходит.

Использование регистрационных объектов и их весов требуется только при задании очень сложных источников и не рекомендуется начинающему пользователю.

На этом описание примитивного источника закончено.

Сложный источник может состоять из любого количества примитивных источников, каждый из которых может использовать свой собственный спектр и генерировать любой из доступных типов частиц. Для этого просто описывается необходимое количество спектров и примитивных источников, координаты которых соответствуют их реальному положению в пространстве. Для упрощения задания сложных источников, содержащих очень большое количество примитивных источников, можно использовать генератор решетки, называющийся S2AR. Этот генератор задает правильную параллелограммную решетку, аналогичную генерируемой генератором решетки G2AR геометрического модуля. Поскольку эта оп-

ция на практике применяется редко, оставим её для самостоятельного изучения.

Рассмотрим несколько практических примеров.

В первую очередь установим простой точечный источник в точку (0, 0, 3). Для этого достаточно задать всего одно предложение. Такое применение простого точечного источника достаточно часто встречается на практике при решении задач на собственное значение:

```
SPNT 0 0 3
FINISH
```

Однако, как говорилось выше, для быстрого выхода на распределение частиц, близкое к собственной функции, целесообразно сразу разбросать нейтроны по некоторой пространственной области. Для этого воспользуемся возможностями сложного источника.

Зададим источник, у которого областью рождения нейтронов является цилиндр с центром нижнего основания в точке (0,0,0), высотой 20 см, радиусом 10 см и осью, совпадающей с OZ:

```
NPS 1 ; имеется лишь 1 примитивный источник
PROB 1 ; поэтому вероятность его срабатывания 1
ANGLN SS ; начало описания спектра с именем SS
MDIS S ; распределение направлений изотропное
EDIS S ; стандартное распределение энергии
TYPE N ; начало примитивного источника нейтронов
RCZ 0,0,0 20 10 ; равномерное распределение
*                по цилиндру
SNAM SS ; энергии и направления определены по SS
REPER 1 0 0 ; репер здесь не важен – изотропность
NOBJ 0 ; ограничение объектами не используется
FINISH
```

Рассмотрим более сложный пример – источник термоядерных нейтронов для установки типа ТОКАМАК. Для простоты будем считать плазменный шнур «тором» с сечением в виде прямоугольника:

```
NPS 1
PROB 1
EQU RSO = 70 ; outer radius of source
EQU RSI = 50 ; inner radius of source
EQU HS2 = 30 ; half of source height
ANGLN CFCS
MDIS S
```

```

EDIS D 1
EMES 14.1E6
EPRO 1.
TYPE N
RCZ 0,0, -HS2 2*HS2 RSO RSI
REPER 1
SNAM CFCS
NOBJ 0
FINISH of source

```

В этом примере энергия источника всегда равна 14,1 МэВ. Поэтому для энергии используется дискретное распределение. Значение 14,1 МэВ задано в предложении EMES. Так как имеется лишь одно значение для энергии, то значение параметра предложения EPRO не играет роли. Контейнер источника представляет цилиндрическое кольцо с внешним радиусом RSO и внутренним радиусом RSI. Здесь эти константы и высота цилиндра определены прямо среди данных источника, хотя в задаче, откуда взят пример, они находились среди данных геометрического модуля.

Следующий пример иллюстрирует задание поверхностных источников, для которых распределение направлений уже не изотропно. Две параллельные прямоугольные пластинки, содержащие поверхностные источники нейтронов, которые излучают по нормальям к пластинкам, равновероятны в обе стороны. Спектр обеих пластинок одинаков по энергии и сосредоточен в пяти точках:

```

NPS 2
PROB 1 1
ANGLEN SPCN
MDIS D 2
EDIS D 5
MMES -1 1
MPRO 0.5 0.5
EMES 10E3 15E3 18E3 19.5E3 22E3
EPRO 1 1 4 2 2
TYPE N
RPP 0,0 -1,1 0,3
REPER 1
SNAM SPCN
NOBJ 0
TYPE N
RPP 0.6,0.6 -1,1 0,3

```

```
REPER 1
SNAM SPCN
NOBJ 0
FINISH
```

Равновероятное направление, перпендикулярное пластинке, достигается тем, что для косинуса угла с реперным вектором допускается только два значения:  $-1$  и  $1$ , т.е. направление вдоль реперного вектора или противоположное. Так как контейнер каждого примитивного источника вырожден в прямоугольник, то реперный вектор можно определить как нормаль к прямоугольнику, что достигается конструкцией REPER 1. Для обоих примитивных источников реперный вектор направлен по оси  $OX$ .

В Руководстве пользователя имеется набор примеров, иллюстрирующих более сложные конструкции.

### 14.3. СМЕЩЕНИЕ РАСПРЕДЕЛЕНИЙ

Смещение распределений является совершенно общей конструкцией для метода Монте-Карло. Его несложно понять для чисто дискретных или чисто непрерывных распределений – разница лишь в том, пишем ли мы суммы с весом вероятностей или интегралы с весом плотностей вероятностей. Рассмотрим случай дискретных распределений.

Пусть имеется случайная величина  $X$ , принимающая значения  $\{x_k\}$  с вероятностями  $\{f_k\}$ . Вместо неё мы будем моделировать другую случайную величину с другими вероятностями  $\{p_k\}$ , а значения новой величины получим из первой с помощью корректирующего множителя  $f_k/p_k$ . Эти множители обычно называются весами, тогда математическое ожидание новой случайной величины

$$f_1/p_1 * x_1 * p_1 + \dots + f_N/p_N * x_N * p_N = f_1 * x_1 + \dots + f_N * x_N.$$

Таким образом, математическое ожидание новой и старой величины совпадают.

Рассмотрим теперь более сложный случай. С вероятностью  $f_k$  выбирается  $k$ -я случайная величина из набора  $\{X_k\}$ ;  $\{x_k\}$  есть их математические ожидания (условные математические ожидания при выборе  $k$ ). Очевидно, что правая часть предыдущей формулы есть математическое ожидание  $X$ . Новую случайную величину введем с помощью тех же весов. При выборе  $k$  множитель, то есть вес, становится константой, поэтому он же есть множитель для условного

математического ожидания, так что левая часть формулы есть математическое ожидание новой случайной величины, оно совпадает с математическим ожиданием старой.

Таким образом, вместо выбора случая, например примитивного источника, по вероятностям  $\{f_k\}$ , которые соответствуют природе моделируемого процесса, мы можем использовать более удобные вероятности  $\{p_k\}$ , добиваясь сохранения математического ожидания с помощью весов. Эта замена случайных величин называется смещением распределений.

При разумном выборе весов или, что то же самое, используемых при моделировании вероятностей  $\{p_k\}$  можно при сохранении математического ожидания существенно уменьшить дисперсию. Однако при плохом их выборе дисперсию можно и увеличить, поэтому использование смещений требует определенного навыка.

