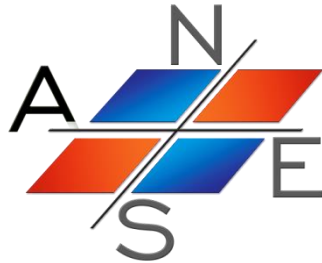


**Anes Team**



***ANES20XE: Код для численного моделирования процессов гидродинамики и теплообмена***

Версия 2.24

Работа с проектом пользователя

Москва 2019 г.

**Оглавление.**

<b>ВВЕДЕНИЕ</b> .....	<b>6</b>
<b>1. ОПИСАНИЕ ПРИКЛАДНОЙ ЗАДАЧИ ПОЛЬЗОВАТЕЛЯ</b> .....	<b>7</b>
1.1 Выбор и настройка макропеременных проекта .....	7
1.2 Выбор решаемых Ф-переменных и настройка их параметров .....	8
1.3 Настройка основных параметров решения .....	8
1.4 Описание расчетной области задачи .....	9
1.5 Задание свойств материалов PO .....	10
1.6 Описание источниковых членов .....	10
1.7 Описание граничных условий .....	10
1.8 Настройка параметров математических моделей кода .....	11
1.9 Настройка параметров численного алгоритма Решателя .....	11
1.10 Настройка содержимого файлов результатов расчета .....	11
1.11 Создание сеточных и скалярных переменных пользователя .....	12
1.12 Настройка Фортран-интерфейса Решателя .....	12
<b>2. СТРУКТУРА И ОПЕРАТОРЫ ФАЙЛА ПРОЕКТА</b> .....	<b>13</b>
2.1 Структура AIL операторов .....	14
2.2 Макропеременные проекта .....	14
2.3 V-переменные проекта .....	16
2.3.1 Постоянное значение .....	17
2.3.2 Формула .....	17
2.3.3 Таблица .....	17
2.3.4 Виртуальная функция .....	18
2.4 Подсистема myForm .....	18
2.4.1 Формулы для сеточных переменных Решателя .....	18
2.4.2 Формулы для скалярных переменных Решателя .....	20
2.5 Подсистема myTable .....	21
2.6 Виртуальные функции .....	22
2.7 Блок-схема работы Решателя .....	23
<b>3. СЕКЦИИ ОБЩИХ ПАРАМЕТРОВ</b> .....	<b>25</b>
3.1 Macro Variables. Описание макропеременных .....	25

3.2	Macro Sub. Описание макрофункций.....	25
3.3	Main. Общие переменные .....	26
3.4	Parall. Декомпозиция расчетной области.....	27
3.5	Special Data. Прямой интерфейс Решателя.....	28
<b>4.</b>	<b>СЕКЦИИ Ф-ПЕРЕМЕННЫХ И ПЕРЕМЕННЫХ ПОЛЬЗОВАТЕЛЯ.....</b>	<b>34</b>
4.1	PHI Variables. Настройка Ф-переменных.....	35
4.1.1	Общие операторы .....	36
4.1.2	Операторы для predetermined Ф-переменных.....	37
4.1.3	Операторы для произвольных Ф-переменных пользователя.....	37
4.2	User Variables. Настройка сеточных User-переменных пользователя.....	38
4.2.1	Внутренние сеточные User-переменные. ....	40
4.3	User Scalars. Настройка скалярных User-переменных пользователя .....	41
4.3.1	Внутренние скалярные User-переменные. ....	41
4.4	Vector Variables. Описание векторных полей.....	41
<b>5.</b>	<b>СЕКЦИИ НАСТРОЙКИ ПАРАМЕТРОВ РЕШАТЕЛЯ .....</b>	<b>42</b>
5.1	Solver. Настройка параметров итерационного процесса .....	42
5.2	Control. Настройка диалога Решателя .....	44
<b>6.</b>	<b>СЕКЦИИ ДЛЯ ОПИСАНИЯ ГЕОМЕТРИИ РАСЧЕТНОЙ ОБЛАСТИ.....</b>	<b>45</b>
6.1	Предварительная информация .....	45
6.1.1	Базовая расчетная область.....	45
6.1.2	Построение структурной сетки БРО.....	46
6.1.3	Формирование расчетной области.....	47
6.1.4	Геометрические объекты и патчи .....	48
6.1.5	Форматы файлов геометрических объектов .....	49
6.1.6	ГеоОбъекты и модификация сетки РО .....	51
6.2	XR(YFI , ZZ) GRID. Структурная сетка БРО.....	53
6.3	Unstructured Cartesian Grid. Неструктурная сетка .....	53
6.3.1	Базовые операторы построения неструктурной сетки .....	53
6.3.2	Специальные операторы построения неструктурной сетки .....	55
6.4	TIME GRID. Сетка по времени .....	56
6.5	PATCHES .....	57
6.5.1	Определение материала БРО .....	57
6.5.2	Геометрия. Стандартный патч.....	57
6.5.3	Геометрия. Стандартный повернутый патч .....	57
6.5.4	Укороченные патчи для внешних границ .....	57
6.5.5	Патчи с объектами 2Dbody .....	58
6.5.6	Патчи с объектами 3Dbody .....	58
6.5.7	Флаги-переключатели секции.....	58
6.5.8	Описание границ расчетной области .....	59
6.5.9	Описание Flow и Struct зон.....	60
6.5.10	Описание POROUS зон .....	60

6.5.11	Описание зон для источниковых членов .....	61
6.5.12	Цвета патчей .....	61
6.5.13	Разбиение патчей на суб-патчи.....	61
6.5.14	Патчи для описания свойств S-фазы .....	62
<b>7.</b>	<b>СПЕЦИФИКА ЗАДАЧИ .....</b>	<b>63</b>
7.1	Properties. Описание свойств материалов PO.....	63
7.1.1	Прямое задание свойств .....	63
7.1.2	Многокомпонентная G-фаза. Предопределенные Ф-переменные.....	64
7.1.3	Использование БД свойств.....	65
7.2	Internal Source. Предопределенные источники.....	66
7.2.1	Силы плавучести в уравнениях движения .....	66
7.2.2	Источники в уравнении энергии для TG .....	67
7.2.3	Стабилизированные и развитые течения .....	68
7.3	Turbulence. Настройка моделей турбулентности .....	68
7.3.1	Алгебраическая модель .....	69
7.3.2	K-ε и k-ω модели турбулентности .....	69
7.4	Boundary Condition. Задание граничных условий.....	69
7.5	Sources. Задание источниковых членов пользователя.....	71
7.6	Porous Model .....	72
<b>8.</b>	<b>ПЕРЕМЕННЫЕ ПОЛЬЗОВАТЕЛЯ .....</b>	<b>74</b>
8.1	User Event. Описание событий пользователя .....	74
8.2	User Fortran Variables. Переменные пользователя.....	74
8.3	User Fortran fields. Массивы пользователя.....	74
8.4	User Data. Передача исходных данных пользователя.....	74
8.5	User Fortran Subroutines. Произвольные подпрограммы .....	75

## Обозначения

Описание операторов файла проекта	- это основной текст документа,
CompileVersion = Y	- это описание АП-оператора или Фортрановского текста в файле описания проекта,
Oper= <i>Var1</i> ! Var1 / Var2/ Var3	- в АП-операторах значения, перечисленные с использованием символа “/”, указывают допустимые значения оператора; при этом первое значение, <i>выделенное красным курсивом</i> , является значением по умолчанию,
Value(Coef,Val [,Color])	- значения, указанные в квадратных скобках, являются необязательными параметрами,
<Имя файла>	- текст в угловых скобках обозначает понятие.

## **Введение**

В данном документе рассмотрены средства кода Anes, предназначенные для описания конкретной прикладной задачи пользователя. Основой для такого описания является файл проекта кода Anes. Файл проекта содержит специальные AIL-операторы языка проекта (Anes Input Language). Ниже эти операторы описаны подробно. Для подготовки и редактирования файла проекта можно использовать либо обычный текстовый редактор, либо диалоговый Препроцессор кода [1,2].

Для удобства «навигации» по AIL-операторам в первой главе документа рассмотрен алгоритм создания нового проекта и некоторые полезные советы. При этом даются ссылки на конкретные AIL-операторы и диалоговые компоненты Препроцессора.

## 1. Описание прикладной задачи пользователя

Самый простой способ познакомиться с правилами описания конкретной прикладной задачи - это изучить примеры, расположенные в каталоге примеров кода (каталог <anes>\project\_lib, см. документ [1]). Эти примеры можно использовать и для создания файла проекта для своей задачи. Для этого нужно просто скопировать файл проекта примера в свой рабочий каталог.

Но в любом случае (как при создании с нуля, так и коррекции другого файла проекта) для описания прикладной задачи необходимо выполнить следующие шаги:

1. Выбрать и настроить макропеременные проекта.
2. Выбрать решаемые Ф-переменные.
3. Настроить основные параметры решения.
4. Описать расчетную область (РО) задачи и построить сетки контрольных объемов.
5. Задать свойства материалов, присутствующих в РО.
6. Описать источники члены в уравнениях сохранения Ф-переменных.
7. Описать граничные условия для всех Ф-переменных на границе РО.
8. Настроить параметры дополнительных математических моделей кода.
9. Настроить параметры численного алгоритма Решателя.
10. Настроить содержимое файлов результатов расчета.
11. Создать сеточные и скалярные переменные пользователя для обработки результатов расчета.
12. Настроить, если необходимо, фортрановский интерфейс Решателя.

Ниже эти шаги рассмотрены подробно. Для удобства навигации пользователя на каждом шаге будет даваться ссылка на секцию файла проекта, в котором располагаются нужные APL-операторы и ссылка на диалоговое окно Препроцессора, в котором настраиваются эти APL-операторы. В Препроцессоре запуск нужного диалога осуществляется нажатием кнопки на панели инструментов главного окна Препроцессора [2]. Ниже для ссылки на нужный диалог будет использоваться название соответствующей кнопки.

### 1.1 Выбор и настройка макропеременных проекта

Макропеременные проекта - это символьные переменные, которые можно использовать во всех APL-операторах проекта. Им с помощью формул можно присвоить действительные или целые значения, которые будут «подставляться» в APL-операторах вместо имен макропеременных. В принципе макропеременные можно не использовать, но их введение может существенно облегчить изменение проекта.

Например, можно ввести макропеременную  $T_{IN}$ , которая содержит значение температуры потока на входе. Эта температура в файле проекта будет входить в несколько операторов проекта (граничные условия, формулы для расчета плотности, вязкости и т.д.). При наличии макропеременной ее значение нужно изменить только в точке ее вычисления, если ее нет, то изменения нужно вносить в разных APL-операторах различных секций (или диалоговых окнах Препроцессора).

В формулы для вычисления значений макропеременных могут входить другие макропеременные, поэтому секцию [Macro Variables] файла проекта или диалоговое окно «Макропеременные» Препроцессора можно рассматривать как обычный «пакетный» калькулятор, в котором можно провести любые предварительные вычисления параметров APL-операторов.

## 1.2 Выбор решаемых $\Phi$ -переменных и настройка их параметров

В соответствии с выбранными математическими моделями (см. документ [3]) необходимо «добавить» необходимые  $\Phi$ -переменные в список используемых  $\Phi$ -переменных и настроить их параметры. Соответствующие AIL-операторы располагаются в секции [PHI Variables] проекта. В Препроцессоре для настройки используется окно « $\Phi$ -переменные».

Напомним, что наряду с «предопределенными»  $\Phi$ -переменными (давлением, компонентами вектора скорости, температурами G- и S-фаз, и т.п.), пользователь может использовать свои собственные  $\Phi$ -переменные, связанные с G-, S- или G+S фазами. Эти  $\Phi$ -переменные добавляются так же, как и предопределенные, но для них нужно указать дополнительные параметры (которые для предопределенных  $\Phi$ -переменных указывать не нужно). Смысл этих параметров описан в документе [1] и ниже в описании секции [PHI Variables]. Заметим, что эти дополнительные параметры в Препроцессоре пользователь должен записать в виде AIL-операторов (для этого используется вторая закладка окна « $\Phi$ -переменные»).

При использовании двухпараметрических моделей турбулентности в Препроцессоре добавление  $\Phi$ -переменных kTUR, epsTUR и omTUR производится автоматически (при смене модели турбулентности). Поэтому вручную их добавлять не нужно.

Перечислим основные настраиваемые параметры  $\Phi$ -переменной:

1. *Коэффициент релаксации.* Он определяет скорость сходимости (или развала) итерационного процесса. Заметим, что в коде можно используются два механизма релаксации: линейная нижняя релаксация и «псевдовременная» FTS релаксация. Тип релаксации определяется знаком (см. [5]).
2. *Начальное значение.* Если выполнение расчета начинается «с нуля», то начальное распределение  $\Phi$ -переменной задается V-переменной оператора Init() файла проекта, что позволяет описать произвольное начальное распределение (об этом смотрите ниже). Если оператор Init() не задан, то в качестве начального значения используется ноль.
3. *Параметры линейного солвера* - точность решения дискретных уравнений, максимальное число внутренних итераций LES-солвера и имя самого LES-солвера. Наиболее интересным для пользователя является точность решения. Подробности можно найти в [5].

## 1.3 Настройка основных параметров решения.

К этой категории параметров относятся:

1. *Заголовок* расчета - произвольный текст, записываемый в файл результатов. Этот текст выводится в заголовок окна Постпроцессора Anes.
2. *Путь-префикс* файлов расчета - путь+имя файлов расчета без расширения файлов.
3. Используемая модель турбулентности.
4. Параметры начала расчета.
5. Выбор типа используемых сеток КО и сетки по времени.
6. Подключение базы данных свойств.

В файле проекта эти параметры задаются AIL-операторами секций [Main] и [Properties], в Препроцессоре для настройки параметров используется диалог «*Основные параметры*».

*Путь-префикс* файлов расчета задает путь к каталогу и имя для создания файлов. Решатель для создания различных файлов расчета использует разные расширения файлов. Например, файл с геометрией РО и сеточной информацией имеет расширение «.agr», файлы с полями  $\Phi$ -переменных для Постпроцессора Anes - «.acr», файл с полями результатов для постпроцессора ParaView - «.vtk». Заметим, что в всех примерах проектов в каталоге project\_lib в качестве пути используется подкаталог data рабочего каталога. Поэтому при копировании проекта в свой каталог не забудьте создать подкаталог data.



*Модель турбулентности* определяется пользователем в макрофункции Modeltur секции [Macro Sub] или в диалоге «**Основные параметры**». Модели турбулентности кода описаны в документе [3].

Для начала итерационного процесса Решателя необходимо проинициализировать значения  $\Phi$ -переменных. Возможно три варианта инициализации:

1. *Новый расчет* - для инициализации используются APL-операторы Init секции [PHI Variables].
2. *Старт с контрольной точки* - поля  $\Phi$ -переменных загружаются из файла контрольной точки (его расширение - «.acr»).
3. *Старт с другого варианта расчета* - поля  $\Phi$ -переменных загружаются из файла результатов Постпроцессора Anes (его расширение - «.ars»).

Файл контрольной точки (файл КТ) создается в процессе выполнения расчета. «Частота» создания файла КТ определяется шагом создания (APL-оператор StepOutCP секции [Main]). Отметим разницу использования старта с КТ и старта с варианта. В первом случае восстанавливаются все параметры предыдущего расчета (коэффициенты релаксации, шаг по времени и значение времени) и фактически продолжается предыдущий расчет. В случае старта с варианта расчета загружаются только поля  $\Phi$ -переменных.

Заметим, что при решении стационарных задач начальное распределение  $\Phi$ -переменных не принципиально, хотя оно влияет на число итераций. При решении нестационарных задач начальное распределение  $\Phi$ -переменных - это начальные условия при  $\tau = \tau_{\text{beg}}$  и они должны быть заданы правильно. Чаще всего в качестве такого распределения используется решение стационарной задачи. В этом случае для инициализации нужно использовать стационарный файл результатов.

Прежде чем проводить описание РО и построение сетки КО необходимо выбрать три основных параметра:

- 1) какая задача решается - стационарная или нестационарная;
- 2) какая система координат используется - декартовая или цилиндрическая;
- 3) какие сетки используются - структурные или неструктурные.

Настройка этих параметров осуществляется APL-операторами IsSteady и TypeMesh секции [Main].

При описании прикладной задачи необходимо для всех материалов G- и S-фаз задать свойства в секции [Properties] проекта. Это можно сделать либо самостоятельно, либо использовать базу данных (БД) свойств кода Anes. Активизация использования БД свойств производится APL-оператором IsUseDB секции [Properties]. В Препроцессоре этот оператор вынесен в диалог «**Основные параметры**». Заметим, что в Препроцессоре при включенном флаге БД свойств, нельзя ввести свои имена материалов. Имена можно выбирать из списка материалов базы данных.

#### **1.4 Описание расчетной области задачи**

Расчетная область (РО) в коде Anes создается в два этапа (см. документы [3,5]). На первом этапе создается базовая расчетная область (БРО), которая всегда представляет собой параллелепипед в обобщенных координатах (x,y,z) (декартовых или цилиндрических). На втором этапе в БРО помещаются геометрические объекты - патчи, которые используются для формирования границы РО и распределения G- и S- фаз в РО.

В файле проекта БРО строится одновременно с построением структурной сетки КО. Для этого используются операторы секций [XY grid], [YFI grid] и [ZZ grid]. При работе с неструктурными сетками эта структурная сетка используется как начальная грубая сетка для построения неструктурной сетки с локальным дроблением. Для описания параметров неструктурной сетки используются операторы секции [Unstructured Cartesian Grid].

Если решается нестационарная задача, то сетка по времени описывается операторами секции [Time Grid].

В Препроцессоре для описания сеток БРО и сетки по времени используются закладки диалога «Сетки КО».

Для описания патчей используются AIL-операторы секции [Patches] или диалог «Патчи» Препроцессора.

### 1.5 Задание свойств материалов РО

Для описания свойств материалов РО используются AIL-операторы Prop() секции [Properties] или диалог «Свойства» Препроцессора. Необходимые свойства должны быть заданы для всех материалов G- и S-фаз (список свойств и их имена - см. документ [3]). Заметим, что если используются Ф-переменные пользователя, определенные во всей РО (Space-переменные), то свойства необходимо задать для материалов G- и S- фаз.

Если используется БД свойств, то операторы Prop() задавать не нужно, они будут автоматически сформированы из БД.

Следует обратить внимание на AIL-операторы P0 и T0. Эти операторы определяют отсчет температуры и давления Решателя  $p_0$  и  $T_0$ . При использовании БД свойств предполагается, что суммы  $(T_g + T_0)$  и  $(T_s + T_0)$  - это температура в градусах Кельвина, а  $(P_f + p_0)$  - это давление в Паскалях.

Заметим, что в диалоге «Свойства» вводятся непосредственно AIL-операторы секции [Properties].

### 1.6 Описание источниковых членов

Источниковые члены пользователя описываются парными AIL-операторами Coef() и Val() секции [Sources]. В Препроцессоре для задания источников используется вторая закладка диалога «Источники». Алгоритм описания источников в коде Anes описан в документе [3].

Внутренние предопределенные источники, которые рассчитываются автоматически, описываются операторами секции [Internal Sources] или полями второй закладки диалога «Источники».

### 1.7 Описание граничных условий

Для замкнутой постановки задачи необходимо задать граничные условия для всех Ф-переменных на всей границе РО. В коде Anes граничные условия задаются как поверхностные источники, привязанные к различным участкам поверхности границы. Для привязки ГУ необходимо при описании РО «раскрасить» границу с помощью специальных патчей (см. документ [3]).

ГУ, как и источники пользователя, описываются парными AIL-операторами Coef() и Val() секции [Boundary Conditions]. В Препроцессоре для задания источников используется вторая закладка диалога «Граничные условия».

Для уменьшения количества задаваемых ГУ в коде Anes предусмотрены ГУ по умолчанию ([3]):

- 1) если граничный патч помечен флагом IsWall и на нем не заданы явные ГУ для компонент вектора скорости, то автоматически задаются условия  $U_{gx} = U_{gy} = U_{gz} = 0$ ;
- 2) если на границе не задано ГУ для давления, то граница считается непроницаемой ( $m_b = 0$ );
- 3) если на непроницаемой границе (плотность массового потока  $m_b = 0$ ) не заданы ГУ, то плотность потока Ф-переменной задается равной нулю;

- 4) если на выходной границе ( $m_b < 0$ ) не заданы ГУ, то автоматически задаются условия  $C_{\Phi,b} = BC\_MASSONLY$ ,  $V_{\Phi,b} = BC\_SAME$  (подробнее см. пункт 3.5.5 [3]);
- 5) если участок границы не связан ни с каким патчем, то на этом участке задаются нулевые плотности потоков для всех  $\Phi$ -переменных.

Пункт (4) позволяет не задавать граничные условия для  $\Phi$ -переменных на выходных границах. Однако при использовании интегральных функций подсистемы `myForm` для выходной границе необходимо явно задать условия  $C_{\Phi,b} = BC\_MASSONLY$ ,  $V_{\Phi,b} = BC\_SAME$ . В противном случае функции возвратят нулевые значения.

### 1.8 Настройка параметров математических моделей кода

При наличии в РО пористых зон необходимо выбрать для них модели FS - взаимодействия и задать параметры этих моделей. Для этого используются AIL-операторы секции [`Porous Models`] или диалог «Пористые модели» Препроцессора. Модели межфазного взаимодействия и их параметры описаны в документе [3].

Если была выбрана модель турбулентности, отличная от ламинарной (секция [`Macro Sub`]), то параметры модели задаются операторами секции [`Turbulence`] или в диалоге «Турбулентность» Препроцессора.

### 1.9 Настройка параметров численного алгоритма Решателя

Параметры, управляющие итерационным процессом решения дискретных уравнений, задаются AIL-операторами секций [`Solver`] и [`Control`]. В Препроцессоре для настройки этих операторов используется диалог «Параметры солвера». Смысл и назначение этих параметров описано в документах [3,5]. Перечислим наиболее часто используемые параметры:

1. *Критерий сходимости итераций.* В Решателе можно использовать три критерия (см. пункт 3.3 документа [5]) : среднюю невязку дискретных уравнений, максимальную невязку или максимальную коррекцию. По умолчанию используется средняя невязка.
2. *Максимальное число SWEEP-итераций.* При превышении этого числа итераций они прекращаются в любом случае.
3. *Численная схема и плотность на грани КО.* Эти параметры определяют алгоритмы расчета  $\Phi$ -переменной и плотности на грани КО (см. документ [5]).
4. *Имя линейного LES-Солвера.* Этот параметр задает имя линейного солвера, который по умолчанию используется для решения линеаризованных дискретных уравнений.
5. *Точка зануления давления  $P_f$ .* Если в ГУ не задано явно давление, то в этой точке Решатель принудительно зануляет гидродинамическое давление  $P_f$ .
6. *Окно сходимости и Окно мониторинга.* Эти параметры определяют информацию, передаваемую программе ДиалогСолвера.
7. *Точка мониторинга.* Точка в РО, значения  $\Phi$ -переменных в которой передаются в программу ДиалогСолвера.

### 1.10 Настройка содержимого файлов результатов расчета

При работе А-компилятора и Решателя создается несколько файлов результатов (см. документ [1]). Число таких файлов определяется операторами-флагами секции [`Main`]. Удобнее всего эти флаги настраивать в диалоге «Результаты» Препроцессора.

Наиболее важными флагами, определяющими размер выводимой информации, являются флаги создания ARS и VTK файлов результатов. Если не планируется использовать

постпроцессор ParaView, то нужно отключить создание VTK-файлов. При работе в ОС Linux, где отсутствует постпроцессор Anes, нужно отключить создание ARS-файлов.

