

Описание применения и инструкция для пользователей программы  
«MCU-PTR с банком данных MDBPT50»

# СОДЕРЖАНИЕ

Введение.....	5
1 Состав программы.....	6
2 Постановка задачи.....	8
2.1 Общие сведения.....	8
2.2 Граничные условия.....	8
2.3 Описание геометрии.....	9
2.4 Моделирование взаимодействий частиц с веществом.....	10
2.4.1 Константное обеспечение.....	10
2.4.2 Моделирование столкновений.....	10
2.5 Моделирование траекторий.....	13
2.6 Накапливаемый источник.....	13
3 Решение однородной задачи.....	14
3.1 Метод поколений.....	14
3.2 Задачи об асимптотической решётке.....	16
3.3 Расчёт эффективной доли запаздывающих нейтронов.....	17
3.4 Метод суммарного изотопа.....	18
4 Вычисление функционалов потока.....	18
4.1 Общие положения.....	18
4.2 Классификация функционалов.....	19
4.2.1 Глобальные функционалы.....	20
4.2.2 Функционалы, вычисляемые для регистрационных областей.....	21
4.3 Описание выдачи на печать модуля регистрации.....	23
4.4 Расчет с использованием активационных сечений.....	23
4.5 Подмодуль USER.....	23
5 Общая схема расчёта.....	23
5.1 Шаг INPUT.....	23
5.2 Шаг CALCULATION.....	24
5.3 Шаг OUTPUT.....	24
5.4 Шаг BURNUP.....	24
5.5 Вызов системных команд.....	25
5.6 Изменение параметров реактора в процессе расчёта выгорания.....	25
6 Режим многопроцессорных вычислений.....	26
7 Схема задания исходных данных.....	27
7.1 Программные средства стандартного ввода.....	28
7.2 Программные средства бесформатного ввода.....	31
8 Исходные данные для физического модуля.....	32
8.1 Заголовок фрагмента исходных данных для физического модуля.....	33
8.2 Описание материалов.....	33
8.3 Изменение значений по умолчанию, заданных в файле DEFAULT.PHY.....	36
8.4 Описание температуры системы.....	37
8.5 Описание суммарного изотопа.....	37
8.6 Управляющие параметры физического модуля.....	38
8.7 Окончание ввода данных для физического модуля.....	40
8.8 Коррекция данных с помощью файлов PDC.....	40
8.9 Примеры задания исходных данных к физическому модулю.....	40
9 Исходные данные для геометрического модуля.....	42
9.1 Введение.....	42
9.1.1 Основные понятия.....	43
9.1.2 Форматы и константы.....	44
9.2 Описание простой системы.....	44
9.2.1 Заголовок.....	44

9.2.2 Секция граничных условий .....	44
9.2.3 Секция тел .....	47
9.2.4 Секция зон .....	54
9.2.5 Таблица объёмов регистрационных зон .....	58
9.2.6 Примеры описания очень простых систем .....	58
9.3 Описание сложных систем .....	59
9.3.1 Сети (основные понятия) .....	60
9.3.2 Описание прототипа ячейки сети .....	63
9.3.3 Описание сетей основной геометрии .....	64
9.3.4 Решётки (основные понятия) .....	67
9.3.5 Описание прототипа элемента решётки .....	69
9.3.6 Описание решётки .....	71
9.4 Дополнительные возможности операции TRANSF .....	77
9.4.1 Описание конкретного движения .....	77
9.4.2 Описание движения тела .....	79
9.4.3 Преобразования в решетке типа GLTL .....	79
9.4.4 Замечания по изменению типов тел .....	81
9.5 Регистрационные возможности геометрического модуля .....	81
10 Исходные данные для модуля источников .....	82
10.1 Простой источник .....	82
10.1.1 Простой точечный источник .....	82
10.1.2 Простой распределённый источник .....	83
10.2 Сложный источник .....	86
10.2.1 Заголовок .....	87
10.2.2 Веса примитивных источников .....	87
10.2.3 Секция описания спектра .....	88
10.2.4 Секция времён рождения .....	92
10.2.5 Секции примитивных источников .....	92
10.2.6 Дополнительные данные для RCZD контейнера .....	97
10.2.7 Ввод данных для регистрационных объектов .....	97
10.2.8 Описание элементов решёток .....	98
10.2.9 Описание решётки .....	98
10.2.10 Примеры данных для сложного источника .....	100
10.3 Накапливаемый поверхностный источник .....	103
10.3.1 Общие характеристики поверхностного источника .....	103
10.3.2 Задание поверхностей накопления и поглощения для накапливаемого источника .....	105
10.3.3 Ввод данных для поверхностного источника .....	105
11 Исходные данные модуля регистрации .....	107
11.1 Заголовок фрагмента исходных данных модуля регистрации .....	108
11.2 Раздел общих параметров .....	108
11.3 Раздел параметров регистрации типа N .....	109
11.4 Окончание ввода данных для модуля регистрации .....	112
12 Исходные данные модуля регистрации для выгорания .....	113
13 Исходные данные для алгоритма моделирования .....	114
14 Исходные данные для управления шагом расчёта .....	115
14.1 Управление неаналоговым моделированием .....	116
14.1.1 Расщепление/рулетка .....	116
14.1.2 Весовое окно по энергии и регистрационным объектам .....	118
14.1.3 Весовое окно по энергии дополнительной геометрии и углу .....	119
15 Исходные данные для модуля выгорания .....	121
15.1 Исходные данные опции STEP .....	122
15.2 Исходные данные опции FINAL .....	127
15.3 Исходные данные опции DELAY .....	127

15.4 Исходные данные опций FINTAB и DELTAB .....	128
15.5 Исходные данные опции FINDEN .....	131
15.6 Исходные данные опции SOURCE .....	132
15.7 Примечание .....	132
16 Описание директорий программы .....	132
17 Установка программы .....	132
18 Генерация программы .....	133
19 Запуск программы на счёт .....	133
20 Тестирование.....	134
21 Msc office.....	134
22 Условия применения .....	135
22.1 На Windows кластере .....	135
22.2 В многопроцессорном режиме на различных компьютерах в сети Windows с MPICH2	137
22.3 На Linux кластере .....	139
22.3.1 Пересылка банка ядерных данных программы на кластер .....	139
22.3.2 Пересылка фортранного текста рабочей версии программы на кластер .....	139
22.3.3 Трансляция текста рабочей версии на кластере .....	139
22.3.4 Запуск задачи на счёт в многопроцессорном режиме.....	140
Заключение.....	141
Обозначения и сокращения .....	142
Список использованных источников.....	143

## ВВЕДЕНИЕ

Программа «MCU-PTR с банком данных MDBPT50» (далее – MCU-PTR) разработана в рамках проекта MCU [1, 2] и является развитием программ MCU-REA/1 [3] и MCU-REA/2 [3], аттестованных Ростехнадзором для расчётов нейтронно-физических характеристик реакторов различного типа.

Программа предназначена для решения аналоговыми и неаналоговыми методами Монте-Карло неоднородных уравнений переноса нейтронов и фотонов. Для нейтронов программа позволяет решать и однородное уравнение (задачи о критичности систем, размножающих нейтроны). Математически это означает, что для рассматриваемой системы решается кинетическое уравнение с заданными граничными условиями, описывающее распределение в ней потока частиц.

Обеспечена возможность расчётного предсказания изотопного состава материалов реактора и его размножающих свойств в зависимости от длительности кампании.

В документе приводятся описание применения и инструкция по работе с программой. Информации, приведённой в нем, достаточно для получения общего представления о программе, схеме её работы, задания исходных данных расчётного варианта и анализа результатов расчёта.

# 1 СОСТАВ ПРОГРАММЫ

Программа имеет модульную структуру и komponуется из следующих модулей пакета MCU-5. В таблице 1.1 представлено краткое описание модулей MCU-PTR.

Таблица 1.1 – Краткое описание модулей MCU-PTR

Название модуля/подмодуля	Обозначение	Функция	Примечание
Управляющий модуль	C	Управление расчетом	Обеспечивает возможность автоматического изменения любых параметров в процессе расчёта выгорания.
Транспортный модуль	TRJ	Моделирование траекторий частиц в системе	-
Составной физический модуль, СОФИЗМ	SFM	Розыгрыш взаимодействия частиц с веществом по информации из банка данных	Состоит из подмодулей
PIN	PIN	Ввод данных для физического модуля	Подмодуль составного физического модуля
PSI	PSI	Реализует метод суммарного изотопа	Подмодуль составного физического модуля
FARION	FAR	Моделирование взаимодействия нейтронов с веществом в области быстрых нейтронов (от 20 до 0.1 МэВ)	Подмодуль составного физического модуля
ФИМБРОЭН	FMB	Моделирование взаимодействия нейтронов с веществом в эпитепловой области	Подмодуль составного физического модуля
ФИМТОЭН	FMT	Моделирование взаимодействия нейтронов с веществом в области термализации нейтронов	Непрерывное слежение за энергией нейтрона с использованием поточечного представления сечений. Подмодуль составного физического модуля
СТЕНЬ	STN	Автоматическая подготовка данных библиотеки VESTA	Подмодуль составного физического модуля

Название модуля/подмодуля	Обозначение	Функция	Примечание
МОФИТТГ	MOF	Моделирование взаимодействия нейтронов с веществом в области термализации нейтронов	Многогрупповое транспортное приближение. Подмодуль составного физического модуля
ТЕРМАК	TRM	Автоматическая подготовка данных библиотеки TERCON	Подмодуль составного физического модуля
MULTIC	FML	Расчет сечений нейтронных реакций по мультигрупповым параметрам	Подмодуль составного физического модуля
RAPAN	RAP	Расчет сечений нейтронных реакций по аналитическим выражениям формализмов	Подмодуль составного физического модуля
POIRES	POR	Расчет сечений нейтронных реакций по данным в поточечном представлении	Подмодуль составного физического модуля
ФОТОН	PNS	Моделирование генерации фотонов в (n, $\gamma$ ) реакциях	Подмодуль составного физического модуля
	PNT	Моделирование взаимодействия фотонов с веществом	Подмодуль составного физического модуля
Геометрический модуль	NCG	Моделирование прямолинейных участков траекторий между столкновениями	-
Модуль источников	SRC	Моделирование фазовых координат частиц источника или нейтронов нулевого поколения при решении задач на критичность	-
Модуль регистрации	RGS	Расчёт широкого набора функционалов нейтронного потока	-

Название модуля/подмодуля	Обозначение	Функция	Примечание
Дозиметрический модуль	DOS	Расчет функционалов, связанных с активационными сечениями и керма-факторами	Подмодуль модуля регистрации
User	URGS	Пользовательская регистрация	Подмодуль модуля регистрации
Модуль регистрации для выгорания	BRG	Расчёт функционалов нейтронного потока для модуля выгорания	-
Модуль оборудования	ENV	Содержит программы, которые могут зависеть от типа компьютера и операционной системы	-
Модуль выгорания	BUR	Расчёт изменения изотопного состава материалов реактора в процессе его эксплуатации	-

## 2 ПОСТАНОВКА ЗАДАЧИ

### 2.1 ОБЩИЕ СВЕДЕНИЯ

Программа предназначена для решения аналоговыми и неаналоговыми методами (в том числе и матричным методом) Монте-Карло неоднородных уравнений переноса нейтронов и фотонов. Для нейтронов программа позволяет решать и однородное уравнение (задачи о критичности систем, размножающих нейтроны). Математически это означает, что для рассматриваемой системы решается кинетическое уравнение с заданными граничными условиями, описывающее распределение в ней потока частиц.

Системой называется любая конечная область пространства с заданными физическими свойствами.

Система состоит из геометрических зон, ограниченных плоскостями или поверхностями второго порядка, параметры которых задаются пользователем. Каждая зона заполнена однородным материалом. Предполагается, что система состоит из конечного числа геометрических зон и материалов.

Каждый материал определяется:

- температурой материала;
- списком нуклидов, содержащихся в нем;
- ядерной концентрацией нуклидов.

### 2.2 ГРАНИЧНЫЕ УСЛОВИЯ

Все граничные условия можно разделить на два класса:

- условия, отображающие симметрии системы,



– условия, отображающие физические свойства системы.

Детализация описания возможных граничных условий приведена в описании геометрического модуля. Здесь мы ограничимся их перечислением.

Симметрия - это движения трёхмерного пространства, по отношению к которым описание системы инвариантно. В программе можно задать следующие симметрии:

- зеркальная симметрия;
- поворотная симметрия;
- сдвиговая (трансляционная) симметрия.

Можно использовать следующие физические граничные условия:

- чёрная поглощающая поверхность;
- белое отражение;
- граничные условия, соответствующие задаче с утечкой, заданной баклингом.

Кроме условий утечки и отражения допускается их комбинация, когда на внешней поверхности системы задаётся некоторое положительное альbedo  $\alpha < 1$ . При пересечении траекторией нейтрона поверхности с вероятностью  $\alpha$  происходит отражение, а с вероятностью  $(1-\alpha)$  – поглощение.

## 2.3 ОПИСАНИЕ ГЕОМЕТРИИ

Программа позволяет моделировать системы, состоящие из объёмных элементов практически произвольной формы.

Геометрический модуль универсального типа NCG позволяет моделировать трёхмерные системы с произвольной геометрией, используя комбинаторный подход, основанный на описании сложных пространственных форм (зоны) комбинациями простых тел или поверхностей с помощью теоретико-множественных операций пересечения, дополнения и объединения.

Имеется некоторый набор типов тел-примитивов. Для каждого типа тела имеется система параметров, полностью описывающая форму конкретного тела и его положение в пространстве. Каждому типу тела соответствует программа, определяющая точки пересечения любой прямой с любым конкретным телом этого типа. Для обеспечения эффективности счёта все тела ограничены плоскостями или поверхностями второго порядка.

Возможно задание решёток, получаемых размножением некоторых исходных элементов, заданных с помощью комбинаторики.

В модуле NCG помимо представления повторяющихся геометрических объектов с помощью решёток, введены так называемые сети. Элемент сети не размножается в процессе ввода, расчёт траектории внутри него производится в локальной системе координат. Таким образом, иерархия геометрических объектов частично сохраняется и на фазе счёта. Это даёт большую экономию памяти, но требует при расчёте использовать локальные системы координат. В модуле NCG сеть всегда состоит из элементов с одинаковой внешней формой и плотно примыкающих друг к другу. Преобразования координат сводятся к сдвигам и не замедляют расчёт. Это радикально упрощает задание геометрии и экономит оперативную память компьютера.

При описании зон методом комбинаторной геометрии все их границы состоят из кусков плоскостей или квадратичных поверхностей, поэтому при наличии деталей с более сложными граничными поверхностями их необходимо аппроксимировать очень большим числом зон.

*Специальный алгоритм* позволяет учитывать эффекты двойной гетерогенности, когда тепловыделяющие элементы содержат десятки тысяч микротвэлов.

## 2.4 МОДЕЛИРОВАНИЕ ВЗАИМОДЕЙСТВИЙ ЧАСТИЦ С ВЕЩЕСТВОМ

### 2.4.1 Константное обеспечение

Константное обеспечение программы составляет банк данных MDBPT50. В таблице 2.4.1.1 представлены библиотеки, составляющие банк данных MDBPT50.

Таблица 2.4.1.1 – Библиотеки, составляющие банк данных MDBPT50

Название библиотеки	Описание
ACE/MCU	Библиотека сечений взаимодействия нейтронов с ядрами в эпитепловой области энергий в поточечном представлении, полученная из файлов ENDF/B-VII.0 и других источников
BNAB/MCU	Расширенная и модифицированная версия 26-групповой системы констант БНАБ-93
LIPAR	Резонансные параметры нуклидов в области разрешённых резонансов
MULTIC	301-групповая библиотека, содержащая, в том числе, данные по температурной зависимости подгрупповых параметров нуклидов в области неразрешённых резонансов
KORT	Библиотека поточечно представленных нейтронно-физических констант в области низких энергий (верхняя граница - 5 эВ)
TEPCON	40-групповые сечения для области термализации с границей 1 эВ
VESTA	Библиотека для моделирования столкновений нейтронов с ядрами замедлителей с учётом непрерывного изменения энергии нейтронов в области термализации; представлена в форме вероятностных таблиц, полученных из законов рассеяния $S(\alpha, \beta)$
BOFS	Библиотека обобщённых фононных спектров замедлителей
DOSIM	Библиотека активационных сечений в поточечном представлении
ABBNL	66 групповые сечения; используется для получения сечений «суммарного изотопа»
PHOTONS	Библиотека многогрупповых сечений генерации фотонов при взаимодействии нейтронов с веществом на основе данных библиотек DLC-41/VITAMIN-C и DLC-184/VITAMIN-B6
PHOTONT	Многогрупповые сечения взаимодействия фотонов с веществом на основе данных библиотек DLC-41/VITAMIN-C и DLC-184/VITAMIN-B6
BURN5	Содержит информацию для задач по выгоранию: периоды полураспада ядер, выходы осколков деления, цепочки радиоактивных превращений и т.д.

### 2.4.2 Моделирование столкновений

Физический модуль программы позволяет проводить моделирование столкновений частиц с веществом на базе перечисленных систем констант, при этом допускается использование различных моделей взаимодействия.

Программа позволяет учитывать эффекты непрерывного изменения энергии частицы при столкновениях, а также как непрерывную, так и ступенчатую зависимость сечений от энергии.

При моделировании переноса нейтронов учитываются следующие эффекты.

При генерации нейтронов деления допускается использование спектра деления мгновенных и запаздывающих нейтронов.

В быстрой энергетической области учитывается анизотропия упругого рассеяния в системе центра масс, имеется возможность проводить моделирование неупругих столкновений с учётом законов, содержащихся в файлах оценённых ядерных данных.

В области неразрешённых резонансов сечения вычисляются по подгрупповым параметрам или с использованием  $f$ -факторов Бондаренко, в обоих случаях с учётом температурной зависимости используемых параметров.

В области разрешённых резонансов допускается как подгрупповое, так и поточечное описание сечений. Сечения наиболее важных нуклидов описываются «бесконечным» числом точек, так как при моделировании в каждой энергетической точке они вычисляются по резонансным параметрам. Такая схема позволяет проводить расчёты непосредственно с использованием данных по резонансным параметрам без предварительной подготовки таблиц сечений и оценивать температурные эффекты через аналитические зависимости сечений от температуры.

Моделирование столкновений в области термализации проводится либо в многогрупповом транспортном приближении, либо по модели непрерывного изменения энергии с учётом корреляций между изменением энергии и угла при рассеянии. В обоих случаях учитываются химические связи, тепловое движение ядер и когерентные эффекты для упругого рассеяния.

Для нейтронов учитываются особенности взаимодействия в разных энергетических областях путём использования набора физических подмодулей. Составной физической модуль (СОФИЗМ) включает в себя набор физических подмодулей, которые используют различные библиотеки констант и реализуют различные модели описания взаимодействия нейтронов с веществом. Составной физической модуль позволяет объединить различные частные модели для разных областей энергий в одну, описывающую взаимодействие нейтронов с веществом во всей области энергий.

В компоновку программы входят следующие подмодули:

– FARION, предназначенный для моделирования взаимодействия нейтронов с веществом в области быстрых нейтронов (от 20 до 0.1 МэВ) по данным библиотеки ACE/MSU;

– ФИМБРОЭН, предназначенный для моделирования взаимодействия нейтронов с веществом в эпитепловой области;

– ФИМТОЭН – предназначенный для моделирования взаимодействия нейтронов с веществом в области термализации нейтронов, в котором реализовано непрерывное слежение за энергией нейтрона с использованием поточечного представления сечений;

– МОФИТТГ - альтернативный ФИМТОЭН подмодуль, предназначенный для моделирования взаимодействия нейтронов с веществом в области термализации нейтронов, в котором реализовано многогрупповое транспортное приближение.

Для каждого из подмодулей определяются свои наиболее характерные понятия, позволяющие на языке ввода исходных данных задать модели взаимодействия нейтронов с ядрами среды, используемые приближения и другую информацию.

В ФИМБРОЭН вводятся понятия резонансного, подгруппового и группового нуклидов.

*Резонансный* - это такой нуклид, для которого сечения захвата, деления и упругого рассеяния в области разрешённых резонансов в каждой энергетической точке рассчитываются всякий раз, когда эти сечения требуются, то есть в поточечном представлении. В области неразрешённых резонансов сечения такого нуклида вычисляются по подгрупповым параметрам библиотек BNAB/MSU (непосредственно подмодулем

ФИМБРОЭН) или MULTIC (с помощью обращения к подмодулю FIMUL), а в случае их отсутствия в библиотеках – с учётом f-факторов. Энергия, начиная с которой в ФИМБРОЭН происходит поточечный расчёт сечений, совпадает с нижней граничной энергией той из групп библиотеки VNAB/MCU, в которую попадает определённая в библиотеке LIPAR граница ERR области разрешённых резонансов данного нуклида. Эту границу пользователь может изменить (в сторону уменьшения) в исходных данных. Нуклид объявляется резонансным путём задания в исходных данных параметра BLOCK равным 3, 4, 8 или 9.

Имеются следующие возможности:

– BLOCK = 3 – в области неразрешённых резонансов сечения вычисляются подмодулем FIMUL с использованием констант MULTIC, а в области разрешённых резонансов расчёт производится по резонансным параметрам из библиотеки LIPAR по аналитическим выражениям формализмов Брейта-Вигнера или Адлер-Адлера в  $\psi$ - $\chi$  приближении с использованием подмодуля RAPAN;

– BLOCK = 4 – в области неразрешённых резонансов сечения вычисляются с использованием констант VNAB/MCU, а в области разрешённых резонансов расчёт производится по резонансным параметрам из библиотеки LIPAR по аналитическим выражениям формализмов Брейта-Вигнера или Адлер-Адлера в  $\psi$ - $\chi$  приближении с использованием подмодуля RAPAN;

– BLOCK = 8 - в области низких энергий расчёт проводится подмодулем POIRES с использованием констант библиотеки KORT, а для энергий нейтронов, больших верхней границы библиотеки KORT, используются константы VNAB/MCU;

– BLOCK = 9 – в области неразрешённых резонансов сечения вычисляются с использованием констант VNAB/MCU, а в области разрешённых резонансов расчёт производится по резонансным параметрам из библиотеки LIPAR по аналитическим выражениям формализмов Рейха-Мура с использованием подмодуля RAPAN.

*Подгрупповой* - это такой нуклид, для описания резонансной структуры сечений которого используется метод подгрупп во всей области резонансных энергий с использованием данных библиотек VNAB/MCU (непосредственно подмодулем ФИМБРОЭН) или MULTIC (с помощью обращения к подмодулю FIMUL). Если в библиотеках подгрупповых параметров нет, то учёт блокировки сечений осуществляется при помощи f-факторов. Нуклид объявляется подгрупповым путём задания в исходных данных параметра

BLOCK = 2 – используются подгрупповые параметры библиотеки VNAB/MCU;

BLOCK = 3 и ERR = EUR – используются подгрупповые параметры библиотеки MULTIC, здесь EUR – верхняя граница области неразрешённых резонансов.

*Групповой* - это такой нуклид, у которого резонансная самоэкранировка сечений либо не учитывается, либо учитывается при помощи f-факторов, которые зависят от температуры и сечения разбавления. Нуклид объявляется групповым заданием параметра BLOCK = 1, когда блокировка учитывается при помощи f-факторов, или BLOCK = 0, когда не учитывается.

Точность расчётов резонансных эффектов увеличивается при переходе от f-факторов к подгрупповому представлению сечений и далее к поточечному. При этом соответственно увеличивается время счёта. Это следует иметь в виду пользователю при выборе способа учёта резонансной структуры сечений каждого изотопа в каждом конкретном варианте.

Для тепловых подмодулей МОФИТТГ и ФИМТОЭН в исходных данных необходимо указать модель, по которой будет происходить розыгрыш рассеяния нейтронов. Под словом модель здесь имеется в виду не только учёт или неучёт химических связей атомов, но и выбранную оценку спектра собственных частот атома при учёте химических связей, а также учёт или неучёт когерентных эффектов.

Пользователь в исходных данных определяет модель термализации и допуск по температуре, с которым разрешается выбирать необходимую информацию из библиотек TEPSON (для МОФИТТГ) и VESTA (для ФИМТОЭН). Из библиотек извлекаются при заданных температурах сечения поглощения, если они не подчиняются закону "1/v", и

сечения рассеяния, если они рассчитываются не по газовой модели и не по модели тяжёлого газа (см. ниже). Признак, подчиняются ли сечения поглощения закону " $1/v$ ", записан в текстовом файле FGI.TPC, расположенном в банке данных программы.

В программе реализована возможность автоматической подготовки данных библиотек TEPCON (с помощью обращения к подмодулю TEPMAK по информации из библиотек BOFS и LIPAR) и VESTA (с помощью обращения к подмодулю СТЕНЬ по информации из библиотеки BOFS) для любой температуры. Эта возможность реализована следующим образом. При отсутствии в библиотеках TEPCON или VESTA в пределах заданного допуска информации по требуемым температурам для того или иного изотопа программа автоматически подготавливает необходимую информацию. Эта информация сохраняется в файлах в разделе TMPDAT используемого при расчете банка данных. При повторной необходимости в этой информации (например, в следующем расчете) программа получит ее из ранее подготовленных файлов, хранящихся в разделе TMPDAT. Иными словами, при отсутствии информации в основной библиотеке (TEPCON или VESTA) программа пытается получить информацию из раздела TMPDAT, а в случае отсутствия ее и в этом разделе автоматически осуществляет ее подготовку, сохранение для дальнейшего использования и ввод.

Моделирование генерации фотонов в  $(n,\gamma)$  реакциях осуществляется подмодулем ФОТОН (по данным библиотеки PHOTONS). Моделирование взаимодействия фотонов с веществом осуществляется подмодулями ФОТОН (по данным библиотеки PHOTONT). При моделировании переноса фотонов в нём учитываются следующие эффекты:

- рождение гамма-квантом (фотоном) в поле атомного ядра пары электрон-позитрон;
- рассеяние гамма-кванта на электроны среды, который (при энергии гамма-кванта больше 10 кэВ) можно считать свободным (комптоновское рассеяние);
- поглощение фотона одним из электронов атома с последующим вылетом этого электрона и испусканием оставшимися электронами серии мягких фотонов в результате перехода на освобождающиеся нижние оболочки (фотоэлектрическое поглощение, фотоэффект).

## 2.5 МОДЕЛИРОВАНИЕ ТРАЕКТОРИЙ

Используемый в программе метод Монте-Карло позволяет моделировать историю каждой частицы в рассматриваемой системе в такой последовательности:

- рождение частицы;
- определение прямолинейных участков траекторий частицы от точки рождения до точки столкновения, между двумя последовательными точками столкновения, от последней точки столкновения до поглощения или вылета частицы из системы;
- моделирование взаимодействия частицы с веществом в точках столкновений с определением типа произошедшей реакции;
- определением параметров вторичных частиц, если они возникают в результате определённого взаимодействия.

Моделирование проводится на основе точного описания геометрической структуры и материального состава рассматриваемых систем и использования законов взаимодействия частицы с веществом, параметры которых содержатся в библиотеках констант.

Важным для моделирования истории частицы весовыми методами Монте-Карло является понятие её *веса*  $W$ . При аналоговом моделировании траектории частицы методом Монте-Карло её вес  $W = 1$ .

## 2.6 НАКАПЛИВАЕМЫЙ ИСТОЧНИК

Допускается решение задачи в несколько этапов – исторически этот метод получил в рамках проекта MSU наименование АЛИГР. Каждый этап есть моделирование переноса методом Монте-Карло с помощью программы MSU. Первый этап является решением

однородной задачи или неоднородной задачи, использующей комбинаторный модуль источника. Все последующие этапы являются решениями неоднородной задачи. На каждом этапе описание геометрии практически идентично.

На каждом этапе производится запись фазовых координат частиц, проходящих через так называемые поверхности накопления в положительном направлении. Эти точки фазового пространства образуют банк частиц и на последующем этапе интерпретируются как источник, а поверхность накопления как внешняя граница системы с условием поглощения. При использовании накопленных частиц их можно расщеплять по тем или иным признакам. Это обеспечивает один из способов неаналогового моделирования. Таким образом, используются два типа поверхностей – поверхность накопления и поверхность поглощения. На первом этапе должна быть определена только поверхность накопления, на последнем этапе используется лишь поверхность поглощения.

Данные для работы накапливаемого источника частично располагаются вместе с описанием геометрии (определение поверхностей накопления и поглощения), а частично с данными источника (порядок расщепления накопленных частиц, т.е. как они формируют источник).

### 3 РЕШЕНИЕ ОДНОРОДНОЙ ЗАДАЧИ

Рассматривается задача о вычислении главного собственного значения  $k_{eff}$  и нормированных скоростей реакций нейтронов, определённых на главной собственной функции  $Q(x)$  уравнения для плотности вторичных нейтронов деления в условно критическом реакторе.

В операторном виде это уравнение можно записать следующим образом.

$$k_{eff}Q = HQ, \quad (3.1)$$

где  $Q = Q(x)$ ,  
 $x = (r, v)$  – координаты точки фазового пространства координат и скоростей,  
 $H$  – интегральный оператор.

Скорости реакций формально можно определить, как дробно-линейные функционалы вида

$$R = \frac{(A, Q)}{(e, Q)} \quad (3.2)$$

где  $A$  – вектор, соответствующий реакции данного типа,  
 $e$  – единичный вектор.

#### 3.1 МЕТОД ПОКОЛЕНИЙ

Решение задач о критичности систем проводится методом поколений.

В каждом поколении моделируются траектории пакета частиц, состоящего из  $NTOT$  нейтронов, где  $NTOT$  – заданное пользователем постоянное число.

Начальные фазовые координаты  $(r, v)$  нейтронов первого поколения задаются пользователем в исходных данных для модуля источников.

Моделирование истории каждого нейтрона в поколении осуществляется следующим образом.

Определяется случайная точка фазового пространства  $(r, v)$ , в которой начинается траектория нейтрона, Далее определяется случайная последовательность точек столкновений нейтрона с ядрами среды от точки его рождения до точки поглощения или вылета из системы. Фазовые координаты точек столкновения, поглощения или вылета нейтронов из системы определяются подпрограммами геометрического и физического модулей, обращения к которым содержатся в транспортном модуле. Для расчёта скоростей реакций и других функционалов нейтронного потока транспортный модуль вызывает подпрограммы

модуля регистрации, входными параметрами которых являются фазовые координаты вышеупомянутых точек и другие величины, необходимые для работы этого модуля.

Для каждой точки поглощения определяется случайное, не обязательно целое число («вес»)  $w_n$  вторичных нейтронов деления и их скорость  $v_n$ .

После окончания моделирования траекторий всех нейтронов очередного поколения, веса нейтронов нового поколения, рождённых в точке с номером  $n$ , подвергаются перенормировке. Новые веса  $w'_n$  определяются так, чтобы полный суммарный вес нейтронов нового поколения был равен заданному числу  $NTOT$ :

$$w'_n = NTOT \frac{w_n}{\sum w_n} \quad (3.3)$$

Точки поглощения нейтронов предыдущего поколения рассматриваются как точки рождения  $NTOT$  нейтронов нового поколения с единичным весом каждый. Новые нейтроны выбираются случайным образом так, чтобы математическое ожидание нейтронов, рождённых в точке  $n$ , было бы равно  $w'_n$ . Для этой цели используется метод систематической выборки, когда в каждой точке рождается либо  $[w'_n]$ , либо  $[w'_n]+1$  нейтронов ( $[w'_n]$  означает целую часть числа  $w'_n$ ), но так, что суммарное число нейтронов нового поколения равно  $NTOT$ .

Число моделируемых поколений может быть сколь угодно велико и задаётся пользователем. В любой момент счёт может быть прерван, а затем продолжен. Результаты, полученные к моменту прерывания, доступны для пользователя.

Процесс моделирования по поколениям эквивалентен итерационному решению задачи об условно критической системе, когда источники нейтронов для каждой итерации нормируются на число  $NTOT$ .

Расчёты статистическим «методом поколений» с фиксированным числом  $NTOT$  нейтронов в поколении являются приближёнными и *всегда* содержат систематическую ошибку в оценке функционалов, убывающую пропорционально  $1/NTOT$ . В программе по умолчанию принято значение  $NTOT = 200$ . При расчётах систем больших, чем ячейка рекомендуется задавать  $NTOT$  не менее 1000.

Другой источник систематических ошибок связан с тем, что все функционалы рассчитываются в предположении асимптотической несмещённости их оценок, которое является корректным только при бесконечном числе промоделированных поколений. На практике всегда приходится моделировать конечное число поколений. Поэтому результат расчётов зависит от выбранного начального приближения, влияние которого уменьшается обратно пропорционально промоделированному числу поколений.

Расчёт статистических ошибок функционалов в программе проводится в приближении, в котором каждые последовательные  $NBAT$  поколений частиц, называемые сериями, считаются независимыми. Точность этого приближения увеличивается с увеличением значения  $NBAT$ , которое задаётся пользователем.

По умолчанию в программе принято значение  $NBAT = 3$ .

При решении задач разного типа рекомендуется использовать следующие значения  $NBAT$ :

- от 5 до 10 при расчёте ячейки,
- от 10 до 50 при расчёте одной или нескольких тепловыделяющих сборок,
- от 100 до 200 и более при расчёте модели энергетического реактора в целом.

Однако приведённые рекомендации не гарантируют получения правильных результатов при решении сложных задач, таких как расчёт полномасштабных энергетических реакторов или слабосвязанных систем. При решении таких задач следует критически относиться к печатаемой программой оценке статистической погрешности, которая зависит от заданного пользователем значения  $NBAT$ .

### 3.2 ЗАДАЧИ ОБ АСИМПТОТИЧЕСКОЙ РЕШЁТКЕ

Наибольший практический интерес представляют две задачи об асимптотической гетерогенной решётке.

В первой рассматривается однородная и бесконечная вдоль оси OZ решётка, имеющая произвольную геометрию в плоскости XOY. Предполагается, что утечка вдоль оси OZ задаётся аксиальной составляющей  $B_z$  вектора баклинга. Значение  $B_z$  может быть как действительным, так и мнимым, что соответствует моделированию экспоненциальных и критических экспериментов соответственно.

Во второй рассматривается элементарная ячейка бесконечной периодической гетерогенной решётки с утечкой, заданной значением  $\mathbf{B}$  вектора баклинга.

В обоих случаях предполагается, что поток нейтронов может быть описан асимптотическим решением уравнения переноса. По определению это решение записывается в форме:

$$\Phi(\mathbf{r}, \mathbf{v}) = \varphi(\mathbf{r}, \mathbf{v}) \exp(i\mathbf{B}\mathbf{r}) \quad (3.4)$$

где  $\varphi(\mathbf{r}, \mathbf{v})$  – периодическая функция с периодом, равным периоду решётки;  
 $\mathbf{r}$  – радиус–вектор точки, в которой находится нейтрон;  
 $\mathbf{v}$  – вектор скорости нейтрона.

Пусть  $\mathbf{r}_s$  есть произвольная точка на границе ячейки и  $\mathbf{r}_s + \mathbf{l}(\mathbf{r}_s)$  есть точка на противоположной границе. Вектор  $\mathbf{l}(\mathbf{r}_s)$  определён однозначно как одна из двух элементарных трансляций в плоскости XOY.

Из уравнения (2.4) следует, что асимптотическое решение может быть получено, если решить уравнение переноса для одной ячейки со следующими граничными условиями:

$$\Phi^+(\mathbf{r}_s + \mathbf{l}(\mathbf{r}_s), \mathbf{v}) = \Phi^-(\mathbf{r}_s, \mathbf{v}) \exp(i\mathbf{B} \mathbf{l}(\mathbf{r}_s)), \quad (3.5)$$

где  $\Phi^+(\mathbf{r}, \mathbf{v})$  – поток нейтронов, влетающих в ячейку,  
 $\Phi^-(\mathbf{r}, \mathbf{v})$  – поток нейтронов, вылетающих из ячейки,  
 $\mathbf{B}$  – вектор баклинга.

Для асимптотической решётки с аксиальной утечкой ситуация упрощается. Рассматривается слой произвольной высоты  $(Z, Z+h)$ . Граничное условие на верхнем и нижнем торцах слоя имеет вид

$$\Phi^+(\mathbf{r}_s, Z+h, \mathbf{v}) = \Phi^-(\mathbf{r}_s, Z, \mathbf{v}) \exp(iB_z h), \quad (3.6)$$

где  $\mathbf{r}_s$  – радиус–вектор произвольной точки решётки в плоскости XOY.

Метод расчёта асимптотических решёток базируется на использовании уравнений (3.5), (3.6) в качестве граничных условий уравнения переноса. Текущая программная реализация предписывает пользователю выбирать величину  $h$  достаточно малой:  $B_z h \ll 1$ .

Решение асимптотических однородных задач проводится в приближении, когда распределение источников нейтронов деления в пределах каждой ячейки совпадает с тем, которое устанавливается в бесконечной решётке при  $\mathbf{B} = \mathbf{0}$ . Поэтому траектории и поколения нейтронов моделируются так же, как описано в предыдущем разделе. Специфика решения асимптотических задач заключается в использовании граничных условий (3.5) и (3.6), которые моделируются следующим образом.

При пересечении нейтроном границы ячейки (слоя)  $\mathbf{r}_s$  нейтрон переносится в точку  $\mathbf{r}_s + \mathbf{l}(\mathbf{r}_s)$  на противоположной стороне ячейки (слоя) без изменения его вектора скорости (здесь  $\mathbf{l}$  – один из векторов элементарных трансляций).

Траектория нейтрона продолжается из новой точки, которая является точкой нового входа нейтрона в ячейку (слой).

Вес нейтрона умножается на комплексный множитель  $\exp(i\mathbf{B} \mathbf{l}(\mathbf{r}_s))$ .



Таким образом, вес нейтрона в любой точке траектории равен произведению комплексных чисел  $\exp(i\mathbf{B} \cdot \mathbf{l}(r_s))$ , соответствующих всем предшествующим пересечениям границ ячейки (слоя).

Так как при решении асимптотических задач интерес представляет только реальная часть комплексного потока, при оценке функционалов используется только реальная часть веса, которая определяется на основе простой алгебраической формулы:

$$W = \operatorname{Re} \exp(i(a_1+a_2+ \dots +a_n)) = \cos(a_1+a_2+ \dots +a_n).$$

### 3.3 РАСЧЁТ ЭФФЕКТИВНОЙ ДОЛИ ЗАПАЗДЫВАЮЩИХ НЕЙТРОНОВ

Уравнение критичности (3.1) можно записать в виде

$$k_{eff}\mathbf{Q} = (\mathbf{H}_{prompt} + \Delta\mathbf{H})\mathbf{Q} \quad (3.7)$$

где оператор  $\mathbf{H}_{prompt}$  описывает деления только на мгновенных, а  $\Delta\mathbf{H}$  - на запаздывающих нейтронах

По определению эффективная доля запаздывающих нейтронов  $\beta_{eff}$  равна:

$$\beta_{eff} = \frac{K_{eff} - K_{prompt}}{K_{eff}} \quad (3.8)$$

где  $K_{eff}$  - эффективный коэффициент размножения системы с учётом как мгновенных, так и запаздывающих нейтронов, определяемый уравнением (3.1);

$K_{prompt}$  - эффективный коэффициент размножения с учётом только мгновенных нейтронов, определяемый уравнением:

$$k_{prompt}\mathbf{Q} = \mathbf{H}_{prompt}\mathbf{Q} \quad (3.9)$$

Используя теорию возмущений, величину  $\beta_{eff}$  можно выразить соотношением:

$$\beta_{eff} = \frac{(Q^+, \Delta\mathbf{H} Q)}{k_{eff}(Q^+, Q)} \quad (3.10)$$

где  $Q$  и  $Q^+$  соответственно главная собственная функция и функция ценности уравнения критичности с учётом деления только на мгновенных нейтронах.

Эта формула и используется в программе для вычисления величины  $\beta_{eff}$  в приближении, что функция ценности  $Q^+(x) = const$ :

$$\beta_{eff} = \frac{(e, \Delta\mathbf{H} Q)}{k_{eff}(e, Q)}, \quad (3.11)$$

Для вычисления эффективной доли запаздывающих нейтронов  $\beta_{eff}$  в каждом поколении моделируются траектории, не только мгновенных, но и запаздывающих нейтронов. Из каждой точки поглощения помимо нейтронов, моделируемых так, как описано в разделе 3.1, моделируются траектории такого же числа запаздывающих нейтронов с весом, равным доле запаздывающих нейтронов на поглотившем нейтрон предыдущего поколения ядре. Энергия запаздывающих нейтронов разыгрывается в соответствии со спектром запаздывающих нейтронов. В каждом поколении помимо нейтронов, родившихся в соответствии с полным спектром деления, моделируется столько же траекторий запаздывающих нейтронов, которые затем используются для оценки  $\beta_{eff}$ .

Эффективная доля запаздывающих нейтронов в асимптотическом режиме всегда вычисляется для  $\mathbf{B} = \mathbf{0}$ .

### 3.4 МЕТОД СУММАРНОГО ИЗОТОПА

При решении задач с выгоранием, а также при расчёте состояний с выгоревшим топливом можно существенно ускорить вычисления и высвободить оперативную память ЭВМ, используя метод суммарного изотопа.

Метод основан на том, что в каждом материале пользователь может заменить малозначачие изотопы (продукты деления и далёкие актиноиды) одним суммарным изотопом с нулевыми сечениями рассеяния и деления и сечениями поглощения и генерации, равными, соответственно

$$\Sigma_{a,eff}(E) = \sum_n \rho_n \sigma_{a,n}(E),$$

$$(v\Sigma)_{f,eff}(E) = \sum_n \rho_n (v\sigma_f)_n(E),$$

где суммирование проводится по всем малозначачим изотопам, определённым пользователем в файле исходных данных.

Это означает, что сечение суммарного изотопа учитывается в эффективном коэффициенте размножения, в сечениях и скоростях реакций с номерами 1 (полное), 3 (поглощения) и 918 (генерации).

Использование суммарного изотопа позволяет на порядок сократить количество изотопов, явно учитываемых в топливе. При этом, путём грамотного выбора списка малозначачих изотопов, можно добиться точности расчёта эффективного коэффициента размножения порядка 0,0001.

Сечение суммарного изотопа в сечениях и скоростях реакции деления не учитывается. Однако, как показывают расчётные исследования, проведённые для тепловых реакторов с топливом из двуокиси урана, если учитывать в топливе всего 12 изотопов, таких как U-235, U-236, U-238, Np-237, Np-238, Pu-238, Pu-239, Pu-240, Pu-241, Pu-242, Am-242m, Cm-245, то погрешность в определении скорости реакции деления в топливе будет меньше, чем 0,05 %.

