

**Anes Team**



***ANES/20XE : Код для численного моделирования процессов гидродинамики и теплообмена***

Версия 2.24

Описание математических моделей кода

Москва 2019 г.

---

## Оглавление

<b>1. ВВЕДЕНИЕ .....</b>	<b>8</b>
<b>1.1 Расчетная область, фазы и Ф-переменные .....</b>	<b>8</b>
1.1.1 Фазы .....	8
1.1.2 Межфазное взаимодействие .....	9
1.1.3 Ф-переменные .....	9
<b>1.2 Размерности переменных .....</b>	<b>10</b>
<b>1.3 V-переменные Anes .....</b>	<b>10</b>
<b>1.4 Файл проекта Anes .....</b>	<b>10</b>
<b>1.5 Сетки контрольных объемов .....</b>	<b>11</b>
<b>2. ОПИСАНИЕ РАСЧЕТНОЙ ОБЛАСТИ .....</b>	<b>12</b>
<b>2.1 Системы координат .....</b>	<b>13</b>
<b>2.2 Патчи .....</b>	<b>14</b>
2.2.1 Описание геометрии патча .....	14
2.2.2 Тип патча .....	16
<b>2.3 Патчи для описания расчетной области .....</b>	<b>17</b>
<b>2.4 Патчи для описания распределения G- и S-фаз .....</b>	<b>18</b>
<b>2.5 Инверсный алгоритм описания РО .....</b>	<b>20</b>
<b>2.6 Патчи для описания источниковых членов .....</b>	<b>20</b>
<b>3. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ОБОБЩЕННОЙ Ф-ПЕРЕМЕННОЙ .....</b>	<b>22</b>
<b>3.1 Уравнение баланса массы .....</b>	<b>22</b>
<b>3.2 Уравнение баланса Ф-переменной .....</b>	<b>23</b>
<b>3.3 Коэффициент диффузионного переноса .....</b>	<b>23</b>
<b>3.4 Источниковые члены .....</b>	<b>24</b>
3.4.1 Размерность источников пользователя .....	25
3.4.2 Источниковые члены для уравнения неразрывности .....	25
3.4.3 Источниковые члены для Ф-переменной .....	25
<b>3.5 Граничные условия .....</b>	<b>26</b>
3.5.1 Граничные условия для давления .....	27
3.5.2 Граничные условия для Ф-переменной .....	28
3.5.3 Граничное условие первого рода .....	28
3.5.4 Граничное условие второго рода .....	29
3.5.5 Граничные условия на входных границах .....	29
3.5.6 Граничные условия на выходных границах .....	30
3.5.7 Граничные условия по умолчанию .....	30
3.5.8 Периодические граничные условия .....	31

---

<b>3.6</b>	<b>Начальные условия</b> .....	<b>32</b>
<b>4.</b>	<b>МАТЕМАТИЧЕСКИЕ МОДЕЛИ G-ФАЗЫ</b> .....	<b>33</b>
<b>4.1</b>	<b>Уравнение неразрывности</b> .....	<b>34</b>
4.1.1	Граничные условия для давления .....	35
<b>4.2</b>	<b>Уравнения движения</b> .....	<b>35</b>
4.2.1	Источники, связанные с цилиндрической системой и тензором напряжений .....	36
4.2.2	Источники, связанные с силой плавучести .....	37
4.2.3	Источники, связанные с межфазным трением в Rogous зонах.....	38
4.2.4	Межфазное трение на границах раздела G- и S-фаз.....	38
4.2.5	Граничные условия .....	38
4.2.6	Моделирование стабилизированного течения .....	38
4.2.7	Развитое периодическое течение.....	40
4.2.8	Моделирование сжимаемых течений.....	41
<b>4.3</b>	<b>Уравнение энергии для однокомпонентной G-фазы</b> .....	<b>43</b>
4.3.1	Энтальпия G-фазы .....	43
4.3.2	Описание источников пользователя.....	45
4.3.3	Работа сил давления и вязкая диссипация .....	45
4.3.4	Уравнение энергии, записанное через энтальпию .....	46
4.3.5	Межфазный теплообмен в Rogous зонах .....	47
4.3.6	Межфазный теплообмен на границах раздела G- и S-фаз .....	47
4.3.7	Граничные условия .....	47
4.3.8	Расчет потоков .....	48
4.3.9	Уравнение энергии для стабилизированного течения .....	48
4.3.10	Уравнение энергии для развитого периодического течения.....	50
4.3.11	Уравнение энергии для сложного периодического течения .....	51
<b>4.4</b>	<b>Уравнения для концентраций</b> .....	<b>52</b>
4.4.1	Источники и граничные условия .....	52
4.4.2	Баро и термодиффузия .....	53
<b>4.5</b>	<b>Уравнение энергии для многокомпонентной G-фазы</b> .....	<b>53</b>
4.5.1	Описание свойств многокомпонентной G-фазы .....	54
4.5.2	Источниковые члены, задаваемые пользователем .....	55
4.5.3	Описание химических реакций.....	55
4.5.4	Описание граничных условий .....	56
4.5.5	Межфазный теплообмен.....	56
4.5.6	Расчет потоков .....	57
<b>4.6</b>	<b>Свойства G-фазы</b> .....	<b>57</b>
4.6.1	Особенности задания свойств сжимаемой G-фазы .....	58
<b>5.</b>	<b>МАТЕМАТИЧЕСКАЯ МОДЕЛЬ S-ФАЗЫ</b> .....	<b>59</b>
<b>5.1</b>	<b>Уравнение энергии S-фазы</b> .....	<b>59</b>
5.1.1	Межфазный теплообмен в зонах Rogous .....	59
5.1.2	Межфазный теплообмен на границах раздела фаз G-S фаз.....	61
5.1.3	Описание источников и граничных условий.....	61
<b>5.2</b>	<b>Плавление S-фазы</b> .....	<b>61</b>
5.2.1	Активация модели плавления.....	62
<b>6.</b>	<b>МАТЕМАТИЧЕСКИЕ МОДЕЛИ <math>\Phi</math>-ПЕРЕМЕННЫХ ПОЛЬЗОВАТЕЛЯ</b> .....	<b>63</b>

<b>6.1</b>	<b>Flow <math>\Phi</math>-переменная</b> .....	<b>63</b>
6.1.1	Источниковые члены и граничные условия .....	63
6.1.2	Межфазный теплообмен в зонах Porous .....	63
6.1.3	Межфазный теплообмен на границах раздела G-S фаз .....	63
<b>6.2</b>	<b>Struct <math>\Phi</math>-переменная</b> .....	<b>64</b>
6.2.1	Источниковые члены и граничные условия .....	64
6.2.2	Межфазный теплообмен в зонах Porous .....	64
<b>6.3</b>	<b>Space <math>\Phi</math>-переменная</b> .....	<b>64</b>
<b>7.</b>	<b>ДВУХФАЗНЫЕ МОДЕЛИ G-ФАЗЫ</b> .....	<b>66</b>
<b>7.1</b>	<b>VOF модель двухфазного потока</b> .....	<b>66</b>
<b>7.2</b>	<b>LDP модель двухфазного потока</b> .....	<b>66</b>
7.2.1	Начальные условия для частиц .....	69
7.2.2	Распределение частиц группы по размерам .....	70
7.2.3	Нестационарные задачи .....	72
7.2.4	Уравнение движения частицы .....	72
7.2.5	Численный алгоритм решения уравнений движения .....	73
7.2.6	Модель турбулентной дисперсии частиц .....	74
7.2.7	Взаимодействие частиц с границей расчетной области .....	76
7.2.8	Анализ результата моделирования .....	78
7.2.9	Интеграция LDP модели с другими физическими моделями .....	78
7.2.10	Примеры модели .....	80
<b>8.</b>	<b>МОДЕЛИ МЕЖФАЗНОГО ВЗАИМОДЕЙСТВИЯ</b> .....	<b>81</b>
<b>8.1</b>	<b>Модели массообмена в пористых зонах</b> .....	<b>81</b>
8.1.1	Модель массообмена msDefault .....	81
8.1.2	Модель массообмена msChemical .....	82
<b>8.2</b>	<b>Межфазное трение в пористых зонах</b> .....	<b>82</b>
8.2.1	Проницаемость системы плоских микроканалов .....	83
8.2.2	Проницаемость шаровой засыпки .....	84
<b>8.3</b>	<b>Межфазный теплообмен в пористых зонах</b> .....	<b>84</b>
<b>8.4</b>	<b>Межфазное взаимодействие для остальных <math>\Phi</math>-переменных</b> .....	<b>85</b>
8.4.1	K-eps модель турбулентности .....	85
8.4.2	Произвольные $\Phi$ -переменные .....	85
<b>8.5</b>	<b>Организация модели пористого межфазного взаимодействия в Решателе</b> .....	<b>85</b>
<b>8.6</b>	<b>Модель пористого взаимодействия POR BALL</b> .....	<b>86</b>
8.6.1	Геометрические характеристики .....	86
8.6.2	Проницаемости .....	87
8.6.3	Коэффициент межфазной теплоотдачи .....	87
8.6.4	Список параметров модели .....	87
<b>8.7</b>	<b>Модель пористого взаимодействия PLAIN PINS</b> .....	<b>87</b>
8.7.1	Геометрические характеристики .....	88
8.7.2	Проницаемости в поперечном сечении (x-y) .....	88
8.7.3	Проницаемость в аксиальном z-направлении .....	88
8.7.4	Коэффициент межфазной теплоотдачи .....	88
8.7.5	Список параметров модели .....	89

<b>8.8</b>	<b>Модель пористого взаимодействия FINE_PINS .....</b>	<b>89</b>
8.8.1	Геометрические характеристики .....	90
8.8.2	Проницаемости по осям координат .....	90
8.8.3	Коэффициент межфазной теплоотдачи .....	91
8.8.4	Список параметров модели .....	92
<b>8.9</b>	<b>Модель пористого взаимодействия PERF_PLATE .....</b>	<b>92</b>
8.9.1	Геометрические характеристики .....	92
8.9.2	Проницаемости по осям координат .....	93
8.9.3	Коэффициент межфазной теплоотдачи .....	93
8.9.4	Список параметров модели .....	93
<b>9.</b>	<b>МОДЕЛИ ТУРБУЛЕНТНОСТИ.....</b>	<b>94</b>
<b>9.1</b>	<b>Турбулентные соотношения для пристенной области .....</b>	<b>95</b>
9.1.1	Логарифмический профиль скорости .....	95
9.1.2	Логарифмический профиль скорости и двухпараметрические модели турбулентности ..	97
9.1.3	Модифицированный логарифмический профиль скорости .....	98
9.1.4	Логарифмический профиль для температуры .....	98
<b>9.2</b>	<b>Модель пристенных функций .....</b>	<b>100</b>
9.2.1	Пристенные функции для скорости .....	100
9.2.2	Пристенные функции для уравнения энергии .....	101
9.2.3	Пристенные функции для произвольных $\Phi$ -переменных.....	101
<b>9.3</b>	<b>Алгебраические модели турбулентности .....</b>	<b>102</b>
9.3.1	Переменная $L_{dis}$ и $\Phi$ -переменная $W_{dis}$ .....	102
9.3.2	LVEL модель турбулентности.....	103
<b>9.4</b>	<b>Двухпараметрические модели турбулентности .....</b>	<b>104</b>
<b>9.5</b>	<b>Модель <math>k-\varepsilon</math> с пристенными функциями.....</b>	<b>104</b>
9.5.1	Граничные условия на твердой стенке.....	105
9.5.2	Входные граничные условия .....	107
9.5.3	Расчет коэффициента турбулентной вязкости .....	107
<b>9.6</b>	<b>Порождение турбулентности силами плавучести <math>G_b</math>.....</b>	<b>108</b>
<b>9.7</b>	<b>Двухслойная <math>k-\varepsilon</math> модель .....</b>	<b>109</b>
9.7.1	Модель tr2LREY.....	109
9.7.2	Модель tr2LWOLF .....	110
9.7.3	Модель tr2LVIA .....	110
9.7.4	Модель tr2LEXP.....	110
9.7.5	Особенности использования двухслойной модели .....	110
<b>9.8</b>	<b>Модель турбулентности <math>k-\omega</math> .....</b>	<b>111</b>
9.8.1	Универсальные пристенные функции для компонент вектора скорости .....	112
9.8.2	Универсальные пристенные функции для температуры .....	113
9.8.3	Граничное условие для $\omega$ на твердой стенке .....	113
9.8.4	Граничное условие для $k$ .....	114
<b>9.9</b>	<b>SST модель Ментера .....</b>	<b>114</b>
9.9.1	Активация моделей в Anes .....	115
<b>9.10</b>	<b>Двухпараметрическая <math>k-\varepsilon</math> модель Лаундера-Шарма .....</b>	<b>115</b>
<b>9.11</b>	<b>Особенности алгоритма решения двухпараметрических уравнений турбулентности.....</b>	<b>116</b>
9.11.1	Линеаризация источниковых членов .....	116
9.11.2	Двухшаговый итерационный процесс.....	117

---

<b>9.12</b>	<b>Экспериментальные k-ε модели.....</b>	<b>118</b>
9.12.1	Модель с F-точкой для неструктурных сеток.....	118
9.12.2	Гибридная модель .....	119
<b>10.</b>	<b>ДОПОЛНИТЕЛЬНЫЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ.....</b>	<b>120</b>
<b>10.1</b>	<b>S2S модель радиационного переноса.....</b>	<b>120</b>
10.1.1	Основные соотношения S2S модели.....	120
<b>10.2</b>	<b>Реализация S2S модели в коде Anes .....</b>	<b>122</b>
<b>10.3</b>	<b>Утилита CalcFView .....</b>	<b>125</b>
<b>10.4</b>	<b>Пример использования S2S модели.....</b>	<b>127</b>

## Обозначения

Описание моделей пакета

- это основной текст документа.

① Описание в проекте прикладной задачи :  
Использование Ф-переменной в задаче определяется оператором

- эта вставка используется для ссылки на операторы файла проекта.

## 1. Введение

Компьютерный CFD код Anes (версия 20хе) предназначен для моделирования стационарных и нестационарных одно-, двух- и трехмерных течений однофазной или двухфазной многокомпонентной сплошной среды в областях сложной геометрической формы, содержащих внутри твердые проницаемые или непроницаемые объекты. В составе кода имеется набор математических моделей и численных алгоритмов для решения широкого класса задач гидродинамики и тепломассообмена.

В данном документе приводятся описания этих моделей. Подробное описание реализованных численных алгоритмов кода представлено в отдельном документе [1]. В данном разделе содержатся лишь те сведения из [1], которые необходимы для понимания особенностей описания математических моделей.

### 1.1 Расчетная область, фазы и $\Phi$ -переменные

В коде Anes поля зависимых переменных прикладной задачи (давление, компоненты вектора скорости, температура и т.д.) определяются в ограниченной области пространства, которая далее будет называться *расчетной областью* (РО). Расчетная область в общем случае представляет собой произвольный трехмерный объем пространства, ограниченный замкнутой поверхностью - *границей* РО. Отметим, что граница РО играет важную роль в постановке прикладной задачи - для замыкания постановки прикладной задачи необходимо на границе РО задать для всех зависимых переменных граничные условия (ГУ).

Принципы построения расчетной области в коде Anes подробно описаны ниже в главе 2.

#### 1.1.1 Фазы

В общем случае в РО могут присутствовать две фазы:

- *S - фаза*: твердые проницаемые или непроницаемые *неподвижные* конструкции, в общем случае с анизотропными и неоднородными свойствами;
- *G - фаза*: сплошная среда - жидкость или газ, с произвольными теплофизическими свойствами.

Распределение фаз по пространству описывается объемной долей G-фазы, называемой также пористостью

$$\varphi = \frac{V_G}{V_G + V_S} \quad (1.1)$$

где  $V_G$ ,  $V_S$  - элементарные объемы, занятые G- и S- фазами. В общем случае РО состоит из набора областей трех типов:

- 1) Flow- область, свободная от твердых объектов и содержащая только жидкость ( $\varphi = 1$ ),
- 2) Struct- область, содержащая непроницаемые конструкции ( $\varphi = 0$ ),
- 3) Porous- область, содержащая проницаемые конструкции ( $0 < \varphi < 1$ ), в которых одновременно присутствуют G- и S- фазы.

Заметим, что в расчетной области одновременно могут присутствовать S- фаза, состоящая из различных материалов и различные G-фазы.

Типичный пример расчетной области показан на рисунке 1.1, где показана РО для моделирования металлгидридного аккумулятора водорода. В этой модели присутствуют два материала G-фазы (водород и охлаждающая вода), несколько материалов непроницаемой S-фазы (сталь, алюминий и медь) и пористые области в которых присутствуют и G-фаза (водород) и S-фаза (пористая засыпка металлгидридного сплава Me).



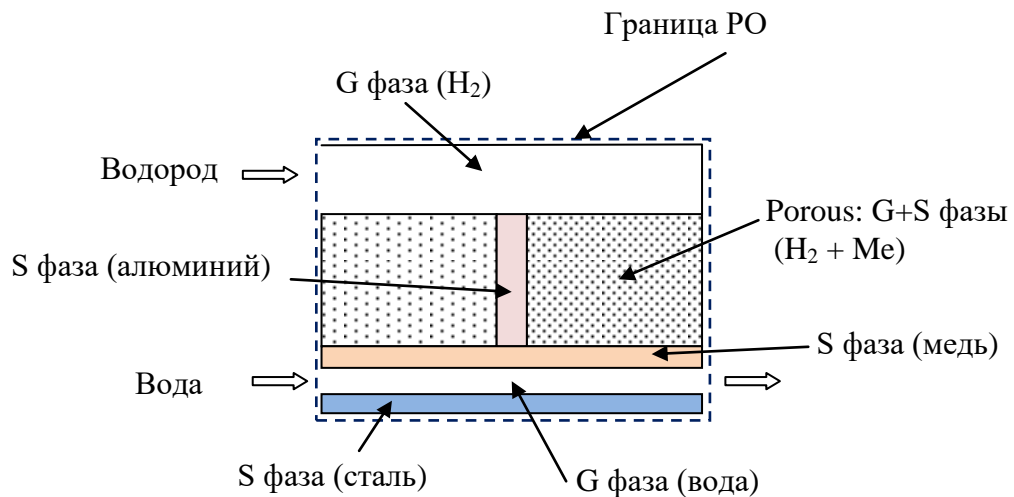


Рисунок 1.1 - Пример РО с несколькими фазами

### 1.1.2 Межфазное взаимодействие

Взаимодействие между G- и S- фазами сводится к обмену массой, импульсом и энергией. Способ описания межфазного взаимодействия зависит от возможности (и необходимости) выделения межфазной поверхности. В коде Anes предусматривается два способа описания межфазного взаимодействия.

*Пористая (дисперсная) модель* - в этой модели межфазная поверхность явно не выделяется, обменные источниковые члены в соответствующих дифференциальных уравнениях сохранения носят объемный характер. Эта модель используется для описания G-S взаимодействия в зонах Porous.

*Модель с явно выделенной границей* - в этой модели фазовая поверхность явно выделена и фиксирована в пространстве. Обменные источниковые члены формируются автоматически на основе поверхностных источников межфазного обмена. Эта модель используется для описания G-S взаимодействия на поверхностях раздела зон Struct-Flow (на FS-патчах).

Пористость Porous-зон определяется для каждой зоны моделью межфазного G-S взаимодействия, при этом математическая модель пакета допускает изменение пористости в функции времени

$$\varphi = f(\tau),$$

что позволяет описать *деформацию* Porous-зон.

### 1.1.3 Ф-переменные

С каждой фазой связывается набор зависимых переменных, которые в дальнейшем будут называться *Ф-переменными*. Поля этих переменных рассчитываются на основе численного решения соответствующих уравнений сохранения. Конкретные Ф-переменные кода и уравнения сохранения описаны в следующих главах.

**Замечание №1.** В Anes Ф-переменные можно разбить на две группы:

- 1) *предопределенные* Ф-переменные, математические модели для которых «защиты» в самом коде,
- 2) произвольные Ф-переменные *пользователя*, вид уравнений сохранения которых полностью определяются пользователем.

Ф-переменные пользователя могут быть как связанными с G- и S- фазами, так и не связанными с ними. Для последнего случая в Anes вводится псевдо-фаза с именем Spase. Эта

фаза заполняет всю РО и ее свойства должны быть определены как в зонах с G-фазой, так и с S-фазой.

## 1.2 Размерности переменных

Уравнения математических моделей, реализованные в коде Anes, основаны на фундаментальных уравнениях сохранения, которые не зависят от используемой системы единиц измерения. Поэтому пользователь вправе использовать любую систему, в том числе и безразмерную форму уравнений. Однако для уменьшения числа ошибок (особенно при записи источниковых членов и граничных условий) в коде Anes рекомендуется использовать систему единиц СИ (метр, секунда, кг, ньютон и т.д.).

Если пользователь хочет воспользоваться безразмерной формой уравнений (которая в ряде случаев позволяет существенно сократить количество вариантов расчетов прикладной задачи), то рекомендуется использовать следующий подход, который продемонстрируем на примере типичной задачи о течении и теплообмена в канале. При постоянных свойствах теплоносителя в уравнения для  $\Phi$ -переменных будут входить только два параметра - число Рейнольдса и число Прандтля:

$$Re = \frac{U_0 d_h}{\nu}, \quad Pr = \frac{\rho c_p \nu}{\lambda}$$

где  $U_0$  - средняя скорость теплоносителя,  $d_h$  - гидравлический диаметр,  $\rho$  - плотность,  $\lambda$ ,  $\nu$  - коэффициенты теплопроводности и кинематической вязкости.

Выберем для моделирования теплоноситель с произвольными свойствами, но такими, чтобы число Прандтля теплоносителя соответствовало заданному значению. Аналогично выберем произвольный размер канала  $d_h$ . А среднюю скорость  $U_0$ , необходимую для задания граничных условий на входном участке границы, определим из заданного значения  $Re$ :

$$U_0 = \frac{Re \nu}{d_h}$$

В итоге расчеты будут проводиться в размерном виде и при этом исходными параметрами задачи будут только число Рейнольдса и Прандтля.

## 1.3 V-переменные Anes

В уравнения сохранения, наряду с  $\Phi$ -переменными, входят дополнительные переменные, которые могут быть функциями  $\Phi$ -переменных, координат, времени и других параметров прикладной задачи (например, плотность и вязкость среды могут зависеть от давления и температуры). Кроме того при задании источниковых членов и граничных условий также могут появиться аналогичные переменные. В коде Anes такие переменные называются V-переменными Решателя (в языке описания проекта пользователя такие переменные реализуются через V-переменные файла проекта [2]). В общем случае V-переменная – это трехмерное поле, аналогичное полю  $\Phi$ -переменной, для которой пользователь может задать произвольный алгоритм их вычисления с помощью соответствующей V-переменной файла проекта.

Ниже при описании моделей всегда будет указываться, какие параметры моделей описываются в Anes с помощью V-переменных.

## 1.4 Файл проекта Anes

В данном документе основное внимание уделяется описанию математических моделей без их явной «привязки» к численным алгоритмам и программной архитектуре кода [1,2]. Однако для удобства работы с кодом в данном документе будут даны ссылки на операто-

ры описания прикладной задачи пользователя. В связи с этим приведем краткие сведения о способе описания прикладной задачи в коде Anes (более подробно он описан в [2]).

Для решения задачи пользователь должен подготовить файл проекта Anes. Файл проекта или А-файл представляет собой текстовый файл с произвольным именем и расширением «.а». Файл содержит операторы языка описания задачи (АИЛ-операторы), сгруппированные в блоки-секции. Большинство АИЛ-операторов имеют «простую» структуру:

```
<Имя АИЛ-переменной> = <Значение>
<АИЛ-Оператор>("<ключ>") = <Значение>
<АИЛ-функция>(<параметр 1>,<параметр 2>, ...)
```

(здесь угловые скобки  $\langle \rangle$  не являются частью операторов и используются только для выделения элемента). Ниже при описании математической модели будет указываться секция и конкретные АИЛ-операторы, «реализующие» эту модель в коде.

### 1.5 Сетки контрольных объемов

При численном решении дифференциальных уравнений сохранения РО разбивается на совокупность ячеек или контрольных объемов (КО), которая называется сеткой КО. Форма контрольных объемов может быть любой, но во всех CFD алгоритмах предполагается, что КО формируется замкнутым набором плоских граней. Грани КО являются важными элементами сетки КО, поскольку они «соединяют» КО друг с другом. При дискретизации уравнений переноса непрерывное поле  $\Phi$ -переменной  $\Phi(\mathbf{x})$  преобразуется в массив дискретных значений в каждом КО  $\Phi_k$  ( $k = 1 \dots \text{NoCells}$ ), где NoCells - число КО расчетной области. Точка внутри КО, к которой «привязывается» дискретное значение  $\Phi_k$ , зависит от численного алгоритма. Но в большинстве алгоритмов в качестве такой точки используется геометрический центр (центр тяжести) КО. В дальнейшем эта точка будет обозначаться буквой «Р».

В коде Anes используются два типа сеток КО: структурные ортогональные сетки и неструктурные сетки с локальным дроблением и моделью дробных ячеек. Особенности этих сеток КО и алгоритмы построения описаны в документе [1]. Приведем некоторые данные, которые понадобятся для понимания математических моделей.

При построении сетки КО объемные патчи (см. ниже) преобразуются в набор КО, поверхностные патчи - в набор граней КО, поверхностные граничные патчи - в набор граней КО, расположенных на границе расчетной области. На рисунке 1.2 показан пример приграничного КО и его характеристики. Ниже при описании граничных условий будут использоваться термины «значение  $\Phi$ -переменной на границе» и «значение  $\Phi$ -переменной вблизи границы». Эти точки соответствуют значениям на приграничной грани «w» и в центре «Р» приграничного КО (ниже "w" - это грань стенки, "b" - грань на любой границе).

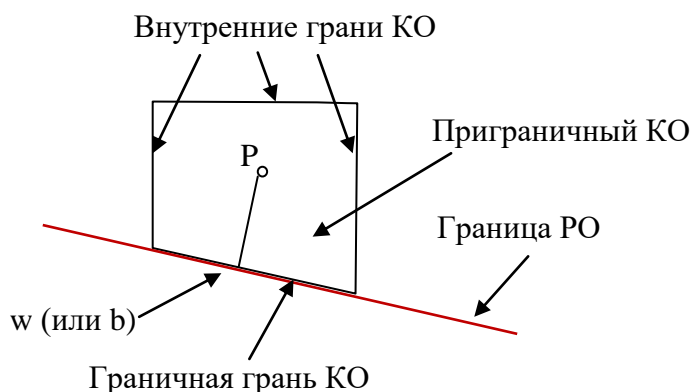


Рисунок 1.2 - Приграничный КО

## 2. Описание расчетной области

Для построения РО в коде Anes используется следующий «простой» алгоритм. На первом этапе создается *базовая расчетная область* (БРО), которая представляет собой параллелепипед в выбранной системе координат. На втором этапе в БРО помещаются специальные элементы кода Anes - Block-зоны, которые блокируют часть БРО:

$$PO = БРО - \sum_m (Block - зона)_m \quad (2.1)$$

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. 2.1 изображены двумерные (2D) расчетная и базовая области. Штриховкой на рис. 2.1 изображены области, занятые Block-зонами.

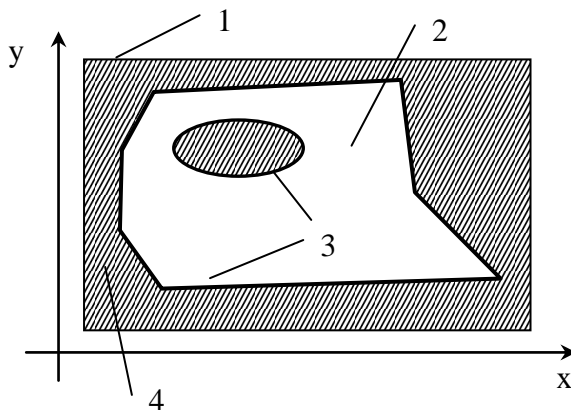


Рисунок 2.1 - Расчетная и базовая расчетная область

- 1 - базовая расчетная область,
- 2 – расчетная область,
- 3 - границы расчетной области,
- 4 - Block-зоны

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

Алгоритм (2.1) 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).

### ❗ Описание в проекте прикладной задачи :

Для выбора алгоритма формирования РО (стандартного или инверсного) используется оператор

```
Domain(FLOW/STRUCT/BLOCK,"Имя-материала")
```

секции [Patches] файла проекта.

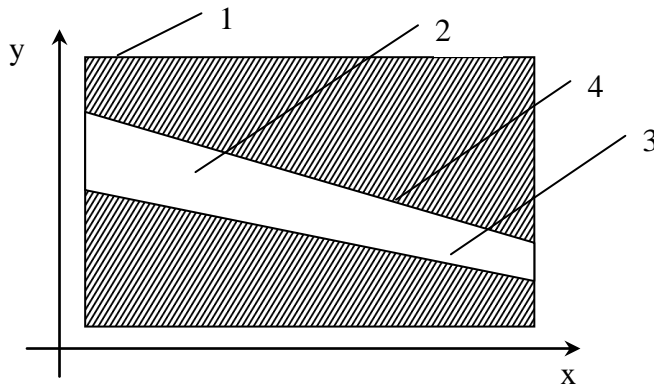


Рисунок 2.2 – Инверсный алгоритм построения РО:

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

## 2.1 Системы координат

В коде Anes для описания РО используются две системы координат (СК): декартова и цилиндрическая. Для универсальности записи уравнений используются обобщенные координаты  $(x, y, z)$ , которые связаны с декартовыми координатами  $(x_c, y_c, z_c)$  соотношениями:

$$x = x_c, \quad y = y_c, \quad z = z_c \quad - \text{декартова СК}$$

$$x_c = x \cos(y), \quad y_c = x \sin(y), \quad z_c = z \quad - \text{цилиндрическая СК,}$$

при этом обобщенная координата  $y$  для цилиндрической СК может изменяться только в пределах  $0 \dots 2\pi$ . Базовые расчетные области (в виде параллелепипеда) для этих систем координат показаны на рисунках 2.3 и 2.4. При использовании декартовой системы координат начало БРО всегда расположено в центре СК, ее положение и размеры -  $(0 - L_x, 0 - L_y, 0 - L_z)$ . При работе с цилиндрической системой координат допускается ее «смещение» вдоль обобщенной координаты  $x$  (радиуса  $r$ ), ее положение и размеры -  $(R_{in} - R_{in} + L_x, 0 - L_y, 0 - L_z)$ .

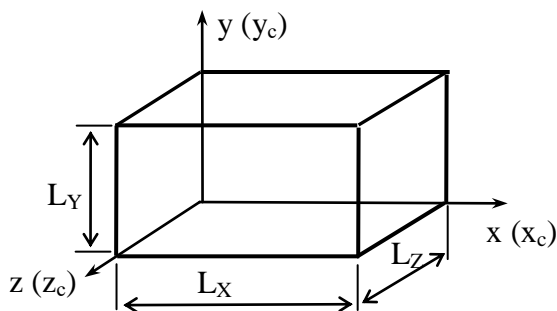


Рисунок 2.3 - Базовая расчетная область в декартовой системе координат

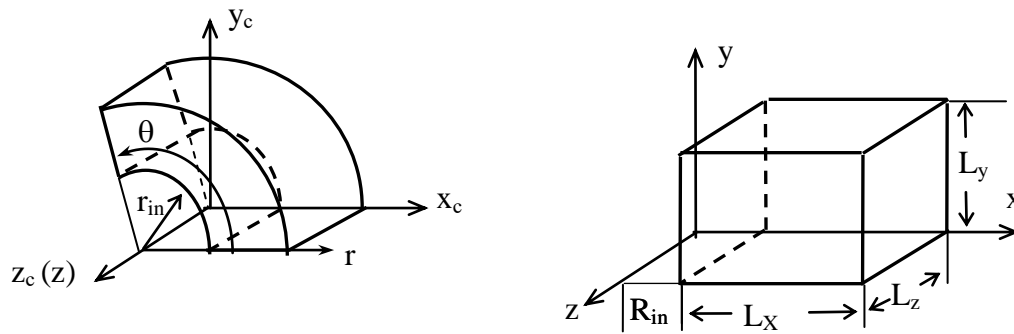


Рисунок 2.4 - Базовая расчетная область в цилиндрической  $(r, \theta, z)$  и обобщенной  $(x, y, z)$  системах координат

При описании математических моделей будут использоваться векторные операторы градиента и дивергенции. Координатный вид градиента и дивергенции в обобщенных координатах  $(x, y, z)$  зависит от системы координат. Для декартовой системы:

$$\nabla \Phi = \left\{ \frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y}, \frac{\partial \Phi}{\partial z} \right\}; \quad \text{div}(\mathbf{F}) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

для цилиндрической системы  $(x = r, y = \theta)$ :

$$\nabla \Phi = \left\{ \frac{\partial \Phi}{\partial x}, \frac{1}{x} \frac{\partial \Phi}{\partial y}, \frac{\partial \Phi}{\partial z} \right\}; \quad \text{div}(\mathbf{F}) = \frac{1}{x} \frac{\partial (x F_x)}{\partial x} + \frac{1}{x} \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

## 2.2 Патчи

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

- 1) уникальным *именем* для ссылки (произвольная комбинация 1-16 символов);
- 2) *типом* патча;
- 3) *геометрией* патча - подобласть пространства типа «точка», «поверхность» или «объем»;
- 4) дополнительными *параметрами*, количество которых и смысл зависит от типа патча.

Патчи являются основой для работы генератора сеток Компилятора Anes (А-компилятора). После работы генератора все объемные патчи переводятся в совокупности КО, а поверхностные патчи - в совокупность граней КО.

### 2.2.1 Описание 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 плоского 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 laShaper2d (см. документ [2]). Для описания 3Dbody-объектов в коде Anes можно использовать два формата:

1. TRI-формат программы ac3d (файл с расширением \*.tri),
2. универсальный STL-формат (файл с расширением \*.stl).

Форматы этих файлов описаны в документе [2].

Для удобства работы с 2Dbody и 3Dbody объектами предусмотрен специальный механизм масштабирования при помещении их в РО. Он позволяет использовать «сторонние» файлы 2GR, TRI и STL-форматов с произвольными размерами. Для масштабирования этих объектов в расчетной области используется следующий алгоритм:

- 1) на первом этапе определяется *граничный бокс* объекта - параллелепипед расчетной области, в который должен быть «вписан» геометрический объект,
- 2) на втором этапе в этот бокс вписывается 2GR, TRI, STL объект.

Граничный бокс описывается двумя тройками координат (рис. 2.5):

$x_0, y_0, z_0$  - точка «привязки» вершины бокса,

$S_x, S_y, S_z$  - размеры бокса вдоль координатных осей.

Для 2Dbody и 3Dbody объектов предусмотрены два способа масштабирования. Если значения всех  $S_x, S_y, S_z$  *положительны*, то объект «растягивается» вдоль всех координат, если значение одного из размеров *отрицательно*, объект масштабируется с сохранением пропорций. На рис. 2.6 на примере 2D задачи показаны два этих способа масштабирования.

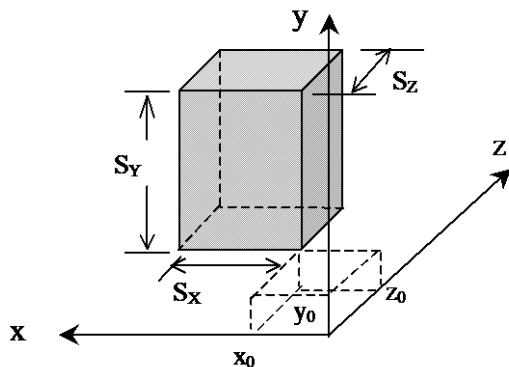


Рисунок 2.5 - Граничный бокс объекта

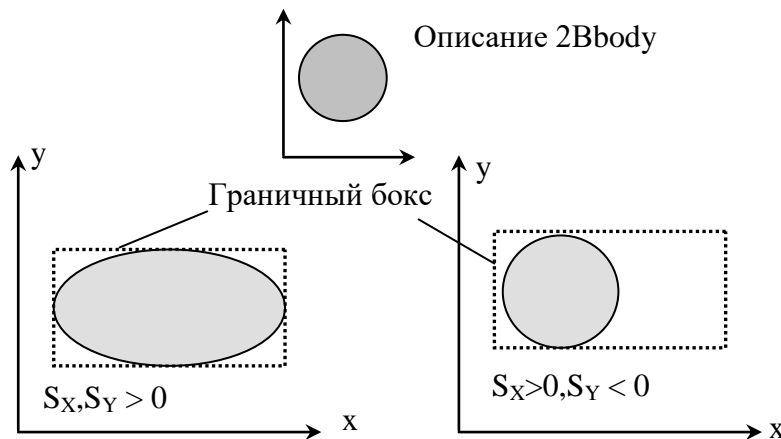


Рисунок 2.6 - Масштабирование объектов в расчетной области

Если пользователю необходимо описать объект в виде параллелепипеда, то нет необходимости использовать 2GR, TRI, STL объекты. В этом случае роль объема патча будет выполнять сам граничный бокс. В Anes такой тип патча называется *стандартным* патчем.

Для стандартного объемного патча допускается еще один способ задания граничного бокса - его поворот относительно одной из координатных осей. В этом случае к параметрам описания граничного бокса добавляется два параметра: ось вращения бокса и угол поворота бокса против часовой стрелки, заданный в градусах.

Объекты *типа «объем»* в коде используются для описания:

- 1) Block-зон,
- 2) распределения G- или S-фазы по пространству (зоны Flow, Struct и Porous),
- 3) объемных источников в уравнениях сохранения.

Объекты *типа «поверхность»* представляют собой плоские или криволинейные поверхности. Они создаются, либо непосредственно пользователем, либо автоматически генератором сетки A-компилятора при обработке объемных патчей. 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. В коде Anes плоские патчи типа «поверхность» используются для описания:

- 1) фрагментов границ расчетной области на поверхностях Block-зон,
- 2) поверхностных граничных источников в уравнениях переноса.

#### Описание в проекте прикладной задачи :

Все патчи описываются в секции [Patches] файла проекта.

Для описания патча типа «точка» используется оператор:

```
Patch("Имя-патча",Point,x0,y0,z0,0,0,0,....)
```

Для описания стандартного патча (без геометрического объекта) используется оператор:

```
Patch("Имя-патча",<Тип>,x0,y0,z0,SX,SY,SZ,....)
```

```
RotPatch("Имя-патча",<Тип>,x0,y0,z0,SX,SY,SZ,<Ось-поворота>,<Угол-поворота> )
```

Для описания патчей с объектом 2Dbody используется оператор:

```
2DPatch("Имя-патча",<Тип>,x0,y0,z0,SX,SY,SZ,<имя 2GR файла >,<направление оси>....)
```

Для описания патчей с объектом 3Dbody используется оператор:

```
3DPatch("Имя-патча",<Тип>,x0,y0,z0,SX,SY,SZ,<имя TRI/STL-файла >,....)
```

### 2.2.2 Тип патча

Тип патча определяет его назначение при формировании расчетной области. В текущей версии кода используются патчи следующих типов:

Тип	Геометрия	Назначение
Block, (BlockWall)	Объем	используется для формирования РО путем создания Block-зон,
BS, (BSWall)	Поверхность	поверхностный патч, описывающий часть границы РО;
Flow	Объем	используется для описания в РО различных Flow-зон.
Struct	Объем	используется для описания Struct-зон расчетной области,
Porous	Объем	используется для описания Porous-зон расчетной области,
Volume	Объем	патч для задания объемного источника,
Surface	Объем	патч для задания поверхностного источника,
Point	Точка	патч для задания точечного источника.



Патчи этих типов создаются пользователем при описании задачи в файле проекта. При обработке файла проекта А-компилятор создает ряд дополнительных патчей, которые будут рассмотрены ниже.

Отметим различия типов патчей Block/BlockWall и BS/BSWall. Если указаны типы BlockWall и BSWall, то патч помечается специальным логическим флагом IsWALL. Этим флагом в коде помечаются границы, представляющие собой непроницаемые *границы с прилипанием* (все три компонента скорости на границе равны нулю). В коде на таких границах автоматически задаются ГУ для компонент вектора скорости и активируются алгоритмы расчета пристенных турбулентных функций.

### 2.3 Патчи для описания расчетной области

Для формирования расчетной области и описания ее границ используются патчи типа Block (BlockWall) и BS (BSwall).

При обработке Block-патча его объем «удаляется» из PO, а сам он переводится в криволинейный в общем случае поверхностный патч типа BS с тем же именем.

При описании граничных условий с *каждым патчем* типа BS можно связать только *одно* граничное условие для *конкретной* Ф-переменной. Для задания *различных* граничных условий на поверхностях Block-объекта в коде реализованы поверхностные патчи типа BS. К BS-патчам предъявляется одно требование - они должны обязательно «касаться» любого Block-объекта. Заметим, что внешние, ограничивающие базовую расчетную область 6 поверхностей считаются поверхностями объектов типа Block и к ним также можно «привязывать» BS-объекты, которые используются для задания граничных условий на внешних границах базовой расчетной области.

На рис. 2.7 показан пример использования патчей Block и BS.

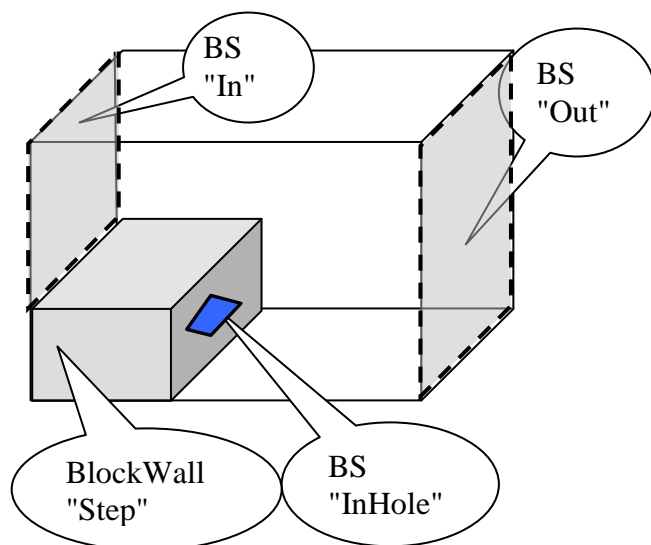


Рисунок 2.7 - Использование BLOCK и BS патчей

В этом примере рассматривается задача о течении за уступом. Уступ описывается BlockWall-патчем «Step». Основной поток поступает через левую границу базовой расчетной области (BS патч «In»), покидает – через правую границу (BS патч «Out»). На поверхности уступа задаются граничные условия прилипания. Если необходимо смоделировать дополнительный вдув потока из уступа, необходимо воспользоваться дополнительным BS-патчем – «InHole».

При обработке Block и BS патчей они обрабатываются *последовательно*, в порядке их описания в файле проекта. При этом используется *принцип перекрытия* – текущий патч перекрывает части всех предыдущих патчей, попавших в его зону.

Как уже отмечалось, Block и BS патчи могут содержать один дополнительный флаг IsWALL, задаваемый через тип патча : BlockWALL и BSWall. В этом случае границы патча рассматриваются как «твердая поверхность с условиями прилипания». На таких границах:

- 1) автоматически формируются граничные условия  $U_{gx}=U_{gy}=U_{gz} = 0$  (если ГУ для скоростей на этих патчах не заданы *явно* пользователем),
- 2) могут автоматически формироваться условия  $\Phi = 0$  для других переменных,
- 3) рассчитываются пристенные функции при использовании двухпараметрических моделей турбулентности с пристенными функциями,
- 4) именно эти границы используются для расчета поля  $L_{dis}$  – расстояния до ближайшей твердой поверхности.

**❗ Описание в проекте прикладной задачи :**

Для автоматического формирования граничных условий на WALL патчах для  $\Phi$ -переменной (кроме компонент скорости) необходимо использовать оператор

WallFixNull("Ф-Имя") = .TRUE.

секции [PHI Variables].

Для удобства описания BS-патчей, совпадающих с внешними границами базовой расчетной области, можно использовать укороченный оператор

DPatch("Имя патча", BS / BSWall, E\_BS / W\_BS / N\_BS / S\_BS / H\_BS / L\_BS)

секции [Patches]. В этом операторе третий параметр – код внешней границы БРО (рисунок 2.3):

W\_BS - «левая» граница  $x = 0$ ; E\_BS - «правая» граница  $x = L_x$ ;

S\_BS - «нижняя» граница  $y = 0$ ; N\_BS - «верхняя» граница  $y = L_y$ ;

L\_BS - «дальняя» граница  $z = 0$ ; H\_BS - «ближняя» граница  $z = L_z$ .

## 2.4 Патчи для описания распределения G- и S-фаз

Для описания распределения G- и S-фаз служат патчи типа Struct, Porous и Flow.

Патч типа Struct используется для описания отдельной Struct-зоны, заполненной *одним* материалом S-фазы. Имя этого материала указывается в качестве дополнительного параметра патча и служит для ссылки на свойства материала.

При обработке Struct-патча А-компилятором Anes в его контрольных объемах значение  $\phi$  пористости автоматически устанавливается равным 0. Сам патч преобразуется в патч типа Volume, что позволяет в дальнейшем связать с ним объемные источники для  $\Phi$ -переменных.

Одновременно при обработке патча этого типа могут быть созданы четыре поверхностных патча с именами:

- 1) FS\_<имя патча> - создается *всегда* на границе раздела данного Struct с областями, содержащими G-фазу (зоны Flow или Porous). Патч имеет тип Surface и привязан к КО со стороны G-фазы. Используется для моделирования межфазного взаимодействия G- и S- фаз. Может использоваться пользователем для задания своих собственных поверхностных источников со стороны G-фазы.
- 2) SF\_<имя патча> - аналог предыдущего патча, но привязанный к КО Struct-зоны. Создается по желанию пользователя для задания поверхностных источников со стороны S-фазы.
- 3) SS\_<имя патча> - поверхностный патч, создаваемый на границах раздела данной Struct-зоны с другими Struct-зонами. Привязка патча – КО данной зоны. Используется *только для задания коэффициента сопротивления* для  $\Phi$ -переменных, связанных с S-фазой (для  $\Phi$ -переменной  $T_s$  – это коэффициент термического сопротивления).
- 4) SI\_<имя патча> - полный аналог предыдущего патча, служащий для задания поверхностного источника пользователя со стороны данного Struct-патча.

- 5) SO\_<имя патча> - аналог предыдущего патча, но с внешней стороны Struct-патча (привязан к КО соседней Struct-зоны). Используется для задания поверхностного источника .

**ⓘ Описание в проекте прикладной задачи :**

Создание SF\_, SS\_, SI\_ и SO\_ патчей определяются флагами-переключателями

IsCreateSF = .TRUE.

IsCreateSS = .TRUE.

IsCreateSIO = .TRUE.

секции [Patches].

Патч типа Porous используется для описания отдельной Porous-зоны, заполненной *одним материалом S-фазы* и имеющую *одну микроструктуру S-фазы* (одну модель межфазного G-S взаимодействия). Имя материала и модели взаимодействия указываются в качестве дополнительных параметров патча и служат для ссылки на свойства материала и модель FS-взаимодействия. Сам патч преобразуется в патч типа Volume, что позволяет в дальнейшем связать с ним объемные источники для  $\Phi$ -переменных. Если в качестве имени материала указан символ «\*», то такой патч описывает двухуровневую L2Porous пористую модель. В этой модели для описания S-фазы используются специальные модели микроуровня.

При работе со *структурной* сеткой Porous-Патч не создает дополнительных поверхностных патчей. При использовании *неструктурных* сеток на границе зон Flow-Porous создается поверхностный патч с префиксом «PF\_». Этот патч используется Решателем Anes для своих целей, однако пользователь может рассматривать его как обычный поверхностный патч и привязывать к нему свои источники.