Поля Ф-переменных выводятся в формате, удобном для построения «плавных» картинок. В частности, для неструктурной сетки вместо значений в центрах ячеек в файлы ARS и VTK выводятся проинтерполированные значения в вершинах ячеек. Если необходимо вывести значения в ячейках (и в постпроцессоре получить «истинные» значения полей), то необходимо использовать операторы секции [Special Data]:

```
L("ARS.OUTCELL") = .True.  
L("VTK.OUTCELL") = .True.
```

### 1.11 Создание сеточных и скалярных переменных пользователя

Пользователь может создать любое количество своих переменных, как связанных с сеткой КО (сеточные User-переменные), так и не связанные с сеткой (скалярные User-переменные).

Сеточные User-переменные аналогичны Ф-переменным, в Решателе они представляются трехмерными (для структурной сетки) или одномерными (для неструктурной сетки) массивами. Скалярные User-переменные являются обычными действительными переменными. Главное достоинство User-переменных - они записываются в файлы результатов для Постпроцессора Anes и их можно в нем проанализировать.

В отличие от Ф-переменных пользователя, которые рассчитываются в Решателе на основе своих уравнений сохранения, расчет значений User-переменных полностью лежит на пользователе. Для этого используются APL-операторы Calc() секций [User Variables] и [User Scalars]. В Препроцессоре для описания этих переменных используются диалоги «User переменные» и «User скаляры».

Для расчета значений User-переменных используются V-переменные файла проекта, поэтому пользователь может задать любые распределения (для сеточных переменных) и значения (для скалярных) User-переменных.

Сеточные User-переменные обычно используются:

- 1) для расчета вторичных полей Ф-переменных; например, для расчета модуля скорости;
- 2) для выдачи в файл результатов внутренних переменных Решателя, например, плотности, вязкости, коэффициента турбулентной вязкости и т.д.

Скалярные User-переменные обычно используются для обработки результатов расчета - расчета различных интегральных потоков и интегральных балансов.

Много примеров использования User-переменных можно найти в файлах проектов в каталоге примеров - <anes>\project\_lib.

Отметим важную особенность User-переменных. Они могут рассчитываться по алгоритму пользователя (заданного V-переменной) в разных «точках» Решателя. По умолчанию они рассчитываются перед записью файлов результатов.

### 1.12 Настройка Фортран-интерфейса Решателя

Если для задания алгоритма V-переменной используются виртуальные функции, то пользователь должен написать «куски» кода на языке Фортран. Совокупность этих «кусков» кода называется Фортран-интерфейсом Решателя.

Для подготовки Фортран-интерфейса используются секции [User Data], [User Event], [User Fortran Variables], [User Fortran Fields], [User Fortran Sub] и набор секций с именами [vf <Имя фортрановской функции>]. В Препроцессоре для работы с Фортрановскими секциями используется диалоги «Fortran-интерфейс» и «Виртуальные функции».

Фортрановский интерфейс Решателя подробно описан в документе [4].

## 2. Структура и операторы файла проекта

Файл проекта или А-файл - это секционный текстовый файл, состоящий из независимых блоков - секций. Секция начинается с заголовка секции (с первой позиции строки) [<Имя секции>]

Конец секции - это либо конец файла, либо начало другой секции. В А-файле имена и число секций фиксированы и не изменяются пользователем:

Таблица 2.1

Название секции	Назначение
Description	секция с произвольным текстом, описанием проекта, не используется А-компилятором;
<b>Общие параметры</b>	
Macro Variables	описание макропеременных проекта и операции с ними;
Macro Sub	вызов макрофункций;
Main	главная общая секция;
Parall	параметры декомпозиции расчетной области;
Special Data	прямой интерфейс с Решателем;
<b>Φ переменные и переменные пользователя</b>	
PHI Variables	настройка Φ-переменных;
User Variables	настройка полей пользователя;
User Scalars	настройка скалярных переменных пользователя;
Vector Variables	описание векторных полей;
<b>Параметры линейного солвера (LES-солвера)</b>	
Solver	параметры итерационного процесса;
Control	настройка диалога Решателя;
<b>Геометрия</b>	
XR Grid	построение сетки контрольных объемов (КО) по оси x;
YFI Grid	построение сетки КО по оси y;
ZZ Grid	построение сетки КО по оси z;
Unstructured Cartesian Grid	описание неструктурной декартовой сетки с локальным дроблением;
Time Grid	управление шагами по времени для нестационарных задач,
Patches	описание геометрических объектов расчетной области;
<b>Специфика задачи</b>	
Sources	описание источниковых членов пользователя;
Boundary Conditions	описание граничных условий;
Internal Sources	настройка внутренних источников;
Properties	описание свойств G- и S- материалов;
Porous Models	описание моделей взаимодействия G-S фаз;
Turbulence	описание модели турбулентности;
<b>Виртуальные функции</b>	
vf <Имя функции>	текст виртуальной функции;
.....	
<b>Переменные пользователя</b>	
User Event	описание функций событий пользователя;
User Fortran Variables	описание фортрановских переменных пользователя;
User Fortran Fields	описание сеточных массивов пользователя;
User Data	исходные данные пользователя;
User Fortran Subroutines	произвольные подпрограммы пользователя.

Все секции А-файла можно разбить на три типа:

- 1) секции с произвольным текстом - это секция Description;
- 2) секции с текстом на языке Фортран - это секции с виртуальными функциями [v XXXXX] и секции [USER Fortran variables] и [USER Fortran subroutines];
- 3) секции с AIL операторами.

## 2.1 Структура AIL операторов

AIL-оператор записывается на одной или нескольких строках. Для продолжения оператора необходимо в конце поставить пробел и символ "&". При обработке строк секции все символы, начиная с символа "!", считаются комментарием и удаляются при компиляции оператора. Допускаются три формы оператора

```
Aname(Aindex) = Par
Aname(Aindex) = AFUN(Par1,Par2, .... ParN)
AFUN(Par1,Par2, .... ParN)
```

где

Aname	-имя А-переменной, для описания которой используется оператор (может отсутствовать),
Aindex	-индекс для А-переменных - массивов; индекс задается либо в виде <i>целого</i> числа, либо в виде <i>символьной</i> строки, либо как имя <i>макропеременной</i> (может отсутствовать);
AFUN	-имя оператора А-переменной (может отсутствовать),
Par, Par1 ... ParN	-параметры оператора, их число определяется смыслом А-переменной.

Имя А-переменной Aname записывается с использованием *любого* регистра букв, при этом пробелы внутри имени не допускаются. В общем случае параметры Aindex и Par1 ... ParN могут принимать три типа значений:

- 1) символьное значение - произвольный текст в *двойных* кавычках ( "xxx" ),
- 2) целое или действительное значение,
- 3) формула с макропеременными.

Числовые значения задаются по правилам записи целых и действительных констант Фортрана:

```
12 , 102.245 , 1.2e-5
```

## 2.2 Макропеременные проекта

Для удобства подготовки числовых параметров в состав AIL включены так называемые макропеременные, над которыми можно производить любые арифметические вычисления.

Допускается использовать три типа макропеременных – действительные, целые и символьные. Макропеременные *описываются* в секции [Macro Variables] с помощью операторов

```
Macro(Mname1, Mname2, ....)
MacroInt(Mname1, Mname2, ....)
MacroChar(Mname1, Mname2, ....)
```

Первый оператор описывает действительные макропеременные, второй – целые, третий – символьные.

Значение символьной переменной не должно превышать 128 символов. Единственная операция с символьными переменными – это присвоение им значений. Сами символьные переменные могут входить в *любые выражения* AIL-операторов в виде (обрамление процентами)

```
%<Имя переменной>%
```

Такое вхождение имени переменной заменяется на ее значение (исключая начальные и конечные пробелы), например:

```
.....
MacroChar(sTH)
sTH = "-QV/2*XP^2 + XP/LX*(1+QV/2*LX^2)"
.....
calc("TS_TH") =MyForm(%sTH%)
```

Отметим, что все вычисления с действительными и целыми макропеременными всегда производятся как с действительными числами, и только при использовании значения целой макропеременной осуществляется ее округление.

Значение макропеременной вычисляется в секции [Macro Variables] с помощью оператора

```
Mname = <Формула>
```

где в правой части могут использоваться:

- числа в любом формате;
- любое число круглых парных скобок;
- знаки операций «- + / \*» и операцию возведения в степень - «^» или «\*\*»;
- имена макропеременных (символьные – в «%»), значение которым присвоено *выше* по тексту секции;
- арифметические функции:  
sin, cos, tan, asin, acos, atan, exp, log, log10, abs, sqrt(квадратный корень), sqr (квадрат), int (целая часть), tanh (гиперболический тангенс);
- арифметические функции двух аргументов:  
sign(a,b) - значение b со знаком a,  
max(a,b) , min(a,b) - максимум или минимум из двух значений;
- функция выбора:  
two(aYes, aNo, bCond) - если bCond = 1, то значение aYES, в противном случае aNO,
- «логические» функции для функции two:  
GE(a,b) - значение 1, если a >= b, в противном случае значение = 0,  
GT, LE, LT, EQ, NE - аналогичные функции.

Макропеременные, описанные и определенные в секции [Macro Variables], можно использовать в качестве значений параметров в APL операторах *любой* секций, за исключением секций с фортрановским кодом. Более того, можно использовать в качестве значения параметров APL-операторов любые формулы с макропеременными (нельзя только вводить новые макропеременные).

#### Три замечания:

1. при обработке А-файла уже существует набор системных макропеременных, которые определены в файле  
<anes>\etc\system.a  
Вводимые макропеременные пользователя *не должны* совпадать с именами системных макропеременных.
2. При работе с *диалоговым Препроцессором* (Дизайнером проекта) нельзя использовать символьные макропеременные.
3. При редактировании файла проекта в *диалоговом Препроцессоре* в макропеременных не допускается использование функций с двумя и тремя аргументами (sign, max, min, two, GE,...). Эти функции можно использовать при редактировании файла проекта в текстовом редакторе.

### 2.3 V-переменные проекта

В файле проекта используются три основных типа АПЛ операторов:

- 1) операторы для описания геометрии расчетной области (РО): zone, patch и т.д;
- 2) операторы инициализации A-переменных (действительных, целых, символьных и логических), типичные примеры – различные логические флаги;
- 3) операторы, служащие для описания *алгоритма расчета значений переменных Решателя или User-переменных пользователя.*

Смысл первых двух типов очевиден, третий тип требует пояснения.

Решатель представляет собой код на языке Фортран и Си. Внутри Решателя используется большое число переменных (целых, действительных, массивов, списков, структур и т.д.). Часть переменных являются внутренними переменными, а для других «внешних» переменных пользователь должен задать алгоритм их расчета. Типичные примеры таких переменных - это свойства фаз, коэффициенты и значения источниковых членов и граничных условий, начальные распределения Ф-переменных и т.д.. Фактически это все параметры математических моделей кода, для которых в их описании в документе [3] используется понятие V-переменная.

Как уже отмечалось выше, пользователь к переменным Решателя может добавить произвольное число своих собственных User-переменных двух типов: сеточных и скалярных (см. раздел 1.11). Для этих переменных также нужно задать алгоритмы расчета.

В принципе пользователь может получить полный доступ ко всем переменным Решателя и своим User-переменным, если он будет использовать Фортран-интерфейс Решателя. Однако в составе кода Apes есть специальные эффективные средства, которые позволяют без фортрановского кодирования описать алгоритмы расчета. Эти средства реализуются с помощью *V-переменных файла* проекта.

Условно «внешние» переменные Решателя и User-переменные пользователя можно разделить на две группы: сеточные и скалярные переменные. *Сеточные* переменные – это массивы, связанные с контрольными объемами (КО) сетки. Типичный пример сеточной переменной Решателя является плотность G-фазы. При использовании структурной сетки плотность хранится в трехмерном массиве

```
RhoG(ixc,iyc,izc),
```

при использовании неструктурной сетки – в одномерном массиве

```
usRhoG(idCV),
```

связанном со списком неструктурных КО. Аналогичным образом хранятся и сеточные User-переменные пользователя:

```
F4(ixc,iyc,izc, UserID) - для структурной сетки,
```

```
usF4(idCV, UserID) - для неструктурной сетки,
```

где UserID - номер переменной пользователя, связанный с ее именем.

В Решателе нет «внешних» скалярных переменных, они используются только в переменных пользователя. *Скалярные* переменные пользователя – это действительные переменные, тем или иным способом, также связанные с сеткой КО или с сеткой по времени. Например, пользователь может их использовать для расчета: площади поверхностного патча, полного теплового потока на патче, сил, действующих на поверхностный патч, и т.д.

Общий алгоритм использования V-переменной для расчета *сеточной* переменной RhoG можно пояснить следующим псевдо-кодом (для структурной сетки):

```
do izC = 1, NZ
  do iyC = 1, NY
    do ixC = 1, NX
      RhoG(ixC,iyC,izC) = <алгоритм расчета V-переменной>
    end do
  end do
end do
```



```
end do
```

Общий алгоритм использования V-переменной для расчета *скалярной* переменной TotVal:

```
TotVal = 0.0
do izC = 1, NZ
  do iyC = 1, NY
    do ixC = 1, NX
      TotVal = TotVal + < алгоритм расчета V-переменной >
    end do
  end do
end do
```

Для определенности ниже в качестве примера будем использовать APL-оператор для описания плотности G-фазы для материала с именем «Air»:

```
Prop("Air.Dens") = <V-переменная>
```

Для описания V-переменной *в общем случае* используются следующие формы:

```
<V-переменная> = <Значение>
<V-переменная> = myForm(<Формула>)
<V-переменная> = myTable(<Описание таблицы>)
<V-переменная> = Virtual(<Имя функции Фортрана>)
<V-переменная> = External(<Имя функции Фортрана>)
<V-переменная> = Internal(<Имя функции Фортрана>)
```

### 2.3.1 Постоянное значение

Для задания постоянного значения V-переменной используется следующая форма:

```
<V-переменная> = <Значение>
```

где <Значение> - либо действительное значение, либо выражение с макропеременными, например

```
Prop("Air.Dens") = 1.25+Tbase**(-0.456)
```

где Tbase - макропеременная, определенная в секции [Macro Variables].

### 2.3.2 Формула

В этом случае алгоритм вычисления V-переменной определяется формулой, которая *интерпретируется в Решателе* с помощью подсистемы myFORM Anes:

```
<V-переменная> = myForm(<Формула>)
```

Идея myFORM и структура формулы описана ниже. Важно отметить, что при использовании формул, не требуется создания локальной версии Решателя. Для расчета, как и в случае постоянного значения, используется Решатель по умолчанию! И, самое главное, при использовании myFORM не нужно знать языка Фортран.

### 2.3.3 Таблица

В этом случае для расчета значений V-переменной используется интерполяция по одномерной таблице:

```
<V-переменная> = myTable(<Описание таблицы>)
```

В качестве аргумента интерполяции можно использовать разные сеточные переменные, например:

```
Prop("Air.Dens") = myTable("TG",10, 997.2, 100, 1002)
```

Более подробно модель myTABLE также описана ниже.

### 2.3.4 Виртуальная функция

Если пользователь хочет определить свой собственный *произвольный* алгоритм расчета V-переменной, необходимо использовать следующую форму представления:

```
<V-переменная> = Virtual(<Имя подпрограммы Фортрана>)
```

и определить подпрограмму на языке Фортран с указанным именем.

## 2.4 Подсистема myForm

Формулы, используемые в myForm, очень похожи на формулы для расчета значений макропеременных. Главное отличие – myForm формулы обрабатываются внутри Решателя, в то время как формулы для макропеременных обрабатываются при компиляции файла проекта. Структура формул V-переменных для описания сеточных и скалярных переменных Решателя немного отличаются, поэтому пишем их отдельно.

### 2.4.1 Формулы для сеточных переменных Решателя

При записи формул для сеточных переменных Решателя можно использовать:


- числа в любом формате;
- любое число круглых парных скобок;
- знаки операций «- + / \*» и операцию возведения в степень - «^» или «\*\*»;
- любые макропеременные (символьные – в «%»);
- стандартные арифметические функции:  
sin, cos, tan, asin, acos, atan, exp, log, log10, abs, sqrt(квадратный корень), sq(квадрат),  
int (целая часть) , tanh (гиперболический тангенс);
- стандартные арифметические функции двух аргументов:  
sign(a,b) - значение b со знаком a,  
max(a,b) , min(a,b) - максимум или минимум из двух значений;
- стандартные функция выбора:  
two(aYes, aNo, bCond) - если bCond = 1, то значение aYES, в противном случае aNO,
- стандартные «логические» функции для функции two:  
GE(a,b) - значение 1, если a >= b, в противном случае значение = 0,  
GT, LE, LT, EQ, NE - аналогичные функции.
- скалярные переменные Решателя: P0, T0, Time, dTime,  
DEV\_DPDZ, DEV\_FLOWRATE, DEV\_DTDZ, DEV\_TBULK, DEV\_TWALL, DEV\_QWALL;
- все Ф - переменные;
- все User - переменные;
- все скалярные User - переменные;
- вспомогательные поля Решателя (плотность, кинематическая вязкость, пористость, теплопроводность G-фазы):  
RhoG ,NuG, Pore, LamdaG;
- сеточные массивы для *структурной* сетки:  
XP, YP, ZP, XFACE, YFACE, ZFACE, Volume, AreaW, AreaS, AreaL;

- сеточные массивы для *неструктурной* сетки:  
XP, YP, ZP, Volume;
- коэффициенты дискретных уравнений для структурной и неструктурной сеток:  
APG, APS, RpG, RpS, FCOR, Coef12G, Coef12S, GamG, GamSX, GamSZ, GamSY;
- коэффициенты дискретных уравнений только для структурной сетки:  
AWG, AWS, AEG, AES, ASG, ASS, ANG, ANS, ALG, ALS, MFlowX, MFlowY, MFlowZ;

Наряду со стандартными функциями можно использовать специфические для CFD-задач функции. Список функций в текущей реализации приведен в таблице 2.2

Таблица 2.2

Функция	Назначение
Функции для структурной и неструктурной сеток	
Grad_X("P#<Ф-имя>"), Grad_Y("P#<Ф-имя>"), Grad_Z("P#<Ф-имя>")	- расчет производных Ф-переменной по координатам x,y,z (компонент вектора градиента). Аргумент функции – строка с именем Ф-переменной и префиксом P#. Например: "P#Tg", "P#UgX"
YPLUS()	- расчет в пристенных КО значение $y^+$ (остальные КО содержат ноль). Подробнее – пункт 9.5.1 документа [1]
L2ZONE()	- расчет распределения пристенных зон и ядра потока для двухслойной модели турбулентности. Подробнее – пункт 9.6.5 документа [4]

 **Замечание.** В формулах в качестве операндов можно использовать как имена макропеременных, так и имена переменных Решателя, например имена Ф-переменных. Как подсистема myFORM различает их? Алгоритм очень прост – на этапе компиляции производится подстановка всех макропеременных проекта, все остальные имена считаются переменными Решателя или User-переменными. Их обработка производится уже при выполнении Решателя. Если такой переменной нет, то Решатель выдаст сообщение об ошибке.

Приведем некоторые примеры использования формул для сеточных переменных.

Пример № 1: Расчет плотности воздуха:

$$\text{Prop}(\text{"Air.Dens"}) = \text{MyForm}(3.48\text{E-}3 * (\text{P0} + \text{PF}) / (\text{TG} + \text{T0}))$$

здесь

P0, T0 - скалярные переменные Решателя (отсчет давления и температуры),

PF, TG - Ф-переменные (давление и температура G-фазы).

Пример № 2: Расчет точного решения для температуры в кольцевом теле, которая сохраняется в User-переменной с именем «TS\_TH»:

[Macro Variables]

```
macro(ROUT,RIN,CENTRE)
  ROUT = 1.0
  RIN = 0.4
  CENTRE = 1.1*ROUT
macro(T00,T11)
  T00 = 1
  T11 = 2
```

[User Variables]

```
calc("RZZ") = MyForm(SQRT((XP-CENTRE)**2+(YP-CENTRE)**2))
calc("TS_TH") = MyForm(T00 + (T11-T00)*log(RZZ/RIN)/log(ROUT/RIN))
```

здесь

RIN, ROUT, T00, T11, CENTRE - макропеременные,

XP, YP - переменные Решателя (координаты центра КО),

RZZ - сеточная User-переменная пользователя (радиус цилиндрической системы координат), которая используется для расчета точного распределения.

#### 2.4.2 Формулы для скалярных переменных Решателя

Основное отличие формул для скалярных переменных – в них можно использовать только:

- - макропеременные проекта,
- - скалярные User-переменные,
- - скалярные переменные Решателя P0, T0, Time, dTime, DEV\_DPDZ, DEV\_FLOWRATE, DEV\_DTDZ, DEV\_TBULK, DEV\_TWALL, DEV\_QWALL;
- интегральные функции myFORM.

Список интегральных функций приведен в таблице 2.3.

Таблица 2.3

Функция	Назначение
<b>Функции для структурной и неструктурной сеток</b>	
TotFlux("<Ф-имя >", "<Имя патча>")	- функция рассчитывает полный поток на поверхностном патче или полный источник на объемном патче <Имя патча> для Ф-переменной <Ф-имя >
AveragePHI ("<Ф-имя >", "<Имя патча>")	- функция рассчитывает среднее значение Ф-переменной <Ф-имя > на любом патче
AreaPatch("<Имя патча>")	- функция возвращает площадь патча <Имя патча>
VolPatch("<Имя патча>/#")	- функция возвращает объем патча <Имя патча>; если имя патча = "#", то возвращается объемом <i>всей</i> расчетной области.
AverageSect("<Ф-имя >", iDir, sBeg, sEnd)	- функция возвращает средне-интегральное значение Ф-переменной в объеме, ограниченном двумя сечениями (sBeg, sEnd) в направлении оси iDIR – 1,2,3
MinPHI ("<Ф-имя >", "<Имя патча>") MaxPHI ("<Ф-имя >", "<Имя патча>")	- функция рассчитывает минимальное и максимальное значение Ф-переменной на патче <Имя патча>
ForceP("X/Y/Z", "<Имя патча>")	- функция рассчитывает X(Y или Z) компоненту силы, действующую на поверхность тела (описываемую патчем BSWall или FS_) и связанную с полем давления на поверхности
ForceTAU("X/Y/Z", "<Имя патча>")	- функция рассчитывает X(Y или Z) компоненту силы, действующую на поверхность тела (описываемую патчем BSWall или FS_) и связанную с касательным напряжением на поверхности
TotPorousSource("<Ф-имя >", "<Имя FS-зоны>")	- функция рассчитывает полный источник пористого взаимодействия для произвольной Ф-переменной в пористой зоне <Имя FS-зоны> (это имя указывается последним параметром в Porous-патче).

TotTimeSource("<Ф-имя >")	- функция рассчитывает интеграл по объему расчетной области от нестационарного члена для произвольной Ф-переменной, например, для Ф-переменной G- фазы $S_{\text{time}} = \int_{\text{PO}} \frac{\partial(\rho_g \Phi)}{\partial \tau} dV$
TotMass()	- функция возвращает полную массу G-фазы в расчетной области
MassOnSurface("Имя Surface патча")	- функция рассчитываем полный массовый поток через патч типа Surface
PhiOnPoint("Ф-Имя",X,Y,Z)	- функция возвращает значение Ф-переменной в точке (x,y,z)
TotRadFLUX("Имя #R_патча")	- функция рассчитывает полный результирующий поток Qrad для радиационного патча при использовании S2S модели излучения

Приведем несколько примеров:

Пример № 1:

```
calc("Qwall") =MyForm(TotFlux("TG","Wall")/ AreaPatch("Wall"))
```

В этом примере скалярной User-переменной пользователя с именем QWall присваивается среднее значение плотности теплового потока на патче Wall.