Для определения списка малозначачих изотопов в выгоревшем топливе используют карту SIZI опции FINTAB (см. описание исходных данных для модуля выгорания).

При моделировании столкновения считается, что нейтрон поглощается суммарным изотопом с вероятностью  $\Sigma_{eff}(E, \mathbf{r}) / \Sigma_{tot}(E, \mathbf{r})$ .

При моделировании отличных от нейтронов частиц изотопы, вынесенные в суммарный изотоп в процессе моделирования не учитываются.

## 4 ВЫЧИСЛЕНИЕ ФУНКЦИОНАЛОВ ПОТОКА

В процессе моделирования поколений модуль регистрации вычисляет некоторый «стандартный» набор линейных и дробно-линейных функционалов потока и их статистические ошибки.

### 4.1 ОБЩИЕ ПОЛОЖЕНИЯ

Все 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  $d\mathbf{r}d\boldsymbol{\Omega}dE$  - область интегрирования в фазовом пространстве, в котором координаты частицы определяются радиус-вектором  $\mathbf{r}$ , направлением скорости  $\boldsymbol{\Omega}$  и значением энергии частицы  $E$ ;

$\varphi(\mathbf{r}, \boldsymbol{\Omega}, E)$  - весовая функция, определяющая конкретный вид функционала;

$\Phi(\mathbf{r}, \boldsymbol{\Omega}, E)$  - функция распределения потока нейтронов.

Все функционалы нормируются на поток от одной первичной частицы источника.

Эффективный коэффициент размножения при такой нормировке определяется интегралом:

$$k_{eff} = \int_V d\mathbf{r} \int_{4\pi} d\boldsymbol{\Omega} \int_{0 < E < \infty} dE v \Sigma_f(\mathbf{r}, \boldsymbol{\Omega}, E) \Phi(\mathbf{r}, \boldsymbol{\Omega}, E).$$

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

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

$$s = D^{0.5},$$

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

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

$$D_z = \frac{\bar{x}^2}{y^2} \left( \frac{D_x}{\bar{x}^2} - 2 \frac{M[(x - \bar{x})(y - \bar{y})]}{xy} + \frac{D_y}{\bar{y}^2} \right),$$

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

При вычислении функционалов используются следующие стандартные области интегрирования (*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.

Для всех областей интегрирования могут вычисляться разные функционалы из стандартного набора функционалов в заданных энергетических группах, которые определяются как энергетические интервалы, заданные своими верхними и нижними границами.

## 4.2 КЛАССИФИКАЦИЯ ФУНКЦИОНАЛОВ

В зависимости от выбора фазовых областей, по которым проводится интегрирование, в модуле регистрации используется следующая классификация функционалов:

- глобальные функционалы - интегралы по всей области фазового пространства, в которой определена рассчитываемая система;
- функционалы для регистрационных областей - интегралы по объёму, занимаемому областью.

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

Используются следующие оценки, вносящие вклад в рассчитываемые функционалы на каждой траектории:

- по точкам столкновений,  $u^{(col)} = \sum_{m=1}^M \frac{\varphi(\mathbf{r}_m, E)}{\Sigma_{tot}(\mathbf{r}_m)} W_m$ , где  $W_m$  - вес в точке столкновения,

а  $m$  - номер точки столкновения;

- по длине пробега,  $u^{(l)} = \sum_{m=1}^M \varphi(\mathbf{r}_m, E) l_{m, m-1} W_m$ , где  $W_m$  - вес в точке регистрации,

$m$  - номер точки пересечения траектории нейтрона с границей геометрической зоны,  $l_{m, m-1}$  - длина траектории в зоне;

- по точкам поглощения,  $u^{(a)} = \frac{\varphi(r_a, E)}{\Sigma_a(r_a, E)} W_a$ , где  $W_a$  - вес в точке поглощения.

Помимо выборочных значений математических ожиданий оцениваются так же их среднеквадратичные отклонения, а также среднеквадратичные отклонения дисперсий и коэффициенты асимметрии.

#### 4.2.1 Глобальные функционалы

Глобальные функционалы оцениваются только для нейтронов. Для них вычисляются следующие функционалы, характеризующие систему в целом: эффективный коэффициент размножения нейтронов; время жизни нейтронов; баланс нейтронов.

*Эффективный коэффициент размножения* нейтронов  $k_{eff}$  определяется как математическое ожидание числа вторичных нейтронов деления на один первичный нейтрон, рождённый в предыдущем поколении. Принято, что к вторичным нейтронам деления не относятся нейтроны, родившиеся в результате  $(n, kn)$  реакций.

Вычисляется несколько оценок коэффициента размножения, которые могут различаться как своими значениями, так и значениями статистических погрешностей:

- по точкам столкновений  $k_{col}$ ;
- по точкам поглощений  $k_{abs}$ ;
- по пробегам  $k_{len}$ ;
- комбинированная оценка по столкновениям и поглощениям

$$k_{com} = a k_{col} + (1-a) k_{abs};$$

- модифицированная комбинированная оценка Бриссендена

$$k_{Bris} = k_{col} + b(1-A),$$

где  $A$  - оценка суммы числа поглощённых и вылетевших из системы первичных частиц;  $a$  и  $b$  - некоторые весовые множители.

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 за исключением случаев неаналогового моделирования, где утечка также оценивается по столкновениям.

Все оценки равноправны. Пользователю рекомендуется принять значение той из них, которая имеет наименьшую статистическую погрешность.

*Утечка нейтронов* (LEAKAGE в листинге, содержащем финальную выдачу) через внешнюю поверхность при  $\mathbf{V} = \mathbf{0}$  подсчитывается прямым моделированием полного числа частиц, вылетающих через внешнюю поверхность рассчитываемой системы для каждой регистрационной группы без учёта размножения в  $(n, kn)$  реакциях. При  $\mathbf{V} \neq \mathbf{0}$  в величине LEAKAGE учитываются нейтроны, утечка которых из системы вычисляется с учётом заданного вектора  $\mathbf{V}$ .

*Баланс нейтронов* подсчитывается как сумма поглощённых и покинувших систему через внешнюю границу первичных частиц и должен быть строго равен 1, если в системе не произошло потерь частицы в результате допустимых приближений геометрического модуля.

Также оценивается *баланс нейтронов для определённого типа реакции* по всем нуклидам, заданным в исходных данных. Тип оценки и перечень реакций указывается пользователем в исходных данных.

Время жизни мгновенных нейтронов определяется формулой

$$\Lambda = \frac{\iiint \tau Q_0^+(x) G(x, x', \tau) Q_0(x') dx dx' d\tau}{\iint Q_0^+(x) H(x, x') Q_0(x') dx dx'}$$

где  $G(x, x', \tau)$  - плотность генерации мгновенных нейтронов деления в точке  $x$  по прошествии времени  $\tau$  после появления в точке  $x'$  нейтрона деления;

$$H(x, x') = \int_0^{\infty} G(x, x', \tau) d\tau$$

$Q_0(x')$  - стационарная плотность делений в условно-критическом реакторе;

$Q_0^+(x)$  - функция ценности, имеет смысл асимптотического числа нейтронов в условно-критическом реакторе по прошествии большого числа поколений при условии, что в начальный момент времени в точке  $x$  находился один нейтрон.

$Q_0(x')$  и  $Q_0^+(x)$  удовлетворяют уравнениям:

$$k_{eff} Q_0 = H Q_0,$$

$$k_{eff} Q_0^+ = H^+ Q_0^+.$$

В программе время жизни мгновенных нейтронов вычисляется в приближении  $Q_0^+(x) = 1$ , где  $x$  – точка рождения вторичных нейтронов деления.

#### 4.2.2 Функционалы, вычисляемые для регистрационных областей

Для каждой регистрационной области в заданных энергетических группах вычисляются скорости реакций нескольких заданных типов, как для каждого нуклида, так и для их композиции (*MIXT*) в данной области:

$$R_n = \int \Sigma_n(\mathbf{r}, E) \Phi(\mathbf{r}, \Omega, E) d\mathbf{r} d\Omega dE,$$

где  $\Sigma_n(\mathbf{r}, E)$  - макроскопическое сечение ядерной реакции с номером  $n$ , либо для отдельного нуклида, либо для всей композиции (*MIXT*) в данной регистрационной области.

В таблице 4.2.2.1 приведена нумерация типов скоростей реакций, доступных в MSU-PTR

Таблица 4.2.2.1 – Нумерация типов скоростей реакций, доступных в MSU-PTR

Номер реакции	Описание реакции	Примечание
Нейтроны		
1	Полное сечение	-
2	Упругое рассеяние	-
3	Поглощение	-
4	Неупругое рассеяние	Отсутствует в ФИМТОЕН, МОФИТГ
16	(n,2n) реакция	Отсутствует в ФИМТОЕН, МОФИТГ
17	(n,3n) реакция	Только в FARION

Номер реакции	Описание реакции	Примечание
18	Деление	-
22	(n,na)	Только в FARION
23	(n,n3a)	Только в FARION
24	(n,2na)	Только в FARION
25	(n,3na)	Только в FARION
28	(n,np)	Только в FARION
41	(n,2np) + (n,3np)	Только в FARION
101	Захват	-
102	Радиационный захват	Отсутствует в ФИМТОЕН, МОФИТТГ
103	(n,p) + (n,n2p) + (n,npa) + (n,pd) + (n,pt) + (n,n2a) + (n,2n2a)	Только в FARION
107	(n,a) + (n,da)	Только в FARION
108	(n,2a) + (n,3a)	Только в FARION
111	(n,2p)	Только в FARION
112	(n,pa)	Только в FARION
113	(n,t2a) + (n,d2a)	Только в FARION
251	Средний косинус угла рассеяния	Отсутствует в ФИМТОЕН
452	Средний выход нейтронов на деление	Отсутствует в FARION
455	Средний выход запаздывающих нейтронов на деление	-
918	Произведение $\nu\Sigma_{fis}$	-
Фотоны		
501	Столкновение	-
504	Комптоновское рассеяние	-
516	Образование пар	-
602	Фотоэффект	-

Также для всех регистрационных областей оцениваются макроскопические сечения реакций по области в целом

$$\bar{\Sigma}_n = \frac{R_n}{\Phi_i},$$

где  $\Phi_i$  – оценка потока в данной регистрационной области.

## **4.3 ОПИСАНИЕ ВЫДАЧИ НА ПЕЧАТЬ МОДУЛЯ РЕГИСТРАЦИИ**

Модуль регистрации печатает значения потока и стандартный набор функционалов потока (Mean), а также их среднеквадратичные отклонения (StdDev) и по требованию среднеквадратичные отклонения дисперсий (StdDevOfDisp) и коэффициенты асимметрии (CoefOfSkew).

## **4.4 РАСЧЕТ С ИСПОЛЬЗОВАНИЕМ АКТИВАЦИОННЫХ СЕЧЕНИЙ**

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 расчете 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 на  $\text{см}^3$ .

## **4.5 ПОДМОДУЛЬ USER**

При компоновке программы в неё всегда включается подмодуль регистрации USER, который может быть модифицирован пользователем и включён в программу вместо стандартного эмулятора пустышки URGS5xx.GUR.

Написание подмодуля USER позволяет пользователю осуществить практически любые дополнительные регистрации и выдачи на печать.

Названия подпрограмм подмодуля дублируют названия подпрограмм основного модуля регистрации, отличаясь от них только дополнительной начальной буквой U.

В случае наличия карты, начинающейся с буквы U в исходных данных модуля регистрации подпрограммы регистрационного модуля после окончания собственной работы вызывают подпрограмму подмодуля USER, имеющую такое же имя, но с добавленной спереди буквой U.

Пользователь имеет право вставить любые осмысленные команды в подпрограммы подмодуля URGS5xx.GUR, модифицировав его в свой подмодуль.

Пользователь самостоятельно программирует накопление, сохранение и обработку требуемой информации.

## **5 ОБЩАЯ СХЕМА РАСЧЁТА**

Для проведения расчёта пользователю 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 и 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.

### **5.1 ШАГ INPUT**

Производится ввод исходных данных, подготовка и запись в рабочие файлы задачи информации для проведения расчёта. С запуска этого шага начинается расчёт каждого варианта. Нормальное завершение работы шага является необходимым условием для запуска и работы остальных шагов. Поэтому после отработки 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 пользователя образуются следующие файлы:

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

NAME.DAT – текстовый файл, который содержит исходные данные для шага CALCULATION;

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.

## 5.2 ШАГ CALCULATION

Осуществляется непосредственное проведение расчёта варианта. Режим работы шага определяется содержанием файла NAME.DAT, который образуется путём автоматического редактирования 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 спасающая запись информации, накопленной в результате моделирования. Это позволяет многократно прерывать и возобновлять расчёт без потери накопленных за время счёта результатов. 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». В противном случае все накопленные результаты будут потеряны, и продолжение расчёта будет невозможно.

В файл NAME.LST поступают сообщения о прохождении задачи.

## 5.3 ШАГ OUTPUT

На шаге происходит финальная обработка и печать результатов расчёта в файл NAME.FIN. Режим работы шага определяется 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.

## 5.4 ШАГ BURNUP

На этом шаге 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 которых добавлено окончание вида «\_Vnnn», где nnn – это количество шагов BURNUP, завершившихся к данному моменту времени.



## 5.5 ВЫЗОВ СИСТЕМНЫХ КОМАНД

В программу добавлена возможность вызова любых системных команд и, следовательно, запускать любые программы между шагами расчётной схемы (ввод, счёт, финальная обработка, выгорание). Для этого в одной директории с файлом исходных данных должен находиться файл MCU5.SYS. В этом файле записываются необходимые системные команды с указанием момента их выполнения. Файл обрабатывается построчно. Максимально возможная длина строки составляет 256 символов. Одна строка не может содержать более одной команды. Команда не может располагаться на нескольких строках. Используется следующий формат записи команд. Сначала записывается номер точки входа, затем двоеточие (:), далее любая системная команда. Строки без двоеточия не считаются командными. Точки входа проверяются по первой строке файла NAME.SYS и соотносятся с шагами расчётной схемы следующим образом:

- 0 – перед копированием исходных данных в файл с расширением dat,
- 1 – после копирования и до шага input;
- 2 – после шага input;
- 3 – до шага calculation;
- 4 – после шага calculation;
- 5 – до шага output;
- 6 – после шага output;
- 7 – до шага burnup;
- 8 – после шага burnup.

Например, добавление в такой файл команды

```
6:copy *.* SAVE\*.*
```

при наличии в расчётной директории поддиректории SAVE вызывает автоматическое копирование в эту папку всех файлов на момент после окончания финальной обработки.

## 5.6 ИЗМЕНЕНИЕ ПАРАМЕТРОВ РЕАКТОРА В ПРОЦЕССЕ РАСЧЁТА ВЫГОРАНИЯ

Данная функция обеспечивает возможность автоматического изменения любых параметров в процессе расчёта выгорания, в том числе и положения стержней при регулировании работы реактора (зависит от версии управляющего модуля).

В соответствии с описанной выше схемой работы программы изменение параметров осуществляется непосредственно перед расчётом очередного состояния, а именно в точке входа 1.

Для обеспечения возможности автоматического изменения параметров модели в процессе расчёта выгорания используется возможность задания части данных в файлах, отличных от файла исходных данных с помощью карты #INCLUDE (см. описание Программных средств стандартного ввода).

Перед каждым шагом INPUT в точке входа 0 программа автоматически выполняет в директории с файлом исходных данных поиск файлов с именем вида BURinnnn для i от 0 до 9 включительно с nnnn равным текущему номеру шага по выгоранию. При этом nnnn – четырехзначное число от 0 до 9999 с нулями в пустых позициях слева, которое соответствует числу из второй строки файла NAME.SYS.

При наличии файлов с именем вида BURinnnn программа переименовывает их в файлы с именем вида BURi. При этом в файле исходных данных должны быть ссылки на файлы BURi, выполненные с помощью карты #INCLUDE (например, #INCLUDE BUR5). Данная схема обеспечивает использование для каждого очередного шага по выгоранию с порядковым номером nnnn своих собственных включений BURi в текст исходных данных. Следует помнить, что реально программой используются только те файлы BURi, на которые

имеются ссылки в файле исходных данных, поэтому файлы BURinnnn с другими  $i$  можно не создавать.

Необходимо отметить, что нумерация шагов по выгоранию начинается с нуля. Это соответствует самому началу счёта, когда ещё ни одного шага по выгоранию выполнено не было. Таким образом, перед самым первым шагом INPUT программа выполняет поиск всех файлов вида BURi0000 (для  $i$  от 0 до 9) и, если такие файлы имеются, переименовывает их в файлы BURi.

Файлы с именем вида BURi в процессе счёта не удаляются. Это означает, что при отсутствии файлов для очередного шага по выгоранию программа автоматически использует данные от предыдущего шага. Таким образом, нет необходимости создавать файлы BURinnnn для каждого шага по выгоранию, а можно ограничиться только теми шагами, на которых в действительности происходит изменение параметров.

Файлы BURi0000 для нулевого шага должны присутствовать обязательно.

Такая автоматизация в совокупности с использованием стандартной для исходных данных карты #INCCLUDE позволяет изменять параметры расчётной модели в процессе расчёта выгорания в зависимости от номера шага по выгоранию.

Описанные возможности могут быть использованы, например, для того, чтобы автоматизировать процесс изменения положения рабочих органов СУЗ (поглощающих стержней) и других параметров расчётной модели в процессе собственно расчёта, не изменяя исходные данные исходной базовой расчётной модели. Для этого в файл исходных данных, например, в раздел геометрического модуля с помощью карты #INLCUDE в необходимые места добавляются ссылки на файлы вида BURi. В соответствующих этим файлам файлам вида BURinnnn определяются изменяемые параметры для соответствующего шага по выгоранию pppp.

*Модуль выгорания должен использоваться в режиме, когда сечения и поток нейтронов рассматриваются постоянными.*

## **6 РЕЖИМ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛЕНИЙ**

Распараллеливание программы осуществлено на базе программного интерфейса MPI (Message Passing Interface). Он является наиболее распространённым 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 остаётся такой же, как и при расчёте на одном процессоре.

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 также, как и в случае не распараллеленных вычислений

На этапе CALCULATION каждый процессор получает свою собственную последовательность случайных чисел, непересекающуюся 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  $K_{эфф}$  для каждого процессора задаётся расчёт только одной серии, а общее количество процессоров задаётся большим. В этом случае суммарное число просчитанных историй будет велико, однако результат на каждом процессоре и, следовательно, и 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.

На этапе BURNUP нулевой процессор снабжает каждый из остальных процессоров данными для расчёта выгорания одного материала, после чего сам начинает расчёт выгорания очередного материала. Закончив свой расчёт, нулевой процессор сохраняет его результаты на внешний носитель. После этого он последовательно собирает результаты вычислений других процессоров и сохраняет их на внешний носитель. Такой подход позволяет экономить оперативную память нулевого процессора, поскольку в каждый момент времени в ней содержится информация только об одном материале. Далее нулевой процессор снова снабжает остальные информацией о необработанных материалах и проводит собственные вычисления. Такой цикл продолжается до тех пор, пока не будут получены решения для всех материалов.

*Результаты расчетов транспортного модуля всегда относятся к одному процессору.*

## **7 СХЕМА ЗАДАНИЯ ИСХОДНЫХ ДАННЫХ**

Расчёты по программе осуществляются на основании содержания файла исходных данных с учётом выбранной расчётной компоновки. Набор исходных данных представляет собой файл с именем, задаваемым пользователем и состоящим из фрагментов, в которые записываются исходные данные для каждого модуля программы.

Файл исходных данных состоит из набора фрагментов:

- а) описание материального состава (физический модуль);
- б) описание геометрии (геометрический модуль);
- в) описание источников (модуль источников);
- г) данные для оценки функционалов (модуль регистрации);

д) данные для подготовки сечений для расчёта выгорания (модуль регистрации для выгорания);

е) данные для шага моделирования траекторий с конкретизацией выбранного алгоритма (модуль траекторий);

ж) данные для управления счётом (управляющий модуль).

з) данные для расчёта изменения изотопного состава материалов в процессе выгорания (модуль выгорания).

Фрагменты должны быть записаны в указанном порядке. После данных позади последнего фрагмента может быть записана любая справочная информация.

Фрагменты с а) по ж) используют программные средства стандартного ввода.

Фрагмент з) использует программные средства бесформатного ввода.

Все единицы, имеющие размерность длины вводятся в сантиметрах.

Нельзя использовать символ табуляции.

## 7.1 ПРОГРАММНЫЕ СРЕДСТВА СТАНДАРТНОГО ВВОДА

По умолчанию символы в строках в позициях больших, чем 200, игнорируются, так что можно считать, что все строки имеют в длину не более 200 позиций. Предельная длина строки может быть переопределена при генерации программы в файле параметров с помощью карты

```
+ $LSTRI          $<число>$
```

где <число> - целое число от 80 до 256, определяющее новую предельную длину строки.

Всё описание состоит из отдельных предложений, каждое из которых занимает одну или несколько последовательных строк.

Первая строка любого предложения в первой позиции обязана содержать букву латинского алфавита, с которой начинается идентификатор из не более, чем 6 символов, называемый меткой предложения. Все буквы латинские; заглавные и строчные буквы различаются.

Смысл метки различен для разных типов предложений и будет описан ниже.

Метка отделяется от остальной части предложения хотя бы одним пробелом.

Любая не первая строка предложения имеет в качестве признака строки-продолжения пробел в первой позиции.

Всё предложение кроме метки трактуется как один символьный массив, называемый параметром предложения.

Параметры состоят из отдельных фрагментов, разделённых пробелами.

Кроме пробелов разделителями фрагментов могут служить запятая или запятая с пробелами.

Если предложение содержит символ «;», то он служит признаком окончания предложения. Любая информация, записанная в предложении после этого символа, трактуется как комментарий и не обрабатывается.

По умолчанию предельная суммарная длина предложения равна 2000000 символов, причём заключительные пробелы в окончаниях строк, а также комментарий в предложение не входят.

Предельная суммарная длина предложения может быть переопределена при генерации программы в файле параметров с помощью карты

```
+ $LBUFS          $<число>$
```

где <число> - целое число, определяющее новую предельную суммарную длину предложения.

Между предложениями, но не разрывая отдельные предложения, могут находиться строки комментариев. Такие строки имеют либо символ «\*» в первой позиции, либо символы «С=» в первой и второй позициях. Эти строки не обрабатываются.

Строки, имеющие в первых трёх позициях символы «C=C», а в четвёртой – пробел, управляют формой выдачи диагностической печати геометрического модуля.

Числа записываются так же, как в языке Фортран–90/95/2003.

Тип числа определяется не только его формой, но и положением в конкретном месте конкретного предложения. Если на данном месте должно быть действительное число, то записи 1 и 1.0 эквивалентны.

Однако, если подразумевается ввод целого числа, то недопустимо использовать форму действительного числа даже без дробной части. Например, использование для целого числа 100 записи 100.0 или 1E2 вызывает диагностику ошибки.

#### Пример

Целые числа без знака:	1	24	333
Целые числа:	24	-5	
Действительные числа:	24	2.4	1. 0.5E-6

Помимо целых и действительных чисел в исходных данных могут использоваться константы символьного типа. Это последовательности из не более чем 6 латинских букв и цифр, начинающиеся с буквы. *Заглавные и строчные буквы считаются различными.*

В программе имеется возможность задания часть данных в файлах, отличных от файла исходных данных. Ссылка на такие файлы осуществляется картой

```
#include <имя файла>
```

или

```
#INCLUDE <имя файла>
```

где *#include* – имя карты исходных данных для MCU,

*<имя файла>* – имя файла, содержимое которого будет вставлено в исходные данные в момент ввода вместо этой карты. Если это не полное имя файла, то файл будет запрошен из директории с исходными данными. Включаемый таким образом файл не должен в свою очередь содержать карт *#include*. При наличии таких карт будет генерироваться сообщение об ошибке.

Само по себе добавление такой функции позволяет выносить часть данных во внешние файлы. Особенно удобным это может оказаться в тех случаях, когда исходные данные для какого-либо раздела MCU генерируются автоматически и содержат большое количество данных, что затрудняет просмотр остальных, заданных вручную частей.

В геометрическом модуле и модуле источников кроме чисел в предложениях могут использоваться идентификаторы, являющиеся константами символьного типа.

Действительная числовая константа обозначается либо соответствующим числом, либо именем, присвоенным некоторому числовому значению. Присвоение имени осуществляется предложением

```
EQU <имя> = <выражение>
```

где *EQU* - метка предложения,

*<имя>* - идентификатор,

*<выражение>* - аналогично арифметическому выражению языка Фортран-77.

Выражение строится из чисел, символических имён, ранее уже присвоенных константам, знаков арифметических операций, скобок, а также пяти встроенных функций *sin* или *SIN*, *cos* или *COS*, *tg* или *TG*, *sqrt* или *SQRT* и *funh* или *FUNH*.

Аргумент тригонометрических функций задаётся в градусах.

Именем *funh* обозначена функция Хевисайда, имеющая 0 при отрицательном аргументе и 1 при нуле и положительном аргументе.

Одному числовому значению может быть присвоено несколько имён. Такие константы эквивалентны.

#### Пример

```
EQU PI = 3.1415926
EQU vol1 = 3.2*(6.-0.3)*(6-0.2)
EQU VOL2 = vol1+3.7*3.7*2
EQU C1 = sqrt(3.)/2
EQU C2 = cos(30)
```

Отметим, что C1 и C2 имеют одно и то же значение с точностью до ошибки округления.

В предложениях с ключом EQU в выражениях допускаются внутренние пробелы.

Предложения присвоения значений константам могут встречаться в любом месте описания. Однако использовать константу можно только после присвоения ей значения.

При описании простых систем любое имя константы может быть определено только один раз. Переопределения не допускаются, то есть наличие в тексте предложений

```
EQU ABC = 10
EQU ABC = 10.5
```

вызывает диагностику и останов программы.

Существует другая форма присвоения имён константам (которых в этом случае более естественно назвать переменными)

```
SET <имя> = <выражение>
```

Разница между первой и второй формой заключается в том, что во втором случае значение константы может быть переустановлено произвольное число раз новой инструкцией SET. Это оказывается удобным, если некоторое имя употребляется для локально используемой величины, нужной только в небольшой части описания. При отсутствии переопределения каждая такая величина требовала бы собственного имени, что увеличивает таблицу имён и вызывает необходимость использования более длинных идентификаторов.

Строка

```
C=C SHOW
```

вызывает выдачу в файл диагностической информации таблицы всех определённых к данному моменту имён констант и их значений.

В случае использования директивы C=C SHOW в таблицу поступают последние на момент появления этой директивы значения констант с точностью до четырех знаков.

Использование констант не зависит от того, какой инструкцией они определены.

#### Пример. Использование конструкции SET

```
EQU A = 48.
EQU R = 50.
SET B = SQRT(R*R-A*A)
* использование значения B
RPP          0,A    -B,B    0,100
* переопределение B и использование нового значения
SET B = B/A
PLG 1,B,0 1
```

Для ввода повторяющихся данных используется следующая конструкция (возможно на нескольких строках с учётом первого пробела, обозначающего дополнительную строку)

*[number|value]*

где *number* – количество повторений, целое число;  
*value* – повторяемая последовательность символов.

Повторяется вся последовательность символов между символами «|» и «|», включая пробелы. При нулевом или отрицательном значении *number* повторение не осуществляется. При вводе исходных данных в предложение входит только собственно конструкция, что при наличии повторяемых конструкций позволяет использовать предложения меньшей длины.

Приведём эквивалентные конструкции:

10., 10., 10.,  
[3|10.,]

и

MCCCU  
M[3|C]U

## 7.2 ПРОГРАММНЫЕ СРЕДСТВА БЕСФОРМАТНОГО ВВОДА

Они позволяют использовать режим по умолчанию и вводить только те величины, для которых не существует стандартных значений, или их следует изменить.

Фрагмент исходных данных при использовании бесформатного ввода представляет собой набор предложений, каждое из которых занимает несколько последовательных строк. Первая строка предложения – основная, остальные, следующие за ней, – дополнительные. Первая строка предложения в позициях 1 – 6 содержит его имя. Имя может состоять из букв, цифр, пробелов и символа «\*».

Первым символом имени должна быть заглавная латинская буква. Как правило, предложение соответствует одному массиву данных.

Позиции 7 – 72 основной строки и 2 – 72 дополнительных строк содержат используемую информацию. Первая позиция каждой дополнительной строки должна содержать пробел. Данные состоят из слов, разделённых пробелами, запятыми и/или круглыми скобками. Запятая и круглые скобки называются терминаторами.

Каждое слово – это цепочка из не более 12 символов, не содержащая пробелов и терминаторов. Слово должно располагаться на одной строке. Если слово заканчивается на позиции 72, то считается, что за ним следует пробел.

Если слово имеет большую длину, то выдаётся предупредительная диагностика, и используются только первые 12 символов.

Среди слов выделены текстовые, целые, целые без знака и действительные числа, синтаксис которых совпадает с синтаксисом языка Фортран–90/95/2003 за исключением действительных чисел с порядком, при записи которых порядок может быть отделён от мантиссы своим знаком (т.е. без использования буквы E). Например,

ABCDEF, GHI, ТЕХТ, Т, DTEM, U235 – текстовые константы;  
38, 421, 47, 5 – целые без знака;  
-1, +21, 312 – целые;  
1.28E1 = +1.28E+1 = 128-1, 314E-2, -45 – действительные;  
.81 = 0.81 -0.24E-10 – действительные.

При разделении слов можно комбинировать пробелы и терминаторы, руководствуясь следующими правилами:

- слова разделяются не более чем одним терминатором;
- до и после терминатора может быть любое количество пробелов;
- если есть терминатор, пробелов может вообще не быть;
- если два последовательных слова находятся на разных строках, то терминатор должен располагаться на той же строке, что и первое слово;
- два терминатора подряд трактуются как пустое слово.

Рекомендуется либо использовать только пробелы, либо ставить терминатор непосредственно после слова.

Приведём пример эквивалентных конструкций:

```
17,28.5
17  ,    28.5
17    28.5
17, 28.5
```

Для ввода повторяющихся данных используются коэффициенты повторения последовательностей слов (возможно на нескольких строках), заключённых в скобки. Перед открывающей скобкой стоит целое без знака, отличное от нуля и означающее число повторений.

Приведём две эквивалентные конструкции:

```
10.,10.,10.
3(10.)
```

Допускается следующее отклонение в синтаксисе: непосредственно после закрывающей скобки можно ставить запятую.

## 8 ИСХОДНЫЕ ДАННЫЕ ДЛЯ ФИЗИЧЕСКОГО МОДУЛЯ

К числу основных понятий в физическом модуле относится понятие МАТЕРИАЛа.

Считается, что вся система состоит из задаваемого пользователем набора материалов. Материал характеризуется номером, температурой и нуклидным составом.

Необязательные карты и параметры файла исходных данных приводятся в квадратных скобках «[...]».

Ввод и обработка фрагмента с данными для модуля производится программными средствами стандартного ввода.

Фрагмент с исходными данными для физического модуля содержит сведения о материальном составе рассчитываемой системы. Признаком начала ввода данных к физическому модулю служит карта PIN. Признаком окончания ввода данных к физическому модулю служит карта FINISH. В общем случае этот фрагмент состоит из следующих разделов.

```
PIN
<описание материалов>
[<описание температуры системы>]
[<описание суммарного изотопа>]
[<задание управляющих параметров>]
FINISH
```

Наличие во фрагменте раздела с описанием материалов является обязательным. Остальные разделы могут быть опущены. В этом случае определяемые на уровне ввода значения соответствующих параметров будут приняты программой по умолчанию.

Порядок ввода карт внутри фрагмента произволен.



## 8.1 ЗАГОЛОВОК ФРАГМЕНТА ИСХОДНЫХ ДАННЫХ ДЛЯ ФИЗИЧЕСКОГО МОДУЛЯ

Фрагмент исходных данных для физического модуля должен начинаться с обязательной карты PIN.

*PIN* [*value1 value2*]

*PIN* - имя карты — заголовка фрагмента.

*value1* - определяет наличие/отсутствие печати карт файла исходных данных в выходном файле (1 – печатать, 0 – не печатать). Значение по умолчанию — 0.

*value2* - определяет наличие/отсутствие отладочной печати в выходном файле, а при наличии печати также и её количество (0 – не печатать, 1,2,... – печатать). Значение по умолчанию — 0. При значении этого параметра равном

1 – непосредственно после описания каждого материала печатается его материальный состав с указанием концентраций и всех параметров нуклидов, после окончания ввода материалов печатается таблица температур материалов;

2 – печатаются параметры нуклидов с учетом переопределений, сделанных пользователем с помощью карт DEF.

3 – печатается содержимое библиотечного файла DEFAULT.PHY;

4 – печатается содержимое библиотечного файла AW.LIB;

5 – печатается дополнительная служебная информация.

## 8.2 ОПИСАНИЕ МАТЕРИАЛОВ

В этом разделе задаётся материальный состав рассчитываемой системы.

Общий вид задания раздела следующий:

<описание первого материала>

...

<описание последнего материала>

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

Общий вид формата описания материала следующий:

```
MATR number [T = t GROUP = group NAME = type DENSAA = densaa  
DENSWA = denswa DENSAW = densaw DENSWW = densww VOL = vol  
BUR = bur CCM1 = ccm1 CCM2 = ccm2 CCM3 = ccm3 CCM4 = ccm4]  
name dens [ACE = ace MODS = mods BLOCK = block EHR = ehr  
DTEM = dtem PHT = pht PHS = phs EUR = eur FCB = fcb WCB = wcb]  
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* – символьное имя группы, к которой относится материал. 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 раздела. Имя группы и указанный в этой карте номер материала могут быть использованы при описании геометрии системы. Константа символьного типа. По умолчанию материал не относится ни к какой группе.

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

– MCU – стандартные 4-символьные имена, принятые в в банке данных MDBPD (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 6-C-12 задание ZZZAAA как 6012 или 006012 соответствует заданию стандартного для MDBPD имени C12).

*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.

*vol* – объём, занимаемый материалом, см<sup>3</sup>; константа вещественного типа. При задании этого параметра хотя бы в одном из материалов в файл выдачи программы на этапе ввода печатается карта VOL с указанием объёмов для каждого материала последовательно, начиная с первого. Это позволяет указывать значения объёмов материала непосредственно при вводе материалов. После выполнения ввода (этап INPUT) пользователь может использовать полученную таким образом карту, например, для копирования её в раздел исходных данных для модуля регистрации для выгорания. В настоящее время автоматическая передача значений, введённых с помощью этого параметра в другие модули, не осуществляется. Значение по умолчанию равно -1, что позволяет быстро определить отсутствующие позиции в печатаемой карте VOL.

*bur* – свойства материала с точки зрения изменения его изотопного состава в процессе облучения; константа символьного типа. Возможные значения параметра: F – выгорает, выделяется энергия, содержит делящиеся изотопы; A – выгорает, энергия не выделяется, состав произволен; P – не выгорает, выделяется энергия, содержит делящиеся изотопы; N – не выгорает, энергия не выделяется, состав произволен (принято по умолчанию). При задании этого параметра хотя бы в одном из материалов в файл выдачи программы на этапе ввода печатаются карты FISZ, ABSZ, POWZ с перечислением соответствующих номеров материалов в формате модуля выгорания. Это позволяет указывать свойства материала при облучении непосредственно при вводе материалов. После выполнения ввода (этап INPUT) пользователь может использовать полученную таким образом карту, например, для копирования её в раздел исходных данных для модуля выгорания. В настоящее время автоматическая передача значений ведённых с помощью этого параметра в другие модули не осуществляется.

Четыре параметра CCMi (i = 1–4) зарезервированы для моделирования электронов и позитронов и в данной программе не используются.

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

*dens* – при отсутствии параметров DENSAA, DENSWW, DENSAAW или DENSAAWA определяет число ядер нуклида в одном кубическом сантиметре (указывается без множителя 10<sup>24</sup>); константа вещественного типа. При наличии параметра DENSAA или DENSAAWA определяет атомную долю нуклида в материале, при наличии параметра DENSAAW или DENSAAWW определяет 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.

*ase* – определяет выбранный для расчёта файл оценённых ядерных данных и совпадает с расширением файла в библиотеке ACE/MCU; используется подмодулем FARION; константа символьного типа.

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

- G – модель идеального одноатомного газа;
- T – модель тяжёлого газа;
- COHR – расчёт сечений для бериллия и графита с учётом когерентных эффектов при упругом рассеянии
- H<sub>2</sub>OK – расчёт сечений для атома H с учётом химической формулы соединения H<sub>2</sub>O;
- CH<sub>2</sub>K – расчёт сечений для атома H с учётом химической формулы соединения CH<sub>2</sub>;
- ZRNK – расчёт сечений для атома H с учётом химической формулы соединения ZrH;
- HYN – расчёт сечений для атома H с учётом химической формулы соединения YH (только для подмодуля МОФИТТГ);
- D<sub>2</sub>OK – 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.

*block* – способ учёта резонансных эффектов для нуклида; используется подмодулем ФИМБРОЭН; константа целого типа:

- 0 – резонансная самоэкранировка сечений не учитывается;
- 1 – резонансная самоэкранировка сечений учитывается при помощи f-факторов Бондаренко с использованием данных библиотеки BNAB/MCU;
- 2 – резонансная самоэкранировка сечений учитывается с помощью подгрупповых параметров с использованием данных библиотеки BNAB/MCU;
- 3 – в области энергий разрешённых резонансов сечения рассчитываются в формализме Брейта-Вигнера или Адлер-Адлера на основе параметров, содержащихся в библиотеке LIPAR (файлы с расширением LIP), а в области энергий неразрешённых резонансов используются данные библиотеки MULTIC;
- 4 – в области энергий разрешённых резонансов сечения рассчитываются в формализме Брейта-Вигнера или Адлер-Адлера на основе параметров, содержащихся в библиотеке LIPAR (файлы с расширением LIP), а в области энергий неразрешённых резонансов используются данные библиотеки BNAB/MCU;
- 8 – в области низких энергий используются константы библиотеки KORT, а для энергий нейтронов, больших верхней границы библиотеки KORT, используются константы BNAB/MCU;
- 9 – в области энергий разрешённых резонансов сечения рассчитываются в формализме Рейха-Мура с использованием параметров, содержащихся в библиотеке LIPAR (файлы с расширением LPW), а в области энергий неразрешённых резонансов используются данные библиотеки BNAB/MCU.

*ehr* – значение верхней границы области энергии разрешённых резонансов, эВ, константа вещественного типа.

*dtem* – допуск по температуре (градусы Кельвина); определяет интервал [Т<sub>к</sub>-dtem, Т<sub>к</sub>+dtem] поиска констант нуклида для температуры Т<sub>к</sub> в библиотеках TEPCON и VESTA; константа вещественного типа.

*pht* – определяет выбранный для расчёта файл оценённых ядерных данных и совпадает с расширением файла в библиотеке PHOTONT; используется подмодулем ФОТОН; константа символьного типа.

*phs* – определяет выбранный для расчёта файл оценённых ядерных данных и совпадает с расширением файла в библиотеке PHOTONS; используется подмодулем ФОТОН; константа символьного типа.

*eur* – верхняя граница области неразрешённых резонансов; используется подмодулем ФИМБРОЭН; константа вещественного типа.

*fcб* – зарезервировано для моделирования электронов.

*wcb* – зарезервировано для моделирования электронов.

*END* - имя карты окончания описания материала. Эта карта может быть пропущена, если следующей картой является карта MATR или карта FINISH.

Параметры перечисляются в произвольном порядке и могут отсутствовать. Если параметр был задан несколько раз, то используется последнее введённое значение.

Если для одного материала было одновременно использовано несколько различных параметров вида DENSxx (DENSAA, DENSWW, DENSAAW или DENSAAWA), то используется последний заданный параметр.

Имя нуклида *name* записывается с первой позиции, а за ним через пробел записывается его ядерная плотность *dens*. Если на одной строке расположено больше одного нуклида, то перед описанием последующего нуклида ставится знак «слеш» («/»).

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.

Файл DEFAULT.PHY *не подлежит редактированию* пользователем.

Файл расположен в банке данных программы и состоит из последовательности строк. Строки с символом «\*» в первой позиции являются комментарными. Строки данных в обязательном порядке завершаются строкой с символом «#» в первой позиции. Строка данных содержит информацию об одном изотопе и имеет следующую структуру параметров, отделяемых друг от друга хотя бы одним пробелом:

- NAME (в формате MCU);
- ACE;
- MODS;
- BLOCK;
- EHR;
- DTEM;
- PHS;
- PHT;
- PRD (зарезервирован для дальнейшего использования);
- EUR;
- FCB;
- WCB;
- порядковый номер строки данных.

Файл AW.LIB расположен в банке данных программы и состоит из последовательности строк. Строки с символом «\*» в первой позиции являются комментарными. Строки данных в обязательном порядке завершаются строкой с символом «#» в первой позиции. Строка данных содержит информацию об одном изотопе и имеет следующую структуру параметров, отделяемых друг от друга хотя бы одним пробелом:

- NAME (в формате MCU);
- ZZZAAA;
- AW (атомный вес изотопа).

### **8.3 ИЗМЕНЕНИЕ ЗНАЧЕНИЙ ПО УМОЛЧАНИЮ, ЗАДАННЫХ В ФАЙЛЕ DEFAULT.PHY**

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

*DEF name [ACE = ace MODS = mods BLOCK = block EHR = ehr  
DTEM = dtem EUR = eur FCB = fcb WCB = wcb]*

*DEF* - имя карты.

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

*ace, mods, block, ehr, dtem, eur, fcb, wcb* – новые значения по умолчанию (см. описание материала) для нуклида *name*.

Все карты DEF должны располагаться в разделе данных для физического модуля до описания первого материала.

Можно использовать любое количество таких карт. Если для какого-либо параметра одного нуклида было задано несколько различных значений, то используется последнее заданное значение.

Изменения, вносимые с помощью этой карты, учитываются при вводе исходных данных для физического модуля и сохраняются

## 8.4 ОПИСАНИЕ ТЕМПЕРАТУРЫ СИСТЕМЫ

В разделе описания температуры системы задаётся значение температуры в градусах Кельвина, приписываемое всем материалам. При отсутствии данного раздела во фрагменте значение температуры системы, принимаемое по умолчанию, равно 300 К. При наличии в файле исходных данных более одной карты TEMPR используется значение, заданное в последней карте.

Общий вид формата задания раздела следующий:

*[TEMPR value]*

*TEMPR* - имя заголовка раздела.

*value* - значение температуры, константа вещественного типа.

## 8.5 ОПИСАНИЕ СУММАРНОГО ИЗОТОПА

В разделе описания суммарного изотопа задаётся список нуклидов, входящих в суммарный изотоп. Нуклиды, входящие в суммарный изотоп, обрабатываются программой иначе, чем нуклиды, не входящие в него. В суммарный изотоп обычно помещаются нуклиды, не оказывающие значительного влияния на результаты моделирования конкретной системы.

При отсутствии данного раздела суммарный изотоп не содержит нуклидов.

Общий вид формата задания раздела следующий:

*[SI list]*

*SI* - имя заголовка раздела.

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

*[SINOT list]*

*SINOT* - имя заголовка раздела.

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

*[SIDEN value]*

*SIDEN* - имя заголовка раздела.

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

Список действует с момента его объявления. Можно использовать любое количество карт описания суммарного изотопа. Это позволяет задавать состав суммарного изотопа

независимо для каждого материала. Можно использовать карту 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.

Карта SIDEN активна всегда, независимо от наличия или отсутствия карт SI и SINOT.

## 8.6 УПРАВЛЯЮЩИЕ ПАРАМЕТРЫ ФИЗИЧЕСКОГО МОДУЛЯ

В этом разделе задаются значения управляющих параметров, определяющих режим работы физического модуля.

Общий вид формата задания раздела следующий:

```
[NEUT   valneut]
[EGRC   valegrc1,valegrc2,valegrc3]
[PHOT   valphot]
[WPHO   valwpho]
[IWPHN  valiwphn]
[EGPH   valegph1,valegph2]
[DELN   valdeln]
[PSIN   valpsin]
[PSGR   valpsgr]
[MATFIL list]
[MATPRN]
[SIPRN]
[DEFPRN]
```

*NEUT* – имя управляющего параметра, определяющего моделирование взаимодействия нейтронов со средой.

При *valneut* равном 1 - взаимодействие нейтронов со средой моделируется, при *valneut* равном 0 - не моделируется. Константа целого типа. По умолчанию *valneut* равно 1 (нейтроны рассматриваются).

*EGRC* – имя управляющего массива параметров, определяющего энергетические границы областей работы нейтронных подмодулей составного физического модуля.

*valegrc1* - нижняя энергетическая граница, эВ, области работы нейтронного подмодуля FARION; константа вещественного типа;

*valegrc2* - нижняя энергетическая граница, эВ, области работы нейтронного подмодуля ФИМБРОЭН; константа вещественного типа;

*valegrc3* - нижняя энергетическая граница, эВ, области работы нейтронного подмодуля МОФИТТГ (многогрупповое транспортное приближение) или ФИМТОЭН (непрерывное слежение за энергией нейтрона с использованием поточечного представления сечений), моделирующих взаимодействие нейтронов со средой в области термализации; константа вещественного типа.

По умолчанию: *valegrc1* равен  $20 \cdot 10^6$  эВ, *valegrc2* равен 1 эВ, *valegrc3* равен 0. Эти значения указывают границы работы для подмодуля ФИМБРОЭН от  $20 \cdot 10^6$  эВ (фактически от  $10.5 \cdot 10^6$  эВ в связи с используемой версией библиотеки VNAV/MCU) до 1 эВ, а для подмодулей МОФИТТГ или ФИМТОЭН 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* менее 20 МэВ.

В компоновках с подмодулем МОФИТТГ граница сшивки *valegrc2* не должна превосходить 1 эВ. Рекомендованное значение: 1 эВ.

В компоновках с подмодулем ФИМТОЭН граница сшивки *valegrc2* должна быть не больше самой низкой границы разрешённых резонансов имеющихся в задаче изотопов. Рекомендованные значения: 1.0; 2.15 и 4.65 эВ.

*RHOT* – имя управляющего параметра, определяющего моделирование взаимодействия фотонов со средой.

Значение *valphot* может принимать значения 0, 1 и 2. По умолчанию *valphot* равно 0 (фотоны не рассматриваются). Константа целого типа.

При *valphot* больше 0 моделируется взаимодействие фотонов со средой. При *valphot* равном 1 не моделируется релаксация атомов (при фотоэффекте и некогерентном рассеянии). При *valphot* равном 2 релаксация атомов моделируется (возможно только при использовании подмодуля ГАММА).

*WRHO* – имя управляющего параметра, определяющего величину, обратную вероятности генерации фотона при реакции (n,γ). Задаётся при наличии управляющего параметра *RHOT* со значением *valphot* больше 0.

*valwpho* - величина, обратная вероятности генерации фотона при реакции (n,γ). По умолчанию *valwpho* равно 10. Константа вещественного типа.

*IWRHN* – имя управляющего параметра, определяющего количество генерируемых фотонейтронов (только для модуля РНТ). Задаётся при наличии управляющего параметра *RHOT* со значением *valphot* больше 0.

*valiwphn* - величина, определяющая количество генерируемых фотонейтронов. При ненулевом значении параметра генерируемые фотонейтроны получают вес равный  $1/valiwphn$ . По умолчанию *valiwphn* равно 0, т. е. фотонейтроны не генерируются. Константа целого типа.

*EGPH* – имя управляющего параметра, определяющего верхнюю и нижнюю границы библиотеки задачи для расчёта взаимодействия фотонов со средой.

Значения *valegph1* и *valegph2* по умолчанию составляют  $14 \cdot 10^6$  и 10000 эВ, соответственно.

При указании в части исходных данных для управления шагом расчёта границы энергии фотонов, ниже которой их траектории прерываются, нельзя использовать значение менее *valegph2*.

*DELN* – имя управляющего параметра, определяющего режим расчёта энергий нейтронов деления.

При *valdeln* равном 0 не производится разделения нейтронов на мгновенные и запаздывающие, а энергия всех нейтронов разыгрывается по спектру мгновенных нейтронов. При *valdeln* равном 1 такое разделение производится, и энергия запаздывающих нейтронов разыгрывается по спектру запаздывающих нейтронов. Константа целого типа. По умолчанию *valdeln* равно 0.

*PSIN* – имя управляющего параметра вывода на печать микроскопических сечений нуклидов.

При *valpsin* равном 1 печать осуществляется в файл с расширением «*lst*», при *valpsin* равном 0 печать отсутствует. Константа целого типа. По умолчанию *valpsin* равно 0.

*PSGR* – имя управляющего параметра вывода на печать макроскопических сечений материалов.

При *valpsgr* равном 1 печать осуществляется в файл с расширением «*lst*», при *valpsgr* равном 0 печать отсутствует. Константа целого типа. По умолчанию *valpsgr* равно 0.

*MATFIL* – имя управляющего параметра вывода в файл с расширением MAT материального состава задачи с учетом списков суммарного изотопа и данных в файлах с расширениями вида PDCn в виде исходных данных для физического модуля.

*list* – список групп (параметр GROUP) вывод которых в файл с расширением MAT не осуществляется. Массив вещественного типа. По умолчанию список пуст, т. е. выводятся все материалы.

*MATPRN* – по окончании ввода для каждого материала в формате исходных данных для физического модуля печатается его материальный состав и характеристики.

*SIPRN* – для каждого материала печатаются сечения полученного суммарного изотопа.

*DEFPRN* – печатаются параметры нуклидов с учетом переопределений, сделанных пользователем с помощью карт DEF.

## 8.7 ОКОНЧАНИЕ ВВОДА ДАННЫХ ДЛЯ ФИЗИЧЕСКОГО МОДУЛЯ

Фрагмент исходных данных для физического модуля должен заканчиваться обязательной картой FINISH.

*FINISH [commentary]*

*FINISH* - признак окончания ввода данных к физическому модулю.

*commentary* - текст произвольного содержания.

## 8.8 КОРРЕКЦИЯ ДАННЫХ С ПОМОЩЬЮ ФАЙЛОВ PDC

При работе с исходными данными для физического модуля возможно выполнить коррекцию некоторых данных с помощью внешних текстовых файлов с расширениями вида PDCn (PIN Data Correction), где n может принимать значения от 1 до 9 или отсутствовать (соответствует n равному 0).

Обработка 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.

Файлы вида PDCn записываются по следующим правилам.

При наличии символа «\*» или пробела в первой позиции строка считается комментарной и не обрабатывается. Пустые строки не обрабатываются.

Первая некомментарная строка (в настоящее время не используется) содержит указательное слово в позициях с 1 по 4. Далее без промежуточных пробелов по 11 позиций на число записываются: количество материалов в файле, максимальное количество нуклидов в одном материале.

Далее следуют разделы материалов. Для ускорения обработки данных рекомендуется записывать материалы без пропусков и в порядке возрастания. Материалы, не подлежащие изменению, можно при этом задавать пустыми.

Описание материала начинается строкой со словом «MATR» в позициях с 1 по 4. Далее без промежуточных пробелов по 11 позиций на число записываются: номер материала, количество изотопов в материале (в настоящее время не используется, но должно присутствовать), температура материала. Если температура материала не должна изменяться, то задается любое отрицательное значение.

Далее следуют строки вида

*<имя изотопа в кодировке MSU> <концентрация>*

при этом имя изотопа записывается с первой позиции, а концентрация с 6 по 16.

Описание материала заканчивается словом «stop» или «STOP» в позициях с 1 по 4.

## 8.9 ПРИМЕРЫ ЗАДАНИЯ ИСХОДНЫХ ДАННЫХ К ФИЗИЧЕСКОМУ МОДУЛЮ

### Пример 1

```
TEMPR 300.  
MATR 1  
* material 1  
U235 1.10E-03
```



```

H 0.0001 MODS = G
O 2.3E-06
MATR 2,T = 350.
* material 2
U238 1.1E-04
END
PSIN 1
PSGR 1
EGRC 8.0E+06,1.,0.
FINISH Physical module: end of the input data

```

В примере рассматривается система, состоящая из двух материалов, которым первоначально присваивается температура 300 К.

Первый материал состоит из трёх нуклидов: U-235, H и O. Для каждого нуклида задана его ядерная плотность в материале. Кроме того, для водорода определена модель идеального газа расчёта сечений в тепловой области энергий, все остальные значения параметров нуклидов приняты по умолчанию.

Второй материал состоит из одного нуклида U-238 с ядерной плотностью 0.00011 ядро/(барн·см). Для него переопределяется значение температуры материала с 300 К на 350 К.

Управляющими параметрами PSIN и PSGR включается печать нейтронных сечений материалов и нуклидов в файл с расширением «lst».

Управляющим параметром EGRC задаются нижние значения энергетических границ областей работы нейтронных подмодулей составного физического модуля. Подмодуль FARION будет использоваться в области энергий от  $20 \cdot 10^6$  до  $8 \cdot 10^6$  эВ, подмодуль ФИМБРОЭН – от  $8 \cdot 10^6$  до 1 эВ, подмодуль МОФИТТГ или ФИМТОЭН (в зависимости от компоновки программы) – в энергетической области от 1 до 0 эВ.

### Пример 2

```

TEMPR 300.0
MATR 1
U235 1.10E-03
H 1.00E-04 ,MODS = G
O 2.30E-06
MATR 2 ,350.
U238 1.10E-04
END
FINISH

```

Материальный состав во втором примере такой же, как в примере 1. Изменена запись значений параметров. Исключены строки–комментарии, начинающиеся символами «\*», и комментарии в строке с именем FINISH.

Поскольку управляющий параметр EGRC отсутствует, то по умолчанию в диапазоне от  $10.5 \cdot 10^6$  до 1 эВ работает подмодуль ФИМБРОЭН, а ниже 1 эВ – подмодуль МОФИТТГ или ФИМТОЭН (в зависимости от компоновки программы).

### Пример 3

```

TEMPR 300.
MATR 1
* material 1
U235 1.1E-03, DTEM = 7.
H 0.0001, MODS = G
O 2.3E-06, ACE = E65
* material 2
MATR 2,350.
U238 1.1E-04

```

```
END
DELN 1
FINISH Physical module: end of the input data
```

В первом материале для нуклида U-235 изменяется значение допуска по температуре *dtem*, определённое по умолчанию, на значение 7.0К.

Для расчёта сечений подмодулем FARION будут использоваться следующие файлы библиотеки ACE: для первого (U235), второго (H) и четвёртого (U238) нуклидов – определённые по умолчанию, а для третьего (O) – O.E65.

Поскольку управляющий параметр DELN задан равным 1, то при моделировании энергия мгновенных нейтронов деления разыгрывается по спектру мгновенных нейтронов, а энергия запаздывающих нейтронов разыгрывается по спектру запаздывающих нейтронов.

#### Пример 4

```
PIN 1 0
SINOT U234 U235 U236 U238 PU39 PU40 PU41 PU42
XE35 SM49 H O AL SI FE NI ZR NB HF
SIDEN 1.E-11
MATR 1
U235 0.0008255
U238 0.022105
O 0.045861
MATR 2
ZR 0.04273
NB 0.000432 MODS = T BLOCK = 4
HF 6.6E-6
MATR 3
H 0.066714
O 0.033357
MATR 4
AL 1.E-6
FINISH
```

При расчёте выгорания данной системы после первого шага делящийся материал MATR 1 будет содержать около 290 изотопов. Благодаря наличию карты SINOT, в реальном расчёте следующего состояния в этом материале будут использоваться только те изотопы, которые в этой карте перечислены при условии, что их концентрация больше, либо равна  $10^{-11}$  яд/см<sup>3</sup>. Все изотопы, не входящие в этот список или входящие в него, но с концентрацией менее  $10^{-11}$  яд/см<sup>3</sup>, будут отнесены к суммарному изотопу, и взаимодействие частиц с ними будет моделироваться по упрощённой схеме.

## **9 ИСХОДНЫЕ ДАННЫЕ ДЛЯ ГЕОМЕТРИЧЕСКОГО МОДУЛЯ**

### **9.1 ВВЕДЕНИЕ**

В данном документе описываются правила задания исходных данных для геометрического модуля NCG с компилятором с языка NCGSIM, а также специального алгоритма учёта микроструктуры топлива.

Полное руководство по использованию языка NCGSIM состоит из двух разделов: описание простой системы и сложные системы. Первый раздел содержит законченную инструкцию для пользователей, не нуждающихся в сложных конструкциях языка NCGSIM. Кроме того, он служит введением, облегчающим понимание информации второго раздела.

В дальнейшем, необязательные секции и параметры файла исходных данных приводятся в квадратных скобках «[...]». Фигурные скобки обозначают альтернативные варианты.

### 9.1.1 Основные понятия

При описании геометрии в модуле NCG используется 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.

В модуле NCG используется комбинаторный способ задания геометрии с помощью тел.

Простейшими геометрическими областями являются так называемые *тела*: шары, параллелепипеды, цилиндры и так далее. Каждое тело имеет тип и параметры, описывающие конкретные размеры тела и его положение в пространстве. Например, для шара параметрами являются положение центра и радиус.

При описании конкретной системы используется конечное число тел, к которым применяются три операции теории множеств: дополнение, пересечение и объединение. Это позволяет строить достаточно сложные формы.

Конечная область пространства, в которой производится расчёт, называется *контейнером системы*. Она разбивается на конечное число областей, называемых *геометрическими зонами* или просто зонами. Геометрические зоны строятся из тел с помощью вышеупомянутых операций.

Любая точка системы входит *только* в одну геометрическую зону (исключением являются точки границ зон). Геометрическая зона считается гомогенной, то есть, однородной по своим свойствам. В частности, это означает, что любая зона состоит из одного материала.

Каждой геометрической зоне присваивается номер соответствующего материала, называемый *материальным номером*. Под *материальной зоной* понимается совокупность всех точек с одним материалом. Очевидно, что материальная зона составляется как набор геометрических зон, взятых целиком. Отдельные части материальной зоны могут не иметь общих границ.

Для работы регистрационного модуля контейнер системы разбивается на два независимых набора областей, называемых *регистрационными зонами* и *регистрационными объектами*. Любые регистрационная зона или регистрационный объект составлены из одной или нескольких геометрических зон, и каждая геометрическая зона входит в одну регистрационную зону, а также в один регистрационный объект. Номер регистрационной зоны, к которой принадлежит данная геометрическая зона, называется её *регистрационным номером*, а номер соответствующего регистрационного объекта - *объектным номером*.

На поверхности контейнера должны быть заданы граничные условия, то есть определена реакция геометрического модуля на вылет частицы из контейнера. Считается, что вся внешняя граница может быть разбита на несколько поверхностей, и в пределах одной поверхности установлено одно граничное условие. Типичными граничными условиями являются: трансляция, зеркальное отражение, белое отражение, утечка и так далее.

Расчёт всегда осуществляется в конечной области пространства. При необходимости выполнить расчёт в бесконечной области, вычисления ведутся в конечной части области, а бесконечность описывается с помощью граничных условий зеркальной симметрии или трансляции (переноса).

Таким образом, описать геометрию системы – это значит задать:

- все геометрические зоны, в совокупности составляющие всю систему;
- соответствие материальных, регистрационных и объектных номеров геометрическим зонам;
- граничные условия на внешней поверхности системы.

## 9.1.2 Форматы и константы

Описание данных для задания геометрии на языке NCGSIM представляет собой последовательность строк текстового файла, заканчивающуюся строкой с символами FINISH в первых 6 позициях.

Ввод и обработка фрагмента с данными для модуля производится программными средствами стандартного ввода.

## 9.2 ОПИСАНИЕ ПРОСТОЙ СИСТЕМЫ

Описание простой системы состоит из последовательности предложений. Некоторые предложения объединены в секции. Порядок секций строго фиксирован и дан в следующей схеме:

```
<заголовок>  
<секция граничных условий>  
<секция тел>  
<секция зон>  
[<таблица объёмов регистрационных зон>]  
FINISH
```

### 9.2.1 Заголовок

Заголовок состоит из одного предложения:

```
HEAD [<целое число 1>[<целое число 2>[<целое число 3>]]]
```

Обязательной в этом предложении является только метка HEAD.

<целое число 1> принимает значения 0,1,2,3,4 (значение по умолчанию есть 1):

- 0 - отсутствие печати на вводе;
- 1 - печать вводимых строк;
- 2 - то же, что 1 и остановка после ввода;
- 3- то же, что 1 и печатаются таблицы тел и зон в памяти;
- 4 - то же, что 3 и остановка после ввода.

<целое число 2> - отвечает за управление трассировкой (значение по умолчанию есть 0), обозначим его Nt:

- Nt равно 0 - отсутствие трассировки;
- Nt больше 0 - выдача первых Nt точек трассировки, затем трассировка выключается;
- Nt равно -1 - трассировка на все время моделирования;
- Nt меньше -1 - выдача |Nt| первых точек трассировки, затем происходит остановка задачи.

<целое число 3> - задаёт количество дополнительных позиций, отводимых под списки поиска (значение по умолчанию равно 400). Если при счёте возникает диагностика

```
NO PLACE FOR LIST
```

то для ускорения счёта можно увеличить это значение. На результатах расчёта это изменение не отражается.

### 9.2.2 Секция граничных условий

Секция граничных условий состоит в задании контейнера, в котором определены тела системы, и условий на поверхностях этого контейнера. Есть два способа определения контейнера.

Первый способ состоит в описании плоскостей симметрии, каждая из которых задана предложением вида:

```
MIR P Q
```

где **P** - вектор,  
**Q** - скаляр.

Компоненты **P** и **Q** - явно заданные константы или имена констант. Разделителями служат пробелы или запятые. Плоскость симметрии имеет уравнение:  $(\mathbf{P}, \mathbf{x}) + Q = 0$ . Вектор **P** направлен внутрь контейнера. Плоскости симметрии задают ограниченный многоугольник, который и является контейнером системы.

Строка с меткой END должна завершать секцию граничных условий.

#### Пример

Контейнер - это конечная призма, которая пересекается плоскостью OXY по треугольнику с вершинами (0,0,0), (1,0,0), (0,1,0), и имеет основания  $\{z = 0\}$  и  $\{z = 1\}$ .

```
MIR 1, 0, 0 0
MIR 0, 1, 0 0
MIR -1, -1, 0 1
MIR 0, 0, 1 0
MIR 0, 0, -1 1
END
```

Второй способ - контейнером служит первое тело в секции описания тел. В качестве контейнера можно использовать тела: SPH, RPP, HEX, RCZ, BOX (см. описание секции тел).

Контейнер задаётся одним предложением:

```
CONT <граничные условия> [S<угол>[<угол>]] PRS<угол>[<угол>]]
```

<список граничных условий> состоит из столькох фрагментов, сколько есть граничных поверхностей тела, служащего контейнером.

Каждый фрагмент имеет структуру: **B** или  $W[<вероятность>]$  или  $M[(<вероятность>)]$  или  $C[<вероятность>]$  или **T**.

<вероятность> - действительное число из интервала (0,1), задающее вероятность отражения. В противном случае происходит поглощение. Если вероятность не задана, всегда происходит отражение.

**B** - чёрная граница (все вылетающие частицы поглощаются);

**W** - белое отражение, соответствующее гипотезе, что поток на границе изотропен по направлениям, а градиент потока перпендикулярен нормали.

Рассмотрим сферическую систему координат  $(t, f)$  с полюсом, соответствующим внутренней нормали, где  $t$  - полярный угол, а  $f$  - азимутальный. Тогда для отражённых частиц  $\cos^2(t)$  распределён равномерно на отрезке  $[0,1]$ , а  $f$  - на единичной окружности. В скобках может быть указана вероятность отражения, если часть частиц поглощается.

**M** - зеркальное отражение, в скобках может быть указана вероятность отражения, если часть частиц поглощается.

**C** - белое цилиндрическое отражение, допустимое на боковой поверхности вертикального цилиндра,  $z$  - компонента направления полёта не меняется,  $x$  и  $y$  компоненты соответствуют изотропному в плоскости XY потоку с градиентом, перпендикулярным внутренней нормали. Для отражённых частиц синус угла между направлением и внутренней нормалью равномерно распределён на  $[-1,1]$ . В скобках может быть указана вероятность отражения, если часть частиц поглощается.

**T** - трансляционная симметрия. Такая симметрия может быть задана на всех гранях тел RPP, BOX, HEX и на верхней и нижней гранях RCZ. Эта симметрия должна быть задана согласованно на противоположных гранях. Трансляционная симметрия допускается на

боковой поверхности RCZ, если ось цилиндра совпадает с осью OZ, и на поверхности шара, если центр шара расположен в начале координат. В первом случае граничная точка (a,b,c) переходит в точку (-a,-b,c), а во втором - в точку (-a,-b,-c).

Фрагменты задаются в порядке нумерации граней, приведённом в секции описания тел.

$S<угол>$  определяет вертикальный угол зеркальной симметрии от 0 (плоскость OXZ) до положения, определяемого параметром  $<угол>$  в градусах. Допустимые значения этого параметра: 180, 90, 60, 45, 30. При наличии угла симметрии трансляция может быть задана только на нижней и верхней гранях тел типа RPP, HEX или RCZ (см. ниже в секции тел).

Дополнительный угол означает поворот угла симметрии на заданное число градусов против часовой стрелки вокруг вертикальной оси.

#### Пример

Задаётся угол симметрии, соответствующий диапазону углов [45,135], то есть прямой угол с вертикальной биссектрисой:

```
CONT T T C S90 45
```

$PRS<угол>$  - соответствует предыдущей, за исключением того, что плоскости симметрии расположены симметрично относительно OXZ и образуют с осью OX угол, определяемый параметром  $<угол>$ . Смысл дополнительного угла тот же, что и в предыдущем случае.

#### Пример

Первое тело есть RCZ:

```
CONT T T W(0.5) PRS60
```

Два способа определения контейнера имеют важное различие. Плоскости симметрии (как заданные предложениями MIR, так и заданные конструкцией S или PRS) не входят явно в определение зон. При проведении траектории учитывается возможность пересечения этих плоскостей с любой зоной. Пользователь может описывать геометрию, не заботясь об этих границах.

Тело-контейнер - это одно из тел, из которых строится комбинаторное описание зон. Пересечения других тел с контейнером должно явно указываться.

#### Пример

Пусть контейнером служит параллелепипед RP, и задан угол симметрии S90. Задан шар SP, который пересекается как с телом RP, так и с плоскостью  $\{y = 0\}$ . Нужно описать зону пересечения шара с контейнером (заштрихована на рисунке 9.2.2.1).

Так как ограничение плоскостью симметрии будет автоматически учтено, то правильное описание будет

```
RP SP
```

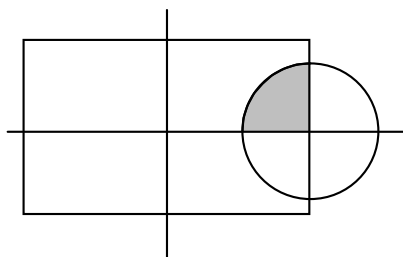


Рисунок 9.2.2.1 – Пример задания зоны пересечения шара с контейнером

В некоторых случаях требуется автоматическая проверка выхода частицы за пределы тела контейнера для любой геометрической зоны. Это требует дополнительных

затрат времени счёта, но часто сильно сокращает подготовку данных. Для этого перед картой CONT в файле исходных данных должна быть карта

CNTAND

На основании этого признака к любой зоне глобальной геометрии, включая и элементы решёток, добавляется операция пересечения с телом-контейнером, что обеспечивает контроль выхода за его пределы.

Начиная с 2010 года, во всех версиях программы карта CNTAND принимает параметр: 1 – включено, 0 – выключено. По умолчанию (при отсутствии карты или параметра) принудительное пересечение всех тел с контейнером включено.

### 9.2.3 Секция тел

В языке NCGSIM тело кодируется с помощью трёх- или четырёхсимвольного ключа, обозначающего тип тела, и набора числовых параметров. Параметры тела должны задаваться в том порядке, в котором они перечисляются в описании.

После ключа в скобках дано однобуквенное обозначение тела, которое используется только при печати таблицы тел.

Если тело используется как контейнер, то на его границах задаются граничные условия. Граница тела делится на составляющие её поверхности, нумеруемые целыми числами 1, 2, .... Для каждого типа тела дано разбиение границы на поверхности и их нумерация.

Каждое тело описывается одним предложением вида:

<метка> <имя тела> <параметры>

<метка> определяет тип одного из перечисленных ниже тел в виде трёхсимвольного ключа.

<имя тела> - служит для ссылок на него. Это есть идентификатор из не более чем 6 латинских букв и цифр, начинающийся с буквы. Вместо этого идентификатора может стоять символ '\*'. В этом случае тело получает имя N<порядковый номер тела>. В качестве имени тела нельзя использовать идентификаторы U и T, которые имеют специальный смысл.

<параметры> - константы, включая числа или выражения по количеству параметров для данного типа тела. Разделителями служат пробелы или запятые (возможно, вместе с пробелами). Поэтому в выражениях не должно быть внутренних пробелов. В описании тела ARB используется символ '/', обозначающий начало описания граней. Этот символ должен быть окружён пробелами.

Раздел заканчивается специальной строкой, содержащей символы END в качестве метки предложения.

Далее приводятся описания тел, с указанием их типа и описанием параметров.

#### 9.2.3.1 Шар, SPH (S)

На рисунке 9.2.3.1.1 приведен шар.

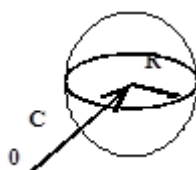


Рисунок 9.2.3.1.1 – Шар

Параметры C, R, где C - радиус-вектор центра, а R - радиус шара. Вся граница представляет одну поверхность.

### 9.2.3.2 Правильный круговой цилиндр, RCC (C)

На рисунке 9.2.3.2.1 приведен правильный круговой цилиндр.

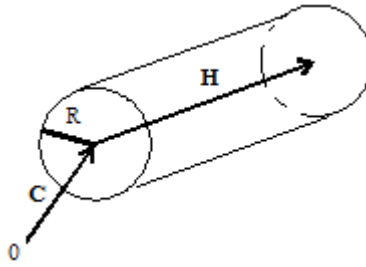


Рисунок 9.2.3.2.1 – Правильный круговой цилиндр

Параметры  $C, H, R$ .  $C$  - радиус-вектор центра одного нижнего основания.  $H$  - вектор, соединяющий центр нижнего основания с центром другого основания.  $R$  - радиус цилиндра.

Поверхности - нижнее основание (номер 1), верхнее основание (2), боковая поверхность (3).

### 9.2.3.3 Эллипсоид вращения, ELL (E)

Имеется два способа задания тела.

Первый способ приведен на рисунке 9.2.3.3.1.

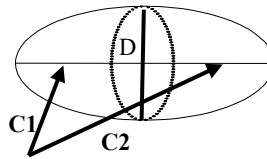


Рисунок 9.2.3.3.1 – Первый способ задания эллипсоида вращения

Параметры  $C_1, C_2, D$ , где  $C_1, C_2$  - радиус-векторы фокусов,  $D$  - длина малой полуоси. Граница состоит из одной поверхности.

Второй способ приведен на рисунке 9.2.3.3.2.

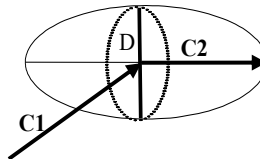


Рисунок 9.2.3.3.2 – Второй способ задания эллипсоида вращения

Параметры  $C_1, C_2, -D$ , где  $C_1$  – радиус-вектор центра эллипсоида,  $C_2$  – вектор полуоси, являющейся осью вращения,  $D$  - длина другой полуоси. Граница состоит из одной поверхности.

### 9.2.3.4 Произвольный параллелепипед, BOX (B)

На рисунке 9.2.3.4.1 приведен произвольный параллелепипед.

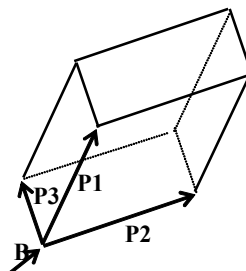


Рисунок 9.2.3.4.1 – Произвольный параллелепипед



Параметры  $\mathbf{V}$ ,  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$ , где  $\mathbf{V}$  - радиус-вектор вершины параллелепипеда,  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$  - векторы, совпадающие по направлениям и длинам с тремя рёбрами, идущими из  $\mathbf{V}$ .

Поверхностями служат 6 граней тела, перечисляемые по порядку номеров: грань, содержащая  $\mathbf{P}_1$  и  $\mathbf{P}_2$ ; противоположная грань, грань с  $\mathbf{P}_2$  и  $\mathbf{P}_3$ ; грань с  $\mathbf{P}_1$  и  $\mathbf{P}_3$ ; грань, противоположная третьей; грань, противоположная четвёртой.

### 9.2.3.5 Призма с треугольником в качестве основания, WED (W)

На рисунке 9.2.3.5.1 приведена призма с треугольником в качестве основания.

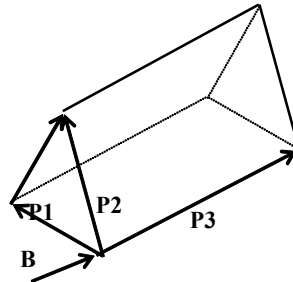


Рисунок 9.2.3.5.1 – Призма с треугольником в качестве основания

Параметры  $\mathbf{V}$ ,  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$ , где  $\mathbf{V}$  - радиус-вектор вершины.  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$ - ребра, исходящие из  $\mathbf{V}$ ,  $\mathbf{P}_1$  и  $\mathbf{P}_2$  лежат в основании, а  $\mathbf{P}_3$  параллельно образующей.

Поверхностями служат 5 граней тела, перечисляемых по порядку номеров: грань, содержащая  $\mathbf{P}_1$  и  $\mathbf{P}_2$ ; противоположная грань, грань, содержащая  $\mathbf{P}_1$  и  $\mathbf{P}_3$ ; грань параллельная  $\mathbf{P}_3$ ; грань, содержащая  $\mathbf{P}_2$  и  $\mathbf{P}_3$ .

### 9.2.3.6 Параллелепипед с рёбрами, параллельными осям координат, RPP (P)

На рисунке 9.2.3.6.1 приведен параллелепипед с рёбрами, параллельными осям координат.

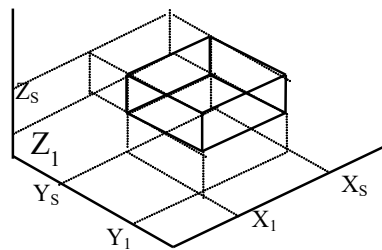


Рисунок 9.2.3.6.1 – Параллелепипед с рёбрами, параллельными осям координат

Параметры  $X_1$ ,  $X_s$ ,  $Y_1$ ,  $Y_s$ ,  $Z_1$ ,  $Z_s$  ( $X_1 < X_s$ ,  $Y_1 < Y_s$ ,  $Z_1 < Z_s$ ); тело задано неравенствами  $X_1 < x < X_s$ ,  $Y_1 < y < Y_s$ ,  $Z_1 < z < Z_s$ .

Поверхностями служат 6 граней тела, перечисляемые по порядку: ( $z = Z_1$ ), ( $z = Z_s$ ), ( $x = X_1$ ), ( $y = Y_1$ ), ( $X = X_s$ ), ( $Y = Y_s$ ).

### 9.2.3.7 Правильная шестиугольная призма с образующей, параллельной оси OZ, HEX (H)

На рисунке 9.2.3.7.1 приведена правильная шестиугольная призма с образующей, параллельной оси OZ.

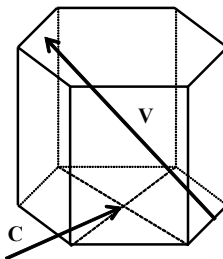


Рисунок 9.2.3.7.1 – Правильная шестиугольная призма с образующей, параллельной оси  $OZ$

Параметры  $C, V$ , где  $C$  - радиус-вектор центра нижнего основания;  $V$  - вектор, соединяющий середину ребра нижнего основания с серединой противоположного ребра верхнего основания (3-я координата  $V$  есть высота призмы; первые 2 координаты  $V$  задают размер шестиугольника "под ключ" и его поворот в плоскости  $X, Y$ ).

Поверхностями служат 8 граней тела, перечисляемые по порядку: нижнее основание, верхнее основание, боковая грань, из которой выходит вектор  $V$ ; далее идут последовательно грани, полученные из третьей поворотами против часовой стрелки вокруг оси  $OZ$  соответственно на 60, 120, 180, 240 и 300 градусов.

Имеется ещё два альтернативных способа задания шестигранника.

### 9.2.3.8 Первая альтернативная форма $HEXX$

Параметры  $C, H, D, [f]$ . Последний параметр не обязателен. Его умалчиваемое значение 0.  $C$  - центр нижнего основания;  $H$  - высота;  $D$  - размер под ключ;  $f$  - угол поворота относительно оси  $OX$  в градусах, то есть параметрам  $H, D, f$  соответствует вектор  $V$  с координатами  $(D \cdot \cos(f), D \cdot \sin(f), H)$ .

Поверхности определяются так же, как в случае  $HEX$ .

### 9.2.3.9 Вторая альтернативная форма $HEXY$

Параметры те же, что и в предыдущем случае, но тело повёрнуто на 90 градусов против часовой стрелки. Угол  $f$  отсчитывается от оси  $OY$ . Вектор  $V$  имеет координаты  $(-D \cdot \sin(f), D \cdot \cos(f), H)$ .

### 9.2.3.10 Прямой круговой цилиндр с образующей, параллельной оси $OZ, RCZ(Z)$

На рисунке 9.2.3.10.1 приведен прямой круговой цилиндр с образующей, параллельной оси  $OZ$ .

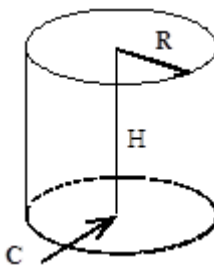


Рисунок 9.2.3.10.1 – Прямой круговой цилиндр с образующей, параллельной оси  $OZ$

Параметры  $C, H, R$ , где  $C$  - радиус-вектор центра нижнего основания,  $H > 0$  - высота, а  $R$  - радиус цилиндра.

Поверхности - нижнее основание, верхнее основание, боковая поверхность.

### 9.2.3.11 Бесконечный цилиндр, ориентированный по оси $X, Y$ или $Z$ , соответственно, $UCX(x), UCY(y), UCZ(z)$

Параметры для  $UCX$  есть  $Y, Z, R$ , где  $Y, Z$  - координаты пересечения оси с плоскостью  $OYZ$ , а  $R$  - радиус.

Параметры для UCSY есть X, Z, R, а для UCZ есть X, Y, R, где первые два параметра задают координаты оси цилиндра. Поверхность одна.

### 9.2.3.12 Произвольно ориентируемое полупространство, PLG(d)

Параметры:  $\mathbf{P}$ ,  $Q$ , где  $\mathbf{P}$  - вектор,  $Q$  - скаляр. Полупространство задано неравенством:  $(\mathbf{P}, \mathbf{x}) \geq Q$ . Поверхность одна.

### 9.2.3.13 Полупространства, заданные плоскостями, перпендикулярными осям координат, PLX(A), PLY(B), PLZ(C)

Параметром PLX служит скаляр  $X_0$ . Полупространство определено неравенством  $X \geq X_0$ . Аналогично, единственный параметр PLY - скаляр  $Y_0$ , PLZ -  $Z_0$ . Поверхность одна.

### 9.2.3.14 Слой между двумя параллельными плоскостями, SLA(V)

На рисунке 9.2.3.14.1 приведен слой между двумя параллельными плоскостями.

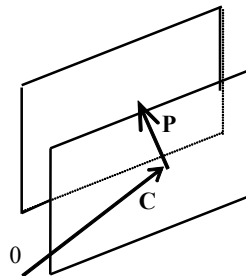


Рисунок 9.2.3.14.1 – Слой между двумя параллельными плоскостями

Параметры:  $\mathbf{C}$ ,  $\mathbf{P}$ .  $\mathbf{C}$  - радиус-вектор произвольной точки одной из плоскостей.  $\mathbf{P}$  - вектор, соединяющий  $\mathbf{C}$  с близлежащей точкой второй плоскости.  $\mathbf{P}$  перпендикулярен плоскостям.

Поверхности - плоскость, откуда выходит  $\mathbf{P}$ ; противоположная плоскость.

### 9.2.3.15 Слой может также быть задан как тело SLB (L)

Параметры:  $\mathbf{V}$ ,  $A$ ,  $B$ .  $\mathbf{V}$  - вектор, перпендикулярный слою.  $A < B$ .  $A$ ,  $B$  - расстояния от начала координат до плоскостей слоя, взятые со знаком так, чтобы направление перпендикуляра к слою задавал  $\mathbf{V}$ . Таким образом, если  $\mathbf{V}$  имеет единичную длину, слой определен неравенством  $A < (\mathbf{V}, \mathbf{x}) < B$ .

Поверхности: плоскости  $\{(\mathbf{V}, \mathbf{X}) = A|\mathbf{V}|\}$  и  $\{(\mathbf{V}, \mathbf{X}) = B|\mathbf{V}|\}$

### 9.2.3.16 Правильный эллиптический цилиндр, REC (O)

На рисунке 9.2.3.16.1 приведен правильный эллиптический цилиндр.

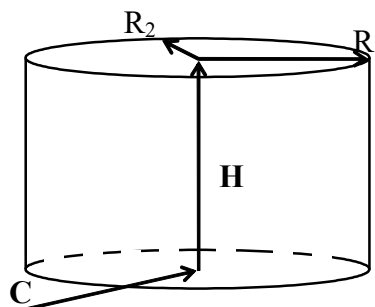


Рисунок 9.2.3.16.1 – Правильный эллиптический цилиндр.

Параметры:  $\mathbf{C}$ ,  $\mathbf{H}$ ,  $\mathbf{R}_1$ ,  $\mathbf{R}_2$ , где  $\mathbf{H}$ ,  $\mathbf{R}_1$ ,  $\mathbf{R}_2$  попарно ортогональны, и  $|\mathbf{R}_1| > |\mathbf{R}_2|$ .

$\mathbf{C}$  - радиус-вектор центра нижнего основания.  $\mathbf{H}$  - вектор высоты, определенным так же, как для RCC.  $\mathbf{R}_1$ ,  $\mathbf{R}_2$  - большая и малая полуоси эллипса.

Поверхности определены так же, как для RCC.

### 9.2.3.17 Правильный круговой усечённый конус, TRC (T)

На рисунке 9.2.3.17.1 приведен правильный круговой усечённый конус.

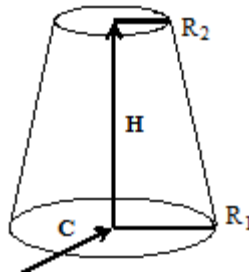


Рисунок 9.2.3.17.1 – Правильный круговой усечённый конус

Параметры:  $C$ ,  $H$ ,  $R_1$ ,  $R_2$ , где  $C$  - радиус-вектор центра нижнего основания;  $H$  - вектор высоты противоположного основания;  $R_1$  - радиус нижнего основания;  $R_2$  - радиус верхнего основания.

Поверхности определены так же, как для RCC.

### 9.2.3.18 Произвольный выпуклый многогранник, имеющий не более 6 граней, ARB (N)

На рисунке 9.2.3.18.1 приведен Произвольный выпуклый многогранник, имеющий не более 6 граней.

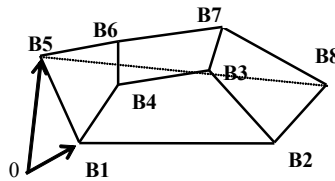


Рисунок 9.2.3.18.1 – Произвольный выпуклый многогранник, имеющий не более 6 граней

Параметры:  $B_1, B_2, B_3, \dots, B_m / M_1, M_2, M_3, \dots, M_n$ .  $B_k$  - радиус-вектор  $k$ -ой вершины многогранника, очевидно, что  $4 \leq m \leq 8$ .  $M_j$  - целое число, описывающее  $j$ -ю грань,  $4 \leq n \leq 6$ . Цифры числа  $M_j$  есть номера вершин, принадлежащих  $j$ -ой грани, которые перечисляются по или против часовой стрелки. Поверхностями служат грани. Первой служит грань, соответствующая  $M_1$ ; второй -  $M_2$  т.д. При описании между последней координатой  $B_m$  и  $M_1$  должен стоять символ '/'.

Пример (египетская пирамида, см. рисунок 9.2.3.18.2)

ARB -1, -1, 0 1, -1, 0 1, 1, 0 -1, 1, 0 0, 0, 4 / 1234, 125, 235, 345, 415

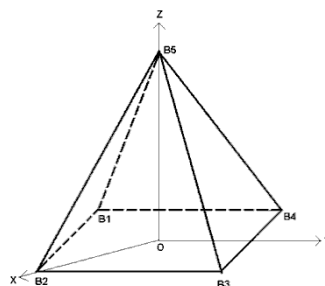


Рисунок 9.2.3.18.2 – Пример египетской пирамиды

Кроме того, имеются два специальных тела, смысл которых будет разъяснён при описании сложных систем.

### 9.2.3.19 Параллелепипед с вершиной в начале координат, SBOX(X)

Параметры:  $P_1, P_2, P_3$  - векторы рёбер тела, исходящие из начала координат. Поверхности нумеруются так же, как в теле BOX.

### 9.2.3.20 Правильная шестигранная призма с осью, совпадающей с $OZ$ , центрами нижнего основания в начале координат, SHEX(I)

Параметры:  $S, H, [f]$ . Величина  $S$  - размер под ключ,  $H$  - высота,  $f$  - угол поворота относительно оси  $OX$  в градусах, то есть параметрам  $H, D, f$  соответствует вектор  $V$  с координатами  $(D\cos(f), D\sin(f), H)$ . Если необязательный параметр  $f$  опущен, то две боковых грани параллельны плоскости  $OYZ$ .

Нумерация граней такая же, как и в теле HEX.

Отметим отличие тела SBOX от тела SHEX помимо формы: начало координат у тела SHEX находится в центре шестигранника, а у тела SBOX - в углу параллелепипеда.

При выборе в качестве контейнера ячейки трёхмерной сети (см. описание сетей) тела SHEX, третий параметр  $f$  является обязательным.

### 9.2.3.21 Пример задания секции тел

#### Пример