При обработке Flow, Struct и Porous патчей они обрабатываются последовательно, в порядке их описания в файле проекта. При этом также используется *принцип перекрытия* – текущий патч перекрывает части все предыдущих патчей.

При стандартном построении РО вначале вся расчетная область заполнена зоной Flow, материал (свойства «среды») которой задан при описании БРО. Для смены материала G-фазы в уже существующей зоне Flow используется патч типа Flow. При определении патча в качестве дополнительного параметра, как и в патче Struct, указывается имя нового материала G-фазы.

Обычно Flow-патчи используются для моделирования течений нескольких теплоносителей в одной расчетной области. Типичный пример приведен на рис. 2.8.

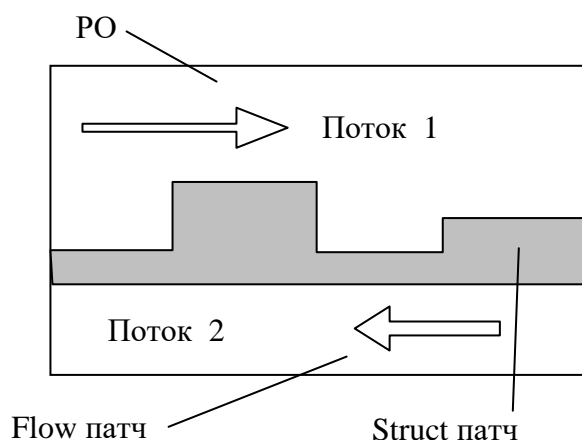


Рисунок 2.8 – Типичный пример использования Flow-патча

В этой задаче рассматривается элемент конструкции теплообменника с двумя теплоносителями, разделенных стенкой. При описании свойств G-фазы (плотности, вязкости, теп-

лоемкости и т.д.) имя G-материала используется пользователем для задания различных значений свойств в КО.

## 2.5 Инверсный алгоритм описания РО

По умолчанию, базовая расчетная область инициализируется Flow-зоной с именем материала G-фазы по умолчанию. Иногда удобнее использовать другой – инверсный режим инициализации.

В этом режим БРО инициализируется одной *блокированной* зоной, заполняющей всю БРО. Расчетная область создается с помощью патчей Struct и Flow, которые *перекрывают* блокированные КО. При этом создаются патчи типа BS или BSwall с именами

BS\_<имя исходного патча>,

которые и используются для задания граничных условий. Тип оператора (BS или BSwall) определяется флагом-переключателем

IsWallBS = .True./.False.

Для активизации этого режима построения РО необходимо, чтобы первым оператором секции [Patches] был оператор:

```
Domain(BLOCK, " ")
```

## 2.6 Патчи для описания источниковых членов

В Anes источниковый член в уравнении сохранения Ф-переменной может быть задан *только* путем его привязки к патчу типа Volume, Surface или Point. Такие патчи автоматически создаются при обработке Flow, Struct или Porous патчей. Однако пользователь может создать дополнительно любое количество источниковых патчей.

Патч типа *Volume* представляет собой 3DBody объект. При обработке патча он переводится в совокупность КО. В дальнейшем при расчете источников в Решателе именно для этих КО рассчитывается объемный источник, заданный пользователем.

Компилятор пакета автоматически создает Volume – патч с именем «\*», связанный со всей расчетной областью. Этот патч может использоваться пользователем для задания источников, связанных со всей РО.

Патч типа *Surface* использует геометрический объект «поверхность» (напомним, что в текущей версии – это плоские поверхности, расположенные в одной из координатных плоскостей). При обработке патча он переводится в совокупность граней КО, с которыми в Решателе связывается поверхностный источник пользователя. Для определения грани КО для патча используется дополнительный параметр патча – направление нормали к стороне поверхности:

- 1) SurfUP - используется сторона поверхности, нормаль которой направлена в *положительном* направлении оси,
- 2) SurfDOWN - используется сторона поверхности, нормаль которой направлена в *отрицательном* направлении оси.

Алгоритм выбора иллюстрируется на рис. 2.9.

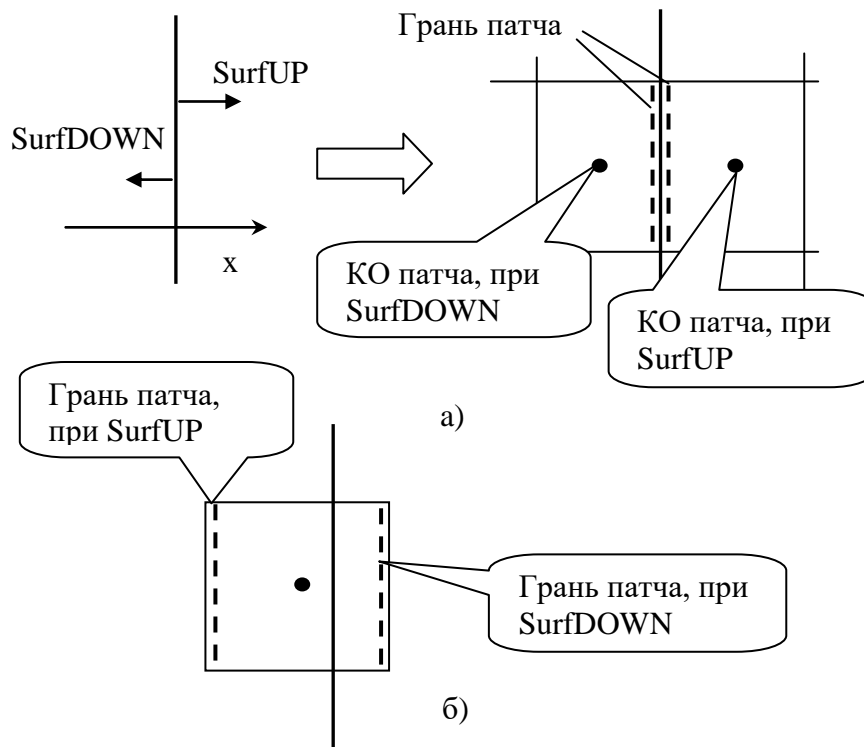


Рисунок 2.9 - Грань и КО Surface-патча:

- а) в случае совпадения поверхности патча с гранью КО,  
 б) в случае прохождения поверхности патча через КО

Патч типа *Point* представляет собой объект типа «точка». При обработке патча он переводится в один КО. В дальнейшем при расчете источников в Решателе для этого КО рассчитывается точечный источник, заданный пользователем.

В отличие от других патчей для патчей типа *Volume*, *Surface* и *Point* не используется алгоритм перекрытия – патчи обрабатываются абсолютно независимо друг от друга и других патчей.

### 3. Математическая модель обобщенной Ф-переменной

Прежде чем переходить к описанию конкретных математических моделей G-, S- фаз, рассмотрим общие принципы реализации моделей в коде Anes. Для этого рассмотрим математические модели «чистой» G-фазы ( $\varphi = 1$ ). Это позволит избежать дублирования при описании моделей фаз. Ниже рассматривается модель для «обобщенной» Ф-переменной, типичным примером которой может быть, например, пассивная примесь или кинетическая энергия турбулентных пульсаций (из двухпараметрической модели турбулентности).

#### 3.1 Уравнение баланса массы

Прежде чем обсуждать уравнение переноса для произвольной Ф-переменной сплошной среды (G-фазы), рассмотрим вопрос об уравнении баланса массы (уравнении неразрывности). В систему уравнений гидродинамики (уравнений Навье-Стокса) входит «странное» уравнение баланса - уравнение неразрывности, которое не связано ни с одной Ф-переменной. С другой стороны в состав Ф-переменных входит поле давления  $p$ , для которого отсутствует свое «родное» уравнение сохранения.

Уравнение неразрывности для «чистой» G-фазы можно записать в виде:

$$\frac{\partial \rho_g}{\partial t} + \operatorname{div}(\rho_g \mathbf{U}_g) = M_p \quad (3.1)$$

где  $\rho_g$  - плотность G-фазы ( $\text{кг}/\text{м}^3$ ),  $\mathbf{U}_g = (U_{gx}, U_{gy}, U_{gz})$  - вектор скорости фазы ( $\text{м}/\text{с}$ ),  $M_p$  - удельный объемный источник массы ( $\text{кг}/(\text{м}^3 \cdot \text{с})$ ), смысл подстрочного индекса "р" будет ясен из дальнейшего.

При моделировании сжимаемых течений (уравнений газовой динамики) уравнение (3.1) обычно рассматривается как уравнение для *плотности*, а поле давления является *свойством среды*. Однако этот подход теряет универсальность для задач с постоянной плотностью.


В коде Anes давление «р» рассматривается как обычная Ф-переменная, а уравнение для нее - это уравнение неразрывности (3.1). Несмотря на неочевидность такого утверждения, оно легко обосновывается на уровне численных алгоритмов (смотрите алгоритм SIMPLE [1]).

Такая привязка давления к уравнению (3.1) имеет принципиальный характер, поскольку в математической модели кода Anes граничные условия для Ф-переменной связываются с плотностью полного потока этой Ф-переменной через границу РО.

Уравнение (2.1) можно переписать в несколько иной форме

$$\frac{\partial \rho_g}{\partial t} + \operatorname{div}(\mathbf{m}_g) = M_p, \quad \mathbf{m}_g = \rho_g \mathbf{U}_g, \quad (3.2)$$

где  $\mathbf{m}_g$  - вектор плотности потока массы ( $\text{кг}/(\text{м}^2 \cdot \text{с})$ ). Утверждение, приведенное выше, означает, что на границе РО плотность массового потока должна зависеть от граничного условия для давления на границе (именно поэтому и появился подстрочный индекс "р" у величины  $M_p$ ). Такой подход, используемый в коде Anes, позволяет реализовать очень гибкий алгоритм задания граничных условий и для других Ф-переменных.

 **Замечание.** В уравнениях (3.1) и (3.2) присутствует объемный источник массы, который в «классических» уравнениях Навье-Стокса отсутствует (масса не исчезает и не возникает!). Причиной появления источника массы в математических моделях кода Anes могут быть:

- 1) точечные источники, связанные с внутренними «подсеточными» границами,

- 2) источники, связанные с гетерогенными химическими реакциями (потоки массы между G- и S- фазами).

Поясним смысл точечных источников. Пусть мы моделируем распространение вредной примеси в озере (характерный размер - 1000 м). Примесь вносится через систему труб с характерным размером 10 см. Единственным способом моделирования таких задач - это использование точечных источников.

### 3.2 Уравнение баланса $\Phi$ -переменной

$\Phi$ -переменные, связанные с чистой G-фазой, удовлетворяют «стандартному» уравнению сохранения (уравнению переноса материальной субстанции). Форма уравнения (3.3) справедлива в том случае, если  $\Phi$  - некоторое переносимое свойство, отнесенное к 1 кг массы (например,  $\mathbf{u}$  - удельный импульс,  $h$  - удельная энтальпия и т.д.):

$$\frac{\partial(\rho_g \Phi)}{\partial \tau} + \text{div}(\rho_g \mathbf{U}_g \Phi - \Gamma_\Phi \nabla \Phi) = S_\Phi^{(1)} + S_\Phi \quad (3.3)$$

Здесь  $\Gamma_\Phi$  - эффективный коэффициент диффузионного переноса,  $S_\Phi^{(1)}$ ,  $S_\Phi$  - объемные источниковые члены (отнесенные к 1 м<sup>3</sup> пространства). В уравнении (3.3) источниковые члены представлены в виде двух составляющих – внутренних источников  $S_\Phi^{(1)}$ , автоматически рассчитываемых в Решателе, и произвольных источников  $S_\Phi$ , смысл которых полностью определяется прикладной задачей пользователя.

Подробно *внутренние* источники рассмотрены в последующих главах. В данном параграфе просто перечислим их:

- 1) градиент давления и напряженность сил плавучести и других массовых сил (н/м<sup>3</sup>) в уравнениях баланса импульса (уравнения движения);
- 2) дополнительные источники в уравнениях движения в цилиндрической системе координат, обусловленные тем, что взаимосвязь тензора касательных напряжений и тензора скоростей деформаций (закон Ньютона) не сводятся к стандартному «градиентному» соотношению;
- 3) работа сил (мощность в 1 м<sup>3</sup>) давления и вязкая диссипация в уравнении энергии;
- 4) межфазный обмен массой, импульсом и энергией между G-S фазами в Porous-зонах и на границах раздела Flow-Struct зон (кг/(м<sup>3</sup>·с));
- 5) источники, описывающие порождение и диссипацию в уравнениях двухпараметрических моделей турбулентности.

Уравнение (3.3) можно записать в виде, аналогичном (3.2):

$$\frac{\partial(\rho_g \Phi)}{\partial \tau} + \text{div}(\mathbf{J}_\Phi) = S_\Phi^{(1)} + S_\Phi, \quad \mathbf{J}_\Phi = \mathbf{m}_g \Phi - \Gamma_\Phi \nabla \Phi \quad (3.4)$$

Здесь  $\mathbf{J}_\Phi$  - вектор плотности полного потока свойства  $\Phi$ , являющийся суммой двух векторов: вектора плотности конвективного потока  $\mathbf{J}_\Phi^c = \mathbf{m}_g \Phi$  и вектора плотности диффузионного потока  $\Phi$ -переменной  $\mathbf{J}_\Phi^d = -\Gamma_\Phi \nabla \Phi$ .

### 3.3 Коэффициент диффузионного переноса

В уравнениях (3.3) и (3.4) диффузионная составляющая переноса представлена в простом градиентном виде. Если диффузионный перенос для конкретной  $\Phi$ -переменной отличается от такой модели (типичный пример - тензор напряжений Ньютона для уравнения баланса импульса), то члены, отличные от градиента переносятся во внутренние источники. В Anes для расчета эффективного коэффициента диффузионного переноса используются два соотношения:

$$\Gamma_{\Phi} = \rho_g \left( \frac{\nu_g}{\sigma_{\Phi}} + \frac{\nu_t}{\sigma_{\Phi,t}} \right) \quad \text{или} \quad \Gamma_{\Phi} = D_{\Phi} + \frac{\rho_g \nu_t}{\sigma_{\Phi,t}}, \quad (3.5)$$

где:  $\nu_g$  - кинематический коэффициент вязкости G-фазы ( $\text{м}^2/\text{с}$ ),  $\nu_t$  - кинематический коэффициент турбулентной вязкости, определяемый используемой моделью турбулентности,  $\sigma_{\Phi}, \sigma_{\Phi,t}$  - молекулярное и турбулентное «число Прандтля» для данной  $\Phi$ -переменной,  $D_{\Phi}$  - молекулярный коэффициент «диффузии» для данной  $\Phi$ -переменной.

Конкретный вид соотношений (3.5) и смысл переменных  $\sigma_{\Phi}, \sigma_{\Phi,t}$  и  $D_{\Phi}$  для отдельных  $\Phi$ -переменных будет приведен ниже, отметим только, что все переменные, входящие в соотношения (3.5) рассматриваются в коде как V-переменные, поэтому пользователь может задать для них любые функциональные зависимости.

Выбор формы представления эффективного коэффициента диффузии зависит от того, что известно пользователю: молекулярный коэффициент диффузии или соответствующее число Прандтля.

**☞ Замечание.** В коде Anes в качестве коэффициента вязкости используется *кинематический* коэффициент вязкости. Это необходимо помнить при задании свойства "Visc" материала G-фазы (для воды коэффициенты динамической  $\mu_g$  и кинематической  $\nu_g$  вязкости могут отличаться в 1000 раз).

**① Описание в проекте прикладной задачи :**

Для конкретной  $\Phi$ -переменной с именем "Ф-имя" используемая форма уравнения (3.5) определяется операторами секции [PHI Variables]

```
TypeDiff("Ф-имя") = ND_PRAN / ND_PROP
NamDiff("Ф-имя") = <имя свойства>
```

где <имя свойства> - имя свойства для «числа Прандтля» или коэффициента диффузии, используемое для задания свойств материалов G-фазы в секции [Properties].

### 3.4 Источниковые члены

В коде Anes источниковый член  $S_{\Phi}$  может быть задан только путем его привязки к патчу PO. Если необходимо привязать источник для объема всей расчетной области (или к ее части, занятой фазой  $\Phi$ -переменной), то в качестве имени патча можно использовать псевдо-имя «\*».

При описании функциональной зависимости источника в коде Anes используется линеаризация источникового члена. Этот подход является типичным для всех CFD кодов [3,4]. Обычно используют две формы линеаризации:

$$S_{\Phi} = S_{\Phi}^{(+)} - S_{\Phi}^{(-)}\Phi, \quad S_{\Phi} = C_{\Phi}(V_{\Phi} - \Phi) \quad (3.6)$$

где  $\{S_{\Phi}^{(+)}, S_{\Phi}^{(-)}\}$  и  $\{C_{\Phi}, V_{\Phi}\}$  - новые составляющие источникового члена  $S_p$ , при этом обязательно выполнение условий  $S_{\Phi}^{(-)} \geq 0, C_{\Phi} > 0$ . Заметим, что математически такая линеаризация носит формальный характер и не имеет никакого смысла. Однако линеаризация источника с  $S_{\Phi}^{(-)} \geq 0$  позволяет существенно улучшить сходимость итерационных алгоритмов [3], т.к. выражения (3.6) описывают источник с отрицательной обратной связью (увеличение  $\Phi$  приводит к уменьшению источника  $S_{\Phi}$ ), что препятствует «разболтке» решения от итерации к итерации. В коде Anes используется вторая форма (3.6). При задании источникового члена пользователь определяет две характеристики:  $C_{\Phi}$  - коэффициент источника и  $V_{\Phi}$  - значение источника. Сразу отметим, что и коэффициент и значение в коде рассматриваются как V-переменные, поэтому пользователь может задать произвольную функциональную зависимость этих параметров.



Использование второй формы линеаризации позволяет в коде Anes логичным образом реализовать специальные частные формы источниковых членов (и граничных условий, что более важно). Для этого в файле проекта используются специальные *символьные* значения коэффициента источника.

### 3.4.1 Размерность источников пользователя

Поскольку в РО могут присутствовать объемные, поверхностные и точечные патчи, то пользователь может задать источниковые члены трех типов: объемные  $S_{\Phi}^{(V)}$ , поверхностные  $S_{\Phi}^{(A)}$  и точечные  $S_{\Phi}^{(P)}$ .

Объемный источник  $S_{\Phi}^{(V)}$  определяет скорость производства  $\Phi$ , отнесенную к  $1 \text{ м}^3$  фазы  $\Phi$ -переменной (напомним, что в данной главе рассматриваются  $\Phi$ -переменные, связанные с G-фазой). В уравнении сохранения (3.3) этот источник записывается в виде

$$S_{\Phi} = S_{\Phi}^{(V)}. \quad (3.6)$$

Поверхностный источник представляет собой скорость производства  $\Phi$ , отнесенную к  $1 \text{ м}^2$  поверхности, занятой фазой  $\Phi$ -переменной. В уравнении сохранения (3.3) такой источник можно записать в виде

$$S_{\Phi} = S_{\Phi}^{(A)} A_V, \quad (3.7)$$

где  $A_V$  - удельная поверхность (площадь поверхности в  $1 \text{ м}^3$  объема).

Точечный источник определяет саму скорость производства  $\Phi$  в уравнениях переноса

$$S_{\Phi} = S_{\Phi}^{(P)} \delta(\mathbf{x} - \mathbf{x}_0), \quad (3.8)$$

где  $\delta(\mathbf{x})$  - дельта-функция.

Преобразования (3.6) - (3.8) производятся в коде автоматически, поэтому при задании источника пользователь не должен преобразовывать источники к объемной форме.

### 3.4.2 Источниковые члены для уравнения неразрывности

Как уже отмечалось уравнение неразрывности (3.1) в коде рассматривается как уравнение для давления  $p$ . Поэтому линеаризованный источник массы задается в виде

$$M_p = C_p (V_p - p), \quad (3.9)$$

где  $C_p$ ,  $V_p$  - коэффициент и значение источника массы (напомним еще раз, что они являются  $V$ -переменными).

В коде для источника массы предусмотрены две специальные формы.

Для задания фиксированного давления  $p_0$  в области патча необходимо для коэффициента источника задать символьное значение:

$$C_p = \text{SR\_FIXPRESS}, \quad V_p = p_0.$$

Если необходимо задать непосредственно массовый поток  $M_0$ , то нужно воспользоваться следующими значениями коэффициента и значения:

$$C_p = \text{SR\_FIXMASS}, \quad V_p = M_0.$$

### 3.4.3 Источниковые члены для $\Phi$ -переменной

Если на патче, для которого задается источниковый член  $S_{\Phi}$  (например, источник импульса), не задан массовый источник  $M_p$ , то источниковый член записывается в стандартном линеаризованном виде (3.6):

$$S_{\Phi} = C_{\Phi} (V_{\Phi} - \Phi). \quad (3.10)$$

Если на патче присутствует отличный от нуля массовый источник  $M_p$ , то используется другая форма линеаризованного источника

$$S_{\Phi} = \max(M_p, 0) \cdot V_{\Phi} - \max(-M_p, 0) \cdot \Phi + C_{\Phi} (V_{\Phi} - \Phi). \quad (3.11)$$

Дополнительные члены в (3.11) связаны с «конвективным» порождением  $\Phi$ -переменной: при «вдуве» массы в РО ( $M_p > 0$ ) в фазу вносится масса со значением  $\Phi = V_\Phi$ , при «уносе» массы из РО ( $M_p < 0$ ) из фазы удаляется масса со значением  $\Phi$  в текущей точке РО.

Для удобства пользователя предусмотрены несколько частных форм задания источниковых членов. Они реализуются через задание специальных символьных значений коэффициента источника.

Для получения в области патча *фиксированного значения*  $\Phi_0$  параметру  $S_\Phi$  при описании источникового члена в файле описания проекта следует присвоить специальное *символьное* значение:

$$S_\Phi = SR\_FIXVAL$$

Для задания *полного источника*  $S_\Phi = S_0$  параметру  $S_\Phi$  при описании источникового члена необходимо присвоить другое специальное *символьное* значение

$$S_\Phi = SR\_FIXSOURCE$$

Заметим, что в этом случае задается *полный* источник, а источники, связанные с массовым источником, полностью игнорируются.

При моделировании источников, связанных с «вдувом» или «отсосом» массы можно использовать нелинеаризованные источники:

$$S_\Phi = \max(M_p, 0) \cdot V_\Phi - \max(-M_p, 0) \cdot \Phi \quad (3.12)$$

Для задания такого источника необходимо использовать символьное значение коэффициента:

$$S_\Phi = SR\_MASSONLY$$

**Замечание.** Если для патча задан массовый источник  $M_p$ , а для какой-то  $\Phi$ -переменной источник не задан, то он все равно будет сформирован Решателем кода Anes для этой  $\Phi$ -переменной в виде

$$S_\Phi = \max(M_p, 0) \cdot \Phi - \max(-M_p, 0) \cdot \Phi = M_p \cdot \Phi$$

Такой вид «скрытого» источника ( $V_\Phi = \Phi$ ) связан со спецификой численного алгоритма Anes (подробнее смотрите в [1]). Если необходимо «явно» занулить источник для  $\Phi$ -переменной, его следует *явно задать* в виде

$$S_\Phi = SR\_FIXSOURCE, \quad V_\Phi = 0.$$

Типичный пример такого явного задания нулевого источника - моделирование химических реакций в однофазной многокомпонентной смеси.

Источниковые члены для отдельных  $\Phi$ -переменных G- и S- фаз имеют ряд особенностей, которые подробно описаны в следующей главе.

#### **Описание в проекте прикладной задачи :**

Источниковые члены для  $\Phi$ -переменной с именем "Ф-имя" на патче с именем Имя-патча задаются парой операторов секции [Sources]

$$\begin{aligned} \text{COEF}(\text{"Имя-патча.Ф-имя"}) &= V\text{-переменная} \\ \text{VAL}(\text{"Имя-патча.Ф-имя"}) &= V\text{-переменная} \end{aligned}$$

### **3.5 Граничные условия**

Для однозначной постановки задачи необходимо на *границе* расчетной области задать граничные условия для всех решаемых  $\Phi$ -переменных. При математической постановке задачи уравнения баланса для  $\Phi$ -переменной рассматривается как дифференциальные уравнения в частных производных (обычно эллиптического типа) и для них используются граничные условия трех типов, например, для уравнения энергии:

1) *ГУ 1-го рода*: на границе задается значение температуры,

- 2) *ГУ 2-го рода*: на границе задается нормальная к границе компонента плотности теплового потока,
- 3) *ГУ 3-го рода*: на границе задается линейная комбинация первых двух условий.

В коде Anes используется другой подход, вытекающий из особенностей численных алгоритмов кода. Он позволяет с одной стороны задать граничные условия любой сложности, а с другой стороны легко реализовать классические условия 1, 2 и 3 рода. Суть этого подхода заключается в задании плотности потока  $\Phi$ -переменной на границе РО в виде обычных поверхностных источников (см. предыдущий раздел):

$$\begin{aligned} m_b &= \mathbf{m}_b \cdot \mathbf{n}_{bc}, & \mathbf{J}_{\Phi,b} &= \mathbf{J}_{\Phi,b} \cdot \mathbf{n}_{bc}, \\ \mathbf{m}_b &= (\rho_g \mathbf{U}_g)_b, & \mathbf{J}_{\Phi,b} &= (\mathbf{m}_g \Phi - \Gamma_\Phi \nabla \Phi)_b, \end{aligned} \quad (3.13)$$

где  $\mathbf{n}_{bc}$  - единичная нормаль к границе, направленная внутрь РО (см. рисунок 3.1). Отличие различных по смыслу ГУ заключается в трактовке параметров источников.

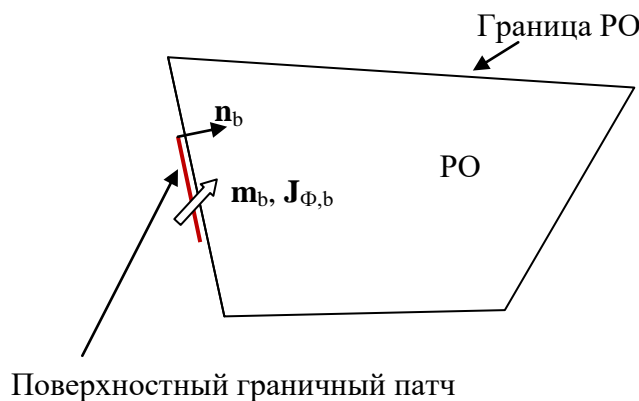


Рисунок 3.1 - Граничные поверхностные источники

### 3.5.1 Граничные условия для давления

Для задания плотности массового потока на границе используется аналог соотношения (3.9):

$$m_b = C_{p,b} (V_{p,b} - p_p) \quad (3.14)$$

Здесь  $C_{p,b}$  и  $V_{p,b}$  - коэффициент и значение граничного условия,  $p_p$  - значение давления внутри РО *вблизи границы* (см. раздел 1.6).

Как и в случае задания объемных источников для ГУ предусмотрены два частных случая.

Для задания на границе *фиксированного давления*  $p_0$  необходимо использовать следующие значения параметров источника:

$$C_{p,b} = \text{VC\_FIXPRESS}, \quad V_{p,b} = p_0$$

Если на границе известна плотность нормального массового потока  $m_0$ , то необходимо использовать следующий набор параметров:

$$C_{p,b} = \text{VC\_FIXMASS}, \quad V_{p,b} = m_0$$

В коде Anes имеются еще несколько специальных граничных условий для уравнения неразрывности. Они будут рассмотрены в следующей главе.

Если на граничном патче пользователем не задано граничное условие для давления, то такая граница по умолчанию считается *непроницаемой* и на ней автоматически задается

$$m_b = 0.$$


В коде предусмотрены еще два специальных граничных условий для случая заданного давления на границе: задание на границе давления торможения и граничное условие BC\_OUTPRESS. Они подробно описаны в пункте 4.1.1.

Границы, на которых плотность массового потока отлична от нуля, можно разделить на два типа:

- 1) *входные*, через которые поток втекает в расчетную область ( $m_b > 0$ ),
- 2) *выходные*, через которые поток покидает расчетную область ( $m_b < 0$ ).

На *входных* границах для массового потока можно задавать любые типы условий, удовлетворяющие (3.14). Однако, если распределение плотности массового потока известно, то рекомендуется использовать граничное условие с  $C_{p,b} = BC\_FIXMASS$ .

На *выходных* границах обычно задается «противодавление» с  $C_{p,b} = BC\_FIXPRESS$ .

 **Замечание.** В Решателе Anes задание постоянного давления на границе соответствует большому значению коэффициента  $C_{p,b}$ . Разумными значениями являются  $C_{p,b} = 10^3 - 10^5$ . При использовании константы BC\_FIXPRESS значение этого коэффициента равно  $10^3$ .

### 3.5.2 Граничные условия для $\Phi$ -переменной

Как и для плотности массового потока, граничные условия для  $\Phi$ -переменной задаются как поверхностные источники, аналогичные (3.11). При этом допускается две формы представления плотности потока  $\Phi$ -переменной:

$$\begin{aligned} J_{\Phi,b} &= \max(m_b, 0) \cdot V_{\Phi,b} - \max(-m_b, 0) \cdot \Phi_P + C_{\Phi,b} (V_{\Phi,b} - \Phi_w), \\ J_{\Phi,b} &= \max(m_b, 0) \cdot V_{\Phi,b} - \max(-m_b, 0) \cdot \Phi_P + |C_{\Phi,b}| (V_{\Phi,b} - \Phi_P). \end{aligned} \quad (3.15)$$

Здесь  $C_{\Phi,b}$  и  $V_{\Phi,b}$  - коэффициент и значение граничного условия, которые в коде описываются  $V$ -переменными. В соотношениях (3.15) используются два значения  $\Phi$ -переменных:  $\Phi_w$  - значения на самой границе РО и  $\Phi_P$  - значения вблизи границы РО (см. раздел 1.6). В коде выбор формы задания ГУ определяется *знаком*  $C_{\Phi,b}$  (напомним, что для устойчивости итерационных алгоритмов коэффициент  $C_{\Phi,b}$  должен быть неотрицательным). Если  $C_{\Phi,b} > 0$  - используется первая форма, если он *отрицателен* - вторая. Выбор той или иной формы зависит от физической постановки задачи, но в большинстве задач как правило используется первая форма поверхностного источника.

Как и для объемных источников для ГУ предусмотрен набор частных граничных условий, которые упрощают процесс задания ГУ. Для этого используются специальные символичные значения коэффициента ГУ  $C_{\Phi,b}$ .

### 3.5.3 Граничное условие первого рода

Для задания на границе значения искомой функции  $\Phi = \Phi_0$  (граничное условие *первого* рода) в случае непроницаемой границы ( $m_b = 0$ ) пользователю необходимо присвоить следующие значения коэффициенту и источнику:

$$C_{\Phi,b} = BC\_FIXVAL, \quad V_{\Phi,b} = \Phi_0$$

Это граничное условие можно использовать и для входных границ с  $m_b > 0$ . В этом случае в плотность потока  $\Phi$ -переменной (3.16) будет добавлена «диффузионная» составляющая потока (3.15), в которой значение  $C_{\Phi,b}$  будет заменено на «диффузионное» сопротивление границы (подробнее об этом - в документе [1]). Такое граничное условие может оказаться эффективным при моделировании нестационарных пульсационных течений, в которых скорость потока на входной границе может падать до нуля.

### 3.5.4 Граничное условие второго рода

Для задания на границе известного значения полной плотности потока  $\Phi$ -переменной  $J_0$  (граничное условие *второго* рода) необходимо указать:

$$C_{\Phi,b} = BC\_FIXFLUX, V_{\Phi,b} = J_0$$

Как и в случае источников, при использовании этого условия для проницаемых границ ( $m_b \neq 0$ ) следует помнить, что полный поток  $J_0$ , задаваемый пользователем, включает в себя как диффузионную, так и конвективную составляющую.

### 3.5.5 Граничные условия на входных границах

Если на входной границе (через которую поток поступает в расчетную область) необходимо учесть только *конвективную* составляющую потока (3.15)

$$J_{\Phi,b} = \max(m_b, 0) \cdot V_{\Phi,b} - \max(-m_b, 0) \cdot \Phi_p, \quad (3.16)$$

то коэффициенту граничного условия следует присвоить значение

$$C_{\Phi,b} = BC\_MASSONLY$$

Отметим, что в подавляющем большинстве прикладных задач это граничное условие является наиболее подходящим для всех  $\Phi$ -переменных G-фазы.

**Замечание.** Внутренние алгоритмы пакета устроены таким образом, что граничные условия необходимо задавать как для плотности массового потока (граничное условие для уравнения неразрывности), так и для компонент вектора скорости (граничные условия для уравнений баланса импульса), хотя обычно эти параметры связаны друг с другом очевидным соотношением - плотность массового потока можно рассчитать как произведение плотности среды и нормальной компоненты вектора скорости. Однако возможны ситуации, когда плотность массового потока известна, а компоненты скорости - нет. Это случай задания давления на входной границе.

Для задания граничных условий для компонент вектора скорости *в случае заданного давления на входной границе* (когда значения компонент вектора скорости на границе не известны) следует использовать специальное граничное условие вида:

$$C_{\Phi,b} = BC\_BYMASS, V_{\Phi,b} = C_{cor}$$

Это условие эквивалентно условию с  $C_{\Phi,b} = BC\_MASSONLY$ , при этом значение  $V_{\Phi,b}$ , рассчитывается по уравнению

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

где  $\rho_{g,p}$  - плотность G-фазы в *приграничной* области.

Поясним использование этого условия на геометрии РО, изображенной на рис. 3.2.

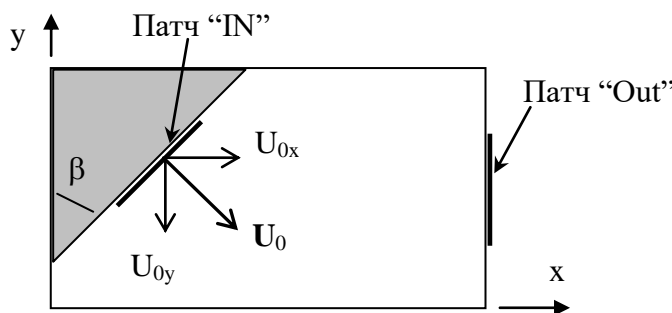


Рисунок 3.2 - Геометрия задачи

В этой задаче поток поступает в РО через границу, описываемую патчем "IN", на котором для давления задано граничное условие постоянного давления, равного 220 Па:

$$C_{p,b} = BC\_FIXPRESS, V_{p,b} = 220$$

Покидает РО поток через выходную границу (патч "Out") с нулевым противодавлением:

$$C_{p,b} = BC\_FIXPRESS, V_{p,b} = 0$$

Для задания граничных условий на входной границе "IN" для компонент вектора скорости  $U_0$  необходимо воспользоваться условиями:

$$C_{U_{gx,b}} = BC\_BYMASS, V_{U_{gx,b}} = \cos(\beta)$$

$$C_{U_{gy,b}} = BC\_BYMASS, V_{U_{gy,b}} = -\sin(\beta)$$

### 3.5.6 Граничные условия на выходных границах

На границах области решения, через которые поток *покидает* РО, для всех  $\Phi$ -переменных также рекомендуется использовать условие с  $C_{\Phi,b} = BC\_MASSONLY$ . Поскольку для данной границы  $\max(m_b, 0) = 0$  (иначе граница окажется не «выходной», а «входной»), то в соответствии с (2.16) значение  $V_{\Phi,b}$  не используется, и ему можно присвоить любое значение. В следующем пункте будет показано, что это соответствует граничному условию, задаваемому по умолчанию. Поэтому для выходной границы вообще нет необходимости задавать граничные условия.

### 3.5.7 Граничные условия по умолчанию

Если на непроницаемой границе ( $m_b = 0$ ) для  $\Phi$ -переменной не задано *явно* граничное условие (граничный источник), то по умолчанию задается «адиабатическое» условие

$$J_{\Phi,b} = 0$$

Если граничное условие не задано на *проницаемой* границе, то Решатель задаст внутренний граничный источник, равный

$$J_{\Phi,b} = \max(m_b, 0) \Phi_P - \max(-m_b, 0) \Phi_P = m_b \Phi_P, \quad (3.17)$$

где  $\Phi_P$  - значение  $\Phi$ -переменной вблизи границы. Заметим, что это именно то ГУ, которое наиболее подходяще для задания ГУ для  $\Phi$ -переменной на выходной границе.

В ряде случаев необходимо явно задать граничное условие по умолчанию (3.17). Это, например, требуется для работы функций расчета полных потоков [1]. Для явного задания такого ГУ необходимо использовать символьные значения:

$$C_{\Phi,b} = BC\_MASSONLY, V_{\Phi,b} = BC\_SAME.$$

В этом случае в Решателе в качестве  $V_{\Phi,b}$  будет использоваться значение  $\Phi_P$ .

Напомним еще раз. Если на проницаемой границе необходимо задать нулевой поток  $\Phi$ -переменной, то для этого необходимо использовать явное ГУ:

$$C_{\Phi,b} = BC\_FIXFLUX, V_{\Phi,b} = 0.$$

Для удобства задания граничных условий прилипания и непроницаемости (равенства нулю всех компонент вектора скорости) используется флаг граничного патча IsWALL. Если для такого патча не заданы явные ГУ, то для всех компонент вектора скорости Решатель Anes автоматически «генерирует» условия

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

#### ❶ Описание в проекте прикладной задачи :

Граничные условия для  $\Phi$ -переменной с именем "Ф-имя" на граничном патче с именем "Имя-патча" задаются парой операторов секции [Boundary Conditions]

COEF("Имя-патча.Ф-имя") = V-переменная

VAL("Имя-патча.Ф-имя") = V-переменная

### 3.5.8 Периодические граничные условия

Особым типом граничных условий являются *периодические граничные условия*. В коде Anes эти условия реализованы для ряда простых геометрий расчетной области.

Если РО имеет две идентичные границы в направлении осей обобщенных координат  $x, y$  или  $z$  и эти плоские границы перпендикулярны осям, то периодические ГУ сводятся к следующим соотношениям, например, для оси  $x$  (см. рисунок 1.6):

$$\Phi(x=0, y, z) = \Phi(x=L_x, y, z)$$

В коде эти условия применяются сразу для *всех* Ф-переменных. На границах периодичности нельзя задавать никакие другие граничные условия.

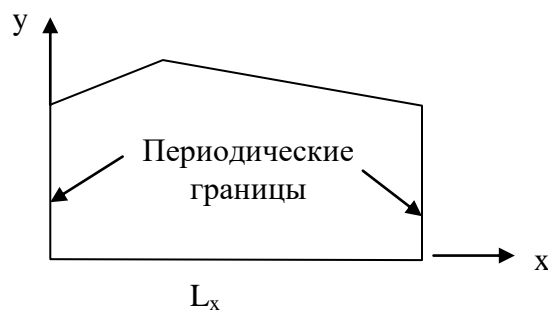


Рисунок 3.3 - Периодические ГУ вдоль оси  $x$

#### ❗ Описание в проекте прикладной задачи :

Для активации периодических ГУ вдоль координатных осей необходимо использовать операторы секции [Main] файла проекта:

```
IsCycleBCX = .TRUE.
IsCycleBCY = .TRUE.
IsCycleBCZ = .TRUE.
```

При использовании неструктурных сеток код Anes допускает особые периодические граничные условия – ПГУ на «наклонных» границах. Этот тип геометрии можно использовать, например, для моделирования теплообменных блоков, в основе которых лежат элементы в треугольной упаковке (см. рисунок 3.4). Для использования «наклонных» периодических границ РО должна удовлетворять следующему условию: рассматривается трехмерная задача в декартовой системе координат с постоянной формой сечения (внешней границей РО) в направлении оси  $z$  в виде шестигранника. В этом случае можно задать периодические граничные условия между следующими участками границы:

$Y_1 - Y_2$  – периодические границы вдоль оси  $y$ ,  
 $XY_a - XY_c$  – первые наклонные периодические границы,  
 $XY_b - XY_d$  – вторые наклонные периодические границы.

#### ❗ Описание в проекте прикладной задачи :

Для активации наклонных периодических ГУ необходимо использовать операторы секции [Main] файла проекта:

```
IsCycleBCX = .TRUE.
IsCycleBCY = .TRUE.
IsCycleBC_XY = .TRUE.
```

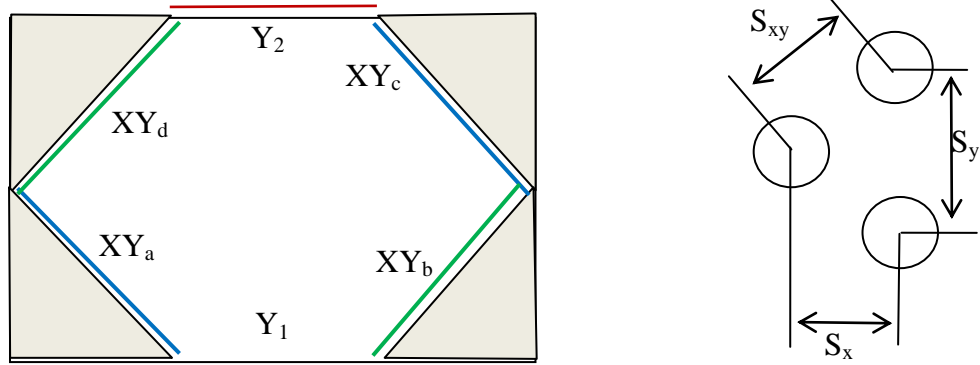


Рисунок 3.4 – Наклонные периодические ГУ

### 3.6 Начальные условия

При решении нестационарных задач наряду с граничными условиями необходимо задать распределения всех  $\Phi$ -переменных в начальный момент времени -  $\Phi(\mathbf{x}, \tau = 0)$ . В коде Anes предусмотрены различные алгоритмы задания начального распределения, они описаны в документе [2].



#### 4. Математические модели G-фазы

Ф-переменная в коде Anes идентифицируется своим уникальным именем (от 1 до 16 символов). Список predefined Ф-переменных, связанных с G-фазой, приведен в таблице 4.1.

Таблица 3.1 - Predefined Ф-переменные G-фазы

Имя	Ф	Назначение
PF	$P_f$	давление
UGX	$U_{gx}$	x-компонента скорости
UGY	$U_{gy}$	y-компонента скорости
UGZ	$U_{gz}$	z-компонента скорости
TG	$T_g$	температура
kTUR	k	кинетическая энергия турбулентных пульсаций
epsTUR	$\varepsilon$	скорость турбулентной диссипации в модели k- $\varepsilon$
omTUR	$\omega$	скорость турбулентной диссипации в модели k- $\omega$
C1	$C_1$	массовая доля первого компонента
...	...	...
C8	$C_8$	массовая доля восьмого компонента
wDIS	$W_d$	вспомогательная переменная для расчета расстояния до твердой стенки

Основными Ф-переменными G-фазы являются давление  $P_f$  и компоненты вектора скорости ( $U_{gx}, U_{gy}, U_{gz}$ ). При решении задач *теплообмена* дополнительно используется температура фазы  $T_g$ . Для моделирования *химических процессов* в G-фазе вводятся Ф-переменные, представляющие собой массовые доли компонентов. В этом случае G-фаза рассматривается как гомогенная смесь  $N_C = 2 \dots 8$  компонентов (Ф-переменные  $C_1 - C_8$ ), поля которых рассчитываются из соответствующих уравнений баланса массы:

$$C_k = \frac{M_k}{\sum_1^{N_C} M_k}, \quad k = 1 \dots N_C \quad (4.1)$$

где  $M_k$  - масса компонента в элементарном объеме. Если компоненты не используются, то предполагается, что G фаза состоит из одного первого компонента.

При использовании двухпараметрических моделей *турбулентности* для описания G-фазы используются еще две Ф-переменные – кинетическая энергия турбулентных пульсаций (далее турбулентная энергия) k и скорость турбулентной диссипации  $\varepsilon$  или диссипативная функция  $\omega$ .

Ф-переменная  $W_d$  в коде используется для расчета поля  $L_{dis}$ , которое содержит расстояние от данной точки пространства до ближайшей твердой границы PO.

#### ❗ Описание в проекте прикладной задачи :

Использование Ф-переменной в задаче определяется оператором

Used(Ф-Имя1,...)

секции [PHI Variables] файла проекта. Этот оператор выделяет память для переменной в Решателе, однако для включения Ф-переменной в список рассчитываемых переменных нужно указать оператор

Solve(Ф-Имя) = .TRUE./.FALSE.

той-же секции.

#### 4.1 Уравнение неразрывности

В коде Anes уравнение сохранения массы G-фазы (уравнение неразрывности) рассматривается как уравнение для давления  $P_f$ . Уравнение неразрывности при наличии в ПО G- и S- фаз имеет вид

$$\frac{\partial(\varphi\rho_g)}{\partial\tau} + \operatorname{div}(\varphi\rho_g\mathbf{U}_g) = M_p^{(l)} + \varphi M_p \quad (4.1)$$


где  $\rho_g$  - плотность G-фазы,  $\mathbf{U}_g$  - вектор скорости G-фазы,  $\varphi$  - пористость. Источник  $M_p^{(l)}$  описывает внутренние объемные источники массы (автоматически рассчитываемые Решателем Anes), связанные с массообменом между S- и G-фазами в зонах Porous -  $M_p^{(gs)}$ . Вид этих источников зависит от модели межфазного взаимодействия и подробно описан в главе 4. Границы раздела G-S фаз (патчи типа "FS\_") считаются непроницаемыми, поэтому никаких внутренних поверхностных источников на них не задаются, однако пользователь может связать с границей раздела свои собственные источники массы.

Источниковый член  $M_p$  описывает источники, определяемые пользователем. Принципы задания таких источников описаны в разделе 3.4.2. Заметим только, что эти источники относятся к  $1 \text{ м}^3$  чистой G-фазы, а не к  $1 \text{ м}^3$  пространства, занятого G- и S- фазами.

Важно отметить, что архитектура кода устроена таким образом, что давление  $P_f$  входит только в источниковые члены уравнений сохранения в виде производных по координатам и времени, поэтому оно определено с точностью до постоянной величины. При решении большинства задач обычно выбирается точка в расчетной области, в которой полагается  $P_f = 0$ . Если силы плавучести не учитываются, то статическое давление  $p$  (необходимое, например, для расчета теплофизических свойств G-фазы) можно рассчитать по соотношению (это соотношение «согласовано» с источниками (4.5b)):

$$\begin{aligned} P_f &= p - p_0 + \frac{2}{3}\rho_g k - \frac{\mu_g}{3}\operatorname{div}(\mathbf{U}_g), \\ p &= P_f + p_0 + p_T, \\ p_T &= \frac{1}{3}\rho_g(v_g D_U - 2 \cdot k), \\ D_U &= \operatorname{div}(\mathbf{U}_g) \end{aligned} \quad (4.2)$$

где  $k$  – удельная кинетическая энергия турбулентных пульсаций,  $p_0$  – постоянный отсчет давления,  $p_T$  – диагональная составляющая тензора напряжений Ньютона и тензора турбулентных напряжений. Обычно член  $p_T$  мал по сравнению с  $p_0$ , и в большинстве задач им можно пренебречь. В этом случае  $p_0$  можно рассматривать как статическое термодинамическое давление в точке с  $P_f = 0$ .

 **Замечание.** Выбор «отсчета» давления  $p_0$  зависит от класса решаемой задачи. При моделировании дозвуковых задач перепад давления  $P_f$  составляет величину порядка сотен Па. Использование в этом случае  $P_f = p$  ( $p_0 = 0$ ) может приводить к большим ошибкам округления. В этом случае удобнее всего перейти в  $P_f$  к относительному давлению: из  $p$  «вычесть» статическое «окружающее» давление, которое составляет величину порядка  $10^5$  Па. При моделировании трансзвуковых и сверхзвуковых течений перепад давления  $P_f$  того же порядка, что и статическое «атмосферное» давление. В этом случае можно положить  $P_f = p$  ( $p_0 = 0$ ).

