

Anes Team



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

Версия 2.24

VOF-алгоритм кода

Москва 2019 г.

Оглавление.

ВВЕДЕНИЕ	3
1. ОСНОВНЫЕ УРАВНЕНИЯ.....	3
1.1. Соотношения на фазовой границе	3
1.2. Уравнения модели взаимопроникающих континуумов	5
1.3. Уравнения VOF-модели	6
1.3.1 Уравнения сохранения массы смеси.....	6
1.3.2 Уравнения движения	7
1.3.3 Уравнения энергии	7
2. ЧИСЛЕННАЯ РЕАЛИЗАЦИЯ VOF-АЛГОРИТМА	8
2.1. Решение уравнения для объемной доли жидкости.....	8
2.1.1 Расчет объемной доли на грани КО.....	8
2.1.2 Предиктор-шаг алгоритма CISIT.....	9
2.1.3 Корректор-шаг алгоритма CISIT.....	10
2.1.4 Граничные условия для VOF-переменной.....	12
2.1.5 Сглаживание VOF-переменной и свойств.....	13
2.1.6 Расчет кривизны межфазной поверхности.....	14
2.1.7 Баланс массы в алгоритме CISIT.....	16
2.2. VOF и алгоритм SIMPLE/PIMPLE	18
2.3. Уравнения движения	19
2.3.1 Структурный силовой источник	20
2.3.2 Неструктурный силовой источник.....	20
2.3.3 BodyForce модель силового источника.....	21
2.3.4 Пример использования силовых источников	22
2.4. Уравнение энергии	25
2.4.1 Аппроксимация конвективных членов.....	25
2.4.2 Аппроксимация диффузионных членов.....	25
2.4.3 Граничные условия на твердой стенке.....	26
2.4.4 Модель массообмена на межфазной границе	27
2.5. Верификация VOF-алгоритма	28
3. РАБОТА С VOF-АЛГОРИТМОМ ANES.....	31
3.1. Активация модели	31
3.2. Настройка численной схемы	31
3.3. Визуализация межфазной поверхности.....	32
3.4. Начальное распределение VOF-переменной.....	33

Введение

В данном документе описан алгоритм VOF, реализованный в коде Anes. Главная идея VOF-подобных алгоритмов - это использование однофазных уравнений гидродинамики для описания двухфазного потока с явно выделенной границей раздела фаз. В исходном VOF алгоритме Хирта [1] газовая фаза отсутствует. Есть только жидкость и свободная поверхность, на которой задано распределение давления. Для описания свободной поверхности на *фиксированной* сетке ячеек используется понятие объемной доли жидкой фазы ϕ (VOF-переменная Хирта). Уравнение для ϕ получается из кинематического условия движения границы вместе с жидкостью

$$\frac{d\phi}{d\tau} = \frac{\partial\phi}{\partial\tau} + \mathbf{U}\nabla\phi = 0.$$

Для описания *двухфазного* потока с выделенной границей используются аналоги VOF алгоритма Хирта. Фактически они связаны с исходным VOF только использованием объемной доли жидкости. Модели двухфазного VOF алгоритма используют уравнения гомогенной модели, дополненные уравнением для функции «раскраски» фаз - ϕ (объемной доли жидкой или газовой фаз). В принципе уравнения такой гомогенной модели можно получить двумя способами:

- 1) на основе двухфазных уравнений модели взаимопроникающих континуумов;
- 2) на основе «сшивания» однофазных уравнений в чистых фазах так, чтобы параметры фаз можно было описать одним параметром, например, гомогенная скорость \mathbf{U} связывается со скоростями фаз \mathbf{U}_f и \mathbf{U}_g соотношением

$$\mathbf{U} = \mathbf{U}_f\phi + \mathbf{U}_g(1 - \phi). \quad (1.2)$$

Ниже для получения уравнений VOF-модели будет использоваться модель взаимопроникающих континуумов.