В модуле источников смещение указывается с помощью символа BIAS. При этом вместо одного набора из  $N$  чисел  $\{f_k\}$  нужно задать два набора:  $\{f_k\}$  и  $\{p_k\}$ . Если после символа BIAS нет дополнительной буквы, то числа задаются парами  $\{f_k, p_k\}$ . Если же используется конструкция BIAS M, то сначала идут все числа  $\{f_k\}$ , а потом все  $\{p_k\}$ .

Смещение может использоваться при выборе примитивных источников. При этом возникают конструкции

```
NPS <n> BIAS  
PROB <массив из 2*n чисел>
```

или

```
NPS <n> BIAS M  
PROB <массив из 2*n чисел>
```

В обоих случаях в предложении PROB содержится  $2 \times n$  чисел, разница лишь в порядке их записи.

Также смещения могут применяться при табличном задании распределений для энергии или косинуса угла отклонения от реперного вектора. Во втором случае могут использоваться предложения

```
MDIS D N BIAS M  
MMES <массив из N чисел>  
MPRO <массив из 2*N чисел>
```

Второе предложение точно такое же, как если бы смещение не использовалось.



В случае отсутствия буквы М в конце первого предложения второе предложение остается без изменения, а в третьем меняется порядок чисел.

Для линейного распределения коррекции точно такие же:

MDIS L N BIAS M

MMES <массив из N+1 чисел>

MPRO <массив из 2\*N чисел>

Дело в том, что с введением смещения меняется лишь предложение MPRO, а его смысл в линейном и дискретном случае одинаков.

Смещение для распределения энергии задается совершенно аналогично с заменой предложений MDIS, MMES, MPRO на EDIS, EMES, EPRO.

## ГЛАВА 15. ИСХОДНЫЕ ДАННЫЕ ДЛЯ МОДУЛЯ РЕГИСТРАЦИИ

Модуль регистрации предназначен для вычисления некоторого «стандартного» набора линейных и дробно-линейных функционалов потока и их статистических ошибок.

Линейные функционалы выражаются интегралами вида

$$I = \int_{\delta V} d\mathbf{r} \int_{\delta \Omega} d\Omega \int_{\delta E} dE \varphi(\mathbf{r}, \Omega, E) \Phi(\mathbf{r}, \Omega, E),$$

где  $d\mathbf{r}d\Omega dE$  – элемент объема в фазовом пространстве, в котором координаты частицы определяются радиусом-вектором  $\mathbf{r}$ , направлением скорости  $\Omega$  и значением энергии частицы  $E$ ;  $\varphi(\mathbf{r}, \Omega, E)$  – весовая функция, определяющая конкретный вид функционала;  $\Phi(\mathbf{r}, \Omega, E)$  – функция распределения потока нейтронов.

Дробно-линейные функционалы (например, средние сечения) являются частными от деления двух линейных функционалов.

Статистическая ошибка расчёта линейных функционалов (статистическая погрешность) определяется как среднеквадратичное отклонение

$$\sigma = \sqrt{D},$$

где  $D$  – выборочное значение дисперсии случайной величины, используемой для оценки функционала.

В расчётах по методу Монте-Карло обычно приводят относительную статистическую ошибку, то есть  $\sigma/m$ , где  $m$  – оценка соответствующего функционала.

Дисперсии дробно-линейных функционалов вида  $z=x/y$  вычисляются по формуле

$$Dz = \frac{\bar{x}^2}{\bar{y}^2} \left( \frac{Dx}{\bar{x}^2} - 2 \frac{M[(x-\bar{x})(y-\bar{y})]}{\bar{x}\bar{y}} + \frac{Dy}{\bar{y}^2} \right),$$

где  $\bar{x}$  и  $\bar{y}$  – выборочные значения математических ожиданий случайных величин  $x$  и  $y$ , а  $D$  и  $M$  – соответственно символы дисперсии и математического ожидания.

Это приближенная формула, но в реальных расчётах ей можно пользоваться.

Функционалы вычисляются с помощью стандартных оценок, используемых в методах Монте-Карло:

по точкам столкновений,  $u^{(col)} = \sum_{m=1}^M \frac{\varphi(\mathbf{r}_m, E)}{\Sigma_{tot}(\mathbf{r}_m)} W_m$ , где  $m$  – номер точки столкновения,  $\Sigma_{tot}$  – полное сечение, а  $W_m$  – статистический вес частицы в точке столкновения (подробно понятие статистического веса дано в руководстве, раздел транспортного модуля или в любой монографии по методу Монте-Карло);

по длине пробега,  $u^{(l)} = \sum_{m=1}^M \varphi(\mathbf{r}_m, E) \ell_{m,m-1} W_m$ , где  $m$  – номер точки пересечения траектории нейтрона с границей геометрической зоны,  $\ell_{m,m-1}$  – длина траектории в зоне, а  $W_m$  – вес в точке регистрации;

по точкам поглощения,  $u^{(a)} = \frac{\varphi(\mathbf{r}_a, E)}{\Sigma_a(\mathbf{r}_a, E)} W_a$ , где  $\Sigma_a$  – сечение поглощения, а  $W_a$  – вес в точке поглощения.

В качестве областей интегрирования в трёхмерном пространстве ( $\delta V$ ) в МСУ используются следующие *регистрационные области*:

- полный объём  $V$ , занятый системой в пространстве;
- регистрационные материалы – объединения нескольких геометрических зон, состоящих из одинаковых материалов;
- регистрационные зоны – объединения нескольких геометрических зон, которые могут состоять из различных материалов;
- регистрационные объекты – альтернативные объединения нескольких геометрических зон, которые могут состоять из различных материалов.

Интегрирование по энергии ведется в энергетических группах ( $\delta E$ ), границы которых определяются пользователем. Важно понимать, что эти группы используются только модулем регистрации и никак не влияют на собственно процесс моделирования траекторий частиц. В MSU поток или скорость реакции всегда интегрируется по всей сфере направлений.

Таким образом, регистрация функционалов определяется выбранным типом оценки, регистрационными областями, энергетическими группами и весовой функцией.

Все функционалы нормируются на поток от одной первичной частицы источника. Такая нормировка общепринята для программ, реализующих метод Монте-Карло. При этом результат накапливается в соответствующем объеме, например в каждом материале. В MSU автоматического деления накопленной статистики на объем регистрационной области не происходит.

При определении параметров регистрации функционалов необходимо учитывать следующие основные особенности метода Монте-Карло. Среднеквадратичное отклонение убывает обратно пропорционально количеству промоделированных частиц. Чем меньше объем регистрационной области, тем меньше в ней регистрируется частиц, а значит, необходимо моделировать больше частиц для достижения необходимого уровня точности. Аналогично при разбиении энергетического диапазона на группы необходимо учитывать, что чем меньше интервал, тем меньше в нем регистрируется нейтронов. И наконец, поглощений обычно меньше, чем столкновений, а пробеги присутствуют всегда.