#### Описание в проекте прикладной задачи :

Значение отсчета давления задается оператором секции [Properties] проекта:

P0 = значение

Точка, в которой давление PF полагается равным нулю, определяется операторами секции [Solver]

XpfZERO = значение

YpfZERO = значение

ZpfZERO = значение

Если на какой-то границе задано ГУ для давления, отличное от BC\_FIXMASS, то эти операторы игнорируются, поскольку отсчет давления определен ГУ.

#### 4.1.1 Граничные условия для давления

Граничные условия для уравнения неразрывности связаны с граничными условиями для давления  $P_f$ . Они подробно описаны в разделе 3.5.1. Опишем еще две разновидности граничного условия заданного давления на границе:

- 1) задание давления торможения,
- 2) задание противодействия.

При решении задач для сжимаемых сред (околозвуковые и сверхзвуковые течения) возникает ситуация, когда как на входной, так и на выходной границе необходимо задавать давление. В этом случае возможна расходимость итерационного процесса решения из-за задания входного давления.

Для решения этой проблемы в коде Anes были реализованы входные граничные условия для давления, основанные на использовании давления торможения. Поясним этот алгоритм на примере слабо сжимаемой среды ( $\rho_g \approx \text{const}$ ). Пусть входная граница расчетной области расположена на границе бака с большим объемом, среда в котором в целом покоится. В этом случае можно связать параметры на входной границе (индекс «in») с параметрами в баке (индекс «0») на основе уравнения Бернулли

$$\frac{p_0 - p_{in}}{\rho_0} = \frac{1}{2} u_{in}^2,$$


которое позволяет получить выражение для плотности массового потока на входной границе в виде:

$$m_{in} = \rho_{in} u_{in} = \sqrt{2\rho_0 (p_0 - p_{in})}$$

Как показали численные эксперименты, это граничное условие позволяет получить сходимость даже для задач с ударными волнами в каналах переменного сечения.

Для реализации этого граничного условия необходимо задать коэффициент  $C_{p,b}$  отрицательным и равным значению плотности  $\rho_0$ :

$$C_p = -\rho_0$$

 **Замечание.** В текущей версии граничное условие заданного входного давления торможения возможно только для неструктурных сеток!

Второй тип граничного условия задается значением коэффициента  $C_{p,b} = \text{BC\_OUTPRESS}$ . Это условие аналогично условию BC\_FIXPRESS, главное отличие условия BC\_OUTPRESS заключается в том, что коэффициент  $C_{p,b}$  рассчитывается внутри Решателя, а не задается равным большому числу ( $10^3$ ). Подробнее этот алгоритм описан в разделах (3.4.4) и (4.5) документа [1].

Какое граничное условие применять (BC\_FIXPRESS или BC\_OUTPRESS) зависит от прикладной задачи. Граничное условие BC\_FIXPRESS дает более быструю сходимость итерационного процесса решения, однако может приводить к искажению полей вблизи выходной границы (в приграничном слое ячеек). При моделировании турбулентных течений эти искажения могут быть особенно заметными. Типичный пример - скачок температуры стенки в двух выходных КО при заданной плотности теплового потока на стенке. При использовании граничного условия BC\_OUTPRESS таких искажений нет. Однако в ряде задач это граничное условие может замедлить сходимость итерационного процесса.

#### 4.2 Уравнения движения

Компоненты вектора скорости G-фазы  $U_{gi}$  ( $i = x, y, z$ ) удовлетворяют уравнению

$$\frac{\partial(\varphi\rho_g U_{gi})}{\partial\tau} + \operatorname{div}(\varphi\rho_g \mathbf{U}_g U_{gi} - \varphi\mu_{\text{eff}} \nabla U_{gi}) = -\varphi(\nabla P_f)_i + S_{U_{gi}}^{(1)} + \varphi S_{U_{gi}} \quad (4.3)$$

Эффективный коэффициент вязкости  $\mu_{\text{eff}}$  рассчитывается в соответствии с первой формой соотношения (3.5):

$$\mu_{\text{eff}} = \rho_g (v_g + v_t) \quad (4.4)$$

где  $v_g$  - коэффициент кинематической вязкости G-фазы,  $v_t$  - коэффициент турбулентной кинематической вязкости, определяемый моделью турбулентности.

Член  $S_{U_{gi}}$  является любым объемным источниковым членом, задаваемым пользователем. Этот источник описывает производство импульса, отнесенное к  $1 \text{ м}^3$  чистой G-фазы ( $\text{н/м}^3$ ). Принципы описания таких источников содержатся в разделе (3.4.3).

Источниковые члены  $S_{U_{gi}}^{(1)}$ , называемые далее «внутренними», связаны с особенностями уравнений сохранения импульса и состоят из *четырёх* слагаемых.

#### 4.2.1 Источники, связанные с цилиндрической системой и тензором напряжений

Первая составляющая появляется, например, в цилиндрической системе координат и обусловлена ее криволинейностью (напомним, что  $x - r$ ,  $y - \theta$ ,  $z - z$ ):

$$\begin{aligned} S_{U_{gx}}^{(1)} &= \varphi \left( \rho_g \frac{U_{gy}^2}{x} - \mu_{\text{eff}} \frac{U_{gx}}{x^2} - \frac{2\mu_{\text{eff}}}{x^2} \frac{\partial U_{gy}}{\partial y} \right), \\ S_{U_{gy}}^{(1)} &= \varphi \left( -\rho_g \frac{U_{gx} U_{gy}}{x} - \mu_{\text{eff}} \frac{U_{gy}}{x^2} + \frac{2\mu_{\text{eff}}}{x^2} \frac{\partial U_{gx}}{\partial y} \right), \\ S_{U_{gz}}^{(1)} &= 0 \end{aligned} \quad (4.5a)$$

Вторая причина появления первой составляющей - взаимосвязь тензора касательных напряжений и тензора скоростей деформаций (закон Ньютона) и соотношения для тензора турбулентных напряжений. При переменных свойствах она не сводится к стандартному «градиентному» виду. В связи с этим, появляются дополнительные источники, например для декартовой системы координат

$$\begin{aligned} S_{U_{gx}} &= \frac{\partial}{\partial y} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gy}}{\partial x} \right) - \frac{\partial}{\partial x} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gy}}{\partial y} \right) + \frac{\partial}{\partial z} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gz}}{\partial x} \right) - \frac{\partial}{\partial x} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gz}}{\partial z} \right), \\ S_{U_{gy}} &= \frac{\partial}{\partial x} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gx}}{\partial y} \right) - \frac{\partial}{\partial y} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gx}}{\partial x} \right) + \frac{\partial}{\partial z} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gz}}{\partial y} \right) - \frac{\partial}{\partial y} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gz}}{\partial z} \right), \\ S_{U_{gz}} &= \frac{\partial}{\partial x} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gx}}{\partial z} \right) - \frac{\partial}{\partial z} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gx}}{\partial x} \right) + \frac{\partial}{\partial y} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gy}}{\partial z} \right) - \frac{\partial}{\partial z} \left( \varphi\mu_{\text{eff}} \frac{\partial U_{gy}}{\partial y} \right) \end{aligned} \quad (4.5b)$$

Эти дополнительные источниковые члены равны нулю, если эффективная вязкость постоянна. При моделировании турбулентных течений эффективная вязкость меняется в широких пределах, однако члены (4.5b) для большинства задач малы по сравнению с градиентом давления и архимедовыми силами. Поэтому по умолчанию эти члены полагаются равными нулю. Для активации их расчета необходимо использовать оператор секции [Special Data]:

`L("IsIntStressSource") = .True.`

В текущей версии эти члены рассчитываются только для неструктурной сетки в декартовой системе координат.

### 4.2.2 Источники, связанные с силой плавучести

Вторая составляющая связана с учетом сил Архимеда или сил плавучести. В векторном представлении эту силу, отнесенную к 1 м<sup>3</sup>, можно записать в виде:

$$\mathbf{S}_{ug} = \varphi \rho_g \mathbf{g} \quad (4.6)$$

где  $\mathbf{g}$  - вектор ускорения свободного падения. В коде Anes предусмотрены два варианта автоматического расчета этого источника - с использованием реальной плотности среды или по соотношениям Буссинеска.

В *первом варианте*, сила плавучести для *декартовой* системы рассчитывается по соотношениям:

$$S_{Ugx}^{(I)} = \varphi g_{cx} (\rho_g - \rho_{арх}), \quad S_{Ugy}^{(I)} = \varphi g_{cy} (\rho_g - \rho_{арх}), \quad S_{Ugz}^{(I)} = \varphi g_{cz} (\rho_g - \rho_{арх}), \quad (4.7)$$

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

$\rho_{арх}$  - характерное постоянное значение плотности.

В цилиндрической системе координат

$$\begin{aligned} S_{Ugx}^{(I)} &= \varphi g_x (\rho_g - \rho_{арх}), & g_x &= g_{cx} \cos(y) + g_{cy} \sin(y) \\ S_{Ugy}^{(I)} &= \varphi g_y (\rho_g - \rho_{арх}), & g_y &= -g_{cx} \sin(y) + g_{cy} \cos(y) \\ S_{Ugz}^{(I)} &= \varphi g_{cz} (\rho_g - \rho_{арх}) \end{aligned} \quad (4.8)$$

При учете сил плавучести по первому варианту связь между статическим давлением  $p$  и  $P_f$  приобретает вид:

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

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

Смысл этого изменения давления легко понять, если положить  $p_0 = 0$  и считать, что движения среды нет (гидростатическое равновесие среды). В этом случае поле статического давления  $p$  линейно возрастет в направлении вектора силы тяжести, а поле  $P_f$  останется постоянным. Таким образом, при использовании (4.8) давление  $P_f$  - это статическое давление за вычетом гидростатического столба с постоянной плотностью  $\rho_{арх}$ .

Во *втором* варианте задания сил плавучести используется приближение Буссинеска, которое для декартовой системы координат можно записать в виде

$$\begin{aligned} S_{Ugx}^{(I)} &= -\varphi g_{cx} \rho_g \beta_T (T_g - T_{арх}), & \beta_T &= -\frac{1}{\rho_g} \left( \frac{\partial \rho_g}{\partial T_g} \right)_p, \\ S_{Ugy}^{(I)} &= -\varphi g_{cy} \rho_g \beta_T (T_g - T_{арх}), \\ S_{Ugz}^{(I)} &= -\varphi g_{cz} \rho_g \beta_T (T_g - T_{арх}) \end{aligned} \quad (4.9)$$

здесь  $T_{арх}$  - характерное постоянное значение температуры G-фазы,  $\beta_T$  - изобарический коэффициент термического расширения.

Характерные значения плотности  $\rho_{арх}$  или температуры  $T_{арх}$  в коде Anes рассматриваются как произвольные постоянные (по умолчанию они равны нулю). Причина использования этих характерных значений заключается в следующем. Для большинства задач, в которых силы плавучести играют заметную роль, изменение плотности (или температуры) мало по сравнению с характерными значениями плотности и температуры. Поэтому возможно появление ошибок округления при вычислении источникового члена. «Вычитание» характерных значений позволяет существенно уменьшить эти ошибки. При этом следует помнить, что давление  $P_f$  - это давление за вычетом гидростатического столба.

### 4.2.3 Источники, связанные с межфазным трением в Poreous зонах

Третья составляющая внутренних источниковых членов описывает межфазный обмен импульсом между G- и S- фазами в Poreous-зонах  $S_{U_{gi}}^{(gs)}$ . Поскольку S-фаза неподвижна, то этот источник равен силе сопротивления, отнесенный к 1 м<sup>3</sup> пространства, при течении G-фазы в проницаемом скелете S-фазы. В коде для моделирования этих источников используются соотношения

$$S_{U_{gi}}^{(gs)} = -\max(-M_p^{(gs)}, 0) \cdot U_{gi} - \varphi \frac{\rho_g v_g}{k_i} U_{gi}, \quad i = x, y, z \quad (4.10)$$

Здесь  $k_i$  - коэффициент *проницаемости* пористой структуры в направлении  $i$ -ой координаты, определяемый моделью G-S взаимодействия (см. главу 8).

### 4.2.4 Межфазное трение на границах раздела G- и S-фаз

Четвертая составляющая внутренних источниковых членов описывает межфазный обмен импульсом между G- и S- фазами на явно выделенных границах раздела G-и S-фаз. Эти границы в коде описываются поверхностными патчами с именами "FS\_..." и рассматриваются как непроницаемые границы с прилипанием. Для них автоматически формируются ГУ

$$U_{gx} = U_{gy} = U_{gz} = 0 \quad (4.11)$$

#### **i** Описание в проекте прикладной задачи :

Параметры источникового члена, описывающего силы плавучести определяется операторами секции [Internal Source] :

RhoArx = значение : *характерное значение плотности*

Tagx = значение : *характерное значение температуры*

ArxForce = ARX\_User/ARX\_RHO/ARX\_T

AgravX = значение : *декартовы составляющие вектора силы тяжести*

AgravY = значение

AgravZ = значение

### 4.2.5 Граничные условия

Для компонент вектора скорости граничные условия задаются точно так же, как и для обычной Ф-переменной (см. раздел 3.5). Подчеркнем лишь некоторые особенности.

На входной границе необходимо задавать «избыточные» на первый взгляд граничные условия: для плотности массового потока (ГУ для  $P_f$ ) и компонент вектора скорости (ГУ для  $U_{gx}$ ,  $U_{gy}$ ,  $U_{gz}$ ). На самом деле это принципиально разные условия. При задании ГУ для компонент скорости задаются плотности потока импульса, которые могут быть не связаны «линейно» с плотностью массового потока.

Если на входной границе для давления задается давление (ГУ с  $C_{p,b} = VC\_FIXPRESS$ ), то для задания неизвестной нормальной к границе компоненты скорости нужно использовать ГУ с  $C_{\phi,b} = VC\_VUMASS$  (см. пункт 3.5.5).

### 4.2.6 Моделирование стабилизированного течения

Стабилизированное течение соответствует случаю, когда компоненты вектора скорости не зависят от координаты  $z$ , а давление линейно зависит от  $z$ . Поле давления можно представить в виде

$$P_f = \left( \frac{dP}{dz} \right)_0 z + p_f(x, y)$$

где  $(dP/dz)_0$  - постоянный градиент давления в направлении оси  $z$ ,  $p_f$  – относительное поле давления в поперечном сечении.

В этом случае уравнение для скорости  $U_{gz}$  превращается в «обычное» уравнение переноса (например, для декартовой системы координат):

$$\frac{\partial}{\partial x} \left( \varphi \rho_g U_{gx} U_{gz} - \varphi \mu_{\text{eff}} \frac{\partial U_{gz}}{\partial x} \right) + \frac{\partial}{\partial y} \left( \varphi \rho_g U_{gy} U_{gz} - \varphi \mu_{\text{eff}} \frac{\partial U_{gz}}{\partial y} \right) = -\varphi \left( \frac{dP}{dz} \right)_0 + S_{U_{gz}}^{(i)} + \varphi S_{U_{gz}} \quad (4.12)$$

Вид уравнений для компонент скорости  $U_{gy}$  и  $U_{gx}$  и для  $p_f$  при этом удовлетворяют уравнениям (4.3). Эти компоненты могут быть отличны от нуля при наличии ненулевых источников. Важно отметить, что при решении подобных задач все поля не должны зависеть от координаты  $z$ , а свойства должны быть постоянны.

В текущей версии кода модель стабилизированного течения реализована только для структурных сеток. При использовании неструктурных сеток для моделирования необходимо использовать модель, описанную ниже.

Для задания градиента давления  $(dP/dz)_0$  код представляет две возможности:

- 1) задается собственно градиент давления,
- 2) задается полный расход среды через канал (через  $x$ -у сечение расчетной области).

Выбор модели определяется оператором секции [Internal Source] файла проекта

```
DEV_Mode = devNONE/devMASSRATE/devDPDZ
```

При использовании режима devDPDZ пользователь должен задать градиент давления  $(dP/dz)_0$  с помощью оператора

```
DEV_DPDZ = <постоянное значение>
```

При использовании режима devMASSRATE пользователь должен задать полный расход через канал

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

с помощью оператора

```
DEV_MassRateZ = значение
```

Параметры  $(dP/dz)_0$  и  $G_0$  однозначно связаны друг с другом. При задании одной из них, вторая рассчитывается автоматически. Значения этих переменных выводятся в листинг расчета в виде:

```
Стабилизированное течение: dP/dZ= 4.0311E-04 Pa/m
Стабилизированное течение: FlowRate= 3.6716E-04 ks/s
```

Для удобства обработки результатов в Решателе определены две переменные, которые можно использовать при работе с любыми  $V$ -переменными:

```
DEV_DPDZ          - это  $(dP/dz)_0$ ,
DEV_FLOWRATE     - это  $G_0$ .
```

Если пользователь укажет в секции [Special Data] оператор

```
L("CreateDEVVEL") = .TRUE.,
```

то по окончании расчета *одномерной структурной задачи* Решатель создаст текстовые файлы с полями используемых переменных с именами:

```
<Префикс файлов результатов>_pf.dat
<Префикс файлов результатов>_ugz.dat
<Префикс файлов результатов>_ktur.dat
<Префикс файлов результатов>_epstur.dat
<Префикс файлов результатов>_tg.dat
```

Формат этих файлов соответствует формату файлов `tuTABLE V`-переменных (см. [2]), что позволяет использовать эти файлы для задания входных граничных условий для многомерных задач.

При решении *двумерной структурной задачи* о стабилизированном течении также создаются текстовые \*.dat файлы, но другой структуры. Формат их записи в Решателе следующий:

```
write(LU,'(2i10)') NX,NY
write(LU,'(6(1pe13.5))') (X(ixC),ixc=1,NX)
write(LU,'(6(1pe13.5))') (Y(iyC),iyc=1,NY)
write(LU,'(6(1pe13.5))') ((F (ixc,iyc),ixc=1,NX),iyc=1,NY)
```

Эти файлы можно использовать для задания 2D граничных условий для 3D задач. Отметим, что подобные файлы используются в LES моделировании для задания начального распределения (см. раздел 2.4 документа `aneLESmod21.pdf`).

При задании расхода через канал  $G_0$  градиент давления рассчитывается в итерационном процессе с использованием нескольких алгоритмов. Выбор алгоритма определяется операторами секции [Special Data]. Подробности алгоритмов приведены в документе [1].

#### **i** Описание в проекте прикладной задачи :

В пакете для расчета полностью развитого течения на структурной сетке необходимо использовать один КО по оси Z (NZ=1).

Модель расчета градиента давления определяется операторами секции [Internal Source]

```
DEV_Mode = devNONE/devMASSRATE/devDPDZ
DEV_DPDZ = значение
DEV_MassRateZ = значение
```

При использовании режима `devMASSRATE` алгоритм расчета градиента давления определяется операторами секции [Special Data]

```
C("ModelMassRate") = "DivDP" / "Newton" / "DBS" / "SIMPLE"
```

### 4.2.7 Развитое периодическое течение

При использовании *неструктурных* сеток для моделирования развитых течений вдали от входа в каналах можно использовать модель периодического течения. В этой модели предполагается, что компоненты поля скорости удовлетворяют периодическим условиям по оси z:

$$\Phi(x,y,0) = \Phi(x,y,L_z),$$

а поле давления можно разделить на линейную по z и периодическую составляющие

$$P_f = p_f + \left( \frac{dp}{dz} \right)_0 z, \quad p_f(x,y,0) = p_f(x,y,L_z)$$

Типичный пример такой задачи – течение в канале (или в трубе) с волнистой стенкой. Отметим, что эту модель можно использовать для расчета стабилизированного течения, описанного выше в разделе 4.2.6. В этом случае для моделирования нужно задать два или более КО по оси z. *Заметим, что в ряде задач для сходимости нужно использовать минимум три КО по оси периодичности!*




Для описания таких задач используются те же операторы, что и для полностью развитого течения:

```
DEV_Mode = devNONE/devMASSRATE/devDPDZ
DEV_DPDZ = <значение>
DEV_MassRateZ = <значение>
.....
C("ModelMassRate") = "DivDP" / "Newton" / "DBS" / "SIMPLE"
```

Единственное отличие – необходимо задать периодические граничные условия по оси z с помощью оператора секции [Main] файла проекта:

```
IsCycleBCZ = .TRUE.
```

 **Замечание.** При решении некоторых задач процесс сходимости уравнения для  $U_{gz}$  носит колебательный характер в случае заданного расхода ( $DEV\_Mode = devMASSRATE$ ). Для улучшения сходимости можно использовать релаксацию для расчета  $(dp/dz)_0$ . Коэффициент релаксации можно задать с помощью оператора секции [Special Data]:

```
R("Relax.DPDZ") = 1.0
```

#### 4.2.8 Моделирование сжимаемых течений

В Решателе кода Anes реализованы численные алгоритмы, позволяющие рассчитывать течения с плотностью, зависящей от давления  $p$ . К этой категории относятся как трансзвуковые и сверхзвуковые течения, так и дозвуковые течения при наличии объемных источников массы и импульса (например – процессы адсорбции водорода в пористых засыпках металлгидридных сплавов).

Система уравнений для несжимаемых, сжимаемых течений и течений с переменными свойствами имеет один и то же вид (3.1) и (3.3). Однако поведение этой системы зависит от характерного значения числа Маха:

$$M = \frac{U}{a_s}, \quad a_s = \sqrt{\frac{1}{\left(\frac{\partial \rho}{\partial p}\right)_s}}$$

Здесь  $U$  - модуль вектора скорости,  $a_s$  - скорость звука, для адиабатических течений частная производная от плотности по давлению рассчитывается при постоянной энтропии  $s$ .

Для существенно дозвуковых течений ( $U \ll a_s$ ) влияние давления на плотность среды не существенно. Заметим, что это связано не с функциональной зависимостью плотности от давления, а с малыми перепадами давления по сравнению с собственно давлением

$$\Delta p \sim \frac{\rho U^2}{2} \ll p.$$

При трансзвуковых ( $M \sim 1$ ) и сверхзвуковых ( $M > 1$ ) течениях изменение давления в потоке сравнимо с самим давлением, что приводит к возникновению новых физических эффектов. Основной эффект заключается в том, что в областях с  $M > 1$  любые возмущения *не могут* распространяться против течения. Для моделирования этого (и других) эффектов в коде Anes реализована «сжимаемая» форма численного алгоритма SIMPLE. Этот алгоритм подробно описан в [1], поэтому в данном документе приведем лишь некоторые идеи этого алгоритма.

В коде Anes для численного решения уравнений переноса используется идея итерационного нахождения поправок для  $\Phi$ -переменных. На каждой итерации ищется решение не для  $\Phi$ -переменной, а для ее поправки  $\delta\Phi$  (символ \* означает значение с предыдущей итерации):

$$\Phi = \Phi^* + \delta\Phi.$$

В «несжимаемой» версии алгоритма SIMPLE уравнение для поправки давления  $\delta p$  получается следующим образом. Используя уравнения движения, можно найти связь между поправкой скорости и давления (это главная «изюминка» алгоритма SIMPLE):

$$\delta \mathbf{U} = - \left( L \frac{\partial \mathbf{U}}{\partial p} \right) \nabla (\delta p) = -a_U \nabla (\delta p)$$

где  $L$  - параметр с размерностью длины. Используя это соотношение выражение для плотности массового потока можно представить в виде:

$$\rho \mathbf{U} = \rho^* \mathbf{U}^* + \rho \delta \mathbf{U} = \rho^* \mathbf{U}^* - \rho a_U \nabla (\delta p) \quad (4.13)$$

Если подставить это выражение в уравнение неразрывности (3.1), то можно получить уравнение для поправки давления в виде уравнения Пуассона:

$$\operatorname{div} [\rho a_U \nabla (\delta p)] = -\operatorname{div} (\rho^* \mathbf{U}^*)$$

Это типичное уравнение эллиптического типа, поэтому возмущения в потоке будут распространяться вдоль любых направлений.

В «сжимаемой» версии SIMPLE в соотношении (4.13) учитывается зависимость плотности от давления:

$$\rho \mathbf{U} = \rho^* \mathbf{U}^* + \rho^* \delta \mathbf{U} + \mathbf{U}^* \frac{\partial \rho}{\partial p} \delta p = \rho^* \mathbf{U}^* + \mathbf{U}^* \frac{\partial \rho}{\partial p} \delta p - \rho^* a_U \nabla (\delta p) = \rho^* \mathbf{U}^* + \frac{\mathbf{U}^*}{U^2} M^2 \delta p - \rho^* a_U \nabla (\delta p)$$

Используя это соотношение уравнение для поправки давления можно записать в следующем виде:

$$\operatorname{div} \left[ \frac{\mathbf{U}^*}{U^2} M^2 \delta p - \rho^* a_U \nabla (\delta p) \right] = -\operatorname{div} (\rho^* \mathbf{U}^*)$$

В этом уравнении наряду с «диффузионным» слагаемым  $-\operatorname{div} [\rho^* a_U \nabla (\delta p)]$  появилось конвективная составляющая, которая принципиально меняет характер распространения возмущений давления. С ростом числа Маха распространение возмущений против потока резко уменьшается.

В сверхзвуковых стационарных течениях невязкого газа (уравнениях газовой динамики) при  $M > 1$  уравнения гидродинамики являются гиперболическими. Параболические и гиперболические уравнения близки по распространению возмущений (обладают характеристиками), но не идентичны. Однако как показывают численные эксперименты, «сжимаемая» форма алгоритма SIMPLE позволяет описать практически все эффекты сверхзвуковых режимов (ударные волны, волны разрежения, нерасчетные режимы при истечении из сопла Ловала и т.д.).

Для активации «сжимаемого» режима в Решателе нужно задать для материала G-фазы новое свойство - коэффициент сжимаемости:

$$\beta_p = \left( \frac{\partial \rho_g}{\partial p} \right)_T,$$

в котором производная берется при постоянной температуре и, естественно, «ввести» в алгоритм расчета зависимость плотности от статического давления ( $P_f + p_0$ ).

В текущей версии алгоритм «параболизации» уравнения для поправки давления (см. [1]) реализован для неструктурной сетки.

#### **ⓘ Описание в проекте прикладной задачи :**

Использование «сжимаемой» или «несжимаемой» формы уравнения для поправки давления определяется значением свойства "BetaP", если оно равно нулю или не задано для материала G-фазы, то используется «несжимаемая» форма.

Для активации алгоритма «параболизации» ударных волн необходимо задать оператор в секции [Special Data]:

L("ComprsPF") = .TRUE.

### 4.3 Уравнение энергии для однокомпонентной G-фазы

В случае, когда не используются уравнения для концентраций  $C_1 - C_8$ , G-фаза рассматривается как однокомпонентная среда. В этом случае тепловое состояние G-фазы описывается уравнением энергии

$$\frac{\partial(\varphi \rho_g H_g)}{\partial \tau} + \text{div}(\varphi \rho_g \mathbf{U}_g H_g - \varphi \lambda_{\text{eff}} \nabla T_g) = S_{T_g}^{(1)} + \varphi S_{T_g}. \quad (4.14)$$

Здесь  $H_g$  - удельная энтальпия G-фазы, которая в коде рассматривается как функция от температуры и давления (V-переменная),  $S_{T_g}$  - объемный источниковый член, задаваемый пользователем, отнесенный к единице объема, занятой G-фазой.

Член  $S_{T_g}^{(1)}$  описывает внутренние источниковые члены в уравнении энергии:


- 1) работу сил давления,
- 2) вязкую диссипацию,
- 3) межфазное взаимодействие в Rogous-зонах,
- 4) сопряженный теплообмен на границе раздела G-S фаз.

Для расчета эффективного коэффициента теплопроводности  $\lambda_{\text{eff}}$  можно использовать две формы соотношения (3.5):

$$\lambda_{\text{eff}} = \rho_g c_{pg} \left( \frac{v_g}{Pr} + \frac{v_t}{Pr_t} \right) \quad \text{или} \quad \lambda_{\text{eff}} = \left( \lambda_g + \frac{\rho_g c_{pg} v_t}{Pr_t} \right) \quad (4.15)$$

Здесь  $\lambda_g$  - коэффициент теплопроводности G-фазы,  $c_{pg}$  - удельная изобарная теплоемкость G-фазы,  $Pr$  и  $Pr_t$  - молекулярное и турбулентное числа Прандтля:

$$c_{pg} = \left( \frac{\partial H_g}{\partial T_g} \right)_p, \quad Pr = \frac{v_g \rho_g c_{pg}}{\lambda_g}$$

 **Замечание.** В коде Anes используется уравнение энергии, записанное через энтальпию и температуру, а не через полную энтальпию (энтальпию «торможения»)

$$H_g + \frac{U_{gx}^2 + U_{gy}^2 + U_{gz}^2}{2}.$$

Об этом следует помнить при моделировании сжимаемых трансзвуковых и сверхзвуковых течений. В этом случае обязательно необходимо активировать расчет работы сил давления.

#### 4.3.1 Энтальпия G-фазы

В уравнение энергии (4.13) входят две переменные, играющие роль Ф-переменных – энтальпия  $H_g$  и температура  $T_g$ . В коде Anes в качестве Ф-переменной для уравнения энергии используется температура фазы  $T_g$ , а энтальпия фазы  $H_g$  рассматривается как V-переменная.

Поскольку энтальпия определена с точностью до произвольной постоянной, то можно записать

$$H_g = h_g(T_g) + H^0 + \int_{p_0}^p \frac{\partial H_g}{\partial p} dp, \quad (4.16)$$

где

$$h_g = \int_{T_0}^T c_{pg} dT \quad - \text{тепловая составляющая энтальпии,}$$

$H^0$  - постоянная, равная энтальпии компонента при  $T=T_0$ ,  $p = p_0$ . Последняя составляющая в правой части (4.16) не учитывается в коде Anes, поскольку она пренебрежимо мала для большинства теплоносителей (газов и капельных жидкостей).

Если нет химических превращений, то значение  $H^0$  может быть любым, поскольку оно в итоге исключается из уравнения энергии. При наличии химических превращений (и, соответственно, при наличии нескольких компонент в G-фазе), эти значения для каждой компоненты также могут быть выбраны произвольными. Однако для получения наиболее *простого* вида уравнения энергии необходимо использовать *согласованные* отсчеты энтальпий компонентов. В этом случае  $H^0$  – это теплота образования компонента при  $(T_0, p_0)$ .

В пакете для организации отсчета температур фаз  $(T_g, T_s)$  при описании свойств в виде V-переменных вводится переменная  $T_0$  и предполагается, что

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

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

Аналогично вводится отсчет давления  $p_0$ :

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

Для «замыкания» свойств в уравнении энергии в коде необходимо задать три переменные:

$c_{pg}$  - теплоемкость (V-переменная),

$h_{g0}$  - тепловая составляющая энтальпии (V-переменная),


$H^0$  - отсчет энтальпии (постоянная, по умолчанию равная нулю).

Заметим, что отсчет энтальпии  $H^0$  используется только в многокомпонентной G-фазе при моделировании источников, связанных с химическими реакциями (см. следующий параграф) и при моделировании однокомпонентной G-фазы при задании источников и граничных условий с коэффициентами SR\_FIXSOURCE и BC\_FIXFLUX.

Если теплоемкость  $c_{pg}$  задана постоянной, то энтальпия автоматически рассчитывается в коде по соотношению

$$h_g = c_{pg} T_g \quad (4.17)$$

Переменную  $h_g$  в этом случае можно не задавать (код ее не использует).

 **Замечание.** Численный алгоритм решения уравнения (4.14) кода Anes позволяет использовать произвольные функциональные зависимости для  $H_g(T_g, p)$ . Подробно этот алгоритм описан в [1].

 **Описание в проекте прикладной задачи :**

При использовании БД свойства материалов G-фазы задаются автоматически. Для задания свойств самостоятельно используются операторы секции [Properties]. При моделировании однокомпонентной G-фазы для задания энтальпии, теплоемкости и отсчета энтальпии используются операторы

Prop("<Имя материала>.H") = V-переменная

Prop("<Имя материала>.CP") = V-переменная

HOG = <значение>

Отсчеты температуры и давления задаются операторами

P0 = <значение>

T0 = <значение>

В уравнении энергии для задания коэффициента теплопроводности можно использовать два подхода. По умолчанию задается коэффициент теплопроводности ( $\lambda_g$ ) и для ее определения необходимо добавить в секцию [Properties] оператор

Prop("<Имя материала>.Cond") = V-переменная

Если в секции [PHI Variables] указать оператор

TypeDiff("TG") = ND\_PRAN

то в коде решателя для расчета коэффициента теплопроводности будет использоваться число Прандтля  $Pr$ , которое необходимо задать оператором описания свойства

Prop("<Имя материала>.Pran") = V-переменная

### 4.3.2 Описание источников пользователя

Для задания источниковых членов в уравнении энергии используется модификация общего подхода, описанного выше в пункте (3.4.3). Эта модификация потребовалась потому, что в конвективной части источника (3.11) необходимо использовать энтальпию потока:

$$S_{T_g} = \max(M_p, 0) \cdot H_0 - \max(-M_p, 0) \cdot H_g + C_T (V_T - T_g) + \max(M_p, 0) \cdot \Delta H_0 \quad (4.18)$$

Здесь

$C_T, V_T$  - коэффициент и значение источника,

$H_0$  - полная энтальпия G-фазы при параметрах «вдува» ( $T_g = V_T, p = V_p$ ),

$H_g$  - полная энтальпия G-фазы в данной точке РО,

$\Delta H_0$  - третий параметр источника для уравнения энергии.

Заметим, что  $H_0$  (полная энтальпия потока в точке вдува) рассчитывается автоматически Решателем при задании источника по соотношению (4.16):


$$H_0 = h_g(V_T, V_p) + H^0$$

где  $V_p$  - давление в точке «вдува».

По умолчанию значение третьего параметра  $\Delta H_0 = 0$ . Это значение используется в том случае, когда при «вдуве» происходит химическое превращение фазы с тепловым эффектом реакции  $\Delta H_0$ . В этом случае энтальпию, «вдуваемая» в расчетную область можно представить в виде

$$H_{\text{вдува}} = H_0 + \Delta H_0 = h_g(V_T, V_p) + H^0 + \Delta H_0 = h_g(V_T, V_p) + \{H^0 + \Delta H_0\}$$

где  $h_g$  - тепловая составляющая энтальпии среды при параметрах «вдува»,  $H^0$  - отсчет энтальпии среды,  $\Delta H_0$  – разность энтальпии «до вдува» и энтальпией среды при параметрах «вдува». В текущей версии этот параметр может быть только постоянной величиной.

 **Замечание.** Особое внимание следует уделить частному случаю задания источника с  $C_T = \text{SR\_FIXSOURCE}$  (и граничных условий с  $C_{b,T} = \text{BC\_FIXFLUX}$ ). Как уже отмечалось в (3.4.3) в этом случае значение источника  $V_T$  – это *полный* источник  $S_{T_g}$ . При наличии массового источника, следует помнить, что абсолютное значение источника для  $T_g$  должно быть согласовано с формулами расчета энтальпии  $h_g$  и ее отсчетом  $H^0$ .

 **Описание в проекте прикладной задачи :**

Коэффициент и значение источника описывается стандартным способом в секции [Sources]. Для описания третьего параметра  $\Delta H_0$  используется оператор DH ("*Имя патча* .TG") = значение

### 4.3.3 Работа сил давления и вязкая диссипация

Работа сил давления рассчитывается по соотношению

$$S_{T_g}^{(1)} = \varphi \left( \frac{\partial(p_0 + P_f)}{\partial \tau} + \mathbf{U}_g \nabla P_f \right), \quad (4.19)$$

а вязкая диссипация вычисляется по формуле

$$S_{T_g}^{(1)} = \varphi \mu_{\text{eff}} G, \quad (4.20)$$

где  $G$  – диссипативная функция. Для декартовой системы

$$G = 2 \left[ \left( \frac{\partial U_{gx}}{\partial x} \right)^2 + \left( \frac{\partial U_{gy}}{\partial y} \right)^2 + \left( \frac{\partial U_{gz}}{\partial z} \right)^2 \right] + \left( \frac{\partial U_{gx}}{\partial y} + \frac{\partial U_{gy}}{\partial x} \right)^2 + \left( \frac{\partial U_{gx}}{\partial z} + \frac{\partial U_{gz}}{\partial x} \right)^2 + \left( \frac{\partial U_{gz}}{\partial y} + \frac{\partial U_{gy}}{\partial z} \right)^2 \quad (4.21)$$

для цилиндрической системы ( $x = r, y = \theta, z = z$ )

$$G = 2 \left[ \left( \frac{\partial U_{gx}}{\partial x} \right)^2 + \left( \frac{1}{x} \frac{\partial U_{gy}}{\partial y} + \frac{U_{gx}}{x} \right)^2 + \left( \frac{\partial U_{gz}}{\partial z} \right)^2 \right] + \left( \frac{1}{x} \frac{\partial U_{gx}}{\partial y} + x \frac{\partial}{\partial x} \left( \frac{U_{gy}}{x} \right) \right)^2 + \left( \frac{\partial U_{gx}}{\partial z} + \frac{\partial U_{gz}}{\partial x} \right)^2 + \left( \frac{1}{x} \frac{\partial U_{gz}}{\partial y} + \frac{\partial U_{gy}}{\partial z} \right)^2 \quad (4.22)$$

**ⓘ Описание в проекте прикладной задачи :**

По умолчанию расчет этих внутренних источниковых членов отключен. Для их активации необходимо в секции [Internal Source] добавить операторы:

IsSrcTG\_PF = .True.  
IsSrcTG\_NU = .True.

### 4.3.4 Уравнение энергии, записанное через энтальпию

При решении задач с сильно переменными свойствами (типичный пример - среды СКД) удобнее в качестве  $\Phi$ -переменной использовать энтальпию  $H_g$ . В этом случае уравнение (4.14) можно записать в виде:

$$\frac{\partial (\varphi \rho_g H_g)}{\partial \tau} + \text{div} \left( \varphi \rho_g \mathbf{U}_g H_g - \varphi \frac{\lambda_{\text{eff}}}{c_{pg}} \nabla H_g \right) = S_{H_g}^{(1)} + \varphi S_{H_g} \quad (4.22a)$$

В текущей версии ANES этот подход имеет ряд ограничений:

- 1) в расчетной области должен быть только один Flow-материал,
- 2) зоны Struct должны отсутствовать (нет  $\Phi$ -переменной TS),
- 3) теплоемкость  $c_{pg}$  должна быть переменной и являться функцией  $T_g$ ,
- 4) нельзя использовать концентрации,
- 5) нельзя моделировать задачи о стабилизированном течении.

В этом режиме температура  $T_g$  является термодинамическим свойством  $T_g = f(p, H_g)$ .

Для описания молекулярного коэффициента переноса используется свойства с именем «Cond» или «Pran», которые задают значения  $\lambda_g$  (а не  $\lambda_g/c_{pg}$ !). Для задания *свойств* в подсистемах myTable и myFORM *нужно* использовать температуру (имя «TG»). Вместо свойства материала с именем «H» (расчет энтальпии по температуре) необходимо определить свойство с именем «T», аргументом которого в myFORM и myTABLE является энтальпия «HG».

Для задания других параметров можно использовать как переменную «HG», так и «TG». При работе с виртуальными функциями для доступа к энтальпии и температуре нужно использовать массивы (для определенности - неструктурная сетка):

usF8(idCV, P\_HG) - для энтальпии,  
usTTG(idCV) - для температуры.

Еще раз отметим, что в секции [Properties] нужно обязательно задать два дополнительных свойства:

Prop("<Имя материала>.T") = V-переменная, зависимость  $T = f(HG, \dots)$   
Prop("<Имя материала>.CP") = V-переменная, зависимость  $C_p = f(TG, \dots)$

причем первый оператор описывает зависимость *температуры* от *энтальпии* (именно это свойство используется в коде для расчета температуры!).

Внутренние источники, описанные в 4.3.3, доступны в этой модели. Источники пользователя, записываются аналогично (4.18):

$$S_{T_g} = \max(M_p, 0) \cdot H_0 - \max(-M_p, 0) \cdot H_g + C_T (V_T - H_g)$$

Аналогично описываются и граничные условия (п. 4.3.7). Следует отметить, что эта модель имеет смысл применять только для LES/DNS моделирования задач с сильно переменной теплоемкостью. Для остальных задач модель с  $\Phi$ -переменной «TG» является эффективной.

#### 4.3.5 Межфазный теплообмен в Porous зонах

Третья составляющая внутреннего источника описывает межфазный теплообмен между G и S фазами в Porous-зонах. В коде эти источники реализованы в виде, аналогичном (4.18):

$$S_{T_g}^{(gs)} = \max(M_p^{(gs)}, 0) \cdot H_g(T_s) - \max(-M_p^{(gs)}, 0) \cdot H_g(T_g) + \alpha_{gs} \tilde{A}_{gs} (T_s - T_g) + \max(M_p^{(gs)}, 0) \cdot \Delta H_{gs}, \quad (4.23)$$

где

$\alpha_{gs}$  - коэффициент межфазной теплоотдачи,

$\tilde{A}_{gs}$  - удельная межфазная поверхность - поверхность раздела G- и S- фаз, отнесенная к единице объема пространства,

$M_p^{(gs)}$  - объемный источник массы, связанный с массообменом между фазами; знак источника положителен, если он направлен в G-фазу,

$\Delta H_{gs}$  - разность энтальпий компоненты G-фазы при переходе из S в G фазу, определяемая при температуре S-фазы

$$\Delta H_{gs}(T) = H_g^{(s)}(T_s) - H_g(T_s)$$

где верхний индекс (s) означает энтальпию компонента G-фазы в S-фазе (в твердом состоянии).

#### 4.3.6 Межфазный теплообмен на границах раздела G- и S-фаз

Четвертая составляющая внутреннего источника описывает межфазный теплообмен между G и S фазами на явно выделенных границах раздела G-и S-фаз. Эти границы в коде описываются патчами с именами "FS\_...". Соответственно с этими границами связывается два *поверхностных источника* со стороны G- и S- фаз. Эти источники рассчитываются из условия сопряжения плотностей тепловых потоков

$$(\lambda_{eff} \nabla T_g) \cdot \mathbf{n}_{sg} = (\lambda_s \nabla T_s) \cdot \mathbf{n}_{sg} \quad (4.24)$$

Здесь  $\mathbf{n}_{sg}$  - единичная нормаль к межфазной границе, направленная в сторону G-фазы,  $\lambda_s$  и  $T_s$  - коэффициент теплопроводности и температура S-фазы.

#### 4.3.7 Граничные условия

Граничные условия (3.15) для уравнения энергии модифицируются аналогично источникам (4.18):

$$J_{T_{g,b}} = \max(m_b, 0) \cdot H_0 - \max(-m_b) \cdot H_p + C_{b,T} (V_T - T_w) + \max(m_b, 0) \cdot \Delta H_0, \quad (4.25)$$

$$J_{T_{g,b}} = \max(m_b, 0) \cdot H_0 - \max(-m_b) \cdot H_p + |C_{b,T}| (V_T - T_p) + \max(-m_b, 0) \cdot \Delta H_0.$$

Здесь  $C_{b,T}$ ,  $V_T$  - коэффициент и значение граничного условия,  $\Delta H_0$  - третий параметр ГУ для уравнения энергии. Использование этого параметра аналогично его использованию в источниковом члене (4.18). В текущей версии этот параметр может быть только постоянной величиной.

При использовании  $\Phi$ -переменной NG используются аналогичные соотношения:

$$J_{T_{g,b}} = \max(m_b, 0) \cdot H_0 - \max(-m_b) \cdot H_p + C_{b,T} (V_T - H_w),$$

$$J_{T_{g,b}} = \max(m_b, 0) \cdot H_0 - \max(-m_b) \cdot H_p + |C_{b,T}| (V_T - H_p)$$

### 4.3.8 Расчет потоков

В коде Anes предусмотрен набор подпрограмм [2] пользователя для расчета компонент полного потока любой  $\Phi$ -переменной. Отметим, что при расчете потоков эти подпрограммы используют в конвективной составляющей плотности потока *абсолютные значения* энтальпий

$$J_{T_{g,k}} = \left( \varphi \rho_g U_{g,k} H_g - \varphi \lambda_{\text{eff}} \frac{\partial T_g}{\partial x_k} \right),$$

$$H_g = h_g + H^0$$

### 4.3.9 Уравнение энергии для стабилизированного течения

При моделировании стабилизированного течения (см. пункт 4.2.6) наряду с моделированием гидродинамики можно моделировать и теплообмен для двух режимов: с постоянной температурой стенки канала или при постоянной плотности теплового потока на стенке канала.

Если на стенке задана постоянная температура  $T_{\text{wall}}$ , то при выходе на стабилизированный режим течения отношение теплового потока на стенке к разнице температуры стенки и среднemasсовой температуры потока (коэффициент теплоотдачи) перестает зависеть от координаты  $z$ . Если на стенке задана постоянная плотность теплового потока  $q_{\text{wall}}$ , то перестает зависеть от координаты  $z$  разница температуры стенки и среднemasсовой температуры потока. Еще раз напомним, что все это справедливо только для постоянных свойств G-фаз. В коде Anes для структурных сеток реализованы оба варианта.

Уравнение энергии для стабилизированного режима можно записать в следующем общем виде:

$$\frac{\partial}{\partial x} \left( \varphi \rho_g c_{pg} U_{gx} T_g - \varphi \lambda_{\text{eff}} \frac{\partial T_g}{\partial x} \right) + \frac{\partial}{\partial y} \left( \varphi \rho_g c_{pg} U_{gy} T_g - \varphi \lambda_{\text{eff}} \frac{\partial T_g}{\partial y} \right) = S_z,$$

$$S_z = - \frac{\partial}{\partial z} (\rho U_{gz} c_{pg} T_g) = - \rho U_{gz} c_{pg} \frac{\partial T_g}{\partial z}$$
(4.26)

#### Граничное условие $T_{\text{wall}} = \text{const}$

Если на границах расчетной области для  $T_g$  заданы либо нулевые потоки, либо граничное условие первого рода с постоянной температурой  $T_{\text{wall}}$ , то можно предположить, что независимой от  $z$  величиной является безразмерная величина [5]

$$\Theta = \frac{T_g - T_{\text{wall}}}{T_b - T_{\text{wall}}}$$
(4.27)

Здесь  $T_b$  – среднemasсовая температура, определяемая соотношением

$$T_b = \frac{1}{G_0} \iint_{xy} \rho_g \varphi U_{gz} T_g dx dy, \quad G_0 = \iint_{xy} \rho_g \varphi U_{gz} dx dy$$
(4.28)

Если записать это условие в виде

$$\ln(\Theta) = \ln(T_{\text{wall}} - T_g) - \ln(T_{\text{wall}} - T_b)$$

и продифференцировать его по  $z$  (напомним, что  $\Theta$  не зависит от  $z$ ), то можно записать

$$\frac{\partial T_g}{\partial z} = \frac{T_{\text{wall}} - T_g}{T_{\text{wall}} - T_b} \left( \frac{dT_b}{dz} \right)$$



Используя это соотношение, источник  $S_z$  в (4.26) можно представить в виде

$$S_z = -\rho_g c_{pg} U_{gz} \frac{T_{wall} - T_g}{T_{wall} - T_b} \left( \frac{dT_b}{dz} \right) \quad (4.29)$$

Градиент среднемассовой температуры можно определить из интегрального баланса тепла, которое можно получить путем интегрирования уравнения (4.26) по ху-сечению

$$\left( \frac{dT_b}{dz} \right) = \frac{Q_{wall}}{G_0 c_{pg}}, \quad Q_{wall} = - \int_{\Pi} \lambda_g \frac{\partial T_g}{\partial n} dS \quad (4.30)$$

где  $Q_{wall}$  – полный поток на единицу длины канала,  $\Pi$  - периметр канала .