Пример № 2:

```
calc("Twall") =MyForm(AveragePHI("TG","Wall"))
calc("Tfluid") =MyForm(AverageSect("TG",3,10,20))
calc("Alfa") =MyForm(TotFlux("TG","Wall")/ AreaPatch("Wall") / &
(TWall - Tfluid) )
```

В этом примере скалярной User-переменной пользователя Alfa присваивается значение среднего коэффициента теплоотдачи.

Интегральные функции в коле реализуются через набор функций Фортран-интерфейса. Подробности алгоритмов интегральных функций описаны в документе [2].

## 2.5 Подсистема myTable

Подсистема myTable предназначена для заполнения сеточной или скалярной V - переменной на основе интерполяции по одномерной таблице. Допустимы две формы вычисления:

```
<V-переменная> = = myTable("VarX",x1,y1,x2,y2,.....)
<V-переменная> = = myTable("VarX", "FILE:<Имя Файла>")
```

Здесь

VarX – аргумент для интерполяции:

Time - время,  
X,Y,Z - декартова координата,  
имя Ф-переменной,  
имя User-переменной.

В первой форме точки таблицы интерполяции (x1,y1), (x2,y2),... задаются непосредственно в вызове функции myTABLE.

Во второй форме таблица интерполяции задается в виде файла, имя которого указывается в последнем параметре myTABLE. Сам файл имеет следующую простую структуру:

- 1) первая строка содержит число точек интерполяции,
- 2) каждая следующая строка содержит пару чисел (x,y) разделенных пробелами или запятыми.

Приведем примеры первой формы, которая используется для задания коэффициента теплопроводности материала "Air" как функции температуры:

```
Prop("Air.Cond") = MyTable("TG", &
    273.0, 0.562, &
    293.0, 0.600, &
    315.0, 0.631, &
    340.0, 0.657, &
    366.0, 0.674, &
    373.0, 0.678 )
```

Вторая форма используется в основном для наполнения базы данных свойств. Поэтому поиск файла с данными интерполяции осуществляется в два этапа:


- 1) сначала в рабочем каталоге WorkDir Anes,
- 2) потом в каталоге <Anes>\property.

Например, для задания коэффициента теплопроводности воды при давлении 240 атм в функции температуры можно использовать оператор:

```
Prop("Water.Cond") = MyTable("TG","FILE:water_240\cond.prp")
```

Файл cond.prp расположен в подкаталоге water\_240 каталога <Anes>/property и имеет следующую структуру:

```
41
273.200000000000 0.576194317237548
292.110769230769 0.610684896316044
313.250307692308 0.641080157295951
336.078307692308 0.665856942544129
360.392153846154 0.684234885768249
.....
```

 **Замечание.** Если в качестве аргумента интерполяции VarX используются имена "TG" или "TS", то в качестве значения температуры используются суммы

TG + T0 и TS + T0.

Если используется имя "PF" - то в качестве значения давления используется сумма PF + P0.

Здесь P0 и T0 - постоянные отсчеты давления и температуры (операторы секции [Properties]). Это сделано для «упрощения» задания свойств материалов: нужно выбрать отсчет T<sub>g</sub> так, чтобы сумма (T<sub>g</sub>+T<sub>0</sub>) соответствовала температуре в градусах Кельвина, а сумма (P<sub>f</sub> + P<sub>0</sub>) - статическому давлению в Паскалях. Естественно в таблицах всегда нужно использовать Кельвины и Паскали.

## 2.6 Виртуальные функции

Если пользователь хочет определить свой собственный произвольный алгоритм расчета V-переменной, то необходимо использовать следующую форму:

```
<V-переменная> = Virtual(<Имя функции>)
```

и определить подпрограмму с указанным именем в секции файла проекта с именем [vi <Имя подпрограммы>] и следующим шаблоном

```
SUBROUTINE <Имя функции>(RetVal)
```

```
real(4) retVal
!-----
.....
!-----
END SUBROUTINE
```

Содержимое этой подпрограммы зависит от типа AIL-оператора, использующего V-переменную. В частности, не всегда необходимо использовать единственный аргумент подпрограммы – RetVal (например, для Event-переменных пользователя), часть параметров, необходимых для вычислений, передаются в виртуальную функцию через специальные COMMON-блоки (например, при расчете свойств материалов). Подробнее эти правила рассмотрены в документе [2].

Пользователь может поместить тексты виртуальных функций в отдельный фортрановский файл и подключить его к проекту на этапе сборки локального Решателя (подробнее см. документ [5]). В этом случае необходимо использовать другой вариант описания виртуальной функции

```
<V-переменная> = External(<Имя функции >)
```

Для описания свойств материалов в Базе Данных свойств используется еще одна форма для описания виртуальной функции

```
<V-переменная> = Internal(<Имя функции >)
```

В этом случае код подпрограммы *уже включен* в код Решателя. Для пользователя такой вариант описания используется при выборе модели F-S взаимодействия или выборе пристенных функций модели турбулентности.

Подробно работа с виртуальными функциями описана в документе [2].

## 2.7 Блок-схема работы Решателя

В ряде AIL-операторов, содержащих V-переменные, необходимо указывать место в алгоритме кода, в котором производится использование алгоритма V-переменной (например, при вычислении User-переменных). Это место в Anes называется «событие пользователя». Ниже приведены эти места для случаев решения стационарной и нестационарной задач.

При решении стационарных задач используется следующий алгоритм решения и порядок вызова событий пользователя:

```
<Setup> ! Инициализация переменных по умолчанию,
<Input> ! Чтение файла инициализации AID/AGR и выделение памяти под массивы
*Вызов «события пользователя» OnInput
<Init> ! Инициализация PHI-полей (по умолчанию и из файла результатов)
*Вызов «события пользователя» OnInit

! ----- Итерационный процесс -----
iSWEEP = 0
Liter = .TRUE.
DO WHILE(LITER)
  iSWEEP = iSWEEP + 1
  *Вызов «события пользователя» BeforeSweep
  <Расчет одной итерации> : *Вызов события OnLESolver
  *Вызов «события пользователя» AfterSweep
ENDDO
<Отчет Решателя>
*Вызов «события пользователя» OnReport
<Создание файлов результатов >

.....
```

STOP: \*Вызов «события пользователя» OnStop

При решении нестационарных задач используется более сложный алгоритм:

```

<Setup>    ! Инициализация переменных по умолчанию,
<Input>    ! Чтение файла инициализации AID/AGR и выделение памяти под массивы
*Вызов «события пользователя» OnInput
<Init>     ! Инициализация PHI-полей (по умолчанию и из файла результатов)
*Вызов «события пользователя» OnInit

! ===== Шаги по времени =====
iTIME = 0
Time = TimeBeg
DO WHILE(Time < TimeEnd)
  <Переприсвоение  $\Phi_k^0 = \Phi_k$ 
  Time = Time + Dtime
  *Вызов «события пользователя» BeforeTime
! ----- Итерационный процесс на шаге по времени -----
  iSWEEP = 0
  Liter = .TRUE.
  DO WHILE(LITER)
    iSWEEP = iSWEEP + 1
    *Вызов «события пользователя» BeforeSweep
    <Расчет одной итерации> : *Вызов события OnLESolver
    *Вызов «события пользователя» AfterSweep
  ENDDO
  *Вызов «события пользователя» AfterTime
  <Отчет на шаге по времени>
  *Вызов «события пользователя» OnReport
  <Создание файлов результатов на шаге по времени>
ENDDO
.....

STOP: *Вызов «события пользователя» OnStop

```



### 3. Секции общих параметров

Эта группа секций описывает общие параметры настройки Решателя. Она включает в себя секции [Macro Variables], [Macro Sub], [Main], [Parall] и [Special Data].

#### 3.1 Macro Variables. Описание макропеременных

Секция предназначена для описания имен используемых макропеременных и формул для вычисления их значений. Операторы этой секции были описаны выше. Приведем просто типичный пример:

```
macro(LX,LY, LZ)
LX = 0.2 ! в метрах
LY = 1*LX
LZ = 0.2*LX

macro(RhoF,NuF, CpF,LamdaF,PrF,BetaF) ! берем воздух
RhoF = 1.189
NuF = 1.544E-05
CpF = 1005.0
LamdaF = 0.0258
BetaF = 3.41E-3

macro(T1,T2)
T1 = 10.0
T2 = 0.0

Macro (Gr, gY)
Gr = 1.0E5
! Определяем силу тяжести из Gras
! Gr = BetaF *gy * (T2-T1) * LX^3 / NuF^2
gY = -1*GR/BetaF/ABS(T2-T1)/LX^2*NuF^2
```

Заметим, что процесс расчета значений производится в порядке следования формул.

#### 3.2 Macro Sub. Описание макрофункций

Эта секция используется для вызова макрофункций APL. Макрофункция - это файл с заготовками APL операторов для разных секций. Файлы макрофункций имеют расширение «.am» и располагаются в подкаталоге <anes>\etc. APL-операторы из этого файла добавляются в конец «заданных» секции при обработке вызова макрофункции

<Имя макрофункции>(<Параметр>)

При этом в зависимости от символьного значения <Параметр> могут генерироваться различные APL-операторы для различных секций.

В текущей версии в APL языке имеется одна макрофункция (и один файл modeltur.am):

ModelTur(trLAMINAR / trVARIANT / trKEWall / trKE2Layer / trKOMsst)

для выбора используемой модели турбулентности:

trLAMINAR	- моделируется ламинарное течение;
trVARIANT	- используется алгебраическая модель турбулентной вязкости, коэффициент турбулентной вязкости задается V-переменной в APL-операторе NuTUR секции [Turbulence];
trKEWall	- k-ε модель с пристенными функциями;
trKE2Layer	- двухслойная k-ε модель турбулентности;
trKOMsst	- k-ω и SST модели турбулентности.

### 3.3 Main. Общие переменные

Эта секция содержит следующие APL операторы.

VariantTitle = "Название варианта"

оператор определяет текст, который записывается в файлы результатов расчета и высвечивается в заголовке окна постпроцессора.

PrefixResFiles = <PreRes >

определяет префикс (путь и начало имени) файлов Решателя. Этот оператор должен быть обязательно указан.

NameStart = " " ! <путь к RES-Файлу>

путь к ".ars" или ".acr" файлу, используемому для инициализации Ф-переменных.

StartMode = **SM\_NEW** ! SM\_NEW/SM\_CP/SM\_RES

оператор определяет способ начала расчета:

SM\_NEW - стандартное начало, Ф-поля инициализируются Решателем на основании значения V-переменных операторов InitPHI(<Ф-имя>);

SM\_CP - нестационарный или стационарный расчет продолжается с файла рестарта (файла КТ – контрольной точки), путь к этому файлу указан в операторе NameStart;

SM\_RES - в качестве начального распределения Ф-полей используется файл результатов или КТ, указанный в NameStart.

Отметим разницу между SM\_RES и SM\_CP. В первом случае загружаются только поля Ф-переменных, во втором – *все настройки* задачи, сохраненные в файле результатов: номер шага по времени, значение времени, значения параметров релаксации.

TypeMesh = TM\_Cartes / TM\_Cylind / TM\_UnCartes / TM\_UnCylind

оператор определяет тип используемой сетки:

TM\_Cartes - используется декартова структурная сетка,

TM\_Cylind - используется структурная сетка в цилиндрической системе координат,

TM\_UnCartes - используется неструктурная декартова сетка с локальным дроблением,

TM\_UnCylind - используется неструктурная сетка с локальным дроблением в двумерной цилиндрической системе координат.

IsSteady = **.TRUE.** / **.FALSE.**

флаг решения стационарной (.TRUE.) или нестационарной (.FALSE.) задачи.

IsCycleBCX = **.FALSE.** / **.TRUE.**

IsCycleBCY = **.FALSE.** / **.TRUE.**

IsCycleBCZ = **.FALSE.** / **.TRUE.**

флаг задания по осям X,Y,Z периодических граничных условий.

WayOfSolve = **?** ! WS\_BALEQU / WS\_SIMPLE / WS\_PIMPLE

оператор выбора алгоритма решения уравнений переноса для Ф-переменных:

WS\_BALEQU - решаются только уравнения для S-фазы ("TS" и Ф- переменные типа STRUCT и SPACE ),

WS\_SIMPLE - решаются уравнения для Flow,Struct и Porous-зон, для решения динамических уравнений используется алгоритм SIMPLE,

WS\_PIMPLE - аналог предыдущего, используется алгоритм PIMPLE (комбинация алгоритмов SIMPLE и PISO).

TypeFlowPhase = **FP\_ONEPHASE** ! FP\_ONEPHASE / FP\_TWOPHASE / FP\_MIXTURE

оператор используется для задания типа G-фазы. В текущей версии этот оператор не используется.

TimeSolution = **-1** ! <время решения в минутах> / -1

оператор определяет время выполнения расчета в минутах. При превышении времени расчета он заканчивается с сохранением результатов. Если задано -1, то время расчета неограниченно.

StepOutCP = **0**

оператор устанавливает шаг создания файла рестарта:

- 0 - файл не создается,
- > 0 - выдача через заданное число шагов в нестационарном расчете или через заданное число итераций в стационарном расчете,
- 1 - в стационарном расчете файл рестарта создается в конце расчета.

Замечание. При использовании неструктурной сетки файл рестарта более удобен для возобновления расчета, чем файл результатов.

Следующая группа операторов представляют собой логические флаги, управляющие созданием и наполнением файлов результатов:

А-переменная	По умолчанию	Назначение
IsBinFormat	.TRUE.	вывод файлов А-Компилятора/ Решателя в двоичном (binary) формате;
IsSaveVTKbin	.TRUE.	сохранение VTK-файлов в BIN формате;
IsSaveStlObj	.FALSE.	сохранение объектов-патчей расчетной области в виде stl-файлов. Имя файла - <PreRes>_<ИмяПатча>.stl. Эти файлы можно использовать в ParaView для «оформления» расчетной области;
IsSaveVTKgrid	.FALSE.	сохранение А-компилятором в виде VTK-файлов: - КО сетки : файл <PreRes>_cells.vtk, - граней патчей : файл <PreRes>_obj_faces.vtk;
IsSaveVTKRES	.TRUE.	сохранение Решателем файлов результатов в виде VTK-файлов. Эти файлы используются для работы с постпроцессором ParaView;
IsSaveANESRES	.TRUE.	сохранение Решателем файлов результатов в виде ARS-файлов. Эти файлы используются для работы с постпроцессором Anes;
IsDBGMain	.FALSE.	флаг активизирует создание файла <PreRes>.deb для вывода отладочной информации;
IsPrintMyTable	.FALSE.	печать в файл листинга Решателя таблицы интерполяции для всех Variant-переменных типа myTABLE;
IsParInfo	.FALSE.	создание файлов INFOPAR.PXXX с информацией о декомпозиции расчетной области.

### 3.4 Parall. Декомпозиция расчетной области

При использовании параллельного расчета производится декомпозиция расчетной области на подобласти субдомены. Подробно декомпозиция для структурной и неструктурной сеток описана в документе [3].

Для *структурной сетки* предусмотрены два механизма декомпозиции – автоматическая и ручная. Для указания механизма используется А-оператор

TypeSplit = **TS\_AUTO** ! TS\_AUTO / TS\_MANUAL

При использовании автоматической декомпозиции (TS\_AUTO) разбиение производится исходя из следующих критериев:

- 1) число субдоменов равно числу используемых логических процессоров,
- 2) направления декомпозиции выбираются из условия минимального числа HALO-ячеек.


При использовании ручной декомпозиции (TS\_MANUAL) структурной сетки пользователь задает число субдоменов по оси X, Y, Z с помощью операторов:

NoSubDomX = 1

NoSubDomY = 1

NoSubDomZ = 1

При задании этих параметров необходимо выполнить условие  
 $\text{NoSubDomX} * \text{NoSubDomY} * \text{NoSubDomZ} = \text{<число логических процессоров>}$

 **Замечание:** Операторы ручной декомпозиции можно указать только при редактировании файла проекта в текстовом редакторе, диалоговый препроцессор не обрабатывает эти операторы.

Декомпозиция *неструктурной* сетки всегда проводится автоматически.

### 3.5 Special Data. Прямой интерфейс Решателя

Операторы этой секции используются для «гибкой» настройки параметров Решателя и для настройки параметров его новых компонентов. Общий вид специальных операторов:

R("ИмяПараметра") = 0.0 ! действительное значение

I("ИмяПараметра") = 0 ! целое значение

L("ИмяПараметра") = .TRUE. ! логическое значение

C("ИмяПараметра") = " " ! символьное значение

V("ИмяПараметра") = vec(0.0, 0.0, 0.0) ! действительный вектор

Имя оператора "ИмяПараметра" может быть любое, но не более 64 символов. Значение символьного параметра ограничено 128 символами.

В текущей версии используются операторы, указанные в таблице 3.1.

Таблица 3.1

ИмяПараметра	По умолчанию	Назначение
<b>Общие параметры</b>		
Relax.<ИМЯ>	1	Коэффициент линейной релаксации для свойства с именем <ИМЯ>;
MINLEVEL. <ИмяПатча>	<MaxLevel>	Минимальный уровень дробления в зонах типа FixLevel (<ИмяПатча> – имя стандартного Volume патча);
X0.CYL , Y0.CYL	0.0	Центр цилиндрических координат для внутренних переменных пользователя с именами RCYL, FiCYL;
X0.SPHE , Y0.SPHE, Z0.SPHE	0.0	Центр сферических координат для внутренних переменных пользователя с именами RSPHE, FiSPHE, TetaSPHE;
IsUPCoeffVE	.FALSE.	Если флаг указан, то в дискретных коэффициентах $a_{nb}$ схем модели Патанкара используются

		конвективные значения «против потока» (по умолчанию используются «точные» конвективные потоки);
<b>Общие настройки моделей</b>		
PropConcModel	?	Модель расчета свойств многокомпонентной среды: byCONC/ byWILKE;
IsWILKE_PABS	.False.	Расчет плотности в модели byWILKE по полному давлению ( $P_0 + P_f$ );
Use_LDIS	FALSE	Независимая активация расчета переменной LDIS (расстояния до ближайшей твердой стенки);
CreateDEVVEL	.FALSE.	Для одномерных стабилизированных задач создаются файлы *.dat с полями Ф-переменных для задания граничных условий в 2D и 3D задачах;
ModelRhie_Chow	"Peric"	Выбор модели расчета скорости на грани КО для неструктурной сетки: "Peric" – стандартный алгоритм Рие-Чоу, "Porous" – алгоритм, адаптированный для пористых зон;
ModelCALC_GRAD	"LSM"	Алгоритм расчета градиентов Ф-переменных: "Gauss" – метод Гаусса-Грина, "LSM" – метод наименьших квадратов;
MaxGradIter	10	Максимальное число итераций для расчета градиента Ф-переменной для модели Гаусса;
ModelLSM_WEIGHT	"W_Gauss"	Весовая функция LSM модели расчета градиента: "W_Gauss", "W_Norm", "W_Unit";
IsPOOR_MESH	.T.	Активизация алгоритма коррекции «плохих» ячеек для давления и уравнений движения ;
IsPOOR_MESH_TGS	.F.	Активизация алгоритма коррекции «плохих» ячеек для уравнения энергии TG и TS;
ComprsPF	.F.	Активация модели расчета неструктурных «звуковых» течений;
AlfaPF_CDS	0	Коэффициент доли центрально-разностной схемы для пульсации плотности для модели неструктурных «звуковых» течений;
NoSecondPStep	0	Число вторичных внутренних итераций при решении уравнения для поправки давления для модели неструктурных «звуковых» течений;
SUBPATCH.<Суб-имя>	?	Задание имени патча-родителя для VOLUME суб-патча с именем <Суб-имя> (суб-патч - это стандартный Volume патч с именем <Суб-имя>, патч-родитель - это любой граничный патч, грани которого «передаются» суб-патчу);
IsBodyForcePorous	.T.	Активация модели BodyForce для скоростных источников в пористых зонах;
BodyForce.<ИмяПатча>	.F.	Активация модели BodyForce для скоростных источников для патча;
BodyForce.*	.F.	Активация модели BodyForce для скоростных источников, связанных со всей РО;
IsSrcConc_TERMO	.F.	Активация расчета термодиффузии;
IsSrcConc_BARO	.F.	Активация расчета бародиффузии;

ConstTD	0.25	Постоянная в формуле расчета коэффициента термодиффузии;
CriCourant	1000 / 0.5	Критическое число Куранта для алгоритма PIMPLE;
IsStopOnIterLES	.F.	Активация модели StopOnIterLES окончания SWEEP-итераций для схемы PIMPLE;
IsBodyForcePorous	.F.	Активация BodyForce модели для пористых силовых источников;
BodyForce.<ИмяПатча>	.F.	Активация BodyForce модели для силовых источников патча;
BodyForce.*	.F.	Активация BodyForce модели силовых источников патча для всей РО;
IsIntStressSource	.F.	Активация расчета дополнительных членов напряжений Ньютона;
IsCoupledTGS	.T.	Активация режима совместного решения уравнений для Tg и Ts для сопряженных задач;
<b>Модели турбулентности</b>		
<ИмяПатча>.HR	-	Высота шероховатости для патча <ИмяПатча>;
IsWALLTUR_EXT	.F.	Использование F-точки в модели пристенных функций;
WALLTUR_EXT_VTK	.F.	Создание vtk-файла с F-точками;
trwDBS.UPLUS	1	Модель универсального логарифмического профиля скорости в пристенных функциях: 1- стандартный логарифмический, 2- профиль Сполдинга;
<b>Двухфазная VOF-модель</b>		
NoSmoVOF	2	Число циклов размазывания свойств и источников;
IsSMO_VOFPROP	.TRUE.	Использование размазывания для расчета свойств (если .TRUE, то для расчета свойств используются точные значения VOF);
IsSurfVOF	.F.	Создание vtk-файла с границей раздела в виде 3D поверхности;
CriCourant	0.5	Критическое число Куранта для VOF модели;
SweepVOFmodelID	OneSweep	Алгоритм расчета VOF на итерациях: OneSweep - только на первой итерации шага по времени, Explicit - на всех итерациях (но по явной схеме);
IsBodyForceVOF	.TRUE.	Использование BodyForce модели расчета силовых источников для неструктурной сетки;
<Имя патча>.ContactAngle	90	Задание краевого угла смачивания для Wall-патча с именем <Имя патча>;
VofForFMassID	LIN	Выбор модели расчета $\phi$ на гранях ячейки: LIN - линейная интерполяция точной VOF, FracTRU - точные значения корректор-шага $\phi_{t,k}$ ;
<b>Двухфазная LDP-модель</b>		
IsUseLDPM	.F.	Активация LDPM модели;
TypeInLet	?	Выбор алгоритма задания начальных условий для частиц: "UserSET" / "VolFrac";