В документе будут использоваться следующие обозначения:

- 1) в качестве индексов первой - жидкой фазы используется индекс "f";
- 2) в качестве индексов второй - газовой фазы используется индекс "g";
- 3) объемные доли фаз будут обозначаться как ϕ_k ($k = f, g$), если индекс у ϕ не указан, то это объемная доля *жидкой* фазы;
- 4) для обозначения значений на межфазной поверхности будет использоваться индекс "int".

1. Основные уравнения

1.1. Соотношения на фазовой границе

Рассмотрим участок фазовой границы, в общем случае криволинейный (рисунок 1.1), в неподвижной системе координат. Уравнения баланса массы, импульса и энергии на фазовой границе можно записать в виде [2]:

$$\begin{aligned} m_{gs} &= \rho_g (\mathbf{U}_g - \mathbf{U}_{int}) = \rho_f (\mathbf{U}_f - \mathbf{U}_{int}), \\ f_{n,gs} &= p_g + m_{gs} \mathbf{U}_g = p_f + m_{gs} \mathbf{U}_f - \sigma \cdot \mathbf{K}, \\ f_{\tau,gs} &= \mu_g \frac{\partial \mathbf{U}_g}{\partial \tau} = \mu_f \frac{\partial \mathbf{U}_f}{\partial \tau}, \\ q_{sf} &= m_{gf} h_g(T_{int}) + q_{g-int} = m_{gf} h_f(T_{int}) + q_{int-f}, \\ q_{g-int} &= [-\lambda_g (\nabla T_g \cdot \mathbf{n}_f)], \quad q_{int-f} = [-\lambda_f (\nabla T_f \cdot \mathbf{n}_f)] \end{aligned} \quad (1.1)$$

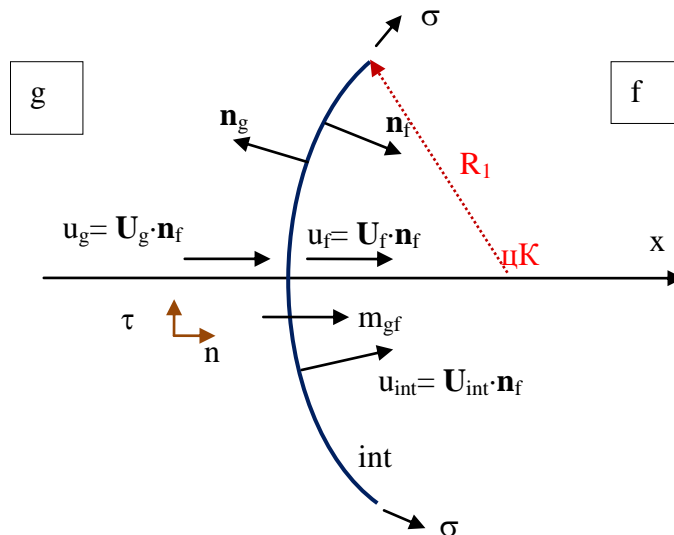


Рисунок 1.1 - Фазовая граница в фиксированной СК

Здесь:

m_{gf} - плотность массового потока ($\text{кг}/(\text{м}^2 \cdot \text{с})$), направленная из g- в f-фазу,

U_{int} - вектор скорости фазовой поверхности,

U_f, U_g - вектора скорости f- и g- фаз,

n_g, n_f - нормали к границе, направленные в сторону g- и f-фаз,

p_g, p_f - давления фаз на фазовой границе,

K - кривизна межфазной поверхности,

σ - постоянный коэффициент поверхностного натяжения.