Остановимся подробнее на выборе оценки для регистрации. Наиболее часто используемая оценка – оценка по пробегам. Это связано с тем, что при необходимости регистрации в регистрационных объемах, в которых происходит мало столкновений, такая оценка – единственно возможный способ набора статистики с необходимой точностью за приемлемое время. В MSU по умолчанию используется регистрация по столкновениям, что является традиционным для этой программы подходом. В MSU-5 была добавлена возможность регистрации по поглощениям. Однако в связи с тем, что в некоторых материалах (например, воде) у всех составляющих такие материалы нуклидов сечение поглощения равно нулю на некоторых энергетических интервалах, а полное сечение в этих ин-

тервалах не равно нулю, происходит занижение расчётного значения потока, поскольку факта регистрации в соответствующем энергетическом интервале не происходит. Поэтому использование регистрации по поглощениям для оценки потока частиц не рекомендуется.

Все рассчитываемые программой MCU функционалы можно условно разделить на две группы:

- глобальные функционалы, являющиеся интегралами по всей области фазового пространства, в которой определена рассчитываемая система;
- функционалы для регистрационных областей и энергетических групп – интегралы по объёму, занимаемому областью.

Глобальные функционалы оцениваются только для нейтронов. Рассмотрим основные рассчитываемые функционалы, характеризующие систему в целом: эффективный коэффициент размножения, утечку и баланс нейтронов.

*Эффективный коэффициент размножения* нейтронов  $k_{eff}$  определяется как математическое ожидание числа вторичных нейтронов деления на один первичный нейтрон, рожденный в предыдущем поколении. При этом в программе MCU принято, что ко вторичным нейтронам деления не относятся нейтроны, родившиеся в результате  $(n, kn)$  реакций.

Вычисляется несколько оценок эффективного коэффициента размножения, которые могут различаться как своими значениями, так и значениями соответствующих статистических ошибок:

- по точкам столкновений  $k_{col}$ ;
- по точкам поглощений  $k_{abs}$ ;
- по пробегам  $k_{len}$ ;
- комбинированная оценка по столкновениям и поглощениям, вычисляемая по формуле  $k_{com} = a \cdot k_{col} + (1 - a) \cdot k_{abs}$ ;
- модифицированная комбинированная оценка Бриссендена, вычисляемая по формуле  $k_{Bris} = a \cdot k_{col} + b \cdot (1 - A)$ .

Здесь  $a$  и  $b$  - некоторые весовые множители,  $A$  - оценка суммы числа поглощённых и вылетевших из системы первичных частиц. Весовые множители  $a$  и  $b$  в комбинированных оценках выбираются из условия минимизации выборочного значения дисперсии данной

величины. Число поглощённых частиц оценивается по столкновениям, а вылетевших из системы - прямым подсчётом.

В данном пособии рассматриваются преимущественно аналоговые методы, когда функции распределения вероятностей, используемые при моделировании траекторий частиц, со всей возможной точностью совпадают с физическими законами. Такой способ оптимален для относительно простых расчётов. В более сложных случаях используют так называемые неаналоговые методы, в которых вероятности событий при моделировании отличаются от реальных, и это отличие компенсируется статистическим весом.

Все оценки являются равноправными, и рекомендуется использовать значение имеющей наименьшую статистическую ошибку.

*Баланс нейтронов* рассчитывается как сумма поглощенных и покинувших систему через внешнюю границу первичных частиц и должен быть строго равен 1, если в системе не произошло потери частиц в результате приближенных вычислений в геометрическом модуле.

Для каждой регистрационной области в заданных энергетических группах вычисляются поток, скорости реакций нескольких заданных типов как для каждого нуклида, так и их композиции, обозначаемой MIXT в файле с результатами расчётов, а также ряд других функционалов, которые здесь рассматриваться не будут. Скорости реакций определяются следующей формулой:

$$R_n = \int \Sigma_n(\mathbf{r}, E) \Phi(\mathbf{r}, \Omega, E) d\mathbf{r} d\Omega dE,$$

где  $\Sigma_n(\mathbf{r}, E)$  – макроскопическое сечение ядерной реакции с номером  $n$  (либо для отдельного нуклида, либо для их композиции в данной регистрационной области).

Нумерация нейтронных скоростей реакций совпадает с принятой в файлах оцененных ядерных данных. Наиболее часто используемые нейтронные реакции:

3 – поглощение;

18 – деление;

918 - произведение  $v\Sigma_{fis}$ .

В отдельную категорию можно выделить так называемые дозиметрические реакции. Вычисления выполняются так же, как и в случае с обычными скоростями реакций, с той лишь разницей, что

вместо макроскопических сечений в искомый функционал подставляются сечения активации из библиотеки DOSIM. При расчёте скоростей реакции детектора с использованием активационных сечений результат измеряется в числе реакций в регистрационной области на один рожденный в системе нейтрон в предположении, что ядерная плотность изучаемого материала равна  $10^{-24}$  ядер/см<sup>3</sup>. Важно понимать, что обработка дозиметрических реакций выполняется исключительно средствами регистрационного модуля и не оказывает никакого влияния на собственно моделирование траектории частицы.

Признаком начала ввода данных к модулю служит предложение RGS, а знаком окончания – предложение FINISH.

В общем случае этот фрагмент состоит из следующих разделов:

```
RGS
<раздел общих параметров>
<дополнительный раздел параметров 1>
. . .
<дополнительный раздел параметров N>
FINISH
```

Все разделы могут быть пропущены. В этом случае модуль регистрации не будет фиксировать никакие события, что ускоряет работу программы. Такой режим может оказаться полезным, например, при расчётах изменения изотопного состава материалов в процессе эксплуатации ядерного реактора, т.е. при расчётах выгорания, поскольку вся необходимая для модуля выгорания регистрация ведется специальным модулем регистрации для выгорания.

Порядок ввода разделов внутри фрагмента произволен.

Предложение с именем, начинающимся с буквы M, относится к материалам, с буквы Z - к зонам, а с буквы O - к объектам. Под параметром *list* будет пониматься список чисел, члены которого можно указывать как через пробел, так и в виде диапазона. Например, запись 1, 3–5, 7 означает 1,3,4,5,7. Использование числа 0 при задании номеров регистрационных областей означает включение регистрации во всех областях соответствующего типа.

Первым обязательным предложением данных для модуля регистрации является предложение RGS. Оно имеет два численных параметра. Первый отвечает за вывод в файл NAME.LST или NAME.FIN строк исходных данных по мере их ввода, а второй - за

вывод в этот файл дополнительной информации. Обычно предложение используется в виде

```
RGS 1 0
```

В разделе общих параметров заказывается регистрация некоторых стандартных функционалов, а также некоторые общие для всех видов регистрации параметры. Заказ осуществляется с помощью предложений. Рассмотрим только наиболее часто употребляемые предложения:

```
KEFF  
BAL  
NUCOFF  
PERC value1  
NRET value1 value2
```

*KEFF* – при наличии этого предложения будет рассчитываться эффективный коэффициент размножения нейтронов в рассчитываемой системе.