Система (4.26), (4.28), (4.29) и (4.30) является замкнутой и позволяет рассчитать поле температур при заданном значении  $T_{wall}$ . Заметим, что полученное поле температур не соответствует ни какому реальному сечению канала по  $z$  на участке стабилизации. Физический смысл имеет только безразмерное поле  $\Theta$ . Рассчитанное поле  $T_g$ , среднемассовая температура  $T_b$  и средняя плотность теплового потока

$$q_{wall} = \frac{Q_{wall}}{\Pi}$$

будут зависеть от граничного значения  $T_{wall}$ . Однако средний коэффициент теплоотдачи

$$\alpha = \frac{q_{wall}}{T_{wall} - T_b} \quad (4.31)$$

как и  $\Theta$  не будут зависеть от граничных условий.

Для удобства обработки результатов в Решателе определены пять переменных, которые можно использовать также как и при работе с любыми V-переменными:

DEV_Twall	- это $T_{wall}$ ,
DEV_Tbulk	- это $T_b$ ,
DEV_Qwall	- это $q_{wall}$ ,
DEV_DTDZ	- это $(dT_b/dz)$ ,
DEV_Tvol	- это средняя температура в РО $T_v$ :

$$T_v = \frac{1}{V_{PO}} \iiint_{xyz} \varphi T_g dx dy dz$$

Значения этих переменных выводятся в листинг расчета в виде:

```

Стабилизированное течение: dT(BULK)/dZ= 2.3529E-02 C/m
Стабилизированное течение: T(Bulk)= -1.203 C
Стабилизированное течение: T(Volume)= 0.0000 C
Стабилизированное течение: T(Wall)= 0.1000 C
Стабилизированное течение: q(Wall)= 4.3411E-02 Wt/m^2

```

#### **Граничное условие $q_{wall} = \text{const}$**

Если на стенке канала задана постоянная плотность теплового потока, то при стабилизированном течении независимой от  $z$  будет величина

$$T_g - T_b$$

Продифференцировав это выражение, источник  $S_z$  в (4.26) можно записать в виде:

$$S_z = -\rho_g U_{gz} c_p \frac{Q_{wall}}{c_p G_0} = -\rho_g U_{gz} \frac{Q_{wall}}{G_0}$$

Поскольку при задании плотности теплового потока на всех границах температура  $T_g$  определена с точностью до постоянной, то Решатель *масштабирует поле температуры*  $T_g$ . Способ масштабирования зависит от значения оператора секции [Special Data]:

$$C(\text{"WhatTZERO"}) = \text{"Tbulk"} / \text{"Tvol"}$$

В первом случае (он включен по умолчанию) поле корректируется так, чтобы средне-массовая температура  $T_b = 0$ , во втором случае поле корректируется так, чтобы средняя по РО температура  $T_v = 0$ .

Как и в случае  $T_{wall} = \text{const}$ , для заданного потока на стенке определены четыре переменные

DEV_Twall	- $T_{wall}$ ,
DEV_Tbulk	- $T_b$ ,
DEV_Qwall	- $q_{wall}$ ,
DEV_DTDZ	- $(dT_b/dz)$ .
DEV_Tvol	- $T_v$ .

**① Описание в проекте прикладной задачи :**

Для выбора режима теплообмена необходимо задать оператор секции [Internal Source]:

DEV\_TG\_MODE = devTG\_CONST / devTG\_FLUX

и задать нужное граничное условие (температуру стенки или плотность теплового потока на стенке).

### 4.3.10 Уравнение энергии для развитого периодического течения

При моделировании развитых периодических течений также можно моделировать теплообмен, если на границах расчетной области (за исключением границ с периодическими условиями) задан *постоянный тепловой поток*  $q_{wall}$  (или на части границ задан нулевой тепловой поток).

Условие периодичности для температуры на длине  $L_z$  можно записать в виде

$$\frac{T_g(x, y, 0) - T_g(x, y, L_z)}{L_z} = \frac{T_g(x, y, L_z) - T_g(x, y, 2L_z)}{L_z}$$

Этому условию удовлетворяет приведенная температура  $T_p$ :

$$T_p = T_g - \left[ T_{b0} + \left( \frac{dT_b}{dz} \right) z \right], \quad \left( \frac{dT_b}{dz} \right) = \frac{Q_{wall}}{G_0 c_{pg} L_z} = \text{const}, \quad (4.32)$$

$$T_p(x, y, 0) = T_p(x, y, L_z)$$

Здесь  $Q_{wall}$  – интеграл от плотности теплового потока по всей поверхности расчетной области,  $T_{b0}$  – произвольная постоянная («формально» это средне-массовая температура при  $z = 0$ ).

При использовании этого режима Решатель в Ф-переменной  $T_g$  сохраняет значение приведенной температуры  $T_p$  (именно это поле сохраняется в файле результатов). Поскольку в этом режиме температура  $T_p$  определена с точностью до постоянной, то Решатель *масштабирует поле так, чтобы либо средне-массовая температура  $T_{b0}$ , либо средняя по объему температура  $T_v$  была равна нулю!* Для выбора способа масштабирования используется оператор секции [Special Data]:

C("WhatTZERO") = "Tbulk" / "Tvol"

Если оператор не указан, то используется модель  $T_{b0} = 0$ .

Характеристики теплообмена сохраняются в тех же переменных Решателя, что и для стабилизированного режима, но смысл их несколько иной:

DEV_Twall	- приведенная температура стенки $\bar{T}_{p,wall}$ , осредненная по поверхности с заданной плотностью теплового потока
DEV_Tbulk	- средне-массовая температура потока при $z = 0$ - $T_{b0}$ ,
DEV_Tvol	- средняя температура по объему РО - $T_v$ ,

DEV\_Qwall - плотность теплового потока на стенке  $q_{wall}$ ,  
 DEV\_DTDZ - постоянный градиент среднemasсовой температуры ( $dT_b/dz$ ).

Для расчета среднего коэффициента теплоотдачи можно воспользоваться соотношением, аналогичным (3.31)

$$\alpha_m = \frac{q_{wall}}{\bar{T}_{p,wall} - T_{b,0}} \quad (4.33)$$

Для расчета локального коэффициента теплоотдачи в сечении  $z$  нужно заменить в (4.33) среднюю температуру стенки на локальное значение температуры стенки. Если необходимы «истинные» распределения температур, то их можно рассчитать из соотношений

$$T_g = T_p + \left( \frac{dT_b}{dz} \right) z, \quad T_b = T_{b,0} + \left( \frac{dT_b}{dz} \right) z$$

**ⓘ Описание в проекте прикладной задачи :**

Для задания режима теплообмена необходимо указать следующий оператор секции [Internal Source]:

DEV\_TG\_MODE = devTG\_FLUX

и задать на поверхности канала постоянную плотность теплового потока.

#### 4.3.11 Уравнение энергии для сложного периодического течения

Если модель DEV\_TG\_MODE = devTG\_FLUX не устраивает пользователя (например, при использовании сложных источников членов), то можно самостоятельно реализовать постановку с ПГУ. При решении *стационарных* задач в случае задания тепловых потоков на стенках канала необходимо «отмасштабировать» температуру, которая определена с точностью до постоянной. При использовании встроенных моделей ANES, описанных выше, это масштабирование проводится автоматически так, чтобы среднemasсовая или среднemasобъемная температура равнялась нулю.

Этот алгоритм можно активировать и для независимой постановки. Для этого используется тот же оператор секции [Special Data]:

C("WhatTZERO") = "TNone" / "Tbulk" / "Tvol"

При использовании второго режима:

- 1) при работе со *структурными* сетками поле  $T_g$  будет отмасштабировано на среднemasсовую температуру  $T_b$ , рассчитанную в сечении  $IZ = 1$ ;
- 2) при работе с *неструктурными* сетками необходимо указать поверхностный патч, на котором будет рассчитываться среднemasсовая температура. Имя этого патча задается оператором:

C("LES.UMPATCH") = <имя поверхностного патча>

если этот оператор не указан, то используется патч с именем 'CYCLE\_BS\_Z'.

Замечание. В этом режиме при задании на стенке плотности теплового потока (и самостоятельного включения источника, связанного с  $dT_b/dz$ ) может наблюдаться плохая сходимость по температуре. В этом случае нужно зафиксировать значение температуры в одной точке, если не используется масштабирование ("TNone"):

COEF("TZERO.TG") = SR\_FIXVAL

VAL("TZERO.TG") = 0

где TZERO - патч типа Point в любой точке расчетной области.

#### 4.4 Уравнения для концентраций

Для моделирования *химических процессов* в G-фазе используются  $\Phi$ -переменные, представляющие собой массовые доли компонентов (концентрации). В этом случае G-фаза рассматривается как гомогенная смесь  $N_C \leq 8$  компонентов ( $\Phi$ -переменные  $C_1 - C_8$ ):

$$C_k = \frac{m_k}{\sum_1^{N_C} m_k}, k=1..N_C,$$

где  $m_k$  - масса компонента в объеме.

Поля концентраций  $C_k$  рассчитываются из уравнений баланса массы компонентов, которые являются стандартными уравнениями переноса (3.3):

$$\frac{\partial(\phi \rho_g C_k)}{\partial \tau} + \text{div}(\phi \rho_g \mathbf{U}_g C_k - \phi D_{\text{eff},k} \nabla C_k) = \phi M_{Ck}, \quad (4.34)$$

где

$$D_{\text{eff},k} = \rho_g \left( \frac{v_g}{Sc_k} + \frac{v_t}{Sc_{k,t}} \right) \text{ или } D_{\text{eff},k} = \rho_g \left( D_k + \frac{v_t}{Sc_{k,t}} \right),$$

$$\sum_k C_k = 1,$$

$Sc_k$ ,  $Sc_{k,t}$  - молекулярное и турбулентное числа Шмидта для k-того компонента,  
 $D_k$  - молекулярный коэффициент диффузии k-того компонента.

Суммирование уравнений (4.34) по всем компонентам приводит к уравнению неразрывности (4.1), поэтому члены в уравнениях (4.34) должны удовлетворять дополнительным условиям

$$M_p = \sum_k M_{Ck}, \quad \sum_k (D_{\text{eff},k} \nabla C_k) = 0, \quad (4.35)$$

выполнение которых должны контролироваться пользователем:

В Решателе Anes используется специальный алгоритм численного решения системы (4.34), который гарантирует, что сумма концентраций всегда равна единице.

Уравнение (4.34) можно записать в виде, который пригодится при описании уравнения энергии для многокомпонентной смеси:

$$\frac{\partial(\phi \rho_g C_k)}{\partial \tau} + \text{div}(\phi \mathbf{m}_{Ck}) = \phi M_{Ck}, \quad (4.36)$$

$$\mathbf{m}_{Ck} = \rho_g \mathbf{U}_g C_k + \mathbf{J}_{Ck}, \quad \mathbf{J}_{Ck} = -D_{\text{eff},k} \nabla C_k$$

где

$\mathbf{m}_{Ck}$  - вектор плотности полного потока массы k-го компонента,

$\mathbf{J}_{Ck}$  - диффузионная составляющая этого вектора.

##### 4.4.1 Источники и граничные условия

Граничные условия и источниковые члены описываются стандартными способами, рассмотренными в разделах 3.4.5 и 3.5.2.

В *зонах Pogois* предусмотрено несколько моделей массообмена, подробнее эти модели описаны в главе 8. Границы *раздела G-S фаз* (патчи типа "FS\_...") считаются непроницаемыми ( $M_p = 0$ ) и химически пассивными ( $M_{Ck} = 0$ ).

Если на границе раздела G-S фаз имеет место химическая реакция, то для ее описания необходимо самостоятельно задавать поверхностные источниковые члены, связанные с патчами "FS\_...".

#### 4.4.2 Баро и термодиффузия

В коде Anes реализована модель, позволяющая учесть процессы термодиффузии и бародиффузии при моделировании *бинарной смеси* (смеси двух компонентов). В этом случае в диффузионный поток компонента добавляются два члена [6,7] (ниже приведено соотношение для первого компонента):

$$\mathbf{J}_{C_1} = -\phi \Gamma_{C_1} \left\{ \nabla C_k - \frac{M_1 - M_2}{M} C_1 C_2 \frac{\nabla p_f}{(p_f + p_0)} + k_T \frac{M_1 - M_2}{M} C_1 C_2 \frac{\nabla T_g}{(T_g + T_0)} \right\}, \quad (4.37)$$

$$M = \frac{1}{\frac{C_1}{M_1} + \frac{C_2}{M_2}}$$

Здесь  $M_i$  – молекулярный вес компонента,  $k_T$  – постоянная термодиффузии. По умолчанию для этой величины используется значение

$$k_T = 0.25$$

#### ❗ Описание в проекте прикладной задачи :

Для задания коэффициента диффузии для каждого компонента используется тот же подход, что и для уравнения энергии. По умолчанию с помощью операторов секции [Properties] необходимо задать коэффициент диффузии  $D_X$  (ниже  $X = 1 \dots 8$ ):

Prop("<Имя материала>.DDif.CX") = V-переменная

Если в секции [PHI Variables] указать операторы  
TypeDiff("CX") = ND\_PRAN

то в коде решателя для расчета коэффициента диффузии будет использоваться число Шмидта ( $Sc_x$ ), которое необходимо задать оператором описания свойства

Prop("<Имя материала>.SC.CX") = V-переменная

Для активации расчета термодиффузии необходимо в секцию [Special Data] файла проекта включить оператор

L("IsSrcConc\_TERMO") = .TRUE.

Аналогичный оператор используется для активации бародиффузии

L("IsSrcConc\_BARO") = .TRUE.

Значение постоянной термодиффузии  $k_T = 0.25$  можно изменить оператором

R("ConstTD") = 0.25

#### 4.5 Уравнение энергии для многокомпонентной G-фазы

В случае использования многокомпонентной G-фазы уравнение энергии записывается для энтальпии смеси компонентов

$$\frac{\partial(\phi \rho_g H_g)}{\partial \tau} + \text{div} \left( \phi \rho_g \mathbf{U}_g H_g + \phi \sum_k \mathbf{J}_{C_k} H_{gk} - \phi \Gamma_{T_g} \nabla T_g \right) = S_{T_g}^{(1)} + \phi S_{T_g}, \quad (4.38)$$

$$H_g = \sum_k C_k H_{gk}$$

или

$$\frac{\partial(\phi \rho_g H_g)}{\partial \tau} + \text{div} \left( \phi \sum_k \mathbf{m}_{C_k} H_{gk} - \phi \lambda_{\text{eff}} \nabla T_g \right) = S_{T_g}^{(1)} + \phi S_{T_g}$$

где

$H_{gk}$  – энтальпия k-ого компонента, представляемая в пакете в виде

$$H_{gk} = h_{gk}(T_g, p) + H_k^0,$$

$c_{pg}$  – теплоемкость смеси

$$c_{pg} = \sum_k C_k c_{pg,k}, \quad c_{pg,k} = \left( \frac{\partial h_{g,k}}{\partial T_g} \right)_p$$

#### 4.5.1 Описание свойств многокомпонентной G-фазы

Термодинамические переменные (энтальпии и теплоемкости) для компонентов в пакете, описываются так же, как и для однокомпонентной среды, только в имя свойства добавляется имя компонента с точкой в начале имени компонента. Например:

```
Prop("<Имя материала>.HG.C1") = V-переменная
Prop("<Имя материала>.CPG.C1") = V-переменная
HOG.C1 = <значение>
```

Алгоритм расчета плотности, вязкости и теплопроводности смеси G-фазы можно задать либо самостоятельно (в этом случае нужно использовать тот же подход, что и для однокомпонентной среды), либо воспользоваться встроенными алгоритмами.

Активация встроенного алгоритма осуществляется символьным оператором секции [Special Data] файла проекта

```
C("PropConcModel") = byCONC/ byWILKE
```

##### **Модель byCONC**

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


$$\lambda_g = \sum_k r_k \lambda_k, \quad \mu_g = \sum_k r_k \mu_k, \quad \rho_g = \frac{1}{\sum_k \frac{c_k}{\rho_k}} = \sum_k r_k \rho_k, \quad \beta_T = \sum_k r_k \beta_{T,k}, \quad \beta_P = 0 \quad (4.39)$$

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

Здесь  $M_i$  – молекулярный вес компонента,  $r_k$  – объемная (мольная) доля компонента.

Для расчета необходимо в секции свойств (или в БД) описать свойства отдельных компонент (здесь  $X = 1 \dots 8$ ) с именами:

```
Cond.CX, DinVisc.CX, Dens.CX, BetaT.CX, MolW.CX
```

 **Замечание.** При использовании этого алгоритма в свойствах DinVisc.CX необходимо задать *динамическую* вязкость компонента, а не кинематическую, как принято в коде Anes.

##### **Модель byWILKE**

В этой модели свойства рассчитываются для идеальной газовой смеси по соотношениям Уилки:

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

$$\lambda_g = \sum_i \lambda_i r_i \frac{1}{\sum_k G_{ik}^* r_k}, \quad G_{ik}^* = \begin{cases} 1.0 & , i = k \\ 1.065 \cdot G_{ik} & , i \neq k \end{cases} \quad (4.40)$$

$$\rho_g = \frac{P_0}{R_0 (T_g + T_0) \sum_k \frac{c_k}{M_k}} \quad (a), \quad \rho_g = \frac{P_0 + Pf}{R_0 (T_g + T_0) \sum_k \frac{c_k}{M_k}} \quad (b)$$

Для расчета необходимо в секции свойств (или в БД) описать свойства отдельных компонент с именами (здесь  $X = 1 \dots 8$ ):

Cond.CX, DinVisc.CX, MolW.CX

Коэффициент термического расширения  $\beta_T$  в этой модели не используется. Если необходимо учесть эффекты плавучести, то необходимо использовать общую модель, основанную на переменной плотности.

Для расчета плотности по умолчанию используется отсчет давления  $P_0$  (соотношение (4.40a)). Если необходимо учитывать полное давление (4.40b), то нужно включить флаг секции [Special Data]:

L("IsWILKE\_PABS") = .True.

#### 4.5.2 Источниковые члены, задаваемые пользователем

При наличии смеси компонентов в соотношении (4.18) источник массы  $M_p$  заменяется на массовые источники  $S_{ck}$  отдельных компонент:

$$S_{Tg} = \sum \max(M_{ck}, 0) \cdot H_{k,0} - \sum \max(-M_{ck}, 0) \cdot H_{gk} + C_T (V_T - T_g) + \sum \max(M_{ck}, 0) \cdot \Delta H_{0k} \quad (4.41)$$

где

- $M_{ck}$  - источниковые члены в уравнениях для компонентов фазы, связанные с патчем источника,
- $C_T, T_0$  - коэффициент и значение источника,
- $H_{gk}, H_{k,0}$  - энтальпии компонент G-фазы при параметрах среды и параметрах вдува (с учетом отсчета энтальпий  $H_k^0$ ),
- $\Delta H_{0k}$  - третий параметр источника для уравнения энергии (в текущей версии этот параметр может быть только постоянной величиной).

По умолчанию массив значений третьего параметра  $\Delta H_{0k} = 0$ . Он используется в том случае, когда при «вдуве» происходит химическое превращение отдельной компоненты фазы. С учетом этого параметра «вдуваемую» энтальпию компонента можно представить в виде

$$H_{\text{вдува}} = H_{gk}(T_0) + \Delta H_{0k}$$

где  $\Delta H_{0k}$  – разность энтальпии компонента «до вдува» и энтальпией компонента среды при параметрах вдува.

#### 4.5.3 Описание химических реакций

При протекании в G-фазы гомогенных химических реакций возникают объемные массовые источники компонентов, связанные законом сохранения массы

$$\sum_k M_{ck} = 0.$$

При этом в уравнении энергии смеси, записанном в балансовом «дивергентном» виде (4.38), не возникает никакого теплового источника, если отсчеты энтальпий компонентов  $H_k^0$  согласованы (т.е. они представляют собой теплоты образования компонентов при  $T=T_0$ , что соответствует  $T_g = 0$ ).

В связи с этим для корректного описания тепловых эффектов химических реакций в коде Anes необходимо:

- 1) согласовать переменные  $H_k^0$ ,
- 2) для объемного патча с именем «\*» (т.е. для всей расчетной области) задать постоянный нулевой источник:  $C_{Tg} = SR\_FIXSOURCE, T_0 = 0$

Важно отметить, что нулевой источник обязательно должен быть описан. Это связано с численным алгоритмом кода: если для патча с массовыми источниками не задан явно источник для уравнения энергии, то Решателем будет сформирован источник по умолчанию

$$S_{Tg} = \sum_k S_{ck} H_{gk}$$

#### 4.5.4 Описание граничных условий

Граничные условия (3.25) для уравнения энергии модифицируются аналогично источникам (3.41):

$$J_{Tg,b} = \sum_k \max(J_{ck,b}, 0) \cdot H_{k,0} - \sum_k \max(-J_{ck,b}, 0) \cdot H_{k,P} + C_{b,T} (V_{b,T} - T_w) + \sum_k \max(J_{ck,b}, 0) \cdot \Delta H_{k0},$$

$$J_{Tg,b} = \sum_k \max(J_{ck,b}, 0) \cdot H_{k,0} - \sum_k \max(-J_{ck,b}, 0) \cdot H_{k,P} + |C_{b,T}| (V_{b,T} - T_p) + \sum_k \max(J_{ck,b}, 0) \cdot \Delta H_{k0},$$

(4.42)

где

$J_{ck}$  - плотность потока массы компонента на границе (ГУ для  $C_k$ ),

$C_{b,T}$  - коэффициент граничного условия,

$V_{b,T}$  - значение граничного условия ( $H_{k,0} = H_{gk}(V_{b,T})$ ),

$\Delta H_{k0}$  - третий массив параметров ГУ для уравнения энергии, его использование аналогично его использованию в источниковом члене.

#### 4.5.5 Межфазный теплообмен

Межфазный теплообмен между G и S фазами в Porous-зонах описывается объемными источниками, аналогичными (4.23)

$$S_{Tg}^{(gs)} = \sum_k \max(M_{ck}^{(gs)}, 0) \cdot H_{gk}(T_s) - \sum_k \max(-M_{ck}^{(gs)}, 0) \cdot H_{gk}(T_g) +$$

$$\alpha_{gs} \tilde{A}_{gs} (T_s - T_g) + \sum_k \max(M_{ck}^{(gs)}, 0) \cdot \Delta H_{gs,k},$$

(4.43)

где

$\alpha_{gs}$  - коэффициент межфазной теплоотдачи,

$\tilde{A}_{gs}$  - удельная межфазная поверхность - поверхность раздела G- и S- фаз, отнесенная к единице объема,

$\Delta H_{gs,k}$  - разность энтальпий компоненты G-фазы при переходе из S в G фазу, определяемая при температуре S-фазы

$$\Delta H_{gs,k}(T) = H_{gk}^{(s)}(T_s) - H_{gk}(T_s)$$

где верхний индекс (s) означает энтальпию компонента G-фазы в S-фазе (в твердом состоянии).



Межфазный теплообмен между G и S фазами на *явно выделенных границах* раздела G-и S-фаз (эти границы в пакете описываются патчами с именами "FS\_...") описывается аналогично случаю однокомпонентной среды.

#### 4.5.6 Расчет потоков

В пакете предусмотрен набор подпрограмм [2] пользователя для расчета компонент полного потока любой  $\Phi$ -переменной. При расчете потоков эти подпрограммы используют в конвективной составляющей значения энтальпий  $H_{g,k}$ :

$$J_{Tg,k} = \left( \varphi \sum_k m_{ck} H_{gk} - \varphi \lambda_{\text{eff}} \frac{\partial T_g}{\partial x_k} \right),$$

$$H_{gk} = h_{gk} + H_k^0$$

#### 4.6 Свойства G-фазы

Для полного определения прикладной задачи, необходимо задать свойства фазы. Эти величины были уже введены выше, в данном разделе они собраны в таблице 4.2. Как уже отмечалось, все свойства задаются операторами секции [Properties] файла проекта следующего вида:

Prop("<Имя материала>.<Имя свойства>") = V-переменная

Таблица 4.2 - Свойства G-фазы

Свойство	Имя в Prop	Назначение
$\rho_g$	Dens	плотность G-фазы;
$\beta_T = -\frac{1}{\rho_g} \left( \frac{\partial \rho_g}{\partial T} \right)_p$	BetaT	-изобарический коэффициент термического расширения; используется при расчете сил плавучести в приближении Буссинеска;
$\beta_p = \left( \frac{\partial \rho_g}{\partial p} \right)_T$	BetaP	-коэффициент сжимаемости; используется при моделировании «сжимаемых» течений;
$T_0$	-	-постоянный отсчет температуры; используется в формулах расчета плотности и энтальпии; сумма ( $T_0 + T_g$ ) или ( $T_0 + T_s$ ) должна соответствовать температуре в градусах Кельвина;
$p_0$	-	-постоянный отсчет давления; используется для расчета статического давления в формулах расчета свойств; сумма $p_0 + P_f$ соответствует локальному термодинамическому давлению в Па;
$\nu_g$	Visc	-кинематический коэффициент вязкости,
$h_g(k)$	H, H.CX	-энтальпия однокомпонентной среды или массив энтальпий компонент G-фазы ( $X = 1 \dots 8$ ); если теплоемкость постоянна, эта переменная <i>не используется</i> , энтальпии рассчитываются по соотношению $c_{pg}(k) T_g$ ;
$H^0(k)$	-	- массив теплот образования компонентов;
$c_{pg}(k) = \left( \frac{\partial h_g(k)}{\partial T_g} \right)_p$	CP, CP.CX	-теплоемкость компонента;

$\rho_{\text{арх}}, T_{\text{арх}}$	-	- постоянные отсчеты плотности и температуры, используемые для моделирования силы плавучести,
$g_{\text{cx}}, g_{\text{cy}}, g_{\text{cz}}$	-	- <i>декартовы</i> компоненты вектора силы тяжести; используются для расчета сил плавучести как в декартовой, так и в цилиндрической системах координат;
$Pr, Sc$	$Pr_{\text{ан}} / Sc$	- молекулярные числа Прандтля и Шмидта.

#### 4.6.1 Особенности задания свойств сжимаемой G-фазы

При решении задач «классической» газодинамики трансзвуковых и сверхзвуковых течений невязкого и нетеплопроводного газа можно поступить следующим образом:

- 1) для моделирования невязкого течения необходимо положить коэффициенты вязкости и теплопроводности, равными малым значениям. В качестве критерия малости можно использовать «характерные» значения для числа Рейнольдса, равными  $10^7 - 10^{10}$ .
- 2) в уравнении энергии необходимо активировать внутренний источник, связанный с работой сил давления.
- 3) при моделировании адиабатических течений можно исключить решение уравнения энергии и использовать уравнение состояния в виде

$$\rho_g = (p_0 + P_f)^{1/\gamma}, \quad \gamma = \frac{c_{pg}}{c_{pg} - R},$$

где  $R$  – газовая постоянная для используемого газа.

Примеры решения таких задач можно найти в секции «Сжимаемые течения» библиотеки примеров кода.

## 5. Математическая модель S-фазы

В текущей версии пакета с S-фазой связаны только две Ф-переменные. Первая предопределенная Ф-переменная - это температура S-фазы  $T_s$  (Ф-имя - "TS"). Вторая Ф-переменная - "Melt" - используется для моделирования плавления фазы.

### 5.1 Уравнение энергии S-фазы

Поскольку S-фаза неподвижна, то для ее описания используется уравнение теплопроводности

$$(1-\varphi)\rho_s c_{ps} \frac{\partial T_s}{\partial \tau} = \operatorname{div}((1-\varphi)\underline{\lambda}_s \otimes \nabla T_s) + S_{T_s}^{(gs)} + (1-\varphi)S_{T_s} + S_{T_s}^{(I)}, \quad (5.1)$$

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,

$S_{T_s}$  - объемный источник, задаваемый пользователем, отнесенный к единице объема S-фазы,

$S_{T_s}^{(gs)}$  - внутренние источники, связанные с межфазным G-S взаимодействием.

$S_{T_s}^{(I)}$  - внутренний источник, связанный с моделями плавления и излучения.

При описании распределения S-фазы с каждым материалом связывается свой набор V-переменных - свойств:

$\lambda_{sx}, \lambda_{sy}, \lambda_{sz}, \rho_s, c_{ps}$

Внутренние источники  $S_{T_s}^{(gs)}$  рассчитываются автоматически в зонах Porous и на границах раздела Struct-Flow.

#### **ⓘ** Описание в проекте прикладной задачи :

Если тензор теплопроводности изотропен, то для его задания используется оператор секции [Properties]:  
Prop("<Имя материала>.Cond") = V-переменная

Если используются ортотропный коэффициент теплопроводности, то необходимо в секции [PHI Variables] задать свои собственные произвольные имена для диагональных компонент тензора теплопроводности, например:

```
NamDiffX("TS") = "CondX"
NamDiffY("TS") = "CondY"
NamDiffZ("TS") = "CondZ"
```

и в секции [Properties] задать свойства для этих имен:

```
Prop("<Имя материала>.CondX") = V-переменная
Prop("<Имя материала>.CondY") = V-переменная
Prop("<Имя материала>.CondZ") = V-переменная
```

Для задания плотности фазы и ее теплоемкости используются операторы

```
Prop("<Имя материала>.Dens") = V-переменная
Prop("<Имя материала>.CP") = V-переменная
```

#### 5.1.1 Межфазный теплообмен в зонах Porous

Уравнение энергии S-фазы (5.1) записано не в «стандартном» виде уравнения баланса (через энтальпию фазы и температуру), а в виде уравнения теплопроводности (только через температуру фазы  $T_s$ ). Это связано с тем, что структура S-фазы не конкретизируется, в частности, не производится явное описание компонентов S-фазы. Однако если имеется массообмен на G-S фазовой границе для компонентов G-фазы, то эти же компоненты

должны присутствовать в «связанном» состоянии и в S-фазе. Поэтому источниковый член, описывающий межфазный теплообмен  $S_{Tg}^{(gs)}$  в G-фазе должен быть согласован с источниковым членом  $S_{Ts}^{(gs)}$  в уравнении (5.1). Рассмотрим это обстоятельство подробнее.

Если S-фаза состоит из набора компонентов, то уравнение баланса энергии S-фазы можно представить в виде:

$$\frac{\partial \rho_s (1-\varphi) H_s}{\partial \tau} = \text{div} \left( (1-\varphi) \underline{\lambda}_s \otimes \nabla T_s \right) + S_{Ts}^{(*)} + S_{Ts},$$

$$H_s = \sum_k C_{s,k} H_{s,k}$$
(5.2)

где

$H_{sk}$  - энтальпия компонентов S-фазы,

$C_{sk}$  - массовые доли компонентов фазы, удовлетворяющие уравнениям баланса массы

$$\frac{\partial (1-\varphi) \rho_s C_{s,k}}{\partial \tau} = -S_{Ck}^{(gs)},$$
(5.3)

где  $S_{Ck}^{(gs)}$  - массовые источники *общих* компонентов G- и S-фаз. Источниковые члены теплообмена на фазовой границе связаны очевидным соотношением:

$$S_{Tg}^{(gs)} + S_{Ts}^{(*)} = 0,$$
(5.4)

Переходя от уравнения (5.2) к уравнению теплопроводности (4.1) можно получить связь между источниковыми членами межфазного взаимодействия в этих уравнениях

$$S_{Ts}^{(gs)} = S_{Ts}^{(*)} + \sum_k S_{Ck}^{(gs)} H_{s,k}$$
(5.5)

Используя это соотношение и (5.4), (5.43), можно получить следующее выражение для источника энергии в уравнении теплопроводности, записанного относительно  $T_s$ :

$$S_{Ts}^{(gs)} = \left( \sum_k \max(-S_{Ck}^{(gs)}, 0) \cdot \bar{c}_k + \alpha_{gs} \tilde{A}_{gs} \right) (T_g - T_s) - \sum_k \max(-S_{Ck}^{(gs)}, 0) \cdot \Delta H_{gs,k},$$

$$\bar{c}_k = \frac{H_{g,k}(T_g) - H_{g,k}(T_s)}{T_g - T_s}$$
(5.6)

 **Замечания:**

- 1) Уравнения баланса массы компонент S-фазы типа (5.3) явно не присутствуют в коде Anes. Для описания этих полей (если необходимо) пользователь может использовать собственные Ф-переменные, связанные с S-фазой (см. раздел 6).
- 2) В коде при обработке результатов расчета с помощью специальных подпрограмм пользователя можно рассчитать интегралы от источниковых членов межфазного взаимодействия (4.43) и (5.6). При этом следует помнить, что абсолютное значение источника в уравнении для  $T_g$  зависит от выбранных отсчетов энтальпий компонентов G-фазы  $H_{0k}$ . В уравнении энергии для  $T_s$  такого отсчета нет и источник определен однозначно. Поэтому неравенство интегралов источников для  $T_s$  и  $T_g$  не должно вызывать удивления.
- 3) Тепловой эффект фазового перехода (последний член в уравнениях (4.43) и (5.6)) относится к G-фазе, если массовый поток положителен (направлен в сторону G-фазы), и к S-фазе, если массовый поток отрицателен.

### 5.1.2 Межфазный теплообмен на границах раздела фаз G-S фаз

Источники межфазного теплообмена на явных границах раздела фаз (патчи типа "FS\_...") рассчитываются автоматически из условия сопряжения плотностей тепловых потоков (так же, как и для G-фазы (4.24)):

$$(\lambda_{\text{eff}} \nabla T_g) \cdot \mathbf{n}_{\text{sg}} = (\underline{\lambda}_s \otimes \nabla T_s) \cdot \mathbf{n}_{\text{sg}} \quad (5.7)$$

Здесь  $\mathbf{n}_{\text{sg}}$  - единичная нормаль к межфазной границе, направленная в сторону G-фазы,  $\underline{\lambda}_s$  и  $T_s$  - тензор коэффициента теплопроводности и температура S-фазы.

### 5.1.3 Описание источников и граничных условий

Для описания источников членов и граничных условий используются стандартные соотношения (4.16) и (4.21), в которых массовый поток *игнорируется!*

В коде Anes на границах раздела Struct-зон с разными материалами (такие границы описываются поверхностными патчами с именем "SS\_...") можно задать термическое сопротивление на границе  $R_T$  (размерность  $(\text{m}^2 \text{K})/(\text{Вт})$ ), которая определяет плотность теплового потока через границу раздела двух материалов 1 и 2 S-фазы:

$$q_{12} = \frac{T_2 - T_1}{R_T}$$

Для задания термического сопротивления необходимо задать псевдо-источник с параметрами:

$$C_{T_s} = \text{SR\_CONTRES}, V_{T_s} = R_T$$

## 5.2 Плавление S-фазы

В коде Anes реализована простая модель, которая позволяет описать *нестационарные* процессы плавления (или затвердевания) неподвижной и непроницаемой S-фазы. Для моделирования используется алгоритм, предложенный Пракашем [8, 9].

В этой модели смесь расплавленного и твердого материала рассматривается как смесь двух фаз: жидкой (индекс "l") и твердой ("m"). Основные ограничения модели состоят в следующем:

- 1) плотности и теплоемкости жидкой и твердых фаз одинаковы ( $\rho_s$  и  $c_{ps}$ ),
- 2) плотности и теплоемкости жидкой и твердых фаз постоянны для каждого материала S-фазы, в котором моделируется плавление.

При отсутствии движения тепловое состояние смеси можно описать с помощью уравнения энергии

$$\frac{\partial (\rho_s H_s)}{\partial \tau} = \text{div} (\lambda_s \nabla T_s) + S_{T_s}, \quad H_s = h_l x_l + h_m (1 - x_l), \quad (5.8)$$

$$h_m = c_{ps} (T_s - T_0), \quad h_l = h_m + Q_m$$

Здесь  $H_s$  - энтальпия смеси,  $x_l$  - массовая доля жидкой фазы,  $Q_m$  - теплота плавления (тепловой эффект фазового перехода).

Уравнение (5.8) легко свести к стандартному уравнению энергии S-фазы (5.1):

$$\rho_s c_{ps} \frac{\partial T_s}{\partial \tau} = \text{div} (\lambda_s \nabla T_s) + S_{T_s}^{(I)}, \quad S_{T_s}^{(I)} = S_m = -\rho_s Q_m \frac{\partial x_l}{\partial \tau} \quad (5.9)$$

Это уравнение отличается от стандартного уравнения теплопроводности наличием источника  $S_m$ , учитывающего эффекты плавления или затвердевания.

Для расчета массовой доли жидкой фазы  $x_l$  используется специальная итерационная процедура, предложенная в работе [8]. Уравнение (5.10) получается из условия постоянства температуры смеси в области, через которую проходит фронт плавления (верхний индекс "\*" означает значение с предыдущей итерации,  $T_m$  - температура плавления материала):

$$x_1 = x_1^* + \frac{c_{ps}}{Q_m} (T_s - T_m), \quad x_1 = \max(0, x_1), \quad x_1 = \min(1, x_1) \quad (5.10)$$

### 5.2.1 Активация модели плавления

Для активации модели плавления необходимо добавить в список рассчитываемых Ф-переменных Ф-переменную с именем «Melt». Эта Ф-переменная будет использоваться для хранения массовой доли жидкой фазы  $x_1$ .

Для каждого материала S-фазы, для которого необходимо смоделировать процессы плавления/затвердевания, нужно задать четыре новых *постоянных* свойства с именами:

- 1) «TMelt» - температура плавления  $T_m$ ,
- 2) «QMelt» - теплота плавления  $Q_m$ ,
- 3) «DensMelt» - плотность при температуре плавления.
- 4) «CpMelt» - теплоемкость при температуре плавления.

Использование модели плавления приведено в проекте [e\\_melt\\_04.a](#) в библиотеке примеров кода (секция «Нестационарные задачи»). В этом примере моделируется процесс плавления льда и сравнение результатов расчета с аналитическим решением задачи Стефана.

## 6. Математические модели $\Phi$ -переменных пользователя

Наряду с «предопределенными»  $\Phi$ -переменными в коде можно использовать любое число произвольных  $\Phi$ -переменных. Вид уравнения переноса для этих переменных зависит от их типа:

- 1) FLOW - переменная связана с G-фазой,
- 2) STRUCT - переменная связана с S-фазой.
- 3) SPACE - переменная привязана к пространству расчетной области.

### ① Описание в проекте прикладной задачи:

Тип уравнения для  $\Phi$ -переменной с именем "Ф-имя" определяется оператор секции [PHI Variables]

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

### 6.1 Flow $\Phi$ -переменная

Уравнение для переменной этого типа совпадает с уравнением (3.3):

$$\frac{\partial(\rho_r \Phi)}{\partial \tau} + \text{div}(\rho_r \mathbf{U}_g \Phi - \Gamma_\Phi \nabla \Phi) = S_\Phi^{(gs)} + \phi S_\Phi \quad (6.1)$$

Для расчета эффективного коэффициента диффузии можно использовать два соотношения (3.5):

$$\Gamma_\Phi = \rho_g \left( \frac{\nu_g}{\sigma_\Phi} + \frac{\nu_t}{\sigma_{\Phi,t}} \right), \quad \Gamma_\Phi = D_\Phi + \frac{\rho_g \nu_t}{\sigma_{\Phi,t}}, \quad (6.2)$$

где:  $\sigma_\Phi, \sigma_{\Phi,t}$  - молекулярное и турбулентное «число Прандтля» для данной  $\Phi$ -переменной,  $D_\Phi$  - молекулярный коэффициент «диффузии» для данной  $\Phi$ -переменной.

#### 6.1.1 Источниковые члены и граничные условия

Для задания источников и граничных условий используются соотношения, описанные в разделе 3.

#### 6.1.2 Межфазный теплообмен в зонах Rogous

Для  $\Phi$ -переменных типа FLOW в зонах Rogous предусмотрены стандартные объемные источниковые члены

$$S_\Phi^{(gs)} = \langle M_p \rangle \Phi_{\text{int}} - \langle -M_p \rangle \Phi + C_\Phi (\Phi_{\text{int}} - \Phi)$$

где  $C_\Phi$  - коэффициент обмена «G-фаза - фазовая граница»,  $\Phi_{\text{int}}$  - значение  $\Phi$ -переменной на фазовой границе,  $M_p$  - объемный массовый поток на фазовой границе G-S фаз. Значение этих переменных определяется моделью межфазного взаимодействия.

#### 6.1.3 Межфазный теплообмен на границах раздела G-S фаз

На границах раздела G-S фаз (патчи типа "FS\_...") для произвольных  $\Phi$ -переменных *не предусмотрено* задания внутренних источниковых членов. Если такие источники имеются, то пользователь может их задать самостоятельно, привязав их к поверхностному FS-патчу.

Важной особенностью уравнения для произвольной  $\Phi$ -переменной является возможность «отключения» отдельных членов (6.1). Можно отключить нестационарный член, конвективную или диффузионную часть плотности потока.

**① Описание в проекте прикладной задачи :**

Для конкретной  $\Phi$ -переменной с именем "Ф-имя" используемая форма уравнения (6.2) определяется операторами секции [PHI Variables]

```
TypeDiff("Ф-имя") = ND_PRAN / ND_PROP
NamDiff("Ф-имя") = <имя свойства>
```

где <имя свойства> - имя свойства для «числа Прандтля» или коэффициента диффузии, используемое для задания свойств материалов G-фазы в секции [Properties].

Если имя свойства (оператор NamDiff) не указано, то по умолчанию для задания свойств используются операторы

```
Prop("<Имя материала>.DDIF.Ф-Имя") = V-переменная           (6.3)
или
Prop("<Имя материала>.PRAN.Ф-Имя") = V-переменная
```

Для включения/отключения членов уравнения используются операторы секции [PHI Variables]:

```
LTurb("Ф-Имя ") = .TRUE.    ! Турбулентная часть коэффициента переноса (6.2)
```

```
LDifX("Ф-Имя ") = .TRUE.    ! Диффузионный член в направлении оси x
```

```
LDifY("Ф-Имя ") = .TRUE.    ! Диффузионный член в направлении оси y
```

```
LDifZ("Ф-Имя ") = .TRUE.    ! Диффузионный член в направлении оси z
```

```
LTrans("Ф-Имя ") = .TRUE.   ! Нестационарный член
```

```
LConv("Ф-Имя ") = .TRUE.   ! Конвективный член
```

**6.2 Struct  $\Phi$ -переменная**

Эта  $\Phi$ -переменная определена в S-фазе и удовлетворяет следующему уравнению баланса:

$$(1 - \varphi) \rho_s \frac{\partial \Phi}{\partial \tau} = \text{div}((1 - \varphi) D_\Phi \nabla \Phi) + (1 - \varphi) S_\Phi + S_\Phi^{(gs)}, \quad (6.4)$$

Для задания коэффициента диффузии  $D_\Phi$  используется свойство с именем "DDIF.Ф-Имя" и оператор (6.3). Для этого уравнения, как и для уравнения (6.2) можно отключить нестационарный член или диффузионный перенос.

**6.2.1 Источниковые члены и граничные условия**

Для задания источников и граничных условий используются соотношения, описанные в разделе 3. При этом *игнорируется массовая составляющая* источника или граничного условия.

**6.2.2 Межфазный теплообмен в зонах Porous**

Для  $\Phi$ -переменных типа STRUCT в зонах Porous предусмотрены стандартные объемные источниковые члены

$$S_\Phi^{(gs)} = C_\Phi (\Phi_{\text{int}} - \Phi)$$

где  $C_\Phi$  - коэффициент обмена «S-фаза - фазовая граница»,  $\Phi_{\text{int}}$  - значение  $\Phi$ -переменной на фазовой границе. Значение этих переменных определяется моделью межфазного взаимодействия.

**6.3 Space  $\Phi$ -переменная**

$\Phi$ -переменные этого типа не связаны конкретными G- и S- фазами, они определены во всей расчетной области. Уравнение сохранения для этого типа переменной имеет вид

$$\frac{\partial \Phi}{\partial \tau} = \text{div}(D_\Phi \nabla \Phi) + S_\Phi \quad (6.5)$$

Уравнения такого типа возникают при описании процессов, не связанных с распределением масс G- или S-фаз, например, процессов электродинамики, магнитостатики и т.д.



Для задания коэффициента диффузии  $D_\Phi$  используется свойство с именем "DDIF.Ф-Имя" и оператор (6.3)

Для Space-переменных также имеется возможность отключения диффузионного переноса и нестационарного члена. Например, для реализации «кинетического» уравнения

$$\frac{\partial \Phi}{\partial \tau} = S_\Phi \quad (6.6)$$

необходимо отключить в уравнении диффузионные члены.

## 7. Двухфазные модели G-фазы

В настоящее время в коде реализуется три двухфазные модели:

- 1) VOF алгоритм для моделирования процессов в двухфазной G-фазе с явным выделением межфазной поверхности.
- 2) Гомогенная модель двухфазного потока,
- 3) LDP модель для моделирования двухфазного потока с дисперсной жидкой или твердой фазой. В этой модели для моделирования частиц используется лагранжевый подход.

### 7.1 VOF модель двухфазного потока

Текущее состояние VOF модели описывается в документе aneVOF21.pdf, который входит в состав документации Anes.

### 7.2 LDP модель двухфазного потока

LDP (Lagrangian Discrete Phase) модель используется для описания двухфазного потока, одна из фаз которого является сплошной средой, а вторая фаза представляет собой набор дискретных частиц. В этой модели для моделирования сплошной (несущей) фазы используются однофазные уравнения Навье-Стокса, а для моделирования дискретной фазы вводится набор частиц, траектории которых рассчитываются на основе уравнений движения Ньютона. Взаимодействие между фазами сводится к обмену массой, импульсом и энергией. Для этого в уравнениях несущей фазы вводятся удельные объемные источники массы, импульса и энергии, а в уравнениях движения частиц появляются соответствующие «точечные» источники (например, в уравнении движения частицы появляется сила трения частицы о несущую фазу). Для «синхронизации» этих источников с каждой моделируемой частицей связывается расход частиц  $\dot{n}_p$  - число реальных частиц, движущихся вдоль траектории в единицу времени. Обычно вместо этого параметра используется более наглядная величина - массовый расход частиц  $\dot{m}_p = m_p \dot{n}_p$  (кг/с), где  $m_p$  - масса частицы.

В текущей версии кода Anes реализована *упрощенная* версия LDP модели, основные особенности которой состоят в следующем:

- 1) не учитывается обратное влияние частиц на несущую фазу,
- 2) отсутствует массообмен и теплообмен между частицами и несущей фазой,
- 3) не учитываются процессы распада и столкновения частиц,
- 4) учитываются только две силы в уравнениях движения частиц - сила трения о несущую фазу и сила тяжести,
- 5) все частицы рассматриваются как сферы различных диаметров,
- 6) для моделирования взаимодействия частиц с твердыми стенками используется упрощенная модель Баи-Госмана [12],
- 7) модель реализована только для неструктурной сетки в декартовой системе координат.

Поскольку в модели не учитывается обратного влияния частиц на несущую фазу, то поля несущей G-фазы можно рассчитать независимо с помощью однофазных уравнений G-фазы кода Anes. Для расчета движения частиц необходимо поле вектора скорости  $\mathbf{u}_g = (u_{gx}, u_{gy}, u_{gz})$ . Если учитывается влияние турбулентных пульсаций на траектории частиц (турбулентная дисперсия частиц), то дополнительно необходимы поля турбулентной энергии  $k$  и скорости ее диссипации  $\varepsilon$  (или  $\omega$ ). В коде Anes блок расчета движения частиц вызывается после расчета уравнений G-фазы, поэтому для моделирования различных ре-

жимов дискретной фазы в *стационарном случае*, не изменяющих поля несущей фазы, можно использовать рестарт расчета с контрольной точки. В нестационарных задачах частицы запускаются с заданным шагом по времени, поэтому движение частиц необходимо рассчитывать одновременно с расчетом полей на шагах по времени.