PatchInlet	?	Имя граничного патча, с которого запускаются частицы;
NoParticles	0	Число частиц для модели LDPM "UserSET";
PartFile	?	Имя файла с параметрами частиц;
NoPartInGroup	1	Число частиц в группе (модель "VolFrac");
PStepFaces	1	Шаг отбора групп из граней патча для варианта "VolFrac";
Coef_Rosin	4.52	Постоянная Росина для распределения частиц по диаметрам;
VolFracIN	0.0	Объемная доля частиц на входе;
DPartMEAN	0.0	Средний диаметр частиц;
DensPartIN	0.0	Плотность частиц;
SigmaG	0.0	Коэффициент поверхностного натяжения;
We_ReBound_MIN	-1.0	Минимальное число WE для отражения;
We_ReBound_MAX	1.0e10	Максимальное число WE для отражения;
We_SPLASH	8.0e10	Граница зоны поглщения по числу WE;
dpmMaxTime	1.0e5	Максимальное время жизни частицы;
<Патч>.REFL_N <Патч>.REFL_T	1.0	Коэффициенты отражения частицы по нормали и касательной к поверхности патча;
IsPartTURB	.F.	Активация модели турбулентной дисперсии частиц;
isPART_DBG	.F.	Вывод в deb-файл полной информации о движении всех частиц;
isNOFACE_DBG	.F.	Вывод в deb-файл информации о частице без выходной грани;
<b>S2S модель Излучения</b>		
FViewFile	?	Имя файла с угловыми коэффициентами;
<ИмяРадЗоны>.epszz	1	Степень черноты для радиационной зоны;
<b>Настройки линейных солверов</b>		
P2P.<Ф-имя>	.F.	Использование P2P LES-солвера для Ф-переменной с именем <Ф-имя>;
SIPSOL.Alfa	0.90	Параметр Alfa для SIPSOL LES солвера;
SPKT.LFIL	5	LES солверы SparsKit2: параметр LFIL;
SPKT.PRECOND	1	LES солверы SparsKit2: индекс прекондиционера;
<b>Вывод результатов</b>		
MarkerRUN	" "	Маркер для пометки консольного расчета; текст выводится на консоль Решателя;
CLUSTER	.FALSE.	Значение всех скалярных переменных пользователя выводятся на консоль;
IsCurTimeRun	.FALSE.	В стоку шага по времени на консоль выводится текущее время расчета в минутах (удобно для работ на кластере) ;
ARS.OUTCELL	.FALSE.	Вывод в неструктурный ARS-файл полей в центрах ячеек (по умолчанию выводятся в вершинах ячеек);
VTK.OUTCELL	.FALSE.	Вывод в VTK-файл полей в центрах ячеек;
IsPrintMinMaxTS	.F.	Печать в листинг максимальных и минимальных температур всех S-зон;
NoOutTime	1	Шаг выдачи информации о шаге по времени на консоль (по умолчанию выводится информация

		о каждом шаге);
<b>Параметры для отладки Решателя</b>		
IsSweepProf	.F.	Активация профилирования кода Решателя с выводом суммарной информации в Deb-файл;
LesProf	0	номер ВНУТРЕННЕЙ итерации, на которой проводится профилирование алгоритмов LES-солвера;
ValidEps	1.e-3	пакетная верификация: точность сравнения
Valid.PHI	?	пакетная верификация: эталонное значение для сравнения $\Phi$ -переменной с именем PHI;
PrintMAP	.FALSE.	распечатка <i>структурного</i> массива MapCV: используется только для двумерных полей
PrintPHI	.FALSE.	распечатка двумерных полей (структурных и неструктурных) в deb-листинг;
PrintUSER	.FALSE.	распечатка двумерных сеточных полей пользователя (структурных и неструктурных) в deb-листинг;
PrintMassFlux	.FALSE.	распечатка двумерных <i>неструктурных</i> полей массовых потоков на гранях ячеек в deb-листинг;
PrintMAXRES	.FALSE.	распечатка дискретных коэффициентов в ячейках с максимальной невязкой;
PrintCFVE	.FALSE.	распечатка коэффициентов дискретных уравнений <i>неструктурной</i> сетки в виде двумерных таблиц;
PrintCFVE.PHI	"*"	Имя $\Phi$ -переменной для оператора PrintCFVE, если указано "*", то коэффициенты выводятся для всех $\Phi$ -переменных;
PrintCFVE.Sweep	0	Номер SWEEP итерации, на которой активируется PrintCFVE, если задан 0, то распечатка производится на итерации MaxSweep;
MaxTimeStep	-1	Максимальное число шагов по времени расчета (по умолчанию используется сетка по времени);
IsP_BN_EXTRAPOLATION	.FALSE.	По умолчанию при расчете градиента PF на непроницаемых стенках используется условие $dP/dn = 0$ ; если данный флаг = .True., то используется экстраполяция;
NoPISOiter	1	Число PISO циклов в алгоритме PIMPLE;
INFO.IX, INFO.IY, INFO.IZ INFO.CellID	0	Индексы ячейки, для которой в deb-файл выводится полная информация о дискретных уравнениях и сходимости; одновременно в файл с именем <Префикс>_cell_<индекс ячейки>.dat выводятся значения $\Phi$ -переменных в ячейке на каждой итерации;
INFO.PHI	*	Имя $\Phi$ -переменной, для которой производится вывод информации (если "*", то информация выводится для всех $\Phi$ -переменных);
Info.PHIGRAD	?	Имя $\Phi$ -переменной, для которой выдается информация о расчете градиента;
INFO.LESIter	.F.	Если флаг включен, то в deb-файл на итерации выдается информация для каждой $\Phi$ -



		переменной о числе LES-итераций, начальной и конечной LES-невязки;
INFO.TimeStep	0	Номер шага по времени, для которого выдает Info-информация;
IsDebugSweepInfo	.F.	Выдача информации о невязках всех Ф-переменных на каждой итерации в файл <Префикс>_sweep.dat ;
StepOutSweepRES	-1	Шаг создания файлов результатов на итерации при решении стационарной задачи (в результатах Time - это номер итерации);
BegOutSweepRES	1	Начало создания файлов результатов на итерации при решении стационарной задачи;
<b>CBL модель слоев ячеек сглаживания</b>		
CBLPatch	"<Имя патча>"	Имя патча, для которого создаются CBL ячейки;
CBLsize. <ИмяПатча>	?	Высота слоев CBL ячеек для данного патча;
CBLNoSubCell.<ИмяПатча>	1	Число слоев CBL ячеек;
CBLqMesh. <ИмяПатча>	1.0	Коэффициент логарифмичности CBL слоев

#### 4. Секции $\Phi$ -переменных и переменных пользователя

Код Anes предназначен для расчета полей зависимых переменных, описывающих сплошную среду. Такие переменные в коде называются  $\Phi$ -переменными [3].

Для описания  $\Phi$ -переменных, связанных с G-фазой (сплошной средой) используются уравнения переноса следующего вида:

$$\frac{\partial(\varphi\rho_g\Phi)}{\partial\tau} + \text{div}(\varphi\rho_g\mathbf{U}_g\Phi - \varphi\Gamma_\Phi\nabla\Phi) = S_\Phi^{(1)} + \varphi S_\Phi \quad (4.1)$$

здесь

- $\rho_g$  - плотность G-фазы (AV-переменная),
- $\varphi$  - пористость,
- $\Gamma_\Phi$  - «коэффициент диффузии»,
- $S_\Phi^{(1)}$  - объемные источниковые члены, связанные со спецификой  $\Phi$ -переменной и автоматически рассчитываемые Решателем (*внутренние* источники),
- $S_\Phi$  - источниковый член пользователя, отнесенный к единице объема, занятой G-фазой,
- $\mathbf{U}_g$  - вектор скорости G-фазы.

Для коэффициента диффузии в коде предусмотрены два варианта представления - через аналог «числа Прандтля» и в виде непосредственного коэффициента «диффузии»:

$$\Gamma_\Phi = \rho_g \left( \frac{\nu_g}{\sigma_\Phi} + \frac{\nu_t}{\sigma_{\Phi,t}} \right) \quad \text{или} \quad \Gamma_\Phi = \rho_g \left( D_\Phi + \frac{\nu_t}{\sigma_{\Phi,t}} \right) \quad (4.2)$$

Для  $\Phi$ -переменной "Tg" используется другая форма этих соотношений

$$\Gamma_{Tg} = \rho_g c_{pg} \left( \frac{\nu_g}{Pr} + \frac{\nu_t}{Pr_t} \right) \quad \text{или} \quad \Gamma_{Tg} = \left( \lambda_g + \frac{\rho_g \nu_t c_{p,g}}{Pr_t} \right)$$

где:

- $\nu_g$  - кинематическая вязкость G-фазы,
- $\nu_t$  - кинематический коэффициент турбулентной вязкости, определяемый моделью турбулентности,
- $\sigma_\Phi$  ,  $\sigma_{\Phi,t}$  - молекулярное и турбулентное «число Прандтля» для  $\Phi$ -переменной,
- $Pr$  ,  $Pr_t$
- $D_\Phi$  - молекулярный коэффициент «диффузии» для  $\Phi$ -переменной.

Уравнение сохранения массы G-фазы (уравнение неразрывности) в коде рассматривается как уравнение для давления  $P_f$  (точнее, давление рассматривается как  $\Phi$ -переменная для уравнения неразрывности, хотя формально давление отсутствует в этом уравнении). Уравнение неразрывности имеет вид

$$\frac{\partial(\varphi\rho_g)}{\partial\tau} + \text{div}(\varphi\rho_g\mathbf{U}_g) = M_p^{(1)} + \varphi M_p \quad (4.3)$$

Для описания  $\Phi$ -переменных, связанных с S-фазой (неподвижные конструкции), используются уравнения вида:

$$(1-\varphi)\rho_s c_{ps} \frac{\partial T_s}{\partial\tau} = \text{div}((1-\varphi)\underline{\lambda}_s \otimes \nabla T_s) + S_{Ts}^{(1)} + (1-\varphi)S_{Ts} , \quad (4.4)$$

где

$\underline{\lambda}_s$  - ортотропный тензор теплопроводности, диагональные компоненты которого равны  $\lambda_{sx}, \lambda_{sy}, \lambda_{sz}$ , для декартовой системы координат

$$\underline{\lambda} \otimes \nabla T_s = \lambda_{sx} \frac{\partial T_s}{\partial x} + \lambda_{sy} \frac{\partial T_s}{\partial y} + \lambda_{sz} \frac{\partial T_s}{\partial z},$$

$\rho_s$  - плотность S-фазы,

$c_{ps}$  - теплоемкость S-фазы,

$S_{Ts}$  - объемный источник пользователя, отнесенный к единице объема S-фазы,

$S_{Ts}^{(l)}$  - внутренние источники, рассчитываемые автоматически Решателем.

В коде Anes предусмотрен набор predefined  $\Phi$ -переменных, название которых фиксировано. В текущей версии – эти переменные приведены в таблице 4.1:

Таблица 4.1

Имя	Фаза	Назначение
PF	G	относительное давление в G-фазе,
UgX	G	X-компонента скорости G-фазы,
UgY	G	Y-компонента скорости G-фазы,
UgZ	G	Z-компонента скорости G-фазы,
Tg	G	температура G-фазы
Ts	S	температура S-фазы
kTUR	G	турбулентная энергия k
epsTUR	G	турбулентная диссипация $\varepsilon$
omTUR	G	турбулентная диссипация $\omega$
C1	G	массовая доля первого компонента
...		...
C8	G	массовая доля восьмого компонента
wDIS	G	Вспомогательная переменная для расчета $L_{dis}$

Наряду с predefined  $\Phi$ -переменными пользователь может использовать свои собственные  $\Phi$ -переменные, связанные либо с G-фазой, либо с S-фазой, либо с пространством. В последнем случае уравнение переноса имеет следующий вид:

$$\frac{\partial(\Phi)}{\partial \tau} = \text{div}(\Gamma_{\Phi} \nabla \Phi) + S_{\Phi} \quad (4.5)$$

Главные особенности таких  $\Phi$ -переменных пользователя:

- 1) они могут иметь произвольное имя,
- 2) уравнения переноса (4.1) – (4.4) для таких переменных являются основой «конструктора» - пользователь может включать и выключать отдельные члены уравнений.

#### 4.1 PHI Variables. Настройка $\Phi$ -переменных

Эта секция используется для описания predefined  $\Phi$ -переменных и  $\Phi$ -переменных пользователя.

### 4.1.1 Общие операторы

Первым оператором (или операторами) секции должен быть:

Used(Ф-имя1, Ф-имя2, ...),

в котором перечисляются используемые Ф-переменные. Далее следуют следующие операторы в произвольном порядке.

Solved("Ф-имя") = **.TRUE.** ! .TRUE. / .FALSE.

оператор активизирует решения уравнения переноса для Ф-переменной. Пользователь может не решать уравнение для Ф-переменной, а ее значения может определить с помощью оператора инициализации.

Restored("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

оператор определяет необходимость при рестарте восстанавливать поле Ф-переменной из файла рестарта.

MaxIter("Ф-имя ") = **20**

оператор определяет максимальное число итераций в линейном LES-солвере.

TypeEndSweep("Ф-имя ") = \* ! TES\_MAXRES / TES\_TOTRES / TES\_MAXCOR

Eps("ИмяФ") = \*

оператор определяет критерий и точность окончания внешних SWEEP-итераций для данной Ф-переменной. По умолчанию используется значения TypeEndSweep и Eps из секции [Solver].

MinValue("Ф-имя ") = **-1.E30** ! Для kTUR и epsTUR =  $10^{-10}$

MaxValue("Ф-имя ") = **1.E30**

операторы определяют границы изменения Ф-переменной в процессе решения.

Relax("Ф-имя ") = **1.0**

оператор определяет коэффициент релаксации для Ф-переменной. По умолчанию для всех переменных, кроме скоростей и давления, релаксация не используется. Коэффициенты релаксации для давления и скоростей зависят от типа сетки. Для структурной сетки:

Relax("PF") = 0.8; Relax("UGX") = Relax("UGY") = Relax("UGZ") = 0.5

для неструктурной сетки:

Relax("PF") = 0.2; Relax("UGX") = Relax("UGY") = Relax("UGZ") = 0.8

Если значение коэффициента релаксации положительно, то задается коэффициент нижней линейной релаксации. Если значение коэффициента отрицательное, то оператор задает значение фиктивного шага по времени  $\Delta t_r$  для FTS-релаксации!

Init("ИмяФ ") = **0** ! V-переменная

оператор определяет алгоритм вычисления начального распределения Ф-переменной. Для этого используется Variant-переменная.

LESSolver("ИмяФ ") = \* ! Имя LES солвера

оператор определяет LES-солвер для данной Ф-переменной. Если указана \*, то используется LES солвер по умолчанию,

FVEScheme("ИмяФ ") = \* / FVES\_....

оператор определяет численную схему для данной Ф-переменной. Если указана \*, то используется численная схема по умолчанию (см. п. 5.1).

### 4.1.2 Операторы для предопределенных Ф-переменных

Большинство нижеописанных параметров используются для настройки уравнений для произвольных Ф-переменных пользователя. Но несколько операторов используется и для предопределенных переменных.

По умолчанию в качестве молекулярного коэффициента «диффузии» для "TG" используется коэффициент теплопроводности – свойство с именем «Cond». Для использования вместо теплопроводности числа Прандтля, необходимо указать операторы

```
TypeDiff("TG") = ND_PRAN
NamDiff("TG") = "ИмяСвойства"
```

Если второй оператор не указан, то в качестве имени свойства используется имя "Pran".

По умолчанию в качестве молекулярного коэффициента «диффузии» для "TS" используется свойство «Cond», а сам процесс переноса считается изотропным. Для задания не-изотропной теплопроводности необходимо использовать операторы

```
NamDiffX("TS") = "ИмяСвойстваX"
NamDiffY("TS") = "ИмяСвойстваY"
NamDiffZ("TS") = "ИмяСвойстваZ"
```

В особо «экзотических» случаях пользователь может отключить ряд членов уравнений переноса для предопределенных уравнений

```
LDifX("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
LDifY("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
LDifZ("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
LTrans("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
LConv("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
LTurb("Ф-имя ") = .TRUE. !.TRUE. /.FALSE.
```

Для задания коэффициентов «диффузии» по умолчанию для предопределенных переменных используются следующие имена свойств:

```
PF, kTUR, epsTUR = ""
UGX, UGY, ULZ   = "Visc",
TG, TS          = "Cond",
C1 .. C8        = "DDIF.C1" ..."DDIF.C8".
Если указан параметр TypeDiff("Ф-имя ") = ND_PRAN, то
TG              = "Pran",
C1 .. C8        = "SC.C1" ..."SC.C8".
```

### Уравнения для концентраций

При работе с концентрациями также существует два способа задания «диффузии». По умолчанию используется оператор (XX = C1 ... C8):

```
TypeDiff("XX") = ND_PROP
```

и в этом случае коэффициент диффузии рассчитывается по соотношению

$$\Gamma_{ck} = \rho_g \left( D_k + \frac{v_t}{\sigma_{ck,t}} \right)$$

а значение коэффициента диффузии задается свойством с именем "DDIF.XX".

При использовании второго варианта вместо коэффициента диффузии используется молекулярное диффузионное число Прандтля  $Sc_k$  (число Шмидта)

$$\Gamma_{ck} = \rho_g \left( \frac{v_g}{Sc_{ck}} + \frac{v_t}{\sigma_{ck,t}} \right), \quad Sc_k = \frac{v_g}{D_k},$$

которое задается свойством с именем "SC.XX".

### 4.1.3 Операторы для произвольных Ф-переменных пользователя

Для настройки уравнений для произвольных Ф-переменных пользователя используется тот же подход. Форма коэффициента переноса (4.2) задается оператором:

TypeDiff("Ф-имя ") = ND\_PROP ! ND\_PRAN / ND\_PROP

Операторы:

NamDiff("Ф-имя ") = "<ИмяСвойства>"

имя свойства для изотропного коэффициента «диффузии» или имя «числа Прандт-ля», если TypeDiff = ND\_PRAN.

NamDiffX("Ф-имя ") = "ИмяСвойстваX"

NamDiffY("Ф-имя ") = "ИмяСвойстваY"

NamDiffZ("Ф-имя ") = "ИмяСвойстваZ"

имена свойств для неизотропных коэффициентов «диффузии».

LDifX("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

LDifY("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

LDifZ("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

LTrans("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

LConv("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

LTurb("Ф-имя ") = **.TRUE.** ! .TRUE. / .FALSE.

включение/выключение членов уравнения переноса.

Phase("Ф-имя ") = **PD\_FLOW** ! PD\_FLOW / PD\_STRUCT / PD\_SPACE

привязка к фазе и выбор типа уравнения переноса.

WallFixNull("Ф-имя ") = **.FALSE.** ! .TRUE. / .FALSE.

разрешение автоматического задания граничных условий первого рода с нулевым значением на граничных патчах с флагом IsWALL (как для компонент вектора скорости).

Для задания коэффициентов «диффузии» используются следующие имена свойств по умолчанию:

<все > = "**DDIF.< Ф-имя >**"

Если указан параметр

TypeDiff("Ф-имя ") = ND\_PRAN

то по умолчанию используются имена

<все > = "**Pran.<имя Ф>**"

## 4.2 User Variables. Настройка сеточных User-переменных пользователя

Сеточные User-переменные – это переменные, связанные с сеткой КО. Их вычисление (и использование) полностью определяется пользователем. Главная их особенность – их можно использовать во всех вычислениях пользователя наряду с Ф-переменными. Кроме того, они сохраняются в файлах результатов вместе с Ф - переменными.

Первыми операторами должны быть операторы описания этих переменных:

Used(ИмяU1,ИмяU2, ...)

Оператор

Saved("ИмяU") = **.TRUE.** ! .TRUE. / .FALSE

определяет нужно ли сохранять User-переменную в файле результатов.

Оператор

Calc("ИмяU") = V-переменная

задает алгоритм вычисления переменной. Для этого всегда используется V-переменная.

Для указания мест в коде Решателе (см. пункт 2.7), где необходимо рассчитывать значения User-переменной используется оператор

EntryCalc("ИмяU ") = **setof(CU\_Report)** ! setof(Point1,Point2, ....)

возможные точки расчета (PointX):

CU\_Init - при инициализации Решателя,  
 CU\_BeforeSweep - перед SWEEP-итерации,  
 CU\_AterSweep - после SWEEP-итерации,  
 CU\_BeforeTime - перед шагом по времени,  
 CU\_AfterTime - после шага по времени,  
 CU\_SAVE - перед записью файлов результатов,  
 CU\_Report - в процессе работы подпрограммы Report,  
 CU\_FVE - после выполнения линейного солвера для Ф-переменной,  
 заданной оператором  
 FVE\_PHI("ИмяУ ") = "ИмяФ"  
 Эта опция используется для отладки кода Решателя.

Для задания подмножества КО, в которых нужно проводить вычисления, используется оператор

```
Phase("ИмяУ") = LT_SPACE ! LT_FLOW / LT_STRUCT / LT_SPACE / LT_PATCH
```

При задании LT\_PATCH вычисления производятся только для КО, связанных с патчем. Имя патча в этом случае должно быть задано оператором

```
PatchName("ИмяУ") = "ИмяПатча"
```

Приведем пример использования User-переменной – расчет точного решения задачи и ее сравнение с рассчитанным полем:

#### [Macro Variables]

```
macro(ROUT,RIN,CENTRE, T00, Q11)
ROUT = 0.6
RIN = 0.15
CENTRE = 1.1*ROUT
T00 = 1
Q11 = 2
.....
```

#### [USER Variables]

```
Used(RZZ,TS_TH, TS$TH)
calc("RZZ") =MyForm("SQRT((XP-CENTRE)**2+(YP-CENTRE)**2)")
calc("TS_TH") =MyForm("T00 + Q11*ROUT*log(RZZ/RIN)")
calc("TS$TH") =MyForm("TS/TS_TH")
```

В этом примере рассчитывается точное решение "TS\_TH" для температуры Ts. Задача использует декартовую систему координат для моделирования осесимметричной задачи. Для вычисления точного решения необходим радиус полярной системы в используемой декартовой системе – переменная RZZ. Для записи в файл результатов точности расчета используется переменная TS\$TH.

Отметим некоторые особенности:

1. В формулах myFORM можно смело использовать как Ф-переменные, так и макро-переменные и User-переменные.
2. В данном примере не задано место вычисления в коде Решателя и фаза, поэтому используются значения по умолчанию (при подготовке окончательного отчета и во всей PO).

### 4.2.1 Внутренние сеточные User-переменные.

Если для User-переменной пользователя *не задан алгоритм расчета* (AИL-оператор Calc), то такая переменная называется внутренней User-переменной Решателя. Значения этого поля пользователя рассчитываются автоматически внутри Решателя, если эта переменная описана в операторе Used. Имена внутренних переменных и точки их расчета приведены в таблице 4.2.

Таблица 4.2

Имя	Точка расчета	Назначение
RCYL	CU_Init	радиус «цилиндрических» координат,
FiCYL	CU_Init	угол «цилиндрических» координат,
RSPHE	CU_Init	радиус «сферических» координат,
TetaSPHE	CU_Init	Угол R-Z (широта) «сферических» координат,
FiSPHE	CU_Init	Угол X-Y (долгота) «сферических» координат,
_AlfaGS	CU_Report	коэффициент теплопередачи в пористых зонах,
_KX, _KY, _KZ	CU_Report	коэффициенты проницаемости в пористых зонах,
SumConcTru	-	сумма концентраций до их коррекции на 1,
iUseWallFun	-	Флаг использования пристенных функций (0 или 1),
iLenTur	-	масштаб турбулентных пульсаций,
UgR , UgTeta	-	Компоненты вектора скорости в псевдополярной СК

«Цилиндрические» координаты рассчитываются для декартовой системы координат (для структурной и неструктурной сеток) по соотношениям:

$$RCYL = \sqrt{(x - x_0)^2 + (y - y_0)^2},$$

$$FiCYL = a \tan\left(\frac{y - y_0}{x - x_0}\right)$$

Угловая координата измеряется в радианах и изменяется в пределах  $0 \dots 2\pi$ . Координаты центра цилиндрической системы  $(x_0, y_0)$  задаются операторами

R("X0.CYL") = x0

R("Y0.CYL") = y0

секции [Special Data]. Если операторы не заданы, то координаты центра равны нулю.

«Сферические» координаты рассчитываются по соотношениям

$$RSPHE = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2},$$

$$FiSPHE = a \tan\left(\frac{y - y_0}{x - x_0}\right), \quad TetaSPHE = a \cos\left(\frac{z}{RSPHE}\right)$$

Координаты центра сферической системы  $(x_0, y_0, z_0)$  задаются операторами

R("X0.SPHE") = x0

R("Y0.SPHE") = y0

R("Z0.SPHE") = z0

секции [Special Data]. Если операторы не заданы, то координаты центра равны нулю.