*BAL* – при наличии этого предложения будут вычисляться функционалы баланса нейтронов в системе, в том числе утечка из системы.

*NUCOFF* – при наличии этого предложения скорости реакций в регистрационных областях вычисляются только для композиции нуклидов, регистрация скоростей реакций для нуклидов в отдельности не производится.

*PERC* – определяет способ печати статистических погрешностей.

*value1* – при значении равном 0 погрешности печатаются в абсолютных величинах, при 1 – в процентах (значение по умолчанию – 1).

*NRET* – определяет порядок перебора характеристик функционала при печати (*NRET* – Nuclide Reaction Energy Tally).

*value1* – четырёхзначное число, определяющее порядок перебора характеристик. Каждая цифра определяет очерёдность перебора соответствующей характеристики. При этом цифра, определяющая тысячи, соответствует нуклиду, сотни – реакции, десятки – энергии, единицы – области регистрации (материалу, зоне или объекту). Чем меньше цифра, тем быстрее изменяется соответствующая характеристика. Например, число 3214 определяет стандартный для собранных на базе пакета MCU-4 программ перебор: быстрее всего изменяется номер энергии, затем – номер реакции, далее – нуклид и

в последнюю очередь – область регистрации. Значение по умолчанию – 4321.

*value2* – символьная константа, определяющая направление перебора наиболее быстро меняющейся характеристики. Может принимать значение UP или DOWN.

Заметим, что предложения PERC и NRET используются только на этапе финальной обработки накопленной статистики. Значения этих предложений могут быть изменены перед выполнением финальной обработки, а выполнять финальную обработку можно неограниченное количество раз. Это позволяет получать любые требуемые для анализа результаты таблицы.

Кроме раздела общих параметров в регистрационном модуле можно задать любое количество дополнительных разделов параметров регистрации. Каждый такой раздел предназначен для регистрации одного типа частиц. Можно определять любое количество разделов для частиц одного типа. В этих разделах можно осуществлять накопление статистики с различными условиями.

Общий вид задания дополнительного раздела:

```
PType value
TType value
MFLU list
ZFLU list
OFLU list
LEAK
MRCT list
ZRCT list
ORCT list
RCT list
MDOS list
ZDOS list
ODOS list
DOS list
ENERGY list
END
```

Каждый дополнительный раздел должен начинаться предложением PTYPE, а заканчиваться – предложением END. Остальные предложения могут отсутствовать.

*PTYPE* – обязательное первое предложение раздела.



*value* – номер типа частиц, для которых при накоплении статистики будут использоваться заданные в этом разделе условия. Нумерация соответствует нумерации, принятой в физическом модуле (1 – нейтроны, 2 – фотоны, 3 – электроны, 4 – позитроны).

*TTYPE* – предложение, определяющее способ оценки функционалов.

*value* – число, определяющее способ оценки функционалов (0 – по точкам столкновений, 1 – по длине пробега, 2 – по точкам поглощений). Значение по умолчанию – 0.

*MFLU*, *ZFLU*, *OFLU* – предложения, определяющие номера регистрационных областей (соответственно материалов, зон и объектов), в которых будут рассчитываться потоки.

*list* – список номеров соответствующих регистрационных областей (соответственно материалов, зон и объектов).

*LEAK* – при наличии этого предложения в регистрационных областях, определённых с помощью предложений *MFLU*, *ZFLU* или *OFLU* (соответственно материалов, зон и объектов), вместо потока будет суммироваться вес частиц, что позволяет рассчитывать утечку. Обычно для этого вокруг интересующей нас области пространства задается достаточно тонкая граничная область, в которой и суммируется вес частиц. Поскольку столкновений и тем более поглощений в такой области будет мало или не будет вообще, в данном случае обычно используется оценка по длине пробега.

*MRCT*, *ZRCT*, *ORCT* – предложения, определяющие номера регистрационных областей (соответственно материалов, зон и объектов), в которых будут рассчитываться скорости реакций, заданные с помощью предложений *RCT*.

*list* – список номеров соответствующих регистрационных областей (соответственно материалов, зон и объектов).

*RCT* – это предложение определяет номера реакций (согласно нумерации физического модуля), которые рассчитываются в областях, определённых предложениями *MRCT*, *ZRCT*, *ORCT*.

*list* – список номеров реакций. Для указанного списка реакций будет вычисляться также понуклидный баланс для всей системы.

*MDOS*, *ZDOS*, *ODOS* – эти предложения определяют номера регистрационных областей (соответственно материалов, зон и объектов) в которых будут рассчитываться скорости дозиметрических реакций, заданные с помощью предложения *DOS*.

*list* – список номеров соответствующих регистрационных областей (материалов, зон и объектов).

*DOS* – предложение, определяющее реакции из библиотеки DOSIM, которые будут рассчитываться в областях, определённых предложениями MDOS, ZDOS, ODOS. Каждая реакция задаётся двумя константами символьного типа, разделёнными пробелом. Первая константа определяет название изотопа (содержится в первых четырёх позициях имени файла в библиотеке, при этом следует игнорировать символ Q), вторая – источник и номер реакции (содержатся в следующих четырёх позициях имени файла в библиотеке). Например, задание пары C12 R16 соответствует файлу C12QR16.ACT.

*list* – список пар символьных констант, разделённых пробелом.

*ENERGY* – предложения, определяющие значения нижних границ энергетических регистрационных групп (в электрон-вольтах) для одной энергетической сетки. Верхняя граница последней энергетической группы равна бесконечности. Каждое предложение задаёт одну независимую энергетическую сетку. Регистрация ведётся отдельно для каждой сетки. Внутри одного раздела может быть использовано любое количество таких предложений. При отсутствии хотя бы одного предложения регистрация для данного раздела не ведётся. Отсутствие значений в предложении приводит к ошибке.

*list* – нижние границы регистрационных групп в электрон-вольтах. Значение 0 должно быть задано явно.

## **ГЛАВА 16. ИСХОДНЫЕ ДАННЫЕ ДЛЯ ТРАНСПОРТНОГО МОДУЛЯ**

Транспортный модуль программы MCU осуществляет организацию моделирования отдельных траекторий частиц. Для всех карт, используемых в разделе данных для транспортного модуля, заданы значения по умолчанию, поэтому раздел может не содержать никаких карт, кроме обязательной карты FINISH в конце раздела. Рассмотрим следующие карты этого раздела:

```
NTOT ntot
NBAT nbatch
NSKI nskip
ISTR istr
MIXG nmixg
FINISH
```

Карта NTOT определяет число частиц в поколении. По умолчанию  $ntot = 200$ . Напомним, что систематическая ошибка расчёта функционалов убывает обратно пропорционально  $ntot$ .

Карта NBAT определяет количество поколений в одной серии. Каждые последовательные  $nbatch$  поколений частиц, называемых сериями, считаются независимыми. По умолчанию  $nbatch = 3$ . Точность расчёта статистических ошибок рассчитываемых функционалов возрастает с увеличением  $nbatch$ .