В VOF-модели Anes в качестве нормали к межфазной границе используется нормаль, направленная в сторону f-фазы:

$$\mathbf{n}_{int} = \mathbf{n}_f,$$

и считается, что кривизна K положительна, если ее центр (цК) расположен в f-фазе. В этом случае для расчета кривизны можно использовать уравнение [4]:

$$K = -\text{div}(\mathbf{n}_f) = -\text{div}(\mathbf{n}_{int}) \quad (1.2)$$

Баланс нормальной компоненты импульса (3 уравнение (1.1)) можно переписать в виде:

$$p_f - p_g = \sigma \cdot K + m_{gs} (U_g - U_f) = \sigma \cdot K + m_{gf}^2 \left(\frac{1}{\rho_g} - \frac{1}{\rho_f} \right) \approx \sigma \cdot K \quad (1.3)$$

В VOF-модели предполагается, что теплоемкости фаз c_{pg}, c_{pf} постоянны, а температура фазовой границы равна температуре насыщения при текущем давлении:

$$h_f(T) = c_{pf} (T - T_{int}) + h_{f,sat}, \quad h_g(T) = c_{pg} (T - T_{int}) + h_{g,sat}, \quad T_{int} = T_{sat}(p), \quad (1.4)$$

$$\Delta h_{gs} = h_{g,sat} - h_{f,sat}$$

где $h_{g,sat}$ и $h_{f,sat}$ - энтальпии фаз на линии насыщения, Δh_{gf} - теплота испарения. Уравнение баланса энергии (4 уравнение (1.1)) можно переписать в другой форме:

$$m_{gf} \Delta h_{gf} = q_{int-f} - q_{g-int} = -(q_{f-int} + q_{g-int}). \quad (1.5)$$

Здесь q_{f-int}, q_{g-int} - плотности тепловых потоков за счет теплопроводности, направленные от фаз к фазовой поверхности.

1.2. Уравнения модели взаимопроникающих континуумов

В модели взаимопроникающих континуумов (МВК) предполагается, что для каждой фазы можно написать свои уравнения сохранения со своими скоростями и температурами [3]. Взаимодействие фаз описывается объемными межфазными источниками массы, импульса и энергии.

Уравнения баланса массы для фаз можно записать в виде:

$$\begin{aligned} \frac{\partial \rho_f \varphi_f}{\partial \tau} + \operatorname{div}(\rho_f \mathbf{U}_f \varphi_f) &= M_{gf} + M_{0f}, \\ \frac{\partial \rho_g \varphi_g}{\partial \tau} + \operatorname{div}(\rho_g \mathbf{U}_g \varphi_g) &= -M_{gf} + M_{0g} \end{aligned} \quad (1.6)$$

Здесь M_{gf} – межфазный *объемный* поток массы (кг/(м³·с)) из газовой фазы "g" в жидкую фазу "f" (он считается положительным, если идет процесс конденсации), M_{0k} – дополнительные источники массы фаз, приведенные к объемному виду. В коде Anes M_{0k} – это источники граничных условий на входных и выходных границах. Объемный массовый источник связан с плотностью потока массы на фазовой границе соотношением:

$$M_{gf} = m_{gf} A_{int}, \quad (1.7)$$

где A_{int} - удельная площадь межфазной поверхности (площадь межфазной поверхности в единице объема). Удельную поверхность A_{int} можно выразить через δ - функцию Дирака (строгий вывод этой формулы можно найти в [4]), а саму δ - функцию можно связать с VOF-переменной [2]:

$$A_{int} = \delta\{\mathbf{n}_{int} \cdot (\mathbf{x} - \mathbf{x}_{int})\} = |\nabla \varphi| = \sqrt{\left(\frac{\partial \varphi}{\partial x}\right)^2 + \left(\frac{\partial \varphi}{\partial y}\right)^2 + \left(\frac{\partial \varphi}{\partial z}\right)^2} \quad (1.8)$$

Уравнения баланса импульса (уравнения движения для скоростей фаз \mathbf{U}_g и \mathbf{U}_f) в МВК имеют следующий вид:

$$\begin{aligned} \frac{\partial \varphi_f \rho_f \mathbf{U}_{f,k}}{\partial \tau} + \operatorname{div}(\varphi_f \rho_f \mathbf{U}_f \mathbf{U}_{f,k} - \varphi_f \mu \nabla \mathbf{U}_{f,k}) &= -\varphi_f \frac{\partial p_f}{\partial x_k} - (p_f - p_{f,int}) \frac{\partial \varphi_f}{\partial x_k} + \varphi_f \rho_f \mathbf{g}_k + \mathbf{F}_{f-int,k}, \\ \frac{\partial \varphi_g \rho_g \mathbf{U}_{g,k}}{\partial \tau} + \operatorname{div}(\varphi_g \rho_g \mathbf{U}_g \mathbf{U}_{g,k} - \varphi_g \mu \nabla \mathbf{U}_{g,k}) &= -\varphi_g \frac{\partial p_g}{\partial x_k} - (p_g - p_{g,int}) \frac{\partial \varphi_g}{\partial x_k} + \varphi_g \rho_g \mathbf{g}_k + \mathbf{F}_{g-int,k} \end{aligned} \quad (1.9)$$

Здесь $p_{f,int}$ и $p_{g,int}$ - давления фаз на фазовой границе. В уравнениях (1.9) для простоты опущены источниковые члены, связанные с отличием закона трения Ньютона от стандартного градиентного вида. Объемные источники \mathbf{F}_{f-int} , \mathbf{F}_{g-int} связаны с взаимодействием фаз с фазовой поверхностью (трение и перенос импульса за счет массообмена). В общем случае эти члены можно записать в виде:

$$\begin{aligned} \mathbf{F}_{f-int} &= \mathbf{f}_{f-int} A_{int}, \quad \mathbf{F}_{g-int} = \mathbf{f}_{g-int} A_{int}, \\ \mathbf{f}_{f-int} &= m_{gf} \mathbf{U}_{f,int} + c_{gf} (\mathbf{U}_g - \mathbf{U}_f), \quad \mathbf{f}_{g-int} = -m_{gf} \mathbf{U}_{g,int} + c_{gf} (\mathbf{U}_f - \mathbf{U}_g), \end{aligned}$$

где c_{gf} - эмпирический коэффициент межфазного трения. Для модели VOF вид этих источников не принципиален, важно только что их сумма равна нулю.

Уравнения энергии фаз в МВК записываются аналогичным образом:

$$\begin{aligned} \frac{\partial \varphi_f \rho_f h_f}{\partial \tau} + \operatorname{div}(\varphi_f \rho_f \mathbf{U}_f h_f - \varphi_f \lambda_f \nabla T_f) &= Q_{int-f}, \\ \frac{\partial \varphi_g \rho_g h_g}{\partial \tau} + \operatorname{div}(\varphi_g \rho_g \mathbf{U}_g h_g - \varphi_g \lambda_g \nabla T_g) &= Q_{int-g}, \end{aligned} \quad (1.10)$$

$$h_f = c_{pf} (T_f - T_{int}) + h_{f,sat}, \quad h_g = c_{pg} (T_f - T_{int}) + h_{g,sat}$$

Здесь h_k – полные энтальпии фаз, $h_{k,sat}$ – энтальпии фаз на линии насыщения, T_{int} – температура межфазной поверхности, Q_{int-f} , Q_{int-g} – объемные источники тепла, описывающие приток тепла к фазе от фазовой границы:

$$\begin{aligned} Q_{int-f} &= A_{int} \left\{ m_{gf} h_f (T_{int}) + q_{int-f} \right\}, \quad Q_{int-g} = A_{int} \left\{ -m_{gf} h_g (T_{int}) + q_{int-g} \right\} \\ Q_{int-f} + Q_{int-g} &= 0 \end{aligned} \quad (1.11)$$

где q_{int-f} , q_{int-g} – плотности теплового потока от межфазной границы к фазам.

Уравнения VOF-модели можно получить, если от двухскоростной и двухтемпературной MBK перейти к гомогенной модели, в которой двухфазная среда описывается как смесь с одним вектором скорости, одним давлением и одной температурой.

1.3. Уравнения VOF-модели

В VOF-модели плотности фаз ρ_g , ρ_f и теплоемкости фаз c_{pg} , c_{pf} считаются *постоянными*. Это важные ограничения модели. Они позволяют вместо обычного уравнения сохранения массы смеси использовать уравнение, записанное через объемные потоки, а уравнение энергии записать через температуру смеси.