При решении осесимметричных задач в *декартовой* СК возникает потребность в расчета «полярных» скоростей ( $U_r$  и  $U_\theta$ ). Типичный пример - LES моделирование в круглой трубе с помощью неструктурных сеток. Для вывода в файл результатов этих компонент скорости используются внутренние переменные с именами "UgR" "UgTeta". Для их вычисления необходимо задать «центр псевдополярной» СК. Для этого используется оператор секции [Special Data]:

```
V("PolarCenter") = vec(xC,yC,0)
```

Здесь (xC,yC) - центр полярной СК.

### 4.3 User Scalars. Настройка скалярных User-переменных пользователя

Операторы настройки скалярных переменных пользователя аналогичны операторам настройки сеточных User-переменных:

```
Used(ИмяUS1, ИмяUS 2, ...)
```

```
Restored("ИмяUS ") = .TRUE. ! .TRUE. / .FALSE. – флаг восстановления при рестарте
```

```
EntryCalc("ИмяUS ") = setof(CU_Report) ! setof(CU_Init , CU_BeforeSweep , CU_AfterSweep ,  
! CU_BeforeTime / CU_AfterTime , CU_Save , CU_Report )
```

```
Calc("ИмяUS ") = 0 ! Variant
```

#### 4.3.1 Внутренние скалярные User-переменные.

Если для скалярной User-переменной не задан алгоритм расчета (оператор Calc), то такая переменная называется внутренней скалярной переменной Решателя. Значение этой переменной пользователя рассчитывается автоматически внутри Решателя, если эта переменная описана в операторе Used. Имена внутренних скалярных переменных и точки их расчета приведены в таблице 4.3.

Таблица 4.3

Имя	Точка расчета	Назначение
_APorous.XX	CU_Report	межфазная поверхность для пористой зоны с индексом XX,
_QPorous.XX	CU_Report	полный тепловой поток между фазами для пористой зоны с индексом XX.

### 4.4 Vector Variables. Описание векторных полей

Для анализа в постпроцессоре Anes векторных полей используются операторы данной секции. Векторное поле – это вектор, компонентами которого являются либо Ф-переменные, либо сеточные User-переменные пользователя. Для «создания» вектора используется оператор:

```
Vector("Имя вектора",ИмяX, ИмяY, ИмяZ)
```

здесь ИмяX, ИмяY, ИмяZ – либо имя Ф/User - переменной, либо "None" или "0", если компонента вектора нет.

При решении задач гидродинамики можно не создавать векторное поле для скорости G-фазы. В этом случае Решатель сам создаст вектор с именем "G-Velocity".

## 5. Секции настройки параметров Решателя

Секция описывает операторы управления итерационным процессом, операторы настройки линейного солвера и операторы настройки Решателя для работы с программой ДиалогСолвера.

### 5.1 Solver. Настройка параметров итерационного процесса

Оператор

TypeEndSweep = **TES\_TOTRES** ! TES\_MAXRES / TES\_TOTRES / TES\_MAXCOR  
определяет критерий окончания SWEEP-итераций, используемый по умолчанию для всех Ф-переменных:  
TES\_TOTRES - по средней невязке уравнения,  
TES\_MAXRES - по максимальной невязке уравнения,  
TES\_MAXCOR - по максимальной коррекции.  
Для отдельной Ф-переменной можно указать свой критерий в секции [PHI Variables].

Eps = **1.e-5**

оператор задает точность решения по умолчанию для всех Ф-переменных. Для отдельной Ф-переменной можно указать свое значение в секции [PHI Variables].

MaxSweep = **100**

оператор определяет максимальное число внешних SWEEP-итераций.

MaxBeforeSweep = **1000**

оператор определяет максимальное число SWEEP-итераций для решения Ф-переменных до основного итерационного цикла.

Операторы

FVEScheme = **FVES\_POWER** ! FVES\_UPWIND / FVES\_POWER / FVES\_SUPERBEE  
! FVES\_QUICK / FVES\_MISCL / FVES\_GAMMA / FVES\_BoundCD/  
FVES\_TrueCD

TemporalScheme = **TTS\_IMPLICIT** ! TTS\_IMPLISIT / TTS\_3LEVEL

RhoFaceModel = **RF\_LINEAR** ! RF\_UPWIND / RF\_LINEAR

используются для выбора схемы *по умолчанию* аппроксимации конвективного потока и нестационарного члена [3]:

FVES\_UPWIND - схема против потока,  
FVES\_POWER - степенная схема,  
FVES\_QUICK - схема против потока второго порядка,  
FVES\_SUPERBEE ,  
FVES\_MUSCL,  
FVES\_GAMMA - TVD схема второго порядка,  
FVES\_TrueCD - центрально-разностная схема,  
FVES\_BoundCD - «смесь» FVES\_GAMMA и FVES\_TrueCD,

TTS\_IMPLICIT - полностью неявная схема первого порядка для нестационарного члена,

TTS\_3LEVEL - трехуровневая схема 2 порядка для нестационарного члена;

и выбора способа вычисления плотности на грани КО:

RF\_LINEAR - линейная аппроксимация,  
RF\_UPWIND - аппроксимация против потока.

Оператор

SpeedRES = **0.5**

используется для управления внутренними итерациями линейных солверов SparsKit2. Оператор определяет коэффициент уменьшения начальной средней невязки в линейном солвере, после которой внутренние итерации прекращаются.

XpfZERO = 0.0

YpfZERO = 0.0

ZpfZERO = 0.0

операторы определяют точку расчетной области, в которой давление  $P_f$  полагается равным нулю. Масштабирование давления производится только в случае, если ни в одном из граничных условий давление не задано явно.

#### Оператор

LESSolver = KIVA

задает имя используемого линейного солвера по умолчанию.

В текущей версии можно использовать следующие LES солверы:

USER	- внешний солвер пользователя ,
KIVA	- Conjugate Residual солвер кода Kiva,
KIVA_ILU	- Kiva солвер с ILU-прекондиционером SparsKit,
ATDMA	- продольно-поперечная прогонка,
SIP_PHO	- вариант Stone-солвера,
P.SIPSOL	- вариант Stone-солвера из книги Перика,
P.CGS	- Conjugate Gradient солвер из книги Перика,
SPKT.CG	- Conjugate Gradient солвер пакета SparsKit2,
SPKT.BCGSTAB	- Bi-Conjugate Gradient Stabilized солвер пакета SparsKit2,
SPKT.FOM	- Full Orthogonalization Method пакета SparsKit2,
SPKT.GMRES	- Generalized Minimum Residual солвер пакета SparsKit2,
SPKT.FGMRES	- Flexible version of GMRES солвер пакета SparsKit2.
HYPRE	- параллельный солвер Lawrence Livermore National Laboratory.

Ограничения солверов приведены в таблице 4.1. Подробности использования солвера HYPRE приведены в разделе 9.2.1 документа [5].

Таблица 5.1

Солвер	Структурная сетка	Неструктурная сетка	Параллельность	Алгоритм	Периодические ГУ
KIVA	да	да	да	любой	да
KIVA_ILU	да	да	да	любой	да
ATDMA	да	-	-	любой	да
SIP_PHO	да	-	-	любой	-
P.SIPSOL	да	-	-	любой	-
P.CGS	да	-	-	BALEQU	да
SPKT.CG	да	да	-	BALEQU	да
SPKT.BCGSTAB	да	да	-	любой	да
SPKT.FOM	да	да	-	любой	да
SPKT.GMRES	да	да	-	любой	да
SPKT.FGMRES	да	да	-	любой	да
HYPRE	да	да	только	любой	да

Для каждой отдельной Ф-переменной можно задать свой индивидуальный LES-солвер. Для этого используется оператор LESSolver секции [PHI Variables].

## 5.2 Control. Настройка диалога Решателя

Операторы этой секции используются для указания информации, передаваемой Решателем программе ДиалогСолвера. Кроме того эта информация сохраняется в файлах результатов расчета, что позволяет проанализировать ее после расчета.

Операторы

TypeResWin = **TRW\_TOTRES** ! TRW\_MAXRES / TRW\_TOTRES / TRW\_MAXCOR  
TypeMonWin = **TMW\_MON** ! TMW\_MON / TMW\_MAXPHI / TMW\_MINPHI

определяют информацию, выводимую в окно «Сходимости» и окно «Мониторинга»:

TRW\_MAXRES - максимальную невязку,  
TRW\_TOTRES - среднюю невязку,  
TRW\_MAXCOR - максимальную коррекцию,  
TMW\_MON - значение Ф-переменной в точке мониторинга,  
TMW\_MAXPHI - максимальное значение Ф-переменной в РО,  
TMW\_MINPHI - минимальное значение Ф-переменной в РО.

Операторы

MonX = 0.0  
MonY = 0.0  
MonZ = 0.0

задают координаты точки мониторинга.

Оператор

TimeOutSweep = **2.0**

задает шаг выдачи информации на консоль Решателя (в сек).

Оператор

NoDialogSweep = **1** ! шаг передачи данных в aDialog

задает шаг передачи информации программе ДиалогСолвера.

## 6. Секции для описания геометрии расчетной области

### 6.1 Предварительная информация

Подробная информация о геометрии расчетной области и используемых методах дискретизации (построение сеток контрольных объемов) приведена в документах [3,5]. Ниже приведены краткие выдержки из этой информации.

Код Apes позволяет решать стационарные или нестационарные задачи в двух (2D)- или трехмерной (3D) постановке. В коде используются четыре типа сеток контрольных объемов (КО или ячеек):

- 1) структурные сетки КО в декартовой системе координат ( $x = x_c, y = y_c, z = z_c$ ),
- 2) структурные сетки КО в цилиндрической системе координат ( $x = r, y = \theta, z = z_c$ ),
- 3) неструктурные сетки с локальным дроблением в декартовой системе координат ( $x = x_c, y = y_c, z = z_c$ ),
- 4) неструктурные сетки с локальным дроблением в *двумерной* цилиндрической системе координат ( $x = r, z = z_c$ ).

Здесь  $(x, y, z)$  - обобщенные координаты,  $(x_c, y_c, z_c)$  - декартовы координаты,  $(r, \theta, z)$  – цилиндрические координаты.

Область пространства, в которой кодом Apes рассчитываются поля зависимых  $\Phi$ -переменных, называется *расчетной областью* (РО). Внутри Решателя РО – это структурный или неструктурный набор КО.

Процесс построения РО для структурной и неструктурной сеток в коде производится в три этапа:

- 1) на первом этапе создается *базовая расчетная область* (БРО), в которой строится *структурная* сетка КО;
- 2) на втором этапе в БРО помещаются геометрические объекты – патчи, которые и формируют РО;
- 3) на третьем этапе происходит модификация структурной сетки БРО (по разному для структурной и неструктурной сеток).

Напомним, что выбор типа используемых сеток и системы координат определяется оператором секции [Main]:

```
TypeMesh = TM_Cartes / TM_Cylind / TM_UnCartes // TM_UnCylind
```

#### 6.1.1 Базовая расчетная область

БРО *всегда* представляет собой 3D параллелепипед (в обобщенных координатах), содержащий внутри РО (рисунки 6.1, 6.2). Расчетная область может быть произвольной, но обязательно «вписанной» в БРО.

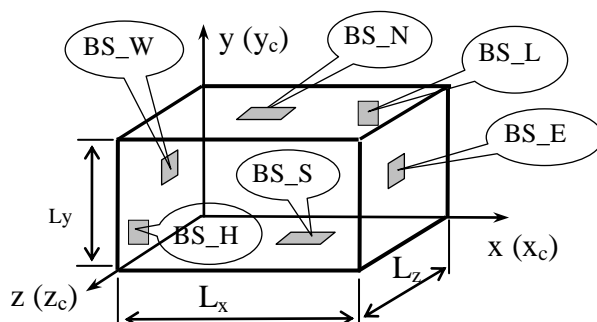


Рисунок 6.1 - Базовая расчетная область в декартовой системе координат

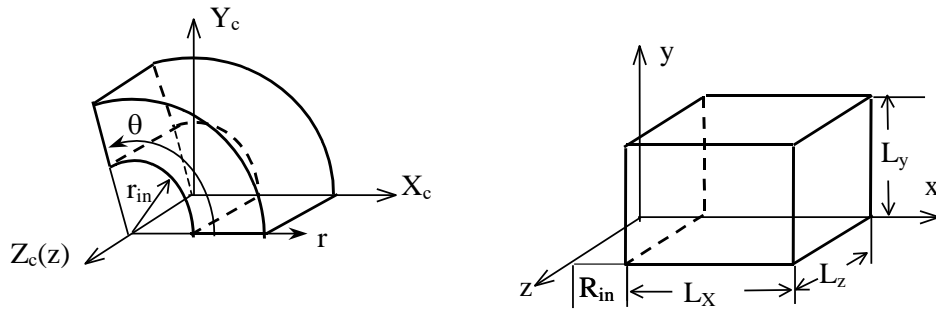


Рисунок 6.2 - Базовая расчетная область в цилиндрической системе координат  $(r, \theta, z)$  и обобщенных координатах  $(x, y, z)$ .

Пределы изменения обобщенных координат  $(x, y, z)$  БРО следующие:

$x = 0 \dots L_x; y = 0 \dots L_y; z = 0 \dots L_z$  для декартовой системы,

$x = R_{in} \dots R_{in} + L_x; y = 0 \dots L_y; z = 0 \dots L_z$  для цилиндрической системы.

Размерности координат в декартовой системе могут быть произвольны, в цилиндрической системе декартовы координаты связаны с обобщенными соотношениями

$$x_c = x \cos(y), \quad y_c = x \sin(y), \quad z_c = z,$$

при этом координата  $y$  может изменяться в пределах  $0 \dots 2\pi$ .

Для обоих типов сеток БРО создается в процессе построения *ее структурной сетки*.

### 6.1.2 Построение структурной сетки БРО

В коде структурная сетка БРО описывается тремя одномерными сетками КО вдоль осей  $x, y, z$ . Для этого вся область БРО вдоль оси  $x = 0 \dots L_x$  ( $y$  или  $z$ ) разбивается на произвольные последовательные зоны -  $\Delta L_m$ :

$$L_x = \sum \Delta L_m$$

Для каждой зоны строится своя сетка КО с использованием одного из четырех алгоритмов:

- 1) "Sym" - логарифмическая симметричная сетка *граней* КО;
- 2) "ASym" - логарифмическая несимметричная сетка *граней* КО;
- 3) "DNS\_COS" - несимметричная сетка *граней* КО, распределенных по закону косинуса;
- 4) "DNS\_TANH" - несимметричная сетка *граней* КО, распределенных по закону гиперболического тангенса.

Для первых двух алгоритмов для каждой зоны задается число КО в зоне  $N_{CV}$  и параметр логарифмичности  $q_L$ . Если  $q_L = 1$ , то в зоне строится равномерная сетка КО с постоянной шириной КО:

$$\Delta x_{ix} = \frac{\Delta L_m}{N_{CV}}.$$

Если используется *асимметричный* алгоритм построения сетки "ASym", то создаются КО разной ширины, удовлетворяющие следующему соотношению:

$$\Delta x_{ix+1} = q_L \cdot \Delta x_{ix}$$

Это позволяет «сгустить» (уменьшить ширину КО) ячейки сетки к началу ( $q_L > 1$ ) или к концу ( $q_L < 1$ ) зоны. При использовании *симметричного* алгоритма построения "Sym" сгущение при  $q_L > 1$  осуществляется симметрично к границам зоны (при  $q_L < 1$  к ее центру).

Алгоритмы "DNS\_COS" и "DNS\_TANH" строят сильно неоднородную сетку граней КО со сгущением к началу зоны ( $q_L > 0$ ) или к концу ( $q_L < 0$ ) по следующим соотношениям. Для "DNS\_COS":

$$x_{Face_i} = \begin{cases} x_{beg} + \Delta L_m [1 - \cos(\alpha_i)], & q_L > 0 \\ x_{beg} + \Delta L_m \sin(\alpha_i), & q_L < 0 \end{cases}$$

$$\alpha_i = \frac{\pi}{2} \frac{i-1}{N_{cv}-1}, \quad i = 1, \dots, N_{cv} + 1$$

Здесь  $x_{beg}$  - координата начала зоны вдоль оси,  $i$  - относительный индекс грани в зоне ( $i=1$  соответствует грани с координатой  $x_{beg}$ ).

Для алгоритма "DNS\_TANH" используется аналогичная зависимость:

$$x_{Face_i} = \begin{cases} x_{beg} + \Delta L_m \left[ 1 - \frac{\tanh(\alpha_i)}{\tanh(q_L)} \right], & q_L > 0 \\ x_{beg} + \Delta L_m \frac{\tanh(\alpha_i)}{\tanh(|q_L|)}, & q_L < 0 \end{cases}$$

$$\alpha_i = q_L \frac{i-1}{N_{cv}-1}, \quad i = 1, \dots, N_{cv} + 1$$

Эти алгоритмы в отличие от логарифмических сеток позволяют создать вблизи левой границы зоны КО с большими значениями  $q_L$ , при этом  $q_L$  вблизи правой границы будет близка к 1. Заметим, что работе с "DNS\_COS" используется только знак  $q_L$ .

### 6.1.3 Формирование расчетной области

На втором этапе в БРО помещаются специальные элементы кода Anes - Block-зоны, которые блокируют часть БРО:

$$PO = \text{БРО} - \sum_m (\text{Block-зона})_m \quad (6.1)$$

Т.е. расчетная область формируется из базовой путем «вычитания» (блокировки) областей, занятых Block-зонами. При этом «тело» Block-зоны удаляется из РО, а его поверхность преобразуется в границу расчетной области. Для наглядности на рис. 6.1 изображены двумерные (2D) расчетная и базовая области. Штриховкой на рис. 6.1 изображены области, занятые Block-зонами.

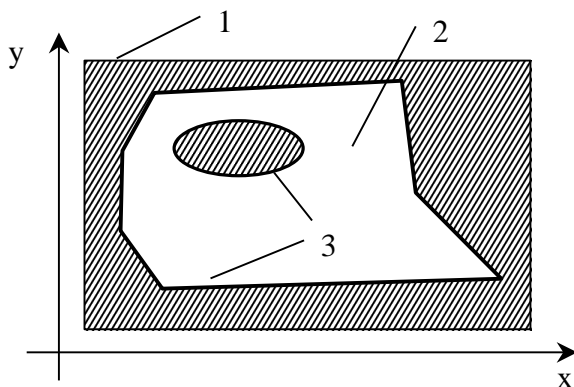


Рисунок 6.1 - Расчетная и базовая расчетная область

- 1 - базовая расчетная область,
- 2 - расчетная область
- 3 - границы расчетной области

Для описания Block-зон (и других геометрических характеристик РО) в коде используются специальные объекты, которые называются *патчами*.

Алгоритм (6.1) соответствует «стандартному» алгоритму построения РО. В этом случае БРО изначально заполнена одним из реальных материалов: либо G-фазы, либо S-фазы. В коде предусмотрен еще один алгоритм – «инверсный» алгоритм построения РО. При использовании инверсного алгоритма БРО изначально полностью «заблокирована» и для создания РО в БРО помещаются не Block-зоны, а зоны типа Flow или Sstruct, которые и формируют РО (см. рисунок 2.2).

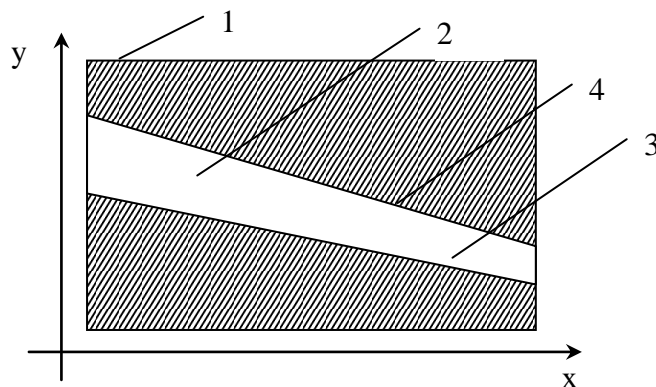


Рисунок 6.2 – Инверсный алгоритм построения РО:

- 1 - базовая расчетная область,
- 2 – зона типа FLOW,
- 3 - расчетная область,
- 4 - граница расчетной области

#### 6.1.4 Геометрические объекты и патчи

Для манипуляции с геометрией РО используются объекты, которые в коде Apes называются патчами. Патч - это:

- 1) уникальное *имя* для ссылки (произвольная комбинация 1-16 символов);
- 2) *тип* патча;
- 3) *геометрия* патча - подобласть пространства типа «точка», «поверхность» или «объем»;
- 4) дополнительные *параметры*, количество которых зависит от типа.

Структура патчей и принципы их использования подробно описаны в документе [3]. В данном документе рассмотрим способы описания геометрии патча.

Для описания объекта типа «точка» достаточно указать его координаты – тройку чисел  $(x_0, y_0, z_0)$  в обобщенных координатах. Объекты типа «точка» используются только для задания точечных источников членов в уравнениях для  $\Phi$ -переменных.

Для описания объекта типа «объем» в коде используются два формата:

- 1) 2Dbody - 3D объем образуется перемещением плоской фигуры в виде *замкнутого многоугольника* в направлении одной из обобщенных осей, при этом плоскость фигуры располагается в одной из координатных плоскостей,
- 2) 3Dbody - 3D объем описывается произвольной замкнутой поверхностью, при этом сама поверхность представляется набором *фасетов* – *треугольников*.

Для указания координаты вершин многоугольника для 2Dbody объекта используются 2GR-формат (файл с расширением \*.2gr). Для создания 2GR-файла можно использовать



либо обычный текстовый редактор, либо утилиту Anes laShaper2d. Для описания 3DBody-объектов в коде Anes можно использовать два формата:

- 1) TRI-формат программы as3d (файл с расширением \*.tri),
- 2) универсальный STL-формат (файл с расширением \*.stl).

Форматы 2GR-, TRI- и STL- файлов описаны ниже.

Объекты типа «*поверхность*», представляющие из себя плоские или криволинейные поверхности, могут быть созданы либо непосредственно пользователем, либо автоматически при обработке Body-объектов А-компилятором кода. В первом случае можно использовать только плоские поверхности, расположенные в одной из координатных плоскостей. Для их описания используется граничный бокс у которого один из размеров  $S_x$ ,  $S_y$  или  $S_z$  равен нулю и два формата описания:

- 1) Rect - поверхность представляет собой прямоугольник, который совпадает с гранью граничного бокса,
- 2) 2Dsurf - поверхность образована *замкнутым многоугольником* 2Dbody объекта.

### 6.1.5 Форматы файлов геометрических объектов

2Dbody объект описывается *замкнутой полигональной линией* (многоугольником) в приведенной системе 2D координат ( $x_1, x_2$ ). Привязка этих координат к осям РО зависит от направления «продления» фигуры – DirAxes:

для DirAxes = z :  $x_1 = x$  ;  $x_2 = y$ ;

DirAxes = y :  $x_1 = z$  ;  $x_2 = x$ ;

DirAxes = x :  $x_1 = y$  ;  $x_2 = z$ ;

Многоугольник описывается заданием координат ( $x_1, x_2$ ) его вершин, начиная с произвольной вершины. Обход вершин всегда осуществляется *против часовой стрелки*. При обработке многоугольника последняя вершина соединяется с первой. Масштаб приведенных координат может быть абсолютно произвольным. Для хранения этого описания используются файлы с расширением \*.2gr. Структура это файла:

- 1) координаты точек многоугольника, разделенные запятой или пробелами, записываются на отдельной строке,
- 2) в самом файле можно использовать комментарии – произвольный текст после символа «!».

Приведем пример использования 2GR-файла следующей структуры:

```
! x , y
0.0000E+000 , 2.0000E+001
1.0000E+002 , 0.0000E+000
8.0000E+001 , 5.0000E+001
5.0000E+001 , 3.0000E+001
3.0000E+001 , 6.0000E+001
```

На рисунке 6.3 показаны сам многоугольник и его 2DBody объекты для разных направлений оси «продления» объекта.