Значения NBAT и NTOT, принятые по умолчанию, разумны при расчёте ячеек или решении задач с источником.

Карта NSKI определяет число отбрасываемых серий, то есть нескольких первых серий, не учитываемых при накоплении статистики. По умолчанию  $nskip = 0$ . Отбрасывание первых серий применяется в задачах на собственное значение для устранения влияния начальных поколений, имеющих далёкое от реального распределение источника. Распространённой практикой является отбрасывание первых трёх поколений, а поскольку в MCU отбрасываются серии, то обычно достаточно отбросить только первую серию.

Карта ISTR определяет тип решаемой задачи. По умолчанию  $istr = 0$ . В этом случае решается задача о критичности системы и вычисляется эффективный коэффициент размножения. При  $istr = 1$  решается задача с заданным внешним источником. При  $istr = 2$  решается задача с заданным внешним источником и блокируется использование нейтронов деления. Этот режим предназначен для задач, в которых применяется рассчитанная каким-то другим методом плотность нейтронов деления.

Карта MIXG управляет нормализацией поколений в многопроцессорном режиме. При  $nmixg = 0$  нормализация выполняется независимо на каждом процессоре, что соответствует стандартной идеологии распараллеливания расчётов MCU. При отличном от нуля  $nmixg$ , после того как все процессоры промоделировали одно поколение нейтронов, выполняются сбор данных по всем процессорам, общая нормализация и раздача нормализованных данных процессорам. Таким образом, в этом режиме многопроцессорный расчёт полностью аналогичен по идеологии обычному однопроцессорному расчёту. По умолчанию  $nmixg = 0$ .

Карта FINISH, обозначающая признак конца фрагмента, является единственной обязательной картой раздела данных для транспортного модуля.

## ГЛАВА 17. ИСХОДНЫЕ ДАННЫЕ ДЛЯ УПРАВЛЯЮЩЕГО МОДУЛЯ

Приведенные здесь данные позволяют пользователю управлять режимом расчёта, все карты после выполнения шага INPUT, записываются в файл NAME.DAT и могут редактироваться перед выполнением шагов CALCULATION и OUTPUT.

Рассмотрим следующие карты:

```
NAMV namvar
MAXS maxser
DTZM nser
NPRI nhys
BETA
NRAN nran
ECUT ecut
FINISH
```

Раздел, начинающийся с обязательной карты NAMV, состоит из нескольких необязательных карт и заканчивается обязательной картой FINISH - признаком окончания ввода данных к разделу.

С помощью карты NAMV задается название варианта *namvar* - текстовая переменная длиной не более 12 символов. Это название выводится на экран в процессе расчёта и служит для визуальной идентификации расчёта.

Карта MAXS определяет количество рассчитываемых серий (*maxser* – номер серии, после которой счёт прекращается). При наличии ненулевого количества отбрасываемых серий (карта NSKI) параметр *maxser* определяет число регистрируемых серий, то есть счёт окончится после моделирования числа серий, равного сумме параметров карт NSKI и MAXS. Значение по умолчанию 1 000 000.

Карта DTZM служит для определения интервала записи промежуточных результатов расчёта на диск (*nser* - количество серий, через которое накопленная информация записывается в файл задачи). Значение по умолчанию 1.

Карта NPRI определяет количество историй *nhys*, через которое на экран поступает информационная строка. Значение по умолча-

нию 0. При использовании этой карты следует иметь в виду, что вывод информационной строки на экран замедляет работу программы.

Карта BETA включает режим вычисления эффективной доли запаздывающих нейтронов. Отметим, что карту BETA следует включать только в случае присутствия в данных физического модуля предложения DELN 1.

Параметр *nran* (произвольное целое число) карты NRAN служит для управления выборкой случайных чисел. Эта карта используется для продолжения расчётов при фатальных авостах. Основная идея состоит в том, что если после фатального останова задачи перезапустить её с другой последовательностью случайных чисел, то вызвавшая эту фатальную ошибку траектория не повторится. Для продолжения расчёта с другой последовательностью случайных чисел нужно вновь запустить шаг, добавив карту NRAN в файл NAME.DAT или, если она уже задана, изменив значение *nran*.

С помощью карты ECUT можно установить границу энергии нейтрона в электрон-вольтах, ниже которой траектории прерываются. Область изменения параметра *ecut* определяется диапазоном работы нейтронных подмодулей, заданных в разделе исходных данных для физического модуля. Значение по умолчанию – 0 эВ. Аналогичные карты существуют и для всех остальных типов частиц, моделируемых программой MCU.

Обязательная карта FINISH завершает раздел данных для управляющего модуля.

## **ГЛАВА 18. ПРИМЕР ИСХОДНЫХ ДАННЫХ ДЛЯ РАСЧЕТА**

В этом разделе представлен пример реально выполнявшегося расчёта жидкосолевого ториевого blankets термоядерного источника нейтронов.

Данный расчёт носил прикидочный характер, поэтому не было необходимости делать точное описание установки. В действительности вакуумная камера установки имеет форму, характерную для сферических токамаков, но в расчётной модели она заменена цилиндрическим кольцом. Аналогично, реальным источником термоядерных нейтронов должен быть плазменный шнур, имеющий

форму бублика с некруговым сечением. Он заменен на эквивалентный источник в форме цилиндрического кольца.

В расчёте были существенны лишь те части конструкции, в которых происходит значительная потеря нейтронов или, напротив, полезные ядерные реакции. Поэтому в расчётной модели, показанной на рис. 47 и 48, используется лишь 13 материалов.

На рис. 47 дан разрез вертикальной плоскостью всей модели, поэтому масштаб не позволяет увидеть четыре материала, составляющие первую стенку – бериллий, бронза, вода и сталь. Они представлены на рис. 48.

В модели, между вакуумной камерой и бланкетом, помещён свинцовый экран, играющий две роли. За счёт реакции  $n \rightarrow 2n$  свинец играет роль дополнительного размножителя нейтронов, кроме того прохождение через свинец несколько смягчает спектр нейтронов, что положительно сказывается на соотношении скоростей ядерных реакций в расплаве солей бланкета. Вся конструкция окружена графитом, играющим роль отражателя нейтронов. В реальной установке такого сплошного отражателя быть не может, но в описываемой работе требовалось оценить эффективность использования нейтронов сверху.

Внешняя поверхность бланкета в модели, послужившей основой рассматриваемой здесь, была образована вращением дуги окружности, лежащей в вертикальной плоскости вокруг оси  $OZ$ , то есть была образована частью поверхности тора. В программе MCU среди тел нет тора, поэтому поверхность, приближена набором конических поверхностей. Для этого использованы тела TRC – усечённые правильные круговые конусы с именами: TM1 – TM4, TP1 – TP4. Эти конические поверхности получены путём вращения вокруг оси  $OZ$  ломанной, вписанной в упомянутую дугу окружности. Параметры усечённых конусов определены с помощью включённых в текст арифметических выражений. Данный способ значительно лучше предотвращает ошибки, чем отдельный расчёт необходимых значений и их последующее использование в исходных данных в виде чисел.