Для интеграции LDP модели с другими физическими моделями предусмотрен *смешанный режим использования LDP*: стационарная LDP модель может быть вызвана в нестационарном расчете гидродинамики на выбранном шаге по времени (подробнее - смотрите п. 7.2.9).

Для моделирования дискретной фазы с участка границы расчетной области (РО) «запускаются»  $N_p$  «представительных» частиц, которые на самом деле описывают движение группы частиц с *одинаковыми* параметрами по *одной* траектории. Далее для простоты термин «представительная» будет опускаться. В качестве границы запуска частиц обычно используется входная граница, через которую несущая G-фаза входит в расчетную область. Для каждой частицы  $k = 1 \dots N_p$  на входной границе задаются (рисунок 3.1):

- 1) ее начальное положение  $\mathbf{x}_{0,k} = (x_{0,k}, y_{0,k}, z_{0,k})$  и скорость  $\mathbf{u}_{p0,k} = (u_{px0,k}, u_{py0,k}, u_{pz0,k})$ ;
- 2) параметры частицы:  $d_p$  - диаметр (м),  $\rho_p$  - плотность частицы ( $\text{кг/м}^3$ );
- 3) расход частиц, связанный с данной представительной частицей -  $\dot{n}_{p,k}$  (1/сек);

В Anes наряду с расходом частиц используются еще две характеристики: массовый расход  $\dot{m}_{p,k}$  (кг/сек) и объемный расход  $\dot{v}_{p,k}$  ( $\text{м}^3/\text{сек}$ ), связанные с  $\dot{n}_{p,k}$  очевидными соотношениями:

$$\begin{aligned} \dot{v}_{p,k} &= v_{p,k} \dot{n}_{p,k}, & \dot{m}_{p,k} &= \rho_{p,k} \dot{v}_{p,k}, \\ v_{p,k} &= \frac{\pi}{6} d_{p,k}^3, \end{aligned} \quad (7.1)$$

где  $v_{p,k}$  - объем частицы ( $\text{м}^3$ ),  $m_{p,k}$  - масса частицы (кг).

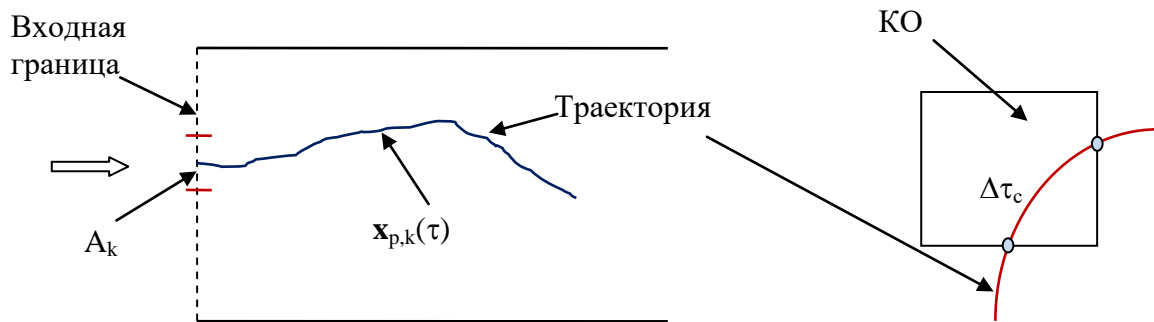


Рисунок 7.1 - Основные параметры движения частицы

Используя уравнения, описанные ниже, можно рассчитать траекторию и скорость движения каждой частицы. Эту траекторию проще всего описать зависимостью вектора положения частицы в РО от времени движения частицы -  $\mathbf{x}_{p,k}(\tau)$ . Аналогичным образом описывается и скорость движения частицы -  $\mathbf{u}_{p,k}(\tau)$ .

Зная эти параметры каждой частицы можно для каждого контрольного объема (КО) сетки расчетной области рассчитать время нахождения частицы в КО -  $\Delta\tau_c$  (см. рисунок 7.1). Используя эти времена можно рассчитать поле объемной доли дисперсной фазы  $\varphi_1(x,y,z)$  в эйлеровом представлении ( $\varphi_1 = 1 - \varphi$ , где  $\varphi$  - объемная доля газовой несущей фазы). Для КО с индексом  $cell$  и объемом  $\Delta V_{cell}$  :

$$\varphi_{1,\text{cell}} = \frac{1}{\Delta V_{\text{cell}}} \sum_{k=1}^{N_p} v_{p,k} \dot{n}_{p,k} \Delta \tau_{c,k} . \quad (7.2)$$

Здесь суммирование проводится по всем траекториям частиц (для частиц, которые не проходят через данный КО,  $\Delta \tau_{c,m} = 0$ ). Аналогичным образом можно найти эйлерову скорость дискретной фазы в КО:

$$\mathbf{u}_{1,\text{cell}} = \frac{1}{\sum_k m_{p,k} \dot{n}_{p,k} \Delta \tau_{c,k}} \sum_m m_{p,k} \mathbf{u}_{p,k} \dot{n}_{p,k} \Delta \tau_{c,k} , \quad (7.3)$$

плотность фазы:

$$\rho_{1,\text{cell}} = \frac{\rho_p}{\Delta V_{\text{cell}}} \sum_k v_{p,k} \dot{n}_{p,k} \Delta \tau_{c,k} ,$$

и массовую долю дискретной фазы ( $x_1 = 1 - x$ , где  $x$  - массовая доля газовой фазы):

$$x_{1,\text{cell}} = \frac{\rho_{1,\text{cell}}}{\rho_g (1 - \varphi_{1,\text{cell}}) + \rho_{1,\text{cell}}} \quad (7.4)$$

Это эйлерово представление дискретной фазы позволяет рассчитать объемные источники массы, импульса и энергии в уравнениях несущей фазы. Поскольку в текущей версии LDP модели Anes влияние дискретной фазы на несущую фазу не учитывается, то эти поля носят в основном информационный характер.

Аналогичная «обратная» связь между эйлеровым и лагранжевым представлением используется в LDP модели Anes для расчета расхода частиц  $\dot{n}_{p,k}$ . Пусть на входной границе известны поля объемной доли G-фазы  $\varphi_{g0}$  или объемной доли  $\varphi_{10}$  и нормальной компоненты скорости дискретной фазы  $u_{10}$ . Разобьем входную границу на площадки с площадью  $A_k$ , число которых равно числу запускаемых частиц  $N_p$ . В этом случае за время  $\delta t$  через площадку  $A_k$  пройдет объем частиц  $= \varphi_{10} \cdot A_k \cdot u_{10} \cdot \delta t$ . С другой стороны этот поток равен  $\dot{n}_p \cdot v_p \cdot \delta t$ . Отсюда следует, что с  $k$ -той частицей будет связан расход:

$$\dot{n}_{p,k} = \frac{\varphi_{10} \cdot A_k \cdot u_{10}}{v_{p,k}} \quad (7.5)$$

**❗ Описание в проекте прикладной задачи :**

Вся настройка параметров LDP модели осуществляется через операторы секции интерфейса солвера [Special Data. Для активизации модели необходимо в эту секцию добавить оператор:

L("IsUseLDPM") = .True.

Замечание: Траектории частиц в коде сохраняются в одномерных массивах переменной длины. Начальный размер массива по умолчанию выбирается равным:

MaxTrackSize = 2 \*(NFX + NFY + NFZ)

В редких случаях этот размер может быть очень большим и при большом числе частиц может не хватить оперативной памяти. В этом случае нужно явно указать начальный размер с помощью оператора:

I("MaxTrackSize") = 1000

Заметим, что размер может быть любым (код сам его увеличит, если необходимо).

### 7.2.1 Начальные условия для частиц

В коде предусмотрены два способа задания начальных характеристик частиц:

- 1) UserSet - начальные параметры частиц задаются в adm-файле,
- 2) VolFrac - начальные параметры частиц рассчитываются автоматически на основе соотношения (7.5).

Выбор способа задания начальных условий задается оператором секции [Special Data]:

```
C("TypeInLet") = "UserSET" / "VolFrac"
```

В обоих вариантах начальное положение частиц задается на *поверхности* одного патча входной границы - PatchInLet. Имя этого патча задается оператором секции [Special Data]:

```
C("PatchInlet") = <Имя патча>
```

При использовании варианта UserSet параметры задаются в текстовом файле следующего формата. Первая строка рассматривается как комментарий и может содержать произвольный текст. Далее идут строки с параметрами отдельной частицы, разделенные пробелами или запятыми:

```
x0 y0 z0 upx0 upy0 upz0 dp DensP MassRate
```

Здесь

- (x0,y0,z0) - начальные координаты частицы, м;
- (upx0,upy0,upz0) - компоненты вектора скорости частицы, м/с;
- dp - диаметр частицы  $d_p$ , м;
- DensP - плотность частицы  $\rho_p$ , кг/м<sup>3</sup>;
- MassRate - массовый поток  $\dot{m}_p$ , связанный с частицей, кг/с.

Количество строк, описывающих частицы должно соответствовать числу частиц  $N_p$ , задаваемому оператором:

```
I("NoParticles") = <Число частиц в файле>
```

Координаты частиц должны «лежать» на поверхности входного патча (с точностью до размера приграничных КО). При обработке файла описания частиц Решатель может скорректировать начальные координаты (x0,y0,z0) так, чтобы частица находилась в плоскости граничной грани, в область которой попала частица.

Заметим, что параметры частиц никак не связаны друг с другом, их значение определяется пользователем. Например, можно задать частицы в виде струи спрея: в этом случае все частицы будут иметь одну начальную скорость, но разные направления вектора скорости. Для удобства просмотра траекторий частиц в постпроцессоре Anes каждой частице присваивается имя "Part.XX.1", где XX - номер частицы (1 ...  $N_p$ ). Если задан оператор

```
I("WFM.NoPartJet") = <Число частиц в «группе»>
```

то частицам присваиваются имена, аналогичные модели VolFrac: - "Part.XX.YY". Значения XX и YY рассчитываются по алгоритму:

```
XX = Np/NoPartJet;    YY = mod(Np,NoPartJet)
if (ll_2 /= 0) then  XX = XX + 1  else YY = NoPartJet
endif
```

При использовании второго способа задания начальных условий - VolFrac - расположение частиц формируется по другому алгоритму.

Все «запускаемые» частицы разбиваются на  $N_g$  групп по  $m_g$  частиц в каждой группе ( $N_p = N_g * m_g$ ). Все частицы в группе имеют одинаковые параметры и начальные данные, кроме диаметра частиц  $d_p$ . Распределение частиц в группе по диаметрам определяется моделью, описанной ниже.

Начальные координаты группы частиц привязываются к центру грани поверхностного патча PatchInLet. Поэтому число групп  $N_g$  равно числу граней входного патча. Если граней слишком много, то можно задать «шаг» отбора граней патча с помощью оператора:

I("PStepFaces") =  $m_{step}$

В этом случае будут отобраны грани патча с шагом  $m_{step}$  (по умолчанию  $m_{step} = 1$ ). Число частиц в группе задается пользователем с помощью оператора:

I("NoPartInGroup") =  $m_g$

Если оператор не задан, то в группе находится одна частица. При использовании второго варианта частице присваивается имя "Part.XX.YY", где XX - номер группы, YY - номер частицы в группе.

. Плотность частиц считается постоянной и одинаковой для всех групп и задается оператором:

R("DensPartIN") = <значение плотности>

Вектор скорости частиц группы принимается равным вектору скорости G-фазы в точке «запуска» частиц. Для расчета расходов частиц  $\dot{n}_{p,k}$ ,  $\dot{m}_{p,k}$  и  $\dot{v}_{p,k}$  используются соотношения (7.1) и (7.5), в которых  $A_k$  - это площадь грани патча, с которой производится частицы группы, а объемная доля дискретной фазы  $\phi_{i0}$  задается оператором:

R("VolFracIN") = <объемная доля>

### 7.2.2 Распределение частиц группы по размерам

В реальных дисперсных двухфазных потоках частицы дисперсной фазы имеют различные размеры и формы. Для учета этого факта в коде Anes при использовании варианта VolFrac можно задать распределение частиц в группе по диаметрам.

Если необходимо задать в группе одинаковые диаметры частиц, то необходимо указать число частиц в группе NoPartInGroup со знаком минус (отрицательным). В этом случае диаметр всех частиц задается оператором:

R("DPartMEAN") = <диаметр частиц>

Если число частиц в группе NoPartInGroup *положительно*, и оно больше 2, то для распределения частиц по диаметрам используется распределение Росина-Раммлера. Это распределение, записанное через объемную (или массовую) долю частиц в группе, имеет вид:

$$f_v(x) = \frac{C_r}{d_m} \left( \frac{x}{d_m} \right)^{C_r-1} \exp \left[ - \left( \frac{x}{d_m} \right)^{C_r} \right] \quad (7.6)$$

Смысл функции  $f_v(d)$  следующий:  $f_v(d) \cdot \Delta d$  - это массовая или объемная доля частиц с диаметрами в интервале  $d - (d + \Delta d)$ . В расчетах обычно используется «накопительная» форма распределения:

$$Y_v(d) = \int_d^{\infty} f_v(x) dx = \exp \left[ - \left( \frac{d}{d_m} \right)^{C_r} \right] \quad (7.7)$$

Значение функции  $Y_v(d_1)$  определяет объемную (массовую) долю частиц в группе с диаметром  $d \geq d_1$ . Распределения (7.6) и (7.7) содержат два «свободных» параметра:  $d_m$  - средний диаметр частиц и  $C_r$  - постоянную распределения (постоянная Росина). Эти параметры задаются операторами:

R("DPartMEAN") = <диаметр частиц>  
 R("Coef\_Rosin") = <постоянная Росина>

В коде Anes соотношения (7.6) и (7.7) используются следующим образом. Для всех групп, состоящих из  $m_g$  частиц, задаются средний диаметр  $d_m$  и параметр  $C_r$ . На первом этапе определяются минимальный и максимальный диаметр частиц в группе по соотношениям:

$$d_{\min} = d_m \left[ 1 - \ln \left( \frac{0.995}{0.368} \right) \right]^{1/C_r}, \quad d_{\max} = d_m \left[ 1 - \ln \left( \frac{0.05}{0.368} \right) \right]^{1/C_r} \quad (7.8)$$

Здесь 0.346 - это значение функции  $Y_v(d_m)$ , 0.995 и 0.005 - значения  $Y_v$  для минимального и максимального диаметра. На втором этапе интервал  $d_{\min} - d_{\max}$  равномерно разбивается на  $m_g$  частиц:

$$d_{p,m} = d_{\min} + \Delta d \cdot (m - 1), \quad \Delta d = \frac{d_{\max} - d_{\min}}{m_g - 1} \quad (7.9)$$

На третьем этапе рассчитываются доли расходов, связанные с частицами группы:

$$\dot{n}_{p,m} = \dot{n}_p \left[ Y_v \left( d_{p,m} - \frac{\Delta d}{2} \right) - Y_v \left( d_{p,m} + \frac{\Delta d}{2} \right) \right] \quad (7.10)$$

Здесь  $\dot{n}_p$  - расход частиц, связанный с группой.

Распределение частиц по размерам выводится в листинг Решателя в виде:

```

-(Распределение объемной фракции по диаметрам)-
- # ---- DPART, мкм --- VolFrac --- VolFrac/DeltaDT
1  6.285      4.6403E-03  3.8945E-03
2  7.476      8.4609E-03  7.1011E-03
3  8.668      1.4044E-02  1.1787E-02
4  9.859      2.1664E-02  1.8182E-02
5  11.05      3.1444E-02  2.6390E-02
6  12.24      4.3256E-02  3.6304E-02
7  13.43      5.6616E-02  4.7517E-02
8  14.63      7.0598E-02  5.9252E-02
9  15.82      8.3820E-02  7.0348E-02
10 17.01      9.4549E-02  7.9353E-02
11 18.20      0.1010      8.4742E-02
12 19.39      0.1016      8.5272E-02
13 20.58      9.5775E-02  8.0382E-02
14 21.77      8.4001E-02  7.0500E-02
15 22.97      6.8013E-02  5.7082E-02
16 24.16      5.0391E-02  4.2293E-02
17 25.35      3.3835E-02  2.8397E-02
18 26.54      2.0371E-02  1.7097E-02
19 27.73      1.0871E-02  9.1241E-03
20 28.92      5.0789E-03  4.2626E-03
Сумма VolFrac .....= 1.000
    
```

### 7.2.3 Нестационарные задачи

При решении нестационарной задачи частицы в каждой группе ее частицы «запускаются» со своей входной грани с шагом  $\Delta\tau_{inj}$ :

$$\Delta\tau_{inj} = \frac{\tau_{end} - \tau_{beg}}{m_g - 1},$$

где  $\tau_{beg}$ ,  $\tau_{end}$  - время начала и окончания запуска частиц. Эти параметры задаются операторами секции [Special Data]:

R("TimeInjectBeg") = < $\tau_{beg}$  - время начала запуска>

R("TimeInjectEnd") = < $\tau_{end}$  - время окончания запуска>

### 7.2.4 Уравнение движения частицы

Уравнение движение частицы - второй закон Ньютона - можно записать в форме (для простоты здесь индекс частицы опущен):

$$m_p \frac{d\mathbf{u}_p}{d\tau} = D_p (\mathbf{u}_g - \mathbf{u}_p) + m_p \mathbf{g} \left( 1 - \frac{\rho_g}{\rho_p} \right), \quad (7.11)$$

$$D_p = \frac{1}{2} \rho_g A_p C_D u_r, \quad u_r = |\mathbf{u}_g - \mathbf{u}_p|, \quad A_p = \frac{\pi}{4} d_p^2$$

Здесь  $\rho_g$  - плотность несущей газовой фазы ( $\text{кг}/\text{м}^3$ ),  $\mathbf{u}_g$  - вектор скорости несущей фазы ( $\text{м}/\text{с}$ ),  $C_D$  - коэффициент сопротивления частицы,  $u_r$  - модуль относительной скорости частицы,  $\mathbf{g}$  - вектор ускорения свободного падения.

Для расчета коэффициента сопротивления используется соотношение работы [13]:

$$C_D = \frac{24}{Re} (1 + 0.15 Re^{0.687}) + \frac{0.42}{1 + 4.25 \cdot 10^4 \cdot Re^{-1.16}}, \quad Re = \frac{u_r d_p}{\nu_g} \quad (7.12)$$

Здесь  $\nu_g$  - коэффициент кинематической вязкости несущей фазы,  $Re$  - число Рейнольдса, рассчитанное по относительной скорости частицы.

Второй силовой источник в (7.11) связан с силой плавучести, действующей на частицу. Декартовы компоненты вектора  $\mathbf{g}$  задаются операторами секции [Internal Source]:

AGravX= 0 / < $g_x$ >

AGravY= 0 / < $g_y$ >

AGravZ= 0 / < $g_z$ >

По умолчанию это член включен. Для принудительного отключения силы тяжести (при ненулевом векторе  $\mathbf{g}$ ) используется оператор секции [Special Data]:

L("IsUseGravForce") = .False.

В текущей версии LDP модели учитывается только две силы, действующие на частицу. В версиях этой модели других CFD кодов учитываются и другие силы:

- 1) сила, связанная с «виртуальной» присоединенной массой и играющая существенную роль, если  $\rho_p \ll \rho_g$ ;
- 2) термофорез - сила, связанная с наличием градиента температуры несущей фазы,
- 3) броуновская сила, связанная со случайными пульсациями молекул несущей фазы;
- 4) сила, связанная с градиентом давления в несущей фазе.

Эти силы будут включены в модель в следующих версиях.

Важным фактором, определяющим движение частицы, является турбулентность несущего потока. В LDP модели это влияние учитывается не путем введения какой-то эффективной силы (как это делается при учете броуновского движения), а путем временной коррекции скорости несущей фазы. Подробнее эта модель описана ниже.



Уравнение (7.11) можно переписать в другой форме, которая будет использоваться при моделировании воздействия турбулентных пульсаций на движение частицы:

$$\frac{d\mathbf{u}_p}{d\tau} = \frac{(\mathbf{u}_g - \mathbf{u}_p)}{\tau_r} + \mathbf{g} \left( 1 - \frac{\rho_g}{\rho_p} \right), \quad \tau_r = \frac{m_p}{D_p} \quad (7.13)$$

Здесь  $\tau_r$  - время релаксации скорости частицы.

### 7.2.5 Численный алгоритм решения уравнений движения

При численном решении уравнения (7.11) шаг интегрирования по времени выбирается из условия нахождения траектории частицы в одном КО. Это связано с тем, что для расчета эйлеровых полей (7.2) - (7.4) необходим именно этот интервал времени. Алгоритмически этот алгоритм реализуется следующим образом.

Шаг по времени определяется из уравнения

$$\mathbf{x}_p^1 = \mathbf{x}_p^0 + \mathbf{u}_p^0 \Delta\tau \quad (7.14)$$

где индексы 0 и 1 соответствуют точкам входа и выхода из ячейки, которые расположены на двух гранях КО. Для определения шага по времени это уравнение рассматривается как уравнение прямой, а шаг по времени рассчитывается из условия пересечения этой прямой с «выходной» гранью 1.

Скорость на выходе рассчитывается с помощью уравнения (7.11), записанного в координатном представлении:

$$\frac{du_{p,i}}{d\tau} = A_i - B u_{p,i}, \quad i = x, y, z \quad (7.15)$$

$$A_i = \frac{D_p}{m_p} u_{g,i} + g_i \left( 1 - \frac{\rho_g}{\rho_p} \right), \quad B = \frac{D_p}{m_p}$$

Это уравнение легко проинтегрировать явно:

$$u_{p,i}^1 = u_{p,i}^0 e^{-B\Delta\tau} + \frac{A_i}{B} (1 - e^{-B\Delta\tau}) \quad (7.16)$$

Таким образом, траектория частицы складывается из участков траектории, расположенных в разных КО.

Пользователь может разбить шаг по времени внутри КО на набор суб-шагов. В этом случае траектория частицы представляет собой полилинию, для каждого отрезка которой применяется описанный выше алгоритм. Число суб-шагов рассчитывается по соотношению:

$$N_s = \frac{\Delta\tau}{C_{dt} \tau_r} + 1$$

где  $\tau_r$  - текущее время релаксации,  $C_{dt}$  - безразмерный параметр, задаваемый пользователем с помощью оператора секции [Special Data]:

$$R("dpmCriDTIME") = 1.e5$$

По умолчанию  $C_{dt} = 10^5$  и в КО реализуется один шаг, а траектория представляет собой один отрезок.

Расчет траектории частицы прекращается, либо если частица достигает границы расчетной области (см. пункт 7.2.6 ниже), либо:

- 1) время движения частицы превысило предельного значение TimeMax,
  - 2) частица попала в зону стагнации,
  - 3) возникла ошибка поиска «выходной» для частицы грани КО.
- Максимальное время задается с помощью оператора

R("dpmMaxTime") = <время>

Ограничение времени необходимо, если частица попадает в зону «возратного» движения несущей фазы и начинает двигаться по замкнутой траектории. Если частица попадает в зону с практически нулевыми скоростями несущей фазы и скорость самой частицы так же близко к нулю, то фактически частица «останавливается». Эти два случая в коде рассматриваются как состояние стагнации частицы.

И, наконец, очень редко возникает ситуация, когда алгоритм не может однозначно определить «выходную» грань КО для частицы (точки пересечения уравнения (7.14) либо не найдена, либо их несколько). В этом случае расчет траектории прекращается. Состояние окончания расчета траектории для каждой частицы выводится в листинг Решателя в виде:

```
Частица= "Part.18.16", Конец= NoFACES !!!, iCELL= 1539, NoCrossFace=2
Частица= "Part.12.1", Конец= STAGNATION, iCELL= 11239
```

Кроме того, в листинг Решателя выводится статистика состояний окончания расчета траекторий частиц в виде:

```
---(число частиц по состояниям окончания)---
Всего частиц.....=      820, 100 %
Выход (EXIT).....=       81 ,   9.9 %
Прилипание (STICK).....=    733 ,  89.4 %
Испарение (SPLASH).....=     4 ,   0.5 %
Стагнация (STAG).....=     0 ,   0.0 %
Ошибка (NOFACES).....=     2 ,   0.2 %
```

Состояния Exit, Stick и Splash будут описаны ниже.

### 7.2.6 Модель турбулентной дисперсии частиц

Для моделирования влияния турбулентности на движение частицы используется «стохастическая» ЕИМ (Eddy Interaction Model) модель, предложенная Госманом [14]. Суть этой модели заключается в следующем.

Поле скорости несущей фазы в уравнении (7.11) представляется в виде:

$$\mathbf{u}_g = \bar{\mathbf{u}}_g(x, y, z) + \mathbf{u}'(\tau) \quad (7.17)$$

Здесь  $\bar{\mathbf{u}}_g$  - поле скорости несущей фазы, полученное при решении уравнений G-фазы с использованием k-ε или k-ω модели турбулентности,  $\mathbf{u}'(\tau)$  - «псевдослучайные» пульсации скорости вдоль траектории частицы. Естественно, что для получения достоверных результатов необходимо запускать не одну «представительную» частицу, а много (при каждом новом запуске пульсации будут разные). В коде Anes это можно реализовать одновременно с разбиением частиц по размерам (см. пункт 7.2.2).

Каждая компонента вектора  $\mathbf{u}'(\tau)$  представляет собой *ступенчатую* функцию от времени со случайной амплитудой и шириной ступеньки по времени  $T_t$ .

Амплитуда пульсации каждой компоненты рассчитывается по соотношению

$$\mathbf{u}' = \mathbf{N} \cdot \mathbf{u}_T, \quad \mathbf{u}_T = \sqrt{\frac{2 \cdot k}{3}}, \quad \mathbf{N} = (r_g(), r_g(), r_g()). \quad (7.18)$$

Здесь k и ε - значения турбулентной энергии и скорости диссипации в точке нахождения частицы (эти поля рассчитаны при решении уравнений G-фазы),  $\mathbf{N}$  - вектор, компоненты которого получены случайным образом с помощью функции  $r_g()$ ,  $r_g()$  - функция случайного значения, вероятность которого φ удовлетворяет нормальному распределению Гаусса с нулевым средним значением и дисперсией, равной 1:

$$\phi = \frac{1}{\sqrt{2\pi}} e^{-\frac{r_g^2}{2}}$$

В текущей версии кода реализованы две EIM модели: ConstEIM и ValAllEIM. Выбор модели определяется оператором:

C("dpmModelEIM") = "VarAllEIM"/"ConstEIM"

По умолчанию используется VarAllEIM модель.

### Модель ConstEIM

В этой модели случайный вектор  $\mathbf{N}$  и амплитуда пульсации  $u_T$  остаются постоянными в течение времени  $T_i$ .

Интервал времени смены текущего вихря  $T_i$  выбирается как минимум из двух времен:

- 1)  $T_e$  - время «жизни» турбулентного вихря,
- 2)  $T_{cr}$  - время «нахождения» частицы в вихре (время «пересечения» частицей вихря).

Для расчета времени «жизни» турбулентного вихря можно использовать два соотношения:

статическое время жизни (dpmlsRandomTE = .False.):

$$T_e = 2C_{TE} \frac{k}{\varepsilon},$$

и стохастическое время жизни (dpmlsRandomTE = .True.):

$$T_e = -C_{TE} \frac{k}{\varepsilon} \ln[r_{01}(0)] \quad (7.19)$$

Здесь  $r_{01}(0)$  - случайное число в интервале  $[0,1]$  с равномерным распределением вероятности. Для постоянной  $C_{TE}$  в разных работах предлагаются различные значения. Чаще всего используются значения

$$2 \cdot C_{TE} = \sqrt{\frac{3}{2}} \cdot C_\mu^{1/2} \quad \text{и} \quad \sqrt{\frac{3}{2}} \cdot C_\mu^{3/4},$$

которым соответствует интервал  $C_{TE} = 0.1 - 0.18$ . В коде по умолчанию используется значение  $C_{TE} = 0.15$ .

При использовании  $k$ - $\omega$  модели турбулентности (см. раздел 9.7) скорость диссипации рассчитывается по соотношению

$$\varepsilon = C_\mu k \omega$$

Время нахождения частицы в вихре рассчитывается по соотношению:

$$T_{cr} = -\tau_r \ln\left(1 - \frac{L_e}{\tau_r |\bar{\mathbf{u}}_g - \mathbf{u}_p|}\right), \quad L_e = C_\mu^{3/4} \frac{k^{3/2}}{\varepsilon}, \quad (7.20)$$

где  $\tau_r$  - релаксационное время частицы (см. уравнение (7.13)),  $L_e$  - характерный размер вихря,  $C_\mu$  - постоянная  $k$ - $\varepsilon$  модели, равная 0.09.

В процессе расчета при входе в новый вихрь рассчитываются параметры  $\mathbf{N}$ ,  $u_T$  и  $T_i$  и устанавливается время нахождения частицы в текущем вихре  $\tau_{rest} = T_i$ . При перемещении частицы с шагом  $\Delta t$  это время уменьшается на этот шаг

$$\tau_{rest} = \tau_{rest} - \Delta t$$

При достижении  $\tau_{rest} = 0$  производится смена текущего турбулентного вихря.

### Модель VarAllEIM

В этой модели в текущем вихре сохраняется постоянным случайный вектор направления пульсаций  $\mathbf{N}$ , а амплитуда пульсаций  $u_T$  обновляется при переходе частицы в другую

ячейку сетки. Одновременно корректируется время нахождения частицы в вихре на каждом шаге интегрирования движения частицы по соотношению:

$$\tau_{\text{rest}} = \tau_{\text{rest}} \frac{T_{i,\text{new}}}{T_{i,\text{pre}}},$$

где  $T_{i,\text{new}}$ ,  $T_{i,\text{pre}}$  - времена взаимодействия частицы с вихрем, рассчитанные по турбулентным полям в точках нахождения частицы на предыдущем и текущем шагах по времени. При использовании модели стохастического времени жизни вихря (7.19), значение случайной величины  $r_{01}()$  рассчитывается при входе в новый вихрь и сохраняется постоянным до конца взаимодействия вихрь-частица.

Для активации учета турбулентной дисперсии, задания постоянной  $C_{TE}$  и выбора модели расчета времени жизни вихря используются операторы секции [Special Data]:

```
L("IsPartTURB") = .FALSE.
R("dpmConstTE") = 0.15
L("dpmlsRandomTE ") = .FALSE.
C("dpmModelEIM") = "VarAllEIM"/"ConstEIM"
```

### 7.2.7 Взаимодействие частиц с границей расчетной области

В LDP-модели границы расчетной области разбиваются на три типа: входная граница, выходные границы и твердые стенки.

С входной границе производится «запуск» частиц. При достижении частицей выходной границы расчет ее траектории прекращается и считается, что она покидает расчетную область. Такое окончание расчета траектории описывается состоянием Exit.

Основные сложности возникают при моделировании взаимодействия частицы с твердой стенкой.

Существует достаточно большое количество работ, посвященной этой проблеме. Но все они в качестве первоисточников используют работы Бай-Госмана [12]. Результат взаимодействия частицы со стенкой зависят от многих параметров: температуры стенки, ее шероховатости, наличием жидкой пленки, скорости частицы и т.д. Результатом взаимодействия может быть:

- 1) отражение частицы от стенки с потерей импульса и энергии (но не массы),
- 2) поглощение частицы стенкой или жидкой пленкой,
- 3) отражение частицы с ее распадом на несколько частиц,
- 4) поглощение частицы жидкой пленкой с образованием струи мелких частиц в виде спрея,
- 5) испарения частицы при ударе о нагретую поверхность.

В математических моделях различных CFD кодов используется та или иная степень реализации этих физических процессов. В текущей версии кода Anes используется простая модель, основанная на модели Бай-Госмана. В ней возможны три режима взаимодействия:

- 1) режим «отражения» - ReBound,
- 2) режим «поглощения» - Stick,
- 3) и режим «испарения» - Splash.

Переход от одного режима к другому определяется числами Вебера и Лапласа:

$$We = \frac{\rho_p u_{1,n}^2 d_p}{\sigma}, \quad La = \frac{\rho_p \sigma d_p}{\mu_p^2} \quad (7.21)$$

Здесь  $\sigma$  - коэффициент поверхностного натяжения,  $\mu_p$  - динамическая вязкость частицы,  $u_{1,n}$  - нормальная к стенке компонента скорости падающей частицы. В текущей версии модели используются следующие границы режимов:

$$\begin{aligned}
 \text{«отражения»} & - We_{c1} < We < We_{c2}, \\
 \text{«поглощения»} & - We < We_{c1} \text{ или } We > We_{c2}, \\
 \text{«испарения»} & - We > We_{c3},
 \end{aligned}
 \tag{7.22}$$

Три критических значения числа Вебера задаются операторами

$$\begin{aligned}
 R(\text{"WE_Rebound\_MIN"}) & = <\text{Значение } We_{c1}> \\
 R(\text{"WE_Rebound\_MAX"}) & = <\text{Значение } We_{c2}> \\
 R(\text{"WE_Splash"}) & = <\text{Значение } We_{c3}>
 \end{aligned}$$

При моделировании движения жидких частиц (капель бензина) в воздухе можно использовать критические значения числа Вебера из работы [12]:

$$We_{c1} = 1, We_{c2} = 5, We_{c3} = 1320 \cdot La^{-0.183}$$

Для расчета «граничного» числа Вебера  $We$  необходим коэффициент поверхностного натяжения  $\sigma$ . По умолчанию он принимается равным очень большому значению. Для задания этого коэффициента используется оператор:

$$R(\text{"SigmaG"}) = <\text{значение коэффициента поверхностного натяжения}>$$

При «поглощении» частицы расчет ее траектории прекращается, а массовый поток частицы добавляется к массовому потоку стенки (это фактически масса жидкой пленки). Режим «испарения» обрабатывается аналогично, только его массовый поток не добавляется к массовому потоку стенки.

При возникновении режима «отражения» расчет траектории *продолжается* от точки касания частицы стенки, при этом меняются нормальная и касательная к стенке компоненты скорости частицы в соответствии с соотношениями [12]:

$$u_{2,t} = e_t u_{1,t}, \quad u_{2,n} = -e_n \cdot u_{1,n},
 \tag{7.23}$$

где  $(u_{1n}, u_{1t})$  - нормальная и касательная составляющие скорости падающей частицы,  $(u_{2n}, u_{2t})$  - нормальная и касательная составляющие скорости отраженной частицы,  $e_n, e_t$  - коэффициенты отражения. По умолчанию для всех граничных патчей типа BS и BlockWall (именно эти патчи описывают твердые стенки) задаются коэффициенты отражения, равные единице. Для задания своих значений используются операторы:

$$\begin{aligned}
 R(\text{"<Имя патча>.REFL\_N"}) & = <\text{значение } e_n> \\
 R(\text{"<Имя патча>.REFL\_T"}) & = <\text{значение } e_t>
 \end{aligned}$$

Если одно из значений коэффициентов ( $e_n$  или  $e_t$ ) отрицательно, то для расчета коэффициентов отражения рассчитываются по соотношениям [12]:

$$e_t = \frac{5}{7}, \quad e_n(\theta_1) = 0.993 - 1.76 \cdot \theta_1 + 1.56 \cdot \theta_1^2 - 0.49 \cdot \theta_1^3$$

Здесь  $\theta_1$  - угол падения частицы относительно плоскости стенки в радианах.

В листинг Решателя выводятся для каждой частицы состояние окончания расчета траектории, например:

```

Стенка: ..Отражение, WE_BOUND = 1.17482
Стенка: ..Отражение, WE_BOUND = 4.74283
Стенка: ..Прилипание, WE_BOUND = 9.39740
Частица= "Part.18.20", Конец=STICK, Time= 1.202703E-02
    
```

и статистика состояний окончания расчета траекторий частиц в виде:

```

---(Число частиц по состояниям окончания)---
Всего частиц.....=      820, 100 %
Выход (EXIT).....=       81 ,   9.9 %
Прилипание (STICK).....=    733 ,  89.4 %
Испарение (SPLASH).....=     4 ,   0.5 %
Стагнация (STAG).....=     0 ,   0.0 %
Ошибка (NOFACES).....=     2 ,   0.2 %
    
```

Для удобства анализа интегральных параметров производится суммирование всех расходов частиц, попавших на различные участки границы расчетной области. Эти значения выводятся в листинг Решателя в виде

```

----- (LDPM: интегральные параметры) -----
- # --- Name --- NumRate [1/Сек] --- VolRate [м^3/Сек] --- MassRate [кг/Сек] ---
1UP          0.3813E+09    0.6566E-05    0.5812E-02
2DOWN       0.4122E+09    0.7061E-05    0.6249E-02
3OUT        0.3991E+09    0.2957E-05    0.2617E-02
4IN        -0.1193E+10   -0.1658E-04   -0.1468E-01

- # --- Name --- NumRate, % --- VolRate, % --- MassRate, % --- (100 - MassRate), % --
-
1UP          32.0      39.6      39.6      60.4
2DOWN       34.6      42.6      42.6      57.4
3OUT        33.5      17.8      17.8      82.2

----- (LDPM: испарение (SPLASH)) -----
- # --- Name --- NumRate [1/Сек] --- VolRate [м^3/Сек] --- MassRate [кг/Сек] ---
1UP          0.000      0.000      0.000
2DOWN       0.000      0.000      0.000
3OUT        0.000      0.000      0.000
4IN         0.000      0.000      0.000

```

### 7.2.8 Анализ результата моделирования

Траектории всех частиц Решатель сохраняет в файлах результатов с расширением \*.aprs и \*.vtk. Одновременно с траекторией сохраняются параметры частицы с именами:

- 1) "Time" - время точки траектории  $\tau$ , сек;
- 2) "x", "y", "z" - координаты частицы  $\mathbf{x}_p$ , м;
- 3) "VelPart" - модуль скорости частицы  $|\mathbf{u}_p|$ , м/с;
- 4) "VelPart" - модуль относительной скорости частицы  $|\mathbf{u}_g - \mathbf{u}_p|$ , м/с;
- 5) "RE" - число Рейнольдса, рассчитанное по относительной скорости частицы (7.12);
- 6) "WE" - число Вебера, рассчитанное по свойствам несущей фазы  $We = \frac{\rho_g u_g^2 d_p}{\sigma}$ ;
- 7) "dTime" - текущий шаг по времени, сек;
- 8) "MaxTimeEddy" - время смены текущего турбулентного вихря  $\Delta\tau_t$ , сек;
- 9) "VelTurb" - модуль пульсационной скорости несущей фазы  $|\mathbf{u}'(\tau)|$ , м/с.

Траектории частиц и указанные параметры можно проанализировать в постпроцессоре Anes или в постпроцессоре ParaView.

Если в файле проекта создана внутренняя сеточная переменная пользователя с именем "PVolFrac", то в это поле будет помещена объемная доля дискретной фазы (7.2). Если создана внутренняя переменная с именем "PMassFrac", то в это поле помещается массовая доля дискретной фазы (7.4).

### 7.2.9 Интеграция LDP модели с другими физическими моделями

В текущей версии реализован один механизм интеграции, который использовался авторами кода для описания процесса пленочной конденсации. Сама идея интеграции заключается в следующем:

1. Производится расчет нестационарной гидродинамической задачи (параметр IsSTEADY = .False.).
2. На выбранном шаге по времени производится формирование файла \*.adpm, содержащего параметры частиц для режима UserSet.
3. На *следующем* шаге по времени вызывается расчет траектории частиц по стационарной LDP модели.

Активация этого режима производится оператором секции [Special Data]:

```
I("ILTimeMakeLDPM") = <номер локального шага по времени>
```

Этот оператор задает локальный шаг по времени  $iTimeLocal$  (в отличие от  $iTime$  этот счетчик всегда начинается с единицы при старте любого расчета) на котором внешняя модель должна создать LDPM файл. Заметим, что эта внешняя модель в своем Фортрановском коде, обязана исправить значение параметра I("NoParticles") файла проекта с помощью функции ANES:

```
call SetSPEDAT_I("NoParticles",NoParticles)
```

На локальном шаге по времени  $ILTimeMakeLDPM+1$  будет выполнен *стационарный* расчет движения частиц, параметры которых описаны в файле. В дальнейшем нестационарный расчет будет продолжен без вызова LDPM модели.

Наиболее удобным является следующий алгоритм:

- 1) проводится независимый расчет нестационарной задачи до нужного момента времени.
- 2) активируется данная модель (например с  $ILTimeMakeLDPM=2$ ) и производится старт с контрольной точки.
- 3) после третьего относительного шага можно остановить расчет.

В этом режиме естественно нужно использовать режим UserSet:

```
C("TypeInlet") = "UserSet"
C("PartFile") = "путь к .adpm файлу"
```

Остальные параметры LDP модели описаны выше.

В качестве примера записи ADPM файла приведем подпрограмму записи из модели пленки:

```
subroutine wftCreateLDPM
use acDATA
use usaDATA
use wftDATA
use atTOOLS
character(32) ch32
integer NoParticles
real MassRate
real VPart(3)
!-----
! Создание файла *adpm
!-----
if(.not. GetSPEDAT('PartFile',PartFile)) return ! Получаем имя файла из проекта
if(.not. GetSPEDAT('TypeInlet',ch32)) ch32 = '?'
if(.not. CompareStr(ch32,'UserSet')) return ! Если не UserSet - не создаем
! ----- Локальный шаг по времени создания -----
if(.not. GetSPEDAT('ILTimeMakeLDPM',ILTimeMakeLDPM)) ILTimeMakeLDPM = 0
! ----- Число частиц в группе -----
if(.not. GetSPEDAT('WFM.NoPartJet',NoPartJet)) NoPartJet = 3
! ----- Создаем ОДИН РАЗ на шаге ILTimeMakeLDPM -----
if(ILTimeMakeLDPM /= iTimeLocal) return
!-----
LU_OUT = 101
call userOpenOutFile(LU_OUT,TRIM(PartFile))
write(LU_OUT,'(a)' ' "X" "Y" "Z" "UX" "UY" "UZ" "Diam" "Dens" "MassRate"
NoParticles = 0

do i=1,Film%NoX
ivFILM = i
if(Film%Entr(i) < 1.e-10) cycle ! ---%Entr(i) - плотность масса конденсата, уходящая из пленки
NoParticles = NoParticles + 1
DPart = Film%D(i) ! ---%Entr(i) - толщина пленки
```

```

MassRate = Film%Entr(i) * Film%DX(i)    ! масса конденсата, уходящая из пленки
iFace   = Film%iFace(i)
XJet    = FaceBS(iFACE)%Centre(1)      ! точка запуска частиц
YJet    = FaceBS(iFACE)%Centre(2)
ZJet    = FaceBS(iFACE)%Centre(3)
VPart(1:3) = FaceBS(iFACE)%PN(1:3) * abs(Film%U(i)) ! вектор скорости частиц
! ----- Запускаем частицы одного диаметра -----
do m=1,NoPartJet
  write(LU_OUT,'(9(1x,1pg12.4))') XJet, YJet, ZJet, VPart(1), VPart(2), VPart(3), &
    Dpart, DensF, MassRate
end do
end do
close(LU_OUT)
NoParticles = NoParticles * NoPartJet
!-----
call SetSPEDAT_I("NoParticles",NoParticles)
!-----
call WriteConsole('Создан ADPM файл: '//trim(PartFile)// &
  ' на share iLocTime=//toSTR(iTimeLocal))

end subroutine

```

### 7.2.10 Примеры модели

В подкаталоге `ldpm` демонстрационных примеров кода расположены три примера использования LDP-модели, которые демонстрируют все возможности модели.



## 8. Модели межфазного взаимодействия

Взаимодействие между G- и S- фазами сводится к обмену массой, импульсом и энергией. Способ описания межфазного взаимодействия зависит от возможности (и необходимости) выделения межфазной поверхности. В пакете предусматривается два способа описания межфазного взаимодействия.

1. *Пористая (дисперсная) модель* - в этой модели межфазная поверхность явно не выделяется, обменные источниковые члены в соответствующих уравнениях сохранения носят объемный характер. Эта модель используется для описания G-S взаимодействия в зоне Porous.
2. *Модель с явно выделенной границей* - в этой модели фазовая поверхность явно выделена и фиксирована в пространстве. Обменные источниковые члены формируются на основе поверхностных источников межфазного обмена. Эта модель используется для описания G-S взаимодействия на поверхностях раздела зон Struct-Flow.

Соотношения для модели с явно выделенной границей для различных Ф-переменных приведены выше. В данной главе основное внимание уделено пористой модели G-S взаимодействия.

Для каждой зоны Porous необходимо вычислить следующие источниковые члены в уравнениях сохранения Ф-переменных (ниже верхний индекс "gs" далее опускается) :

$\phi$	-пористость;
$M_P$	- объемный массовый поток из S в G фазу, отнесенный к единице объема пространства, знак потока положителен, если он направлен в G-фазу;
$M_{Ck}$	-массовые потоки компонентов G-фазы ( $k = 1 \dots 8$ ) $M_P = \sum_k M_{Ck};$
$S_{ugx}, S_{ugy}, S_{ugz}$	-объемные силы трения в уравнениях движения G-фазы;
$S_{Tg}, S_{Ts}$	-объемный теплообмен между фазами;
$S_\Phi$	-обменные источники для произвольных Ф-переменных типа FLOW или STRUCT.

### 8.1 Модели массообмена в пористых зонах

Массообмен между фазами связан с переходом компонентов G-фазы из G- в S-фазу и наоборот (процессы растворения, испарения, химические реакции). В пакете предусмотрены три модели описания массовых источников:

- 1) MassMode = msNone : межфазная граница *непроницаема*, т.е.

$$M_{Ck} = 0, \quad M_P = \sum_k m_{Ck} = 0 \quad (8.1)$$

- 2) MassMode = msDefault : массообмен описывается источниками стандартного вида,
- 3) MassMode = msChemical : массовые источники для компонентов G-фазы связаны стехиометрическими соотношениями.

#### 8.1.1 Модель массообмена msDefault

В этой модели пользователь в модели взаимодействия должен определить, исходя из постановки прикладной задачи, полный поток массы из S в G-фазу –  $M_P$ .

Массовые источники для отдельных компонентов рассчитываются по «стандартным» для источников соотношениям

$$M_{Ck} = \max(M_p, 0) \cdot C_{k,int} - \max(-M_p, 0) \cdot C_k + C_{0,k} (C_{k,int} - C_k), \quad (8.2)$$

Модель взаимодействия должна определить значение *коэффициента*  $C_{0,k}$  и значение *концентрации* на фазовой границе  $C_{k,int}$ . Напомним, что эти параметры должны быть согласованы для выполнения условий

$$M_p = \sum_k M_{Ck}, \quad \sum_k C_{k,int} = 1$$

### 8.1.2 Модель массообмена msChemical

При использовании этой модели, как и для модели msDefault, вначале определяется полный массовый источник  $M_p$ , а массовые источники для отдельных компонентов рассчитываются по соотношениям

$$M_{Ck} = \xi_{ch,k} M_p, \quad \sum_k \xi_{ch,k} = 1 \quad (8.3)$$

здесь  $\xi_{ch,k}$  - постоянные «стехиометрические» коэффициенты модели. Эти коэффициенты могут быть отрицательными, нулевыми или большими единицы, важно, чтобы их сумма равнялась единице. Обычно соотношения (8.3) используются в случае, когда на фазовой границе идет химическая реакция.