1.3.1 Уравнения сохранения массы смеси

Если разделить уравнения (1.6) на плотности и сложить, то можно получить уравнение неразрывности в виде:

$$\begin{aligned} \text{div}(\mathbf{U}) &= M_{gf} \left(\frac{1}{\rho_f} - \frac{1}{\rho_g} \right) + \frac{M_{of}}{\rho_f} + \frac{M_{og}}{\rho_g}, \\ \mathbf{U} &= \mathbf{U}_f \varphi + \mathbf{U}_g (1 - \varphi) \end{aligned} \quad (1.12)$$

Если просто сложить уравнения (1.6), то можно получить обычное уравнение неразрывности.

$$\begin{aligned} \frac{\partial \rho}{\partial \tau} + \text{div}(\rho \mathbf{U}) &= M_0, \\ M_0 &= M_{of} + M_{og}, \quad \rho = \rho_f \varphi + (1 - \varphi) \rho_g, \\ \mathbf{U} &= \frac{\rho_f \varphi \mathbf{U}_f + (1 - \varphi) \rho_g \mathbf{U}_g}{\rho}, \end{aligned} \quad (1.13)$$

В VOF-алгоритме используется первая - объемная форма - уравнения неразрывности (1.12).

Уравнение для объемной доли жидкой фазы φ получается из первого уравнения (1.6):

$$\frac{\partial \varphi}{\partial \tau} + \text{div}(\mathbf{U} \varphi) = \frac{M_{gf}}{\rho_f} + \frac{M_{of}}{\rho_f} \quad (1.14)$$

Следует отметить, что в уравнения неразрывности (1.12) и (1.13) входят разные скорости смеси. Однако эти скорости будут отличаться только вблизи межфазной границы. При переходе к дискретным уравнениям это означает, что скорости будут отличаться только в ячейках с межфазной границей. Эти ячейки в коде обрабатываются особым образом. Поэтому при записи уравнений будем считать, что $\mathbf{U} = \mathbf{U}_f = \mathbf{U}_g$.

1.3.2 Уравнения движения

Если сложить уравнения (1.9), то уравнение для смеси можно записать в виде ($k = x, y, z$):

$$\frac{\partial \rho U_k}{\partial \tau} + \text{div}(\rho U U_k - \mu \nabla U_k) = - \frac{\partial [\varphi p_f + (1 - \varphi) p_g]}{\partial x_k} + (p_{f,int} - p_{g,int}) \frac{\partial \varphi}{\partial x_k} + \rho g_k \quad (1.15)$$

Здесь свойства смеси рассчитывается по соотношениям

$$\rho = \rho_f \varphi + (1 - \varphi) \rho_g, \quad \mu = \mu_f \varphi + (1 - \varphi) \mu_g \quad (1.16)$$

Если положить, что давление смеси рассчитывается по среднеобъемному осреднению

$$p = p_f \varphi + (1 - \varphi) p_g, \quad (1.17)$$

а для скачка давления на фазовой поверхности использовать соотношение (1.3), то уравнения (1.15) можно записать в виде стандартных однофазных уравнений движения:

$$\frac{\partial \rho U_k}{\partial \tau} + \text{div}(\rho U U_k - \mu \nabla U_k) = - \frac{\partial p}{\partial x_k} + F_{b,k}, \quad (1.18)$$

в которых появляется объемный силовой источник:

$$\mathbf{F}_b = K \sigma \nabla \varphi + \rho \mathbf{g}$$

Этот источник содержит две составляющие: силы поверхностного натяжения на межфазной поверхности, приведенные к объемному виду, и силы плавучести. Для численного решения системы (1.18) для постоянных плотностей фаз с большими значениями ρ_f/ρ_g удобнее использовать другую форму записи сил плавучести [5]. Введем новое давление, которое представляет собой статическое давление за вычетом локального гидростатического столба:

$$p_g = p - \rho(\mathbf{g} \cdot \mathbf{x}), \quad (1.19)$$

где \mathbf{x} - радиус-вектор точки расчетной области, p - статическое давление. В этом случае силовой источник \mathbf{F}_b можно представить в следующем виде:

$$\mathbf{F}_b = \left[-(\rho_f - \rho_g)(\mathbf{g} \cdot \mathbf{x}) + \sigma K \right] \nabla \varphi \quad (1.20)$$