Еще один полезный приём иллюстрирует строка EQU TOFS = 5. Предполагалось провести серию расчётов с различными толщинами свинцового экрана с целью выбрать лучший вариант.

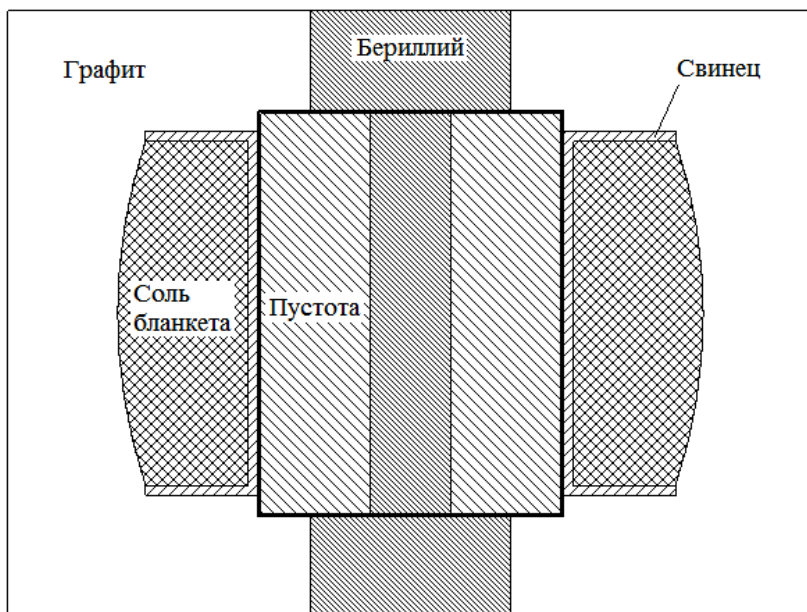


Рис. 47. Материальные зоны для расчёта бланкета

Константа TOFS – толщина экрана. Через эту толщину определяются другие величины, зависящие от неё (например, внутренний радиус бланкета – радиус цилиндра U1).

Следующий использованный приём также обусловлен тем, что данный расчёт входил в большую серию, связанную с выбором различных параметров. Легко видеть, что указанные в разделе данных для физического модуля материалы с номерами 6 и 10 не используются – в описании геометрии нет зон с такими материалами.

В данном варианте в качестве материала бланкета используется расплав смеси фторидов лития, натрия и калия, в которых растворен тетрафторид тория. Исследовались варианты с наличием в расплаве и бифторида бериллия. Так как имелось значительное число вариантов, отличавшихся лишь составом солей, было удобно всегда иметь описание обоих вариантов, чтобы замена одного на второй сводилась к формальной подстановке в геометрическом модуле :9 → :10 с помощью контекстной замены, доступной практически в любом текстовом редакторе.

По тем же причинам имеется материал 6 – тяжелая вода.

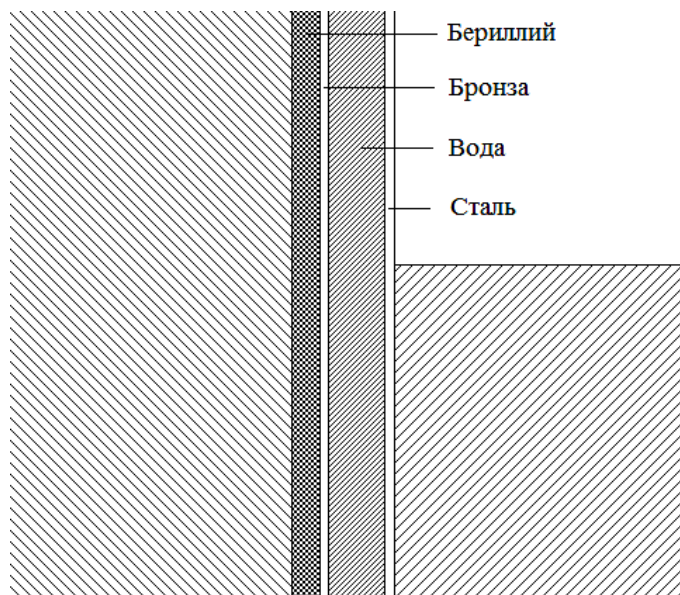


Рис 48. Материалы стенки вакуумной камеры и свинцового экрана

Точно так же рассматривались варианты, в которых центральный столб состоит из меди или бериллия. Именно поэтому в описании имеется два бериллия: материал 4 - бериллий стенки вакуумной камеры, а также материал 1 – бериллий центрального столба. В этом материале в виде комментария присутствует заготовка описания меди.

```
PIN 1 0
MATR 1 DENSWA=1.848
** BE
BE 1.0
** CU
** CU 8.4350E-02
MATR 2
** VOID 1
O 0.1E-8
MATR 3
```



```

** H2O
H 0.0641209
D 9.61958E-6
O 0.0320653
MATR 4 DENSWA=1.848
** BE
BE 1.0
MATR 5 DENSAW= 8.4350E-2
** Cu 99.1% Cr 0.8% Zr 0.1%
CU 0.991
ZR 0.001
CR 0.008
MATR 6
* D2O
D 0.0669953
O 0.0334976
MATR 7
** VOID 2
O 0.1E-8
MATR 8
** Pb
PB 3.29634E-02
MATR 9
** FLiNaK 30% ThF4
LI7 8.1585E-03
NA 1.8488E-03
K 7.3690E-03
TH32 7.5194E-03
F 4.7454E-02
PA31 1.0E-9
PA32 1.0E-9
PA33 1.0E-9
MATR 10
** FLiNaK 15% ThF4 + 5% BeF2
LI7 9.2596E-03
NA 2.2900E-03
K 8.3635E-03
TH32 3.7337e-03
BE 1.2446E-03
F 3.7337E-02
MATR 11
** Li7 screen

```

```

LI7 4.6330E-02
MATR 12 DENSWA=7.8
** fe
FE 1.0
MATR 13
** grafit
C 0.084182
END
EGRC 1.0, 1.0, 0.
FINISH PHYSIC
HEAD 1 0 40000
CONT B B B
* Main constants
EQU HC = 100 ; chamber top/bottom
EQU HA2 = 150.8 ; all top/bottom
EQU RAL = 200.8 ; global radius
EQU RCM = 75 ; radius of chamber
EQU RSO = 70 ; outer radius of source
EQU RSI = 50 ; inner radius of source
EQU HS2 = 30 ; half of source height
EQU RCT = 20 ; radius of central conductor
EQU RPL = 50 ; radius of polliadal
EQU RD = 5 ; width of detector
EQU HDE = 5 ; half height of detector
*
EQU HPLD = 50
EQU BTCL = HC+0.8
EQU HD2 = 5
*
* thickness of screen
EQU TOFS = 5
* constants of cone
EQU HC0 = 182
EQU HC1 = HC0-2*TOFS
EQU RM = RCM+1.1+70
EQU RF = 273
EQU AC = RF-RM
EQU Z1 = HC1/8
EQU Z2 = 2*Z1
EQU Z3 = 3*Z1
EQU Z4 = HC1/2
EQU RZ0 = RM