## 8.2 Межфазное трение в пористых зонах

Источники межфазного взаимодействия в уравнениях движения представляют собой силу сопротивления, отнесенную к  $1 \text{ м}^3$  пространства. Для записи этих источников можно использовать различные параметры (коэффициент трения, коэффициент гидравлического сопротивления). В коде Anes используется понятие коэффициента проницаемости

$$S_{U_{gi}} = - < -M_p > U_{gi} - \varphi \frac{\rho_g v_g}{k_i} U_{gi}, \quad i = x, y, z \quad (8.4)$$

где

- $k_i = \frac{2 d_{e,i}^2}{\xi_i Re_i}$  - коэффициент проницаемости вдоль  $i$ -ой оси,
- $d_{e,i} = \frac{4A_i}{\Pi_i}$  - эквивалентный диаметр при течении в  $i$ -том направлении,
- $A_i, \Pi_i$  - площадь поперечного сечения и смоченный периметр микроэлемента пористой среды при течении в  $i$  – направлении,
- $\xi_i = \frac{8\bar{\tau}_{w,i}}{\rho_g \bar{U}_g^2}$  - коэффициент гидравлического сопротивления в  $i$ -том направлении,
- $\bar{U}_g = \sqrt{U_{gx}^2 + U_{gy}^2 + U_{gz}^2}$  - модуль скорости G-фазы,
- $\bar{\tau}_{w,i}$  - среднее трение в  $i$ -ом направлении,
- $Re_i = \frac{\bar{U}_g d_{e,i}}{v_g}$  - число Рейнольдса.

Для определения коэффициентов проницаемости необходима информация о геометрической микроструктуре пористой среды и данные либо по коэффициенту гидравлического сопротивления, либо экспериментальные данные по перепаду давлений  $\Delta P_F$  в слое порис-

той среды ( $L$ -толщина слоя), которые позволяют рассчитать коэффициент гидравлического сопротивления по соотношению

$$\xi = \left( \frac{\Delta P_f}{L} \right) \frac{2d_e}{\rho_g \bar{U}_g^2} \quad (8.5)$$

При использовании экспериментальных соотношений для коэффициента гидравлического сопротивления необходимо обратить внимание на то, какая скорость  $U_g$  используется в определении коэффициента. Обычно используются два типа скорости:

- 1) *средняя скорость в микроканалах* : в этом случае необходимо использовать скорости, рассчитываемые в коде -  $U_{gx}, U_{gy}, U_{gz}$ ,
- 2) *скорость на бесконечности* или *фильтрационная скорость*: в этом случае для скоростей необходимо использовать значения  $\varphi \cdot U_{gx}, \varphi \cdot U_{gy}, \varphi \cdot U_{gz}$

Рассмотрим два типичных примера.

### 8.2.1 Проницаемость системы плоских микроканалов

Пусть пористая среда представляет систему плоских микроканалов в одном из направлений. Типичный пример – пластинчатый теплообменник, микроструктура которого – набор плоских пластин, перпендикулярных оси  $z$  (рисунок 8.1).

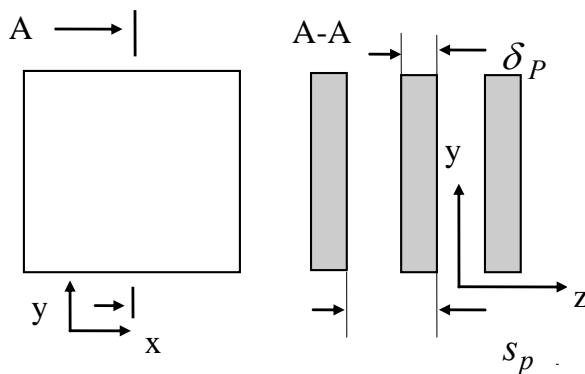


Рисунок 8.1 - Микроструктура пористой среды

Пористость для такой системы легко определить из геометрических соображений

$$\varphi = 1 - \frac{\delta_p}{S_p}$$

где  $\delta_p, S_p$  - толщина пластин и шаг между ними. Эквивалентный диаметр в направлении осей  $x$  и  $y$  равен удвоенной ширине микроканала

$$d_{e,x} = d_{e,y} = 2h = 2(S_p - \delta_p)$$

Для ламинарного стабилизированного течения коэффициент гидравлического сопротивления легко получить из профиля скорости в плоском канале

$$U_y(z) = \frac{3}{2} \bar{U} \left[ 1 - \left( \frac{2z}{h} \right)^2 \right]$$

(начало оси  $z$  - в середине канала, шириной  $h$ ). Отметим, что средняя скорость в микроканале  $\bar{U}$  - это и есть компонента скорости  $U_{gx}$  или  $U_{gy}$ , которые рассчитываются в пакете.

Зная профиль скорости легко рассчитать трение и коэффициент гидравлического сопротивления

$$\bar{\tau}_w = -\rho_g v_g \left( \frac{\partial U_y}{\partial z} \right)_{z=h/2} = \frac{12\rho_g v_g \bar{U}}{(2h)}, \xi = \frac{8\bar{\tau}_w}{\rho_g \bar{U}^2} = \frac{96}{Re}, Re = \frac{\bar{U}(2h)}{v_g}$$

и получить выражение для коэффициентов проницаемости

$$k_x = k_y = \frac{2d_e^2}{96}, k_z = 0$$

### 8.2.2 Проницаемость шаровой засыпки

Для такой микроструктуры среды значение пористости  $\phi$  зависит от способа «упаковки» шаров и обычно определяется из экспериментальных данных.

Для расчета проницаемости (слоя из сферических шаров одинакового диаметра) можно использовать эмпирическое соотношение Эргуна:

$$\frac{\Delta P}{L} = 150 \frac{\mu_g W(1-\phi)^2}{d^2 \phi^3} + \frac{1.75 \rho_g W^2(1-\phi)}{d \phi^3} \quad (8.6)$$

где  $\mu_g$  - коэффициент динамической вязкости,  $d$  - диаметр сферической частицы,  $W$  - скорость потока *на входе в слой*,  $\Delta p$  - перепад давления на слое засыпки толщиной  $L$ . Выражение для коэффициента проницаемости можно получить из определения коэффициента проницаемости

$$\phi \mu_g \frac{U}{k} = -\phi \frac{\Delta P_f}{L}, W = \phi U$$

Подставляя (8.6) в это соотношение, получим

$$k_x = k_y = k_z = d^2 \left( \frac{\phi}{1-\phi} \right)^2 \frac{1}{C_{k1} + C_{k2} Re \frac{\phi}{1-\phi}}, Re = \frac{\bar{U}d}{v_g} \quad (8.7)$$

где значения коэффициентов модели по умолчанию  $C_{k1} = 150, C_{k2} = 1.75$

### 8.3 Межфазный теплообмен в пористых зонах

Межфазный теплообмен в пористых зонах описывается двумя сопряженными источниками в уравнениях для  $T_g$  и  $T_s$  (4.43) и (5.6):

$$\begin{aligned} S_{T_g} &= \sum_k \max(M_{Ck}, 0) \cdot H_{g,k}(T_s) - \sum_k \max(-M_{Ck}, 0) \cdot H_{g,k}(T_g) + \alpha_{gs} \tilde{A}_{gs} (T_s - T_g) \\ &\quad + \sum_k \max(M_{Ck}, 0) \cdot \Delta H_{gs,k}, \\ S_{T_s} &= \sum_k \max(-M_{Ck}, 0) \cdot H_{g,k}(T_s) - \sum_k \max(M_{Ck}, 0) \cdot H_{g,k}(T_g) + \alpha_{gs} \tilde{A}_{gs} (T_g - T_s) \\ &\quad - \sum_k \max(-M_{Ck}, 0) \cdot \Delta H_{gs,k} \end{aligned} \quad (8.8)$$

Для расчета этих источников в модели межфазного взаимодействия должны быть определены три величины:

- 1)  $\alpha_{sg}$  - коэффициент межфазной теплоотдачи, отнесенный к  $1 \text{ м}^2$  межфазной поверхности,
- 2)  $\tilde{A}_{gs}$  - удельная межфазную поверхность,
- 3)  $\Delta H_{gs,k}$  - массив тепловых эффектов реакций.

## 8.4 Межфазное взаимодействие для остальных $\Phi$ -переменных

### 8.4.1 K-eps модель турбулентности

В текущей версии не учитывается влияние пористой среды на распределение турбулентных характеристик.

### 8.4.2 Произвольные $\Phi$ -переменные

Для  $\Phi$ -переменных типа SPACE эффекты межфазного взаимодействия в пористых зонах не учитываются.

Для  $\Phi$ -переменных, связанных с S-фазой (тип STRUCT) межфазный массообмен не учитывается и межфазный источниковый член записывается в виде

$$S_{\Phi} = C_{\Phi} (\Phi_{int} - \Phi) \quad (8.9)$$

Модель взаимодействия должна рассчитать значение коэффициента  $C_{\Phi}$ , отнесенного к  $1 \text{ м}^3$  пространства и значение  $\Phi$ -переменной на фазовой границе  $\Phi_{int}$ .

Для  $\Phi$ -переменных, связанных с G-фазой (тип FLOW) источник рассчитывается с учетом массовой составляющей

$$S_{\Phi} = \langle S_p \rangle \Phi_{int} - \langle -S_p \rangle \Phi + C_{\Phi} (\Phi_{int} - \Phi) \quad (8.10)$$

Модель взаимодействия в этом случае определяет значение коэффициента  $C_{\Phi}$ , отнесенного к  $1 \text{ м}^3$  пространства и значение  $\Phi$ -переменной на фазовой границе  $\Phi_{int}$ .

## 8.5 Организация модели пористого межфазного взаимодействия в Решателе

В пакете с каждой Porous-зоной связаны три элемента модели межфазного взаимодействия:

- 1) переменная MassMode, определяющая алгоритм расчета массообмена (msNone, msDefault, msChemical),
- 2) массив действительных переменных - параметров настройки модели для текущей Porous зоны,
- 3) подпрограмма (или набор подпрограмм) на языке Фортран, содержащая алгоритмы расчета характеристик межфазного взаимодействия.

В состав дистрибутива Anes входит каталог открытой части кода – openfor. В подкаталоге openfor/sourceporous расположены подпрограммы Решателя, в которых реализованы все внутренние модели пористого взаимодействия. Изучив эти тексты, пользователь легко может реализовать свои собственные модели. В данном параграфе рассмотрим структуру этих подпрограмм.

Подпрограмма модели в Решателе вызывается для каждого контрольного объемов Porous-зоны. При вызове ей в качестве параметра передается значение переменной iENTRY, а подпрограмма в зависимости от значения этой переменной должна вернуть значения двух действительных переменных RetCoef и RetVal(3):

iENTRY	RetVal(1:3)	Ret Coef	Назначение
fmPOROUS	(1) = $\phi$	-	подпрограмма модели возвращает значение пористости среды,
fmKXYZ	(i) = $k_i$	-	- возвращает коэффициенты проницаемости вдоль i-оси,
fmALFAFS	(1) = $\alpha_{sg}$	-	- возвращает поверхностный коэффициент теплоотдачи, отнесенный к единице межфазной поверх-

			ности,
fmALFAV	$(1) = \alpha_{sg} \tilde{A}_{gs}$	-	- возвращает объемный коэффициент теплоотдачи, отнесенный к единице объема пространства,
fmMASS	$(1) = S_p$	-	- общий массовый поток, отнесенный к единице объема пространства,
fmCH1 - fmCH8	$(1) = \xi_{ch,k}$		-используется при MassMode = msChemical; возвращает значение стехиометрических коэффициентов,
fmCONC1- fmCONC8	$(1) = C_{0,k}$	$C_{k,int}$	-используется при MassMode = msDefault; возвращается значение коэффициента и концентрации на фазовой границе в соответствии с (4.2),
fmDQ1 - fmDQ8	$(1) = \Delta H_{gs,k}$	-	- возвращает тепловой эффект перехода через фазовую границу (4.8),
fmUSER + idPHI	$(1) = \Phi_{int}$	$C_\Phi$	- возвращает коэффициент и значение на фазовой границе для произвольной $\Phi$ -переменной (4.9) или (4.10) с индексом idPHI,
fmFV	$(1) = \tilde{A}_{gs}$		- возвращает значение удельной поверхности.

В состав текущей версии пакета входит четыре внутренние модели межфазного взаимодействия:

- 1) POR BALL – модель шаровой засыпки,
- 2) PLAIN\_PINS – цилиндрические стержни в треугольной упаковке, расположенные вдоль оси Z,
- 3) FIN\_PINS – цилиндрические оребренные стержни в треугольной упаковке, расположенные вдоль оси Z,
- 4) PERF\_PLATE - пластина с цилиндрическими каналами, расположенными в направлении одной из оси.

Пользователь может создавать свои собственные модели, как на основе внутренних моделей (их исходные тексты расположены в каталоге открытых текстов), так и самостоятельно.

**❗ Описание в проекте прикладной задачи :**

Для подключения пористой модели в патчах типа POROUS указывается имя, которое будет называться «Имя FS-зоны». В секции [Porous Models] для каждой «Имя FS-зоны» должны быть записаны операторы, определяющие модель пористого взаимодействия:

```
InterFace("Имя FS-зоны") = Internal(ИмяВнутреннейМодели) / Virtual(Имя функции)
MassMode("Имя FS-зоны") = msNONE / msDEFAULT / msChemical
```

```
Param("Имя FS-зоны.ИмяПараметра ") = значение
.....
```

Первый оператор определяет имя подпрограммы модели. Если используется внутренняя модель, то ее имя задается оператором Internal(). Второй оператор определяет модель массообмена. Далее указываются операторы Param(), которые задают входные параметры модели. Число этих параметров и их имена зависят от конкретной модели. Для внутренних моделей эти параметры будут описаны ниже.

## 8.6 Модель пористого взаимодействия POR BALL

В этой модели микроструктура пористой среды представляет собой засыпку из сфер одного диаметра.

### 8.6.1 Геометрические характеристики

Пористость  $\phi$  является параметром модели и задается пользователем в параметре с именем "Porous". Удельная поверхность рассчитывается по соотношению

$$\tilde{A}_{gs} = \frac{6(1-\varphi)}{d}$$

### 8.6.2 Проницаемости

Проницаемости в модели рассчитываются по соотношению

$$k_x = k_y = k_z = \left( \frac{d_s \varphi}{1-\varphi} \right)^2 \frac{1}{C_{k1} + C_{k2} \operatorname{Re} \frac{\varphi}{1-\varphi}}, \operatorname{Re} = \frac{\bar{U}d}{v_g},$$

где

$d_s$  - диаметр сферы, задается параметром с именем "DS",

$C_k$  - коэффициенты сопротивления, задаются параметрами с именами "СК1" и "СК2".

Если в описании модели не заданы параметры  $C_k$ , то для них используются значения по умолчанию

$$C_{k1} = 150, C_{k2} = 1.75$$

### 8.6.3 Коэффициент межфазной теплоотдачи

Коэффициент теплоотдачи вычисляется по соотношению

$$\operatorname{Nu} = C_{\operatorname{Nu}} + 1.1 \operatorname{Pr}^{0.33} (\operatorname{Re} \varphi)^{0.6},$$

$$\operatorname{Re} = \frac{U_g d_s}{v_g}, U_g = \sqrt{U_{gx}^2 + U_{gy}^2 + U_{gz}^2}, \alpha_{gs} = \frac{\operatorname{Nu} \lambda_g}{d_s}$$

Параметр  $C_{\operatorname{Nu}}$  задается пользователем в параметре с именем "NUS". Если параметр не задан, то используется значение по умолчанию  $C_{\operatorname{Nu}} = 2$ .

### 8.6.4 Список параметров модели

Полный список параметров модели:

DS = <Диаметр сферы>

POROUS = <Пористость>

NUS = <Число Нуссельта: 2.00>

СК1 = <Первый к-т в формуле Эргуна: 150>

СК2 = <Второй к-т в формуле Эргуна: 1.75>

## 8.7 Модель пористого взаимодействия PLAIN\_PINS

В этой модели микроструктура пористой среды представляет собой цилиндрические стержни в треугольной равносторонней упаковке, расположенные вдоль оси z (рис. 8.2).

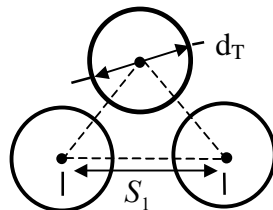


Рисунок 8.2 - Расположение стержней в плоскости x-y

### 8.7.1 Геометрические характеристики

Пористость и удельная поверхность рассчитывается из очевидных геометрических соотношений:

$$\varphi = 1 - \frac{\pi}{2\sqrt{3}} \left( \frac{d_T}{S_1} \right)^2, \quad \tilde{A}_{sg} = \frac{2\pi}{\sqrt{3}} \frac{d_T}{S_1^2}$$

Диаметр стержня  $d_T$  задается параметром с именем "DextTUBE", шаг решетки  $S_1$  – параметром с именем "STEP1".

### 8.7.2 Проницаемости в поперечном сечении (x-y)

Проницаемости в направлении x и y рассчитываются с использованием коэффициента сопротивления  $\xi$ :

$$k_x = k_y = \frac{2d_T^2}{\xi \text{Re}} \left\{ \frac{\sqrt{3}}{2} (x_1 - 1) \frac{1}{\varphi} \right\},$$

в котором число Рейнольдса определяется через скорость в узком сечении:

$$\text{Re} = \frac{U_{\max} d_T}{\nu_g}, \quad U_{\max} = \varphi \sqrt{U_{gx}^2 + U_{gy}^2} \frac{x_1}{x_1 - 1}, \quad x_1 = \frac{S_1}{d_T}$$

Коэффициент гидравлического сопротивления рассчитывается по следующим соотношениям:

$$\begin{aligned} \text{Re} < 600 & : \xi = 1.42(x_1 - 1)^{-0.33} (600)^{-0.15} \\ 600 < \text{Re} < 7000 & : \xi = 1.42(x_1 - 1)^{-0.33} \text{Re}^{-0.15} \\ \text{Re} > 7000 & : \xi = 5.2(x_1 - 1)^{-0.25} \text{Re}^{-0.29} \end{aligned}$$

### 8.7.3 Проницаемость в аксиальном z-направлении

Коэффициент проницаемости в z-направлении также рассчитывается через коэффициент гидравлического сопротивления:

$$k_z = \frac{2d_{ek}^2}{\xi \text{Re}}, \quad d_e = \left( x_1^2 \frac{2\sqrt{3}}{\pi} - 1 \right) d_T, \quad \text{Re} = \frac{|U_{gz}| d_e}{\nu_g}$$

Для вычисления коэффициента гидравлического сопротивления используются соотношения:

$$\begin{aligned} \text{Re} < 2300 & : \xi = \frac{64}{\text{Re}} (0.41 + 1.9 \sqrt[3]{x_1 - 1}) \\ \text{Re} > 2500 & : \xi = \frac{0.210}{\text{Re}^{0.25}} [1 + (x_1 - 1)^{0.32}] \\ 2300 < \text{Re} < 2500 & : \text{линейная интерполяция по числу Рейнольдса.} \end{aligned}$$

### 8.7.4 Коэффициент межфазной теплоотдачи

Коэффициент межфазной теплоотдачи определяется через число Nu:

$$\alpha_{sg} = \text{Nu} \frac{\lambda_g}{d_{ek}}$$

в зависимости от режима обтекания стержней:

- 1)  $\theta_U < 24^\circ$  : «продольное» обтекание стержней,
- 2)  $\theta_U > 30^\circ$  : «поперечное» обтекание стержней,
- 3)  $24^\circ < \theta_U < 30^\circ$  : линейная интерполяция по углу.



Здесь  $\theta_U$  – угол между вектором скорости и осью стержней (осью z)

$$\theta_U = \arctan\left(\frac{U_{xy}}{U_{gz}}\right), U_{xy} = \sqrt{U_{gx}^2 + U_{gy}^2}, U_g = \sqrt{U_{gx}^2 + U_{gy}^2 + U_{gz}^2}$$

При *продольном обтекании* для расчета Nu используется число Рейнольдса, определенное по эквивалентному диаметру  $d_e$  и модулю скорости  $U_g$

$$Re = \frac{U_g d_e}{\nu_g}$$

при  $Re < 2500$

$$\begin{aligned} Nu &= 30x_1 - 28.5, \quad x_1 < 1.2 \\ &= 12.5x_1 - 7.5, \quad 1.2 < x_1 < 1.4 \\ &= 8.5x_1 - 1.9, \quad 1.4 < x_1 < 2.8 \\ &= 10x_1 - 6.1, \quad 2.8 < x_1 < 4.0 \end{aligned}$$

при  $Re > 3000$

$$Nu = \left[ 0,0165 + 0,02 \left( 1 - \frac{0,91}{x_1^2} \right) x_1^{0,15} \right] Re^{0,8} Pr^{0,4} .$$

В области чисел  $Re = 2500 \dots 3000$  используется линейная интерполяция по числу Re.

При *поперечном обтекании* в качестве эквивалентного диаметра в числе Re используется диаметр стержня  $d_T$ , а в качестве скорости – скорость в узком сечении канала

$$U_{max} = U_g \frac{x_1}{x_1 - 1}$$

Число Nu вычисляется по следующим соотношениям:

$$\begin{aligned} Nu &= C(\theta_U) \tilde{Nu}, \\ C(\theta_U) &= 0.396473 + 0.0142268\theta_U - 7.06788 \cdot 10^{-5} \theta_U^2, \\ \tilde{Nu} &= A Re^m Pr^n \end{aligned}$$

где параметры A, m и n зависят от числа Re

Re	A	m	n
< 40	1.04	0.4	0.36
40 – 1000	0.71	0.5	0.36
$10^3 - 2 \cdot 10^5$	0.35	0.6	0.36
$> 2 \cdot 10^5$	0.031	0.8	0.4

### 8.7.5 Список параметров модели

Полный список параметров модели:

DextTUBE = <Внешний Диаметр трубы трубы>  
STEP1 = <Шаг треугольной упаковки>

### 8.8 Модель пористого взаимодействия FINE\_PINS

Эта модель аналогична модели PLAIN\_PIN, но на трубах с внешней стороны расположены ребра, т.е. микроструктура пористой среды представляет собой оребренные цилиндрические стержни в треугольной упаковке, расположенные вдоль оси z (рис. 8.2, 8.3).

В этой модели используется две межфазные поверхности между G и S фазами. Для расчета пористости межфазная поверхность проходит по внешней поверхности оребренной трубы, включая поверхность ребер (граница B - физическая межфазная поверхность).

Для расчета удельной поверхности, пористости и коэффициента теплоотдачи используется условная межфазная поверхность, представляющую собой внешнюю поверхность гладкой трубы (граница А).

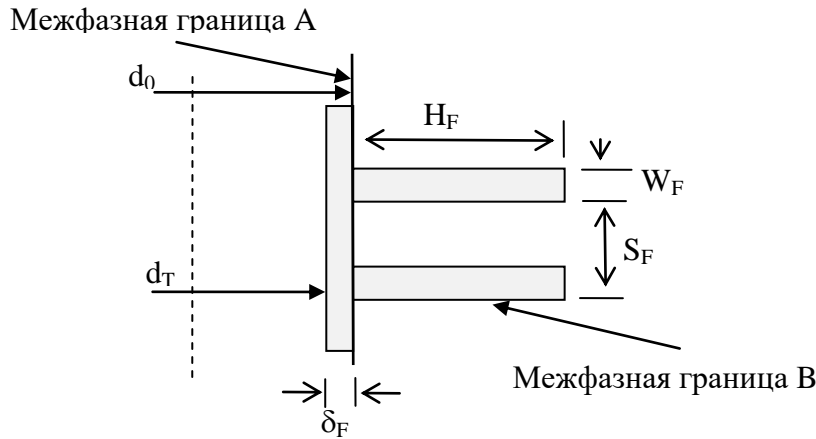


Рисунок 8.3 – Параметры оребрения

Использование в модели двух межфазных поверхностей и введение толщины оребренной цилиндрической поверхности  $\delta_F$  связано с тем, что данная модель используется в двухуровневых пористых моделях. В этих моделях для расчета температуры  $T_s$  S-фазы вместо уравнения теплопроводности (5.1) используются внешние к коду уравнения (например, одномерные уравнения для течения теплоносителя внутри трубок).

### 8.8.1 Геометрические характеристики

Пористость и удельная поверхность рассчитывается из очевидных геометрических соотношений:

$$\phi = 1 - \frac{\pi}{2\sqrt{3}} \left( \frac{d_0}{S_1} \right)^2 \left[ 1 + \frac{W_F}{S_F} \left( \left( \frac{d_F}{d_0} \right)^2 - 1 \right) \right],$$

$$\tilde{A}_{sg} = \frac{2\pi d_T}{\sqrt{3} S_1^2}, \quad d_0 = d_T + 2\delta_F, \quad d_F = d_0 + 2H_F$$

Диаметр стержня  $d_T$  задается параметром с именем "DextTUBE", шаг решетки  $S_1$  – "Step1", толщина цилиндрического оребрения  $\delta_F$  – "DeltF", толщина ребра  $W_F$  – "W\_F", высота ребра  $H_F$  – "H\_F".

### 8.8.2 Проницаемости по осям координат

Проницаемости в направлении  $x$  и  $y$  рассчитываются через коэффициент гидравлического сопротивления по общему соотношению:

$$k = \frac{2d_{base}^2}{\xi(Re_m) \cdot Re_m} \left( \frac{S_2}{d_{base}} \right) \frac{1}{C_m}, \quad Re_m = \frac{U_m d_{base}}{\nu}, \quad S_2 = S_1 \frac{\sqrt{3}}{2}$$

Здесь  $d_{base}$  – характерный диаметр трубок,  $U_m$  – скорость потока в «узком» сечении,  $C_m$  – коэффициент связи скорости  $U_m$  со скоростью кода Anes.

В модели Fine\_Pins в качестве характерного диаметра и «узкого» сечения используются параметры гладкой трубы с диаметром  $d_0$ . В этом случае

$$U_m = C_m U_0, \quad C_{m0} = \varphi \frac{x_1}{x_1 - 1}, \quad U_0 = \sqrt{U_{gx}^2 + U_{gy}^2 + U_{gz}^2},$$

и коэффициенты сопротивления равны:

$$k_x = k_y = \frac{2d_0^2}{\xi Re_m} \left\{ \frac{\sqrt{3}}{2} (x_1 - 1) \frac{1}{\varphi} \right\}.$$

Коэффициент гидравлического сопротивления рассчитывается по следующим соотношениям:

для  $Re_m < 10$ :

$$\xi = 13.1 x_1^{-0.55} x_2^{-0.5} \frac{(1 - x_{sf})^{1.8}}{(1 - x_{hf})^{1.4}} (10)^{-0.25}$$

для  $Re_m < 10^5$ :

$$\xi = 13.1 x_1^{-0.55} x_2^{-0.5} \frac{(1 - x_{sf})^{1.8}}{(1 - x_{hf})^{1.4}} (Re_m)^{-0.25}$$

для  $Re_m > 10^5$ :

$$\xi = 0.74 x_1^{-0.55} x_2^{-0.5} \frac{(1 - x_{sf})^{1.8}}{(1 - x_{hf})^{1.4}}$$

где

$$x_1 = \frac{S_1}{d_0}, \quad x_2 = x_1 \frac{\sqrt{3}}{2}, \quad x_{sf} = \frac{S_f}{d_0}, \quad x_{hf} = \frac{H_f}{d_0}, \quad x_{wf} = \frac{W_f}{d_0}$$

Поскольку при течении в оребренном пучке из-за наличия ребер течение в аксиальном направлении затруднено, то коэффициент проницаемости  $k_z$  считается малой величиной, равной

$$k_z = \frac{k_x}{100}$$

### 8.8.3 Коэффициент межфазной теплоотдачи

Коэффициент межфазной теплоотдачи определяется по соотношениям

$$\alpha_{gs} = C_A \alpha_m, \quad \alpha_m = E_{eff} Nu \frac{\lambda_g}{d_0}, \quad C_A = \frac{A_F}{A_0}$$

где

$\alpha_{gs}$  - коэффициент теплоотдачи, отнесенный к межфазной поверхности А (внешняя поверхность неоребренной трубы),

$\alpha_m$  - коэффициент теплоотдачи, отнесенной к межфазной поверхности В (внешняя поверхность трубы с ребрами).

$C_A$  - коэффициент пересчета коэффициента теплоотдачи с поверхности А на поверхность В.

$A_F, A_0$  - площади внешней поверхности оребренной трубы и неоребренной трубы:

$$A_0 = \pi d_0^2, \quad A_F = A_{FT} + A_{RT},$$

$$A_{FT} = \pi \left[ \frac{1}{2} (d_F^2 - d_0^2) + d_F W_F \right], \quad A_{RT} = \pi d_0 (S_f - W_f)$$

При расчете числа Нуссельта используется число Рейнольдса, рассчитанное по скорости в «узком» сечении *оребреной* трубы и диаметру гладкой трубы  $d_0$ :

$$Re_m = \frac{U_m d_0}{\nu}, \quad U_m = C_{mf} U_0, \quad C_{mf} = \frac{\varphi}{1 - \frac{1}{x_1} - 2 \frac{H_f W_f}{S_f S_1}}, \quad x_1 = \frac{S_1}{d_0}$$

Число Нуссельта рассчитывается по соотношениям:  
для  $Re_m < 1$ :

$$Nu = C_{geo} \cdot 0.192 \cdot Pr^{0.36}, \quad C_{geo} = \left( \frac{x_1}{x_2} \right)^{0.2} x_{sf}^{0.18} x_{hf}^{-0.14}$$

для  $1 < Re_m < 2 \cdot 10^4$ :

$$Nu = C_{geo} \cdot 0.192 \cdot Re_m^{0.65} Pr^{0.36}$$

для  $2 \cdot 10^4 < Re_m < 2 \cdot 10^5$ :

$$Nu = C_{geo} \cdot 0.0507 \cdot Re_m^{0.8} Pr^{0.4}$$

для  $Re_m > 2 \cdot 10^5$ :

$$Nu = C_{geo} \cdot 0.0081 \cdot Re_m^{0.95} Pr^{0.4}$$

Коэффициент эффективности ребра  $E_{eff}$  определяется по формулам:

$$H_e = H_F \left( 1 + \frac{W_F}{2H_F} \right) \left[ 1 + 0.35 \cdot \ln \left( \frac{d_F}{d_0} \right) \right],$$

$$\beta_e = H_e \sqrt{\frac{2\alpha_m}{\lambda_F H_e}}, \quad \Omega_f = \frac{\tanh(\beta_e)}{\beta_e} [1 - 0.058\beta_e],$$

$$E_{eff} = \left[ \frac{A_{FT}}{A_0} \Omega_f + \frac{A_{RT}}{A_0} \right]$$

Здесь  $\lambda_F$  - коэффициент теплопроводности материала ребра, который задается параметром с именем "CondF".

### 8.8.4 Список параметров модели

Полный список параметров модели:

DextTUBE = <Внешний Диаметр несущей трубы>  
STEP1 = <Шаг треугольной упаковки>  
H\_F = <ВЫСОТА РЕБРА>  
W\_F = <ТОЛЩИНА РЕБРА>  
S\_F = <ШАГ ОРЕБРЕНИЯ>  
DELTF = <Толщина стенки ребра>  
CONDf = <Теплопроводность ребра>

## 8.9 Модель пористого взаимодействия PERF\_PLATE

В этой модели микроструктура пористой среды представляет собой S-фазу с внутренними цилиндрическими каналами для G-фазы в треугольной упаковке. В каком-то смысле эта модель является «инверсией» модели PLAIN\_PINS.

### 8.9.1 Геометрические характеристики

Пористость и удельная поверхность рассчитывается из очевидных геометрических соотношений:

$$\varphi = \frac{\pi}{2\sqrt{3}} \left( \frac{d_T}{S_1} \right)^2, \quad \tilde{A}_{sg} = \frac{2\pi}{\sqrt{3}} \frac{d_T}{S_1^2}$$

Диаметр канала  $d_T$  задается параметром "DTUBE", шаг решетки S1 – "STEP1", направление расположения каналов задается целым параметром «iDIR» (для iDIR = 1 каналы расположены вдоль оси x, 2 – вдоль оси y, 3 – вдоль оси z).

### 8.9.2 Проницаемости по осям координат

Проницаемость в направлении оси iDIR рассчитывается с использованием коэффициента гидравлического сопротивления

$$k_{iDIR} = \frac{2d_T^2}{\xi Re}, \quad Re = \frac{U_{g,iDir} d_T}{v_g}$$

Коэффициент гидравлического сопротивления определяется по соотношениям

$$Re < 2200: \quad Re \xi = 64,$$

$$2400 < Re < 5 \cdot 10^4: \quad Re \xi = 0.316 Re^{0.75},$$

2200 < Re < 2400 : интерполяция,

$$Re > 5 \cdot 10^4: \quad Re \xi = 0.184 Re^{0.8}$$

Проницаемости в направлениях других осей принимаются равными:

$$k = k_{iDIR}/100$$

### 8.9.3 Коэффициент межфазной теплоотдачи

Коэффициент теплоотдачи рассчитывается по значению числа Нуссельта:

$$\alpha_{sg} = Nu \frac{\lambda_g}{d_T}.$$

Для расчета числа Нуссельта используются соотношения:

для  $Re < 2400$ :

$$Nu = Nu_L$$

для  $Re > 10^4$ :

$$Nu = \frac{\sqrt{\frac{\xi}{8}} Re Pr}{1.07 + 12.7 \sqrt{\frac{\xi}{8}} (Pr^{0.666} - 1)}, \quad \xi = \frac{1}{(1.82 \lg(Re) - 1.64)^2}$$

для  $2400 < Re < 10^4$ : интерполяция по числу Re.

Постоянное число Нуссельта для ламинарного режима течения  $Nu_L$  задается параметром модели с именем "LNU". Если этот параметр не задан, то по умолчанию используется значение  $Nu_L = 4$ .

### 8.9.4 Список параметров модели

Полный список параметров модели приведет:

DTUBE = <Диаметр трубы>

STEP1 = <Шаг треугольной упаковки>

LNU = <Ламинарное число Нуссельта>

IDIR = <Направление труб 1 .. 3>

## 9. Модели турбулентности

В текущей версии кода Anes для моделирования турбулентного переноса массы, импульса и энергии используются модели, основанные на концепции кинематической турбулентной вязкости -  $\nu_t$ .

В этом случае эффективный коэффициент переноса в уравнении сохранения для  $\Phi$ -переменной, связанной с G-фазой, рассчитывается по соотношению (2.5)

$$\Gamma_{\Phi} = \rho_g \left( \frac{\nu_g}{\sigma_{\Phi}} + \frac{\nu_t}{\sigma_{\Phi,t}} \right) \quad (9.1)$$

Здесь  $\sigma_{\Phi,t}$  - турбулентное «число Прандтля», смысл которого зависит от конкретной  $\Phi$ -переменной (для уравнения энергии - это турбулентное число Прандтля, для уравнения для концентрации - турбулентное число Шмидта, для уравнений движения он равен единице). В коде эти переменные рассматриваются как V-переменные, алгоритм расчета которых определяется пользователем. По умолчанию  $\sigma_{\Phi,t} = 1$  для всех  $\Phi$ -переменных G-фазы.

В текущей версии пакета для расчета коэффициента турбулентной вязкости  $\nu_t$  можно использовать шесть моделей:

- 1) алгебраическая модель,
- 2) двухпараметрическая k- $\epsilon$  модель с пристенными функциями,
- 3) двухслойная двухпараметрическая k- $\epsilon$  модель для малых турбулентных чисел Рейнольдса,
- 4) двухпараметрическая k- $\epsilon$  модель Лаундера-Шарма для малых турбулентных чисел Рейнольдса,
- 5) двухпараметрическая k- $\omega$  модель Вилкокса-Ментера,
- 6) SST модификация двухпараметрической k- $\omega$  модели.

В алгебраической модели коэффициент турбулентной вязкости рассматривается как V-переменная, алгоритм расчета которой определяется пользователем. В текущей версии кода реализована одна встроенная алгебраическая модель турбулентности – LVEL модель турбулентности, предложенная Б. Сполдинггом. Подробности модели изложены ниже.

При использовании k- $\epsilon$  моделей коэффициент турбулентной вязкости рассчитывается на основе полей двух  $\Phi$ -переменных:

- 1) k - удельной кинетической энергии турбулентных пульсаций,
- 2)  $\epsilon$  - скорости диссипации турбулентной энергии.

При использовании k- $\omega$  и SST моделей вместо скорости диссипации используется ее аналог – переменная  $\omega$ , которая представляет собой удельную скорость диссипации турбулентной энергии.

### Описание в проекте прикладной задачи :

Выбор модели турбулентности определяется макрофункцией секции [Macro Sub] ModelTur(TM)

где

TM = trLAMINAR / trVARIANT / trKEWall / trKE2Layer / trKOM / trSST / trKELOW

Эта макрофункция сама осуществляет все настройки модели, в частности, определяет операторы Used() и Solve() для  $\Phi$ -переменных kTUR и epsTUR или omTUR.

При использовании алгебраической модели турбулентности значение кинематического коэффициента турбулентной вязкости задается оператором секции [Turbulence]

NuTVAR = V-переменная

Для задания турбулентного «числа Прандтля» для  $\Phi$ -переменной с именем "Ф-имя" используется оператор секции [Turbulence]

SigmaTUR("Ф-имя") = V-переменная

### 9.1 Турбулентные соотношения для пристенной области

Для понимания материала, приведенного ниже, напомним некоторые соотношения для пристенной турбулентности (см. например, [10]).

Рассмотрим *одномерный* сдвиговый слой вблизи твердой стенки, на которой выполняются условия прилипания ( $U_g = 0$ ). В этой модельной задаче имеется только одна касательная компонента скорости  $U_{gx} = U(y)$ , а градиентом давления вдоль стенки  $dp/dx$  пока будем пренебрегать (рисунок 9.1). Для краткости ниже будет опускаться нижний индекс «g» у всех переменных.

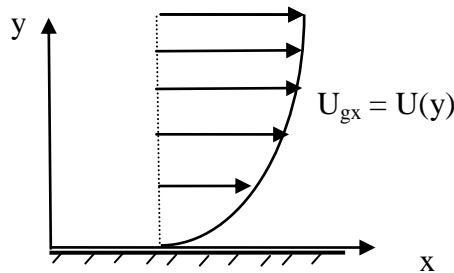


Рисунок 9.1 – Одномерный сдвиговый слой

Распределение скорости в одномерном сдвиговом слое описывается уравнением движения:

$$\frac{d}{dy} \left[ \rho(v + v_t) \frac{dU}{dy} \right] = 0, \text{ или } \rho(v + v_t) \frac{dU}{dy} = \tau_{xy} = \tau_w = \text{const}, \tag{9.2}$$

$$\tau_w = \rho v \left( \frac{dU}{dy} \right)_{y=0}$$

где  $\tau_w$  – касательное напряжение на стенке.

Одномерной слой можно условно разбить на три зоны: область ламинарного *вязкого подслоя*, где турбулентной вязкостью  $v_t$  можно пренебречь ( $v_t \ll v$ ), область *развитой турбулентности* ( $v \ll v_t$ ), и буферная переходная зона, в которой молекулярная и турбулентная вязкости одного порядка.

Для расчета турбулентной вязкости  $v_t$  в области развитой турбулентности обычно используется формула Прандтля, которая подтверждается огромным числом экспериментальных исследований:

$$v_t = l_v^2 \left| \frac{dU}{dy} \right|, l_v = \kappa y \tag{9.3}$$

Здесь  $l_v$  - длина пути смешения Прандтля,  $\kappa$  - эмпирическая постоянная Кармана (типичное значение  $\kappa = 0.41$ ).

#### 9.1.1 Логарифмический профиль скорости

Если ввести безразмерные переменные

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad \text{- динамический масштаб скорости,}$$

$$U^+ = \frac{U_x}{u_\tau}, y^+ = \frac{u_\tau y}{\nu} \quad \text{- безразмерные скорость и расстояние до стенки,}$$

то систему уравнений (9.2), (9.3) можно записать в следующем безразмерном виде: для зоны вязкого подслоя

$$\frac{dU^+}{dy^+} = 1$$

и для зоны развитой турбулентности

$$\frac{dU^+}{dy^+} = \frac{1}{\kappa y^+}$$

Интегрируя эти уравнения и сшивая их, получим так называемый «стандартный» профиль скорости для сдвигового слоя

$$U^+ = \begin{cases} y^+, & y^+ < y_v^+ \\ \frac{1}{\kappa} \ln(Ey^+), & y^+ > y_v^+ \end{cases}, \quad (9.4)$$

В этом соотношении  $E$  – эмпирическая постоянная, которая определяется координатой «сшивки» решений (9.4)  $y_v^+$ :

$$E = \frac{\exp(\kappa y_v^+)}{y_v^+} \quad (9.5)$$

Типичные экспериментальные значения  $E$  лежат в пределах 8.6 .. 9.8 (значения  $y_v^+ = 11.1 .. 11.5$ ).

Профиль скорости (9.4) является приближенным, поскольку в нем зона развитой турбулентности (область логарифмического профиля скорости) непосредственно сшивается с ламинарным слоем. В реальности между ними существует важная буферная зона, в которой турбулентная и молекулярная вязкость соизмеримы. Для более точного описания универсального профиля скорости предложено много аппроксимаций, наиболее оригинальной является аппроксимация Б. Сполдинга, которая использует «обратную» зависимость  $y^+ = f(U^+)$ :

$$y^+ = U^+ + \frac{1}{E} \left\{ e^{\kappa U^+} - 1 - (\kappa U^+) - \frac{1}{2} (\kappa U^+)^2 - \frac{1}{6} (\kappa U^+)^3 - \frac{1}{24} (\kappa U^+)^4 \right\} \quad (9.6)$$

Соотношения (9.5) и (9.6) справедливы для течений, в которых градиентом давления в уравнении (9.2) можно пренебречь по сравнению со значением касательного напряжения на стенке  $\tau_w$ . Для большинства канальных течений *вблизи стенки* это условие выполняется. Для течений с отрывом потока, где касательное напряжение на стенке близко к нулю, необходимо учитывать градиент давления в уравнении движения

$$\frac{d}{dy} \left[ \rho(v + v_t) \frac{dU}{dy} \right] = \frac{dP}{dx}, \text{ или } \rho(v + v_t) \frac{dU}{dy} = \tau_w + \frac{dP}{dx} y,$$

Для таких решений в литературе также имеется большое число универсальных профилей скорости, в которых наряду с динамической скоростью  $u_\tau$  используется еще одна характерная скорость  $u_p$ :

$$u_p = \left( \frac{v}{\rho} \left| \frac{dP}{dx} \right| \right)^{1/3}$$

Турбулентная составляющая профиля скорости (9.4) справедлива для гладких стенок. Если стенка имеет шероховатость, то постоянная  $E$  начинает зависеть от эффективной высоты выступов шероховатости –  $h_r$ . Для учета влияния шероховатости на логарифмический профиль Б. Сполдингом были предложены следующие соотношения:



$$E = \begin{cases} E_0, h_r^+ < 3.7, \\ \frac{1}{\sqrt{a \left(\frac{h_r^+}{b}\right)^2 + (1-a) \frac{1}{E_0^2}}}, 3.7 < h_r^+ < 100, \\ \frac{b}{h_r^+}, h_r^+ > 100 \end{cases}$$

$$b = 29.7, \quad a = (1 + 2x^3 - 3x^2) \tag{9.7}$$

$$x = 0.02248 \frac{100 - h_r^+}{(h_r^+)^{0.564}}$$

где

$E_0$  - постоянная логарифмического профиля для гладкой стенки,

$h_r^+ = \frac{u_\tau h_r}{\nu}$  - безразмерная координата  $y^+$ , рассчитанная по высоте шероховатости.

### 9.1.2 Логарифмический профиль скорости и двухпараметрические модели турбулентности

В двухпараметрических моделях турбулентности  $k$ - $\varepsilon$  и  $k$ - $\omega$  (подробности этих моделей описаны ниже) для описания турбулентного переноса используется поля двух характеристик: удельной кинетической энергии турбулентных пульсаций  $k$  и скорости диссипации турбулентной энергии  $\varepsilon$  (или ее разновидности - удельной диссипации  $\omega$ ).

В зоне развитой турбулентности одномерного сдвигового слоя уравнение для  $k$  можно записать в виде:

$$v_t \left( \frac{dU}{dy} \right)^2 - \varepsilon = 0, \quad v_t = C_\mu \frac{k^2}{\varepsilon} \tag{9.8}$$

Если для диссипации  $\varepsilon$  использовать выражение

$$\varepsilon = C_D \frac{k^{3/2}}{l_v}, \quad l_v = \kappa y, \quad C_D = C_\mu^{3/4}, \tag{9.9}$$

то для профиля скорости можно получить то же решение, что и для модели Прандтля (9.3).

Это позволяет получить следующие важные соотношения, связывающие модель длины смешения Прандтля и  $k$ - $\varepsilon$  ( $k$ - $\omega$ ) модель:

- турбулентная энергия в зоне логарифмического профиля скорости постоянна и связана с динамической скоростью соотношением:

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \tag{9.10}$$

- диссипацию можно рассчитать по соотношению

$$\varepsilon = \frac{u_\tau^3}{l_v}, \quad l_v = \kappa y \tag{9.11}$$

- а коэффициент турбулентной вязкости по соотношению

$$v_t = C_\mu^{1/4} \sqrt{k} l_v \quad (9.12)$$

### 9.1.3 Модифицированный логарифмический профиль скорости

Соотношение (9.10) позволяет получить другую форму для стандартного профиля скорости (9.4). Если ввести следующие безразмерные переменные

$$U^* = \frac{U_x \sqrt{k}}{u_\tau^2}, \quad y^* = \frac{y^+}{C_\mu^{1/4}} = R_k = \frac{\sqrt{k} y}{v}, \quad (9.13)$$

$$\kappa^* = \kappa C_\mu^{1/4}, \quad E^* = C_\mu^{1/4} E, \quad y_v^* = \frac{y_v^+}{C_\mu^{1/4}}$$

то профиль (9.4) можно записать в «эквивалентном» виде

$$U^* = \begin{cases} y^*, & y^* < y_v^* \\ \frac{1}{\kappa^*} \ln(E^* y^*), & y^* > y_v^* \end{cases}, \quad (9.14)$$

Как было показано Лаундером [10] при использовании пристенных функций профиль (9.14) дает лучшее совпадение расчетов с экспериментом, чем (9.4).

В моделях  $k-\omega$  в качестве модифицированного масштаба скорости используется не-много другая величина  $u^*$ :

$$u^* = C_\mu^{1/4} \sqrt{k}$$

Используя эту скорость можно ввести новую безразмерную координату

$$y^* = \frac{u^* y}{v},$$

которая отличается от определения Лаундера только численным коэффициентом. Однако у этого определения есть одно преимущество - в области логарифмического слоя

$$y^+ = y^*$$

Эта особенность используется ниже при построении пристенных функций в  $k-\omega$  моделях.