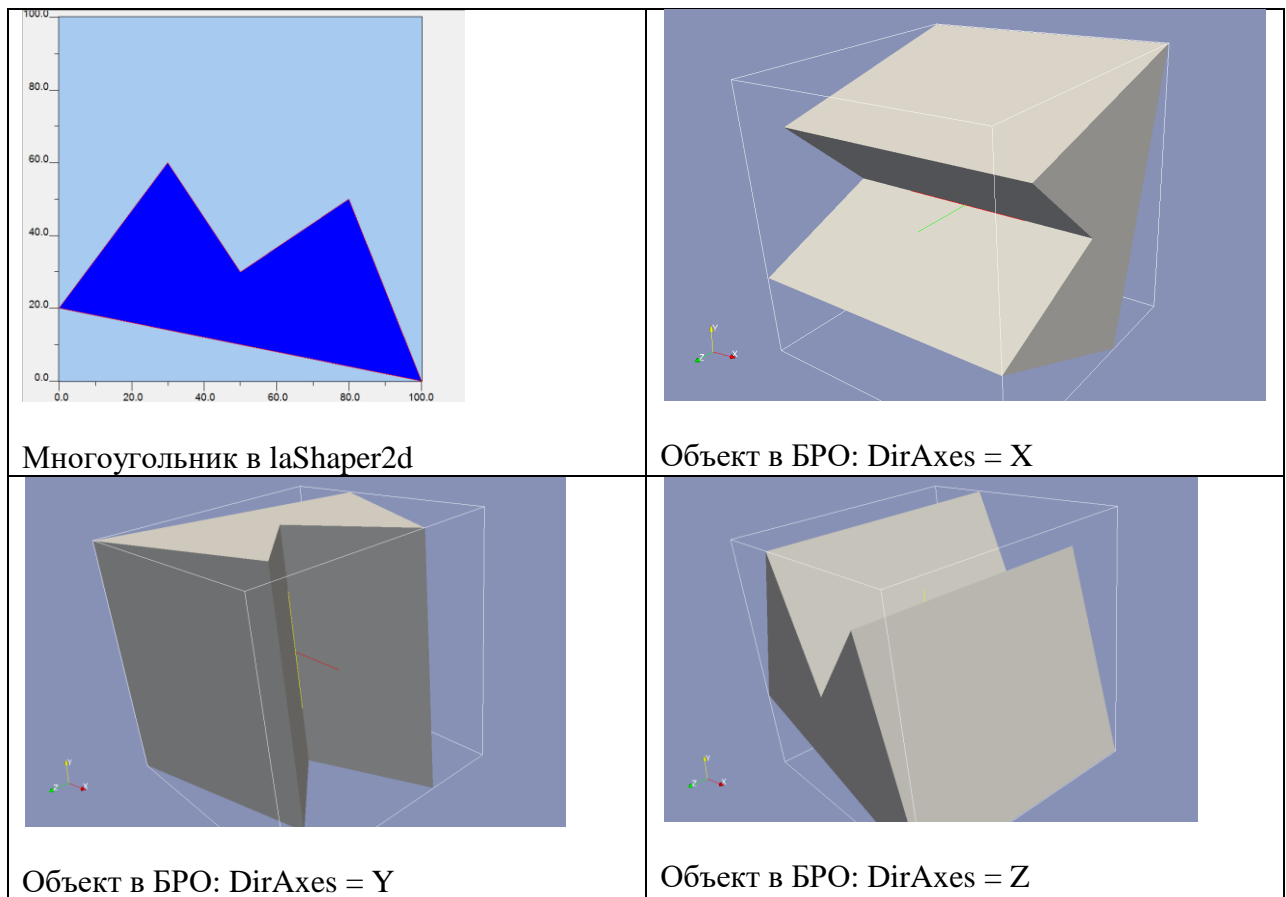


Рис. 6.3 – Объект 2DBody

Для описания 3DBody в коде можно использовать файлы двух форматов: TRI и STL.

*TRI-файл* представляет собой текстовый файл (с расширением \*.tri), на каждой строке которого описывается *один треугольник* поверхности тела. Сам треугольник задается декартовыми координатами своих вершин (всего в строке  $3 * 3 = 9$  действительных чисел). Вершины треугольника должны обходиться *против часовой стрелки* с вершины его *внешней* нормали. Числа разделяются запятыми или пробелами. Допускаются строки комментарии – в первой позиции символ «!».

*STL-формат* является универсальным форматом для описания поверхности трехмерных объектов (расширение файла - \*.stl). Допускаются два формата файла – двоичный и текстовый. Текстовый формат по сути похож на формат TRI-файла: это набор треугольников, описанных через свои три декартовы точки:

```

solid <комментарий>
facet normal 0.000000e+000 -1.000000e+000 5.000000e-001 <- координаты внешней нормали
  outer loop
    vertex 5.000000e-001 5.000000e-001 1.000000e+000 <- координаты вершин
    vertex 0.000000e+000 0.000000e+000 0.000000e+000 <- координаты вершин
    vertex 1.000000e+000 0.000000e+000 0.000000e+000 <- координаты вершин
  endloop
endfacet
.....
endsolid

```

Для формата STL порядок обхода вершин не играет роли (внешняя нормаль записана в файле), однако лучше всего обходить их *против часовой стрелки со стороны внешней нормали*.

Для создания 3Dbody примитивов можно воспользоваться любым САД-пакетом или пакетом 3D графики. Для создания простых объектов можно воспользоваться бесплатной утилитой OpenSCAD, которая работает во всех ОС.

### 6.1.6 ГеоОбъекты и модификация сетки РО

На третьем этапе построения расчетной области производится модификация исходной структурной сетки БРО. Алгоритмы этого этапа существенно отличаются для структурных и неструктурных сеток.

Для *структурных сеток* обработка всех патчей сводится к «раскраске» КО, сама сетка при этом не меняется. А-компилятор с каждым КО связывает ряд характеристик, в частности тип КО: заблокированный, содержащий G- (и или) S-фазу.

Для *неструктурных сеток* А-компилятор вызывает *генератор построения неструктурной сетки*, который использует исходную структурную сетку БРО в качестве начальной грубой сетки (CoarseMesh).

В качестве исходной информации для генерации неструктурной сетки используется эта *грубая* структурная сетка и набор патчей, влияющих на процесс дробления сетки. В качестве таких патчей используются:

- 1) BLOCK, STRUCT, FLOW и POROUS – патчи (далее эти патчи будут называться Blockage-патчи);
- 2) 2D BSWALL патчи;
- 3) 2D BS патчи;
- 4) VOLUME патчи с заданным параметром FixLevel.

В качестве исходных данных для процесса дробления используются:

- 1) MaxLevel – максимальный уровень дробления сетки (исходная грубая сетка соответствует нулевому уровню);
- 2) NoLayers – число слоев ячеек одного уровня для сглаживания перехода между уровнями (по умолчанию = 2);
- 3) BSLevel – минимальный уровень дробления ячеек вблизи BS-патчей;
- 4) BSWallLevel – минимальный уровень дробления ячеек вблизи BSWall-патчей;
- 5) для использования любого VOLUME патча в качестве FixLevel патча необходимо задать параметр FixLevel (с помощью оператора секции [Special Data]).

Для построения неструктурной сетки используются следующие простые правила:

- 1) вблизи Blockage-патчей должны быть ячейки с MaxLevel уровнем дробления;
- 2) вблизи BS-патчей должны быть ячейки с уровнем дробления не меньшим, чем BSLevel;
- 3) вблизи BSWall-патчей должны быть ячейки с уровнем дробления не меньшим, чем BSWallLevel;
- 4) ячейки, расположенные внутри BLOCK-патчей удаляются из сгенерированной сетки;
- 5) ячейки одного уровня должны располагаться слоями: для любой ячейки можно выбрать направление, в котором будет NoLayers ячеек ее уровня;
- 6) поверхность раздела ячеек разного уровня должна быть достаточно гладкой: для любой ячейки число соседей другого уровня не должно превышать 3 в двумерных задачах и 4 в трехмерных;
- 7) не допускается соседство ячеек, уровни которых отличаются больше чем на единицу.
- 8) ячейки, попадающие в Volume-патчи с заданным FixLevel, должны иметь уровень дробления не меньший чем FixLevel.

Сам алгоритм работы генератора достаточно простой:

1. При разбиении ячейки создаются четыре (для 3D задачи) или 2 (2D задачи) одинаковые ячейки;
2. В процессе разбиения создается дерево ячеек, начиная с ячеек грубой сетки. Для этого при разбиении ячейки ее индекс запоминается в ячейках-потомках. Глубина дерева равна MaxLevel. Дерево внутри генератора сетки используется для быстрого нахождения соседей для любой ячейки просмотра всех низовых ячеек.
3. Ячейка требует дробления, если:
  - поверхность любого объекта проходит через ячейку и еще не достигнут нужный уровень разбиения (MaxLevel, BSLevel, BSWallLevel, FixLevel);
  - ячейка помечена на разбиение алгоритмами сглаживания и создания слоев;
4. Процесс дробления осуществляется в цикле, начиная с ячеек грубой сетки. Процесс прекращается, когда ячейки, помеченные для разбиения, отсутствуют.
5. Более подробно алгоритм работы описан в [5].

Для ячеек, расположенных на границе Blockage-патчей (BLOCK, FLOW, STRUCT, POROUS), предусмотрено две модели аппроксимации границы объекта:

1. модель целых ячеек,
2. модель дробных ячеек.

Отличия этих моделей показаны на рисунке 6.4.

Для быстрой проверки прохождения поверхности объекта через любую ячейку любого уровня в генераторе сетки используется алгоритм *лучевого сканирования* объектов-патчей, участвующих в построении сетки. Кратко суть этого алгоритма сводится к следующему.

Наряду с первоначальной грубой структурной сеткой можно построить еще одну структурную сетку – так называемую мелкую сетку - FineMesh. Эта сетка получается из грубой сетки путем разбиения ее KO MaxLevel раз.

Луч – это линия, направленная вдоль одной из осей x,y,z и проходящая через центр KO fine mesh (для модели целых ячеек) или через ребро KO (для модели дробных ячеек). При сканировании на луче рассчитываются и запоминаются точки пересечения (и их характеристики) для всех объектов-патчей, участвующих в построении сетки.

В процессе сканирования проводятся лучи через все KO FineMesh в трех направлениях.

Эта информация в дальнейшем позволяет легко и быстро определить прохождение поверхности через любую ячейку.

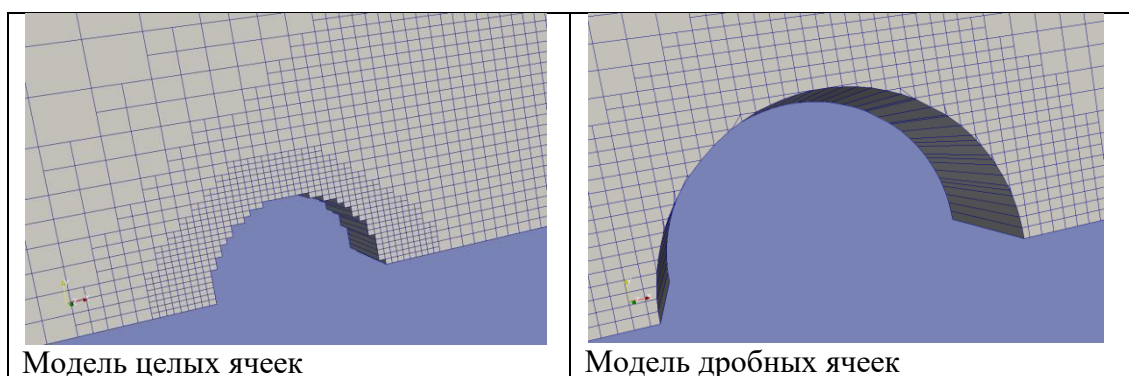


Рисунок 6.4 – Модели целых и дробных ячеек

**Замечание №1.** Несмотря на то, что для работы с неструктурными сетками в Решателе используется полностью неструктурная концепция (произвольные одномерные списки ячеек и граней), «остатки» структурности сохраняются.

Если наложить на неструктурную сетку ячеек (с разными уровнями дробления) структурную сетку FineMesh, соответствующую максимальному уровню дробления, то с каж-

дой ячейкой любого уровня можно связать структурные индексы (ifx,ify,ifz). Эти индексы называются Fine-индексами ячейки. Смысл их очень простой – это индексы FineMesh левой, нижней, дальней Fine-ячейки, входящей в состав данной ячейки.

Эти индексы существенно облегчают работу с полями Ф-переменных в постпроцессоре Anes.

 **Замечание №1.** Если положить уровни дробления

$\text{MaxLevel} = \text{BSLevel} = \text{BSWallLevel} = 0,$

то неструктурная сетка ничем не будет отличаться от структурной сетки в декартовой системе координат (но при этом можно использовать алгоритм дробных ячеек!).

## 6.2 XR(YFI , ZZ) GRID. Структурная сетка БРО

Для построения структурной сетки БРО используются три секции:

[XR Grid] , [YFI Grid] , [ZZ Grid].

которые строят одномерные сетки КО вдоль обобщенных координатных осей x,y,z. Построение сеток осуществляется последовательностью операторов, которые реализуют алгоритм, описанный в пункте 6.1.2:

Zone(Len,NoCV, QCV, ASYM/ SYM/ DNS\_COS/ DNS\_TANH)

Здесь

Len - ширина зоны  $L_m,$

NoCV - число КО в зоне  $N_{cv},$

qCV - параметр логарифмичности сетки  $q_L.$

При использовании цилиндрической системы координат граница расчетной области по x (r) начинается со значения, определяемого оператором

Rin = 0.0

Если одна из секций пустая или число КО по координате равно единице, то по этой координате нет *размерности*. Такой вариант используется для решения 2D задач.

## 6.3 Unstructured Cartesian Grid. Неструктурная сетка

Эта секция используется для настройки генератора неструктурной сетки. Описание неструктурных сеток, используемых в коде, приведено в документе [5].

Она используется, если оператор секции [Main] равен

TypeMesh = TM\_UNCARTES или TM\_UNCYLIND

Операторы этой секции можно разбить на две категории: базовые операторы и операторы «тонкой» настройки.

### 6.3.1 Базовые операторы построения неструктурной сетки

К этой категории относятся операторы, определяющие основные параметры дробления грубой сетки [5]:

MaxLevel = 3

оператор задает максимальный уровень разбиения вблизи поверхностей Block, BlockWall, Struct, Flow и Porous патчей.

NoLayers = 2

оператор определяет число слоев сглаживания.

BSLevel = <MaxLevel>

оператор задает уровень разбиения для поверхностных BS-патчей.

BSWallLevel = <MaxLevel>

оператор задает уровень разбиения для поверхностных BSWall-патчей.

IsNoRefZ = .F.

оператор отключает процесс дробления ячеек в направлении оси z для трехмерных задач,

IsCUTCELL = .F.

оператор активизирует модель дробных ячеек.

IsCycleBC\_XY = .F.

оператор активизирует модель наклонных периодических граничных условий в плоскости (x,y).

### Оператор

AlgoSFC = **SFC\_NONE** ! SFC\_NONE,SFC\_Metis,SFC\_Morten,SFC\_BYIX,  
SFC\_BYIY, SFC\_BYIZ, SFC\_BYIXYZ\_METIS,

задает алгоритм нумерации неструктурных ячеек:

SFC\_NONE - ячейки расположены в том же порядке, как и в дереве дробления.

SFC\_byIZ - ячейки нумеруются в порядке возрастания одномерного индекса  
 $ifxyz = ifx + (ify-1)*NFY + (ifz-1)*NFX*NFY$ .

SFC\_byIY - ячейки нумеруются в порядке возрастания одномерного индекса  
 $ifxyz = ifz + NFZ*(ifx-1)+NFZ*NFX*(ify-1)$ .

SFC\_byIX - ячейки нумеруются в порядке возрастания одномерного индекса  
 $ifxyz = ify + NFY*(ifz-1)+NFZ*NFY*(ixf-1)$ .

SFC\_Morten - ячейки нумеруются в соответствии с алгоритмом Мортена.

SFC\_Metis - ячейки нумеруются в соответствии с алгоритмами пакета MeTis.

SFC\_IXYZ\_Metis - это наиболее эффективный алгоритм перенумерации для проведения параллельных вычислений.

В этом режиме ячейки перенумеруются в два этапа. На первом этапе используется перенумерация SFC\_byIX/ SFC\_byIY / SFC\_byIZ (выбирается направление для максимального NFX,NFY,NFZ), на втором этапе используется алгоритм MeTis.

Приведем типичный пример построения неструктурной сетки. На рисунке 6.5 показаны патчи, используемые для построения сетки, на рисунке 6.6 – построенная сетка КО. Ниже приведены операторы, используемые для построения:

#### [Patches]

```
Domain(FLOW,"WAT")
DPatch("In",BS,W_BS)
DPatch("Out",BS, E_BS)
2DPatch("CYL",BlockWall, LX_B,-1*RCYL,0, 2*RCYL, 2*RCYL,RCYL,"CIRCLE",Z_DIR)
Patch("FGE_AFTER",VOLUME, LX_B+2*RCYL,0,0, 6.0*RCYL,2.2*RCYL,RCYL)
```

#### [Special Data]

```
I("MINLEVEL.FGE_AFTER") = 3
```

#### [Unstructured Cartesian Grid]

```
MaxLevel = 4
NoLayers = 4
BSLevel = 0
BSWallLevel = 0
IsCUTCELL = .T.
```

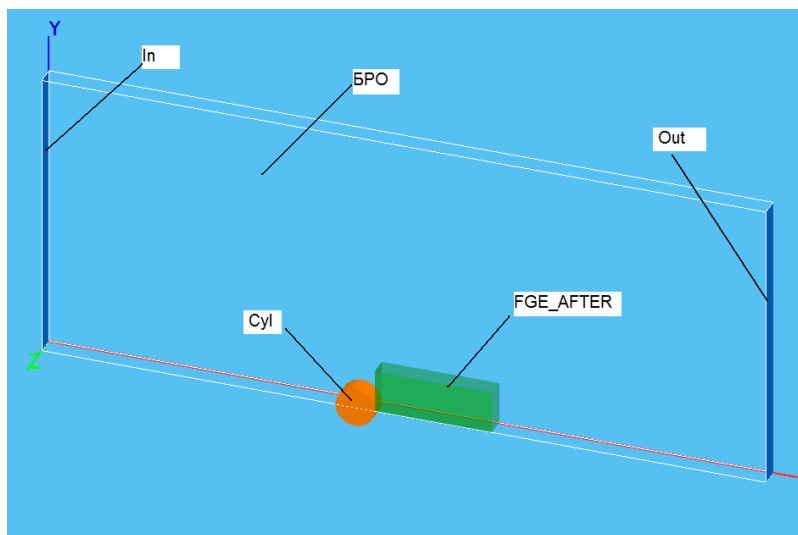


Рисунок 6.5 – БРО и патчи

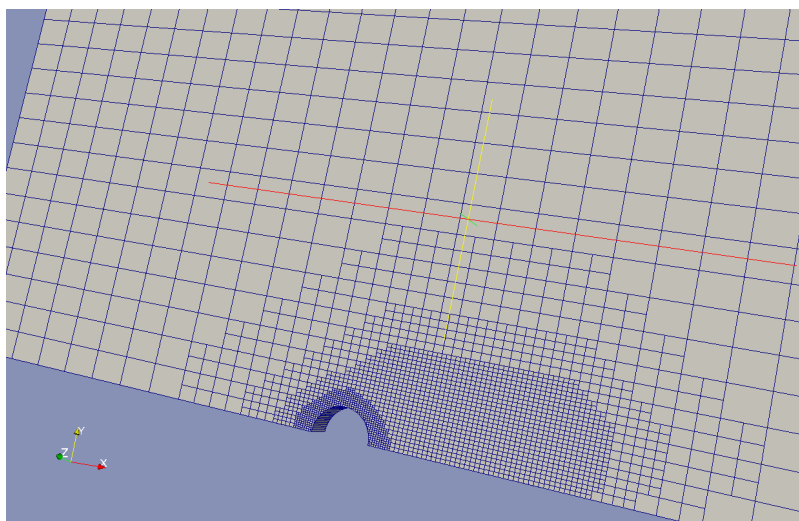


Рисунок 6.6 – Сетка КО

### 6.3.2 Специальные операторы построения неструктурной сетки

Эта группа операторов представляет интерес только для разработчиков кода. Они в первую очередь предназначены для развития и отладки алгоритмов построения неструктурной сетки.

`CriFracEdge = 0.1`

оператор определяет предельную долю пересечения на ребре, меньше которой точка пересечения переносится в вершину КО.

`EQUOTOL = 1.e-5`

оператор задает относительную точность сравнения точек пространства в алгоритме сканирования лучей.

`IsScanX = .T.`

`IsScanY = .T.`

`IsScanZ = .T.`

флаги отключения сканирования в направлении координатной оси. По умолчанию сканирование производится по всем направлениям.

BASECROSS = 20

оператор определяет шаг выделения памяти для точек пересечения на луче.

Для формирования выдачи отладочной информации в log-файл А-компилятора используются следующие флаги:

Оператор	По умолчанию	Назначение
DbgStGrid	.FALSE.	печать структурной fine-сетки;
DbgUnCreate	.FALSE.	печать ошибок при создании неструктурных ячеек из дерева ячеек;
DbgTestScan	.FALSE.	печать информации о трех первых ошибках сканирования по осям;
DbgScan2D	.FALSE.	печать всех лучей в 2D случае;
DbgRef	.FALSE.	печать информации о процессе дробления;
DbgTestCell	.FALSE.	Печать информации о процессе проверки правильности создания неструктурных ячеек;
DbgCutCell	.FALSE.	печать ошибок при обработке дробных ячеек;
DbgAllCutCell	.FALSE.	печать информации обо всех дробных ячейках;
IsSplitPH	.TRUE.	при выводе POLYHED ячейки в VTK разбивать ее на линейные под-ячейки;

#### Операторы

iDbgIX = 0

iDbgIY = 0

iDbgIZ = 0

задают fine-индекс А-ячейки (это ячейка в дереве ячеек) для вывода полной отладочной информации о А-ячейке. Заметим, что вывод осуществляется только если DbgCutCell = .F.

#### Оператор

iDbgCellID = 0

индекс неструктурной ячейки сетки для выдачи полной информации о ее топологии: создании VTK-файла ячейки и вывода ее fine-индексов ifx,ify,ifz.

## 6.4 TIME GRID. Сетка по времени

Эта секция описывает операторы управления шагами по времени. При решении стационарных задач эта секция не используется.

Управление шагами по времени и шагом выдачи в файл результатов определяется последовательностью операторов:

TZone(TimeBeg,TimeEnd,DTime,StepRes,ModelChange)

где

TimeBeg,TimeEnd - интервал зоны по времени,

DTime - шаг по времени,

StepRes - шаг по времени выдачи в файл результатов,

ModelChange - параметр, определяющий алгоритм автоматического изменения шага.

В текущей версии шаг не меняется в пределах зоны и для последнего параметра необходимо указать значение tsCONST.



## 6.5 PATCHES

Эта секция служит для описания патчей РО. Напомним, что патч - это структура, которая связывает геометрический объект расчетной области (типа «точка», «поверхность» или «объем») с набором характеристик, используемых:

- 1) для формирования расчетной области из базовой расчетной области,
- 2) для привязки к зонам расчетной области источниковых членов и граничных условий,
- 3) для описания зон Struct, Flow и Porous.

### 6.5.1 Определение материала БРО

Первым оператором секции должен быть оператор определения материала базовой расчетной области. Вид этого оператора следующий:

```
Domain(ТипБРО, "ИмяМатериала")
```

здесь

ТипБРО - тип материала FLOW / STRUCT / BLOCK; значение BLOCK соответствует «инверсному» алгоритму создания РО;  
ИмяМатериала - имя материала G- или S- фазы, для BLOCK указывается пробел.