```

```

EQU RZ1 = SQRT(RF*RF-Z1*Z1)-AC
EQU RZ2 = SQRT(RF*RF-Z2*Z2)-AC
EQU RZ3 = SQRT(RF*RF-Z3*Z3)-AC
EQU RZ4 = SQRT(RF*RF-Z4*Z4)-AC
*
RCZ CNT 0,0,-HA2 2*HA2 RAL
RCZ VAC 0,0,-HC 2*HC RCM
RCZ BE 0,0,-HC-0.3 2*HC+0.6 RCM+0.3
RCZ BRON 0,0,-HC-0.4 2*HC+0.8 RCM+0.4
RCZ H2O 0,0,-HC-1.0 2*HC+2.0 RCM+1.0
RCZ STL 0,0,-HC-1.1 2*HC+2.2 RCM+1.1
RCZ CUI 0,0,-HC 2*HC RCT
RCZ CUP 0,0,-HA2 2*HA2 RPL
*
* Pb screen
RCZ U1 0,0,-BTCL-15 2*BTCL+30 RCM+1.1+TOFS
* model
RCZ BLAN0 0,0,-HC0/2 HC0 RAL
TRC TM4 0,0,-Z4 0,0,Z1 RZ4 RZ3
TRC TM3 0,0,-Z3 0,0,Z1 RZ3 RZ2
TRC TM2 0,0,-Z2 0,0,Z1 RZ2 RZ1
TRC TM1 0,0,-Z1 0,0,Z1 RZ1 RZ0
TRC TP1 0,0, 0 0,0,Z1 RZ0 RZ1
TRC TP2 0,0, Z1 0,0,Z1 RZ1 RZ2
TRC TP3 0,0, Z2 0,0,Z1 RZ2 RZ3
TRC TP4 0,0, Z3 0,0,Z1 RZ3 RZ4
RCZ RSB 0,0, -Z4-TOFS TOFS RZ4
RCZ RST 0,0, Z4 TOFS RZ4
END
*
CU1 CUI /1:1
CU2 CUP -STL /2:1
VCH VAC -CUI /7:7
WBE BE -VAC /4:4
WBR BRON -BE /5:5
WAT H2O -BRON /3:3
STL STL -H2O /12:12
* Vacuum
BLANM1 TM1 -U1 /6:9
BLANM2 TM2 -U1 /6:9
BLANM3 TM3 -U1 /6:9
BLANM4 TM4 -U1 /6:9
BLANP1 TP1 -U1 /6:9
BLANP2 TP2 -U1 /6:9

```

```

BLANP3 TP3 -U1 /6:9
BLANP3 TP4 -U1 /6:9
* Pb04 screen
PB RSB -STL /15:8
PT RST -STL /15:8
P1 U1 -STL BLAN0 /8:8
P2 CNT -CUP -STL -BLAN0 /13:13
P3 BLAN0 -TM1 -TM2 -TM3 -TM4
  -TP1 -TP2 -TP3 -TP4
  -RSB -RST /14:13
END
FINISH GEOMETRY
NPS 1
PROB 1
ANGLEN CFCS
MDIS S
EDIS D 1
EMES 14.1E6
EPRO 1.
TYPE N
RCZ 0,0,-HS2 2*HS2 RSO RSI
REPER 1
SNAM CFCS
NOBJ 0
FINISH SOURCE
RGS 1 0
BAL
NRET 2314 DOWN
PTYPE 1
TTYE 1
ENERGY 10E6 1E6 1E5 1E4 1E3 100 10 1 0
RCT 2, 3, 4, 16, 17, 18, 22, 28, 101, 102, 103, 107
ZFLU 1-15
ZRCT 1-15
END
FINISH TALLY
ISTR 1
FINISH
NAMVAR G0305
MAXSER 2400
NPRIN 0
DTZML 200
FINISH

```

Стоит отметить, что геометрические параметры источника – константы HS2, RSO, RSI – описаны среди прочих данных в разделе геометрического модуля. Это удобно, так как геометрия материальных и регистрационных зон и геометрия источника, хотя и не совпадают, но, несомненно, связаны.

В источнике для энергии выбран дискретный тип распределения с единственным значением (EDIS D 1). Это связано с тем, что нейтроны синтеза «дейтерий – тритий» всегда предполагаются с энергией 14,1 МэВ. Данное значение присутствует в предложении EMES в электрон-вольтах – стандартной единице энергии в программе MCU. Предложение EPRO должно присутствовать обязательно, хотя при одном значении энергии содержащееся в этом предложении число не играет роли.

Точно так же должно присутствовать предложение REPER, хотя при изотропном источнике (предложение MDIS S) значение его параметров роли не играет.

Рассмотрим теперь данные регистрационного модуля. Первое предложение – стандартный заголовок.

Далее следует запрос печати в файл результатов (файл с расширением FIN) информации о балансе нейтронов. В файле результатов будут присутствовать строки:

```
Abs. by col: 1.61598E+00 1.22245E-02
```

```
Leakage n.: 1.37728E-01 3.95825E-02
```

Первая строка означает оценку числа поглощенных нейтронов, вторая – число нейтронов покинувших систему, то есть утечку. Оба числа нормированы на один нейтрон источника. Не следует удивляться, что сумма этих чисел превышает единицу. В данной системе активно идут реакции  $n \rightarrow 2n$  на свинце и тории, а также  $n \rightarrow 3n$  и деления с генерацией вторичных нейтронов на тории.

Смысл предложения NRET 2314 DOWN уже был разъяснён. В данном случае при выдаче в файл результатов самый внешний перебор – это регистрационные зоны. Для каждой зоны перебираются ядерные реакции. Для реакции перебираются нуклиды, и внутри выдается таблица по энергетическим группам.

Предложение PTYPE 1 определяет, что дальнейшее относится к регистрации нейтронов, а предложение TTYPE 1 означает регистрацию по пробегам, что для задач с источником почти стандарт.

Следующее предложение определяет разбиение энергетической шкалы на группы. Границы групп, заданные в электрон-вольтах, перечисляются в убывающем порядке, номера групп по традиции идут от больших энергий к малым. Верхняя граница первой группы подразумевается как бесконечность.