### 9.1.4 Логарифмический профиль для температуры

Для одномерного сдвигового слоя уравнение энергии можно записать в виде, аналогичном (9.2)

$$\frac{d}{dy} \left[ \rho c_p \left( \frac{v}{Pr} + \frac{v_t}{Pr_t} \right) \frac{dT}{dy} \right] = 0, \quad \text{или} \quad \rho c_p \left( \frac{v}{Pr} + \frac{v_t}{Pr_t} \right) \frac{dT}{dy} = q_w = \text{const}, \quad (9.15)$$

$$q_w = \frac{\rho c_p v}{Pr} \left( \frac{dT}{dy} \right)_{y=0}$$

Если ввести безразмерные переменные (где  $T_w$  – температура стенки)

$$\Theta^+ = \frac{T_w - T}{T_\tau}, \quad T_\tau = \frac{q_w}{\rho c_p u_\tau} \quad (9.16)$$

то систему (9.15) можно записать в виде: для зоны вязкого подслоя

$$\frac{d\Theta^+}{dy^+} = Pr,$$

и для зоны развитой турбулентности

$$\frac{d\Theta^+}{dy^+} = \frac{Pr_t}{\kappa y^+}.$$

Решение этой системы:

$$\Theta^+ = \begin{cases} Pr y^+, & y^+ < y_T^+ \\ \frac{Pr_t}{\kappa} \ln(E_T y^+), & y^+ > y_T^+ \end{cases}, \quad (9.17)$$

где  $E_T$  – новая эмпирическая постоянная, значение которой зависит от чисел Прандтля  $Pr$  и  $Pr_t$ . Безразмерная координата сшивки решений для температуры определяется из решения уравнения

$$E_T = \frac{\exp\left(\frac{Pr}{Pr_t} \kappa y_T^+\right)}{y_T^+}$$

Обычно турбулентная составляющая профиля (9.17) записывается в другой форме, более удобной для реализации в пристенных функциях

$$\Theta^+ = Pr_t \left[ U^+ + Pi \right], \quad Pi = Pi\left(\frac{Pr}{Pr_t}\right), \quad (9.18)$$

которая связывает универсальный профиль температуры  $\Theta^+$  с универсальным профилем скорости  $U^+$ . В (9.18)  $Pi$  – это так называемая «Пи-функция», которая учитывает отличие динамического и температурного логарифмических слоев. Эта функция может быть получена из обработки экспериментальных данных. В настоящее время чаще всего используются два соотношения:

Б. Солдинга:

$$Pi(x) = 9 \frac{(x-1)}{x^{0.25}}, \quad x = \frac{Pr}{Pr_t} \quad (9.19)$$

и Джаятиллака:

$$Pi(x) = 9.24 \left( x^{3/4} - 1 \right) \left[ 1 + 0.28 \exp(-0.007x) \right] \quad (9.20)$$

Профиль (9.18) можно записать через модифицированную координату  $y^*$  Лаундера, если ввести новые безразмерные переменные

$$\Theta^* = \frac{\rho c_p (T_w - T) \sqrt{k}}{q_w}, \quad Pi^* = \frac{Pr}{C_\mu^{1/4}}, \quad (9.21)$$

$$\Theta^* = Pr_t \left[ U^* + Pi^* \right]$$

При наличии шероховатости значение  $Pi$ -функции меняется. Для учета влияния шероховатости на  $Pi$ -функцию обычно используется соотношение, предложенное Джаятиллаке:

$$Pi = Pi_0 \left( \frac{E}{E_0} \right)^{0.6} + 3.15 Pr^{0.695} \left( \frac{1}{E} - \frac{1}{E_0} \right)^{0.359}$$

где  $E, E_0$  – постоянные логарифмического профиля для шероховатой и гладкой стенок,  $Pi_0$  – Пи-функция для гладкой стенки.

**ⓘ Описание в проекте прикладной задачи :**

В описанных выше соотношениях используются две «универсальные» константы – постоянная Кармана  $\kappa$  и константа логарифмического закона  $E$ . Они используются во всех моделях турбулентности. Пользователь может изменить их значения по умолчанию с помощью операторов секции [Turbulence]:

StKappa = 0.41  
StE = 9.79

При использовании универсальных профилей скорости и температуры в модели турбулентности с пристенными функциями можно для каждого патча с флагом `ISWALL` можно задать свою эффективную высоту шероховатости  $h_r$  с помощью оператора секции `[Special Data]`:  $R(\text{"Имя Патча.HR"}) = \text{значение}$

## 9.2 Модель пристенных функций

При моделировании турбулентных течений наибольшие градиенты  $\Phi$ -переменных наблюдается в пристеночной области, где вблизи стенки всегда имеется ламинарный вязкий подслой. Одним из способов моделирования таких течений на достаточно грубых сетках является применение описанных выше законов стенки для расчета параметров в пристенных КО.

### 9.2.1 Пристенные функции для скорости

Рассмотрим задание граничных условий для скорости на твердой границе с прилипанием (рисунок 9.2).

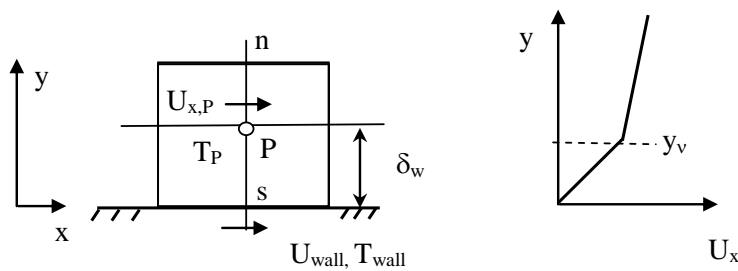


Рисунок 9.2 - Схема пристенного КО для скорости

Естественными граничными условиями для скоростей является равенство на поверхности нормальной компоненты скорости нулю и равенство касательной компоненты скорости движения поверхности (для неподвижной стенки – нулю). В коде такие граничные условия реализуются путем задания плотности потока импульса на граничной грани "s" (с параметрами ГУ:  $C_{ux} = \text{BC\_FIXVAL}$ ,  $V_{ux} = U_{\text{wall}}$ , см. пункт 3.5.3). Для касательной составляющей скорости плотность потока импульса на граничной грани "s" в этом случае будет рассчитан по формуле

$$J_{ux,s} = \tau_w = -\frac{v_{\text{eff},P}}{\delta_n} U_{\text{slip}}, U_{\text{slip}} = U_{x,P} - U_{\text{wall}}, \quad v_{\text{eff},P} = \left[ \rho_g (v_g + v_T) \right]_P \quad (9.22)$$

Если узел P находится вне вязкого подслоя, то вычисление трения по уравнению (9.22) приводит к значительным погрешностям, так как пристенный КО характеризуется большими градиентами скорости и турбулентной вязкости.

С другой стороны выражение для касательного напряжения на стенке можно записать в следующем обобщенном виде

$$\tau_w = -\rho_{g,P} |U_{\text{slip}}| c_{fw} U_{\text{slip}} = -\frac{\rho_g v_{\text{eff},P}}{\delta_w} U_{\text{slip}},$$

$$c_{fw} = \frac{\tau_w}{\rho_{g,P} U_{\text{slip}}^2}, \quad \text{Re}_P = \frac{|U_{\text{slip}}| \delta_w}{v_{g,P}} \quad (9.23)$$

$$v_{\text{eff},P} = v_g (c_{fw} \text{Re}_P)$$

здесь  $c_{fw} = c_w(\text{Re}_P)$ - «эмпирический» коэффициент трения. Именно этот коэффициент и есть так называемая *пристенная функция* для скорости. Заметим, что в (9.23) трение на стенке рассчитывается по «стандартному» для граничного потока соотношению (9.22),

изменилось только выражение для расчета эффективного коэффициента кинематической вязкости.

Для определения коэффициента трения  $c_{fw}$  можно использовать следующую идею. Пусть узел «Р» расположен в области логарифмического закона стенки (9.4). Соотношение (9.4) легко записать через коэффициент трения в виде

$$\frac{1}{\sqrt{c_{fw}}} = \frac{1}{\kappa} \ln(E Re_P \sqrt{c_{fw}}) \quad (9.24)$$

Это нелинейное уравнение позволяет по заданной касательной скорости в точке Р рассчитать коэффициент трения и «точно» задать касательное трение на стенке (это уравнение легко решается методом простой итерации).

Закон стенки (9.14), записанный через  $R_k$ , также легко переписать в виде соотношения для определения коэффициента трения

$$c_{fw} = \frac{R_k}{Re_P} \frac{\kappa^*}{\ln(E^* R_k)}, \quad R_k = \frac{\sqrt{k_P} \delta_w}{\nu} \quad (9.25)$$

Важно отметить, что соотношения (9.23) – (9.25) справедливы, если центр пристенного КО Р находится в области развитой турбулентности. В качестве критерия можно использовать значение безразмерной длины  $y^+$  или  $R_k$  в точке Р. Это естественно накладывает условие на размер пристенного КО (он должен быть достаточно большим, чтобы  $y^+ > 20-40$ ).

### 9.2.2 Пристенные функции для уравнения энергии

Для уравнения энергии ( $\Phi$ -переменная  $T_g$ ) можно использовать аналогичные соотношения

$$J_{T_{g,s}} \equiv q_w = -\rho_{g,P} c_{pg,P} |U_{slip}| St_w (T_{g,P} - T_w) = -\frac{\lambda_{eff,P}}{\delta_w} (T_{g,P} - T_w),$$

$$St_w = \frac{q_w}{\rho_{g,P} c_{pg,P} |U_{slip}| (T_{g,P} - T_w)}, \quad (9.26)$$

$$\lambda_{eff,P} = \frac{\rho_g \nu_g c_{pg,P}}{Pr_g} (St_w Re_P Pr).$$

Здесь  $St_w$  - «пристенное» число Стентона, которое рассматривается как пристенная функция для температуры. Число Стентона легко связать с коэффициентом трения, используя профили (9.17) и (9.21)

$$St_w = \frac{c_{fw}}{Pr_t (1 + Pi \sqrt{c_{fw}})}, \quad Pi = f(\tilde{x}), \quad \tilde{x} = \frac{Pr}{Pr_t},$$

$$St_w = \frac{c_{fw}}{Pr_t \left[ 1 + Pi \cdot c_{fw} \frac{U_{slip}}{C_\mu^{0.25} \sqrt{k_P}} \right]} \quad (9.27)$$

### 9.2.3 Пристенные функции для произвольных $\Phi$ -переменных

В коде Anes для моделирования пристенных функций для произвольных  $\Phi$ -переменных, связанных с G-фазой, используется подход, аналогичный подходу для уравнения энергии:

$$J_{\Phi,s} \equiv -\rho_{g,P} |U_{slip}| St_{\Phi,w} (\Phi_P - \Phi_w) = -\frac{\Gamma_{\Phi,P}^{(eff)}}{\delta_w} (\Phi_P - \Phi_w),$$

$$St_{\Phi,w} = \frac{J_{\Phi,s}}{\rho_{g,P} |U_{slip}| (\Phi_P - \Phi_w)}, \quad (9.28)$$

$$\Gamma_{\Phi,P}^{(eff)} = \frac{\rho_g v_g}{\sigma_\Phi} (St_{\Phi,w} Re_P \sigma_\Phi).$$

### 9.3 Алгебраические модели турбулентности

В текущей версии Anes реализована одна алгебраическая модель - LVEL модель Б. Сполдинга. В этой модели (и других моделях) требуется знание расстояния от любой точки РО до ближайшей твердой стенки  $\delta_w$ . Для хранения этого поля в коде используется поле  $L_{dis}$ , а для его расчета используется специальная  $\Phi$ -переменная с именем  $W_{dis}$ .

#### 9.3.1 Переменная $L_{dis}$ и $\Phi$ -переменная $W_{dis}$

В коде Anes реализованы два алгоритма расчета поля  $L_{dis}$ : «прямой» алгоритм и расчет на основе поля  $W_{dis}$ .

При использовании «прямого» алгоритма поле  $L_{dis}$  рассчитывается следующим образом:

- 1) для каждого КО просматриваются все грани поверхностных патчей с флагом IsWALL;
- 2) в качестве  $L_{dis}$  для данного КО выбирается минимальное расстояние от его центра до центра грани.

Другой алгоритм вычисления LDIS основан на решении уравнения для переменной  $W_{dis}$ . Этот алгоритм был предложен Б. Сполдингом и он позволяет найти *приближенное расстояние* до ближайшей твердой стенки из решения уравнения

$$\text{div}(\nabla W_{dis}) = -1 \quad (9.29)$$

с граничными условиями  $W_{dis} = 0$  на твердых стенках. Значение переменной  $L_{dis}$  определяется из соотношения

$$L_{dis} = \delta_w = \sqrt{|\nabla W_{dis}|^2 + 2W_{dis}} - |\nabla W_{dis}| \quad (9.30)$$

Заметим, что решение (9.29) для сложной геометрии дает приемлемые результаты только вблизи стенки! Вдали от стенки расстояние будет неточным. Например, для трубы

$$\text{расстояние } L_{dis} = \frac{r_0}{\sqrt{2}} \approx 0.7r_0.$$

Поэтому рекомендуется использовать прямой метод расчета LDIS. В текущей версии он работает и в параллельном режиме!

При наличии в задаче STRUCT-зон в них LDIS рассчитывается со знаком минус!

При решении больших задач (порядка 5-30 млн. ячеек) прямое вычисление LDIS может занимать большое время! Для решения этой проблемы в текущей версии реализованы следующие возможности:

1. LDIS сохраняется в отдельном файле с расширением \*.aldis. При рестарте или старте с варианта LDIS не рассчитывается, а считывается из файла.
2. При расчете «канальных» задач, в которых стенка создается 2D объектами можно использовать «быстрый» режим расчета, когда LDIS определяется только гранями в том же сечении по iz, что и ячейка. В этом случае вычисление LDIS ускоряется на несколько порядков.

**ⓘ Описание в проекте прикладной задачи :**

Расчет LDIS активируется в Решателе если:

- используется двухслойная модель турбулентности trKE2Layer или SST k- $\omega$  модель;
- решается уравнение для  $\Phi$ -переменной WDIS;
- пользователь указал логический оператор

L("USE\_LDIS") = .TRUE.

в секции [Special Data].

Поле LDIS рассчитывается до выполнения основного SWEEP- итерационного цикла решения как стационарной так и нестационарной задач. Выбор алгоритма расчета Ldis зависит от использования  $\Phi$ -переменной WDIS. Если WDIS используется, то LDIS рассчитывается по рассчитанному полю WDIS. В противном случае используется «прямой» алгоритм расчета.

Для включения «быстрого» режима расчета LDIS для канальных течений необходимо в секцию [Special Data] добавить оператор:

L("LDIS.CalcByZ") = .True.

### 9.3.2 LVEL модель турбулентности

При проведении первоначальных отладочных расчетов часто используется простейшая модель турбулентности, в которой турбулентная вязкость принимается постоянной

$$\nu_t = \text{const}$$

Очевидно, что данная модель будет давать нереальные значения характеристик, особенно вблизи твердых стенок. Более удобной для оценок является LVEL-модель Б. Сполдинга, которая «расширяет» пристенные сдвиговые соотношения на всю расчетную область. Суть этой модели заключается в следующем.

Рассмотрим пристенную область, в которой можно записать соотношение (9.2) немного в другом виде

$$\rho v_{\text{eff}} \frac{dU}{dy} = \tau = \tau_w = \text{const}$$

Будем считать, что  $U$  – это модуль скорости в *любой* точке РО а  $y$  – расстояние от данной точки РО до *ближайшей* твердой стенки. Приводя к безразмерному виду, это соотношение можно представить в виде

$$v^+ = \frac{v + \nu_t}{\nu} = \frac{dy^+}{dU^+}, \quad U^+ = \frac{U}{u_\tau}, \quad y^+ = \frac{u_\tau y}{\nu}, \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (9.31)$$

Будем считать, что во *всей области* решения справедливо соотношение (8.6)

$$y^+ = U^+ + \frac{1}{E} \left\{ e^{\kappa U^+} - 1 - (\kappa U^+) - \frac{1}{2}(\kappa U^+)^2 - \frac{1}{6}(\kappa U^+)^3 - \frac{1}{24}(\kappa U^+)^4 \right\} \quad (9.32)$$

Дифференцируя это соотношение можно связать безразмерную эффективную вязкость с безразмерной скоростью

$$v^+ = 1 + \frac{\kappa}{E} \left\{ e^{\kappa U^+} - 1 - (\kappa U^+) - \frac{1}{2}(\kappa U^+)^2 - \frac{1}{6}(\kappa U^+)^3 \right\}$$

Таким образом, для расчета безразмерной вязкости необходимо знать значение безразмерной скорости в любой точке РО. В модели LVEL предлагается следующая идея:

- 1) мы можем рассчитать «пристенное» число Рейнольдса

$$R = \frac{U \delta_w}{\nu}$$

где  $U$  – модуль скорости в данной точки,  $\delta_w$  – расстояние до ближайшей твердой стенки, которая определяется полем  $L_{\text{dis}}$ -переменной;

- 2) число  $R$  можно записать в виде

$$R = U^+ y^+$$

Эти два условия позволяют записать неявное уравнение для вычисления  $U^+$  (используя (9.32)):

$$R = U^+ \left[ U^+ + \frac{1}{E} \left\{ e^{\kappa U^+} - 1 - (\kappa U^+) - \frac{1}{2} (\kappa U^+)^2 - \frac{1}{6} (\kappa U^+)^3 - \frac{1}{24} (\kappa U^+)^4 \right\} \right]$$

Отметим главный недостаток LVEL-модели – с ростом удаления от стенки турбулентная вязкость будет расти всегда. Фактически эта модель распространяет зону логарифмического слоя на всю РО. Поэтому эта модель дает приемлемые результаты только для течений типа канальных.

Описание в проекте прикладной задачи :

Для выбора LVEL-модели необходимо в секции [Macro Sub] выбрать алгебраическую модель

ModelTur(trVARIANT)

а в секции [Turbulence] задать имя модели в операторе

NuTVAR = Internal(trLVEL)

## 9.4 Двухпараметрические модели турбулентности

В текущей версии кода реализованы несколько версий двух двухпараметрических моделей: k-ε и k-ω модели. Версии отличаются способом описания турбулентных характеристик в пристеночной области.

Пристеночная область турбулентных потоков характеризуется большими градиентами зависимых переменных и недостаточной изученностью закономерностей турбулентного переноса массы, импульса и энергии. Для описания полей зависимых переменных вблизи твердой поверхности необходимо использовать очень мелкую сетку КО, неприемлемую для многих трехмерных задач. Однако основная трудность заключается в отсутствии моделей турбулентности, надежно описывающих характеристики турбулентности в непосредственной близости стенки (вязком подслое и буферной области).

Все разновидности k-ε и k-ω моделей можно разделить на три группы:

- 1) Модели с *пристенными функциями* - в этом случае для расчета k и ε используется «стандартная» система их уравнений для всей расчетной области за исключением пристенных КО. В пристенных КО уравнения модели турбулентности и уравнения для Φ-переменных модифицируются с использованием пристенных функций, описанных выше.
- 2) Модели для *малых турбулентных чисел* Рейнольдса – в этом случае «стандартная» система модели вблизи стенки модифицируется; что позволяет расширить ее применение вплоть до твердой стенки.
- 3) Универсальные модели, которые объединяют указанные выше модели.

В текущей версии пакета реализованы по одному представителю всех трех моделей. Пользователь может создавать свои собственные версии этих моделей. В качестве образца можно использовать исходные коды моделей Решателя, которые располагаются в подкаталоге openfor/source/turbulence каталога Anes.

## 9.5 Модель k-ε с пристенными функциями

В качестве базовой модели в коде реализована стандартная k-ε модель турбулентности [10] для больших турбулентных чисел Рейнольдса, использующая дифференциальные уравнения переноса для двух Φ-переменных k и ε:

$$\begin{aligned} \frac{\partial(\varphi \rho_g k)}{\partial \tau} + \operatorname{div} \left( \varphi \rho_g \mathbf{U}_g k - \varphi \rho_g \left( \mathbf{v}_g + \frac{\mathbf{v}_t}{\sigma_k} \right) \nabla k \right) &= S_k^{(I)} + \varphi S_k, \\ \frac{\partial(\varphi \rho_g \varepsilon)}{\partial \tau} + \operatorname{div} \left( \varphi \rho_g \mathbf{U}_g \varepsilon - \varphi \rho_g \left( \mathbf{v}_g + \frac{\mathbf{v}_t}{\sigma_\varepsilon} \right) \nabla \varepsilon \right) &= S_\varepsilon^{(I)} + \varphi S_\varepsilon. \end{aligned} \quad (9.33)$$



Источниковые члены пользователя  $S_k$  и  $S_\varepsilon$  задаются общими соотношениями, описанными в главе 2. Внутренние источниковые члены, автоматически рассчитываемые в пакете, можно записать в виде

$$S_k^{(1)} = \varphi \rho_g (P_k + G_b - \varepsilon),$$

$$S_\varepsilon^{(1)} = \varphi \rho_g \left[ C_{\varepsilon 1} (P_k + C_{\varepsilon 3} G_b) - C_{\varepsilon 2} \varepsilon \right] \frac{\varepsilon}{k},$$
(9.34)

где

- $P_k = \nu_t G$  - скорость порождения турбулентной энергии,
- $G$  - диссипативная функция (4.21) и (4.22),
- $G_b$  - порождение турбулентной энергии, связанное с силами плавучести (см. пункт 9.6).

Коэффициент кинематической турбулентной вязкости рассчитывается по соотношению

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}$$
(9.35)

Стандартная турбулентная модель содержит шесть постоянных

$$C_\mu = 0.09, C_{\varepsilon 1} = 1.44, C_{\varepsilon 2} = 1.92, C_{\varepsilon 3} = 0.0, \sigma_k = 1, \sigma_\varepsilon = 1.3.$$

**① Описание в проекте прикладной задачи :**

Основные постоянные k-е модели пользователь может изменить с помощью операторов секции [Turbulence]:

CtMU	= 0.09	!	постоянная $C_\mu$
CtEPS1	= 1.44	!	$C_{\varepsilon 1}$
CtEPS2	= 1.92	!	$C_{\varepsilon 2}$
CtEPS3	= 1.0	!	$C_{\varepsilon 3}$
CtSigmaK	= 1.0	!	$\sigma_k$
CtSigmaEps	= 1.3	!	$\sigma_\varepsilon$

### 9.5.1 Граничные условия на твердой стенке

Еще раз напомним, что в коде к этой категории границ относятся все участки границы РО, для которых пользователем установлен флаг IsWALL. Для межфазной границы между G- и S-фазами (описываемой патчами "FS\_") этот флаг устанавливается автоматически.

Для всех Ф-переменных G-фазы (за исключением k и ε) на данных границах с использованием пристенных функций задаются модифицированные граничные условия (9.23), (9.26) и (9.28).

Для турбулентной энергии и диссипации используется аналогичный подход - в приграничных КО система уравнений (9.33), (9.34) модифицируется с использованием пристенных моделей. Типичными представителями таких моделей являются равновесная модель Сполдинга и неравновесная модель Лаундера-Сполдинга.

В *равновесной* модели значения турбулентной энергии и диссипации в пристенном КО просто фиксируются с помощью соотношений (9.10) и (9.11):

$$k_p = \frac{u_\tau^2}{\sqrt{C_\mu}}, \quad \varepsilon_p = \frac{u_\tau^3}{\kappa \delta_w},$$
(9.36)

а значение скорости  $u_\tau$  рассчитывается с помощью коэффициента трения  $c_{fw}$  (9.24).

В *неравновесной* модели значение диссипации в пристенном КО также фиксируется

$$\varepsilon_p = \frac{C_\mu k_p^{3/2}}{\kappa^* \delta_w}.$$
(9.37)

Для турбулентной энергии на стенках задаются «адиабатические» граничные условия, а источниковые члены в уравнении баланса турбулентной энергии вычисляются с учетом

допущения о нулевом вкладе вязкого подслоя в порождение турбулентной энергии и постоянстве касательного напряжения в КО [10]

$$S_k^{(1)} = \overline{\varphi \rho_g (P_k - \varepsilon)} = \rho \left\{ \frac{u_\tau^2}{\sqrt{k_p}} \frac{1}{2\kappa^* \delta_w} \ln \left( \frac{2y_p^*}{y_v^*} \right) - \frac{\sqrt{k_p}}{\delta_w} \left[ \frac{1}{y_v^*} + \frac{C_\mu}{2\kappa^*} \ln \left( \frac{2y_p^*}{y_v^*} \right) \right] k_p \right\}, \quad (9.38)$$

где верхняя черта над сомножителями обозначает среднюю величину в пристенном КО.

Заметим, что эти соотношения в пристенных КО формируются Решателем автоматически. Пользователю нужно только указать флаг IsWALL на нужных границах.

В текущей версии кода Anes реализованы обе модели пристенных функций:

- 1) trwDBS - равновесная модель,
- 2) trwLAUND - неравновесная модель.

#### **Модель trwDBS**

В этой модели в качестве пристенных функций для  $k$  и  $\varepsilon$  используются равновесные соотношения (9.36). В качестве универсального профиля скорости используется профили скорости  $U^+(y^+)$  (9.4) или (9.6). Для расчета коэффициента трения используется соотношение (9.24).

По умолчанию используется «стандартный» логарифмический профиль (9.4). Для использования профиля Сполдинга (9.6) необходимо в секции [Special Data] задать оператор

I("trwDBS.UPLUS") = 2

#### **Модель trwLAUND**

В этой модели в качестве пристенных функций для  $k$  и  $\varepsilon$  используются неравновесные соотношения (9.38) и (9.39). Для расчета коэффициента трения и чисел Стентона используются пристенные соотношения, связанные с переменной  $R_k$  (9.25) и (9.27).

Информация о параметрах модели турбулентности с пристенными функциями выводится в листинг Решателя в виде следующей таблицы:

```
------(K-EPS (Wall Fun))-----
1.SweepBeforeKE.....= 0
2.CtMasK.....= 0.00000
3.CtMasL.....= 2.00000
4.CtMU.....= 9.000000E-02
5.CtEPS1.....= 1.44000
6.CtEPS2.....= 1.92000
7.CtEPS3.....= 1.00000
8.CtSigmaK.....= 1.00000
9.CtSigmaEps.....= 1.30000
10.LinearKEPS.....= keEQU
11.CtMAXLENTUR.....= 0.748623 m
12.CtKappa.....= 0.410000
13.CtE.....= 8.60000
14.CtYPLUS_LAM.....= 11.1238
15.IsWALLTUR_EXT.....= F
16.WallFun model.....= Равновесная модель Сполдинга
17.UPLUS model.....= Универсальный профиль Сполдинга
```

Заметим еще раз, что модель пристенных функций справедлива, если центр пристенного КО находится в области развитой турбулентности. В качестве критерия можно использовать значение безразмерной длины  $y^+$  или  $R_k$  в точке P. Решатель рассчитывает минимальные и максимальные значения этих величин для всех твердых стенок и выводит эту информацию в свой листинг в следующем виде:

===== Turbulence Parameters =====

# Patch Name	RetN(min)	RetN(max)	YPLUS(min)	YPLUS(max)
1 BOT	0.31386E-03	1.1452	0.23854	5.5793
2 UP	0.32065E-03	14.875	0.23566	6.2850

Для правильности использования модели пристенных функций величина  $y^+$  для *большинства* пристенных КО должна быть больше 20.

В состав подсистемы *tuFORM* [2] включена функция-переменная с именем *YPLUS*, которая сохраняет в пристенных КО значение  $y^+$  (в остальные КО помещается ноль).

**❗ Описание в проекте прикладной задачи :**

Для включения модели с пристенными функциями необходимо в секции [Macro Sub] указать оператор *ModelTur(trKEWall)*

и настроить параметры секции [Turbulence]. Для выбора модели пристенных функций используются операторы:

*TKEFun = Internal(trwDBS)*

*TKEFun = Internal(trwLAUND)*

По умолчанию используется модель Сполдинга-Лаундера для расчета пристенных функций.

### 9.5.2 Входные граничные условия

При использовании «типичных» ГУ на входной границе ( $C_k = C_\varepsilon = BC\_MASSONLY$ ) необходимо задать значения турбулентной энергии  $k_0$  и диссипации  $\varepsilon_0$  в качестве значения ГУ (см. раздел 3.16). Нулевые значения  $k$  и  $\varepsilon$  на входных границах не являются лучшим вариантом для сходимости итерационного процесса. Рекомендуется на входе задавать следующие постоянные значения

$$k_0 = \frac{3}{2}(C_{k0}U_0)^2, \quad \varepsilon_0 = C_\mu^{3/4} \frac{k_0^{3/2}}{0.1L_0}, \quad (9.39)$$

где

$C_{k0}$  – характерная интенсивность турбулентных пульсаций ( $C_{k0} = 0.05 \dots 0.01$ ),

$U_0$  – характерная скорость на входе,

$L_0$  – характерный размер входной границы.

В ряде задач на входной границе можно использовать «слабые» условия:

$C_k = C_\varepsilon = BC\_MASSONLY$ ;  $V_k = V_\varepsilon = BC\_SAME$

В коде *Anes* это означает, что *на входной границе не нужно вообще задавать ГУ!* Какие ГУ на входе использовать - можно определить только численными экспериментами.

### 9.5.3 Расчет коэффициента турбулентной вязкости

Для расчета коэффициента турбулентной вязкости в стандартной  $k$ - $\varepsilon$  модели используется соотношение (9.35). Это соотношение может приводить к ошибкам итерационного процесса в областях, где  $k$  и  $\varepsilon$  близки к нулю. В Решателе *Anes* используется модификация соотношения (8.35) следующего вида

$$\nu_t = C_\mu^{1/4} \sqrt{k} l_v, \quad l_v = \min\left(C_\mu^{3/4} \frac{k^{3/2}}{\varepsilon}, L_{\max,v}\right) \quad (9.40)$$

Здесь  $L_{\max,v}$  - «ограничитель» масштаба турбулентных пульсаций. Этот параметр рассчитывается в коде по соотношению

$$L_{\max,v} = C_{\text{mas}L} L_{\text{dom}}$$

где  $L_{\text{dom}}$  – минимальный размер базовой расчетной области,  $C_{\text{mas}L}$  – коэффициент, задаваемый пользователем в операторе секции [Turbulence]:

$CtMasL$  = значение

Если оператор не указан, то используется значение по умолчанию = 2.

Значение  $L_{\max,v}$  выводится в листинг Решателя в таблице настройки модели турбулентности (см. ниже в пункте 5.7.5) под именем `StMAXLENTUR`.

Если пользователь ввел *внутреннюю* переменную пользователя с именем "iLENTUR", то эта переменная будет содержать значение поля линейного масштаба турбулентных пульсаций  $l_v$ .

### 9.6 Порождение турбулентности силами плавучести $G_b$

Член порождения турбулентной энергии  $G_b$  в уравнениях  $k$ - $\varepsilon$  моделей в общем случае можно записать в виде:

$$G_b = g_i \overline{u_i' \rho'} = -\beta_T \sum_{i=1}^3 g_i \overline{u_i' T'}$$

где верхний штрих означает пульсацию поля, а символ подчеркивания - осреднение по времени.

В коде ANES реализованы две модели для расчета этого члена: SGD (Simple Gradient Diffusion) и GGD (Generalised Gradient Diffusion).

В модели SGD для аппроксимации турбулентного потока тепла используется та же аппроксимация, что и в уравнении энергии:

$$\overline{u_i' T'} = -\frac{v_t}{Pr_t} \frac{\partial T}{\partial x_i}, \quad G_b = \beta_T \frac{v_t}{Pr_t} \sum_{i=1}^3 g_i \frac{\partial T}{\partial x_i} \tag{9.41}$$

Во второй модели GGD (Generalised Gradient Diffusion), предложенной Лаундером, для расчета корреляции скорости и температуры используется соотношение:

$$\begin{aligned} \overline{u_i' T'} &= -C_\theta \frac{k}{\varepsilon} \sum_{j=1}^3 \overline{u_i' u_j'} \frac{\partial T}{\partial x_j}, \quad \overline{u_i' u_j'} = -\left\{ 2v_t S_{ij} - \frac{2}{3} \delta_{ij} k \right\}, \quad S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \\ G_b &= C_\theta \frac{k}{\varepsilon} \beta_T \sum_{i=1}^3 g_i \sum_{j=1}^3 \left\{ \frac{2}{3} \delta_{ij} k - 2v_t S_{ij} \right\} \frac{\partial T}{\partial x_j} = \\ &= \beta_T \left( \frac{2}{3} C_\theta \right) \frac{k^2}{\varepsilon} \sum_{i=1}^3 g_i \frac{\partial T}{\partial x_i} - \beta_T C_\theta \frac{2v_t k}{\varepsilon} \sum_{i=1}^3 \sum_{j=1}^3 S_{ij} g_i \frac{\partial T}{\partial x_j} \end{aligned} \tag{9.42}$$

Здесь  $C_\theta$  - эмпирическая постоянная. Сравнение (9.41) и (9.42) дает для этой постоянной значение

$$C_\theta = \frac{3C_\mu}{2Pr_t} \approx 0.15$$

В различных работах используются значения  $C_\theta = 0.15 \dots 0.3$ . Для «подключения» члена  $G_b$  в уравнении для диссипации используется постоянная  $C_{\varepsilon 3}$ . В коде используются два алгоритма добавления  $G_b$ . Если значение постоянной  $C_{\varepsilon 3} > 0$ , то член добавляется в уравнение для  $\varepsilon$  в обычном виде:

$$S_\varepsilon^{(1)} = \phi \rho_g \left[ C_{\varepsilon 1} (P_k + C_{\varepsilon 3} G_b) - C_{\varepsilon 2} \varepsilon \right] \frac{\varepsilon}{k},$$

Если  $C_{\varepsilon 3} < 0$ , то добавляется только «чистое порождение» :

$$S_\varepsilon^{(1)} = \phi \rho_g \left[ C_{\varepsilon 1} (P_k + |C_{\varepsilon 3}| \max(0, G_b)) - C_{\varepsilon 2} \varepsilon \right] \frac{\varepsilon}{k},$$

Значение  $C_{\varepsilon 3}$  по умолчанию равно единице.

**❶ Описание в проекте прикладной задачи :**

Для активации модели расчета члена  $G_b$  используется оператор секции [Special Data]:

`C("ModelKE_BUOY") = "None" / "SimpleGrad" / "GeneralGrad"`

Значение постоянной  $C_\theta$  и  $C_{\varepsilon_3}$  определяются оператором секции [Turbulence]:

$C_tTETA = 0.3$   
 $C_tEPS3 = 1$

### 9.7 Двухслойная $k$ - $\varepsilon$ модель

Двухслойная модель (Two-layer Model) относится к категории моделей для малых турбулентных чисел Рейнольдса. В этой модели расчетная область разбивается на две зоны (рис. 5.2):

- 1) пристенную зону, расположенную вблизи твердых границ РО,
- 2) зону ядра потока.

КО «P» попадает в пристенную зону, если его центр удовлетворяет условиям:

$$R_k = \frac{\sqrt{k_p} \delta_w}{v_g} \leq R_{k,cr} \text{ и } \delta_w < L_{cr} \quad (9.43)$$

Здесь  $R_{k,cr}$  – граничное значение турбулентного числа Рейнольдса  $R_k$  (значение по умолчанию  $R_{k,cr} = 200$ ),  $L_{cr}$  – предельная ширина пристенной зоны (по умолчанию равно бесконечности). Второе условие (9.43) защищает от появления пристенных зон вдали от стенок в областях с маленькими  $R_k$ .

Уравнение для турбулентной энергии  $k$  (первое уравнение (9.33)) решается во всей РО с граничными условиями  $k=0$  на твердых границах. Уравнение для скорости диссипации (второе уравнение (9.33)) решается только в ядре потока. Соотношение для расчета турбулентной вязкости (9.35) также используется только в ядре потока.

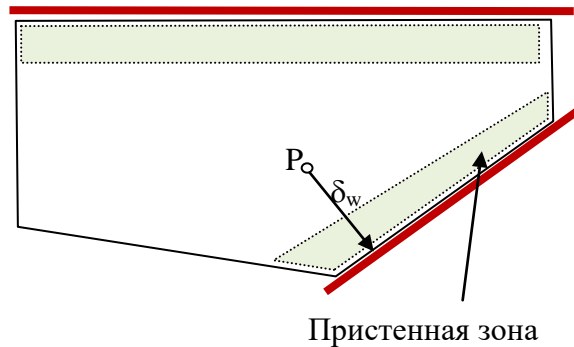


Рисунок 9.3 – Зоны двухслойной модели

Для расчета вязкости в *пристенной* зоне используется соотношение, аналогичное (9.12)

$$v_t = C_\mu^{1/4} f_\mu (R_k) \sqrt{k} l_v, \quad (9.44)$$

а для расчета скорости диссипации – соотношение

$$\varepsilon = C_D f_\varepsilon (R_k) \frac{k^{3/2}}{l_v}, \quad l_v = \kappa y, \quad C_D = C_\mu^{3/4} \quad (9.45)$$

Различные версии двухслойной модели отличаются собственно видом «демпфирующих» функций  $f_\mu$  и  $f_\varepsilon$ . Ниже эти функции описаны для моделей, реализованных в Anes.

#### 9.7.1 Модель tr2LREY

Эта модель реализует разновидность двухслойной модели, предложенной Норрисом и Рейнольдсом (L. Norris & W. Reynolds):

$$f_{\mu} = 1 - \exp\left(-\frac{R_k}{A_{\mu}}\right), \quad f_{\varepsilon} = \left(1 + \frac{A_{\varepsilon}}{R_k}\right),$$

$$A_{\mu} = 50.5; \quad A_{\varepsilon} = 5.3$$

### 9.7.2 Модель tr2LWOLF

Эта модель реализует разновидность двухслойной модели, предложенной Вольфштейна (M. Wolfshtein):

$$f_{\mu} = 1 - \exp\left(-\frac{R_k}{A_{\mu}}\right), \quad f_{\varepsilon} = \frac{1}{1 - \exp\left(-\frac{R_k}{A_{\varepsilon}}\right)},$$

$$A_{\mu} = 70.0; \quad A_{\varepsilon} = 5.1$$

### 9.7.3 Модель tr2LVIA

Эта модель реализует разновидность двухслойной модели одного из разработчиков кода Anes. Модель близка к модели, предложенной Хассидом и Порэ (S. Hassid, M. Poreh):

$$f_{\mu} = 1 - \exp\left(-\frac{R_k}{A_{\mu}}\right), \quad f_{\varepsilon} = f_{\mu} + \frac{A_{\varepsilon}}{R_k},$$

$$A_{\mu} = 83.4; \quad A_{\varepsilon} = 11.0$$

### 9.7.4 Модель tr2LEXP

Это экспериментальная версия модели, основанная на диссертации R. Diaz [11]. В этой версии

$$f_{\mu} = 1 - \exp\left(-A_{\mu 1} \sqrt{R_k} - A_{\mu 2} R_k^{3/2}\right), \quad A_{\mu 1} = 0.028; \quad A_{\mu 2} = 0.0012;$$

$$\varepsilon = A_{\varepsilon 1} \frac{k^{3/2}}{\delta_n} f_{\varepsilon 1} + 2\nu_g \frac{k}{\delta_n^2} f_{\varepsilon 2},$$

$$f_{\varepsilon 1} = 1 - \exp(-A_{\varepsilon 2} R_k), \quad f_{\varepsilon 2} = \exp(-A_{\varepsilon 3} R_k)$$

$$A_{\varepsilon 2} = 0.02; \quad A_{\varepsilon 3} = 0.15$$

### 9.7.5 Особенности использования двухслойной модели

Для использования двухслойной модели необходимо в секции [Macro Sub] задать оператор

```
ModelTur(trKE2Layer)
```

По умолчанию в качестве версии модели используется модель tr2LVIA. Для смены версии необходимо использовать операторы секции [Turbulence]

```
TKEFun = Internal(tr2LREY)
TKEFun = Internal(tr2LWOLF)
TKEFun = Internal(tr2LEXP)
TKEFun = Internal(tr2LVIA)
```

При использовании двухслойной модели необходимо поле  $L_{dis}$ . Ее активация описана в пункте 8.3.2. Имя этого поля можно использовать в формулах `myFORM`, поэтому оно лег-

ко может быть выведено в файл результатов. Поле  $R_k$  сохраняется в поле с именем "RETN", которое также можно использовать в формулах tuFORM.

Для анализа распределения пристенных зон и ядра потока удобно использовать переменную подсистемы tuFORM с именем "L2ZONE". Эта функция возвращает 1, если КО располагается внутри пристенной зоны, и 0 – если КО находится в ядре потока.

Если пользователь ввел *внутреннюю* переменную пользователя с именем "iEPS\_2L", то эта переменная будет содержать значение диссипации, рассчитанной по соотношению (8.43) для всей PO (а не только в пристенных зонах).

Коэффициенты версий двухслойной модели указаны выше. При создании своих собственных версий или при изменении параметров существующих версий пользователь может изменить коэффициенты  $A_\mu$  и  $A_\varepsilon$ . Для этого можно использовать операторы секции [Turbulence]

CtANU = значение

CtAEPS = значение

Критические значения  $R_{k,cr}$  и  $L_{cr}$  задаются операторами

CtRETNCri = значение

CtLDISCri = значение

Информация о настройке любой модели турбулентности выводится в листинг расчета в виде:

```
------(K-EPS (Two Layers))-----
1.SweepBeforeKE.....= 100
2.CtMasK.....= 0.00000
3.CtMasL.....= 2.00000
4.CtMU.....= 9.000000E-02
5.CtEPS1.....= 1.44000
6.CtEPS2.....= 1.92000
7.CtEPS3.....= 0.00000
8.CtSigmaK.....= 1.00000
9.CtSigmaEps.....= 1.30000
10.LinearKEPS.....= keEQU
11.CtMAXLENTUR.....= 0.748623 m
12.CtKappa.....= 0.410000
13.CtE.....= 8.60000
14.CtYPLUS_LAM.....= 11.1238
15.CtRETNCri.....= 200.000
16.CtLDISCri.....= 1.000000E+20
17.CtANU.....= 83.4000
18.CtAEPS.....= 11.0000
19.IsWALLTUR_EXT.....= T
20.TwoLayer model.....= VIA & Hassid
```

Поскольку в двухслойной модели не используются пристенные функции, то в них невозможно учесть влияние шероховатости поверхности (см. пункт 9.1).

## 9.8 Модель турбулентности $k-\omega$

В текущей версии Anes реализованы две версии двухпараметрической модели турбулентности, в которой для описания диссипации турбулентной энергии используется поле  $\omega$ : стандартная  $k-\omega$  модель и ее модификация – SST модель Ментера. В данном разделе описана  $k-\omega$  модель, SST модель описана в следующем разделе.

В коде используется низкорейнольсовская  $k-\omega$  версия модели с «универсальными» пристенными функциями, предложенными Ментером:

$$\begin{aligned} \frac{\partial(\rho k)}{\partial \tau} + \operatorname{div} \left( \rho \mathbf{U} k - \rho \left( \mathbf{v} + \frac{\mathbf{v}_t}{\sigma_k} \right) \nabla k \right) &= \rho \left[ v_t G - C_\mu f_k \omega k \right], \\ \frac{\partial(\rho \omega)}{\partial \tau} + \operatorname{div} \left( \rho \mathbf{U} \omega - \rho \left( \mathbf{v} + \frac{\mathbf{v}_t}{\sigma_\omega} \right) \nabla \omega \right) &= \rho \left[ C_{\omega 1} f_\omega G - C_{\omega 2} \omega^2 \right], \end{aligned} \quad (9.46)$$

$$v_t = f_\mu \frac{k}{\omega}, \quad R_T = \frac{k}{\omega \nu}$$

Здесь  $G$  – диссипативная функция (3.21), (3.22) :

$$G = \frac{\partial U_i}{\partial x_j} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \equiv \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) = S^2, \quad (9.47)$$

$$S = \sqrt{2 S_{ij} S_{ij}}, \quad S_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

В модели использует следующий набор постоянных:

$$C_\mu = 0.09, \quad C_{\omega 1} = \frac{5}{9} = 0.556, \quad C_{\omega 2} = \frac{3}{40} = 0.075, \quad \sigma_k = \sigma_\omega = 2 \quad (9.48)$$

Учет низкорейнольдсовских эффектов производится с помощью трех функций, зависящих от турбулентного числа Рейнольдса  $R_T$ :

$$f_\mu = \frac{0.024 + \frac{R_T}{6}}{1 + \frac{R_T}{6}}, \quad f_\omega = \frac{0.111 + \frac{R_T}{2.95}}{1 + \frac{R_T}{2.95}}, \quad f_k = \frac{0.2667 + \left( \frac{R_T}{8} \right)^4}{1 + \left( \frac{R_T}{8} \right)^4} \quad (9.49)$$

В текущей версии реализована версия «универсальных» граничных условий на твердых стенках, которые совмещают идеи пристенных функций и низкорейнольдсовского подхода. Если сетка в области стенки грубая, то реализуется алгоритм пристенных функций, которые на мелкой сетке переходят к фактически «точным» граничным условиям:

$$U = 0, \quad k = 0, \quad \omega \sim 1/y^2$$

Главная хитрость модели – использование двух «пристенных» скоростей:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}, \quad u^* = C_\mu^{1/4} \sqrt{k_p}$$

### 9.8.1 Универсальные пристенные функции для компонент вектора скорости

Для задания граничного условия для скоростей используется идея «универсальных» пристенных функций, основанная на сшивке зависимостей для вязкого подслоя и логарифмической области.

Для расчета касательного напряжения на стенке используются следующие соотношения:

$$\begin{aligned} \tau_w &= \rho u_\tau u^* = c_{fw} \rho U_p^2, \\ c_{fw} &= \sqrt{c_\tau} \sqrt{c_*}, \quad \sqrt{c_\tau} = \frac{u_\tau}{U_p}, \quad \sqrt{c_*} = \frac{u^*}{U_p} \end{aligned} \quad (9.50)$$

Для расчета «пристенных» скоростей используются сопрягающие соотношения:



$$u^* = \left[ u_1^4 + \left( C_\mu^{1/4} \sqrt{k_p} \right) \right]^{0.25}, \quad u_1 = \sqrt{\frac{\nu U_p}{\delta_w}},$$

$$u_\tau = C_L u_1 + C_T u_t, \quad u_t = \frac{\kappa U_p}{\ln(Ey^+)}, \quad y^+ = \frac{u_\tau \delta_w}{\nu} \quad (9.51)$$

$$C_L = \exp(-\Gamma_U), \quad C_T = \exp\left(-\frac{1}{\Gamma_U}\right), \quad \Gamma_U = \frac{0.01(y^*)^4}{1+5y^*}$$

Здесь  $U_p$  и  $\delta_w$  – касательная составляющая скорости в пристенном КО и расстояние от центра этого КО до стенки.