```
RPP KOP1 -10,10 -10,10 0,h
RCZ * 0,0,0 h dout*0.5
RCZ tws 0,0,0 h din/2
END
```

Отметим, что первый цилиндр получил имя N2.

### 9.2.3.22 Ортогональное преобразование

Кроме прямого описания тела, его можно задать в виде результата ортогонального преобразования (сдвига, поворота, отражения) ранее описанного тела. Для такого описания служит предложение вида:

```
TRANSF <имя нового тела> <имя тела-прототипа>
      <тип преобразования> <параметры преобразования>
```

<имя нового тела> - имя вновь образуемого тела, которое имеет ту же структуру, что и в прямом описании.

<имя тела-прототипа> - имя тела, из которого будет образовано новое. Оно должно задаваться идентификатором из букв и цифр в явном виде.

<тип преобразования> - буква M или R.

<параметры преобразования> - три действительных числа, обозначим их A, B и f. Смысл этих чисел определяется типом преобразования.

Тип преобразования M означает отражение от вертикальной плоскости, проходящей через точку  $(A, B, 0)$ , причём её пересечение с плоскостью  $OXY$  образует с осью  $OX$  угол в  $f$  градусов. Угол отсчитывается от оси  $OX$  против часовой стрелки. Очевидно, что одну и ту же плоскость можно задать разными способами. Рекомендуется брать угол в диапазоне от минус 90 до плюс 90.

Тип преобразования R означает поворот тела против часовой стрелки на  $f$  градусов вокруг вертикальной оси, проходящей через точку с координатами  $(A, B, 0)$ .

Тип тела при преобразовании не меняется. Тела RPP, SBOX, SHEX, PLX, PLY, UCX, UCSY не могут служить телами-прототипами.

### Пример:

TRANSF CYLFT CYLRG M 10.5, 0 90

## 9.2.4 Секция зон

Для построения зон используются три известные 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 U (заглавная латинская буква), окружённым 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,  $A U C D$  понимается как объединение  $A$  с пересечением  $C$  и  $D$ .

На рисунках 9.2.4.1, 9.2.4.2, 9.2.4.3 приведены примеры построения зон.

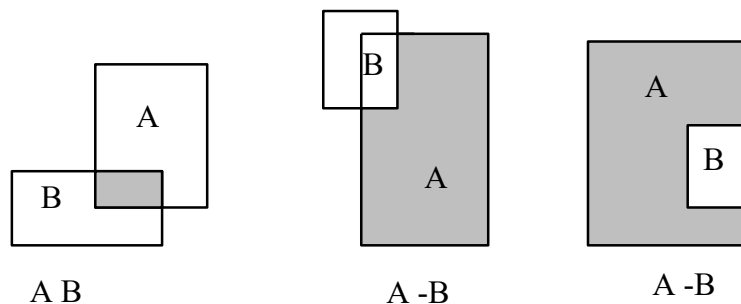


Рисунок 9.2.4.1 – Пример 1 построения зон

На рисунке 9.2.4.1 заштрихованы  $A B$  и  $A - B$ , соответственно. Выражение  $A - B$  можно трактовать как разность: совокупность точек  $A$ , не входящих в  $B$ .

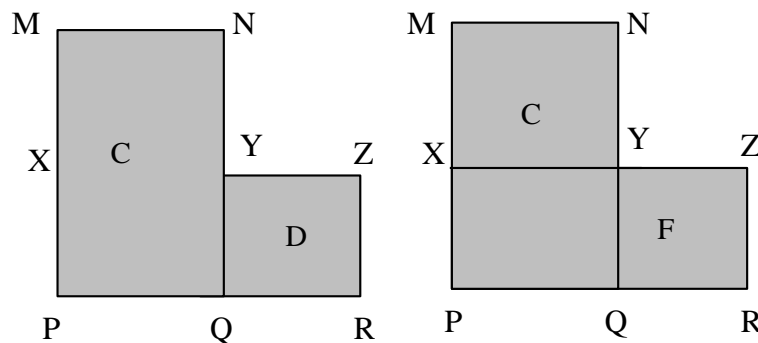


Рисунок 9.2.4.2 – Пример 2 построения зон

На рисунке 9.2.4.2 заштрихованы  $C U D$  и  $C - D$ .

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

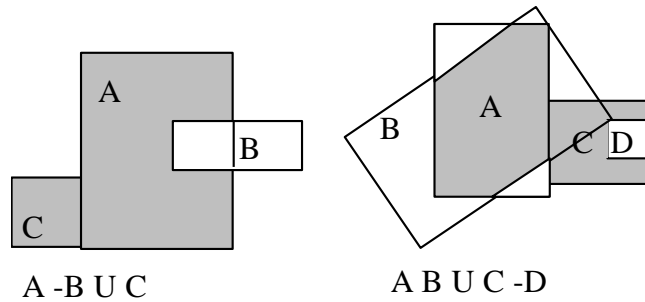


Рисунок 9.2.4.3 – Пример 3 построения зон

На рисунке 9.2.4.3 приведены примеры использования всех трёх операций. Заштрихованы результаты.

Комбинаторное описание позволяет строить достаточно сложные формы. При этом некоторые распространённые геометрические объекты приходится задавать не совсем привычным способом, например, труба задаётся как пересечение цилиндра с внешним радиусом и дополнения цилиндра с внутренним радиусом.

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

Каждая зона описывается одним предложением следующего вида:

*<имя зоны> [*<тип поиска>*]*<имя тела>*[*<знак операции>* *<имя тела>* [*<знак операции>**<имя тела >*[...]]]*<конечный фрагмент>**

Раздел зон заканчивается предложением END.

*<имя зоны>* - метка предложения, которая служит для ссылки на данную зону. Это любой идентификатор, являющийся последовательностью из не более чем 6 латинских букв и цифр, начинающийся с буквы. Все имена зон должны быть различными.

*<тип поиска>* - может принимать следующие значения:

- буква T означает табличный алгоритм поиска соседних тел для всех тел или их дополнений для данной зоны;
- конструкция вида /*<положительное целое число>* означает списковый алгоритм с дополнительным увеличением массива списков на указанное число элементов;
- отсутствие параметра означает списковый алгоритм без дополнительного увеличения массива списков.

Начинающему пользователю рекомендуется не использовать этот параметр.

*<имя тела>* - имя-идентификатор тела, описанного в секции тел, или дополнение к этому телу, если *<имя тела>* содержит в первой позиции знак минуса. Если тело имеет имя N*<натуральное число>*, то можно использовать просто это число.

Пример

Труба, образованная двумя цилиндрами (см. пример из раздела описания тел), может быть описана двумя способами:

N2 -tws

или

2 -tws

В качестве первой ссылки на тело допускается 0 (ноль), означающий все пространство как область.

#### Пример. Описание дополнения к параллелепипеду КОР1

0 -КОР1

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

<конечный фрагмент> может определяться двумя альтернативными способами. Разные зоны можно описывать разными способами.

Первый способ является наследием и некоторым развитием подхода, принятого в пакете MSU4. Он обеспечивает возможность использования исходных данных для геометрического модуля написанных для программ, созданных из моделей пакета MSU4. При использовании этого способа <конечный фрагмент> имеет одну из следующих форм:

– /<натуральное число 1>:<натуральное число 2>[/<натуральное число 3>], где <натуральное число 1> означает регистрационный номер данной зоны, <натуральное число 2> означает соответствующий материальный номер данной зоны, <натуральное число 3> означает объектный номер данной зоны. При отсутствии этого фрагмента объектный номер принимает умалчиваемое значение – один;

– /<натуральное число 1>[/<натуральное число 3>]. Такая форма употребляется, если данный регистрационный номер уже встречался, и, следовательно, соответствующий материальный номер уже определён. Если регистрационный номер встречался с разными материальными номерами, то выбирается последний из них;

– :<натуральное число 2>. Такая форма употребляется, если для данной зоны не требуется задания номера регистрационной зоны и регистрационного объекта. В этом случае их значения будут приняты равными 1;

– /<идентификатор>. Это обозначение специальной зоны, задающей краевые условия. Идентификатор означает тип условия, соответствующего пересечению границы этой зоны. Допускаются следующие типы: В - поглощение, W - белое отражение, М - зеркальное отражение, С - цилиндрическое белое отражение. Кроме того, допускаются поворотные симметрии вокруг оси OZ. Они кодируются одним из следующих символов R45, R60, R90, R120, R135, R180, R225, R240, R270, R300, R315. Число после R означает угол поворота в градусах против часовой стрелки.

При задании граничных условий с помощью специальных зон в процессе ввода не осуществляется проверка корректности их задания, например, согласования поворотов на разных границах. Поэтому начинающему пользователю лучше определять граничные условия с помощью контейнера или плоской зеркальной симметрии (см. описание секции граничных условий).

#### Пример

```
ZON1 1 -5 U 6 U 10 11 /2:2
ZON2 7 -8 15 /2/3
ZON3 T C1 SL2 U C1 SL4 /4:1/2
ZON4 /8 RCII /4
BLAK 0 -1 /B
ROTH 0 -1 2 /R300
```

Второй способ разработан в рамках пакета MSU-5 и обеспечивает большее количество возможностей по работе с геометрическими зонами. Именно этот способ будет развиваться в дальнейшем. Разработка этого способа вызвана особенностями пакета MSU-5. Так, например, программы, созданные на базе этого пакета, позволяют вести регистрацию непосредственно в области, занятой одним материалом.



При использовании второго 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:

- М или m – для задания материального номера;
- IM или im – для задания условного материального указателя (см. Описание прототипа ячейки сети), в этом случае, в отличие от первого способа, используется положительное целое число;
- Z или z – для задания регистрационного номера;
- IZ или iz – для задания условного регистрационного указателя (см. Описание прототипа ячейки сети), в этом случае, в отличие от первого способа, используется положительное целое число;
- O или o – для задания объектного номера;
- IO или io – для задания условного объектного указателя (см. Описание прототипа ячейки сети), в этом случае, в отличие от первого способа, используется положительное целое число;
- 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, этот номер и присваивается соответствующей геометрической зоне. Значением параметра является константа символьного типа. При отсутствии параметра номер материала присваивается геометрической зоне обычным образом без дополнительной обработки.

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

Специальные зоны для задания граничных условий задаются так же, как и в первом способе.

#### Пример

```
ZON1 1 -5 U 6 U 10 11 # m = 2 z = 2
ZON2 7 -8 15 # z = 2 m = 2 o = 3
ZON3 T C1 SL2 U C1 SL4 #M = 1 O = 2 Z = 4
ZON4 /8 RCII # Z = 4 m = 1
BLAK 0 -1 /B
ROTH 0 -1 2 /R300
```

Для финальной обработки в программах, собранных на базе пакета MCU-4, в отличие от MCU-5, было существенно, чтобы любая регистрационная зона содержала только один материал. В языке NCGSIM такого ограничения нет. Для возможности контроля этого служит строка

```
C=C RGMM
```

При её наличии в конце работы компилятора выдаётся список номеров регистрационных зон, содержащих более одного материала, и для каждой из них список этих материалов.

## 9.2.5 Таблица объёмов регистрационных зон

Геометрический модуль NCG может использоваться не только в рамках программы MCU, но и в составе других программ. Настоящий раздел для работы программы MCU не существен, так как в этой программе объёмы регистрационных зон вводятся среди карт для других модулей.

Таблица объёмов должна подготавливаться вручную или отдельной программой.

Обозначим через  $N_M$  максимальный регистрационный номер, использованный в описании. Объёмы задаются последовательно для первой, второй и так далее регистрационных зон. Если задано  $N$  объёмов, и  $N$  меньше чем  $N_M$ , то для  $N+1, N+2, \dots, N_M$  объёмы регистрационных зон считаются равными 1.

Таблица состоит из одного или нескольких предложений с метками: V01, V02, ..., V10, V11, ... Разбивка таблицы на предложения совершенно произвольна. Она возникает ввиду ограниченности длины одного предложения.

Объёмы задаются с помощью положительных чисел или имён констант, соответствующих положительным числам.

Они разделяются пробелами и/или запятыми.

Перед числом или именем константы может стоять коэффициент повторения вида  $N^*$ , где  $N$  целое число большее или равное 2.

Группа чисел и/или имён констант может быть взята в скобки, перед которыми стоит коэффициент повторения того же вида. В этом случае  $N$  раз повторяется вся группа. После закрывающей скобки должен стоять пробел или запятая.

Пример. Использование коэффициентов повторения  
Предложение

```
V01 VV 3*5.2 7.1 2*(3, VK, VL) 8
```

полностью эквивалентно предложению:

```
V01 VV 5.2, 5.2, 5.2 7.1 3, VK, VL 3, VK, VL 8
```

## 9.2.6 Примеры описания очень простых систем

### Пример 1

Описана двумерная (однородная по оси 0Z) модель ячейки сборки TRX. Ячейка - правильная шестигранная призма с замедлителем. В центре расположен цилиндрический стержень, состоящий из топлива и оболочки, между которыми есть тонкий зазор. Бесконечность вдоль оси 0Z задаётся с помощью граничных условий трансляции на торцах призмы. Условие бесконечности решётки шестигранных ячеек задаётся с помощью условия зеркального отражения на боковых поверхностях призмы.

Материальные номера имеют следующий смысл: 1 - топливо; 2 - замедлитель; 3 - материал оболочки; 4 - воздух. В качестве объёмов здесь взяты площади поперечных сечений.

```
HEAD 3 0
CONT T T M M M M M
*      секция тел, первое тело - контейнер.
HEX  C  0,0,0  1.806,0,100
RCZ  FU 0,0,0  100    0.4915
RCZ  ZA 0,0,0  100    0.5042
RCZ  CL 0,0,0  100    0.5753
END
*      секция зон.
FUEL  FU      /1:1
SPACE ZA -FU  /2:4
CLAD  CL -ZA  /3:3
```

```

WATR      C   -CL   /4:2
END
*          таблица объёмов.
V01      0.7589   0.0397  0.2411  1.7724
FINISH

```

### Пример 2

Описывает ту же самую геометрию, но при условии, что придётся часто менять следующие параметры: *rt* - радиус топливной таблетки, *dz* - толщина воздушного зазора, *rc* - радиус покрытия, *step* - шаг решётки.

```

HEAD      3      0
CONT      T      T M M M M M M
EQU       rt = 0.4915
EQU       dz = 0.0127
EQU       rc = 0.5753
EQU       step = 1.806
*          вычисления объёмов.
EQU       VF = 3.1416*rt*rt
EQU       VZ = 3.1416*(rt+rt+dz)*dz
EQU       VC = 3.1416*(rc-rt-dz)*(rc+rt+dz)
EQU       VW = step*step*cos(30)-3.1416*rc*rc
*          тела.
HEX       C      0,0,0   step,0,100
RCZ       FU     0,0,0   100 rt
RCZ       ZA     0,0,0   100 rt+dz
RCZ       CL     0,0,0   100 rc
END
*          зоны.
FUEL      FU          /1:1
SPACE     ZA -FU     /2:4
CLAD      CL -ZA     /3:3
WATR      C      -CL   /4:2
END
V01       VF, VZ, VC, VW
FINISH

```

## 9.3 ОПИСАНИЕ СЛОЖНЫХ СИСТЕМ

Существует возможность упростить задание геометрических объектов, имеющих повторяющиеся элементы. Такие объекты можно задавать с помощью сетей и решёток.

Сети позволяют заполнить плотно (или почти плотно) часть пространства ячейками, имеющими одинаковую внешнюю форму.

Решётки позволяют поместить любое количество экземпляров своих элементов в любую точку пространства объекта.

Сеть может быть использована для задания внутренней структуры элемента решётки.

Описание сложной системы состоит из последовательности предложений. Некоторые предложения объединены в секции. Порядок секций дан в следующей схеме:

```

<заголовок>
<раздел граничных условий>
<секция тел>
<секция зон>
[<описание прототипов ячеек сетей>]
[<описание сетей основной геометрии>]
[<описание прототипов элементов решёток>]

```

[<описание решёток>]  
[<таблица объёмов регистрационных зон>]  
FINISH

Описание прототипов ячеек сетей, прототипов элементов решёток и собственно описания сетей и решёток не обязательно должны следовать в указанном порядке. Должно лишь быть соблюдено следующее правило: при описании сети и/или решётки все используемые в ней прототипы должны быть уже описаны.

<заголовок>, <раздел граничных условий>, <секция тел>, <секция зон>, <таблица объёмов регистрационных зон> задаются практически так же, как для простой системы. Отличия перечислены ниже.

Для сложной системы введена конструкция, определяющая, что зона является зоной-носителем сети. Каждая зона-носитель сети описывается одним предложением, следующего вида:

```
<имя зоны> (<имя сети>) [<тип поиска>]<имя тела>..  
[<знак операции> <имя тела>[<знак операции><имя тела>  
[...]]]<конечный фрагмент>
```

Все фрагменты, кроме нового (<имя сети>), имеют тот же смысл, что и в случае зоны, описанной для простых систем. Для фрагмента <тип поиска> запрещён тип T - табличный поиск соседей.

Новый фрагмент, отличающий зону-носитель сети, есть:

(<имя сети>) – взятый в скобки идентификатор, являющийся последовательностью из не более чем 6 латинских букв и цифр, начинающийся с буквы, определён при описании сети и здесь приводится для ссылки на неё.

*Запрещается использовать одну зону в качестве зоны-носителя нескольких сетей.*

Пример

```
Z1 (LTH) A1 -A2 /5:1
```

При описании тел возможно присвоение им так называемых дополнительных имён. Тело-контейнер по умолчанию имеет дополнительное имя С. Для остальных тел они могут задаваться в скобках после основного имени тела. Эти дополнительные имена используются при задании решёток в списках дополнительных ограничений (см. описание решётки).

Пример

```
RPP BR0 (CT0) -100,100 0,200, 0,1000
```

Здесь тело имеет дополнительное имя СТ0.

Дополнительные имена не могут быть использованы при описании зон. Они могут совпадать с обычными именами зон, именами констант и т.д., так как по контексту всегда ясно, что имя трактуется как дополнительное имя тела.

В разделе <таблица объёмов регистрационных зон> задавать объёмы нужно с учётом размножения при генерации решёток, а также их наложения на зоны-носители.

### **9.3.1 Сети (основные понятия)**

Структура геометрических данных, связанных с системой координат, имеет два уровня.

Конструкции в целом описываются в глобальной системе координат. Это верхний уровень. Сети описываются в системах координат нижнего уровня. *Сетью* называются двумерные или трёхмерные массивы ячеек одинаковой формы, плотно примыкающих друг к другу.

Системы координат нижнего уровня - это локальные системы координат ячеек (не зависят от сдвига). Каждая сеть порождает набор систем координат нижнего уровня, по числу составляющих сеть ячеек.

Ячейки сети могут быть трёх типов:

- произвольный параллелепипед;
- параллелепипед, ребра которого параллельны осям системы координат;
- правильная шестигранная призма с осью, параллельной оси OZ.

Сеть может быть двумерной или трёхмерной.

### 9.3.1.1 Двумерная сеть

Двумерная сеть образуется из ячеек с помощью сдвигов вида:

$$C + P(i-1) + Q(j-1),$$

где  $i, j$  - целые числа;

вектор  $C$ , называемый *корневым вектором*, задаёт положение в пространстве ячейки с индексами (1,1).

Отметим, что:

- для параллелепипеда векторы  $P$  и  $Q$  являются векторами его первых двух рёбер;
- для призмы вектор  $P$  лежит в плоскости OXY, имеет длину размера под ключ, а поворот этого вектора соответствует ориентации призмы; вектор  $Q$  получается из вектора  $P$  поворотом на 120 градусов против часовой стрелки.

Примеры задания двумерных сетей приведены на рисунках 9.3.1.1.1 и 9.3.1.1.2.

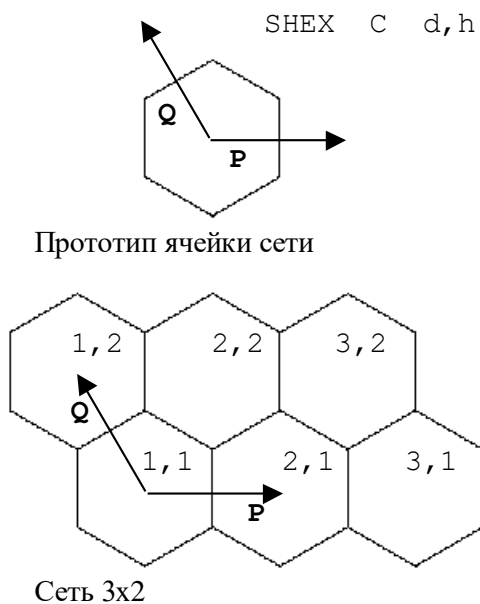


Рисунок 9.3.1.1.1 – Сеть 3x2

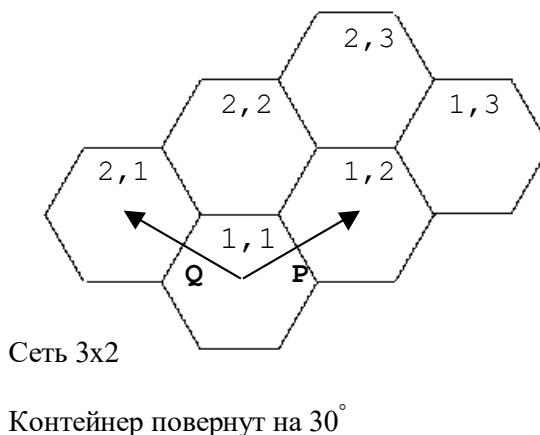
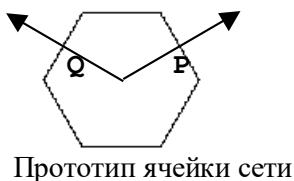


Рисунок 9.3.1.1.2 – Сеть 3x2 с поворотом контейнера

**Пример**

Две боковые поверхности призмы параллельны плоскости OYZ,  $P = r l_x$  и  $Q = -0.5 l_x + 0.5\sqrt{3} l_y$ , где  $l_x, l_y$  – базисные векторы, а  $r$  – размер шестигранника под ключ.

**9.3.1.2 Трёхмерная сеть**

Трёхмерная сеть образуется из ячеек с помощью сдвигов вида:

$$C + P(i-1) + Q(j-1) + R(k-1),$$

где  $i, j, k$  – целые числа;  
векторы  $C, P$  определяются так же, как в двумерном случае.  
Отметим, что:

- для параллелепипеда вектор  $R$  соответствует его третьему ребру;
- для призмы вектор  $R$  определён как  $h l_z$ , где  $h$  – высота призмы.

Индексы  $i, j, k$  принимают значения:

$$1 \leq i \leq N,$$

$$1 \leq j \leq M,$$

$$1 \leq k \leq L.$$

Пример задания трехмерной сети приведен на рисунке 9.3.1.2.1.

Каждая сеть вкладывается в одну или несколько зон, описанных в глобальной системе координат, называемых *зонами-носителями сети*.

Частица всегда покидает сеть при пересечении границы зоны-носителя, то есть происходит переход в глобальную систему координат.

Если сеть не заполняет всю зону-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 зоны-носителя (по этой оси не происходит дополнение сети фиктивными ячейками).

Зона-носитель может содержать не все ячейки сети или неполные ячейки, то есть «обрезанные» границей зоны.

Все ячейки одной сети должны иметь одинаковую внешнюю форму, но их внутренняя геометрия совершенно независима.

При использовании сетей уменьшение быстродействия невелико, а затраты памяти существенно меньше, чем при прямом описании.

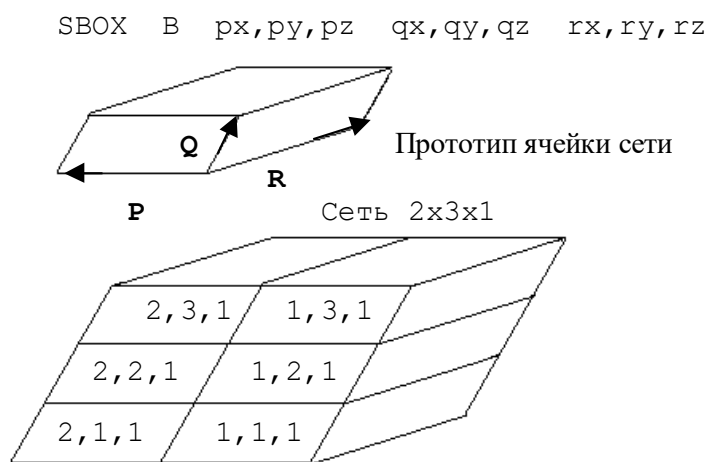


Рисунок 9.3.1.1.2 – Сеть 2x3x1

### 9.3.2 Описание прототипа ячейки сети

Описание прототипа ячейки сети имеет следующую структуру:

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

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

<имя прототипа ячейки сети> - произвольный идентификатор, являющийся последовательностью из не более чем 6 латинских букв и цифр, начинающийся с буквы. Оно служит для ссылки на данный прототип при задании картограммы сети.

Это имя должно быть уникальным среди имён 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, определённые 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 соответствующего раздела в описании глобальной геометрии (см. описание простых систем).

Нумерация тел ведётся от единицы. В разделе зон необходимо ссылаться только на тела данного описания прототипа. Это относится не только к телам в описаниях зон, но и к телам в конструкциях TRANSF.

Для определения регистрационных, объектных и материальных номеров, которые зависят не только от зоны прототипа ячейки сети, но и от положения ячейки в зоне-носителе, могут быть использованы условные регистрационные указатели (УРУ), условные объектные указатели (УОУ) и условные материальные указатели (УМУ). Каждый УРУ, УОУ и УМУ кодируется отрицательным целым числом.

Перекодировка условных указателей в номера для разных сетей и для разных прототипов ячеек происходит независимо, поэтому *рекомендуется* для каждого прототипа ячейки кодировать эти указатели, начиная с -1.

#### Пример

Зона в описании прототипа ячейки сети имеет условный регистрационный указатель 6, условный объектный указатель 2, а материальный номер 1.

```
ZPE A5 A6 -C /-6:1/-2
```

#### Пример

Описание прототипа ячейки сети. Предполагается, что константа REM определена в основном блоке. Здесь зона ZN1 имеет "безусловные" регистрационный и объектный номера. Они равны 4 и 5 независимо от положения ячейки в сети.

```
CELL NC
SBOX S 10.,0.,0. 5.,5.,0. 0.,0.,3.
EQU RS = REM*0.5
SPH H 7.5,2.5,1.5 RS
END
ZN1 S -H /4:2/5
ZNTE H /-1:1/-1
END
```

### 9.3.3 Описание сетей основной геометрии

Описание сети имеет следующую структуру:

```
<заголовок>
<картограмма имён прототипов ячеек, заполняющих ячейки сети>
[<картограммы регистрационных номеров>]
[<картограммы объектных номеров>]
[<картограммы материальных номеров>]
[<картограммы символьных имён групп>]
END
```

#### 9.3.3.1 Заголовок

Заголовок состоит из одного предложения вида

```
NET <имя сети> <корневой вектор> <число столбцов>
<число строк>[<число слоёв>]
```

NET - метка предложения.

<имя сети> - любой идентификатор, являющийся последовательностью из не более чем 6 латинских букв и цифр, начинающийся с буквы. Оно служит для ссылки на данную сеть при описании зоны-носителя сети.

<корневой вектор> - параметр, определённый тремя координатами корневого вектора  $C_x$ ,  $C_y$ ,  $C_z$ , 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.

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

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

*<число слоёв>* - то же по третьему вектору (L).

Первый, второй и третий векторы определяются телом-контейнером ячеек сети. Это тело должно быть одним и тем же для всех прототипов ячеек одной сети.

Если параметр *<число слоёв>* отсутствует, то сеть определена как двумерная.

### 9.3.3.2 Картограмма имён прототипов ячеек

Картограмма имён прототипов ячеек сети для двумерного случая задаётся как M предложений, содержащих по N фрагментов каждая, где N – число сдвигов по первому вектору, M – число сдвигов по второму вектору. Если сеть трёхмерная, то картограмма задаётся послойно, то есть совокупность из M предложений по N фрагментов каждое задаётся L раз, где L – число сдвигов по третьему вектору.

Метками M предложений должны быть идентификаторы, состоящие из латинской буквы T и номера строки, то есть T01, T02, T03,..., T09, T10, T11,... и т.д.

Каждое предложение отвечает одной строке, то есть набору индексов (1,j),..., (N,j), и содержит N фрагментов. Один фрагмент имеет вид:

*[ - ] <имя прототипа ячейки сети >*

Знак минус [ - ] означает, что ячейка не пересекается с границей зоны-носителя сети. Указание, что ячейка не пересекается с границей зоны-носителя сети, несколько увеличивает быстродействие, но требует большой аккуратности при задании исходных данных. Начинающему пользователю не рекомендуется пользоваться знаком минус.

*<имя прототипа ячейки сети >* даёт ссылку на прототип, который соответствует данной ячейке и задан при описании прототипа.

Если подряд идут несколько ячеек, имеющих один и тот же прототип, то возможна сокращённая запись с помощью коэффициента размножения *<k>*.

*[ <k>\* ] [ - ] <имя прототипа ячейки сети >*

*<k>* - целое число, равное числу повторений следующего за ним фрагмента.

#### Пример. Два эквивалентных предложения

```
T01  -C4  3*-CC  2*TW  -C4
T01  -C4  -CC  -CC  -CC  TW  TW  -C4
```

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

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

Для трёхмерной сети картограмма задаётся послойно. Задаётся L двумерных картограмм (размерности NxM), каждая из которых начинается картой вида:

*T LAYER <номер слоя >*

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

После этой карты следует M предложений с метками T01, T02, T03,..., T09, T10, T11,... (Метки повторяются для каждого слоя).

### 9.3.3.3 Картограммы регистрационных номеров

Условные регистрационные указатели (УРУ) должны вводиться по порядку, желательно без пропусков, начиная с единицы. Для каждого регистрационного указателя должна вводиться своя картограмма. Значения условных регистрационных указателей локальны в каждом прототипе ячейки, и значение регистрационных номеров, помеченных таким образом зон ячеек сети, присваиваются в соответствии с картограммой, то есть в разных ячейках в зависимости от их положения в сети одному и тому же условному регистрационному указателю могут соответствовать разные регистрационные номера зон.

Все регистрационные номера, заданные в прототипах ячеек без использования условных указателей, сохраняются во всех ячейках независимо от того, что указано в картограммах. Их перекодировка никогда не производится.

Если число  $N_{урu}$  означает максимальное значение условного регистрационного указателя во всех прототипах ячеек, составляющих сеть, то описание двумерной сети должно включать  $N_{урu}$  картограмм значений регистрационных номеров.

Каждая картограмма регистрационных номеров состоит из  $M$  предложений, содержащих  $N$  номеров:

$P\langle k\rangle\langle j\rangle$  *<регистрационный номер> <регистрационный номер>...*

$P\langle k\rangle\langle j\rangle$  - метка предложения для  $k$ -ой картограммы и  $j$ -ой строки;

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

$\langle j\rangle$  - натуральное число, имеющее два символа (для чисел меньше 10 левый 0 записывается), номер строки ячеек сети, меняется от 1 до  $M$ .

#### Пример

Если  $j = 5$ , а  $k = 11$ , то метка будет

P1105

*<регистрационный номер>* - натуральное число, имеющее значение регистрационного номера, который будет присвоен зоне с условным регистрационным указателем  $k$  в ячейке, которая будет помещена в сети путём сдвига на  $j$  позиций по второму вектору и на число позиций, соответствующих положению в строке, по первому вектору.

Допустимы коэффициенты повторения отдельных чисел и групп чисел.

*Использование имён констант не разрешается.*

Если в картограмме имен прототипов в соответствующей позиции ( $j,k$ ) задан ноль, то в картограмме регистрационных номеров в этой позиции также может быть указан ноль. В противном случае должно быть указано отличное от нуля значение, даже если в соответствующем прототипе ячейки сети не используется условный регистрационный указатель  $k$ .

#### Пример. Две аналогичных строки картограммы регистрационных номеров

Такая строка означает, что зоны с третьим условным указателем в ячейках с индексами по векторам сдвигов (1,1), (2,1), (3,1) получают регистрационный номер 2, в ячейке (4,1) – номер 7, (5,1) – номер 5 и так далее.

P0301 3\*2 7 2\*(5,6,9)

P0301 2,2,2 7 5,6,9 5,6,9

Если вся  $k$ -ая картограмма состоит из одного и того же значения, то вместо  $M$  предложений можно написать одно предложение вида:

$P\langle k\rangle ALL$  *<регистрационный номер>*

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

#### Пример. Строка картограммы регистрационных номеров

Такая строка картограммы регистрационных номеров означает, что второй условный указатель в помеченных этим условным указателем зонах всех ячеек получит значение 23.

P02ALL 23

Если сеть трёхмерная, то картограммы регистрационных номеров задаются послонно.

Задаётся  $L \times N_{ур}$  картограмм размерности  $M \times N$ . Перед каждой картограммой вводится карта вида:

P $\langle k \rangle$ LAY  $\langle$ номер слоя $\rangle$

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

$\langle$ номер слоя $\rangle$  - натуральные числа от 1 до L по порядку, имеющий значение номера слоя.

#### Пример

P05LAY 11

Затем вводятся M предложений, построенных аналогично двумерному случаю.

#### **9.3.3.4 Картограммы объектных номеров**

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

#### **9.3.3.5 Картограммы материальных номеров**

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

#### **9.3.3.6 Картограммы символьных имен групп**

Если в разделе PIN для описания материала используется символьное имя группы, то после описания этого материала в картограмме материальных номеров требуется ввести картограмму(ы) символьных имён групп. Картограммы символьных имён групп вводятся после картограмм материальных номеров. Эти картограммы аналогичны картограммам материальных номеров, но вместо номера материала вводится символьное имя группы. В тех местах картограммы, где символьное имя группы не определено, ставится цифра "0". Кроме того, в метках предложений латинская буква 'M' должна быть заменена латинской буквой 'G'.

### **9.3.4 Решётки (основные понятия)**

Решётка состоит из набора элементов. *Элемент решётки* – это некоторое тело, разбитое на геометрические зоны с приписанными регистрационными, объектными и материальными номерами. Элементы решётки размещаются в зонах глобальной геометрии. Обычно прототипом элемента решётки служит какой-либо технический элемент конструкции, например, твэл или пэл. Можно выделить в системе какой-то другой повторяющийся объём, границы которого могут не являться поверхностями деталей, например, шестигранная ячейка в кассетах водо-водяных реакторов.

Контейнером прототипа элемента решётки может быть любой тип тела (см. раздел тел).

Прототип элемента решётки описывается в локальной системе координат.

Прототип элемента решётки может содержать в себе зоны-носители сетей и описания этих сетей. Зоны прототипа элемента решётки могут быть носителями только тех сетей, которые принадлежат данному прототипу элемента решётки. Корневые векторы этих сетей даются в локальной системе координат прототипа.

Прототипы ячеек этих сетей описываются отдельно от прототипа элемента решётки и от самой решётки. Одна и та же ячейка может использоваться в сетях нескольких прототипов элементов решёток и одновременно в сетях глобальной геометрии.

Прототип элемента решётки накладывается на зоны основной геометрии с помощью сдвигов. Сдвиг можно трактовать как вектор переноса прототипа на конкретное место или как точку в глобальных координатах, в которой помещается начало локальной системы координат для данного элемента решётки.

Различные элементы не должны иметь общих точек, не считая граничных точек, которые могут быть общими. Наложение понимается в том смысле, что области, занятые элементами, изымаются из зон. То есть, зоны модифицируются путём пересечения с дополнением контейнеров всех элементов.

Сдвиги вырабатываются подпрограммами генерации сдвигов, которых в настоящее время имеется две. При описании решётки необходимо указать, какая из этих подпрограмм будет использоваться, и задать для неё строку параметров.

На рисунке 9.3.4.1 изображена решётка из трёх квадратов.  $S_1, S_2, S_3$  - векторы сдвигов для элементов решётки,  $I$  - прототип элемента решётки. Зона  $B$  преобразуется в

$$B - S_1(I) - S_2(I) - S_3(I),$$

где  $S_i(I)$  -  $i$ -тый элемент  $I$ .

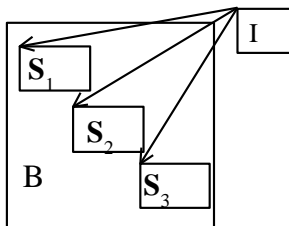


Рисунок 9.3.4.1 – Решётка из трёх квадратов

Контейнер прототипа решётки может быть расположен произвольно относительно начала локальной системы координат.

Одна решётка может быть наложена на несколько зон, а на одну зону может быть наложено несколько решёток. Элементы всех наложенных решёток не должны иметь общих точек кроме граничных.

При описании прототипов элементов решёток кроме обычных регистрационных и объектных номеров можно задавать условные указатели. Зоне с условным указателем настоящий регистрационный или объектный номер будет приписан только после генерации решётки.

Решётка может состоять из элементов нескольких прототипов. Генератор сдвигов помимо положения элемента задаёт его тип.

На рисунке 9.3.4.2 изображена решётка, образованная элементами различных прототипов.

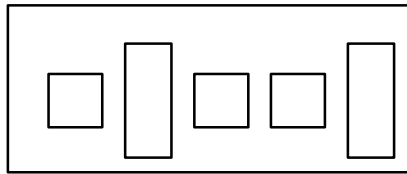


Рисунок 9.3.4.2 – Решётка, образованная элементами различных прототипов

Имена зон, на которые накладывается решётка, указываются при описании этой решётки.

Число элементов в решётке обычно значительно меньше, чем в сети, а их расположение в пространстве менее регулярно.

Допускается наложение решётки на зону-носитель сети. Если точка одновременно лежит в ячейке сети и элементе решётки, приоритет отдаётся решётке.

Пометки граничных ячеек сети следует делать, учитывая обрезание ячеек элементами решётки. В этом случае рекомендуется все ячейки описать как граничные.

### 9.3.5 Описание прототипа элемента решётки

Описание прототипа элемента решётки имеет следующую структуру:

```
LCELL <имя прототипа элемента решётки>  
<описание тел>  
<описание зон>  
[<описание сетей>]  
ENDL
```

Описание прототипа элемента решётки подобно описанию прототипа ячейки сети.

<имя прототипа элемента решётки> - произвольный идентификатор, являющийся последовательностью из не более чем 6 латинских букв и цифр, начинающийся с буквы. Оно служит для ссылки на данный прототип при задании решётки.

Это имя должно быть уникальным среди имён прототипов ячеек сетей и имён прототипов элементов решёток. Имена, использованные для констант или тел, могут совпадать с именами прототипов ячеек сетей и прототипов элементов решёток.

Контейнером служит тело, описанное первым. Оно может быть любым. *Не рекомендуется* использовать для этого тела SBOX и SHEX, а также неограниченные тела, то есть SLA, SLB, UCX, UCY, UCZ, PLG, PLX, PLY, PLZ.

При описании прототипа элемента решётки могут использоваться константы, определённые 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 элемента решётки.

Раздел тел ничем не отличается от соответствующего раздела в описании глобальной геометрии.

В разделе зон необходимо ссылаться только на тела данного раздела описания прототипа. Это относится не только к телам в описаниях зон, но и к телам в конструкциях TRANSF.

Для определения регистрационных и объектных номеров могут быть использованы УРУ и УОУ, образование которых объяснено выше. Они кодируются отрицательными числами. Абсолютные значения этих чисел есть значения указателей.

Для преобразования условного указателя в регистрационный или объектный номер применяется следующий алгоритм.

Обозначим через  $N_{pm}$  максимальный регистрационный номер, использованный к моменту генерации решётки. Пусть в первом элементе решётки имеются условные регистрационные указатели со значениями до  $k$  включительно. Тогда зоны с первым

условным указателем получают регистрационный номер  $N_{pm}+1$ , со вторым -  $N_{pm}+2$  и так далее до  $N_{pm}+k$ . После этого  $N_{pm}$  будет увеличен на  $k$ , и начнётся размещение второго элемента решётки. Если набор используемых условных указателей имеет пропуски (например, 1,2,5,6), то и набор номеров регистрационных зон также будет иметь пропуски.

Алгоритм замены условных объектных указателей точно такой же, только вместо  $N_{pm}$  используется  $N_{om}$  - максимальный объектный номер, использованный к моменту генерации решётки.

Описание сети внутри прототипа элемента решётки подобно описанию сети в глобальной геометрии.

Имя такой сети также должно быть уникальным во всем описании геометрии.

В картограммах регистрационных и объектных номеров сети могут использоваться отрицательные целые числа. Они трактуются как условные регистрационные и объектные указатели элементов решётки и будут получать истинные значения при генерации решётки по приведённым выше правилам.

#### Пример

Приведено описание прототипа элемента решётки, зона Z1 которого содержит сеть EN1, состоящую из ячеек типа CE1, CE2. В ячейках CE1 и CE2 использовано по одному УРУ и УОУ (в соответствии с числом картограмм). Отрицательные числа в картограмме регистрационных зон сети являются УРУ элемента решётки и при генерации решётки заменяются положительными регистрационными номерами. На рисунке 9.3.5.1 в скобках даны индексы элементов сети, через «/» приведены регистрационные и объектные номера.