В следующем предложении определён список ядерных реакций, для которых рассчитываются их скорости. Данный расчёт делался с целью определения основных ядерных реакций, приводящих к изменению нуклидного состава. Поэтому список содержит практически все реакции, происходящие при энергии нейтронов ниже 20 МэВ. Обычно при расчётах ядерных реакторов список нужных скоростей реакций намного меньше. В начале файла результатов для каждой реакции будут напечатаны скорости этой реакции на каждом нуклиде (присутствующем в рассчитываемой системе), проинтегрированные по всему объёму и всем энергиям, то есть интегральные скорости реакций по нуклидам. Ниже приведена таблица для реакции 16 ( $n \rightarrow 2n$ ). Вторая колонка содержит скорость реакции на один нейтрон источника, третья – относительное среднеквадратичное отклонение в процентах. Присутствуют все нуклиды, даже те, для которых такая реакция невозможна (например, углерод).

Reaction - 16

	Mean	StdDev
BE	2.60814E-01	2.87347E-02
O	1.32810E-12	6.80194E+01
H	0.00000E+00	9.99999E+01
D	1.56615E-06	9.88282E-03
CU	7.79415E-03	9.67979E-03
ZR	9.00102E-06	9.89652E-03
CR	3.87171E-05	9.11936E-03
PB	3.26297E-01	1.13489E-02
LI7	1.68228E-03	2.41457E-02
NA	1.26060E-04	2.47095E-02
K	1.00207E-04	2.39935E-02
TH32	4.86005E-02	2.49424E-02
F	0.00000E+00	9.99999E+01
PA31	3.62154E-09	2.44287E-02
PA32	4.96739E-09	2.43586E-02
PA33	7.02159E-09	2.41601E-02
FE	4.91210E-03	1.03047E-02
C	0.00000E+00	9.99999E+01

Предложение ZFLU 1–15 определяет, что интегралы потока надо рассчитывать для регистрационных зон с номерами с 1 по 15, в данной задаче это все регистрационные зоны. По каждой из перечисленных регистрационных зон в файле результатов будет присутствовать таблица вида

FLUX. ZONE: 6

Energy	Mean	StdDev
1.00000E+07	4.24407E+00	2.44394E-02
1.00000E+06	9.87598E+00	2.27023E-02
1.00000E+05	2.35059E+01	1.63123E-02
1.00000E+04	4.51118E+01	1.64349E-02
1.00000E+03	2.72033E+01	2.27738E-02
1.00000E+02	1.52139E+01	3.23994E-02
1.00000E+01	6.40366E+00	4.70630E-02
1.00000E+00	5.46553E+00	5.64119E-02
0.00000E+00	8.29751E+00	3.53082E-02

В ней приведены интегралы потока в регистрационной зоне 6 в каждой энергетической группе. В первой колонке находятся нижние границы групп. Вторая колонка содержит интегралы потока на один нейтрон источника, третья – относительные среднеквадратичные отклонения в процентах.

Предложение ZRCT 1–15 определяет, что интегралы скоростей реакций надо рассчитывать для регистрационных зон с номерами с 1-го по 15-й. В данной задаче это все регистрационные зоны. По каждой из перечисленных регистрационных зон, реакций и присутствующих в системе нуклидов в файле результатов будет присутствовать таблица вида

ZONE: 6, REACTION: 18, NUCLIDE: TH32

Energy	Mean	StdDev
1.00000E+07	1.09770E-02	2.41925E-02
1.00000E+06	8.11725E-03	2.61662E-02
1.00000E+05	2.81067E-05	2.69732E-02
1.00000E+04	7.37635E-07	1.64367E-02
1.00000E+03	1.23390E-07	2.27128E-02
1.00000E+02	5.06277E-08	3.40669E-02
1.00000E+01	7.11024E-08	4.87794E-02
1.00000E+00	2.10104E-07	5.66881E-02
0.00000E+00	0.00000E+00	9.99999E+01

В данном примере в таблице приводятся скорости реакции 18 (деление) в 6-й регистрационной зоне на тории. Как и при выдаче

потоков в первой колонке находятся нижние границы групп, во второй – интеграл скорости реакции на один нейтрон источника, а в третьей – относительное среднеквадратичное отклонение в процентах. В отличие от таблиц интегральных скоростей реакций, если для какого-либо нуклида реакция не происходит, то соответствующая таблица не выдается.

Предложения данных для транспортного и управляющего модулей объяснены достаточно подробно в соответствующих разделах.

В начале файла результатов всегда есть информация о таких общих характеристиках расчёта, как число используемых процессоров, число промоделированных серий и частиц, общее время расчёта и среднее время моделирования одной траектории.

В файле входных данных в разделе регистрационного модуля есть ряд параметров, управляющих содержанием файла результатов расчёта. Этот файл кроме указанных, может содержать еще ряд оценок физических величин. Для понимания их смысла и более содержательного анализа результатов расчёта требуется понимание физических основ теории переноса частиц, кроме того – знание особенностей работы конкретных физических установок.



## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Norman E. Holden A SHORT HISTORY OF CSISRS. BNL-75288-2005-IR, 2005

2. Абагян Л.П., Базазянц Н.О., Николаев М.Н., Цибуля А.М. Групповые константы для расчета реакторов и защиты. М.: Энергоиздат, 1981, – 233 с.

3. Николаев М. Н. РОСФОНД - Российская национальная библиотека оцененных нейтронных данных // В мире науки. 2006. № 9. С.78–81.

4. International Handbook of Evaluation Criticality Safety Benchmark Experiments. NEA/NSC/DOC (95)03, September 2010 Edition.

5. International Handbook of Evaluated Reactor Physics Benchmark Experiments. NEA/NSC/DOC(2006)1, March 2009 Edition.

6. Ермаков С.М., Михайлов Г.А. Курс статистического моделирования. М.: Наука, 1976. – 320 с.

7. Михайлов Г.А. Оптимизация весовых методов Монте-Карло. М.: Наука, 1987. – 241 с.

8. MacFarlane R.E. and Muir D.W. The NJOY Nuclear Data Processing System. LA-12740, LANL, 1994.

9. Абагян Л.П., Базазянц Н.О., Бондаренко И.И., Николаев М.Н. Групповые константы для расчета ядерных реакторов. М.: Атомиздат, 1964. – 139 с.

Более подробно теоретические вопросы рассмотрены в следующих работах:

1. Соболев И.М. Численные методы Монте-Карло. М.: Наука, 1973. – 312 с.

2. Спанье Дж., Гельбард Э. Метод Монте-Карло и задачи переноса нейтронов. М.: Атомиздат, 1972. – 272 с.

*Михаил Исаевич Гуревич, Денис Александрович Шкаровский*

Расчет переноса нейтронов методом Монте-Карло  
по программе MCSU

Учебное пособие

Редактор Е.Г.Станкевич

Подписано в печать 15.11.2012. Формат 60×84 1/16.

Печ.л. 10,75. Уч.-изд.л. 10,75. Тираж 92 экз.

Изд. № 006-1. Заказ 66.

Национальный исследовательский ядерный университет «МИФИ»  
115409, Москва, Каширское шоссе, 31  
ООО «Полиграфический комплекс «Курчатовский»  
144000, Московская область, г.Электросталь, ул.Красная, д.42