Используя выражение для коэффициента трения (9.50) можно записать граничное условие для скорости в стандартном для Anes «универсальном» виде

$$\tau_w = \rho v_{\text{eff}} \frac{U_p}{y_p}, \quad v_{\text{eff}} = \nu \cdot (c_f \text{Re}_p) \quad (9.52)$$

### 9.8.2 Универсальные пристенные функции для температуры

Для расчета граничного условия для температуры используется пристенное число Стентона, аналогичное фактору трения. Единственное отличие – вместо скорости  $u_\tau$  используется скорость  $u^*$ :

$$\text{St} = \frac{q_w}{\rho c_p U_p (T_w - T_p)} = \frac{u^*}{U_p} \frac{1}{\Theta^+(y^*)}, \quad y^* = \frac{u^* \delta_w}{\nu} \quad (9.53)$$

Число Стентона позволяет задать граничное условие для температуры в «универсальном» виде, используемом в Anes:

$$q_w = \lambda_{\text{eff}} \frac{T_w - T_p}{\delta_w}, \quad \lambda_{\text{eff}} = \lambda \cdot (\text{St} \cdot \text{Re}_p \cdot \text{Pr}) \quad (9.54)$$

Для расчета безразмерной температуры используется подход Кадера, основанный на сшивке решений для ламинарной и развитой турбулентной зон:

$$\Theta^+ = \Theta_1 e^{-\Gamma} + \Theta_t e^{-1/\Gamma}, \quad \Gamma = \frac{0.01(\text{Pr} y^*)^4}{1+5\text{Pr}^3 y^*},$$

$$\Theta_1 = \text{Pr} y^*, \quad (9.55)$$

$$\Theta_t = \text{Pr}_t \left[ \frac{1}{\kappa} \ln(Ey^*) + \text{Pi} \right], \quad \text{Pi} = 9.24 \left[ \left( \frac{\text{Pr}}{\text{Pr}_t} \right)^{3/4} - 1 \right] \left[ 1 + 0.28 \exp\left(-0.007 \frac{\text{Pr}}{\text{Pr}_t}\right) \right]$$

### 9.8.3 Граничное условие для $\omega$ на твердой стенке

Значение  $\omega$  фиксируется в пристенном КО в соответствии с модифицированной моделью Ментера:

$$\omega_p = \sqrt{\omega_1^2 + \omega_t^2}, \quad \omega_1 = \frac{6\nu}{C_\mu y_p^2} = \frac{(u^*)^2}{\nu} \omega_1^+, \quad \omega_1^+ = \frac{6}{C_\mu (y^*)^2} \quad (9.56)$$

$$\omega_t = \frac{u^*}{\sqrt{C_\mu} \kappa y_p} = \frac{(u^*)^2}{\nu} \omega_t^+, \quad \omega_t^+ = \frac{1}{\sqrt{C_\mu} \kappa y^*}$$

При учете шероховатости меняется выражение для безразмерной «ламинарной»  $\omega$ :

$$\omega_1^+ = \min[\omega_r^+, \omega_0^+], \quad \omega_0^+ = \frac{6}{C_\mu (y^+)^2}, \quad \omega_r^+ = \begin{cases} \left(\frac{50}{h_s^+}\right)^2, & h_s^+ < 25 \\ \frac{100}{h_s^+}, & h_s^+ > 25 \end{cases}, \quad h_s^+ = \max\left(1, \frac{u^* h_r}{v}\right)$$

### 9.8.4 Граничное условие для k

В качестве граничного условия для k используется условие

$$\frac{\partial k}{\partial n} = 0, \tag{9.57}$$

а значение G-функции в приграничном КО вычисляется по соотношению

$$G = \left(\frac{\partial U}{\partial n}\right)^2 = \left(\frac{u^*}{\kappa y_p}\right)^2 \tag{9.58}$$

### 9.9 SST модель Ментера

Эта модель представляет собой комбинацию двух моделей: k-ε в ядре потока и k-ω вблизи твердых стенок. В этой модели для обозначения диссипации используется переменная ω. Сам такой переход осуществляется с помощью специальной сглаживающей функции F<sub>1</sub>. Там где F<sub>1</sub> = 1 используется «чистая» k-ω модель, при F<sub>1</sub> = 0 используется k-ε модель. Все коэффициенты уравнений рассчитываются с помощью линейной интерполяции

$$C = C^{(1)}F_1 + C^{(2)}(1 - F_1) \tag{9.59}$$

Ниже коэффициент с верхним индексом (1) соответствует коэффициенту для k-ω модели, с индексом (2) - k-ε.

Система уравнений SST модели, используемая в Anes, имеет вид:

$$\begin{aligned} \frac{\partial(\rho k)}{\partial \tau} + \text{div} \left( \rho \mathbf{U}k - \rho \left( \nu + \frac{\nu_t}{\sigma_k} \right) \nabla k \right) &= \rho [ \nu_t G - C_k \omega k ], \\ \frac{\partial(\rho \omega)}{\partial \tau} + \text{div} \left( \rho \mathbf{U}\omega - \rho \left( \nu + \frac{\nu_t}{\sigma_\omega} \right) \nabla \omega \right) &= \rho [ C_{\omega 1} G - C_{\omega 2} \omega^2 ] + \\ &+ 2(1 - F_1) \frac{\rho}{\omega \sigma_\omega^{(2)}} (\nabla k \cdot \nabla \omega) \end{aligned} \tag{9.60}$$

$$\nu_t = \frac{k}{\omega} \frac{1}{\max \left\{ \frac{1}{f_\mu}, \sqrt{G} F_2 \right\}}$$

Здесь:

**Турбулентные числа «Прандтля»:**

$$\sigma_k = \frac{1}{\frac{F_1}{\sigma_k^{(1)}} + \frac{1 - F_1}{\sigma_k^{(2)}}}, \quad \sigma_\omega = \frac{1}{\frac{F_1}{\sigma_\omega^{(1)}} + \frac{1 - F_1}{\sigma_\omega^{(2)}}},$$

$$\sigma_k^{(1)} = 1.176, \quad \sigma_k^{(2)} = 1, \quad \sigma_\omega^{(1)} = 2, \quad \sigma_\omega^{(2)} = 1.168$$

**Коэффициент C<sub>k</sub> в диссипативном члене для k:**

$$C_k = C_\mu f_k$$

**Коэффициенты  $C_{\omega 1}$  и  $C_{\omega 2}$ :**

$$C_{\omega 1} = f_\omega \left[ C_{\omega 1}^{(1)} F_1 + C_{\omega 2}^{(2)} (1 - F_1) \right], \quad C_{\omega 1}^{(1)} = 0.556, \quad C_{\omega 2}^{(2)} = 0.44,$$

$$C_{\omega 2} = C_{\omega 2}^{(1)} F_1 + C_{\omega 2}^{(2)} (1 - F_1), \quad C_{\omega 2}^{(1)} = 0.075, \quad C_{\omega 2}^{(2)} = 0.0828$$

**Функция-переключатель в формуле для турбулентной вязкости  $F_2$ :**

$$F_2 = \tanh \left[ \left[ \max \left( \frac{2\sqrt{k}}{C_\mu \omega \delta_w}, \frac{500\nu}{\delta_w^2 \omega} \right) \right]^2 \right]. \quad (9.61)$$

Здесь  $\delta_w$  – расстояние до ближайшей твердой стенки.

**Главная функция-переключатель  $F_1$**

Эта функция является основным сглаживающим фактором перехода от одной модели к другой:

$$F_1 = \tanh(A^4),$$

$$A = \min \left[ \max \left( \frac{\sqrt{k}}{C_\mu \omega \delta_w}, \frac{500\nu}{\delta_w^2 \omega} \right), \frac{4\rho k}{\sigma_w^{(2)} D_{k\omega} \delta_w^2} \right], \quad (9.62)$$

$$D_{k\omega} = \max \left( \frac{2\rho}{\sigma_w^{(2)} \omega} (\nabla k \cdot \nabla \omega), 10^{-10} \right),$$

Для задания граничных условий используется модель стандартной k-omega модели.

### 9.9.1 Активация моделей в Anes

Для активации k- $\omega$  модели необходимо в секции [Macro Sub] указать оператор ModelTur(trKOM)

Для включения SST версии необходимо в секции [Special Data] задать оператор ModelTur(trSST)

При использовании этих моделей *внутренние* переменные пользователя с именами "iF1SST", "iF2SST". Значение поля  $R_T$  сохраняется в переменной подсистемы tuFORM с именем "RETN".

### 9.10 Двухпараметрическая k- $\epsilon$ модель Лаундера-Шарма

Эта модель является типичной k- $\epsilon$  моделью 80-ых годов прошлого столетия для низких турбулентных чисел Рейнольдса:

$$\begin{aligned} \frac{\partial(\varphi\rho_g k)}{\partial\tau} + \operatorname{div}\left(\varphi\rho_g \mathbf{U}_g k - \varphi\rho_g \left(v_g + \frac{v_t}{\sigma_k}\right)\nabla k\right) &= \varphi\rho_g (P_k + G_b - \varepsilon - D_t) + \varphi S_k, \\ \frac{\partial(\varphi\rho_g \varepsilon)}{\partial\tau} + \operatorname{div}\left(\varphi\rho_g \mathbf{U}_g \varepsilon - \varphi\rho_g \left(v_g + \frac{v_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right) &= \varphi\rho_g [C_{\varepsilon 1}(P_k + C_{\varepsilon 3}G_b) - C_{\varepsilon 2}f_\varepsilon \varepsilon] \frac{\varepsilon}{k} \\ &+ \varphi\rho_g E_t + \varphi S_\varepsilon, \end{aligned} \quad (9.63)$$

$$v_t = C_\mu f_\mu \frac{k^2}{\varepsilon}$$

В этой модели демпфирующие функции имеют вид:

$$f_\mu = \exp\left[-\frac{3.4}{\left(1 + \frac{R_T}{50}\right)^2}\right], \quad f_\varepsilon = 1 - 0.3\exp(-R_T^2), \quad R_T = \frac{k^2}{\varepsilon \cdot \nu} \quad (9.64)$$

а дополнительные источники:

$$D_t = 2v_g \sum_{k=1}^3 \left(\frac{\partial\sqrt{k}}{\partial x_k}\right)^2, \quad E_t = 2v_g v_t \sum_{i=1}^3 \sum_{k=1}^3 \left(\frac{\partial^2 u_i}{\partial x_k^2}\right)^2 \quad (9.65)$$

На твердых стенках задаются естественные граничные условия:

$$k = 0, \quad \varepsilon = 0$$

Особенности использования модели Лаундера-Шарма:

1. Данная модель была разработана для задач с простой геометрией расчетной области. Поэтому она реализована только для структурных сеток.
2. Из-за наличия членов  $D_t$  и  $E_t$  модель обладает «плохой» сходимостью. Для улучшения сходимости необходимо задавать значение оператора секции [Turbulence]  $NuTVAR = 0$ .

В этом случае на начальном этапе расчета ( $iSWEEP < SweepBeforeKE$ ) для расчета турбулентной вязкости используется алгебраическая LVEL модель турбулентности (дополнительно необходимо активировать  $\Phi$ -переменную с именем "WDIS").

### 9.11 Особенности алгоритма решения двухпараметрических уравнений турбулентности

Внутренние источниковые члены (9.34) и (9.58) обладают одной неприятной особенностью – сильной нелинейностью. Это часто приводит к медленной сходимости итерационного процесса (а иногда к его расходимости). В пакете предусмотрено два механизма улучшения сходимости:

- 1) различные схемы линеаризации внутренних источников,
- 2) использование двух шагового итерационного процесса.

#### 9.11.1 Линеаризация источниковых членов

Как уже не раз отмечалось, под линеаризацией источников подразумевается представление их в виде

$$S^{(1)} = B - A\Phi,$$

который легко преобразовать к стандарту Anes

$$S^{(1)} = C(V - \Phi).$$

В Решателе реализованы несколько вариантов линеаризации источниковых членов для  $k$  и  $\varepsilon$  ( $\omega$ ). Рассмотрим эти варианты на примере источникового члена в уравнении для  $k$ . В пакете предусмотрено три варианта.

1) Вариант lkeDEFAULT: в этом случае используется «естественная» линеаризация

$$B_k = \varphi \rho_g P_k + \varphi P_a, \quad A_k = \varphi \rho_g \frac{\varepsilon}{k}$$

2) Вариант lkeEQU: в этом случае используются соотношения

$$B_k = \varphi \rho_g \left( \frac{3}{2} P_k + (c_{\varepsilon 2} - 1) \varepsilon \right) + \varphi P_a, \quad A_k = \varphi \rho_g \left( c_{\varepsilon 2} \frac{\varepsilon}{k} + \frac{P_k}{2k} \right)$$

3) Вариант lkeLENM: в этом случае используются соотношения

$$B_k = \varphi \rho_g \left( P_k + \frac{\varepsilon}{2} \right) + \varphi P_a, \quad A_k = \varphi \rho_g \frac{3\varepsilon}{2k}$$

Какой вариант выбрать, зависит от задачи и определяется экспериментально. По умолчанию используется вариант lkeDEFAULT. Для указания алгоритма линеаризации используется оператор секции [Turbulence]:

LinearKEPS = LkeEQU / LkeLENM / lkeDEFAULT

### 9.11.2 Двухшаговый итерационный процесс


Для улучшения сходимости необходимо задать разумные начальные поля как для турбулентной энергии и диссипации, так и для поля скорости.

В пакете для этого предусматривается следующий алгоритм:

1. В начале итерационного процесса ( $iSWEET < SweepBeforeKE$ )  $k$ - $\varepsilon$  ( $\omega$ ) уравнения решаются, однако для расчета турбулентной вязкости используется алгебраическая модель, заданная пользователем (обычно используется постоянное значение турбулентной вязкости или LVEL-модель).
2. На итерации  $iSWEET = SweepBeforeKE$  производится переключение расчета турбулентной вязкости по соотношениям модели и начинается полный расчет уравнений для  $k$  и  $-\varepsilon$  ( $\omega$ ).
3. Если расчет начинается с рестарта или другого варианта, то с первой итерации начинается решение  $k$  и  $-\varepsilon$  ( $\omega$ ) уравнений, поскольку турбулентные поля уже определены.

Значение параметра SweepBeforeKE можно задать с помощью оператора секции [Turbulence]:

SweepBeforeKE = значение

 **Замечание.** Значение турбулентной вязкости на интервале  $iSWEET < SweepBeforeKE$  определяется оператором NuTVAR секции [Turbulence]. Если этот параметр не задан или значение равно нулю, то в качестве алгебраической модели используется LVEL модель турбулентности!

При решении нестационарных задач момент «включения» расчета турбулентной вязкости на шаге по времени определяется переменной TimeBeforeKE. Включение производится при  $iTime > TimeBeforeKE$ . Значение параметра TimeBeforeKE можно задать с помощью оператора секции [Turbulence]:

TimeBeforeKE = значение

## 9.12 Экспериментальные $k$ - $\varepsilon$ модели

В текущей версии кода реализованы новые модели, которые находятся в состоянии отладки. Ниже кратко описаны идеи этих моделей.

### 9.12.1 Модель с F-точкой для неструктурных сеток

При использовании модели турбулентности с пристенными функциями главным условием ее применимости является условие попадания узла пристенного КО в область развитого турбулентного течения. По другому это условие можно сформулировать как  $Y^+(\text{пристенного КО}) > 20$ .

При работе с структурными сетками и простой геометрией РО это условие легко выбрать путем подбора сетки. При работе с неструктурными сетками и сложной геометрией из-за использования дробных ячеек дистанция от центра пристенной ячейки до границы  $\delta_n$  может меняться в широких пределах. Это приводит к появлению пульсаций касательного напряжения на стенке и температуры стенки (или теплового потока).

В литературе предложены ряд моделей для борьбы с этим недостатком сеток с дробными ячейками (см., например [9]). Одна из таких моделей реализована для равновесной турбулентной модели trwDBS.

Основная идея заключается в следующем. Для расчета коэффициента сопротивления  $c_{fw}$  (4.34) и числа Стентона (4.37) используется не центр  $P$  пристенного КО, а точка, расположенная на фиксированном расстоянии от твердой стенки – точка  $F$  (см. рисунок 4.4).

Это позволяет вместо сильно переменной величины  $\delta_p$  использовать постоянное расстояние  $\delta_F$ .

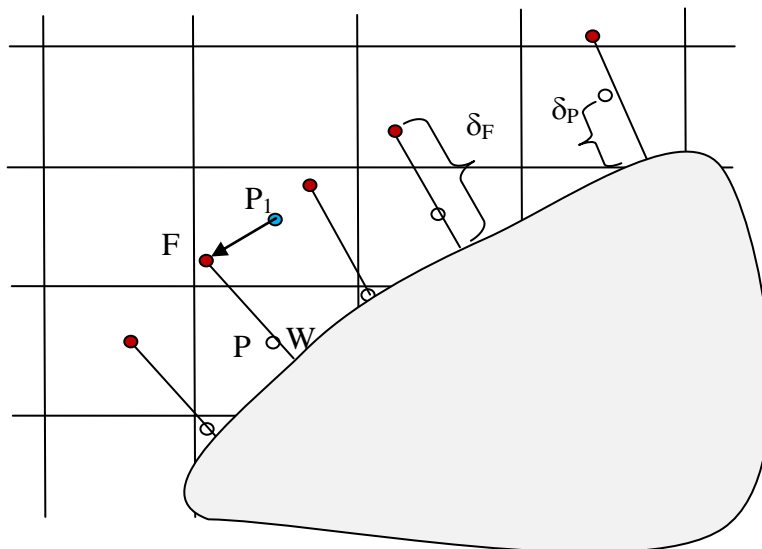


Рисунок 9.4 – Расположение точек  $F$  для пристенных КО

Для расчета коэффициента трения необходимо вычислить касательную составляющую скорости в точке  $F$ . Для этого используются компоненты вектора скорости и их градиенты из КО, в который «попадает» точка  $F$ :

$$\vec{U}_F = \vec{U}_{P_1} + (\nabla U)_{P_1} \cdot (\vec{F} - \vec{P}_1)$$

В текущей экспериментальной версии точка  $F$  используется только для расчета касательного напряжения на стенке. Все остальные соотношения равновесной модели приме-

няются к пристенному дробному КО. Это не совсем правильно, по этому модель требует дльнейшего развития.

Модель активируется с помощью оператора секции [Special Data]:  
`L("IsWALLTUR_EXT") = .TRUE.`

Для задания фиксированной длины  $\delta_F$  используется соотношение  

$$\delta_F = C_\delta \min(dx, dy, dz)$$

где  $dx, dy, dz$  – размеры пристенных КО по осям  $x, y, z$ ,  $C_\delta$  - коэффициент коррекции, по умолчанию  $C_\delta = 1.2$ . Значение этого коэффициента можно изменить с помощью оператора

`R("DIST_WALL_F") = 1.2`

Для визуализации положения F точки для всех пристенных КО можно использовать оператор секции [Special Data]

`L("WALLTUR_EXT_VTK") = .TRUE.`

В этом случае Решатель создает vtk-файл с именем

<Префикс Результатов>\_\_walltur.vtk,

в котрый помещаются линия P<sub>1</sub>-F-W для всех пристенных ячеек. Этот файл можно просмотреть в постпроцессоре ParaView (вместе с файлами описания ячеек и граничных граней).

### 9.12.2 Гибридная модель

При использовании двухслойной модели требуется, чтобы в пристенные КО находились в области вязкого подслоя ( $y^+ < 2$ ). При расчете сложных течений, несмотря на использование мелкой сетки, возможно, что в некоторых пристенных КО  $y^+$  будет больше 2. Очень часто это происходит в процессе сходимости.

Гибридная модель подключает пристенную неравновесную модель trwLAUND к двухслойной модели tr2LREY.

В этой модели во всей РО используется двухслойная модель. Однако, если в пристенном КО

$R_k(P) > R_{hyb}$ ,

то для расчета всех характеристик этого КО используются пристенные функции неравновесной модели. По умолчанию критическое значение  $R_{hyb} = 10$ . Это значение можно изменить с помощью оператора секции [Special Data]

`R("CtRETN_HYBRIDCri") = 10`

Для анализа пристенных зон «включения» пристенных функций можно использовать внутреннюю переменную пользователя \_USEWALLFUN. Это поле содержит 1 для пристенных КО, в которых «включены» пристенные функции, в остальных КО поле содержит 0.

## 10. Дополнительные математические модели

### 10.1 S2S модель радиационного переноса

Моделирование процесса теплообмена за счет излучения является чрезвычайно сложной задачей, особенно если теплоноситель не является оптически прозрачным, и оптические свойства зависят от спектра излучения. Если газ не поглощает излучение (оптическая длина для всех частот спектра значительно больше характерных размеров каналов), а сами поверхности можно считать диффузными [15], то для расчета теплообмена излучением можно использовать модель «Система поверхностей» или S2S модель (Surface – to – Surface модель).

В текущей версии кода Anes эта модель реализована для структурных сеток.

#### 10.1.1 Основные соотношения S2S модели

Рассмотрим замкнутую систему поверхностей (радиационных зон) с площадями  $A_k$ ,  $k = 1 \dots N$  (см. рисунок 10.1).

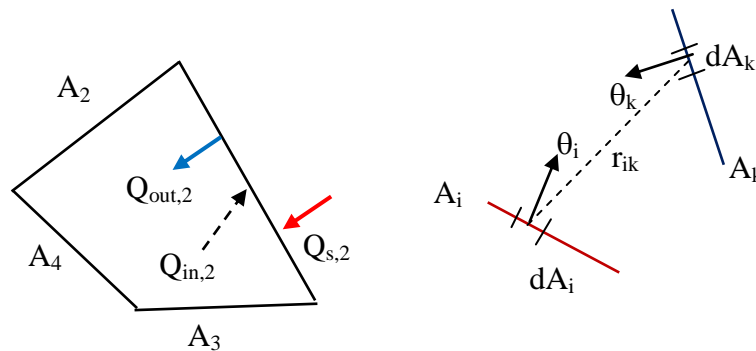


Рисунок 10.1 – К определению параметров S2S модели излучения

Будем обозначать ниже через  $Q_i$  поток тепла (Вт), связанный с зоной, а через  $q_i$  – плотность этого потока тепла ( $\text{Вт}/\text{м}^2$ ):

$$Q_i = q_i A_i \tag{10.1}$$

Будем считать, что каждая зона имеет постоянную температуру  $T_i$ . Результирующий поток тепла  $i$ -ой зоны  $Q_{\text{rad},i}$  можно записать в виде[15]:

$$Q_{\text{rad},i} = Q_{\text{in},i} - Q_{\text{out},i} \tag{10.2}$$

где  $Q_{\text{out},i}$  – эффективный поток излучения от  $i$ -ой зоны к другим зонам,  $Q_{\text{in},i}$  – падающий поток излучения на  $i$ -ую зону от других зон. Будем считать, что знак результирующего потока положителен, если он направлен к зоне.

Для описания «распределения» потоков от  $i$ -ой зоны к другим зонам вводится понятие среднего углового коэффициента  $F_{ik}$ . Доля эффективного потока  $Q_{\text{out},i}$  с  $i$ -ой поверхности, попавшего на  $k$ -поверхность  $Q_{\text{in},i-k}$ , равна:

$$Q_{\text{in},i-k} = F_{ik} Q_{\text{out},i} \tag{10.3}$$

Угловые коэффициенты являются чисто геометрическими характеристиками зон и в общем случае могут быть рассчитаны по формуле:



$$F_{ik} = \frac{1}{A_i} \int_{A_i} \int_{A_k} \frac{\cos(\theta_i) \cos(\theta_k)}{\pi r_{ik}^2} dA_i dA_k \quad (10.4)$$

Из соотношений (10.3) и (10.4) следуют два свойства угловых коэффициентов:

$$\sum_k F_{ik} = 1, \quad F_{ik} A_i = F_{ki} A_k \quad (10.5)$$

Эффективный поток излучения состоит из двух потоков: собственного излучения и отраженного излучения:

$$Q_{out,i} = E_i A_i + \rho_i Q_{in,i}, \quad \rho_i = (1 - \varepsilon_i), \quad E_i = \varepsilon_i \sigma T_i^4 \quad (10.6)$$

Здесь  $\varepsilon_i$  - степень черноты поверхности,  $\rho_i$  – коэффициент отражения,  $\sigma$  - постоянная Стефана-Больцмана ( $\sigma = 5.67032 \cdot 10^{-8}$  Вт/(м<sup>2</sup>·К<sup>4</sup>)). В S2S модели поверхности считаются непрозрачными, поэтому сумма степени черноты и коэффициента отражения равна единице.

Второе соотношение (10.5) (соотношение взаимности) позволяет переписать (10.3) через плотности потоков:

$$q_{in,i-k} = \frac{Q_{in,i-k}}{A_k} = \frac{F_{ik} Q_{out,i}}{A_k} = \frac{F_{ik} A_i}{A_k} q_{out,i} = F_{ki} q_{out,i}$$

Используя вышеприведенные соотношения можно перейти от *потоков* к *их плотностям* и записать систему уравнений (10.6) в следующем виде:

$$\begin{aligned} q_{out,i} &= \sigma \varepsilon_i T_i^4 + (1 - \varepsilon_i) q_{in,i} = \sigma \varepsilon_i T_i^4 + (1 - \varepsilon_i) \sum_k q_{in,k-i} = \\ &= \sigma \varepsilon_i T_i^4 + (1 - \varepsilon_i) \sum_k F_{ik} q_{out,k} \end{aligned}$$

или

$$\sum_k B_{ik} q_{out,k} = \sigma \varepsilon_i T_i^4, \quad B_{ik} = \delta_{ik} - (1 - \varepsilon_i) F_{ik} \quad (10.7)$$

Система линейных уравнений (10.7) относительно эффективных плотностей потоков позволяет при заданных температурах зон найти эти потоки. Эта система является основой S2S модели различных CFD кодов (например, Fluent). В коде Anes используется другая форма «радиационных» уравнений (10.7), записанных через плотности результирующих потоков  $q_{rad,i}$ . Подобная форма уравнений используется в частности CFD коде Phoenix.

Используя очевидные соотношения систему (10.7) можно записать в виде [16]:

$$q_{rad,i} = \sum_k R_{ik} \sigma (T_k^4 - T_i^4) \quad (10.8)$$

Здесь  $R_{ik}$  - матрица коэффициентов «обмена излучением» (Radiation Exchange Matrix). Эта матрица рассчитывается через матрицу Гибхарта  $B_{ik}$  [16]:

$$\begin{aligned} R_{ik} &= \varepsilon_i \varepsilon_k B_{ik}, \quad B_{ik} = [m_{im}]^{-1} F_{mk}, \\ m_{ik} &= \delta_{ik} - (1 - \varepsilon_k) F_{ik} \end{aligned} \quad (10.9)$$

где  $[m_{im}]^{-1}$  обозначает обратную матрицу.

Замечание: Матрицы  $R_{ik}$  и Гибхарта  $B_{ik}$  обладают следующими свойствами:

$$\sum_i \varepsilon_i B_{ik} = 1, \quad \sum_k R_{ik} = \varepsilon_i, \quad k = 1..N, \quad (10.10)$$

которые можно использовать (наряду с (10.5)) для проверки правильности построения геометрии поверхностей.

### 10.2 Реализация S2S модели в коде Anes

Для моделирования радиационного переноса в расчетную область необходимо добавить поверхностные патчи типа BS, Surface или SF\_, имена которых начинаются с "#R\_" (см. рисунок 10.2). Эти патчи рассматриваются в коде как радиационные поверхности (радиационные патчи). Если такие патчи присутствуют в проекте задачи, то модель S2S активируется.

Важно отметить, что:

- 1) радиационные патчи типа BS\_ (описывающие границу РО), должны соприкасаться только с G-фазой,
- 2) радиационные патчи типа Surface должны касаться либо FS-границы, либо располагаться целиком в G-фазе (они не могут располагаться внутри S-фазы).

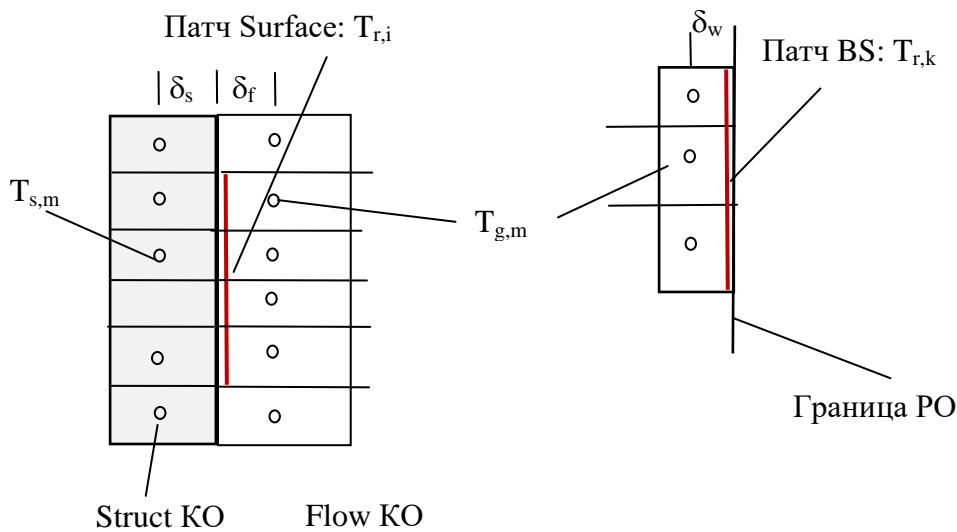


Рисунок 10.2 – Геометрия S2S модели

Основой радиационной модели кода является система уравнений (10.8). Если предположить, что радиационная поверхность имеет нулевую толщину и постоянную радиационную температуру  $T_{r,i}$ , то уравнение *баланса тепла* радиационной зоны можно записать в виде:

$$\begin{aligned}
 q_{flow,i} + q_{struct,i} + q_{bc,i} + q_{rad,i} &= 0, \\
 q_{rad,i} &= \sum_k R_{ik} \sigma (T_{r,k}^4 - T_{r,i}^4), \\
 q_{flow,i} &= \frac{1}{A_i} \sum_m \alpha_{flow,m} \Delta A_m (T_{g,m} - T_{r,i}), \quad \alpha_{flow,m} = \frac{\lambda_{g,eff}}{\delta_f}, \\
 q_{struct,i} &= \frac{1}{A_i} \sum_m \alpha_{struct,m} A_m (T_{s,m} - T_{r,i}), \quad \alpha_{struct,m} = \frac{\lambda_s}{\delta_s}, \\
 q_{bc,i} &= \frac{1}{A_i} \sum_m C_{b,m} A_m (V_{b,m} - T_{r,i})
 \end{aligned} \tag{10.11}$$

Здесь  $m$  - индекс суммирования по всем граням радиационного патча,  $\Delta A_m$  - площадь грани приграничной ячейки радиационного патча,  $T_{g,m}$  и  $T_{g,s}$  - температуры G- и S- фаз в приграничных ячейках,  $\lambda_{g,eff}$  - эффективный коэффициент теплопроводности приграничной G-ячейки,  $\lambda_s$  - коэффициент теплопроводности приграничной S-ячейки,  $C_b$ ,  $V_b$  - коэффициент и источник граничного условия. Знаки плотностей потоков в уравнении баланса выбраны в соответствии с идеологией, принятой в Anes: *поток положителен, если он*

направлен к радиационной зоне. Если  $q_{rad,i} > 0$ , то результирующий поток тепла за счет излучения направлен к  $i$ -ой зоне, если  $q_{struct,i} > 0$ , то это означает, что Struct-ячейки «греют» радиационную зону (поток направлен от Struct-ячейки к радиационной зоне).

Величина  $q_{struct,i}$  - это средняя плотность теплового потока от Struct-ячеек, грани которых принадлежат радиационному патчу с индексом  $i$ , к радиационной зоне. В текущей версии кода принимается, что на FS-границе теплообмен с радиационной зоной осуществляют только со Struct-ячейками, поэтому плотность потока  $q_{flow,i}$  полагается равной нулю (фактически это означает, что радиационная зона «немного» перемещается с границы внутрь Struct-ячейки). Для описания FS- радиационной зоны можно использовать либо поверхностный Surface-патч, либо SF-патч, создаваемый автоматически вместе с FS-патчем. В последнем случае радиационный патч может описывать «кривую» поверхность.

Отметим, что в этом случае S2S модель корректирует площади граней SF-патча так, чтобы их суммарная площадь была равна истинной плотности поверхности патча (напомним, что при работе со структурными сетками используется модель целых ячеек).

Если радиационная зона описывается граничным патчем BS, то граничное условие для  $T_g$ , заданное пользователем (коэффициент  $C_b$  и значение  $V_b$ ), «переключается» на радиационную поверхность. Для температуры автоматически формируется новое граничное условие первого рода с  $V_b$ , равной радиационной температуре зоны.

Для «правильного» расчета радиационного переноса необходимо, чтобы система радиационных зон была замкнута (для каждой зоны сумма угловых коэффициентов должна быть равна единице). Поэтому при моделировании расчетных областей с входными и выходными границами необходимо использовать «замыкающие» радиационные зоны, расположенные на таких границах. Для их описания можно использовать либо Surface-патчи, целиком расположенные в G-фазе, или BS-патчи с заданными адиабатическими граничными условиями. Для таких зон в S2S модели плотности потоков  $q_{struct,i}$ ,  $q_{flow,i}$  б  $q_{bc,i}$  равны нулю. Это условие эквивалентно «отражению» излучения от таких зон.

При заданных температурах G- и S- фаз система (10.11) позволяет рассчитать радиационные температуры патчей  $T_{r,i}$  и все плотности радиационных потоков. В Решателе для решения (10.11) используется итерационный метод Ньютона. Для этого система записывается через поправку радиационной температуры:

$$\begin{aligned} T_{r,i} &= T_{r,i}^* + \delta T_i, \quad a_i \delta T_i = r_i, \\ a_i &= \frac{1}{A_i} \sum_m \alpha_{struct,m} \Delta A_m + \frac{1}{A_i} \sum_m \alpha_{flow,m} \Delta A_m + \frac{1}{A_i} \sum_m C_{b,m} \Delta A_m + 4(T_{r,i}^*)^3 \sum_k R_{ik} \sigma, \\ r_i &= \sum_k R_{ik} \sigma [(T_{r,i}^*)^4 - (T_{r,k}^*)^4] + \frac{1}{A_i} \sum_m \alpha_{struct,m} A_m (T_{s,m} - T_{r,i}^*) + \\ &+ \frac{1}{A_i} \sum_m \alpha_{flow,m} A_m (T_{g,m} - T_{r,i}^*) + \frac{1}{A_i} \sum_m C_{b,m} A_m (V_{b,m} - T_{r,i}^*) \end{aligned} \quad (10.12)$$

Для «обратной» связи радиационного переноса и теплообмена в уравнении энергии для  $T_s$  появляются сопряженные поверхностные источники на FS-границах:

$$S_{T_s,m} = \alpha_{struct,m} A_m (T_{r,i} - T_{s,m})$$

и поверхностный источник для  $T_g$  на граничных BS-патчах:

$$S_{T_g,m} = \alpha_{flow,m} A_m (T_{r,i} - T_{g,m}).$$

Для замыкания системы радиационных уравнений пользователь должен задать следующие параметры.

1. Для каждого радиационного патча нужно в секции [Special Data] задать степень черноты с помощью оператора:

$$R(\text{«Имя радиационного патча»}.Reps) = \varepsilon_i$$

2. Необходимо подготовить файл со значениями угловых коэффициентов  $F_{ik}$ . Имя этого файла должно быть указано в операторе секции [Special Data]:

C("FViewFile") = <путь к файлу>

Это файл представляет собой текстовый файл следующей структуры:

```
<Число радиационных поверхностей: NRS>
<Имя поверхности 1 >
<Угловые коэффициенты: (F1,k, k = 1, NRS) >
.....
<Имя поверхности NRS >
<Угловые коэффициенты: (FNRS,k, k = 1, NRS) >
```

Важно, что индекс k радиационных зон в массиве  $F_{i,k}$  должен соответствовать переименованию имен зон <Имя поверхности i> в этом файле и число радиационных зон должно быть равно числу радиационных патчей. Угловые коэффициенты должны быть выведены по формату Фортрана "6E13.6".

Если пользователь независимо подготовил угловые коэффициенты в массиве  $F_{i,k} = FView(1:N_{rs}, 1:N_{rs})$  и массив имен радиационных зон - NameRZ(i), то файл можно создать следующими операторами Фортрана:

```
WRITE(LU_OUT,'(i3)') NoRS
DO IRZ = 1,NoRS
  WRITE(LU_OUT,'(a16)') Name(IRZ)
  WRITE(LU_OUT,'(6E13.6)') (FView(IRZ,JRZ),JRZ=1,NoRS)
ENDDO
```

Для осесимметричных задач файл угловых коэффициентов можно создать с помощью утилиты aCalcFView, которая входит в состав дистрибутива Anes.

Результаты решения S2S уравнений (10.2) выводятся в листинг расчета в виде. Ниже приведен пример листинга демонстрационной задачи, описанной ниже.

```
----- (Результаты S2S Модели Излучения) -----
----- ПЛОТНОСТИ потоков радиационных зон -----
--- Имя Зоны -- Trad,K -- (Trad-T0), C -- qRAD(Вт/м^2) -- qSRTUCT ---- qBC ---
- qSUM ---
#R_RIGHT    2067.5    194.53    -48177.    0.0000    48177.    -0.25949E-01
#R_BOT      1984.5    111.54    0.10159E+06  0.0000    -0.10159E+06  0.32983E-01
#R_UP       1898.0    24.970    0.21973E+06  0.0000    -0.21973E+06 -0.18622E-01
#R_IN1      2033.4    160.35    -3984.2    3984.2    0.0000    -0.44890E-03
#R_IN2      2031.3    158.32    -4567.4    4567.4    0.0000    0.50375E-05
#R_IN3      2018.6    145.55    -5943.6    5943.6    0.0000    -0.19556E-03
#R_OUT1     2052.9    179.91    3827.5    -3827.5    0.0000    -0.32947E-02
#R_OUT2     2053.9    180.90    4381.3    -4381.3    0.0000    -0.33115E-02
#R_OUT3     2047.7    174.65    5738.3    -5738.3    0.0000    -0.31698E-02

----- ПОТОКИ радиационных зон -----
--- Имя Зоны -- Trad,K --- Qrad(кВт) -- Qstruct ---- Qbc ---- Qsum ---
#R_RIGHT    2067.5    -90.333    0.0000    90.333    -0.48654E-04
#R_BOT      1984.5    28.571    0.0000    -28.571    0.92765E-05
#R_UP       1898.0    61.800    0.0000    -61.800    -0.52374E-05
#R_IN1      2033.4    -0.29881    0.29881    0.0000    -0.33668E-07
#R_IN2      2031.3    -0.34256    0.34256    0.0000    0.37782E-09
#R_IN3      2018.6    -0.44577    0.44577    0.0000    -0.14667E-07
#R_OUT1     2052.9    0.29663    -0.29663    0.0000    -0.25534E-06
#R_OUT2     2053.9    0.33955    -0.33955    0.0000    -0.25664E-06
#R_OUT3     2047.7    0.44472    -0.44472    0.0000    -0.24566E-06
Сумма qRAD(+)*AreaZone.....= 91.452 кВт
Сумма qRAD(-)*AreaZone.....= 91.420 кВт
Разница (+) и (-).....= 0.31982E-01 кВт
```

В первой таблице:

- 1) первая колонка: имена всех радиационных патчей,
- 2) вторая: радиационная температура патча  $T_{r,i}$  в градусах Кельвина ,
- 3) третья: радиационная температура патча  $T_{r,i}$  в размерности, принятой в проекте (температура в Кельвинах за вычетом отсчета температуры  $T_0$ ),
- 4) четвертая: плотность результирующего потока  $q_{rad,i}$  (еще раз напомним, что поток положителен, если он направлен к радиационной зоне),
- 5) пятая и шестая : плотности потоков  $q_{struct,i}$  и  $q_{bc,i}$ ,
- 6) седьмая - невязка баланса плотностей потоков.

Вторая таблица содержит аналогичные поля для самих потоков ( $Q_i = q_i \cdot A_i$ )

Аналогичные поля выводятся в текстовые файлы с расширениями \*.as2s и \*.cs2s. Эти файлы используются для продолжения расчета с контрольной точки (\*.cs2s) и для начала расчета с другого варианта (\*.as2s).

При работе с подсистемой muFORM можно использовать функцию

TotRadFLUX("Имя патча"),

для получения доступа к  $Q_{rad}$  радиационного патча.

### 10.3 Утилита CalcFView

Утилита предназначена для расчета угловых коэффициентов  $F_{ik}$  для осесимметричной геометрии. Радиационные поверхности представляют «отрезки» в двумерной цилиндрической системе координат  $(r,z)$  (см. рисунок 10.3):

- 1) верхняя или нижняя поверхность диска (1) или кольца диска (2),
- 2) внутренняя или внешняя поверхность цилиндра (3),
- 2) внутренняя или внешняя поверхность конуса (4).

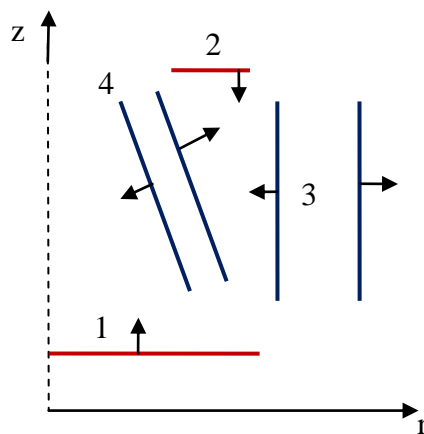


Рисунок 10.3 - Формы радиационных поверхностей утилиты aCalcFView

Утилита представляет собой консольную программу. Команда запуска:

```
<anes>\bin\CalcFView.exe -i:<in файл> [-f: <fview файл>]
```

Здесь

<in файл> - имя файла с исходными данными утилиты,

<fview файл> - имя выходного файла с матрицей угловых коэффициентов.

Если имя выходного файла не указано, то оно формируется из имени файла с исходными данными путем замены расширения на ".fview".

Формат файла исходных данных:

```
NoSurf NoRZ NoTETA
"<Имя зоны 1>"
rF rL zF zL NormR NormZ
.....
.....
"<Имя зоны NoSurf>"
rF rL zF zL NormR NormZ
```

Первая строка содержит три целых числа:

NoSurf - число радиационных зон,

NoRZ - число разбиения поверхности в направлении осей  $r$  и  $z$ ,

NoTeta - число разбиения поверхности в направлении угловой координаты  $\theta$ .

Далее идет NoSurf пар строк. Первая строка пары содержит имя радиационной зоны, вторая содержит 6 действительных чисел, описывающих «отрезок» поверхности в  $(r,z)$  СК и ее нормаль:

$rF, rL$  - начало и конец отрезка поверхности по оси  $r$ ,

$zF, zL$  - начало и конец отрезка поверхности по оси  $z$ ,

NormR , NormZ - одно из трех значений (-1, 0, 1), которые определяют *знак* компоненты нормали к поверхности.

Рассмотрим пример расчета угловых коэффициентов для демонстрационной задачи, описанной в следующем разделе. В этой задаче рассматриваются 9 радиационных зон, изображенных на рисунке 10.4.

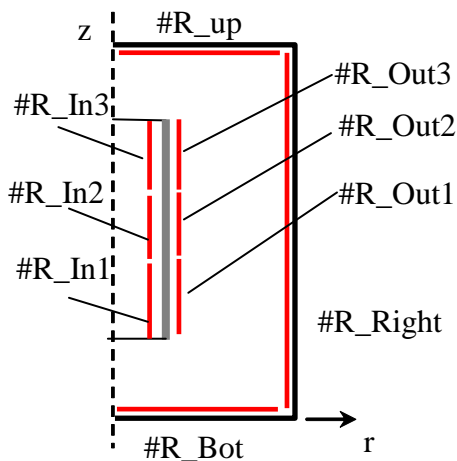


Рисунок 10.4 - Радиационные зоны демонстрационной задачи

Файл исходных данных для этой задачи имеет следующий вид;

```
9 20 20 ! NoSurf, NRZ , NTETA
"#R_RIGHT"
0.750 0.750 0.0 2.5 -1 0
"#R_BOT"
0.000 0.750 0.0 0.0 0 1
"#R_UP"
0.000 0.750 2.5 2.5 0 -1
"#R_IN1"
0.249 0.199 0.5 1.0 -1 -1
"#R_IN2"
0.198 0.148 1.0 1.5 -1 -1
```

```

"#R_IN3"
0.147 0.097 1.5 2.0 -1 -1
"#R_OUT1"
0.251 0.201 0.5 1.0 1 1
"#R_OUT2"
0.202 0.152 1.0 1.5 1 1
"#R_OUT3"
0.153 0.103 1.5 2.0 1 1

```

Для расчета угловых коэффициентов утилита использует «прямое» вычисление интеграла (10.4) методом прямоугольников. Для этого поверхности двух зон, для которых рассчитывается угловой коэффициент, разбивается на прямоугольные ячейки (число ячеек вдоль осей rz равно NoRZ, по углу  $180^\circ - \text{NoTeta}$ ) и производится суммирование подинтегральной функции по всем парам ячеек. При этом учитывается эффект «затенения» другими поверхностями: если луч между центрами ячеек пересекается с другими зонами, то эти ячейки не учитываются в интеграле.

После расчета угловых коэффициентов производится проверка качества расчета, для чего используются соотношения (10.5). Результат проверки выводится в листинг утилиты в виде:

```

===== СУММА угловых коэффициентов должно = ЕДИНИЦЕ =====
---- ID ---- Name ----- SUM(FView) -----
 1 #R_RIGHT          0.9816
 2 #R_BOT            0.9911
 3 #R_UP             0.9910
 4 #R_IN1            1.002
 5 #R_IN2            1.002
 6 #R_IN3            1.003
 7 #R_OUT1           1.000
 8 #R_OUT2           1.000
 9 #R_OUT3           1.000

===== Свойство взаимности =====
Поверхность 1 ="#R_RIGHT"
---- Поверхность 2 ---- F12 ---- F12*A12 ---- F21*A21 ----
      #R_BOT      0.1317      1.552      1.552
      #R_UP       0.1317      1.552      1.552
      #R_IN1      0.1886E-02  0.2222E-01  0.2222E-01
      #R_IN2      0.1053E-05  0.1240E-04  0.1240E-04
      #R_IN3      0.1883E-02  0.2219E-01  0.2219E-01
      #R_OUT1     0.3686E-01  0.4342      0.4342
      #R_OUT2     0.3833E-01  0.4515      0.4515
      #R_OUT3     0.3691E-01  0.4348      0.4348

```

#### 10.4 Пример использования S2S модели

В библиотеке примеров (каталоге project\_lib\rad) расположен пример задачи, использующий S2S модель переноса излучения. В этой задаче рассматривается нагрев полого цилиндра с тонкой стенкой, расположенного внутри цилиндрической печи, заполненной парами кремния. Каталог содержит:

- 1) проект s\_rad\_cyl9.a - файл проекта с S2S моделью излучения,
- 2) проект s\_norad\_cyl9.a - файл проекта без учета излучения,
- 3) MakeFView.bat, cyl9.in - скрипт и файл исходных данных для утилиты aCalcFView.

## Литература

1. Код Anes20хе. «Описание численных алгоритмов кода», версия 2.24, 2019.
2. Код Anes20хе. «Работа с проектом пользователя», версия 2.24, 2019.
3. С. Патанкар. Численные методы решения задач тепломассообмена и динамики жидкости: Пер. с англ. - М.: Энергоатомиздат, 1984. - 152 с.
4. Ferziger, J.H. and Peric, M.: Computational methods for fluid dynamics: Springer Verlag, Berlin-New York, 1995.
5. Патанкар С.В. Численное решение задач теплопроводности и конвективного теплообмена при течении в каналах// М.: Издательство МЭИ, 2003.
6. Ю.В. Лапин, М.Х.Стрелец. Внутренние течения газовых смесей.// М.: Наука, 1989, 368 с.
7. Д.А. Франк-Каменецкий. Основы макрокинетики. Диффузия и теплопередача в химической кинетике: Учебник-монография// Долгопрудный: ИД «Интеллект», 2008, 408 с.
8. V. R. Voller and C. Prakash. A fixed grid numerical modeling methodology for convection-diffusion mushy region phase-change problems// Int. J. Heat Mass Transfer. 1988, Vol. 30, No 8, pp. 1709 – 1719.
9. V. R. Voller and A.D. Brent, C. Prakash. The modeling of heat, mass and solute transport in solidification systems. Int. J. Heat Mass Transfer. 1989, Vol. 32, No 9, pp. 1719 – 1731.
10. В.Е. Launder. О расчете конвективного теплообмена в сложных турбулентных течениях// Современное машиностроение, серия А, 1989, № 9, с. 69-89.
11. R. H. Diaz. Critical evaluation and development of one-equation near-wall turbulence models// Dissertation Doctor of Philosophy, University of Maryland, College Park, 2003.
12. Bai C., Gosman A. D. Development of methodology for spray impingement simulation. – SAE Technical Paper, 1995. – №. 950283.
13. R. Clift, J. Grace, M. Weber . Bubbles, Drops, and Particles// Academic Press, 1978.
14. A. Gosman, E. Ioannides. Aspects of computer simulation of liquid-fuelled combustors// AIAA-81-0323, AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri, USA, 1981.
15. А.Г. Блох, Ю.А. Журавлев, Л.Н. Рыжков. Теплообмен излучением// М.: Энергоатомиздат, 1991, 432 с.
16. Справочник по теплообменникам: В 2 т. Т. 1// Перевод с англ. под редакцией Б.С. Петухова, В.К. Шикова, М.: Энергоатомиздат, 1987, 560с.