### 6.5.2 Геометрия. Стандартный патч

Стандартный оператор патча для описания геометрического объекта используем сам граничный бокс патча:

```
Patch("ИмяПатча",PType,x0,y0,z0,SX,SY,SZ, ...)
Patch("ИмяПатча",POINT,x0,y0,z0, 0,0,0)
Patch("ИмяПатча",SURFACE,x0,y0,z0, SX,SY,SZ, SurfUP/SurfDOWN)
```

где

ИмяПатча - уникальное имя (до 16 символов) используемое для ссылки на патч;  
Ptype -тип патча: Block (BlockWall), BS (BSWall), Volume, Struct, Porous, Flow;  
x0,y0,z0 -координата начала граничного бокса патча,  
SX, SY, SZ -размеры граничного бокса патча; для патча с типом SURFACE и BS один из размеров *должен быть равен нулю*.

Параметры SurfUP/SurfDOWN в поверхностном патче SURFACE используются для определения грани КО, к которой «привязывается» поверхностный патч (см. раздел 2.6 документа [3]).

### 6.5.3 Геометрия. Стандартный повернутый патч

Этот стандартный патч, который поворачивается относительно одной из осей RotDir (X\_DIR / Y\_DIR / Z\_DIR) на угол Angle:

```
RotPatch("ИмяПатча",PType,x0,y0,z0,SX,SY,SZ, RotDir,Angle,...)
```

Угол поворота задается в градусах, положительное значение соответствует повороту *против* часовой стрелки. Данный патч можно применять для Ptype = Block, BlockWall, Struct, Flow и POrous.

### 6.5.4 Укороченные патчи для внешних границ

Эти патчи являются модификациями стандартного патча типа BS и используются для описания внешних границ параллелепипеда базовой расчетной области:

```
DPatch("ИмяПатча",BS/BSWall,TBound)
```

где параметр Tbound может принимать следующие значения (см. рисунок 6.1):

E\_BS, W\_BS - правая и левая граница ( по x),  
 N\_BS, S\_BS - верхняя и нижняя граница ( по y),  
 H\_BS, L\_BS - передняя и задняя граница ( по z).

### 6.5.5 Патчи с объектами 2Dbody

Для описания патчей, использующих 2Dbody объекты, служат операторы 2DPatch:

```
2DPatch("ИмяПатча",PType ,x0,y0,z0,SX,SY,SZ, Name2GR, DirAxes, ...)
```

здесь

ИмяПатча - уникальное имя (до 16 символов) используемое для ссылки на патч;  
 Ptype -тип патча: Block (BlockWall), BS (BSWall), Volume, Struct, Flow;  
 DirAxes - X\_DIR / Y\_DIR / Z\_DIR - ось, вдоль которой «продлевается» 2D фигура;  
 Name2GR - имя 2GR файла (без расширения!).

При обработке патчей в А-компиляторе поиск 2GR-файлов осуществляется в следующем порядке:

1. сначала в рабочем каталоге решения – WorkDir,
2. потом в каталоге с файлом проекта,
3. и наконец, в каталоге <Anes>/geobj.

### 6.5.6 Патчи с объектами 3Dbody

Для описания патчей, использующих 3Dbody объекты служат операторы 3DPatch:

```
3DPatch("Имя патча",PType ,x0,y0,z0,SX,SY,SZ, GeoFile, ...)
```

где GeoFile - имя файла с описанием фасетов примитива. В текущей версии кода можно использовать два формата файла TRI и STL.

При обработке оператора используется следующий алгоритм поиска файла:

- 1) файл <GeoFile>.tri в рабочем каталоге WorkDir,
- 2) файл <GeoFile>.stl в рабочем каталоге WorkDir,
- 3) файл <GeoFile>.tri в каталоге А-файла,
- 4) файл <GeoFile>.stl в каталоге А-файла,
- 5) файл <GeoFile>.tri в каталоге <Anes>/geobj,
- 6) файл <GeoFile>.stl в каталоге <Anes>/geobj.

### 6.5.7 Флаги-переключатели секции

Для управления созданием дополнительных патчей Решателя используются специальные флаги-переключатели. Эти флаги действуют на все операторы описания патчей, расположенных после них в секции [Patches]:

IsCreateSF = **.FALSE**/.TRUE.

флаг-переключатель создания SF-патча (сопряженный к FS\_ патчу со стороны Struct). Действует на патчи типа STRUCT.

IsCreateSS = *.FALSE./TRUE.*

флаг-переключатель создания SS-патча (создается внутри Struct для задания термического сопротивления). Действует на патчи типа STRUCT.

IsCreateSIO = *.FALSE./TRUE.*

флаг-переключатель создания SI- и SO-патчей (создается внутри и вне Struct границы с другой Struct-зоной). Действует на патчи типа STRUCT. Этот флаг работает только для *структурных сеток*.

IsWallBS = *.FALSE./TRUE.*

флаг-переключатель признака WALL для патчей с именем BS\_..., которые создаются в «инверсном» режиме для патчей типа FLOW, STRUCT.

### 6.5.8 Описание границ расчетной области

Как уже отмечалось, границы расчетной области (PO) в пакете строятся по простому алгоритму

POЗ = <Параллелепипед базовой PO> – Сумма(патчей типа Block)

в котором для описания патчей используются операторы

```
Patch("ИмяПатча",BLOCK/BlockWall ,x0, y0, z0, SX,SY,SZ)
RotPatch("ИмяПатча",BLOCK/BlockWall ,x0, y0, z0, SX,SY,SZ,RotDir,Angle)
2DPatch("ИмяПатча",BLOCK/BlockWall,x0,y0,z0,SX,SY,SZ,GeoName, DirAxe)
3DPatch("ИмяПатча",BLOCK/BlockWall,x0,y0,z0,SX,SY,SZ,GeoFile)
```

Для привязки к зоне BLOCK "другой" границы служит патчи

```
Patch("ИмяПатча",BS/BSWall,x0,y0,z0,SX,SY,SZ)
2DPatch("ИмяПатча",BLOCK/BlockWall,x0,y0,z0,SX,SY,SZ,GeoName, DirAxe)
DPatch("ИмяПатча",BS/BSWall, E_BS/W_BS/ N_BS/S_BS/ H_BS/L_BS)
```

при этом один из размеров SX/SY/SZ должен быть равным нулю и плоскость патча должна *касаться* зоны патча Block, описанного *выше* в секции. Отметим, что по умолчанию, внешние границы расчетной области представляют собой зоны Block и к ним можно привязывать BS - патчи.

Напомним, что флаг Wall в патчах типа Block и BS, используются для указания границ расчетной области, на которых рассчитываются пристенные функции для моделей турбулентности и задаются нулевые граничные условия «по умолчанию».

Пример использования таких патчей показан на рисунке 6.7.

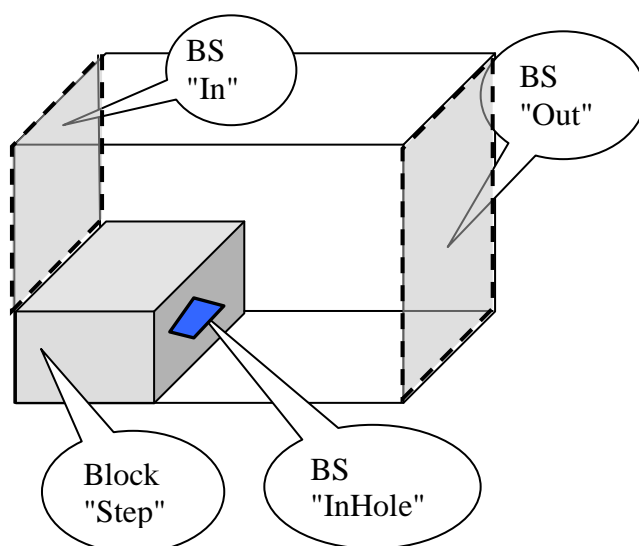


Рисунок 6.7 - Использование BLOCK и BS патчей

### 6.5.9 Описание Flow и Struct зон

Для описания распределения S -фазы (Struct или Porous -зон) используются операторы:

```
Patch("ИмяПатча",Struct/Flow,x0,y0,z0,SX,SY,SZ,"ИмяМатериала")
RotPatch("ИмяПатча", Struct/Flow ,x0, y0, z0, SX,SY,SZ,RotDir,Angle,"ИмяМатериала")
2DPatch("Имя_патча", Struct/Flow,x0,y0,z0,SX,SY,SZ,GeoName, DirAxe,"ИмяМатериала")
3DPatch("Имя_патча", Struct/Flow,x0,y0,z0,SX,SY,SZ,GeoFile,"ИмяМатериала")
```

где

ИмяМатериала - имя материала для задания свойств S-фазы или G-фазы в секции [Properties].

По умолчанию после обработки операторов Struct и Flow-патчей создается два патча Решателя:

- 1 патч с именем "ИмяПатча" типа VOLUME для привязки источниковых членов,
- 2 поверхностный патч с именем "FS\_ИмяПатча", используемый для описания границы раздела фаз G и S.

Пользователь может создать ряд дополнительных поверхностных патчей, используя флаги-переключатели.

### 6.5.10 Описание POROUS зон

Для описания пористых зон используются операторы двух типов:

```
Patch("ИмяПатча",POROUS,x0,y0,z0,SX,SY,SZ,"Имя S-материала", "Имя FS-зоны ")
RotPatch("ИмяПатча", POROUS ,x0, y0, z0, SX,SY,SZ,RotDir,Angle,
"Имя S-материала", "Имя FS-зоны ")
```

и

```
Patch("ИмяПатча",POROUS,x0,y0,z0,SX,SY,SZ,"*", "ИмяFS-зоны ")
RotPatch("ИмяПатча", POROUS ,x0, y0, z0, SX,SY,SZ,RotDir,Angle,"*", "ИмяFS-зоны ")
```

где

Имя S-материала - имя материала S-фазы;

Имя FS-зоны - имя для ссылки на модель межфазного F-S взаимодействия в секции [Porous Models].

Патчи Porous всегда используются в режиме «наложения», поэтому материал G-фазы не указывается. Используется материал, установленный предыдущими патчами.

Вторая группа патчей, в которой не задан материал S-фазы, используется для описания зон *двухуровневой пористой модели*.

### 6.5.11 Описание зон для источниковых членов

Для привязки объемных, поверхностных и точечных источников используются патчи:  
Объемный источник

Patch("Имя патча",VOLUME,x0,y0,z0,SX,SY,SZ)

Поверхностный источник (одно из pSX, pSY, pSZ должно быть = 0)

Patch("Имя патча",SURFACE,x0,y0,z0,pSX,pSY,pSZ,SurfUP/SurfDOWN)

здесь SurfUP/SurfDOWN - направление нормали поверхности в положительном направлении оси, определяющую используемую сторону поверхности.

Точечный источник

Patch("Имя патча",POINT,x0,y0,z0,0,0,0)

Отметим, что патчи типа STRUCT, FLOW и POROUS также можно использовать для привязки объемных источников.

### 6.5.12 Цвета патчей

Для более наглядного представления патчей в постпроцессоре предусмотрена возможность для патчей типа BLOCK, STRUCT и POROUS задать индекс цвета в виде *последнего необязательного* параметра

ICOLOR = 1 .. 8

Если параметр не задан, то используются индексы цвета по умолчанию:

1 : для STRUCT, POROUS и BLOCK,

2 : для BLOCKWall.

Сами цвета для заданных индексов настраиваются в Постпроцессоре Anes [6].

### 6.5.13 Разбиение патчей на суб-патчи

При построении PO для сложных задач с использованием *неструктурных* сеток наиболее правильным и удобным является использование операторов 3DPatch типа Block, Struct или Flow.

В этом случае граница расчетной области, связанная с объектом, будет описываться одним патчем Решателя типа BS (граница PO) или FS (граница Flow- и Struct- зон). Соответственно к этой границе можно «привязать» только одно граничное условие для каждой Ф-переменной или источник для FS (SF) межфазной границы.

Для задания разных ГУ или источников такой поверхностный патч можно разбить на несколько патчей. Для этого можно использовать следующий алгоритм:

1. Необходимо создать Volume патчи с помощью операторов Patch. Имена этих патчей будут использоваться в качестве имен новых суб-патчей. Их граничный бок (x0,y0,z0,SX,SY,SZ) будет использоваться для отбора граней ячеек патча-родителя.

2. Для связи суб-патчей с патчем-родителем необходимо в секции [Special Data] указать «символьный» оператор

```
C("SUBPATCH.<Имя суб-патча>") = "<Имя патча-родителя>".
```

3. При обработке патчей в компиляторе Anes Volume патчи будут удалены из списка патчей и они будут преобразованы в патч-родитель. При этом в списке граней патча останутся только грани ячеек, центры которых попадают в граничный бокс суб-патча. При этом грани нового патча удаляются из списка граней родителя.

Рассмотрим типичный пример – моделирование солнечного коллектора. Объем коллектора описывается с помощью STL-объекта:

[Special Data]

```
C("SubPatch.BoxUP") = "WALL"
C("SubPatch.BoxBOT") = "WALL"
```

.....

[Patches]

```
3DPatch("WALL",BlockWall,0,0,0, LX,LY,LZ,"box_9")
Patch("BoxUP",VOLUME,0,0.95*LY,0,LX,0.06*LY,LZ)
Patch("BoxBOT",VOLUME,0,0,0,LX,0.05*LY,LZ)
```

Компилятор Anes преобразует поверхность объекта box\_9 в одну границу расчетной области - WALL. Для выделения верхней и нижней части бака коллектора (это нужно для расчета температуры верхней и нижней поверхностей бака) используются суб-патчи BoxUP и BoxBOT.

Замечание. При работе с патчем типа SF следует помнить, что вместе с ним существует патч FS с тем же списком граней. Если создается суб-патч для SF (или FS) то его грани будут одновременно удалены и из списка SF и FS родителя!

#### 6.5.14 Патчи для описания свойств S-фазы

При описании S-зоны со сложной геометрией, заполненной разными материалами используется набор 3D патчей типа Struct. При этом патчи либо перекрывают друг друга, либо «касаются» друг друга. При работе с неструктурными сетками все эти патчи используются в алгоритме построения неструктурной сетки. Часто это приводит к появлению ошибок построения неструктурных ячеек.

В этом случае можно использовать так называемые Property-патчи. Property-патч - это обычный Patch, 2DPatch или 3DPatch патч типа Struct, имя которого начинается с символов "#X\_...". Такие патчи не используются для построения сетки. Их обработка производится *после построения сетки*. Сама обработка сводится к следующему:

- 1) перебираются все ячейки сетки и проверяется попадание центра ячейки внутрь объекта патча;
- 2) если ячейка внутри объекта, то ее индекс материала S-фазы заменяется на индекс S-фазы Property-патча.

Заметим, что Property-патчи обрабатываются после обработки других патчей (которые участвуют в построении сетки) в порядке их записи в секции [Patches].

## 7. СПЕЦИФИКА ЗАДАЧИ

### 7.1 *Properties. Описание свойств материалов PO*

Эта секция используется для описания свойств материалов G- и S- фаз. Логика задания свойств достаточно простая – для всех материалов, введенных в секции [Patches] необходимо задать свойства.

#### 7.1.1 Прямое задание свойств

Свойства материалов задаются оператором

PROP("ИмяМатериала.ИмяСвойства") = V-переменная

где в качестве значения используется любая форма V-переменной. В качестве индекса оператора используются два имени, разделенные точкой.

#### Однокомпонентная G-фаза. Предопределенные Ф-переменные

В этом случае необходимо задать следующие свойства:

Dens - плотность,  
 Visc - кинематическая вязкость,  
 Cp - теплоемкость однокомпонентной G-фазы

$$c_{pg} = \left( \frac{\partial h_g}{\partial T_g} \right)_p$$

BetaT, BetaP - значения коэффициентов термического расширения и изотермической сжимаемости

$$\beta_{Tg} = -\frac{1}{\rho_g} \left( \frac{\partial \rho_g}{\partial T_g} \right)_p, \quad \beta_{Pg} = \left( \frac{\partial \rho_g}{\partial p} \right)_{Tg}$$

Если эти свойства не указано, то считается, что значения равны нулю,

H - тепловая составляющая энтальпии однокомпонентной фазы  $h_g$ . Это свойство используется только, если теплоемкость переменная. В случае постоянной теплоемкости тепловая составляющая энтальпии рассчитывается по соотношению

$$h_g = c_{pg} \cdot T_g$$

Важное замечание: Если задан алгоритм вычисления энтальпии, то энтальпия и теплоемкость должны быть согласованы, т.е.

$$c_{pg} = \left( \frac{\partial h_g}{\partial T_g} \right)_p$$

Если в секции [PHI Variables] для температуры выбрано задание числа Прандтля, то вместо "Cond" нужно задавать свойство "Pran".

В коде предполагается, что тепловая энтальпия  $h_g$  равна нулю при  $T_g = 0$ . При наличии граничных условий типа BC\_FIXFLUX или источников с SR\_FIXSOURCE для температуры  $T_g$  и наличии для этих источников массовых потоков в коде учитываются «полное» значение энтальпии:

$$H_g = h_g + H^0$$

где  $H^0$  – отсчет энтальпии (постоянный тепловой эффект образования компонента при  $T_g=0$ ). Для задания этого отсчета используется оператор:

NOG = 0

Если теплоемкость фазы переменна, то в Решателе используются и энтальпия и теплоемкость [5]. Если на грани КО перепад температур мал, вместо энтальпии используется значение теплоемкости. Критерий такой «малости» задается оператором

DelTMIN = 1

#### Отсчеты температур и давления

В коде для организации отсчета температур фаз ( $T_g$ ,  $T_s$ ) при использовании *tuFORM* и расчета *энтальпий* вводится переменная  $T_0$  и предполагается, что:

$T = T_g + T_0$  - это температура G-фазы в градусах Кельвина,

$T = T_s + T_0$  - это температура S-фазы в градусах Кельвина.

Аналогично вводится отсчет давления  $p_0$ :

$p = P_f + p_0$  - абсолютное давление в Па.

Эти переменные задаются операторами

P0 = 0.0

T0 = 0.0

#### S-фаза. Предопределенные Ф-переменные

Для каждого материала S-фазы необходимо задать следующие свойства:

Dens - плотность, используется только в нестационарных задачах;

Cond - изотропная теплопроводность;

CP - теплоемкость.

При использовании неизотропной теплопроводности необходимо задать свойства для имен свойств, заданных в операторах NamDiffX/ NamDiffY/ NamDiffZ.

#### Ф-переменные пользователя

Для всех имен свойств, указанных в операторах NamDiffX/ NamDiffY/ NamDiffZ нужно задать свойства.

### 7.1.2 Многокомпонентная G-фаза. Предопределенные Ф-переменные

При использовании концентраций часть свойств задается для всей G-фазы, часть – для отдельных компонент.

Для отдельных компонент *всегда* задаются следующие свойства:

Ср.С1 .. Ср.С8 - теплоемкости компонент,

Н.С1 .. Н.С8 - тепловые составляющие энтальпии.

Аналогично задаются тепловые отсчеты энтальпий с помощью операторов

NOG.C1 = 0.0

...

NOG.C8 = 0.0

Если в секции [Special Data] указан параметр

C("PropConcModel") = byCONC/ byWILKE

то для расчета Dens, Visc, Cond, BetaT, BetaP смеси используются внутренние алгоритмы, описанные ниже. Если этот параметр не задан, то пользователь должен указать алгоритмы расчета перечисленных свойств.

#### Модель byCONC

Если выбрана эта модель, то свойства рассчитываются по следующим соотношениям:



$$\lambda_m = \sum_k r_k \lambda_k, \quad \mu_m = \sum_k r_k \mu_k, \quad \rho_m = \frac{1}{\sum_k \frac{c_k}{\rho_k}} = \sum_k r_k \rho_k, \quad \beta_T = \sum_k r_k \beta_{T,k}, \quad \beta_P = 0$$

$$r_i = \frac{c_i}{M_i} \frac{1}{\sum_k \frac{c_k}{M_k}},$$

Здесь  $M_i$  – молекулярный вес компонента,  $r_k$  – объемная (молярная) доля компонента. Для расчета необходимо в секции свойств (или в БД) описать свойства отдельных компонент (здесь XX = C1 .. C8):

Cond.XX, DinVisc.XX, Dens.XX, BetaT.XX, MolW.XX

Замечание. В данном случае необходимо задать динамическую вязкость (а не кинематическую!).

#### **Модель byWILKE**

В этой модели свойства рассчитываются для идеальной газовой смеси по соотношениям Уилки (Wilke, 1950) для вязкости и соотношениями Масона (Mason & Saxena, 1958) для теплопроводности:

$$\mu_m = \sum_i \mu_i r_i \frac{1}{\sum_k G_{ik} r_k}, \quad G_{ik} = \frac{1}{\sqrt{8 \left(1 + \frac{M_i}{M_k}\right)}} \left[ 1 + \sqrt{\left(\frac{\mu_i}{\mu_k}\right)} \left(\frac{M_k}{M_i}\right)^{0.25} \right]^2,$$

$$\lambda_m = \sum_i \lambda_i r_i \frac{1}{\sum_k G_{ik}^\lambda r_k}, \quad G_{ik}^\lambda = \frac{1.065}{\sqrt{8 \left(1 + \frac{M_i}{M_k}\right)}} \left[ 1 + \sqrt{\left(\frac{\mu_i}{\mu_k}\right)} \left(\frac{M_k}{M_i}\right)^{0.25} \right]^2, \quad G_{kk}^\lambda = 1$$

$$\rho_m = \frac{P_0}{R_0 T_g \sum_k \frac{c_k}{M_k}}, \quad \beta_P = \frac{1}{R_0 T_g \sum_k \frac{c_k}{M_k}}, \quad \beta_T = 0$$

Для расчета необходимо в секции свойств (или в БД) описать свойства отдельных компонент (здесь XX = C1 .. C8):

Cond.XX, DinVisc.XX, MolW.XX

Свойство BetaT для этой модели не используется. Если необходимо учесть эффекты плаву-чести используйте общую модель, основанную на плотности.

#### **Свойства для уравнений диффузии**

Для уравнений диффузии компонент C1 .. C8 необходимо определить либо коэффициент диффузии – свойство DDIF.XX, либо число Шмидта SC.XX.

### **7.1.3 Использование БД свойств**

В коде Anes предусмотрена возможность воспользоваться заготовками свойств для набора материалов G- и S- фаз. Эти заготовки хранятся в файле

<Anes>/property/pdb.a

который и является БД свойств. Для использования свойств из этого файла необходимо в секции в качестве первого оператора использовать оператор


IsUseDB = **.FALSE.** !.TRUE. / .FALSE.

В принципе можно смешивать использование БД и непосредственное задание. Алгоритм использования БД свойств следующий:

- 1) после обработки патчей А-компилятор создает список G- и S- материалов, необходимых для расчета;
- 2) свойства необходимых материалов «копируются» из БД в *начало* секции [Properties];
- 3) после этого производится обработка операторов Prop(), расположенных в секции (как из БД, так и непосредственно помещенных в секцию).

Структуру БД легко понять из куска текста файла dbp.a:

```
[F#AIR_IDG]
!#RusName= Воздух: идеальный газ
Dens = MyForm(3.48E-3*P0/(TG+T0))
Visc = MyForm(3.7228E-07*(TG+T0)**0.683/RhoG)
Cp = 1005.0
Cond = MyForm(2.45322E-04*(TG+T0)**0.82 )
BetaT = MyForm(1.0/(TG+T0))
.....
[S#AL_27]
!#RusName= Алюминий: постоянные свойства при T=27 C
Dens = 2700.0
Poisson = 0.35
Cp = 896.0
Cond = 204.0
BetaT = 2.35E-5
MYoung = 0.68e11
```

 **Замечание.** Свойства, описанные в БД, при компиляции файла проекта просто вставляются в секцию Properties. Поэтому при использовании формул для описания свойств можно использовать макропеременные, которые в проекте определит пользователь. В качестве примера рассмотрим расчет плотности для влажного воздуха (смесь двух идеальных газов – воздуха и пара):

$$\text{Dens} = \text{myFORM}( P0/(TG+T0)/8.314e3*(28.96-10.944*RVAP) )$$

Здесь RVAP – это макропеременная, определяющая объемную долю пара в воздухе (отношение парциального давления пара к атмосферному давлению). При использовании этого свойства из БД пользователь должен определить значение этой макропеременной.