```

HEAD  1  0  0
CONT  W
EQU   RT = 1.8
SPH   *  0,0,0    200
END
QQ    1          /1:1
END
*    описание прототипов ячеек сетей
CELL  CE1
SBOX  A1  1,0,0    0,-1,0    0,0,2
RCZ   A2  0.5,-0.5,0  2    0.25
END
ZA1   A1  -A2    /2:2/1
ZA2   A2          /-1:6/2
END
CELL  CE2
SBOX  B1  1,0,0    0,-1,0    0,0,2
RCZ   B2  0.5,-0.5,0  2    0.4
END
ZB1   B1  -B2    /3:3/3
ZB2   B2          /-1:5/-1
END
*    описание прототипа элемента решётки
LCELL EXAMP
RCZ   *  0,0,0    10    RT
EQU   R1 = 1.5
RCZ   *  0,0,0    10    R1
END
Z1    (EN1)  2          /-1:1/1
Z2    1 -2    /31:4/-1
END
*    константа RT определена в глобальной геометрии
NET   EN1  -1,1,0  2  2
T01   CE1    CE2

```

```

T02   CE2   CE1
P0101 2*23
P0102 -2,-3
O01ALL 9
END
ENDL
*
LATT      GLTL   QQ
LISTEL    EXAMP
PARM      0,0,0
FINISH

```

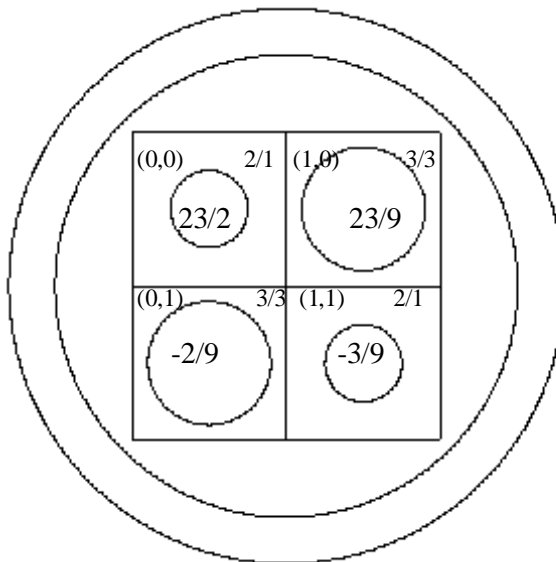


Рисунок 9.3.5.1 – Пример использования прототипа элемента, содержащего сеть.

### 9.3.6 Описание решётки

Описание решётки имеет вид:

```

LATT <имя генератора сдвигов> <имя зоны> [<имя зоны>
                                         [<имя зоны>]... [<имя зоны>]]
LISTEL <имя ПЭР> [ (список ограничения) ]
        [<имя ПЭР> [ (список ограничения) ]
        [<имя ПЭР> [ (список ограничения) ] ]
...
        [<имя ПЭР> [ (список ограничения) ] ] ]
PARM <строка параметров>

```

<имя генератора сдвигов> - имя подпрограммы, которая размещает элементы решётки. Можно использовать стандартные подпрограммы GLTL и G2AR. Параметры для них задаются строкой PARM.

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

<имя ПЭР> - имя прототипа элемента решётки описанного выше. Должны быть приведены имена всех использованных прототипов, они могут встречаться более одного раза. 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

программы генератора сдвигов. Какой-либо элемент списка может быть не использован в расчёте, но нумерация ведётся по списку.

(*<список ограничения>*) - имя или несколько имён тел, описанных в глобальной геометрии; имена могут быть разделены пробелами или запятыми. Если имя приведено со знаком минус, то оно трактуется как дополнение данного 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. Проверяется пересечение каждого элемента, после которого приведён такой список, со всеми указанными в списке телами, поэтому для увеличения быстродействия *рекомендуется* заведомо не обрезанные граничными поверхностями элементы указывать без списка ограничения, а затем, в случае необходимости, повторно привести их имя со списком высечения.

Тело-контейнер основной геометрии всегда по умолчанию имеет глобальное имя *C* и может в таком виде использоваться в списках ограничения. Дополнительное имя тела-контейнера *C* оказывается полезным, если часть экземпляров элементов решётки выходит за пределы тела-контейнера. Такая конфигурация часто встречается при расчёте полиячеек.

#### Пример

Все экземпляры элемента TVEL могут иметь пересечение с границей контейнера, тело TVEL будет иметь номер один в списке, PELA - номер 2, PELB - номер 3 для параметров программы генератора сдвигов.

```
LISTEL TVEL(C) PELA PELB
```

#### Пример

Задан контейнер и поворотная симметрия на 90 градусов.

```
HEAD 1 0 400
CONT T T W
RCZ CNT 0, 0, 0 H RGL ; контейнер имеет номер1
PLY P0(C0) 0 ; тело имеет номер 2
PLX P1(C1) 0 ; тело имеет номер 3
```

Далее пусть имеется описание поворотной симметрии

```
ROTP CNT -P0 /R90
ROTM CNT -P1 P0 /R270
```

В описании решётки применена конструкция

```
LISTEL KAC(C,C0,C1)
```

В этом случае к каждой зоне любого экземпляра элемента KAC будет добавлено пересечение с телами 1, 2, 3, т.е. четвертой частью цилиндра. Злоупотребление такой конструкцией несколько уменьшает быстродействие, но значительно упрощает задание входных данных.

Структура *<строки параметров>* зависит от того, какая подпрограмма использована для генерации сдвигов.

### **9.3.6.1 Генератор сдвигов GLTL**

Подпрограмма GLTL задаёт радиус-векторы положения экземпляров элементов решётки путём прямого перечисления их координат.

*<строка параметров>* состоит из последовательности фрагментов следующего вида:



[/*<целое число>*]*<радиус-вектор положения>*

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

*<радиус-вектор положения>* - параметр, определённый тремя координатами радиус-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 распространяется только на один следующий за ним фрагмент, в отличие от генератора G2AR (см. ниже).

#### Пример

Описание решётки содержит 2 прототипа элементов: первый и второй, поскольку наибольший номер в строке параметров 2. Элемент второго типа имеется только один с координатами радиус-вектора X4,Y4,0.5, остальным по умолчанию дается тип 1. Если бы в строке параметров вместо цифры два была бы цифра три, то при описании решётки в строке LISTEL должно было бы быть три имени, хотя происходило бы размещение только прототипов один и три.

```
LATT          GLTL      ZN1 , FUT
LISTEL        TPIC      TP2D
PARAM         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
```

### 9.3.6.2 Генератор сдвигов G2AR

Подпрограмма G2AR используется в том случае, когда решётка получается из регулярного двумерного массива исключением некоторых элементов. Регулярный двумерный массив образуется как набор сдвигов вида:  $\{A+iB+jC: I_i \leq i \leq I_s, J_i \leq j \leq J_s\}$

*<строка параметров>* имеет следующую структуру:

*<размерность массива>* *<векторы A, B, C>* *<список исключений>*  
[/*2<список положений элементов типа 2>*]  
[/*3<список положений элементов типа 3>*]...

*<размерность массива>* - целые числа, задают пределы изменения индекса *i* и пределы изменения индекса *j* в форме:

$[I_i:]I_s$   $[J_i:]J_s$

$I_i, I_s, J_i, J_s$  – нижняя и верхняя границы изменения индексов. По умолчанию нижняя граница полагается равной нулю.

*<векторы A, B, C>* - параметр определён тремя координатами корневого радиус-вектора **A**, тремя координатами вектора сдвига **B** и тремя координатами вектора сдвига **C**. Каждая из координат может быть задана действительным числом, ранее определённым именем константы или выражением, содержащим числа и константы. Параметры должны быть разделены запятыми или пробелами.

*<список исключений>* задаёт те положения в массиве сдвигов, которые не входят в решётку, то есть на эти места не накладывается никакого элемента. Положения в списке задаются в виде:

[*<целое число 3>:*]*<целое число 1>*, [*<целое число 4>:*]*<целое число 2>*

*<целое число 1>*, *<целое число 2>* задают значения пары индексов *i, j* массива сдвигов по векторам **B** и **C** соответственно. Последовательность пар вида *a, j, a+1, j, ..., b, j* может быть сокращённо записана в виде *a:b, j*. Аналогично *i, a:b* означает *i, a, i, a+1, ..., i, b*. Можно также использовать конструкцию *a:b,c:d*.

Порядок перечисления исключаемых положений произволен.

[/2<список положений элементов типа 2>] задаёт положения элементов, прототип которых задан вторым в строке LISTEL. Положения в списке задаются в виде пары индексов  $i, j$  по тем же правилам, что и в списке исключений.

[/3<список положений элементов типа 3>]... - аналогичен списку для второго прототипа и так далее.

Положения, не входящие ни в какой список, заполняются элементами, прототип которых указан первым в строке LISTEL.

Номера должны идти по порядку возрастания, допустимы пропуски.

Если одно положение попадает в два списка и более, то даётся предупредительная диагностика, и считается действительным вхождение в тот список, в который положение попало в первый раз. Список исключений является самым приоритетным.

### Пример

В куб вкладывается решётка прямоугольных каналов. Рисунок 9.3.6.2.1 дан в глобальной системе координат. Изображён радиус-вектор  $A$  - положение "нулевой" точки решётки и векторы  $B$  и  $C$ . Решётка состоит из двух типов элементов, второй тип отмечен вертикальной чертой.

Описание решётки (вместо 0:5 можно записать просто 5):

```
LATT      G2AR   CUB
LISTEL    TW1   TW2
PARM      0:5  -1:4  A1,A2,A3  B1,B2,B3  C1,C2,C3
          0:1,-1  0:1,0  0,1  0,3:4  4:5,2:4
          /2   3,0   2,1:2
```

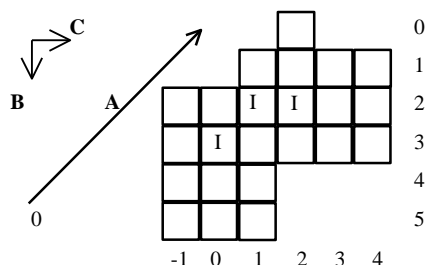


Рисунок 9.3.6.2.1 – Задание решетки прямоугольных каналов

### Пример

Используются все конструкции многоуровневой геометрии.

```
HEAD      7   0  100
CONT      W
*         глобальная геометрия
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      ;зона-носитель решётки
ZN (NT)   HOCN   /4:2      ;зона-носитель сети
END
*         первый прототип ячейки сети, используемой в 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 сети, используемой в сети NT
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, лежащая в зоне глобальной геометрии
NET      NT  -6,-6,-2  4,3
T01      4*CN1
T02      CN2 CN1  CN2  CN1
T03      CN1 CN2  CN1  CN1
*        картограмма условных регистрационных указателей
P0101    4*0
P0102    5 0 6 0
P0103    0 7 0 0
P0201    4*0
P0202    9 0 9 0
P0203    0 10 0 0
END
*        прототип ячейки сети, используемой в сети решётки
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      /-1:14
END
*        первый прототип элемента решётки; он не содержит сетей
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
C = C SHOW
END
LZZ      BL -C1 -C2     /-1:1
C12      C1 U C2       /-2:5
END
ENDL
*        второй прототип элемента решётки; он содержит сеть
LCELL    BVV
RPP      BL  -0.8,0.8  -0.7,0.7  0,3
END
CC3      (VNT) BL      /2:2
END
*        внутренняя сеть
NET      VNT  -0.8,-0.7,0  2,2
T01      CLN  CLN
T02      CLN  CLN

```

```

P0101      -1  3
P0102       4 -1
END
ENDL
*          описание решётки
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
*
C = C SHOW
*          C = C RGMM
V01     4.1888  0.1  3612.8  0.2
FINISH

```

### 9.3.6.3 Генератор сдвигов G2MP

Генератор сдвигов G2MP, так же как и генератор G2AR используется для определения решётки, получаемой из регулярного двумерного массива элементов. В отличие от сложного и ненаглядного формата генератора G2AR, генератор G2MP использует картограмму, аналогичную той, которая используется при описании сетей.

Решётка получается из регулярного двумерного массива, образуемого как набор сдвигов вида:  $\{A+(i-1)B+(j-1)C: 1 \leq i \leq I, 1 \leq j \leq J\}$

*<строка параметров>* имеет следующую структуру:

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

*<число столбцов I>, <число строк J>* – параметры, определяющие размерность двумерной решётки.

*<вектор A>* – определён тремя координатами корневого вектора **A**.

*<вектор B>* – определён тремя координатами вектора сдвига **B**.

*<вектор C>* – определён тремя координатами вектора сдвига **C**.

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 имён прототипов элементов>* – картограмма имён 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, то есть 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 решетки>*

*<имя прототипа элемента решетки>* – имя элемента решётки, определённое в списке прототипов LISTEL

Ниже приведён пример задания решётки кассет реактора ВВЭР-1000 с помощью генератора сдвигов G2MP

#### Пример

```

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
PARM   15,15                    ; dimension 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 K3 0
L14 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 K3 0 0 0 0 0 0 0

```

## 9.4 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ОПЕРАЦИИ TRANSF

В операции движения тела TRANSF допускаются произвольные движения трехмерного пространства. Кроме того, эти движения могут быть использованы в решетке с прямым перечислением элементов GLTL.

### 9.4.1 Описание конкретного движения

Описание конкретного движения представляет собой набор фрагментов предложения и имеет вид

<код> [G]  $p_1, p_2, \dots$

<код> состоит из одной или двух букв и определяет тип движения (см. ниже). Строчные и заглавные буквы считаются эквивалентными.

Не обязательная буква G (или g) означает трактовку сдвига, как изменения начала координат, что лучше пояснить формулами.

<код> и совокупность параметров  $p_1, p_2, \dots$  определяют ортогональную матрицу  $U$  и вектор  $s$ . При наличии буквы G преобразование  $r \rightarrow r_1$  осуществляется по формуле

$$r_1 = Ur + s.$$

При отсутствии буквы G считается, что однородное преобразование  $U$  делается в системе координат с началом в точке  $s$ . Таким образом, преобразование делается по формуле

$$r_1 = U(r-s) + s = Ur + (s - Us).$$

Число и смысл параметров зависят от значения <код>, что описано ниже.

Каждый параметр есть числовая константа, имя числовой константы или арифметическое выражение из констант и имен.

Общее преобразование имеет вид

$G[G]$   $s$   $U$

где вектор  $s$  задается координатами;  
ортогональная матрица  $U$  записывается по столбцам.  
Вращение вокруг оси X имеет вид

$$RX[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

где  $\varphi$  угол в градусах вращения вокруг оси  $OX$  в положительном направлении.  
Вращение вокруг оси  $Y$  имеет вид

$$RY[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

Вращение вокруг оси  $Z$  имеет вид

$$RZ[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

Отражение от плоскости, проходящей через ось  $OX$  имеет вид

$$MX[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

где  $\varphi$  угол в градусах между осью  $OY$  и пересечением плоскости отражения и плоскости  $OYZ$  (см. рисунок 9.4.1.1).

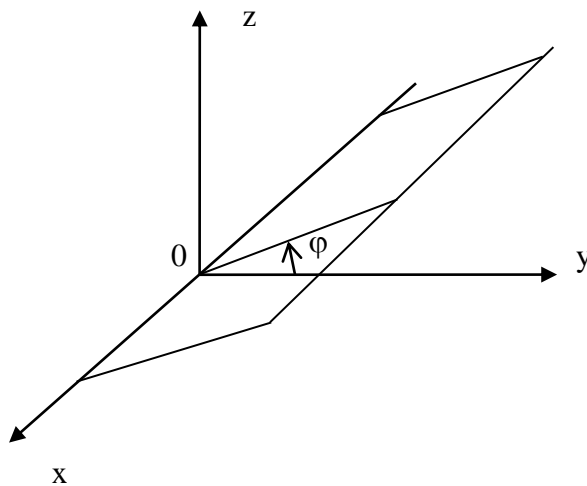


Рисунок 9.4.1.1 – Определение угла  $\varphi$

Отражение от плоскости, проходящей через ось  $OY$  имеет вид

$$MY[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

где  $\varphi$  угол в градусах откладывается от оси  $OZ$  к оси  $OX$ .  
Отражение от плоскости, проходящей через ось  $OZ$  имеет вид

$$MZ[G] \quad \quad \quad \mathbf{s} \quad \quad \quad \varphi$$

где  $\varphi$  угол в градусах откладывается от оси  $OX$  к оси  $OY$ .  
Произвольное вращение задается в виде

$$RA[G] \quad \quad \quad \mathbf{s} \quad \quad \mathbf{a} \quad \quad \varphi$$

где вектор  $\mathbf{a}$  задает направление ( $\mathbf{a}$  определен с точностью до положительного множителя) оси, вокруг которой происходит вращение;

параметр  $\varphi$  задает угол вращения в градусах в положительном направлении.

Например, следующие 3 предложения определяют одно и то же преобразование

$$RX \quad 0 \quad 0 \quad 1 \quad 45$$

```
G    0  0  1  1  0  0  0  cos(45)  sin(45)
      0 -sin(45)  cos(45)
RA   0  0  1  1  0  0  45
```

(как хорошо известно, любая ортогональная матрица с единичным определителем представляет вращение вокруг некой оси).

Отражение вокруг произвольной плоскости, проходящей через начало координат имеет вид

```
MN[G]          s          n
```

где вектор  $n$  ( $n$  определен с точностью до множителя) задает направление, перпендикулярное плоскости отражения.

Отметим, что не любое ортогональное преобразование с определителем -1 можно задать в таком виде. Примером этого служит матрица

```
cos(φ)  -sin(φ)  0
sin(φ)   cos(φ)  0
0         0       -1
```

Поэтому существует возможность задать матрицу ее компонентами.

Для сохранения возможности использования ранее подготовленных входных файлов сохранены две старые формы:

```
R    x0, y0, φ
```

вращение на угол  $\varphi$  вокруг оси, параллельной  $OZ$  и проходящей через точку  $x_0, y_0, 0$ ;

```
M    x0, y0, φ
```

отражение от плоскости параллельной оси  $OZ$ , чье пересечение с плоскостью  $OXY$  образует с осью  $OX$  угол  $\varphi$ , откладываемый от оси  $OX$ .

Очевидно, что это описание эквивалентно описанию

```
MNG    x0, y0, 0    sin(φ), -cos(φ), 0
```

Для описаний типа R и M дополнительная буква G не применяется.

### 9.4.2 Описание движения тела

Описание движения тела имеет вид

```
TRANSF <новое имя> <старое имя> <описание движения>
```

#### Пример

Отметим, что последнее движение есть просто сдвиг.

```
RCC    A    0  0  0    0  4  0    2
TRANSF B  A  RX  0  0  8  45
TRANSF C  A  MNG 0 10 5  -1 1 0
TRANSF D  A  RX  0  0 -9  0
```

### 9.4.3 Преобразования в решетке типа GLTL

При задании параметров решетки типа GLTL каждому элементу решетки соответствовали совокупность фрагментов:

[<изменение номера прототипа>] <положение>

где <положение> представлялось 3 координатами сдвига. Кроме того, положение элемента может быть задано с помощью описаний движений. Перед описанием движения стоит знак "«/»". Для всех типов движения, кроме R и M, параметры соответствуют приведенным выше.

Для R и M, для совместимости с предыдущими версиями после описания движения вводятся еще 3 числа, определяющие сдвиг, который производится после преобразования данного в описании движения.

Например, элемент, положение которого определено выражением

```
/R 2 1 90 5 7 100
```

подвергается следующему преобразованию

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} -(y-1)+2+5 \\ (x-2)+1+7 \\ z+100 \end{pmatrix}$$

Перед символами R и M также ставится символ «/».

#### Пример списка положения элементов

Каждое положение дано в отдельной строке

```
/RAG 0 0 0 1 1 0 90
0 2 10
/RXG 0 0 -10 90
/R 1 2 60 0 0 20
```

#### Пример, пригодный для просмотра в MCU Office.

```
HEAD 3 0 5000
CONT B B B
RCZ CNT 0 0 -10 20 10
END
Z0 CNT /8:8
END
*
LCELL A
RCZ K 0,0,-1 3 2
RPP L -1,1 -0.5,0.5 -1,1
END
K K -L /1:1
L L /-1:2
END
ENDL
*
LCELL B
HEXY K 0,0,-1 3 4
RPP L -1,1 -0.5,0.5 -1,1
END
K K -L /1:1
L L /-1:2
END
ENDL
*
*
LCELL C
```



```

RCC K 0,0,-1 0 0 2 2
RPP L -1,1 -0.5,0.5 -1,1
END
K K -L /1:1
L L /-1:2
END
ENDL
*
LATT GLTL Z0
LISTEL A B C
PARM /3 0,0,0
/RZG 5 0 0 90
/RZ -3 0 0 90
/2 /RZG 0 -4.5 0 15
/3 /RXG 0 0 4 90
FINISH

```

#### 9.4.4 Замечания по изменению типов тел

В геометрическом модуле NCG движения относятся к трем классам

- сдвиги, то есть движения вида  $r \rightarrow r+s$ ;
- движения, сохраняющие вертикальные направления, то есть движения вида  $r = (rx, ry, z) \rightarrow (rxa+ryb+sx, rxc+ryd+sy, z+sy)$  причем матрица  $(a,b;c,d)$  отлична от единичной;
- движение общего вида  $r \rightarrow Ur+s$  причем матрица  $U$  не сохраняет координатный вектор  $ez$  ( $u_{33} < 1$ ).

Любое тело может быть подвергнуто сдвигу.

Тела RPP, UCX, UCY, PLX, PLY не могут быть подвергнуты даже любому преобразованию 2 класса.

Тела RCZ, HEX, UCZ, PLZ не могут быть подвергнуты любому преобразованию 3 класса. Поэтому при осуществлении движения тип тела может меняться, что описывает следующим правилами:

- для тел UCX, UCY и UCZ не может быть применено движения, отличного от сдвига;
- для тел HEX, RCZ, SBOX, SHEX не может быть применено движению 3-го класса;
- тело RPP сохраняет свой тип лишь при сдвиге или перестановке координат вместе со сдвигом. Иначе оно переходит в тело BOX;
- тела PLX, PLY, PLZ при движении отличном от сдвига переходят в тело PLG.

Во всех прочих случаях тип тела не изменяется.

### 9.5 РЕГИСТРАЦИОННЫЕ ГЕОМЕТРИЧЕСКОГО МОДУЛЯ

### ВОЗМОЖНОСТИ

В геометрическом модуле имеется возможность регистрировать попадания частиц в регистрационные области, а также наличие или отсутствие отражений частиц в этих областях.

Под регистрационными областями понимаются материальные зоны, регистрационные зоны, либо регистрационные объекты. В одном расчёте можно регистрировать частицы только по одному типу регистрационных областей.

Для включения этого режима непосредственно перед картой FINISH геометрического модуля ставится одна из трёх карт:

*C=C MUSE*

для регистрации по материальным зонам,

*C=C ZUSE*

для регистрации по регистрационным зонам,

*C=C OUSE*

для регистрации по регистрационным объектам.

При наличии одной из этих карт в файле финальной обработки в разделе статистики геометрического модуля выдаётся список из двух колонок. В первой колонке даны номера соответствующих областей (те же, что используются регистрационным модулем), а во второй – одна из трёх цифр: 0 – в области не было частиц, 1 – в области были частицы, но не происходило отражений, 2 – в области были отражения.

Отражения от специальной плоскости симметрии и от грани контейнера при регистрации не различаются.

## 10 ИСХОДНЫЕ ДАННЫЕ ДЛЯ МОДУЛЯ ИСТОЧНИКОВ

В программе используется модуль источников SRC.

Ввод и обработка фрагмента с данными для модуля производится программными средствами стандартного ввода.

Примитивность задания источников для генерации параметров нейтронов нулевого поколения несущественна, т.к. при решении задач на критичность все функционалы, вычисляемые в программе, являются суммами случайных величин, включающими вклад от всех поколений нейтронов. Вклад каждого поколения в оцениваемую величину имеет порядок  $1/N$  при достаточно больших  $N$ , где  $N$  - число промоделированных поколений. Поэтому вклад первого поколения в результирующую ошибку много меньше статистической ошибки, которая уменьшается как  $(1/N^{0.5})$ .

В качестве исходных данных к модулю можно задать только три пространственные координаты  $x, y, z$  точки рождения нейтронов нулевого поколения. Энергетическое распределение нейтронов нулевого поколения задаётся спектром Уатта с константами, соответствующими спектру деления  $U^{235}$  тепловым нейтроном. Угловое распределение принимается изотропным.

Модуль источника предоставляет возможность выполнять расчёты с одним из двух типов источников: простым или сложным. Для пользователя, который занимается только решением задач на собственное значение, достаточно ознакомиться только с первым разделом данной главы. Разделы 2 и 3 он может при чтении опустить.

### 10.1 ПРОСТОЙ ИСТОЧНИК

#### 10.1.1 Простой точечный источник

Простой точечный источник – это всегда точечный источник с изотропным распределением по углу. Его спектр по энергиям может быть либо дельта-функцией, либо ступенчатой функцией.

Признаком использования простого источника является карта SPNT, которая должна быть первой картой в секции данных для модуля источника.

```
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> - числа определяют границы энергетических интервалов, данных в убывающем порядке. Энергия задаётся в электронвольтах. В результате работы

источника будут вырабатываться энергии, равные серединам интервалов. По умолчанию границы интервалов есть:

```
1.0e7, 6.5e6, 4.0e6, 2.5e6, 1.4e6, 8.0e5, 4.0e5, 2.0e5, 1.0e5,
4.65e4, 2.15e4, 1.0e4, 4.65e3, 2.15e3, 1.0e3, 465.0, 215.0, 100.0,
46.5, 21.5, 10.0, 4.65, 2.15, 1.0, 0.465, 0.215, 0.0
```

<массив из 26 чисел> - массив определяет вероятности энергетических интервалов в виде ненормированных весов. По умолчанию вероятности соответствуют спектру деления  $U^{235}$  тепловым нейтроном:

```
0.016, 0.088, 0.184, 0.270, 0.202, 0.141, 0.061, 0.024, 0.010,
0.003, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
```

<число> - энергия в электронвольтах.

При наличии карты с меткой ENSO спектр состоит из одной дельта-функции, то есть все частицы вырабатываются с одинаковой энергией. Карту с меткой ENSO нельзя использовать в сочетании с картами ESET и SPEC.

Отметим, что единственное ограничение на значение энергии в картах SPEC и ENSO есть неотрицательность, но по смыслу они должны согласовываться с используемым физическим модулем.

### 10.1.2 Простой распределённый источник

При решении задач на Кэф источник используется только для формирования 1-го поколения нейтронов. Поэтому нет нужды использовать 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.

В настоящем подразделе на примерах показано, как задать источник со следующими свойствами.

Энергетическое распределение соответствует стандартному спектру деления урана, упомянутому в предыдущем разделе, как умалчиваемый спектр.

Направление полёта частиц равномерно распределено по сфере.

Точки рождения частиц равномерно распределены по призме или цилиндру или сектора цилиндра с вертикальной образующей.

Для реализации используется описанный ниже сложный источник, поэтому некоторые конструкции могут показаться усложнёнными. Смысл всех используемых карт подробно объяснён в следующих разделах.

Во всех примерах предполагается, что нижнее основание области рождения частиц лежит в плоскости, перпендикулярной оси Z, с аксиальной координатой равной минус 50.0 см, а высота призмы есть 800 см., таким образом, верхнее основание лежит на уровне 750 см. Эти числа могут быть изменены, но естественно, согласованно между собой.

Во всех примерах изменяется только одна карта (возможно со строкой-продолжением), определяющая форму области рождения нейтронов.

#### Пример 1

Область рождения нейтронов есть цилиндр с радиусом 12 см и осью, совпадающей с OZ.

```
NPS      1          ; две карты означают только
PROB     1          ; одну область рождения частиц
*
ANGLEN   SS        ; три карты означают
MDIS     S          ; изотропный
EDIS     S          ; стандартный спектр
```

```

*
TYPE          N                      ; источник нейтронов
RCZ           0,0,-50.0   800.0   12.0 ; область рождения
SNAM          SS                      ; ссылка на спектр
*
REPER         1  0  0                  ; смысл этих двух карт
NOBJ          0                        ; будет описан далее
FINISH

```

### Пример 2

Область рождения нейтронов есть сектор цилиндра с радиусом 12 см и осью, совпадающей с OZ. Проекция основания на плоскость OXY лежит в угле от минус 30 градусов от оси OX до плюс 30 градусов от оси OX.

```

NPS           1                      ; две карты означают только
PROB          1                      ; одну область рождения частиц
*
ANGLEN        SS                      ; три карты означают
MDIS          S                      ; изотропный
EDIS          S                      ; стандартный спектр
*
TYPE          N                      ; источник нейтронов
RCZ           0,0,-50.0   800.0   12.0 ; область рождения
              0.0       -30   30
SNAM          SS                      ; ссылка на спектр
*
REPER         1  0  0                  ; смысл этих двух карт
NOBJ          0                        ; будет описан далее
FINISH

```

### Пример 3

Область рождения нейтронов есть сектор цилиндра с радиусом 12 см и осью, совпадающей с OZ. Проекция основания на плоскость OXY лежит в угле от оси OX до плюс 60 градусов от оси OX.

```

NPS           1                      ; две карты означают только
PROB          1                      ; одну область рождения частиц
*
ANGLEN        SS                      ; три карты означают
MDIS          S                      ; изотропный
EDIS          S                      ; стандартный спектр
*
TYPE          N                      ; источник нейтронов
RCZ           0,0,-50.0   800.0   12.0 ; область рождения
              0.0       0   60
SNAM          SS                      ; ссылка на спектр
*
REPER         1  0  0                  ; смысл этих двух карт
NOBJ          0                        ; будет описан далее
FINISH

```

### Пример 4

Область рождения нейтронов есть призма с квадратным основанием размером 25x25 см<sup>2</sup>, центр которого лежит на оси OZ. Таким образом, координаты призмы удовлетворяют неравенствам

$$|x| \leq 12.5 \quad |y| \leq 12.5 \quad -50.0 \leq z \leq 750.0.$$

```

NPS          1          ; две карты означают только
PROB         1          ; одну область рождения частиц
*
ANGLEN       SS        ; три карты означают
MDIS         S         ; изотропный
EDIS         S         ; стандартный спектр
*
TYPE         N          ; источник нейтронов
RPP          -12.5  12.5  -12.5  12.5
            -50.0  750.0          ; область рождения
SNAM         SS        ; ссылка на спектр
*
REPER        1  0  0          ; смысл этих двух карт
NOBJ         0          ; будет описан далее
FINISH

```

### Пример 5

Область рождения нейтронов есть с призма шестиугольным основанием размером под ключ 23.6 см., центр которого лежит на оси OZ. Призма повёрнута так, что две стороны основания параллельны оси OX.

```

NPS          1          ; две карты означают только
PROB         1          ; одну область рождения частиц
*
ANGLEN       SS        ; три карты означают
MDIS         S         ; изотропный
EDIS         S         ; стандартный спектр
*
TYPE         N          ; источник нейтронов
HEXY         0  0  -50.0  800  23.6          ; область рождения
SNAM         SS        ; ссылка на спектр
*
REPER        1  0  0          ; смысл этих двух карт
NOBJ         0          ; будет описан далее
FINISH

```

### Пример 6

Область рождения нейтронов есть призма с шестиугольным основанием размером под ключ 23.6 см., центр которого лежит на оси OZ. Призма повёрнута так, что две стороны основания параллельны оси OY.

```

NPS          1          ; две карты означают только
PROB         1          ; одну область рождения частиц
*
ANGLEN       SS        ; три карты означают
MDIS         S         ; изотропный
EDIS         S         ; стандартный спектр
*
TYPE         N          ; источник нейтронов
HEXY         0  0  -50.0  800  23.6  90      ; область рождения
SNAM         SS        ; ссылка на спектр
*
REPER        1  0  0          ; смысл этих двух карт
NOBJ         0          ; будет описан далее
FINISH

```

### Пример 7

Вся область рождения нейтронов есть с призма шестиугольным основанием размером под ключ 23.6 см., центр которого лежит на оси OZ. Призма повернута так, что две стороны основания параллельны оси OX. Однако из этой призмы взята лишь одна треугольная призма с проекцией нижнего основания на OXY в виде прямоугольного треугольника с углом в 30 градусов в нуле координат и гипотенузой, лежащей на оси OX.

```
NPS          1          ; две карты означают только
PROB         1          ; одну область рождения частиц
*
ANGLEN       SS        ; три карты означают
MDIS         S         ; изотропный
EDIS         S         ; стандартный спектр
*
TYPE         N          ; источник нейтронов
HEXY         0 0 -50.0   800   23.6
0 0 0        ; область рождения
SNAM         SS        ; ссылка на спектр
*
REPER        1 0 0      ; смысл этих двух карт
NOBJ         0          ; будет описан далее
FINISH
```

### Пример 8

Вся область рождения нейтронов есть призма с шестиугольным основанием размером под ключ 23.6 см., центр которого лежит на оси OZ. Призма повернута так, что две стороны основания параллельны оси OX. Однако из этой призмы взята лишь одна треугольная призма с проекцией нижнего основания на OXY в виде равностороннего треугольника с углом в нуле координат и одной стороной, лежащей на оси OX.

```
NPS          1          ; две карты означают только
PROB         1          ; одну область рождения частиц
*
ANGLEN       SS        ; три карты означают
MDIS         S         ; изотропный
EDIS         S         ; стандартный спектр
*
TYPE         N          ; источник нейтронов
HEXY         0 0 -50.0   800   23.6
0 0 1        ; область рождения
SNAM         SS        ; ссылка на спектр
*
REPER        1 0 0      ; смысл этих двух карт
NOBJ         0          ; будет описан далее
FINISH
```

## 10.2 СЛОЖНЫЙ ИСТОЧНИК

Вероятности всюду задаются в виде ненормированных весов. Ниже под термином веса вероятностей понимается набор величин, отличающихся от вероятностей общим для них множителем, то есть:

$$P_i = w_i / (w_1 + w_2 + \dots + w_n).$$

Стандартную единицу измерения энергии, эВ, можно заменить с помощью предложения:

UNIT <размерность>

<размерность> - величина, принимающая символьные (текстовые) значения MeV, keV, eV (с сохранением регистра).

### Пример

UNIT MeV

### Предложение

PRISOU

включает печать внутренних массивов модуля источников.

Некоторые предложения объединены в секции. Порядок секций строго фиксирован и дан в следующей схеме:

<заголовок>

<веса примитивных источников>

<секции описания спектра>

[<секции времён рождения>]

<секции примитивных источников>

FINISH

## 10.2.1 Заголовок

Заголовок состоит из одной карты:

NPS <число> [BIAS [M]]

<число> - натуральное целое число равное числу примитивных источников.

BIAS – характеризует способ задания смещения выборки источника:

- отсутствие BIAS означает, что все частицы в источнике имеют единичный вес;
- BIAS - смещения выборки источников; интенсивности и веса вероятностей задаются попеременно:  $v_1, f_1, v_2, f_2, \dots$ , где  $v_i$  означает интенсивность, а  $f_i$  - вес вероятности;
- BIAS M - смещения выборки источников; интенсивности и веса вероятностей задаются последовательно:  $\{v_i\}, \{f_i\}$ , где  $v_i$  означает интенсивность, а  $f_i$  - вес вероятности.

### Пример 1

NPS 4

### Пример 2

NPS 22 BIAS M

## 10.2.2 Веса примитивных источников

Веса вероятностей срабатывания примитивных источников задаются картой PROB.

PROB <массив>

<массив> - массив положительных чисел, содержащий веса вероятностей срабатывания примитивных источников. При отсутствии смещения выборки массив имеет длину  $N_s$ . При наличии смещения выборки –  $2*N_s$ . Отметим, что вес частиц  $i$ -го источника пропорционален  $v_i/f_i$ .

### Пример

PROB            0.2, 0.2, 0.5, 0.1

## **10.2.3 Секция описания спектра**

Описание спектра задаёт правило генерации величин энергии, обозначаемой  $E$  и величин косинуса угла между вектором скорости частицы и реперным вектором, обозначаемым  $\mu$ . При каждой генерации места рождения частицы вырабатывается и реперный вектор, зависящий только от координат рождения частицы.

Секция описания спектра состоит из двух или трёх частей, далее описываемых по порядку, в котором они должны быть заданы. Внутри частей порядок произволен.

### **10.2.3.1 Заголовок спектра**

Заголовок спектра состоит из одной карты:

*ANGLEN* <имя спектра> [*<зависимость>* [*L*]]

<имя спектра> - служит для ссылок на него. Оно есть идентификатор, содержащий не более 4 заглавных латинских букв и цифр.

<зависимость>:

– отсутствие параметра <зависимость> означает независимость  $E$  и  $\mu$  и эквивалентно конструкции *NO*;

– *NO* - означает независимость  $E$  и  $\mu$ ;

– *МТОЕ* - означает зависимость  $E$  от  $\mu$ ;

– *ЕТОМ* - означает зависимость  $\mu$  от  $E$ ;

– *FUNC* означает функциональную зависимость;

*L* - означает, что переход линейный, а по умолчанию переход дискретный, употребляется только совместно с *МТОЕ* и *ЕТОМ*.

### Пример

*ANGLEN*        *SPN2*            *ЕТОМ*            *L*

### **10.2.3.2 Общие характеристики распределений**

*MDIS* <тип распределения  $\mu$ > [*<число>* [*BIAS* [*M*]]]

*EDIS* <тип распределения  $E$ > [*<число>* [*BIAS* [*M*]]]

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

– *D* - дискретное распределение;

– *L* - линейное распределение;

– *S* - означает 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 урана 235 тепловым нейтроном;

– *F* <имя нуклида> – спектр деления нуклида с параметрами, взятыми из библиотеки *BNAB/MCU*, <имя нуклида> - это идентификатор, содержащий не более 4 символов, как в файле *DEFAULT.PHY*.

*S* и *F* допустимы, только если величины  $\mu$  и  $E$  независимы.

Параметры *D* и *L* обязательно требуют наличия параметра <число> и могут иметь параметр *BIAS* или *BIAS M*.



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

*BIAS* – характеризует способ задания смещения выборки энергии или угла:

- отсутствие *BIAS* означает, что все частицы имеют единичный вес;
- *BIAS* - смещения выборки; интенсивности и веса вероятностей задаются попеременно:  $v_1, f_1, v_2, f_2, \dots$ , где  $v_i$  означает интенсивность, а  $f_i$  - вес вероятности;
- *BIAS M* - смещения выборки; интенсивности и веса вероятностей задаются последовательно:  $\{v_i\}, \{f_i\}$ , где  $v_i$  означает интенсивность, а  $f_i$  - вес вероятности.

В настоящей версии смещение допускается только для независимых величин  $E$  и  $\mu$ .

При функциональной зависимости (параметр *FUNC* в строке *ANGLEN*) параметры в строках *EDIS* и *MDIS* должны быть одинаковы.

### Пример

*MDIS*    *D*    19

### Пример

Задание спектра деления  $^{252}\text{Cf}$  из библиотеки *BNAB*.

*EDIS*            *F*            *CF52*

### **10.2.3.3 Дополнительные данные для дискретного и линейного распределений**

Если для случайной величины  $E$  (энергия частицы) или  $\mu$  (косинус угла между направлением полёта и реперным вектором) используется линейное или дискретное распределение (параметры *L* или *D* в картах *EDIS* и *MDIS*, соответственно), то для неё необходимо задать два массива: массив для узлов возможных значений этой величины и массив вероятностей выборки значений этой величины.

Обозначим параметр *<число>* в картах *EDIS* и *MDIS* через *NE* и *NM*, соответственно.

Массивы узлов вводятся картами вида:

*EMES*    *<массив значений узлов>*

*MMES*    *<массив значений узлов>*

*<массив значений узлов>* - действительные числа, определяющие узлы возможных значений величины  $E$  в карте *EMES* или узлы возможных значений величины  $\mu$  в карте *MMES*. Значения узлов всегда вводят в порядке возрастания. Значения  $E$  всегда неотрицательны, значения  $\mu$  могут меняться от  $-1$  до  $1$  (как косинус угла).

Массивы вероятностей вводятся с помощью карт:

*EPRO*    *<массив значений вероятностей>*

*MPRO*    *<массив значений вероятностей>*

*<массив значений вероятностей>* - действительные числа, определяющие значения вероятностей для величины  $E$  в карте *EPRO* или значения вероятностей для величины  $\mu$  в карте *MPRO*.

Данные, содержащиеся во всех этих массивах, зависят от характеристик спектра, заданного в картах *ANGLEN*, *EDIS*, *MDIS*.

Возможны следующие варианты задания параметров в картах дополнительных данных для дискретного и линейного распределений.

### 10.2.3.3.1 Независимые случайные величины с дискретным распределением (параметр NO или отсутствие этого параметра в карте ANGLEEN, параметр D в картах EDIS, MDIS)

Длины *<массива значений узлов>* и *<массива значений вероятностей>* для каждой случайной величины одинаковы и равны числу NE из карты EDIS или NM из карты MDIS, соответственно.

*<массив значений узлов>* - это всё множество значений, которые может принимать случайная величина.

Для энергии E *<массив значений узлов>* это последовательность:  $E_1 < E_2 < E_3 \dots < E_{NE}$ , где  $E_1 \geq 0$ .

Для  $\mu$  *<массив значений узлов>* это последовательность:  $\mu_1 < \mu_2 < \dots < \mu_{NM}$ , где  $\mu_1 \geq -1$ ,  $\mu_{NM} \leq 1$ .

Если отсутствует смещение (нет параметра BIAS в картах EDIS, MDIS), то *<массив значений вероятностей>* состоит из NM (или NE) действительных чисел  $p_1, p_2 \dots p_N$ , где  $p_k$  есть ненормированная вероятность для величины  $\mu$  (или E) принять значение  $\mu_k$  (или  $E_k$ ). При вводе программа производит нормировку вероятностей.

Если есть смещение (параметр BIAS в картах EDIS, MDIS), то *<массив значений вероятностей>* состоит из  $2 \cdot NM$  (или  $2 \cdot NE$ ) действительных чисел  $f_1, f_2 \dots f_N, p_1, p_2 \dots p_N$ , где  $f_k$  есть ненормированная вероятность для величины  $\mu$  (или E) принять значение  $\mu_k$  (или  $E_k$ ) для реально существующего источника, а  $p_k$  есть ненормированная вероятность, с которой программа будет генерировать это значение. Чтобы скомпенсировать расхождение между  $p_k$  и  $f_k$ , при этом будет вырабатываться весовой множитель  $w = (f_k/p_k)(S_P/S_F)$  где  $S_P$  – сумма  $p_k$ , а  $S_F$  – сумма  $f_k$ . Вес генерируемой частицы будет умножен на  $w$ .

Порядок задания величин  $f_k$  и  $p_k$  определяется наличием или отсутствием параметра M в картах EDIS, MDIS:

- если M отсутствует, то задаётся последовательность:  $f_1, p_1, f_2, p_2, \dots, f_N, p_N$ ;
- если M присутствует, то задаётся последовательность:  $f_1, f_2, \dots, f_N, p_1, p_2, \dots, p_N$ .

### 10.2.3.3.2 Независимые случайные величины с линейным распределением (параметр NO или отсутствие этого параметра в карте ANGLEEN, параметр L в картах EDIS, MDIS)

Длина *<массива значений узлов>* для каждой случайной величины равна числу NE+1 из карты EDIS или NM+1 из карты MDIS, соответственно.

*<массив значений узлов>* содержит все значения концов примыкающих друг к другу отрезков, и множество возможных значений есть отрезок от минимального узла до максимального.

Для энергии E *<массив значений узлов>* есть последовательность:  $E_0 < E_1 < \dots < E_{NE}$ , где  $E_0 \geq 0$ .

Для  $\mu$  *<массив значений узлов>* есть последовательность:  $\mu_0 < \mu_1 < \dots < \mu_{NM}$ , где  $\mu_0 \geq -1$ ,  $\mu_{NM} \leq 1$ .

Длина *<массива значений вероятностей>* для каждой случайной величины равна числу NE из карты EDIS или NM из карты MDIS, соответственно.

*<массив значений вероятностей>* определяется так же, как в случае независимых случайных величин с дискретным распределением в зависимости от наличия или отсутствия смещения (BIAS или BIAS M). Однако в линейном случае  $p_k$  есть вероятность того, что вырабатываемая величина  $\mu$  (или E) лежит в отрезке  $[\mu_{k-1}, \mu_k]$  (или  $[E_{k-1}, E_k]$ ) и конкретное значение из этого отрезка далее выбирается по равномерному распределению.

### 10.2.3.3.3 Дискретная зависимость E от $\mu$ (параметр MTOE в карте ANGLEEN)

*<массив значений узлов>* определяется в карте MMES так же, как в случае независимых случайных величин с дискретным распределением в зависимости от наличия параметра L или D в карте MDIS (имеет длину NM+1 или NM).

<массив значений узлов> определяется в карте EMES так же, как в случае независимых случайных величин с дискретным распределением в зависимости от наличия параметра L или D в карте EDIS (имеет длину NE+1 или NE).

<массив значений вероятностей> определяется в карте MPRO, как в случае независимых случайных величин с дискретным распределением (имеет длину NM).

<массив значений вероятностей> определяется в карте EPRO и является матрицей условных вероятностей. Эта матрица имеет NE столбцов и NM строк. При генерации величины  $\mu$  вырабатывается случайный индекс k, означающий k-ый узел или отрезок в зависимости от распределения величины  $\mu$ . Этот же индекс определяет строку матрицы, которая даёт вероятности для выработки величины E по дискретному или линейному алгоритму в зависимости от параметра в карте EDIS.

#### **10.2.3.3.4 Линейная зависимость E от $\mu$ (параметры MTOE и L в карте ANGLEN)**

Если в карте ANGLEN вводится параметр L, тогда в карте MDIS также должен быть параметр L.

<массив значений узлов> определяется в карте MMES так же, как в случае независимых случайных величин с дискретным распределением с учётом параметра L в карте MDIS (имеет длину NM+1).

<массив значений узлов> определяется в карте EMES так же, как в случае независимых случайных величин с дискретным распределением в зависимости от наличия параметра L или D в карте EDIS (имеет длину NE+1 или NE).

<массив значений вероятностей> определяется в карте MPRO, как в случае независимых случайных величин с дискретным распределением (имеет длину NM).

<массив значений вероятностей> определяется в карте EPRO и является матрицей условных вероятностей. Эта матрица имеет NE столбцов и NM+1 строк. При генерации  $\mu$  вырабатывается случайный индекс k, означающий k-ый отрезок, а также вырабатывается случайное равномерно распределённое на отрезке [0,1] число  $\xi$ . Значение  $\mu$  получается как взвешивание двух узлов, а строки вероятностей для получения E - как взвешивание двух строк матрицы k-1 и k с весами  $(1-\xi)$  и  $\xi$ .

#### **10.2.3.3.5 Дискретная зависимость $\mu$ от E (параметр ETOM в карте ANGLEN)**

Аналогично заданию данных для дискретной зависимости E от  $\mu$ .

#### **10.2.3.3.6 Линейная зависимость $\mu$ от E (параметры ETOM и L в карте ANGLEN)**

Аналогично заданию данных для линейной зависимости E от  $\mu$ .

#### **10.2.3.3.7 Функциональная зависимость (параметр FUNC в карте ANGLEN)**

Обе величины должны иметь одинаковый тип распределения – дискретный или линейный (параметр D или L в картах MDIS, EDIS).

Должны присутствовать карты MMES и EMES для ввода узлов. <массив значений узлов> определяется так же, как и в случае независимых случайных величин с дискретным распределением.

Для E и  $\mu$  используется один и тот же <массив значений вероятностей>. Этот массив определяет значение как E так и  $\mu$ . Его длина определяется в картах EDIS и MDIS одним из параметров NM или NE, которые должны быть равны ( $NM = NE$ ). Этот массив вводится с помощью любой из карт EPRO или MPRO.

#### **Пример**

В данном примере истинные вероятности на разных линиях спектра мало различаются, но при моделировании будут преимущественно выбираться две линии с наименьшими энергиями.

MDIS	S					
EDIS	D	5	BIAS			
EMES	20	45.5	100	120	130.9	
EPRO		1.2	4			
		1.3	2			
		1.5	1			
		1.3	1			
		1.2	1			

## 10.2.4 Секция времён рождения

В расчётах стационарных задач время рождения частиц вообще не используется, и данная секция в данных для источника может отсутствовать.

Секция времён рождения состоит из трёх предложений, описываемых в том порядке, в каком они должны задаваться.

*TIME* <имя распределения времени рождения> [*L*]

<имя распределения времени рождения> служит для ссылок на него. Это идентификатор из не более чем 4 заглавных латинских букв и цифр.

*L* - означает линейное распределение. В противном случае распределение дискретно.

*TMES* <массив 1>

<массив 1> - значения узлов времени для распределения времени рождения, заданные в секундах.

*TPRO* <массив 2>

<массив 2> - вероятности, заданные как вес. Если распределение дискретное, то размер <массива 1> должен совпадать с размером <массива 2>, а если распределение линейное то <массив 1> содержит на один элемент больше, чем <массив 2>.

### Пример

```
TIME      T2      L
TMES 0, 0.01, 0.02, 0.08
TPRO 0.5 0.8 0.3
```

## 10.2.5 Секции примитивных источников

Секция примитивного источника состоит из двух, трёх или четырёх частей, перечисляемых в порядке ввода.

Порядковый номер секции в полном описании источника есть номер данного примитивного источника, поэтому общее число секций этого типа должно совпадать с числом примитивных источников, определённом в секторе заголовка.

### 10.2.5.1 Первая часть

Первая часть состоит из одной строки вида

*TYPE* <тип частицы> [<имя распределения времени рождения>]

<тип частицы>:

- *N* - означает источник нейтронов;
- *PH* - означает источник фотонов;
- *EL* - означает источник электронов;

– *PO* - означает источник позитронов.

*<имя распределения времени рождения>* - задано в карте TIME, при наличии данного параметра будет использовано распределение с заданным именем.

В случае отсутствия параметра *<имя распределения времени рождения>* у всех генерируемых частиц время рождения будет 0.

### 10.2.5.2 Вторая часть

Вторая часть состоит из четырёх предложений.

#### 10.2.5.2.1 Описание контейнера источника

Контейнер – это область пространства, в которой для данного примитивного источника разыгрываются точки рождения. Контейнер определяется типом и параметрами, которые задаёт предложение вида:

*<имя тела контейнера>* *<массив параметров>*

*<имя тела контейнера>* - задаёт тело контейнера, оно может быть RPP, либо RCZ, либо SPH, либо RCZD, либо HEXU

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

*RPP* - означает прямоугольный параллелепипед, ориентированный по осям координат (см. рисунок 10.2.5.2.1.1).  $\{(x_i): a_i \leq x_i \leq b_i\}$ , равенство  $a_i = b_i$  возможно только для одного индекса  $i$ .

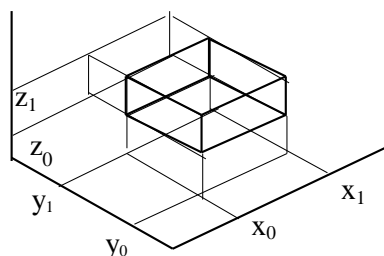


Рисунок 10.2.5.2.1.1 – Прямоугольный параллелепипед, ориентированный по осям координат

*<массив параметров>* есть  $x_0, x_1, y_0, y_1, z_0, z_1$ .

*RCZ* - означает сектор цилиндра (см. рисунок 10.2.5.2.1.2),  $\{(\rho, \varphi, z): r \leq \rho \leq R, \varphi_H \leq \varphi \leq \varphi_K, 0 \leq z \leq h\} + (x_0, y_0, z_0)$ . 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 либо допускается трёхмерная область, либо сектор круга (в частности полный круг), либо кусок цилиндрической поверхности

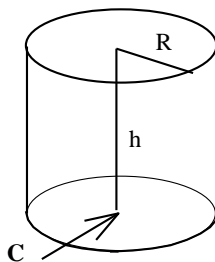


Рисунок 10.2.5.2.1.2 – Сектор цилиндра

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

Вектор  $C$  определяется параметрами  $x_0, y_0, z_0, h$  – высота цилиндра,  $R$  – радиус цилиндра. Параметры  $\varphi_H, \varphi_K$  задаются в градусах. Отсутствие параметров  $\varphi_H, \varphi_K$  означает полный круг:  $\varphi_H = -180^\circ, \varphi_K = 180^\circ$ .

Отсутствие параметра  $r$  означает  $r = 0$ .

$SPH$  - означает сектор шара (см. рисунок 10.2.5.2.1.3).  $\{(\rho, \varphi, \theta): r \leq \rho \leq R, \varphi_H \leq \varphi \leq \varphi_K, 0 \leq \theta \leq \pi\} + (x_0, y_0, z_0)$ . Точка  $(x_0, y_0, z_0)$  - центр шара;  $\pi \leq \varphi_H < \varphi_K \leq \pi; r \leq R$ , это часть шара или сферы;

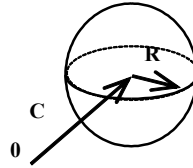


Рисунок 10.2.5.2.1.3 – Сектор шара

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

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

$RCZD$  - означает цилиндр с внутренним распределением вероятностей рождения.

В настоящее время реализовано лишь независимое по высоте и радиусу азимутально-симметричное распределение. Оба независимых распределения могут быть дискретными и кусочно-линейными.

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

Первые три означают координаты «центра нижнего основания».

При генерации вырабатываются случайные числа  $h$  и  $r$ , которые есть внутренние координаты цилиндра вместе с равномерно распределённым по окружности углом  $\psi$ . Координаты точек рождения есть

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

Случайная величина  $h$  может принимать при желании и отрицательные значения, в этом случае  $z_0$  не является центром нижнего основания.

Величина  $r$  всегда неотрицательна.

$N_h$  - целое, отличное от нуля число, означает тип распределения  $h$ .

– Если  $N_h < 0$ , то распределение дискретно,  $|N_h|$  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, означает тип распределения  $r$ .

– Если  $N_r < 0$ , то это соответствует дискретному распределению по  $r$ , с  $|N_r|$  возможными значениями величины  $r$ .

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

$HEXY$  - означает сектор правильной шестигранной вертикально ориентированной призмы (см. рисунок 10.2.5.2.1.4).

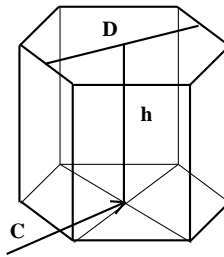


Рисунок 10.2.5.2.1.4 – Сектор правильной шестигранной вертикально ориентированной призмы

<массив параметров> есть  $x_0, y_0, z_0, h, D [\varphi [Sb, Se]]$

$x_0, y_0, z_0$  – координаты центра нижнего основания;  $h$  – высота,  $D$  – размер под ключ. Если прочие параметры опущены, то полная шестигранная призма расположена, так что две её грани параллельны плоскости  $0XZ$ . Параметр  $\varphi$  – означает поворот данной призмы вокруг оси  $Z$  на  $\varphi$  градусов.

Номера треугольников  $Sb, Se$  задаются в соответствии с нумерацией, приведённой на рисунке 10.2.5.2.1.5. Выделение секторов производится по направлению против часовой стрелки от меньшего номера к большему, поэтому используется как положительная, так и отрицательная нумерация.

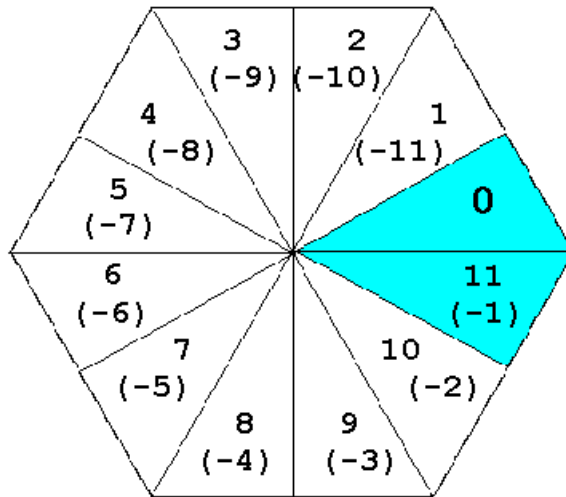


Рисунок 10.2.5.2.1.5 – Нумерация треугольников в НЕХУ

Например, для закрашенной на рисунке части шестигранной призмы можно использовать описание

НЕХУ X, Y, Z H D 0 -1, 0

Для оставшейся не закрашенной части можно использовать описание

НЕХУ X, Y, Z H D 0 1, 10

Описание

НЕХУ X, Y, Z H D 60 9, 10

выделяет часть шестигранной призмы отличную от закрашенного, однако, за счёт поворота на 60 градусов распределение генерируемых частиц будет таким же.

### 10.2.5.2.2 Правило выработки реперного вектора $v_p$

Правило выработки реперного вектора определяется строкой вида

*REPER* <массив параметров>

<массив параметров>:

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

– /P и 3 действительных числа, которые задают координаты характерной точки системы **a** (положение индикатора). В этом случае при рождении частицы реперный вектор в точке рождения определяется по формуле  $v_p = (\mathbf{a} - \mathbf{x}) (\mathbf{a} - \mathbf{x}, \mathbf{a} - \mathbf{x})^{-0.5}$ ;

– число 1 либо -1, которое определяет знак нормали, то есть sgn. При каждой генерации точки рождения примитивный источник вырабатывает зависящий от этой точки вектор нормали к контейнеру  $v_n$ . Реперный вектор определяется по формуле  $v_p = \text{sgn} * v_n$ .

Нормаль к контейнеру определяется по следующим формулам.

Пусть для RPP  $a_i = b_i$  для некоторого  $i$ . В этом случае  $v_n$  есть  $i$ -ый координатный вектор. Если  $a_i < b_i$  для всех  $i$  и задан sgn или  $a_i = b_i$  для 2 или 3 значений  $i$ , то происходит аварийная остановка при вводе.

Для RCZ при  $h = 0$   $v_n = (0, 0, 1)$ . В остальных случаях при генерации точки

$$(x_0, y_0, z_0) + (\rho \cos(\varphi), \rho \sin(\varphi), z)$$

одновременно происходит генерация нормали по следующей формуле

$$v_n = .(\cos(\varphi), \sin(\varphi), 0).$$

Для SPH при генерации точки

$$(x_0, y_0, z_0) + (\rho \cos(\varphi) \sin(\theta), \rho \sin(\varphi) \sin(\theta), \rho \cos(\theta))$$

происходит генерация нормали по следующей формуле

$$v_n = .(\cos(\varphi) \sin(\theta), \sin(\varphi) \sin(\theta), \cos(\theta)).$$

В вырожденном случае вектор  $v_n$  не определён.

Для HEXY автоматической выработки нет.

Для RCZD  $v_n$  всегда равен  $(\cos \varphi, \sin \varphi, 0)$ .

### 10.2.5.2.3 Имя используемого спектра

Карта указывает, какой спектр будет использован для данного примитивного источника.

*SNAM* <имя спектра>

<имя спектра> - ссылка на вышеописанный спектр (его заголовок - карта ANGLN). Оно есть идентификатор, содержащий не более четырёх заглавных латинских букв и цифр, начинающихся с буквы.

### 10.2.5.2.4 Количество регистрационных объектов

Карта указывает количество регистрационных объектов задания сложной для формы области рождения частиц

*NOBJ* <число>



<число> - неотрицательное целое число, которое указывает количество регистрационных объектов, используемых для задания области рождения.

### 10.2.6 Дополнительные данные для RCZD контейнера

Дополнительные данные вводятся для описания функции распределения рождения частиц внутри контейнера типа RCZD, для контейнеров всех других типов точки рождения считаются распределёнными равномерно.

Если контейнер задаётся предложением вида RCZD  $x_0, y_0, z_0, N_h, N_r$ , то кроме этой карты должны быть введены 4 предложения в том порядке, в каком они описываются ниже. 2 предложения определяют распределение  $h_i$  и зависимость весов от  $h$ .

HGRI <массив чисел>

<массив чисел> - массив действительных чисел, задающий положения  $h_i$

Если  $N_h < 0$ , то вводятся  $|N_h|$  возможных значений высоты:  $h_1 < h_2 < \dots < h_M$ ; ( $M = |N_h|$ )

Если  $N_h > 0$ , то вводятся концы  $|N_h|$  отрезков:  $h_0 < h_1 < h_2 < \dots < h_M$ . Внутри каждого из этих отрезков случайная величина равномерно распределена.

HPC <массив чисел>

<массив чисел> - массив из пар чисел  $(v_i, f_i)$ . Числа  $v_i$  означают реальные интенсивности моделируемого источника. Числа  $f_i$  означают с точностью до множителя вероятности, с которыми будут генерироваться частицы в программе.

Два следующих предложения определяют распределение  $r_i$  и зависимость весов от  $r$ .

RGRI <массив чисел>

<массив чисел> - массив действительных чисел, задающий положения  $r_i$ .

Если  $N_r < 0$ , то вводятся  $|N_r|$  возможных значений радиуса  $r_1 < r_2 < \dots < r_N$ . ( $N = |N_r|$ ).

Если  $N_r > 0$ , то вводятся  $|N_r|$  возможных концов отрезков:  $r_0 < r_1 < \dots < r_N$ .

Отметим, что для всех значений  $i$   $r_i \geq 0$ .

RPC <массив чисел>

<массив чисел> - массив из пар чисел  $(v_i, f_i)$ . Числа  $v_i$  означают реальные интенсивности моделируемого источника. Числа  $f_i$  означают с точностью до множителя вероятности, с которыми будут генерироваться частицы в программе.

### 10.2.7 Ввод данных для регистрационных объектов

Если число регистрационных объектов, заданных картой NOBJ, не равно 0, то вводится ещё одна часть, состоящая из двух предложений, задающих номера этих объектов и соответствующие веса.

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 данном контейнере, не попадают ни в один объект из списка, то они отбрасываются и генерируется новая точка рождения. Использование объектов одновременно позволяет ввести пространственную зависимость весовых множителей. Вместе со списком объекта вводится список весовых множителей той же длины. Если точка рождения оказывается в  $k$ -ом объекте по списку, то для неё используется  $k$ -ый весовой множитель из списка множителей.

Порядок ввода этих предложений произволен.

Карта задания номеров объектов имеет вид:

*LOBJ* <массив чисел>

<массив чисел> - массив из  $M$  целых чисел, где  $M$  – количество регистрационных объектов, заданное в карте *NOBJ*.

Карта задания весов объектов имеет вид:

*WOBJ* <массив чисел>

<массив чисел> - массив из  $M$  действительных чисел, где  $M$  – количество регистрационных объектов, заданное в карте *NOBJ*.

Так как эти величины по смыслу являются весами, а не вероятностями, то никаких перенормировок не происходит. Использование регистрационных объектов и их весов требуется только при задании очень сложных источников.

### 10.2.8 Описание элементов решёток

Описание элемента решётки отличается от описания примитивного источника только первой строкой, которая имеет вид

*ELEM* <имя> <тип частицы> [<имя распределения времени рождения>]

<имя> - служит для ссылок на данный элемент решётки. Оно есть идентификатор, содержащий не более 4 латинских букв и цифр и начинающийся с буквы (прописные и заглавные буквы различаются).

<тип частицы>:

- *N* - означает источник нейтронов;
- *PH* - означает источник фотонов;
- *EL* - означает источник электронов;
- *PO* - означает источник позитронов.

<имя распределения времени рождения> - задано в карте *TIME*, при наличии данного параметра будет использовано распределение с заданным именем.

В случае отсутствия параметра <имя распределения времени рождения> у всех генерируемых частиц время рождения будет 0.

Далее все строки описываются как в примитивном источнике. Следует помнить, что пространственные координаты в элементе ещё не окончательные, так как элементы решётки будут помещаться в конкретных позициях решётки, которые определяются векторами сдвига.

### 10.2.9 Описание решётки

Собственно решётка определяется множеством, состоящим из четвёрок символов следующего вида:

{(T, I, F, S)},

где T определяет ссылку на элемент, который будет использован в данной позиции решётки, в качестве примитивного источника;

I определяет интенсивность этого источника;

F есть вероятность срабатывания источника в данной позиции решётки, при условии, что решётка выбрана как примитивный источник верхнего уровня. Очевидно, что вес выработанной частицы должен быть пропорционален  $I/F$ ;

S обозначает вектор сдвига элемента решётки на данную позицию.

Решётка определяет набор позиций источника. Каждой позиции соответствует четвёрка (T,I,F,S). Во время моделирования используется модифицированная четвёрка

( $T, v, p, S$ ), где  $v$  - вес, а  $p$  - условная вероятность рождения частицы в данной позиции. Вероятность  $p$  отличается от числа  $F$  лишь нормировкой на единицу.

Число  $v$  в случае включённой нормализации NORM ON определяется по формуле  $v = I/(p \cdot G)$ , где  $G$  есть сумма чисел  $I$  по всем позициям решётки. При отключённой нормализации NORM OFF  $v = I/p$ .

Следует сделать пояснение относительно весов. Источник - элемент решётки сам может выработать отличный от единицы вес частицы  $W$ . В этом случае окончательный вес частицы будет равен произведению  $v \cdot W$ .

Описание решётки начинается со строки

LATT <имя генератора сдвигов>

<имя генератора сдвигов> - S2AR

Структура дальнейших строк зависит от используемого генератора сдвигов. В настоящее время реализован только один такой генератор, называющийся S2AR, исходные данные для которого будут описаны. Этот генератор соответствует правильной параллелограммной решётке, подобной генерируемой генератором сдвигов G2AR в модуле NCG. Входные данные состоят из следующих строк.

BOUN < $m_i, n_i$ > < $m_j, n_j$ >

< $m_i, n_i$ > - пара целых чисел, которые определяют диапазон изменения первого индекса  $i$ ,

< $m_j, n_j$ > - пара целых чисел, которые определяют диапазон изменения второго индекса  $j$ .

ROOT <массив из 3 чисел> <массив из 3 чисел> < массив из 3 чисел >

<массив из 3 чисел> - это параметры, которые задают три вектора **A**, **B**, **C** (см. рисунок 10.2.9.1). Векторы сдвигов решётки образуются по правилу:

$$S = A + i \cdot B + j \cdot C \quad (i = m_i, m_i+1, \dots, n_i) \quad (j = m_j, m_j+1, \dots, n_j).$$

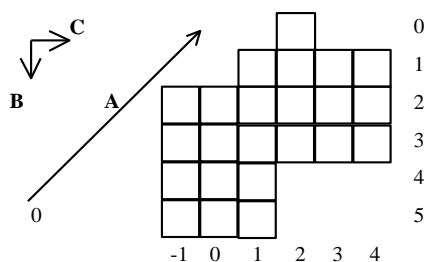


Рисунок 10.2.9.1 – Задание решетки примитивных источников

Далее следуют  $n_j - m_j + 1$  предложений, образующих картограмму типов элементов. Метка  $j$ -го предложения должна иметь вид  $E \langle j \rangle$ , причём для неотрицательных индексов используется трёхзначное представление с лидирующими нулями, если надо, например, E000, E004, E021. Для отрицательных индексов используется двузначное представление с лидирующим нулём, а во второй позиции стоит знак минус, например, E-02, E-26.

В каждом предложении должно быть ровно  $n_i - m_i + 1$  фрагментов, каждый из которых может быть или именем элемента или нулём. Если  $i$ -ый фрагмент  $j$ -го предложения есть имя элемента, то именно этот элемент будет соответствовать позиции решётки  $(i, j)$ . Если этот

фрагмент ноль, то данная позиция решётки исключена: при её выборе вместо выработки частицы производится новый выбор позиции. В следующих двух картограммах на данных позициях должны быть также нули.

Далее следует картограмма интенсивностей, состоящая из того же количества предложений с метками вида  $I\langle j \rangle$ . Конструкция  $\langle j \rangle$  образуется по тем же правилам, что и в случае с картограммой типов элементов  $E\langle j \rangle$ . Предложение состоит из  $n_i - m_i + 1$  неотрицательных действительных чисел, являющихся интенсивностями для соответствующих позиций решётки.

Далее следует картограмма вероятностей, состоящая из того же количества предложений с метками вида  $F\langle j \rangle$ . Конструкция  $\langle j \rangle$  образуется по тем же правилам, что и в случае с картограммой типов элементов  $E\langle j \rangle$ . Предложение состоит из  $n_i - m_i + 1$  неотрицательных действительных чисел, являющихся вероятностями выборки соответствующих позиций решётки. Важно отметить, что если позиция не исключена, то для неё недопустимо задавать нулевую вероятность.

### 10.2.10 Примеры данных для сложного источника

#### Пример 1

Цилиндр, содержащий изотропный по направлениям источник нейтронов со стандартным спектром деления.

В этом примере параметр в строке REPER не имеет никакого значения.

```
NPS          1
PROB         1
ANGLN       SPC1      NO
MDIS        S
EDIS        S
TYPE        N
RCZ         0, 0, -3    6    2
REPER       1
SNAM        SPC1
NOBJ        0
FINISH
```

#### Пример 2

Две параллельные прямоугольные пластинки, содержащие поверхностные источники нейтронов, которые излучают по нормальям к пластинкам, равновероятно в обе стороны. Спектр обеих пластинок одинаков по энергии и сосредоточен в нескольких точках.

```
NPS          2
PROB         1    1
ANGLN       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

### Пример 3

Совпадает по конфигурации со случаем 2, но каждая пластинка излучает нейтроны только в сторону другой пластинки.

В этом примере строка «REPER -1» во втором примитивном источнике вызывает формирование вектора  $\mathbf{v}_p$  как  $\mathbf{v}_n = (-1, 0, 0)$ . Того же эффекта можно было бы достичь, задав строку «REPER -1, 0, 0»

```

NPS 2
PROB 1 1
ANGLN SPCK
MDIS D 1
EDIS D 5
MMES 1
MPRO 1
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 SPCK
NOBJ 0
TYPE N
RPP 0.6,0.6 -1,1 0,3
REPER -1
SNAM SPCK
NOBJ 0
FINISH

```

### Пример 4

Совпадает по конфигурации с 1, добавлено распределение времени рождения, и стандартный спектр деления замен спектром деления калифорния.

```

NPS 1
PROB 1
ANGLN SCF NO
MDIS S
EDIS F CF52
TIME TC L
TMES 0 0.1 0.2 1
TPRO 0.45 0.3 0.25
TYPE N TC
RCZ 0,0,-3 6 2
REPER 1
SNAM SCF
NOBJ 0
FINISH

```

### Пример 5

Совпадает с 1, добавляется распределение по радиусу (дискретное) и высоте (линейное) цилиндра, задано в узлах сети.

```

NPS 1
PROB 1
ANGLN SPC1
MDIS S

```

```

EDIS      S
TYPE      N
RCZD      0,0,-3      -4      4
REPER     1
SNAM      SPC1
NOBJ      0
HGRI      0.1  0.5  0.9  1.
HPC       1,1.5 0.5,2 0.5,2 1,1
RGRI      0      0.5  0.8  1      1.2
RPC       1,1  1,1  1,1.5 0.5,2 0.5,3
FINISH

```

### Пример 6

Совпадает с 3, но использована функциональная связь между E и  $\mu$ , которые заданы дискретными.

```

NPS       2
PROB      1      1
ANGLLEN   SPCK  FUNC
MDIS      D      3
EDIS      D      3
MMES      0.5  0.866 1
EMES      0.3F6 0.4E6 0.5E6
MPRO      0.3  0.3  0.4
TYPE      N
RPP       0.6,0.6      -1,1  0,3
REPER     -1
SNAM      SPCK
NOBJ      0
FINISH

```

### Пример 7

Решётка состоит из 2\*3 позиций, одна из которых исключена. Используются 2 типа элементов: EK1 и EK2.

```

NPS              1
PROB             1
ANGLLEN         SPC1  NO
MDIS            L      5
EDIS            D      3
EMES            10.      50.      1E2
MMES            -1.      -0.8  -0.4      0.      0.5      1.
MPRO            1,1,2,1,5
EPRO            1      2      2
ELEM            EK1      PH
RCZD            0,0,-3      -3      -4
REPER           1
SNAM            SPC1
NOBJ            0
NORM            OFF
HGRI            0.      1.      1.5
HPC             1.      2.      1.      1.      1.      2.
NORM            ON
RGRI            0      1      2      3
RPC             1 1      2 1      1 1      4 1
ELEM            EK2  PH
RCZ             0,0,0      8      0.0001
REPER           1

```

```

SNAM      SPC1
NOBJ      0
LATT      S2AR
BOUN      0,1  -1,1
ROOT      0,0,-5  2,0,0  0,1,0
E-01      EK2  EK1
E000      EK2  EK2
E001      EK2  0
I-01      0.5  0.5
I000      1  2
I001      1  0
F-01      1  1
F000      1  1
F001      1  0
PRISOU
FINISH

```

### 10.3 НАКАПЛИВАЕМЫЙ ПОВЕРХНОСТНЫЙ ИСТОЧНИК

Для использования в качестве источника банка частиц, накопленных на предыдущем этапе необходимо определить, как эти частицы будут расщепляться. Следует отметить, что не генерируется ни одной точки в фазовом пространстве, не соответствующей частице из банка. Более того, математическое ожидание суммы весов частиц из определённой точки равно весу запомненной частицы. Поэтому можно говорить о расщеплении этих частиц.

При генерации программы MSU метод поверхностного источника включается двумя строками в файле параметров с расширением MEM:

```

+::GSURSO
+::GADDOB

```

Эти ключи нельзя совмещать с ключами WOOD – метод выровненных сечений.

При генерации должен быть указан размер блока записи для файлов, в которых копится источник, так называемые *файлы накопления*. Для этих целей в файле параметров должна присутствовать строка:

```

+$LBIUFP  $<целое число>$

```

<целое число> - число частиц в блоке записи для файлов накопления.

#### Пример

```

+$LBIUFP  $4096$

```

При расчёте не первого этапа следует использовать карту ISTR со значением 12 (см. Исходные данные для алгоритма моделирования), что включает режим, специально предназначенный для не начального этапа работы с накапливаемым источником. При этом дисперсия оценивается правильно – по сериям первого этапа.

*Использование накапливаемого поверхностного источника возможно только при расчёте на одном процессоре.*

#### 10.3.1 Общие характеристики поверхностного источника

Частицы выбираются из файла накопления последовательно в том же порядке, что они туда поступали. По каждой частице из файла генерируется серия частиц с одинаковыми фазовыми координатами. Математическое ожидание суммарного веса серии равно весу выбранной частицы.

Пусть из файла выбрана частица с пространственными координатами  $\mathbf{X}$ , энергией  $E$ , единичным вектором направления полёта  $\mathbf{U}$  и весом  $W$ . Модуль источника по вектору  $\mathbf{X}$

определяет коэффициент расщепления  $c_x = C_x(\mathbf{X})$ , а также единичный вектор основного для этой точки направления  $\mathbf{r}_x = \mathbf{R}_x(\mathbf{X})$ . Обе эти функции определены пользователем. Также пользователь определяет функции коэффициентов расщепления по энергии  $C_E(E)$  и по отклонению  $C_{MU}(\mu)$ . Отклонение определено как  $\mu = (\mathbf{U}, \mathbf{r}_x)$ . Так как оба вектора единичной длины, то  $\mu$  есть косинус угла между ними.

Общий коэффициент расщепления определяется как

$$C_p = c_x C_E(E) C_{MU}(\mu).$$

Все частицы в серии получают вес  $W / C_p$ .

Прочие параметры совпадают с параметрами исходной частицы из банка. Положим  $N$  - целая часть  $C_p$  и  $q = C_p - N$ . Тогда с вероятностью  $q$  порождается  $N+1$  частица и с вероятностью  $1-q$  -  $N$  частиц. Среднее число есть  $N+q = C_p$ .

Отметим, что вполне допустимо значение  $C_p < 1$ . В этом случае расщепление превращается в русскую рулетку.

В настоящее время реализованы функции скалярных аргументов, поэтому  $C_x$  должна представляться как произведение функций от 3-х координат, допустимы:

- декартовы координаты:  $C_x(\mathbf{X}) = C_x(x)C_y(y)C_z(z)$ ;
- цилиндрические координаты:  $C_x(\mathbf{X}) = C_r(r)C_\varphi(\varphi)C_z(z)$ .

Пусть аргумент обозначается буквой  $t$ , а значение  $s$ . В обоих случаях пользователь задаёт сетку точек  $t_0 < t_1 < t_2 < \dots < t_m$ .

Функции скалярного аргумента допускаются двух видов:

- кусочно-линейные: задаются  $m+1$  значения в узлах:  $s_0, s_1, \dots, s_m$ .

Для точки  $t \in [t_{i-1}, t_i]$ :  $s(t) = s_i - ((s_i - s_{i-1}) / (t_i - t_{i-1})) * (t_i - t)$ ;

- ступенчатые: задаются  $m$  значений на отрезках  $s_1, \dots, s_m$ .

Для точки  $t \in [t_{i-1}, t_i]$ :  $s(t) = s_i$ .

В случае  $t$  меньше  $t_0$  или  $t$  больше  $t_m$  по определению  $s$  равно 1, то есть, нет влияния на коэффициент расщепления.

Для точки, совпадающей по значению с одним из узлов за исключением первого и последнего, может быть выбран как левый отрезок, так и правый. Для случая кусочно-линейной функции это безразлично, для ступенчатой даёт разные результаты. Вероятность такого события может быть не нулевой.

Файл накопления образуется в той же директории, где ведётся расчёт. Он имеет имя: *<имя варианта>.fso*. В нем хранятся параметры зарегистрированных частиц по 18 слов на каждую. Файл записывается блоками с заданным числом частиц.

Там же образуется файл указателей с именем: *<имя варианта>.fsk*. В нем записаны 2 целых числа - номер последнего использованного блока  $n$  и указатель места в нем  $k$ .

Число  $n$  считается с 1. Число  $k$  есть  $(18 * \text{число частиц в блоке})$ . В случае полного заполнения  $n$  блоков в файле будут записываться числа  $n+1$  и 0. Далее в этом файле записаны три действительных числа и два целых числа, необходимые для правильной нормировки оценок функционалов на одну частицу *первого* этапа расчёта:

- главный нормировочный коэффициент  $C_{\text{main}}$  (см. ниже);
- суммарный вес частиц  $W_{\text{main}}$ , взятых в данном расчёте из банка для однородной задачи или из источника для неоднородной; на первом этапе расчёта именно на эту величину делятся накопленные суммы при нормировке оценок на один нейтрон;
- суммарный вес частиц  $W_{\text{sour}}$ , записанных в файл накопления;
- число частиц в пакете на первом этапе;
- число пакетов в серии на первом этапе.

Главный нормировочный коэффициент определяется по следующему правилу. Обозначим через  $C_0$  этот коэффициент, полученный на предыдущем этапе, чей файл накопления сейчас используется как источник. Для первого этапа будем считать  $C_0$  равным 1. Тогда по определению  $C_{\text{main}} = C_0 * W_{\text{sour}} / W_{\text{main}}$ . Легко понять, что при нормировке на



каждом этапе накопленные суммы должны не только делиться на  $W_{\text{main}}$ , но умножаться на  $C_0$ .

### 10.3.2 Задание поверхностей накопления и поглощения для накапливаемого источника

Для задания поверхностей накопления, а также поглощения (т.е. поверхностей накопления предыдущего этапа, которые сейчас являются границами системы) в геометрическом модуле непосредственно перед строкой FINISH должны быть заданы предложения:

```
LFIXSO      <список 1>
LBLACK      <список 2>
```

<список 1> определяет поверхность накопления источника. <список 2> определяет поверхность поглощения. На первом этапе используется лишь карта LFIXSO, на последнем - только LBLACK. На промежуточных этапах – обе карты.

Каждый <список> состоит из пар различных неотрицательных чисел, разделённых запятыми или пробелами. Внутри одного списка все пары должны быть различными. Порядок перечисления пар роли не играет.

#### Пример

```
LFIXSO      1, 2   1, 3
LBLACK      1, 0
```

Каждая пара определяет поверхность, состоящую из всех кусков границ зон, обладающих свойством, что с одной стороны лежит зона с одним значением объектного номера из пары, а по другую сторону - с другим. Ориентация поверхности соответствует направлению от первого элемента пары ко второму. Дадим точное определение, пользуясь замкнутостью зон.

Пусть  $\{Z_{1j}, \dots, Z_{N(j)j}\}$  множество зон с дополнительным атрибутом  $j$ . Тогда по паре  $j, k$  строится поверхность:

$$S(j, k) = (Z_{1j} \cup Z_{2j} \cup \dots \cup Z_{N(j)j}) \cap (Z_{1k} \cup Z_{2k} \cup \dots \cup Z_{N(k)k}).$$

Ориентация в каждой точке такой поверхности задаётся нормалью, идущей внутрь зоны с атрибутом  $k$ .

Список  $j_1, k_1, \dots, j_m, k_m$  определяет поверхность, определённую их объединениям:

$$S = S(j_1, k_1) \cup S(j_2, k_2) \cup \dots \cup S(j_m, k_m).$$

### 10.3.3 Ввод данных для поверхностного источника

Ввод данных осуществляется в разделе данных для модуля источника. В разделе ввода данных источника не должны одновременно присутствовать предложения с ключами поверхностного источника и какого-либо другого источника.

Порядок предложений в разделе источника произволен.

```
XOYOZO      <массив из 3 чисел>
```

<массив из 3 чисел> - три действительных числа, задаёт начало координат системы, в которой происходит расчёт функций  $C_x$  и  $R_x$ .