## 7.2 Internal Source. Предопределенные источники

Эта секция описывает источники, автоматически рассчитываемые Решателем.

### 7.2.1 Силы плавучести в уравнениях движения

В векторном представлении эта сила имеет вид

$$\mathbf{S}_{ug} = \varphi \rho_g \mathbf{g}$$

В пакете предусмотрены три варианта учета этого источника:

- 1) источник автоматически не рассчитывается и определяется пользователем (значение ARX\_USER),
- 2) источник рассчитывается через плотность среды (ARX\_RHO),
- 3) источник рассчитывается по соотношениям Буссинеска (ARX\_T).

Во *втором варианте*, сила плавучести для *декартовой* системы рассчитывается по соотношениям

$$S_{U_{gx}}^{(I)} = \varphi g_{cx} (\rho_g - \rho_{arx}),$$

$$S_{U_{gy}}^{(I)} = \varphi g_{cy} (\rho_g - \rho_{arx}),$$

$$S_{U_{gz}}^{(I)} = \varphi g_{cz} (\rho_g - \rho_{arx}),$$

где  $g_{cx}, g_{cy}, g_{cz}$  - компоненты вектора силы тяжести в декартовой системе координат,

$\rho_{arx}$  - характерное значение плотности.

В цилиндрической системе координат

$$S_{U_{gx}}^{(I)} = \varphi g_x (\rho_g - \rho_{arx}), \quad g_x = g_{cx} \cos(y) + g_{cy} \sin(y)$$

$$S_{U_{gy}}^{(I)} = \varphi g_y (\rho_g - \rho_{arx}), \quad g_x = -g_{cx} \sin(y) + g_{cy} \cos(y)$$

$$S_{U_{gz}}^{(I)} = \varphi g_{cz} (\rho_g - \rho_{arx})$$

Напомним, что при учете сил плавучести связь между термодинамическим давлением и  $P_f$  приобретает вид

$$p = P_f + p_0 + \rho_{arx} (g_{cx}x + g_{cy}y + g_{cz}z) \quad \text{— для декартовой системы,}$$

$$p = P_f + p_0 - \rho_{arx} (g_{cx} \cos(y)x + g_{cy} \sin(y)x + g_{cz}z) \quad \text{— для цилиндрической системы.}$$

В *третьем* варианте используется приближение Буссинеска, которое для декартовой системы координат можно записать в виде

$$S_{ugx}^{(I)} = -\varphi g_{cx} \rho_g \beta_{Tg} (T_g - T_{arx}), \quad \beta_{Tg} = -\frac{1}{\rho_g} \left( \frac{\partial \rho_g}{\partial T_g} \right)_p,$$

$$S_{ugy}^{(I)} = -\varphi g_{cy} \rho_g \beta_{Tg} (T_g - T_{arx}), \quad S_{ugz}^{(I)} = -\varphi g_{cz} \rho_g \beta_{Tg} (T_g - T_{arx})$$

здесь  $T_{arx}$  - характерное значение температуры.

Для выбора и настройки сил плавучести используются операторы:

DensArx = 0 ! характерное значение плотности  
 Tarx = 0 ! характерное значение температуры  
 ArxForce = ARX\_User ! ARX\_User / ARX\_RHO / ARX\_T  
 AgravX = 0 ! декартовая X-компонента ускорения свободного падения  
 AgravY = 0  
 AgravZ = 0

## 7.2.2 Источники в уравнении энергии для TG

В текущей версии для уравнения энергии предусмотрено два внутренних источника: работа сил давления и вязкая диссипация.

Работа сил давления рассчитывается по соотношению

$$S_{Tg}^{(I)} = \varphi \left( \frac{\partial P_f}{\partial \tau} + \mathbf{U}_g \nabla P_f \right),$$

если включен флаг SrcTG\_PF данной секции:

IsSrcTG\_PF = .FALSE. ! .TRUE. / .FALSE.

Вязкая диссипация вычисляется по формуле

$$S_{Tg}^{(I)} = \varphi \mu_{eff} G,$$

где  $G$  – диссипативная функция. Для декартовой системы координат эта функция имеет вид:

$$G=2\left[\left(\frac{\partial U_{gx}}{\partial x}\right)^2+\left(\frac{\partial U_{gy}}{\partial y}\right)^2+\left(\frac{\partial U_{gz}}{\partial z}\right)^2\right]+\left(\frac{\partial U_{gx}}{\partial y}+\frac{\partial U_{gy}}{\partial x}\right)^2+\left(\frac{\partial U_{gx}}{\partial z}+\frac{\partial U_{gz}}{\partial x}\right)^2+\left(\frac{\partial U_{gz}}{\partial y}+\frac{\partial U_{gy}}{\partial z}\right)^2$$

Для активизации вычисления вязкой диссипации используется оператор

IsSrcTG\_NU = *.FALSE.* !.TRUE. / *.FALSE.*

### 7.2.3 Стабилизированные и развитые течения

Операторы

DEV\_MODE = *devNONE* / devMASSRATE / devDPDZ  
 DEV\_MassRateZ = V-переменная  
 DEV\_DPDZ = *0.0*  
 DEV\_TG\_MODE = *devTG\_NONE* / devTG\_CONST / devTG\_FLUX

используются при моделировании *полностью развитого* течения в канале или трубе и стабилизированного течения с периодическими граничными условиями.

Режим полностью развитого течения активизируется, если используются структурные сетки, NZ = 1 и решается уравнение для компоненты UgZ.

Если DEV\_MODE = devDPDZ, то аксиальный градиент давления пользователь должен задать через оператор DEV\_DPDZ, в противном случае градиент давления рассчитывается автоматически. В этом случае пользователь должен задать *полный расход* через канал

$$G_0 = \iint_{xy} \rho_g \phi U_{gz} dx dy$$

используя оператор DEV\_MassRateZ. Сам полный расход задается как V-переменная.

При использовании для этой переменной виртуальной функции, функция вызывается только один раз для КО с

lxC=2, lyc=2, lzc=1 или idCV = 1

Режим моделирования теплового состояния определяется оператором DEV\_TG\_MODE. Подробности моделей стабилизированных течений изложены в документе [5].

### 7.3 Turbulence. Настройка моделей турбулентности

В текущей версии пакета предусмотрено пять моделей турбулентности:

- trVariant - алгебраические модели турбулентности,
- trKEWall - k-ε модель турбулентности с пристенными функциями,
- trKE2Layer - двухслойная k-ε модель турбулентности для малых турбулентных чисел Рейнольдса.
- trKELOW - k-ε модель турбулентности Лаундера-Шарма для малых турбулентных чисел Рейнольдса.
- trKOM - k-ω модель Вилхокса с универсальными пристенными функциями,
- trSST - k-ω SST-модель Ментера с универсальными пристенными функциями.

Основная настройка модели турбулентности осуществляется автоматически макрооператором ModelTur в секции [Macro Sub], которая и формирует содержимое этой секции.

### 7.3.1 Алгебраическая модель

Для *алгебраической модели* пользователь задает распределение турбулентной кинематической вязкости с помощью V-переменной

```
NuTVAR = 0.0 / V-переменная
```

В коде реализована встроенная алгебраическая LVEL модель турбулентности [3]. Для использования этой модели нужно использовать следующий вариант V-переменной

```
NuTVAR = Internal(trLVEL)
```

### 7.3.2 К-ε и k-ω модели турбулентности

Настройка этих моделей в данной секции сводится к организации начала итерационного процесса и выбора алгоритма линеаризации источниковых членов.

В пакете предусмотрен следующий алгоритм начала итерационного процесса в случае решения стационарных задач:

Если в операторе

```
SweepBeforeKE = NoXX
```

задано ненулевое значение, то NoXX итераций уравнения для k и ε решаются, однако турбулентная вязкость рассчитывается не средствами самой модели, а с помощью алгебраической модели, определенной переменной NuTVAR.

Начиная с NoXX-итерации турбулентная вязкость рассчитывается по собственной модели. Значение SweepBeforeKE игнорируется (Решатель устанавливает значение SweepBeforeKE=0), если расчет начинается с контрольной точки и в файле рестарта уже сохранены турбулентные поля. Если сохранение произошло до включения решения турбулентных уравнений, то необходимо запретить чтение полей из файла рестарта с помощью операторов секции [PHI Variables]

```
Restored("kTUR") = .false.  
Restored("epsTUR") = .false.
```

В этом случае Решатель не будет занулять значение SweepBeforeKE.

Для выбора модели линеаризации источниковых членов [3] используется оператор

```
LinearKEPS = lkeDefault ! lkeDefault / lkeEQU/ lkeLENM
```

При использовании модели с пристенными функциями в пакете предусмотрены две модели пристенных функций [3]. Выбор модели осуществляется оператором

```
TKEFun = Internal(trwLAUND/trwDBS)
```

Для задания турбулентного числа Прандтля для предопределенных Φ-переменных TG и C1 .. C8 и Φ-переменных пользователя используется оператор

```
SigmaTUR("Φ-имя") = 1.0 / Variant
```

## 7.4 Boundary Condition. Задание граничных условий

Эта секция используется для описания граничных условий. Для Φ-переменных на границе расчетной области задается значение нормальной к граничной поверхности компо-

ненты вектора плотности полного потока -  $J_{\Phi,b}$  (напомним, что нормаль к границе направлена *в сторону* РО). При этом допускается две формы представления [3]:

$$J_{\Phi,b} = \max(m_b, 0) \cdot V_{\Phi,b} - \max(-m_b, 0) \cdot \Phi_p + C_{\Phi,b} (V_{\Phi,b} - \Phi_w),$$

$$J_{\Phi,b} = \max(m_b, 0) \cdot V_{\Phi,b} - \max(-m_b, 0) \cdot \Phi_p + |C_{\Phi,b}| (V_{\Phi,b} - \Phi_p)$$

где

$C_{\Phi,b}$  - *коэффициент* граничного условия,

$\Phi_{\Phi,b}$  - *значение* граничного условия.

В этих выражениях в конвективной составляющей потока используется значение  $\Phi$ -переменной в узловой точке  $P$  *приграничного КО G-фазы*, а в диффузионной составляющей:

- значение функции  $\Phi$  на границе РО ( $\Phi_w$ ) для первой формы,
- значение функции  $\Phi$  в узловой точке  $P$  ( $\Phi_p$ ) для второй формы (эта форма идентична форме задания ГУ в виде поверхностного источникового члена).

В пакете выбор формы задания ГУ определяется *знаком*  $C_{\Phi,b}$  (напомним, что с точки зрения *физики* коэффициент  $C_{\Phi}$  всегда положителен или равен нулю). Если  $C_{\Phi,b} > 0$  – используется первая форма, если он *отрицателен* – вторая.

Для давления  $P_f$  используется другая форма для плотности потока массы:

$$m_b = C_p (V_p - P_{f,p})$$

Значение коэффициента  $C_{\Phi,b}$  ( $C_p$ ) и значения  $V_{\Phi,b}$  ( $V_p$ ) задаются с помощью операторов:

COEF("ИмяПатча.Ф-имя") = V-переменная

VAL("ИмяПатча.Ф-имя") = V-переменная

В качестве патчей, к которым «привязываются» ГУ используются патчи типа Block (BlockWall) и BS (BSWall).

Допускается задание «специальных» типов ГУ, путем задания символьных значений коэффициенту COEF:

- |             |  |
|-------------|--|
| BC_FIXVAL   | -граничное условие первого рода, переменная $V_{\Phi,b}$ задает значение $\Phi$ на границе ( $\Phi_w = V_{\Phi,b}$ ),  |
| BC_FIXFLUX  | -граничное условие второго рода, переменная $V_{\Phi,b}$ задает <i>полную</i> плотность нормального потока $\Phi$ на границе $J_{\Phi,b} = V_{\Phi,b}$ ,   |
| BC_MASSONLY | -используется для входных и выходных границ, задается только конвективная составляющая полного потока, $V_{\Phi,b}$ - это значение $\Phi$ , вносимое в расчетную область потоком (для выходной границы $V_{\Phi,b}$ не используется и может быть любое), |
| BC_FIXMASS  | -используется только для давления $P_f$ ; в этом случае $m_b = V_p$ (на границе задается нормальная компонента плотности потока массы),  |
| BC_FIXPRESS | -используется только для давления $P_f$ , в этом случае $V_p$ задает давление на границе (значение $P_{f,w}$ ),  |
| BC_BYMASS   | - это условие типа BC_MASSONLY в котором значение ГУ $V_{\Phi,b}$ пересчитывается по формуле   |

$$V_{\Phi,b} = V_{\Phi,b} \frac{m_b}{\rho_{g,p}}$$

(напомним, что  $m_b > 0$ , если поток «втекает» в расчетную область),

Если для  $\Phi$ -переменной не задано явно граничное условие для существующего Block/BC патча, то подразумевается:

$$C_{\Phi,b} = 0 \quad , \quad V_{\Phi,b} = 0$$

Если для компоненты скорости G-фазы не заданы граничные условия, то для граничных патчей с типом Wall (BSWall и BlockWall) компилятором автоматически генерируются условия прилипания:

$$C_{\Phi,b} = BC\_FIXVAL \quad , \quad V_{\Phi,b} = 0$$

Для других  $\Phi$ -переменных этот подход можно активизировать с помощью оператора WallFixNull( $\Phi$ -имя).

Наряду с «специальными» значениями коэффициента существует одно «специальное» значение коэффициента:

$$V_{\Phi,b} = BC\_SAME$$

В этом случае значение берется из самого приграничного КО с *предыдущей* итерации:

$$V_{\Phi,b} = \Phi_p^*$$

Граничные условия для уравнения энергии ( $\Phi$ -переменная "TG") задаются в более сложной форме (см. раздел 4.5.4 документа [3]):

$$J_{Tg,b} = \sum_k \max(J_{Ck,b}, 0) \cdot H_{k,0} - \sum_k \max(-J_{Ck,b}, 0) \cdot H_{k,P} + C_T (T_0 - T_w) + \sum_k \max(J_{Ck,b}, 0) \cdot \Delta H_{k,0},$$

$$J_{Tg,b} = \sum_k \max(J_{Ck,b}, 0) \cdot H_{k,0} - \sum_k \max(-J_{Ck,b}, 0) \cdot H_{k,P} + |C_T| (T_0 - T_p) + \sum_k \max(J_{Ck,b}, 0) \cdot \Delta H_{k,0},$$

Для описания коэффициента  $C_T$  и значения  $T_0$  используются стандартные операторы описания ГУ. Для задания теплоты «фазового перехода» на границе  $\Delta H_{k,0}$  используются операторы

$$DH(\text{"ИмяПатча.TG"}) = 0.0$$

$$DH.C1(\text{"ИмяПатча.TG"}) = 0.0$$

....

$$DH.C8(\text{"ИмяПатча.TG"}) = 0.0$$

Важное замечание: Эти операторы должны быть описаны в секции *после* соответствующих операторов описания самого граничного условия.

## 7.5 Sources. Задание источниковых членов пользователя

Эта секция используется для описания источников для  $\Phi$ -переменных:

$$S_{\Phi} = \max(M_p, 0) \cdot V_{\Phi} - \max(-M_p, 0) \cdot \Phi + C_{\Phi} (V_{\Phi} - \Phi)$$

$$M_p = C_p (V_p - P)$$

$$S_{Tg} = \sum \langle S_{Ck} \rangle H_{k,0} - \sum \langle -S_{Ck} \rangle H_k + C_T (T_0 - T_g) + \sum \langle S_{Ck} \rangle \Delta H_{0k},$$

вид которых аналогичен поверхностным источникам ГУ.

Источниковые члены должны быть привязаны к патчам типа Volume, Surface или Point. Напомним, что именно типом патча определяется тип источника (объемный, поверхностный или точечный) и его размерность.

Значение коэффициента  $C_{\Phi}$  ( $C_p$ ) и значения  $\Phi_{\Phi}$  ( $V_p$ ) задаются с помощью V-переменных:

$$COEF(\text{"ИмяПатча.}\Phi\text{-имя"}) = V\text{-переменная}$$

$$VAL(\text{"ИмяПатча.}\Phi\text{-имя"}) = V\text{-переменная}$$

Для задания теплот «фазового перехода» на границе  $\Delta H_{k,0}$  используются операторы  $DH(\text{"ИмяПатча.TG"}) = 0.0$

DN.C1("ИмяПатча.TG")= 0.0

....

DN.C8("ИмяПатча.TG")= 0.0

Эти операторы должны быть описаны в секции *после* соответствующих операторов описания самого источника.

Как и для ГУ, для источников допускаются специальные частные формы:

SR\_FIXVAL -источник задает фиксированное значение  $\Phi$ -переменной  $\Phi = V_\Phi$ ,  
 SR\_FIXSOURCE -задается непосредственно **полное** значение источника  $S_\Phi = V_\Phi$ ,  
 SR\_MASSONLY -используется для источников с массовой составляющей, задается только конвективная составляющая источника,  $V_\Phi$  - это значение  $\Phi$ , вносимое в расчетную область вдуваемой массой,  
 SR\_FIXMASS -используется только для  $P_f$ , в этом случае задается сам массовый источник  $M_P = V_P$ ,  
 SR\_FIXPRESS -используется только для  $P_f$ , это условие задает постоянное значение давления в области источника  $P_f = V_P$ ,  
 SR\_BYMASS - это условие типа SR\_MASSONLY в котором значение  $V_\Phi$  пересчитывается по формуле

$$V_\Phi = V_\Phi \frac{M_P}{\rho_g},$$

## 7.6 Porous Model

Эта секция используется в случае, когда имеются патчи типа Porous. В патче указывается символьная ссылка "ИмяFS-зоны" на модель межфазного взаимодействия. Модель межфазного взаимодействия - это «встроенная» или ваша собственная подпрограмма для расчета характеристик межфазного взаимодействия [3]. Заметим, что одна и та же модель (например, модель стержней в треугольной упаковке) может использоваться для описания различных зон (например, с разными диаметрами стержней). Для настройки модели на конкретные условия используются набор параметров модели, значения которых устанавливает пользователь.

В коде Anes реализовано две принципиально разные пористые модели:

- 1) обычная пористая модель, в которой S-фаза описывается так же, как и G-фаза; в S-фазе рассчитывается поле предопределенной  $\Phi$ -переменной типа TS;
- 2) двухуровневая пористая модель, в которой нет на уровне сетки КО расчетной области S-фазы, а есть подсеточная модель, которая используется для расчета аналога температуры фазы TS.

В данном документе не рассматривается организация подпрограмм моделей как FS-взаимодействия, так подсеточных моделей. Об этом подробно написано в [3].

В секции для каждой FS-зоны должны быть указаны следующие операторы:

InterFace("ИмяFS-зоны") = Internal(ИмяFSМодели) ! Имя подпрограммы вычисления свойств  
 = Virtual(Имя функции) ! Виртуальная подпрограмма в A-файле  
 = External(Имя функции) ! Функция в f90-файле  
 MassMode("ИмяFS-зоны") = msNONE ! Режим массообмена: msNONE /msDEFAULT/msChemical  
 Param("ИмяFS-зоны.ИмяПараметра") = 0 ! Параметры-постоянные модели

Операторы Param() задают значение параметров модели, число и смысл которых зависит от FS-модели. Например, для модели POR\_BALL (сферическая засыпка):

Param("ИмяFS-зоны.POROUS") = 0.5 ! пористость  
 Param("ИмяFS-зоны.DS") = 5.e-6 ! диаметр сферы  
 Param("ИмяFS-зоны.СК1") = 150 ! коэффициент СК1 в формуле трения  
 Param("ИмяFS-зоны.СК2") = 1.75 ! коэффициент СК2 в формуле трения  
 Param("ИмяFS-зоны.NUS") = 2



При использовании двухуровневой пористой модели для ее описания используются дополнительные операторы:

```
L2Model("ИмяFS-зоны") = "Имя модели"  
Param("ИмяFS-зоны.ИмяПараметра") = 0
```

**! Параметры-постоянные модели**

## 8. ПЕРЕМЕННЫЕ ПОЛЬЗОВАТЕЛЯ

Операторы этих секций используются для создания программного интерфейса пользователя с использованием языка Фортран. Более подробно этот интерфейс описан в документе [4].

### 8.1 *User Event. Описание событий пользователя*

Эта секция используется для описания виртуальных функций-событий пользователя. Секция состоит из операторов описания 9 Event-переменных, представляющих собой разновидность Virtual функции:

```
OnInput    = nil / EventValue
OnInit     = nil / EventValue
BeforeSweep = nil / EventValue
AfterSweep  = nil / EventValue
BeforeTime = nil / EventValue
AfterTime  = nil / EventValue
OnReport   = nil / EventValue
OnLESolver = nil / SubValue
OnStop     = nil / SubValue
OnResume   = nil / SubValue
```

где

```
EventValue = Virtual(<Имя виртуальной функции>)
            = External(<Имя функции>)
```

### 8.2 *User Fortran Variables. Переменные пользователя*

Эта секция не является секцией с AIL операторами. В этой секции пользователь может записать операторы описания *своих переменных Фортрана*, которые будут доступны в любых виртуальных функциях и подпрограммах пользователя.

### 8.3 *User Fortran fields. Массивы пользователя*

Эта секция используется, если пользователю необходимо в своих виртуальных функциях использовать структурные или неструктурные массивы, связанные с КО. Для создания таких массивов используются AIL операторы:

```
ArrCells(Name1, Name2, ...)
```

При использовании структурной сетки будут созданы трехмерные массивы  
<Name>(1:NX,1:NY,1:NZ)

Для неструктурной сетки будут созданы одномерные массивы  
<Name>(1:NoCells)

### 8.4 *User Data. Передача исходных данных пользователя*

Эта секция используется Решателем для передачи данных целого, действительного и символьного типов в подпрограммы пользователя.

В Решателе имеются три массива, доступные во всех подпрограммах пользователя

```
IUDat(1 .. 64)   - массив типа integer,
RUDat(1 .. 64)   - массив типа real(4),
CUDat(1 .. 64)   - массив типа character(32).
```

По умолчанию эти массивы инициализируются нулями и пробелами. Пользователь может присвоить любые значения элементам этих массивов с помощью операторов:

```
IUDat(1 .. 64) = 0    ! IntegerValue / <Макро-выражение>
RUDat(1 .. 64) = 0.0 ! RealValue / <Макро-выражение>
CUDat(1 .. 64) = " " ! "<Текст>"
```

Ненулевые значения этих массивов распечатываются в листинге Решателя.

### **8.5 *User Fortran Subroutines. Произвольные подпрограммы***

Эта секция не является секцией с APL операторами. В этой секции пользователь может записать любые свои подпрограммы и функции, которые необходимы для создания виртуальных функций.

## Литература

1. Код Anes20хе. «Настройка и работа с кодом», версия 2.24, 2019.
2. Код Anes20хе. «Дизайнер проекта. Программа - препроцессор», версия 2.24, 2019.
3. Код Anes20хе. «Описание математических моделей кода», версия 2.24, 2019.
4. Код Anes20хе. «Фортран-интерфейс кода», версия 2.24, 2019.
5. Код Anes20хе. «Описание численных алгоритмов кода», версия 2.24, 2019.
6. Код Anes20хе. «Визуализация результатов расчетов. Программы - постпроцессоры», версия 2.24, 2019.