```
COORDS      <текстовый параметр>
```

*<текстовый параметр>* - может принимать значения XYZ или RFIZ:

- XYZ - декартова система координат, направления осей совпадают с направлениями осей в основной системе ввода геометрических данных; преобразование координат сводятся к сдвигу;

- RFIZ - цилиндрическая система координат  $(r, \varphi, z)$ , направление Oz совпадает с направлением в системе ввода геометрических данных, а равенства  $\varphi = 0, z = 0$  определяют прямую, параллельную оси OX.

REPER *<параметр определения  $R_x$ >*

*<параметр определения  $R_x$ >* - может принимать 4 варианта значений:

- 1 - цилиндрическая система координат,  $R_x((r, \varphi, z)) = (\cos(\varphi), \sin(\varphi), 0)$ ;

- минус 1 - цилиндрическая система координат,  $R_x((r, \varphi, z)) = (-\cos(\varphi), -\sin(\varphi), 0)$ ;

- *<массив из 3 чисел>* - функция  $R_x$  - постоянна и равна вектору  $v/|v|$ , где  $v$  имеет координаты равные 3 параметрам предложения;

-  $/P$  *<массив из 3 чисел>* - массив определяет точку, единичный вектор направления на которую есть  $R_x$ .

FILNAM *<имя файла>*

*<имя файла>* - строка из не более, чем 45 символов, означающих имя файла, содержащего банк источника, без расширения.

### Пример

FILNAM ..\V2\BAP01

Банк будет браться из файла ..\V2\BAP01.fso, а файл с номером последнего использованного блока и указателем занятости в нем будет ..\V2\BAP01.fsk.

Вышеперечисленные 4 предложения обязательно должны присутствовать.

MV1 *<массив действительных чисел>*

*<массив действительных чисел>* - сеть узлов и массив значений для первой координаты вектора  $X(x$  и  $r)$ . Тип функции определяется чётным или нечётным количеством чисел в нем:

- чётное число чисел  $(2m+2)$  - кусочно-линейная функция; первые  $m+1$  трактуются как узлы сети с индексами  $0, 1, \dots, m$ ; значения должны быть упорядочены по возрастанию; следующие  $m+1$  элементов являются значениями в этих узлах;

- нечётное число чисел  $(2m+1)$  - функция ступенчатая; первые  $m+1$  также есть сеть узлов, заданная в возрастающем порядке, а следующие  $m$  есть значения на отрезках.

Если предложение с ключом MV1 отсутствует, то коэффициент расщепления не зависит от первой координаты.

MV2 *<массив действительных чисел >*

То же самое, что описано для ключа MV1, но по отношению ко второй координате ( $y$  или  $\varphi$ ). Значения  $\varphi$  даются в градусах.

MV3 *<массив действительных чисел >*

То же самое, что описано для ключа MV1, но по отношению к третьей координате  $z$ .

MVEN *<массив действительных чисел >*

То же самое, что описано для ключа MV1, но по отношению к энергии.

*MVMU* <массив действительных чисел >

То же самое, что описано для ключа MV1, но по отношению к отклонению  $\mu$ .

*PRINT*

Предложение не имеет параметров и вызывает печать содержимого внутренних массивов модуля источников по окончанию ввода.

*BLAD* <целое число>

Если ключ отсутствует, то считается, что в предыдущем и текущем этапе геометрические зоны и их нумерация никак не изменились. Для пользователя это означает, что единственное, что он может изменить в данных для геометрического модуля (помимо предложений LFIXSO и LBLACK) – это замена номеров регистрационных зон и регистрационных объектов, а также, (что вообще-то нарушает математическую точность метода) номеров материалов в описаниях геометрических зон. Такое ограничение позволяет ускорить работу программы.

В некоторых расчётах удобно допускать небольшие коррекции в описании геометрии, для чего и служит данный ключ. В этом случае считается, что при выборке из накопленного файла источника частицы текущий номер геометрической зоны не определён, и, следовательно, геометрические зоны и их нумерация могут быть изменены. Это несколько снижает быстродействие, так что без нужды строку BLAD лучше не вставлять.

<целое число> - параметр, который определяет сдвиг стартовой точки, что сделано для ликвидации лишних поисков зон, вызванных ошибками округления:

– <целое число> меньше или равно 0 - точки берутся из накопленного файла без всяких изменений;

– <целое число> больше 0 и равно N - каждая стартовая точка сдвигается в направлении своего движения на  $0.00001 \cdot N$  см.

## 11 ИСХОДНЫЕ ДАННЫЕ МОДУЛЯ РЕГИСТРАЦИИ

Признаком начала ввода данных к модулю служит карта RGS. Признаком окончания ввода данных к модулю служит карта FINISH.

Ввод и обработка фрагмента с данными для модуля производится программными средствами стандартного ввода.

В общем случае этот фрагмент состоит из следующих разделов.

*RGS*

<общие параметры>

<параметры регистрации типа 1>

...

<параметры регистрации типа N>

*FINISH*

Все разделы могут быть пропущены. В этом случае модуль регистрации не будет регистрировать никакие события, что ускоряет работу программы.

Порядок ввода разделов внутри фрагмента произволен.

Для всех карт, начинающихся с M, Z или O и имеющих параметр list, значения в этом параметре 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.

## 11.1 ЗАГОЛОВОК ФРАГМЕНТА ИСХОДНЫХ ДАННЫХ МОДУЛЯ РЕГИСТРАЦИИ

Фрагмент исходных данных для модуля регистрации должен начинаться с обязательной карты RGS.

*RGS [value1 value2]*

*RGS* - имя карты — заголовка фрагмента;

*value1* - определяет наличие/отсутствие печати карт файла исходных данных в выходном файле (1 – печатать, 0 – не печатать). Значение по умолчанию — 0.

*value2* - определяет наличие/отсутствие отладочной печати в выходном файле, а при наличии печати также и её количество (больше 0 – печатать, 0 – не печатать). Значение по умолчанию — 0.

## 11.2 РАЗДЕЛ ОБЩИХ ПАРАМЕТРОВ

В этом разделе задаются параметры для регистрации некоторых стандартных функционалов, а также некоторые общие для всех видов регистрации параметры.

Общий вид задания раздела следующий:

*[KEFF]*  
*[LIFE]*  
*[BAL]*  
*[MOM4]*  
*[BUCL B<sub>x</sub>, B<sub>y</sub>, B<sub>z</sub>]*  
*[BCRIT value]*  
*[Uxxxxx values]*  
*[EABS]*  
*[MPHEN list]*  
*[ZPHEN list]*  
*[OPHEN list]*  
*[NUCOFF]*  
*[SPCMTX]*  
*[PERC value1]*  
*[FIXED]*  
*[NRET value1 value2]*

*KEFF* – при наличии этой карты будет рассчитываться эффективный коэффициент размножения нейтронов в рассчитываемой системе.

*LIFE* – при наличии этой карты будет рассчитываться среднее время жизни нейтронов в моделируемой системе.

*BAL* – при наличии этой карты будут вычисляться функционалы баланса нейтронов в системе, в том числе утечка из системы.

*MOM4* – при наличии этой карты помимо первого и второго центральных моментов для всех рассчитываемых функционалов будут определяться также и третий и четвёртый центральные моменты.

*BUCL* – задаёт компоненты вектора геометрического параметра ***B*** (баклинга), характеризующего утечку нейтронов, и используется только при решении асимптотической задачи. Если решается асимптотическая задача, то в данных к геометрическому модулю требуется задать граничное условие трансляции вдоль направлений с ненулевыми значениями ***B***.

*B<sub>x</sub>, B<sub>y</sub>, B<sub>z</sub>* – задают вектор баклинга, умалчиваемое значение 0,0,0, т.е. баклинг не используется.

*BCRIT* – вычисляет значения эффективного коэффициента размножения (необходимо наличие карты *KEFF*) для *value* значений параметра  $B_z$  (необходимо наличие карты *BUCL*):  $B_z(i) = B_z * i / value$ ,  $i = 1, value$ . На этапе финальной обработки по полученным значениям эффективного коэффициента размножения вычисляется критическое значение баклинга с помощью линейной интерполяции.

*Uxxxxx* – при наличии карты начинающейся с буквы U при работе программы вызываются подпрограммы подмодуля USER регистрационного модуля. На этапе ввода исходных данных обработка параметров карт *values*, начинающихся с литеры U, передаётся подмодулю USER.

*EABS* – при наличии этой карты вычисляется средняя энергия поглощения нейтрона.

*MPHEN, ZPHEN, OPHEN* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов), в которых вычисляется поглощённая энергия вследствие взаимодействия фотонов с веществом.

*list* – список номеров соответствующих регистрационных областей (соответственно материалов, зон и объектов).

*NUCOFF* – при наличии этой карты регистрация скоростей реакций для нуклидов в отдельности не производится.

*SPCMTX* – при наличии этой карты регистрационный модуль производит вычисление элементов матриц в специальном режиме. А именно, предполагается, что будет проведено число расчётов равное числу объектов, запрошенному в карте *OMTX* с различным распределением нейтронов нулевого поколения и рождением вторичных частиц только в *i*-м объекте (для *i*-го расчёта) см. карту *WOBJ*.

*PERC* – определяет способ печати статистических погрешностей.

*value1* – при значении равном 0 погрешности печатаются в абсолютных величинах, при 1 – в процентах (значение по умолчанию – 1).

*FIXED* – при наличии этой карты создаётся файл, содержащий результаты работы модуля регистрации в фиксированном формате.

*NRET* – эта карта определяет порядок перебора характеристик функционала при печати в файл стандартной выдачи. (*NRET* – Nuclide Reaction Energy Tally).

*value1* – четырехзначное число, определяющее порядок перебора характеристик. Каждая цифра этого числа определяет очерёдность перебора соответствующей характеристики. При этом цифра, определяющая тысячи, соответствует нуклиду, сотни — реакции, десятки — энергии, единицы — области регистрации (материалу, зоне или объекту). Например, число 3214 определяет стандартный для собранных на базе пакета *MCU4* 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.

Примечание - Карты *PERC, FIXED, NRET* используются только на этапе финальной обработки накопленной статистики. Значения этих карт могут быть изменены перед выполнением финальной обработки.

### 11.3 РАЗДЕЛ ПАРАМЕТРОВ РЕГИСТРАЦИИ ТИПА N

Модуль регистрации позволяет осуществлять накопление статистики с различными условиями. Пользователь определяет эти условия для каждого раздела параметров регистрации независимо. Каждый раздел предназначен для регистрации одного типа частиц. Можно определять любое количество разделов для частиц одного типа.

Общий вид задания раздела следующий:

```
PTYPE value1  
[TTYPE value1]  
[MFLU list]
```

```

[ZFLU list]
[OFLU list]
[LEAK]
[MIGL]
[MRCT list]
[ZRCT list]
[ORCT list]
[MDOS list]
[ZDOS list]
[ODOS list]
[MPOW value1]
[ZPOW value1]
[OPOW value1]
[MMUL value1, value2, value3, ..., valuemaxmat]
[ZMUL value1, value2, value3, ..., valuemaxmat]
[OMUL value1, value2, value3, ..., valuemaxmat]
[MMAX list]
[ZMAX list]
[OMAX list]
[MADF list]
[ZADF list]
[OADF list]
[MCUR list]
[ZCUR list]
[OCUR list]
[MSMT list]
[ZSMT list]
[OSMT list]
[RCT list]
[FRM list]
[DOS list]
[ENERGY list]
[SPECTR value]
[Uxxxxx values]
END

```

В этом разделе карты, начинающиеся с буквы М относятся к материалам, Z — зонам, а О — к объектам.

*P*TYPE – обязательная первая карта раздела.

*value1* – 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).

*T*TYPE – эта карта определяет способ оценки функционалов.

*value1* – число, определяющее способ оценки функционалов (0 - по точкам столкновений, 1 – по длине пробега, 2 - по точкам поглощений). Значение по умолчанию — 0.

*MFLU*, *ZFLU*, *OFLU* – эти карты определяют 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 этой карты в регистрационных областях, определённых с помощью карт *MFLU*, *ZFLU* или *OFLU* (соответственно материалов, зон и объектов), вместо потока будет суммироваться вес частиц, что позволяет рассчитывать утечку.

*MIGL* – при наличии этой карты вычисляется средний квадрат длина смещения нейтрона до его поглощения или смены энергетической группы в многогрупповом разбиении согласно заданным в данном разделе энергетическим сеткам и интегрально по всей энергии в пространстве, а также вдоль осей *OX*, *OY*, *OZ*.

*MRCT*, *ZRCT*, *ORCT* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов) в которых будут рассчитываться скорости реакций, заданные с помощью карт *RCT* и *FRM*.

*list* – список номеров соответствующих регистрационных областей (соответственно материалов, зон и объектов).

*MDOS*, *ZDOS*, *ODOS* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов) в которых будут рассчитываться скорости дозиметрических реакций, заданные с помощью карты *DOS*.

*list* – список номеров соответствующих регистрационных областей (соответственно материалов, зон и объектов).

*MPOW*, *ZPOW*, *OPOW* – при наличие этих карт на этапе финальной обработки значения функционалов во всех регистрационных областях (соответственно материалах, зонах и объектах) умножаются на *value1*.

*MMUL*, *ZMUL*, *OMUL* – при наличие этих карт на этапе финальной обработки значение функционала в *i*-й регистрационной области (соответственно материале, зоне и объекте) умножается на *value<sub>i</sub>*.

*MMAX*, *ZMAX*, *OMAX* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов), среди которых на этапе финальной обработки будет производиться поиск регистрационной области, в которой значение скорости реакции деления максимально при заданном энергетическом разбиении. Данные карты требуют указания скорости реакции деления при задании карты *RCT*.

*MADF*, *ZADF*, *OADF* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов), в которых будут вычисляться (и целиком по системе) следующие функционалы: поток (в финальной выдаче – *Flux*); скорость реакции рождения нейтронов, появившихся в системе за счёт как внешних источников, так и рассеявшихся из других групп (*Source*); скорость реакции рождения нейтронов, появившихся только в результате деления ядер (*Birth*); макроскопические сечения деления, поглощения и генерации нейтронов; макроскопическое сечение увода нейтронов из группы в результате их поглощения в группе или рассеяния (соответственно – *Fiss. C/S*, *Absorpt. C/S*, *Nu\*Fiss. C/S*, *Removing C/S*); средний квадрат длины смещения  $R^2$  нейтрона до его поглощения, вылета из регистрационной области или смены энергетической группы в пространстве и вдоль осей *OX*, *OY*, *OZ*; средний квадрат длины миграции  $M^2 = R^2 / (6 * Flux * Removing C/S)$ ,  $M^2(x) = R^2(x) / (2 * Source)$  и т.д.; коэффициент диффузии  $D = M^2 * Removing C/S$ ,  $D(x) = M^2(x) * Removing C/S$  и т.д.; асимптотической решётки методом утечки Бенуа ( $D(B)$ ). Величины  $R^2$ ,  $M^2$  и  $D$  имеют смысл при нулевом векторе баклинга и при отсутствии утечки в системе, а величина  $D(B)$  – при не нулевом векторе баклинга.

*MCUR*, *ZCUR*, *OCUR* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов), в которых будут вычисляться односторонние и двусторонние токи.

*MSMT*, *ZSMT*, *OSMT* – эти карты определяют номера регистрационных областей (соответственно материалов, зон и объектов), для которых вычисляются полное сечение с транспортной поправкой и нейтронные групповые макроконстанты перехода из одной энергетической группы в другую, взвешенные по потоку, а также их сечения.

*RCT* – эта карта определяет номера реакций (согласно нумерации физического модуля) которые рассчитываются в областях, определённых картами *MRCT*, *ZRCT*, *ORCT*.

*list* – список номеров реакций. Для указанного списка реакций будет вычисляться понуклидный баланс для всей системы.

*FRM* – эта карта определяет реакцию, определяемую формулой, задаваемой пользователем. Эта реакция будет рассчитываться в областях, определённых картами *MRCT*,

ZRCT, ORCT. Таких карт может быть произвольное количество. Формулы нумеруются согласно порядку их появления в разделе. Формулы записываются по определённым правилам. Возможные арифметические операции: сложение (кодируется знаком (+), вычитание (-), умножение (\*) и деление (/). В качестве переменных указываются порядковые номера реакций в карте RCT. Порядок вычисления соответствующей скорости реакции слева направо. Все арифметические операции равнозначны (т.е. выражение  $1+2*3$  равно 9, а не 7). Математические операции вычисляются непосредственно в момент регистрационного события (например, столкновения) и с точки зрения регистрационного модуля являются такими же реакциями, как и реакции, определённые в карте RCT.

*list* – формула реакции.

*DOS* – эта карта определяет реакции из библиотеки DOSIM, которые будут рассчитываться в областях, определённых картами MDOS, ZDOS, ODOS. Каждая реакция задаётся двумя константами символьного типа, разделёнными пробелом. Первая константа определяет название изотопа (содержится в первых 4-х позициях имени файла в библиотеке, при этом следует игнорировать символ Q), вторая — источник и номер реакции (содержатся во вторых 4-х позициях имени файла в библиотеке). Например, задание пары C12 R16 соответствует файлу C12QR16.ACT.

*list* – список пар символьных констант, разделённых пробелом.

*ENERGY* – эти карты определяют значения нижних границ энергетических регистрационных групп в электрон-вольтах для одной энергетической сетки. Верхняя граница последней энергетической группы равна бесконечности. Каждая карта задаёт одну независимую энергетическую сетку. Регистрация ведётся отдельно для каждой сетки. Внутри одного раздела может быть использовано любое количество таких карт. При отсутствии хотя бы одной карты регистрация для данного раздела не ведётся. Отсутствие значений в карте приводит к ошибке.

*list* – нижние границы регистрационных групп в электрон-вольтах. Значение 0 должно быть задано явно.

*SPECTR* – при наличии этой карты функционалы потока и скоростей реакций на этапе финальной выдачи делятся на ширину энергетической группы (*value* = 1) или на разницу логарифмов от границ энергетической группы (*value* = 2). При *value* = 0 (значение по умолчанию) финальная выдача происходит стандартно.

*Uxxxxx* – при наличии карты начинающейся с буквы U при работе программы вызываются подпрограммы подмодуля USER регистрационного модуля. На этапе ввода исходных данных обработка параметров карт *values*, начинающихся с литеры U, передаётся подмодулю USER.

*END* – окончание раздела. Эта карта является обязательной.

Для общих регистрационных областей в картах *MFLU* и *MRCT* (*ZFLU* и *ZRCT*, *OFLU* и *ORCT* – аналогично) будут вычисляться макроскопические сечения по всей области для тех типов реакций, которые перечислены в карте RCT.

## 11.4 ОКОНЧАНИЕ ВВОДА ДАННЫХ ДЛЯ МОДУЛЯ РЕГИСТРАЦИИ

Фрагмент исходных данных для модуля регистрации должен заканчиваться обязательной картой FINISH.

*FINISH* [*commentary*]

*FINISH* – признак окончания ввода данных к модулю регистрации.  
*commentary* - текст произвольного содержания.



## 12 ИСХОДНЫЕ ДАННЫЕ МОДУЛЯ РЕГИСТРАЦИИ ДЛЯ ВЫГОРАНИЯ

Модуль регистрации для выгорания BRG предназначен для подготовки и передачи в модуль выгорания потоков и сечений, а также объёмов материалов. *Модуль работает независимо от модуля регистрации.*

Данные для модуля вводятся после данных для модуля регистрации, либо после данных для модуля теплофизического анализа, если он присутствует в задании.

В случае отсутствия раздела данных для модуля регистрации для выгорания модуль не используется.

Ввод и обработка фрагмента с данными для модуля производится программными средствами стандартного ввода.

Общий вид задания раздела следующий:

```
BRG [value1 value2]
[BURALL]
[BMAX maxbur]
[VOL V1, V2, ..., Vn, ..., Vnmat]
[BUCL value1 value2 value3]
FINISH commentary
```

*BRG* – заголовок исходных данных к модулю, первая обязательная карта.

*value1* - определяет наличие/отсутствие печати карт файла исходных данных в выходном файле (1 – печатать, 0 – не печатать). Значение по умолчанию — 0.

*value2* - определяет наличие/отсутствие отладочной печати в выходном файле, а при наличии печати также и её количество (больше 0 – печатать, 0 – не печатать). Значение по умолчанию — 0.

Если после заголовка нет ни одной карты, то в этом случае модуль не работает.

*BURALL* – при наличии этой карты модуль готовит сечения и потоки для всех материалов. Это рекомендуемый режим работы, позволяющий наиболее полно использовать возможности модуля выгорания.

*BMAX* – карта позволяет выполнить подготовку сечений только для первых *maxbur* материалов, описанных в разделе исходных данных для физического модуля. Это позволяет увеличить скорость расчёта состояния. Карта не работает при наличии в исходных данных карты *BURALL*.

При задании *maxbur* меньше, чем полное количество материалов, концентрации изотопов во всех выгорающих материалах будут рассчитаны правильно, но *k<sub>inf</sub>* *будет иметь неверное значение.*

По умолчанию *maxbur* = 0. При *maxbur* = 0 и отсутствии карты *BURALL* модуль не работает, и расчёт выгорания не проводится.

*VOL* – задаёт объёмы материалов. Может отсутствовать, только если модуль не работает.

*V1, V2, ..., Vn, ..., Vnmat* – значения объёмов (в единицах см<sup>3</sup>), занятых как выгорающими, так и невыгорающими материалами, *nmat* – полное количество материалов в исходных данных к физическому модулю. Объёмы перечисляются последовательно, начиная с первого.

Рекомендуется задавать реальные объёмы для всех материалов, что в сочетании с использованием карты *BURALL* позволяет наиболее полно использовать возможности модуля выгорания.

При  $n > maxbur$  объёмы  $Vn$  можно задавать любыми (например, равными 1), что не повлияет на результаты работы модуля выгорания. При  $n \leq maxbur$  объёмы можно задавать любыми (например, равными 1), только если соответствующий материал не выгорает, и энергия в нем не выделяется (тип N, см. раздел модуля выгорания). В этом случае

концентрации изотопов в материалах и интегральные параметры этих материалов будут рассчитаны модулем выгорания правильно, *однако  $k_{inf}$ , потоки нейтронов в материалах и полный поток нейтронов во всей системе будут иметь неверные значения.*

*BUCL* – задаёт компоненты вектора геометрического параметра **B** (баклинга), характеризующего утечку нейтронов, и *используется только при решении асимптотической задачи.* Если решается асимптотическая задача, то в данных к геометрическому модулю требуется задать граничное условие трансляции вдоль направлений с ненулевыми значениями **B**.

*value1, value2, value3* – задают вектор баклинга, умалчиваемое значение 0,0,0, т.е. баклинг не используется.

*FINISH* – признак окончания ввода данных к модулю регистрации.

*commentary* - текст произвольного содержания.

Примечания

1 Этот раздел данных может отсутствовать полностью. В таком случае расчёт выгорания не осуществляется.

2 Если при генерации программы в неё не был включён модуль BRG, то данные этого раздела игнорируются.

## 13 ИСХОДНЫЕ ДАННЫЕ ДЛЯ АЛГОРИТМА МОДЕЛИРОВАНИЯ

Фрагмент данных для алгоритма моделирования включает предложения, все из которых могут отсутствовать за исключением обязательного последнего предложения *FINISH*.

Для ввода данных модуль использует программные средства стандартного ввода.

*ISTR istr*

Число *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 используется рассчитанная каким-то другим методом плотность нейтронов деления.

При *istr = 12* режим специально для не начального этапа работы с накапливаемым источником. При этом дисперсия оценивается правильно – по сериям первого этапа.

*NTOT ntot*

Число частиц в поколении. По умолчанию *ntot = 200*. Систематическая ошибка расчёта функционалов убывает обратно пропорционально *ntot*.

*NBAT nbatch*

Каждые последовательные *nbatch* поколений частиц, называемых сериями, считаются независимыми. По умолчанию *nbatch = 3*. Точность расчёта статистических ошибок рассчитываемых функционалов возрастает с увеличением *nbatch*.

*NSKI nski*

Число отбрасываемых серий, т. е. серий, не учитываемых при накоплении статистики. По умолчанию *nski = 0*.

*WTOB w1, w2, ... , wkobj*

Данная карта предназначена для использования при корректировке числа нейтронов деления на один акт деления (однако распространяется на все типы моделируемых частиц). Карта содержит массив  $w_i$ , где  $i$  соответствует номеру регистрационного объекта. Каждое  $w_i$  есть неотрицательное число, на которое умножается вес любой вторичной частицы, рождённой в регистрационном объекте  $i$ , перед помещением её в банк. При  $w_i = 0$  вторичные частицы, рождённые в регистрационном объекте  $i$ , в банк не помещаются и не моделируются. При отсутствии карты все  $w_i$  равны 1.

*MIXG nmixg*

Карта управляет нормализацией поколений в многопроцессорном режиме. При  $nmixg = 0$  нормализация выполняется независимо на каждом процессоре, что соответствует стандартной идеологии распараллеливания расчетов MSU. При  $nmixg$  отличном от нуля после, того как все процессоры промоделировали одно поколение нейтронов, выполняется сбор данных по всем процессорам, общая нормализация и раздача нормализованных данных процессорам. Таким образом, в этом режиме многопроцессорный расчет полностью аналогичен по идеологии обычному однопроцессорному расчету. По умолчанию  $nmixg = 0$ .

*FINISH*

Признак конца фрагмента.

## 14 ИСХОДНЫЕ ДАННЫЕ ДЛЯ УПРАВЛЕНИЯ ШАГОМ РАСЧЁТА

Данные этого фрагмента позволяют пользователю управлять режимом расчёта и задавать управляющие параметры для режима финальной обработки результатов моделирования.

Для ввода данных модуль использует программные средства стандартного ввода.

Фрагмент задаётся в общем наборе исходных данных задачи и затем, после выполнения шага INPUT, запоминается в файле NAME.DAT, где NAME - имя файла с исходными данными. Перед выполнением шагов CALCULATION и OUTPUT файл NAME.DAT может редактироваться пользователем.

Фрагмент для шага расчёта начинается с карты NAMV, состоит из нескольких необязательных строк и обязательной строки FINISH - признака окончания ввода данных к разделу.

Строки данных относятся к трём типам – общее регулирование режимом счёта, управление финальной выдачи, управление неаналоговым моделированием.

*NAMV namvar*

*namvar* - название варианта; текстовая переменная длиной не более 12 символов; редактированию не подлежит.

*MAXS maxser*

*maxser* – номер серии, после которой счёт прекращается; может редактироваться в процессе счёта. При наличии ненулевого количества отбрасываемых серий (карта NSKI) параметр *maxser* определяет число регистрируемых серий, т.е. счёт окончится после моделирования числа серий, равного сумме параметров карт NSKI и MAXS.

*DTZM nser*

*nser* - количество серий, через которое накопленная информация записывается в файл задачи; может редактироваться в процессе счёта.

*NPRI nhys*

*nhys* - количество историй, через которое на экран будет поступать информационная строка; может редактироваться в процессе счёта.

*BETA*

При наличии этой карты вычисляется эффективная доля запаздывающих нейтронов; редактированию не подлежит.

*NRAN n*

Параметр, управляющий началом выборки случайных чисел; используется для продолжения расчётов при фатальных авоствах; для продолжения счёта нужно вновь запустить шаг, задав произвольное целое число *n*; может редактироваться в процессе счёта.

*ECUT e1*

Указывается граница энергии нейтрона в электронвольтах, ниже которой траектории прерываются. Область изменения параметра определяется диапазоном работы нейтронных подмодулей, заданных в разделе исходных данных для физического модуля. Значение по умолчанию 0 эВ. Может редактироваться в процессе счёта.

*ECUP e1*

Указывается граница энергии фотона в электронвольтах, ниже которой траектории прерываются. Область изменения параметра определяется диапазоном работы фотонного подмодуля, заданным в разделе исходных данных для физического модуля. Значение по умолчанию 10000 эВ. Может редактироваться в процессе счёта.

## **14.1 УПРАВЛЕНИЕ НЕАНАЛОГОВЫМ МОДЕЛИРОВАНИЕМ**

Допустимо использование нескольких альтернативных методов неаналогового моделирования. Какой из них будет использован определяется ключами генерации. Для нейтронов и фотонов используется один и тот же метод, но параметры для разных типов частиц могут быть разными.

Строки данных для неаналогового моделирования зависят от применяемого метода, но в любом случае может быть использована строка задания типа частиц

*SETT nr*

Управляющий параметр *nr* может принимать значения N, PH. (по умолчанию *nr* = N).

Значение N означает, что все дальнейшие строки данных для неаналогового моделирования относятся к нейтронам, значение PH – к фотонам. Таким образом, если решается чисто нейтронная задача, то строка задания типа частицы не нужна.

### **14.1.1 Расщепление/рулетка**

Метод расщепления/рулетки подробно описан научной литературе. Возможности MCU для данного метода аналогичны возможностям других кодов метода Монте-Карло.

Этот метод имеет несколько модификаций. Далее приведено описание его алгоритма, которое использовано в программе MCU.

Введём следующие обозначения. Производятся два независимых разбиения фазового пространства на непересекающиеся области:

- разбиение геометрического пространства на регистрационные объекты;

– разбиение энергетической оси на непересекающиеся интервалы

$$R_+ = [0, E_1) \cup [E_1, E_2] \cup \dots \cup [E_n, \infty).$$

Это разбиение не зависит от разбиения на регистрационные энергетические группы и задаётся вне связи с ним.

Каждому регистрационному объекту приписывается характерный вес  $W > 0$  и обратная к нему величина - ценность  $V = 1/W$ .

При переходе из одного регистрационного объекта  $O_i$  в другой объект  $O_j$  происходит расщепление /рулетка по следующим правилам:

Пусть объект  $O_i$  имеет характерный вес  $W_i$ , а объект  $O_j - W_j$ . Частица имеет вес  $w$ .

Обозначим^

–  $N = [W_i/W_j]$  - целая часть;

–  $\gamma = (W_i/W_j - N)$  – дробная часть.

Образуется  $M$  частиц с теми же фазовыми координатами, что и у исходной частицы, и весом  $w' = w/(W_i/W_j)$ , причём с вероятностью  $\gamma$   $M = N+1$ , а с вероятностью  $1-\gamma$   $M = N$ .

Данная процедура осуществляется геометрическим модулем. Если  $M > 0$ , то геометрический модуль продолжает моделирование полёта одной частицы. При  $M > 1$ , остальные  $M-1$  частиц помещаются в очередь расщепления.

При  $M = 0$ , траектория частицы прерывается со специальным признаком обрыва по рулетке.

Аналогичный алгоритм работает при пересечении границы отрезка энергетической оси, что осуществляется после рассеяния (включая  $n \rightarrow n+n$  реакцию), если  $W_i$  – характерный вес для энергии до рассеяния, а  $W_j$  – характерный вес энергии после рассеяния.

Полезно помнить очевидные выражения описанных величин в терминах ценностей:

$$N = [V_j/V_i], \gamma = V_j/V_i - N, w' = w V_i/V_j.$$

Режим расщепления/рулетки включается в файл параметров с расширением MEM по ключу:

*+SPLRUL*

*Режим не совмещается с режимом весового окна и с режимом поверхностного источника.*

Границы отрезков по энергии вводятся предложением

*WWEN* <массив границ отрезков>

<массив границ отрезков> - массив должен быть упорядочен по возрастанию, энергия вводится в электронвольтах. Любая энергия должна попадать в какой-либо отрезок, поэтому последнее число должно быть очень большим. Число отрезков совпадает с числом введённых чисел, поскольку левая граница первого интервала есть 0.

Ценности для отрезков на оси энергии задаются предложением

*INPE* <массив ценностей>

<массив ценностей> - массив положительных действительных чисел. Число этих чисел должно быть таким же, как число отрезков.

Ценности регистрационных объектов задаются предложением

*INPO* <массив ценностей>

<массив ценностей> - массив положительных действительных чисел. Длина этого массива может не совпадать с реальным числом регистрационных объектов. Если длина массива меньше этого числа, недостающие позиции заполняются единицами.

### 14.1.2 Весовое окно по энергии и регистрационным объектам

Метод весового окна имеет несколько модификаций.

Весовое окно работает в двух случаях:

- пересечение геометрической границы ячейки фазового пространства, что отслеживается геометрическим модулем;
- при изменении скоростных координат – после рассеяния или выборки частицы из банка.

Если весовое окно работает в обоих описанных случаях, будем говорить о полном весовом окне.

Можно не пользоваться первой возможностью, а проверять геометрическое положение частицы только после рассеяния или выборки из банка. Такой способ будем называть весовым окном по рассеянию.

Работу окна по рассеянию можно пояснить следующим образом.

Моделирование траектории частицы после её рождения представляет последовательность меняющихся этапов: свободный полет (СП), взаимодействие со средой (ВС). После каждого этапа частица может исчезнуть (утечки в этапе СП и поглощение в этапе ВС). Эту схему можно пояснить рисунком 14.1.2.1.



Рисунок 14.1.2.1 – Схема работы весового окна по рассеяниям

Алгоритм весового окна включается в точке, обозначенной на рисунке буквой W; то есть перед свободным полётом и после взаимодействия со средой, или выработки частицы источником или выборкой её из банка.

Каждое весовое окно характеризуется тремя параметрами:

$$0 < w_i \leq w_m < w_s.$$

В программе MCU имеются две константы  $c_m$  и  $c_s$  такие, что для любого весового окна  $w_m = c_m w_i$ ,  $w_s = c_s w_i$ .

Очевидно, что  $1 \leq c_m < c_s$ .

Кроме того, требуется выполнение неравенства:

$$2 c_m \leq c_s.$$

По умолчанию упомянутые константы имеют значения

$$c_m = 2.5, \quad c_s = 5.$$

Эти значения могут быть изменены предложением, которое вводится как данные алгоритма моделирования:

WISU <значение  $c_m$ > <значение  $c_s$ >

Если вес частицы  $w$  лежит в весовом окне  $w_i \leq w \leq w_s$ , то ни расщеплений, ни рулетки не производится.

Если  $w > w_s$ , то производится расщепление на целое число  $N = [w/w_m]$ . Очевидно,  $N \geq 2$ . Вес образованных  $N$  частиц будет  $w' = w/N$ , причём  $w_m \leq w' < w_s$ .

Если  $w < w_s$ , то с вероятностью  $w/w_m$  вес частицы становится равным  $w_m$ , в противном случае частица уничтожается.

*Работу весового окна запрещено совмещать с режимом с расщеплением/рулеткой.*

В описываемом режиме работает полное весовое окно. Режим включается в файл параметров с расширением MEM по ключу:

*+WIOBRG*

*Работу весового окна в этом режиме запрещено совмещать с режимом с поверхностным источником.*

Ячейки фазового пространства устроены так же как в режиме расщепления/рулетки – геометрическое пространство разбито на регистрационные объекты, а энергетическая ось – на отрезки.

Параметры весовых окон вводятся предложениями:

*WWEN* <массив границ отрезков по энергии>

<массив границ отрезков по энергии> - массив должен быть упорядочен по возрастанию, энергия вводится в электронвольтах. Любая энергия должна попадать в какой-либо отрезок, поэтому последнее число должно быть очень большим. Число отрезков совпадает с числом введённых чисел, поскольку левая граница первого интервала есть 0.

Ценности веса для отрезков на оси энергии задаются предложением

*INPE* <массив ценностей>

<массив ценностей> - массив положительных действительных чисел. Число этих чисел должно быть таким же, как число отрезков.

Это предложение должно следовать за описанием границ отрезков.

Ценности регистрационных объектов задаются предложением

*INPO* <массив ценностей>

<массив ценностей> - массив положительных действительных чисел. Длина этого массива может не совпадать с реальным числом регистрационных объектов. Если длина массива меньше этого числа, недостающие позиции заполняются единицами.

В этом режиме характерные веса  $W_i = 1/V_i$  трактуются как множители нижней границы окна, т.е. если частица лежит в объекте номер  $j$ , а её энергия в отрезке номер  $k$ , то нижняя граница окна будет

$$w_i = 1(V_{oj}V_{ek}).$$

(индексы  $o$  и  $e$  введены для обозначения характерных весов для объектов и энергий, соответственно).

### **14.1.3 Весовое окно по энергии дополнительной геометрии и углу**

Все общие положения о весовых окнах даны в предыдущем пункте. В описываемом режиме используется весовое окно по рассеянием.

Параметры весового окна также зависят от энергии и положения точки в пространстве, но также могут зависеть от направления полёта частицы. Алгоритм весового

окна работает только после события рассеяния, что даёт возможность использовать независимую геометрию.

Описываемый режим совместим с режимом накопления поверхностного источника.

Режим включается в файл параметров с расширением MEM по ключу:

*+WIUSER*

Дальнейшие данные для этого режима вводятся как данные алгоритма моделирования, так что в принципе могут быть изменены в течение одного расчёта. Если эти предложения опущены, то весовые окна не работают. Имеются следующие предложения для ввода параметров:

*WWEN* <массив границ отрезков по энергии>

<массив границ отрезков> - массив действительных, положительных чисел, задаёт разбиение оси энергии на NE отрезков (NE- размерность массива), также как для метода расщепления/рулетки.

*XYZ0* <координаты центра>

Если <координаты центра> состоят из трёх чисел, они задают X0 точку в пространстве с помощью координат в основной геометрии. Эта точка будет центром сферической системы координат весовых окон. Использоваться будут лишь расстояния до точки X0 т.е. геометрическое пространство разбивается на сферические слои.

Если <координаты центра> состоят из двух чисел, они задают ось OZ цилиндрической системы координат  $(r, \theta, z)$  в пространстве основной геометрии. Для определения параметров весовых окон использоваться будут лишь координата r, т.е. геометрическое пространство разбивается на цилиндрические слои.

*RADS* <массив радиусов слоёв >

<массив радиусов слоёв> - массив действительных, положительных чисел:  $r_1 < r_2 < \dots < r_{NR}$ . (NR- размерность массива).

1-ый слой определён неравенством  $0 \leq r \leq r_1$

j-ый слой определён неравенством  $r_{j-1} < r \leq r_j$ .

*INPM* < массив из NR·NE чисел >

<массив из NR·NE чисел> - массив понимается как матрица  $VE_{ij}$  вводимая по строкам и определяющая ценности по радиусам и энергиям.

*SANG* < массив косинусов >

<массив косинусов> - вводится массив из NA-1 чисел  $-1 < s_1 < s_2 < \dots < s_{NA-1} < 1$  Определяет разбиение пространства по углам. (NA-1 - размерность массива).

*INRA* < массив из NR·NA чисел >

<массив из NR·NA чисел> - массив понимается как матрица VA размерности NR\*NA, вводимая по строкам.

Последние 2 предложения могут отсутствовать. При отсутствии этих карт зависимость от направления полёта отсутствует.

При угловом разбиении и сферической системе координат весовые окна определены следующим алгоритмом.



Пусть точка фазового пространства имеет координаты  $(X, E, U)$   $X$  – геометрическое положение,  $E$  – энергия,  $U$  – направление полёта.

Числа  $j, k, n$  определены так, что выполнены соотношения:

$$\begin{aligned} |\bar{X} - \bar{X}_0| &\in (r_{j-1}, r_j] \quad (r_0 = 0) \\ E &\in (E_{k-1}, E_k] \quad (E_0 = 0) \\ \cos(U, \bar{X}_0 - \bar{X}) &= \frac{(U, \bar{X}_0 - \bar{X})}{|\bar{X}_0 - \bar{X}|} \in [s_{n-1}, s_n] \quad (s_0 = -1, s_{N_A} = 1) \end{aligned}$$

Тогда параметр  $w_m$  окна определён выражением  $w_m = 1/(V_{Ejk} V_{Ajn})$  и  $w_i = w_m/c_i$ ,  $w_s = w_i/c_s$ .

При использовании цилиндрической системы используются те же самые формулы, но вместо векторов  $X, X_0, U$  используются их проекции на плоскость с нулевой координатой  $z$ .

При отсутствии предложений SANG и INRA формула есть просто  $w_m = 1/V_{Ejk}$ .

При работе с весовым окном по рассеянию с ненулевой вероятностью возможен большой пробег частицы без изменения весов. Следовательно, дисперсии оценки растёт. Поэтому можно задать искусственное ограничение максимального оптического пути, проходимого без столкновений. Это максимальное число вводится следующей картой.

*ELCU* <максимальное значение>

Если разыгранный оптический путь больше, чем <максимальное значение>, то по достижении этой искусственной границы моделируется  $\delta$ -рассеяние. Если весовое окно не изменяется, то ничего не происходит, в противном случае может сработать расщепление или русская рулетка.

Карта доступна при наличии в файле параметров с расширением MEM ключа:

*+ELCUT*

## 15 ИСХОДНЫЕ ДАННЫЕ ДЛЯ МОДУЛЯ ВЫГОРАНИЯ

Рассчитываемая система состоит из произвольного числа материалов. Материалы могут быть четырёх типов:

- F – выгорает, выделяется энергия, содержит делящиеся изотопы;
- A – выгорает, энергия не выделяется, состав произволен;
- P – не выгорает, выделяется энергия, содержит делящиеся изотопы;
- N – не выгорает, энергия не выделяется, состав произволен.

Номера материалов типа F, A и P указываются в исходных данных, остальные материалы имеют тип N.

Как правило, тип F приписывается твэлам, тип A – органам СУЗ, тип P ориентирован на использование в модельных задачах.

Энергия выделяется в материалах типа F и P, а в материалах типа A и N равна нулю. Эффективная энергия деления  $E$  включает кинетическую энергию продуктов деления, энергию мгновенного излучения и запаздывающего распада радиоактивных ядер, энергию, выделяющуюся при захвате нейтронов без деления.

Расчёт выполняется в одном из двух режимов:

- с заданной зависимостью от времени мощности;
- с заданной зависимостью от времени потока нейтронов.

Режим задаётся пользователем. При расчётах реакторных задач, как правило, используется режим с заданной мощностью.

По умолчанию программа «настроена» на тепловые реакторы, для расчета быстрых систем необходимо в библиотеке BURN5 банка данных программы скопировать файл

PARAMETE.FST в PARAMETE.BUR. Для возврата к установкам для тепловых реакторов необходимо там же скопировать файл PARAMETE.THR в PARAMETE.BUR

Информация записывается в файле исходных данных для программы MCU после раздела исходных данных для управления шагом расчёта.

Для ввода данных BURNUP использует программные средства бесформатного ввода.

Исходные данные для модуля начинаются с обязательной строки BURN.

*BURN library*

Эта строка обозначает начало исходных данных для модуля BURNUP. Наличие этой строки обязательно.

*library* – имя библиотеки констант, записанной в файле с указанным именем. При отсутствии этого параметра расчёт производится со стандартной библиотекой констант. Имя стандартной библиотеки задаётся при генерации модуля.

Опция работы программы задаётся картой CODE. Наличие этой карты обязательно.

*CODE option*

Определяет название используемой опции работы модуля. Наличие этой строки обязательно.

*option* – может принимать следующие значения:

- RSTP – используется опция STEP;
- RFNL – используется опция FINAL;
- RDEL – используется опция DELAY;
- RFTB – используется опция FINTAB;
- RDTB – используется опция DELTAB;
- RFDN – используется опция FINDEN;
- RSOU – используется опция SOURCE.

Далее в произвольном порядке следуют карты для выбранной опции работы модуля.

Фрагмент исходных данных для модуля выгорания должен заканчиваться обязательной картой FINISH.

*FINISH commentary*

Служит признаком окончания ввода данных к модулю выгорания.

*commentary* - текст произвольного содержания.

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

При наличии ошибок в процессе работы модуля информация о них поступает в файл VAR.ERR.

При повторном запуске шага выгорания после ошибки необходимо предварительно удалить файл VAR.ERR.

## 15.1 ИСХОДНЫЕ ДАННЫЕ ОПЦИИ STEP

Для работы опции STEP должна быть определена интегральная мощность реактора или интегральный поток. Как правило, задаётся мощность. Поток задаётся при расчёте выгорания образцов в облучательном канале. Мощность или поток задаётся в виде кусочно-постоянной функции.

Кроме концентраций изотопов STEP сохраняет в рабочем архиве значения их сечений и потоки нейтронов для каждого временного шага. Они используются другими программами модуля BURNUP.

Файлы исходных данных:

- VAR.DAT – исходные данные (используется при первом обращении);
- VAR.INI – исходные данные (используется при всех обращениях, кроме первого);
- VAR.DB – исходная концентрация изотопов и объёмы выгорающих материалов (используется при первом обращении);
- VAR.SB – сечения изотопов и потоки в начале шага.

Файлы с результатами расчёта:

- VAR.DEN – концентрация изотопов в начале и конце временного шага;
- VAR.PDC – концентрация изотопов в конце шага (используется для расчёта состояния);
- VAR.FNL – архив результатов всех шагов.

Рабочие файлы:

- VAR.D – исходная концентрация изотопов (только при первом обращении);
- VAR.SIG – сечения изотопов в начале и конце всех посчитанных временных шагов;
- VAR.FLU – значения мощности и потока нейтронов в начале и конце всех посчитанных временных шагов;
- VAR.PRT – протокол последнего обращения к любой подпрограмме модуля BURNUP, при первом обращении к STEP сюда записывается список изотопов, отсутствующих в константной базе MCU;
- END\_TIME – флаг конца расчёта всех шагов.

При анализе результатов работы STEP пользователю могут быть полезны файлы VAR.DEN и VAR.PRT.

Далее приводятся строки, определяющие режим работы модуля при выбранной опции STEP.

*FISZ list*

Список материалов типа F. Параметр *list* определяет номера материалов типа F. Номера материалов в списке *list* указываются интервалами. Нечётные числа обозначают начала интервалов, чётные – их окончания.

#### Пример 1

FISZ 3

#### Пример 2

FISZ 3 3

В примерах 1 и 2 только материал номер 3 имеет тип F.

#### Пример 3

FISZ 3 3, 5 8, 15

В примере 3 тип F имеют материалы номер 3, с 5 по 8, 15

*ABSZ list*

Список материалов типа A. Те же правила ввода, что и для строки FISZ. Может отсутствовать.

*POWZ list*

Список материалов типа P. Те же правила ввода, что и для раздела FISZ. Может отсутствовать.

Одна из строк FISZ или ABSZ со списком материалов должна присутствовать обязательно, иначе программа выдаст сообщение об ошибке.

POWE q1 t1 q2 t2 q3 t3 ...

Устанавливается режим работы "заданная мощность" и, одновременно задаётся суммарное по всем материалам энерговыделение  $Q$  в зависимости от времени  $T$ .

Единицы измерения:  $Q$  – киловатты,  $T$  – сутки.

$Q(T)$  представляется как кусочно-постоянная функция и вводится парами чисел  $q$  (энерговыделение на данном временном интервале) и  $t$  (верхняя граница временного интервала).

#### Пример 4

POWE 0.2

В примере 4  $Q = 0.2$  при всех  $T$ .

#### Пример 5

POWE 0.2 500.

#### Пример 6

POWE 0.2 500., 0.

В примерах 5 и 6  $Q = 0.2$  при  $T \leq 500.$ ;  $Q = 0$  при  $T > 500.$

#### Пример 7

POWE 0.2 500., 0.1 600., 0.3 1000.

В примере 7  $Q = 0.2$  при  $T \leq 500.$ ;  $Q = 0.1$  при  $500. < T \leq 600.$ ;  $Q = 0.3$  при  $600. < T \leq 1000.$ ;  $Q = 0$  при  $T > 1000.$

DPOW q1 t1 q2 t2 q3 t3 ...

Устанавливается режим работы "заданная мощность" и, одновременно задаётся суммарное по всем материалам энерговыделение  $Q$  в зависимости от времени  $T$ .

Единицы измерения:  $Q$  – киловатты,  $T$  – сутки.

$Q(T)$  представляется как кусочно-постоянная функция и вводится парами чисел  $q$  (энерговыделение на данном временном интервале) и  $dt$  (временной интервал);  $t_i = t_{i-1} + dt_i$ .

#### Пример 8

DPOW 0.2

В примере 8  $Q = 0.2$  при всех  $T$ .

#### Пример 9

DPOW 0.2 500.

### Пример 10

DPOW 0.2 500., 0.

В примерах 9 и 10  $Q = 0.2$  при  $T \leq 500.$ ;  $Q = 0$  при  $T > 500.$

### Пример 11

DPOW 0.2 500., 0.1 600., 0.3 1000.

В примере 11

$Q = 0.2$  при  $T \leq 500.$ ;

$Q = 0.1$  при  $500. < T \leq 1100.$ ;

$Q = 0.3$  при  $1100. < T \leq 2100.$ ;

$Q = 0$  при  $T > 2100.$

*Можно использовать только строку POWE или строку DPOW, иначе программа выдаст сообщение об ошибке.*

FLUX f1 t1 f2 t2 f3 t3 ...

Устанавливается режим работы "заданный поток нейтронов" и, одновременно, задаётся средний по всем материалам поток  $\Phi$  в зависимости от времени  $T$ .  $\Phi(T)$  представляется как кусочно-постоянная функция и вводится парами чисел  $f$  (поток,  $\text{н}\cdot\text{см}^{-2}\cdot\text{с}^{-1}$ , на данном временном интервале) и  $t$  (верхняя граница временного интервала).

Правила ввода строки те же, что и для строки POWE.

*Одна из строк FLUX, POWE или DPOW должна присутствовать обязательно, иначе программа выдаст сообщение об ошибке.*

STEP t1 n1 t2 n2 t3 n3 ...

Задаются значения последовательные значения времени  $T$ . Значения задаются парами:  $t$  – время,  $n$  – количество шагов на временном отрезке  $T_i - T_{i-1}$  ( $T_0 = 0$ ).

Единицы измерения – сутки.

### Пример 12

STEP 1000.

### Пример 13

STEP 1000. 1

В примерах 12 и 13 задан один временной шаг  $dT = 1000.$

### Пример 14

STEP 1000. 4, 3000. 3, 3500. 1

В примере 14 задано четыре шага размером  $dT = 1000./4$ , далее три шага  $dT = 2000./3$  и один шаг  $dT = 500.$

DSTEP t1 n1 t2 n2 t3 n3 ...

Задаются значения последовательных отрезков  $T$ . Значения задаются парами:  $t$  – длина временного отрезка,  $n$  – количество шагов на этом временном отрезке.

Единицы измерения – сутки.

### Пример 15

DSTP 1000.

### Пример 16

DSTP 1000. 1

В примерах 15 и 16 задан один временной шаг  $dT = 1000$ .

### Пример 17

DSTP 1000. 4, 3000. 3, 3500. 1

В примере 17 задано четыре шага размером  $dT = 1000./4$ , далее три шага  $dT = 3000./3$  и один шаг  $dT = 3500$ .

*Можно использовать только строку STEP или строку DSTP, иначе программа выдаст сообщение об ошибке.*

*Одна из строк STEP или DSTP должна присутствовать обязательно, иначе программа выдаст сообщение об ошибке.*

COLI k1 t1 k2 t2 k3 t3 ...

Устанавливается режим работы "интерполяция сечений и потоков" и, одновременно, задаётся признак линейной интерполяции микросечений и потоков нейтронов при решении уравнений изотопной кинетики по всем материалам в зависимости от времени  $T$ .

Единицы измерения:  $t$  – сутки.

COLI представляется как кусочно-постоянная функция и вводится парами чисел  $k$  и  $t$ ; где  $k$  – признак интерполяции микросечений и потоков на данном временном интервале ( $k = 0$  – сечения и поток нейтронов рассматриваются постоянными;  $k = 1$  – линейное изменение сечений и потоков в данном временном интервале, используется итерационная процедура предиктор-корректор) и  $t$  - верхняя граница временного интервала. Если строка с COLI не введена, то по умолчанию для всех временных интервалов задаётся режим, когда сечения и поток нейтронов рассматриваются постоянными ( $k = 0$ ).

### Пример 18

COLI 1

В примере 18  $k = 1$  при всех  $T$ .

### Пример 19

COLI 0 500.

В примере 19  $k = 0$  при  $T \leq 500.$ ;  $k = 1$  при  $T > 500.$

### Пример 20

COLI 0 500., 1 600., 0 1000.

В примере 10  $k = 0$  при  $T \leq 500.$ ;  $k = 1$  при  $500. < T \leq 600.$ ;  $k = 0$  при  $600. < T \leq 1000.$ ;  $k = 1$  при  $T > 1000.$

eps – минимальное значение концентрации изотопов при передаче информации в MCU.

Если концентрация изотопа меньше, чем eps, то в MCU этот изотоп не передаётся.

Если eps = 0, передаются все изотопы с концентрацией, отличной от нуля.

Если eps < 0, передаются все изотопы, в том числе и с концентрацией, равной нулю.

Строка может отсутствовать, в этом случае используется стандартное значение (eps = 0.0), задаваемое при генерации модуля.

## 15.2 ИСХОДНЫЕ ДАННЫЕ ОПЦИИ FINAL

Опция позволяет получить концентрацию изотопов при временах, не совпадающих с временами, определёнными в файле исходных данных VAR.DAT при работе с опцией STEP.

В представлении кусочно-линейной зависимости (см. описание карты COLI) требуемые значения локальной мощности фрагмента зоны и сечений нейтронов получаются интерполяцией между вычисленными ранее и сохранёнными в архиве узловыми точками. При кусочно-постоянном представлении (см. описание карты COLI) значения мощности фрагментов зоны и сечения рассматриваются постоянными в соответствующих временных интервалах, заданных при работе с опцией STEP.

Результаты расчёта во всех, основных и дополнительных временных точках записываются в файл VAR.FNL. Он заменяет файл, сформированный ранее опцией STEP.

Обращение к FINAL может выполняться многократно. В исходных данных задаются только значения дополнительных временных точек. Для остальных параметров используются те же значения, что и при работе STEP.

Входные файлы: VAR.DAT, VAR.INI, VAR.D, VAR.SIG, VAR.FLU

Результат: VAR.FNL.

*TIMP* t1 n1 t2 n2 t3 n3 ...

Задание дополнительных временных точек. Исходные данные записываются так же, как в строке STEP опции STEP.

*Строка TIMP должна присутствовать обязательно, иначе программа выдаст сообщение об ошибке.*

## 15.3 ИСХОДНЫЕ ДАННЫЕ ОПЦИИ DELAY

Опция позволяет рассчитывать зависимость концентраций изотопов от времени в остановленном реакторе. Работает после опции STEP или FINAL.

Исходными данными являются время остановки и временные точки после остановки (время выдержки). По умолчанию время остановки – максимальное время в исходных данных STEP или FINAL. Время выдержки может быть задано в любых единицах времени: секунды, минуты, часы, дни, годы.

Обращение к DELAY может выполняться многократно. Результаты расчёта записываются в файл VAR.DLT той же структуры, что и VAR.FNL.

Входные файлы: VAR.DAT, VAR.INI, VAR.FNL.

Результат: VAR.DNL

*TIMP* t1 n1 t2 n2 t3 n3 ...

Задаёт время прекращения облучения. Исходные данные записываются так же, как и для опции FINAL. Используется максимальное значение времени.

Строка может отсутствовать. В этом случае используется максимальное время при последнем обращении к FINAL или STEP, если обращения к FINAL не было.

*TSEC* t1 n1 t2 n2 t3 n3 ...

```
TMIN  t1 n1 t2 n2 t3 n3 ...
THOU  t1 n1 t2 n2 t3 n3 ...
TDAY  t1 n1 t2 n2 t3 n3 ...
TYEA  t1 n1 t2 n2 t3 n3 ...
```

Эти строки задают времена после окончания облучения для расчёта и вывода на печать концентрации изотопов и данных о запаздывающем излучении. Отличаются разделы единицами измерения. Это, соответственно, TSEC – секунда, TMIN – минута, THOU – час, TDAY – сутки, TYEA – год. Отсчёт времени задаётся строкой TIMP.

Исходные данные записываются так же, как в строке TIMP.

```
GLIB  library
```

Задаёт названия библиотеки выходов гамма-квантов.

*library* – имя библиотеки, записанной в файле с указанным именем. При отсутствии этого параметра, расчёт производится со стандартной библиотекой. Имя стандартной библиотеки задаётся при генерации модуля.

## 15.4 ИСХОДНЫЕ ДАННЫЕ ОПЦИЙ FINTAB И DELTAB

Опция FINTAB предназначена для постобработки и печати результатов работы STEP и FINAL. Выводятся результаты последнего обращения к FINAL, а если обращения не было, то результаты опции STEP.

Постобработка включает в себя расчёт следующих интегральных характеристик материалов и системы в целом:

- мощность, кВт;
- энергия, выделившаяся за время работы, МВт·сут/кг $U_{исх}$ ;
- выгорание, г/кг $U_{исх}$ ;
- коэффициент размножения нейтронов  $K_{\infty}$ ;
- макроскопические сечения  $\Sigma_c$ ,  $\Sigma_f$ ,  $\Sigma_{n,2n}$ ,  $\nu\Sigma_f$ ;
- поток нейтронов, нормированный на мощность.

На печать выводятся интегральные характеристики, концентрации изотопов и их макроскопические сечения поглощения и деления. Концентрации и сечения могут быть упорядочены по убыванию их значений.

Общий объём информации велик, поэтому обеспечена возможность выборочной печати. Может быть напечатана только нужная информация в конкретных материалах и временных точках. Какая именно информация интересует пользователя, указывается в исходных данных.

Обращение к FINTAB может выполняться многократно.

Входные файлы: VAR.DAT, VAR.INI, VAR.SIG, VAR.FNL.

Результат записывается в файл VAR.REZ. Ранее существовавший файл уничтожается.

Опция DELTAB предназначена для постобработки и печати результатов работы последнего обращения к опции DELAY.

Постобработка включает в себя расчёт следующих интегральных характеристик материалов и системы в целом:

- активность, Ки;
- энергия, выделяющаяся при распаде, Вт;
- энергия, выделяющаяся в виде гамма-излучения, Вт;
- локальное энерговыделение, Вт (полная энергия минус энергия гамма-квантов распада, минус энергия тормозного излучения; тормозное излучение только для  $UO_2$ );
- спектр нейтронов спонтанного деления и реакции ( $\alpha, n$ ) на кислороде;
- медицинский стандарт, м<sup>3</sup>; объём, в котором следует растворить материал, после чего воду можно безопасно использовать;



- спектр гамма-излучения; включает тормозное излучение (только для UO<sub>2</sub>);
- спектр нейтронов спонтанного деления и реакции (α,n) на кислороде.

На печать выводятся интегральные характеристики материалов, концентрация изотопов и их радиационные параметры. Концентрация и параметры могут быть ранжированы по убыванию их значений.

Так же как и в FINTAB обеспечена возможность выборной печати.

Обращение к DELTAB может выполняться многократно.

Входные файлы: VAR.DAT, VAR.INI, VAR.DNL.

Результат записывается в файл VAR.RDZ. Ранее существовавший файл уничтожается.

*ZONP list*

Задаёт значения номеров материалов, для которых результаты расчёта изменения изотопного состава выводятся на печать.

Информация записывается так же, как для карт FISZ, ABSZ и POWZ.

Строка может отсутствовать, в этом случае на печать выводятся характеристики всех материалов.

*TIMP t1 n1 t2 n2 t3 n3 ...*

Задаёт значения времени, при котором выдаются на печать значения концентрации изотопов и их характеристик. Исходные данные записываются так же, как и для опции FINAL. Строка может отсутствовать, в этом случае используются все значения времени.

Строка используется только опцией FINTAB. DELTAB выдаёт результат для всех рассчитанных времён.

*SUMZ list*

Указание о выводе на печать интегральных характеристик всей системы и отдельных материалов.

Список *list* содержит указатели для вывода характеристик:

- SUMB – характеристики выгорания всей системы;
- ZONB – характеристики выгорания отдельных материалов;
- SUMS – макроскопические сечения всей системы;
- ZONS – макроскопические сечения отдельных материалов;
- SUMR – радиационные характеристики всей системы;
- ZONR – радиационные характеристики отдельных материалов;
- SUMG – спектры гамма-квантов всей системы;
- ZONG – спектры гамма-квантов отдельных материалов.

Любая величина в списке может быть опущена, тогда соответствующая характеристика на печать не выводится. При пустом списке в опции FINTAB выводятся характеристики выгорания всей системы (SUMB), в опции DELTAB выводятся радиационные характеристики всей системы (SUMR).

### Пример 1

SUMZ      SUMB    ZONB    SUMS    ZONS    SUMR    ZONR    SUMG    ZONG

*CONT list*

Указание о выводе на печать характеристик всех изотопов.

Список *list* содержит указатели для вывода характеристик:

- DENS – концентрация;

- SIGM – макроскопическое сечение.
- CURI – активность;
- HEAT – энерговыделение при радиоактивном распаде;
- HGAM – энергия гамма квантов при радиоактивном распаде;
- HBET – энергия тормозного излучения электронов и позитронов при радиоактивном распаде
- QUOT – объём воды, нужный для разбавления изотопа до допустимой концентрации;
- SPNU – число нейтронов спонтанного деления и (alpha,n) реакции.

Элементы списка могут следовать в произвольном порядке; любой элемент может быть опущена, тогда соответствующая характеристика на печать не выводится; при пустом списке выводится концентрация (DENS).

При использовании опции FINTAB выводятся характеристики DENS, SIGM; при использовании опции DELTAB выводятся все характеристики, кроме SIGM.

### Пример 2

```
CONT    DENS    HEAT    CURI    QUOT    SPNU    SIGM
```

```
ACTI list
```

Указание о выводе на печать характеристик продуктов деления.  
Те же правила ввода, что и для строки CONT.

```
FISP list
```

Указание о выводе на печать характеристик продуктов деления.  
Те же правила, что и для строки CONT.

```
LIST list
```

```
PARA list
```

В строке LIST задаётся список изотопов, а в строке PARA задаются их характеристики.

Элемент в списке строки LIST задаётся его химическим символом, а изотоп – в обозначениях MCU.

Характеристики в списке строки PARA задаются по тем же правилам, что и для карты CONT. При отсутствии этой карты на печать выводится только концентрация (DENS).

### Пример 3

На печать выводится концентрация всех изотопов заданного списка элементов. В данном примере это U, Pu и Gd.

```
LIST    U    Pu    Gd
```

### Пример 4

На печать выводится концентрация всех изотопов урана и изотопов PU39, PU40 и XE35.

```
LIST    U    PU39    PU40    XE35
```

### Пример 5

На печать выводится концентрация (DENS) и макроскопическое сечение (SIGM) U235, PU39 и всех изотопов Gd.

```
LIST    U235  PU39  Gd
PARA    DENS,  SIGM
```

```
SIZI    eps  t
```

Эта строка позволяет упорядочить данные в порядке их убывания и вывести на печать только достаточно большие значения. Строка может отсутствовать, в этом случае характеристики изотопов выводятся в порядке возрастания значений  $Z$  и  $A$ .

#### Пример 6

```
SIZI
```

#### Пример 7

```
SIZI    0.
```

В примерах 6 и 7 на печать выводятся все изотопы. Упорядочивание производится по последнему значению времени.

#### Пример 8

На печать выводятся все изотопы. Упорядочивание производится по 3-му значению времени.

```
SIZI    0.  3
```

#### Пример 9

Упорядочивание производится по последнему значению времени; изотопы, суммарный вклад которых в относительное значение параметра не превышает величину 0.01, на печать не выводятся.

```
SIZI    0.01
```

#### Пример 10

То же, что и в предыдущем примере, но упорядочивание производится по 3-му значению времени.

```
SIZI    0.01  3
```

## **15.5 ИСХОДНЫЕ ДАННЫЕ ОПЦИИ FINDEN**

Опция формирует исходные данные для расчёта нейтронно-физических характеристик выгоревшего реактора.

Исходными данными служит длительность кампании, которая должна обязательно присутствовать в исходных данных опции STEP или FINAL.

Входные файлы: VAR.DAT, VAR.INI, VAR.D.

Результат записывается в файл VAR.NEW того же формата, что и файл VAR.PDC.

```
TIMP    t1  n1  t2  n2  t3  n3  ...
```

Задание времени работы реактора. Исходные данные записываются так же, как и для опции FINAL. Используется последнее значение  $t$ .

Строка может отсутствовать. По умолчанию используется максимальное время работы реактора.

## 15.6 ИСХОДНЫЕ ДАННЫЕ ОПЦИИ SOURCE

Опция служит для формирования источников запаздывающего излучения (фотонов и нейтронов).

Запаздывающим излучением обычно называют излучение радиоактивных ядер, накопленных за время работы реактора. В основном это продукты деления и минорные актиноиды. В работающем реакторе запаздывающее излучение вносит примерно 30 % в радиационный нагрев его материалов, а в остановленном реакторе это единственный источник энергии.

Расчёт источника проводится для максимального времени в опции DELAY, а если DELAY не использовалась, то для максимального времени в опции STEP.

Входные файлы: VAR.DAT, VAR.INI, VAR.FNL.

Результат работы программы записывается в файл VAR.SOU.

Данная опция не требует данных.

## 15.7 ПРИМЕЧАНИЕ

Если при работе модуля выгорания имелась ошибка, её диагностика записывается в файл NAME.ERR.

При повторном запуске модуля выгорания после исправления ошибки следует:

- удалить файл NAME.ERR;
- в первой строке файла VAR.SYS заменить имеющуюся цифру на 6.

## 16 ОПИСАНИЕ ДИРЕКТОРИЙ ПРОГРАММЫ

Директория MDBPT50 содержит файлы банка ядерных данных. Названия её поддиректорий соответствуют названиям библиотек, входящих в состав банка ядерных данных.

Директория MСUPTR содержит средства для расчёта задач. В её состав входят следующие поддиректории:

- поддиректория MСUDOC содержит документацию по программе;
- поддиректория EXE предназначена для хранения исполняемых файлов программы;
- поддиректория MСUOFF3 содержит файлы программы MСU Office;
- поддиректория МАКЕМСU содержит средства генерации программы;
- поддиректория MEMORY содержит файлы, определяющие параметры генерации программы;
- поддиректория RESTEST содержит результаты расчётов тестовых вариантов из директории RUN;
- поддиректория RUNTEST содержит исходные данные тестовых задач и средства запуска тестовых расчётов;
- поддиректория TEXTGUR содержит исходные тексты программы.

## 17 УСТАНОВКА ПРОГРАММЫ

Создайте папку на диске, в которой вы хотите разместить программу. Скопируйте (распакуйте) в эту папку содержимое дистрибутива.

После этого необходимо скорректировать с учётом имени этой папки пути в переменных pathexe и pathdb файла MСUPTR\EXE\msu5.bat.

Так, например, при установке программы в папку C:\MСUPTR необходимо изменить две строки в файле C:\MСUPTR\MСUPTR\EXE\msu5.bat следующим образом:

```
set pathexe = C:\MСUPTR\MСUPTR\EXE\  
set pathdb = C:\MСUPTR\MDBPT50\  

```

## 18 ГЕНЕРАЦИЯ ПРОГРАММЫ

Для получения исполняемых модулей (exe файлов) программу необходимо сгенерировать, т.е. перевести тексты с языка Гуртран на язык Фортран 95 и скомпилировать полученный текст.

Перевод с языка Гуртран на язык Фортран осуществляется программой MCUREP (Инструкция по использованию программы MCUREP находится в Приложении А, однако простому пользователю ознакомление с ней не требуется). В процессе перевода используется информация, записанная в так называемом «файле памяти» (такие файлы располагаются в папке MEMORY и имеют расширение MEM). Эти файлы содержат ключи, определяющие параметры сборки рабочей программы (например, наличие или отсутствие в рабочей программе того или иного подмодуля физического модуля, использование одинарной или двойной точности и т.д.), а также максимальную длину некоторых массивов.

Для компиляции итогового текста на языке Фортран 95 можно использовать компиляторы g95 (распространяется по лицензии GNU и может быть свободно скачан с <http://www.g95.org>), Intel 10.

Для генерации программы необходимо в папке МАКЕМСУ выполнить команду:

```
makemcu.bat <M или F> <имя памяти> <компилятор> <опция>
```

<M или F> – определяет какой подмодуль физического модуля будет использоваться в области термализации. Используйте M или m для генерации программы с подмодулем МОФИТТГ, F или f для генерации программы с подмодулем ФИМТОЕН.

<имя памяти> – имя файла памяти из папки MEMORY без расширения MEM (см. п.3 Инструкции по использованию программы MCUREP);

<компилятор> – параметр, отвечающий за выбор компилятора с языка Фортран 95. Используйте G или g для g95, I или i для Intel, C или c для Compaq. При отсутствии данного параметра компиляция программы будет выполняться с помощью компилятора Intel.

<опция> – Если данный параметр задан как g, то будет выполнен только перевод текстов программы с языка Гуртран на язык Фортран. 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. При отсутствии этого параметра выполняется полная генерация программы.

Настройки компиляторов записаны в файле setcomp.bat. При необходимости пользователь может изменять заданные в нем параметры, в частности включить оптимизацию.

### Пример

```
makemcu f pd i
```

## 19 ЗАПУСК ПРОГРАММЫ НА СЧЁТ

Программа устроена следующим образом. При запуске исполняемого (exe) модуля в той папке, из которой был выполнен запуск, программа ищет файл с фиксированным именем MSU5.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. Ниже описан один из наиболее часто используемых способов.

Каждый вариант рассчитывается в своей собственной папке, в которую копируется упоминавшийся выше файл msu5.bat.

Файл mcs5.bat запускается из папки с вариантом с помощью командной строки или какого-либо файл-менеджера, например, FAR, доступного по адресу <http://www.rarsoft.com>. Эта команда имеет вид:

```
mcs5.bat <M или F> <имя варианта> <опция>
```

<M или F> — определяет какой подмодуль физического модуля будет использоваться в области термализации. Используйте M или m для использования подмодуля МОФИТТГ, F или f для использования подмодуля ФИМТОЕН.

<имя варианта> — имя варианта.

<опция> — при отсутствии 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

```
mcs5 f inputdata i
```

## 20 ТЕСТИРОВАНИЕ

В папке RUNTEST находятся файлы исходных данных для тестирования работы программы. В папке RESTEST находятся файлы с результатами, полученными для версии программы, собранной с помощью компилятора G95.

Для запуска тестового варианта достаточно запустить файл runtest.bat, находящийся в одной папке с исходными данными для тестового варианта.

## 21 MCU OFFICE

Программный комплекс MCU Office является графическим интерфейсом для программы MCU. Основными его функциями являются следующие:

- визуализация исходных данных для геометрического модуля NCG 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;
- запуск задачи на счёт в отдельном окне DOS;
- Кроме того, пользователю предоставляется ряд дополнительных возможностей:
- генерация рабочей версии программы MCU;
- сравнение двух рисунков;
- отображение цветом на рисунке скоростей реакций, полученных в процессе расчёта;
- отображение в текстовом виде на рисунке скоростей реакций, полученных в процессе расчёта, а также номеров материалов, зон, объектов и другой информации;
- и др.

После установки программы рекомендуется создать на рабочем столе ярлык к программе MCUPTR\MCUOFF3\mcsioffice.exe. Для этого необходимо щёлкнуть по правой кнопке мыши на рабочем столе, выбрать Создать->Ярлык и следовать указаниям системы.

С помощью этой программы можно, в том числе, осуществить ввод команд на генерацию программы и запуск варианта на счёт с помощью диалоговых окон, привычных для пользователя Windows.

На рисунке 21.1 приводится диалоговое окно, предназначенное для генерации программы, которое можно вызвать с помощью пункта меню Starter->Make MCU.

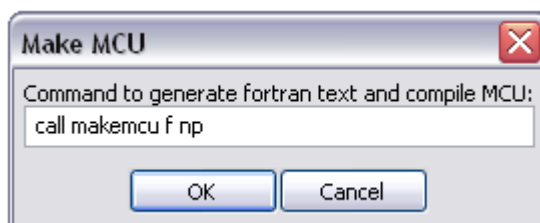


Рисунок 21.1 – Диалоговое окно, предназначенное для генерации программы

При работе с MCU Office файл исходных данных для программы MCU вызывается с помощью команды меню File->Open, поэтому в приводимом на рисунке 21.2 окне диалога для запуска задачи на счёт (Starter->Run MCU) вместо имени варианта используется символ «\*», который будет автоматически заменён на имя варианта при выполнении команды.

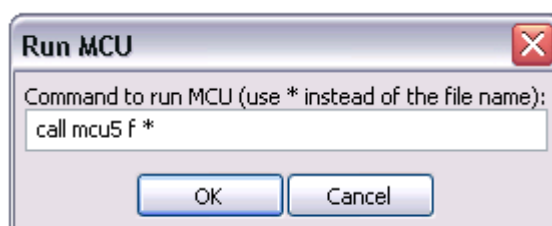


Рисунок 21.2 – Окно диалога для запуска задачи на счёт

Обратите внимание на то, что в обоих случаях перед командой используется слово call. Это не обязательно, однако иногда оказывается полезным.

В обоих случаях в рабочей папке MCUOFF3 программы mscuoffice.exe создаётся файл пакетной обработки (MakeMCU.bat в первом случае и RunMCU.bat во втором), который содержит все команды, необходимые для выполнения соответствующей операции, и запускается автоматически.

## 22 УСЛОВИЯ ПРИМЕНЕНИЯ

Тексты программы с банком данных и документацией занимают около 2 Гб на жёстком диске, рекомендуемый минимальный объём оперативной памяти – 1 Гб на одно ядро процессора.

### 22.1 НА WINDOWS КЛАСТЕРЕ

Кластером в данном случае считается система, которую пользователь «видит» как компьютер с несколькими процессорами, у которых общая дисковая и оперативная память. В этом смысле обычный компьютер с двухъядерным процессором является кластером.

Процедура установки программы на Windows кластер не отличается от стандартной. При генерации многопроцессорной версии программы в файле памяти следует включить ключи:

*+UNIMPI*  
*+RNDMAR*

Можно использовать файл памяти `rtgmpi.mem`, в котором эти ключи уже включены.

Программа поставляется с настройками, предполагающими, что в качестве библиотеки MPI будет использован пакет MPICH2, установленный в папку «C:\Program

Files\MPICH2\», а в качестве компилятора с языка Фортран 90 будет использован Intel® Visual Fortran Compiler. Данные настройки вызываются при генерации программы указанием `impi` в качестве параметра <компилятор>. Общий синтаксис команды при генерации, например, с подмодулем ФИМТОЕН будет следующим:

```
makemcu f ptrmpi impi
```

При генерации программы может потребоваться отключить опцию оптимизации, обычно установленную по умолчанию. Для этого необходимо исправить установки компилятора, которые меняются так же, как и в случае с однопроцессорной версией программы.

Для запуска программы следует пользоваться командой `mscu5mpi.bat` (файл `MCUPTR\EXE\mscu5mpi.bat`, в котором, как и в случае стандартной установки, необходимо скорректировать пути в переменных `pathexe` и `pathdb`). В нем также предполагается использование пакета `MPICH2`.

Эта команда имеет вид, аналогичный стандартной команде, однако добавляется указание требуемого количества процессоров:

```
mscu5mpi.bat <М или F> <имя варианта> <опция> <количество процессоров>
```

<М или F> — определяет какой подмодуль физического модуля будет использоваться в области термализации. Используйте `M` или `m` для использования подмодуля `МОФИТТГ`, `F` или `f` для использования подмодуля `ФИМТОЕН`.

<имя варианта> — имя варианта.

<опция> — при отсутствии последовательно выполняются все три шага, возможны следующие значения:

- `A` – последовательное выполнение всех шагов, аналогично отсутствию этого параметра в стандартной версии;
- `I` – только ввод;
- `C` – только счёт;
- `F` – только финальная обработка;
- `B` – только шаг выгорания;
- `D` – удалить файлы, созданные при предыдущем запуске.

<количество процессоров> – количество процессоров (ядер), на которых будет выполняться этап счета.

Пример (для четырех процессоров):

```
mscu5mpi f inputdata a 4
```

Выбор интерфейса `MPICH2`, обусловлен тем, что он является высокопроизводительной и наиболее широко распространённой реализацией стандарта `Message Passing Interface (MPI)` (как `MPI-1`, так и `MPI-2`). `MPI` в свою очередь является наиболее распространённым стандартом интерфейса обмена данными в параллельном программировании. Важной особенностью `MPICH2` является то, что это свободно распространяемое программное обеспечение.

Выбор `Intel Fortran` в качестве компилятора непосредственно связан с использованием интерфейса `MPICH2`, который и рекомендует его в качестве такового.

При этом использование `MPICH2` и компилятора `Intel` не является принципиальным требованием. Достаточно использовать любую реализацию `MPI` с любым компилятором `Fortran 90/95`, при условии, что с их помощью можно скомпилировать работающее в многопроцессорном режиме приложение. При этом будет необходимо перенастроить соответствующие `bat` файлы программы `MSU`, или воспользоваться следующим способом. Выполняется генерация программы с опцией `g`. После этого в папке `MAKEMCU\tmp`



образуется файл MSU.F90, который содержит полный фортранный текст рабочей версии, соответствующей заданным в файле памяти параметрам. Этот файл компилируется с учётом требований, предъявляемых выбранными MPI и Фортран 90/95.

## 22.2 В МНОГОПРОЦЕССОРНОМ РЕЖИМЕ НА РАЗЛИЧНЫХ КОМПЬЮТЕРАХ В СЕТИ WINDOWS С MPICH2

Ниже описывается рекомендуемый способ работы с программой при проведении многопроцессорных расчётов на нескольких компьютерах с ОС Windows, связанных между собой по сети.

Для проведения таких расчётов рекомендуется использование свободно распространяемого интерфейса MPICH2, который является высоко-производительной и наиболее широко распространённой реализацией стандарта Message Passing Interface (MPI) (как MPI-1, так и MPI-2). MPI в свою очередь является наиболее распространённым стандартом интерфейса обмена данными в параллельном программировании. Важной особенностью MPICH2 является возможность одновременно использовать процессоры на физически различных компьютерах.

В качестве компилятора в этом случае рекомендуется Intel Fortran, что непосредственно связано с рекомендациями по использованию интерфейса MPICH2.

Возможно использование любой другой реализации MPI с любым компилятором Фортран 90/95, при условии, что с их помощью можно скомпилировать приложение, работающее в многопроцессорном режиме на физически разных компьютерах. При этом будет необходимо перенастроить соответствующие bat файлы программы MSU, а возможно и создать необходимые вспомогательные программы.

Далее будет описан применяемый на практике способ использования MPICH2.

На всех компьютерах заводится пользователь с одним и тем же именем и паролем и правами администратора, например, пользователь MSU с паролем MSUPASS.

MPICH2 устанавливается на все компьютеры в папку в папку «C:\Program Files\MPICH2\» и регистрируется для пользователя MSU.

На каждом компьютере создаётся папка с одним и тем же именем, например, «C:\MSU\». К этой папке открывается свободный доступ по сети.

На одном из компьютеров, далее называемом основным, в эту папку устанавливается программа MSU.

Процедура установки программы не отличается от стандартной. При генерации многопроцессорной версии программы в файле памяти следует включить ключи:

```
+UNIMPI  
+RNDMAR
```

Можно использовать файл памяти ptrmpi.mem, в котором эти ключи уже включены.

Программа поставляется с настройками, предполагающими, что в качестве библиотеки MPI будет использован пакет MPICH2, установленный в папку «C:\Program Files\MPICH2\», а в качестве компилятора с языка Фортран 90 будет использован Intel® Visual Fortran Compiler. Данные настройки вызываются при генерации программы указанием impi в качестве параметра <компилятор>. Общий синтаксис команды при генерации, например, с подмодулем ФИМТОЕН будет следующим:

```
makemcu f ptrmpi impi
```

Все расчёты выполняются в этой же папке «C:\MSU\».

Перед расчётом в папке с файлом исходных данных для MSU должен находиться файл с фиксированным именем mscmpirun.job. Этот файл определяет компьютеры, участвующие в расчёте и количество процессоров, которые будут задействованы для каждого такого компьютера.

Файл имеет следующую структуру.

В первой строке с первой позиции записывается имя пользователя, для которого на всех компьютерах устанавливался MPICH2. В соответствии с примером выше это MCU.

Во второй строке записывается соответствующий этому имени пароль. В соответствии с тем же примером это MCUPASS.

Далее возможно использование комментарных строк. Такие строки должны иметь символ «\*» в первой позиции.

Каждая некомментарная строка содержит сетевое имя компьютера, который будет использован в расчёте, затем двоеточие и после него число, определяющее количество процессоров, которые будут задействованы в расчёте на этом компьютере.

Имя того компьютера, с которого запускается счёт, должно быть указано первым.

Пустые строки не допускаются.

### Пример

```
MCU
MCUPASS
* This is an example file that defines the job for mpi at several PCs.
ani2:1
  denis :2
      gur : 4
```

В этом примере задача запускается на счёт с компьютера ani2, на котором будет задействован 1 процессор, кроме того будет задействовано 2 процессора на компьютере denis и 4 процессора на компьютере gur.

Для запуска программы следует пользоваться командой mcs5mpibynet.bat (файл MSCUPTR\EXE\mcs5mpibynet.bat, в котором, как и в случае стандартной установки, необходимо скорректировать пути в переменных pathexe и pathdb). В нем также предполагается использование пакета MPICH2.

Эта команда имеет вид, аналогичный стандартной команде:

```
mcs5mpibynet.bat <M или F> <имя варианта> <опция>
```

<M или F> — определяет какой подмодуль физического модуля будет использоваться в области термализации. Используйте M или m для использования подмодуля МОФИТТГ, F или f для использования подмодуля ФИМТОЕН.

<имя варианта> — имя варианта.

<опция> — при отсутствии последовательно выполняются все три шага, возможны следующие значения:

- A – последовательное выполнение всех шагов, аналогично отсутствию этого параметра в стандартной версии;
- I – только ввод;
- C – только счёт;
- F – только финальная обработка;
- B – только шаг выгорания;
- D – удалить файлы, созданные при предыдущем запуске.

### Пример

```
mcs5mpibynet f inputdata
```

Поскольку при счёте на не основные компьютеры производится копирование ряда файлов, и на их дисках создаются рабочие файлы, то удалять файлы после расчёта лучше с помощью опции D, которая стирает файлы не только на основном, но и на остальных

компьютерах. Рекомендуется периодически контролировать удаление файлов с помощью удалённого доступа, либо непосредственно на каждом компьютере.

В заключение, хочется отметить, что основным требованием описанной схемы расчёта является одинаковое расположение всех файлов на участвующих в расчёте компьютерах. Доступность всех папок с исполняемым модулем и исходными данными для MCU на чтение и запись с любого компьютера.

## 22.3 НА LINUX КЛАСТЕРЕ

Кластером в данном случае считается система, которую пользователь «видит» как компьютер с несколькими процессорами, у которых общая дисковая и оперативная память. В этом смысле обычный компьютер с двухъядерным процессором является кластером.

Ниже описывается рекомендуемый способ работы с программой на Linux кластере.

Суть всех действий по подготовке программы к использованию на кластере сводится к следующему. Необходимо

- переписать в директорию пользователя на кластере банк ядерных данных (библиотеку) программы;
- переписать в директорию пользователя на кластере фортранный текст рабочей версии программы с необходимой пользователю функциональностью;
- средствами программного обеспечения кластера этот текст оттранслировать;
- средствами программного обеспечения кластера запустить задачу на счёт в многопроцессорном режиме.

### 22.3.1 Пересылка банка ядерных данных программы на кластер

Пересылка банка ядерных данных программы на кластер осуществляется с помощью стандартных средств связи с кластером. Рекомендуется пересылать библиотеку в виде одного архивного (zip или tar) файла, а после распаковать её непосредственно на кластере.

### 22.3.2 Пересылка фортранного текста рабочей версии программы на кластер

Для получения рабочей версии программы необходимо установить её на компьютере пользователя обычным способом. При этом если на данном компьютере не планируется выполнять вычисления, то наличия на ней компилятора с языка Фортран не требуется, однако операционная система должна быть Windows, кроме того, можно не устанавливать библиотеку программы.

После установки программы необходимо выполнить её генерацию с опцией g. После этого в папке МАКЕМCU\tmp образуется файл MCU.F90, который содержит полный фортранный текст рабочей версии, соответствующей заданным в файле памяти параметрам.

Для большей конкретности можно переименовать полученный MCU.F90 по шаблону «<M или F><имя памяти>.f90». Например, программа, сгенерированная с подмодулем ФИМТОЕН и памятью nmpi.mem, будет называться «fnmpi.f90» (это имя будет использовано во всех примерах ниже).. Например, программа, сгенерированная с подмодулем ФИМТОЕН и памятью nmpi.mem, будет называться «fnmpi.f90» (это имя будет использовано во всех примерах ниже).

Далее этот текст необходимо переслать на кластер.

Примечание - Если файла памяти предназначен для получения рабочих версий для многопроцессорных вычислений, то он обычно содержит слово «mpi». В таком файле активны ключи UNIMPI и RNDMAR.

### 22.3.3 Трансляция текста рабочей версии на кластере

Для трансляции текста рабочей версии на кластере рекомендуется использовать отдельную директорию, поскольку в процессе её будет создано много служебных файлов.

Собственно трансляция осуществляется средствами кластера командой (зависит от кластера):

```
mpif90 -O3 fnmpi.f90 -o fnmpi.exe
```

### 22.3.4 Запуск задачи на счёт в многопроцессорном режиме

Рекомендуется проводить каждый расчёт в своей собственной директории, что связано с большим количеством создаваемых в процессе расчёта файлов. В эту директорию копируется файл исходных данных. В этой же директории создаётся файл MCU5.INI (обязательно заглавными буквами!). 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).

#### Пример

```
fn206  
/share/home/mcuteam/MDBPT50/
```

Если скопировать исполняемый файл рабочей версии программы в эту же директорию, то запуск на счёт можно осуществить стандартной командой кластера (зависит от кластера):

```
mpirun -np 10 -maxtime 1440 ./fnmpi.exe
```

Этой командой мы поставили задачу в очередь для расчёта на 10 процессорах в течение 1440 минут, то есть на сутки.

Примечание - Иногда необходимо при указании исполняемого файла добавлять спереди точку и слеш «./» (зависит от кластера).

Каждой задаче автоматически присваивается номер.

## ЗАКЛЮЧЕНИЕ

В отчёте приводятся описание применения и инструкция по работе с программой MSU-PTR, которая позволяет проводить моделирование процессов переноса нейтронов и фотонов аналоговыми и весовыми методами Монте-Карло на основе оценённых ядерных данных в системах с трёхмерной геометрией изотопного состава материалов реактора.

Из особенностей программы по сравнению с предыдущими версиями программ семейства MSU отметим возможность параллельных вычислений с использованием многопроцессорных вычислительных комплексов, возможность моделирования фотонов, электронов и позитронов, переход на использование динамической памяти.

Кроме того, обеспечена возможность, позволяющая при решении уравнения переноса нейтронов вычислять законы рассеяния для нужных температур на этапе ввода исходных данных для рассчитываемого варианта.

Библиотека констант MDBPT50 содержит информацию для 375 изотопов.

Информации, приведённой в отчёте, достаточно для получения общего представления о возможностях программы и схеме её работы, для составления задания исходных данных расчётного варианта и анализа результатов расчёта.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

MCU	– Monte Carlo Universal
Модуль	– одна или несколько программных единиц с предписанной архитектурой пакета предназначением и межмодульными связями, которые регламентируются интерфейсами с полностью описанными точками входа и выхода
Банк данных	– определенная совокупность библиотек констант
Библиотека констант	– совокупность данных по свойствам ядер или данных, необходимых для математического моделирования физических процессов переноса определенного вида излучения и взаимодействия его с веществом, возможно, в определенном энергетическом диапазоне, которая получена из измерений физических величин или их теоретических оценок с применением некоторой компьютерной обработки теоретико-экспериментальной информации
Семейство программ MCU	– совокупность всех специализированных рабочих программ, собранных из модулей всех версий пакета MCU

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Kalugin M.A., Oleynik D.S., Shkarovsky D.A. Overview of the MCU Monte Carlo Software Package // *Annals of Nuclear Energy*, v. 82, p. 54–62, 2015.
- 2 Alekseev N.I., Bol'shagin S.N., Gomin E.A., Gorodkov S.S., Gurevich M.I., Kalugin M.A., Kulakov A.S., Marin S.V., Novosel'tsev A.P., Oleynik D.S., Pryanichnikov A.V., Sukhino-Khomenko E.A., Shkarovskiy D.A. and Yudkevich M.S. The Status of MCU-5 // *Physics of Atomic Nuclei*, v. 75(14), pp. 1634–1646, 2012
- 3 Гомин Е.А. Статус MCU-4. ВАНТ, сер.: Физика ядерных реакторов, вып. 1, М., 2006, стр. 6-32.