1.3.3 Уравнения энергии

Складывая уравнения энергии (1.10) можно получить уравнение для смеси:

$$\frac{\partial (\rho h_m)}{\partial \tau} + \text{div}(\rho U h_m - \lambda \nabla T) = 0,$$

$$h_m = \frac{\rho_f h_f \varphi + \rho_g h_g (1 - \varphi)}{\rho}, \quad \lambda = \varphi \lambda_f + (1 - \varphi) \lambda_g$$

Это уравнение содержит энтальпию смеси, и не содержит межфазных источников тепла. В VOF-алгоритме для решения используется уравнение энергии смеси, записанное через температуру. Используя выражения для энтальпии (1.10) это уравнение можно записать в следующем виде:

$$\frac{\partial (\rho c_p T)}{\partial \tau} + \text{div}(\rho c_p U T - \lambda \nabla T) = M_{gf} \left[\Delta h_{gf} + (c_{pf} - c_{pg}) T_{sat} \right], \quad (1.21)$$

$$\rho c_p = \varphi \rho_f c_{pf} + (1 - \varphi) \rho_g c_{pg}$$

Заметим, что появление в этом уравнении объемного источника тепла связано только с использованием формы записи через температуру.

Система уравнений (1.12), (1.14), (1.18) и (1.21) образует систему уравнений VOF-модели.

2. Численная реализация VOF-алгоритма

Главная идея VOF-модели заключается в способе реконструкции межфазной поверхности на основе непрерывного поля объемной доли жидкости ϕ , которая рассчитывается на основе обычного уравнения сохранения массы жидкой фазы (1.14). В исходной работе Хирта для борьбы со схемной диффузией, которая быстро размывает межфазную границу, был предложен «донор-акцептор» алгоритм, который фактически вводил специальную схемную анти-диффузию. Для реконструкции поверхности использовался простой алгоритм «граница более горизонтальна» - «граница более вертикальна». Все последующие модификации VOF-алгоритма фактически отличаются способами решения этих двух задач: борьба со схемной диффузией и реконструкции межфазной поверхности.

В коде Anes используется CISIT (Conservative Interpolation Scheme for Interface Tracking) версия VOF-алгоритма, предложенная в работе [6]. Этот алгоритм фактически является развитием алгоритма «донор-акцептор», но его оригинальный способ реконструкции границы раздела практически не зависит от размерности задачи.

2.1. Решение уравнения для объемной доли жидкости

Интегрируя уравнение (1.14) по КО, можно получить дискретное уравнение для ϕ :

$$\frac{\phi_P - \phi_P^0}{\Delta\tau} \Delta V_P = - \sum_k \phi_{f,k} V_{f,k} + V_{gf,P}^0 \Delta V_P, \quad (2.1)$$

$$V_{f,k} = U_{n,k}^* \Delta A_k = (\mathbf{U}_k^* \cdot \mathbf{n}_k) \Delta A_k, \quad V_{gf,P}^0 = \left(\frac{M_{gf}}{\rho_f} \right)_P$$

Здесь \mathbf{n}_k - внешняя нормаль k-ой грани КО "P", $V_{f,k}$ - *вытекающий* объемный поток через k-ую грань КО, остальные обозначения те же, что и в документе [7].

Для решения системы уравнений (2.1) можно использовать как явную, так и неявную схемы. В коде Anes используется *явная* схема, в которой источник V_{gf}^0 и значения VOF-переменной на гранях $\phi_{f,k}$ рассчитываются по полям с предыдущего шага по времени ($\tau - \Delta\tau$), а значения объемных потоков $V_{f,k}$ на гранях рассчитываются по скоростям, рассчитанным на предыдущей итерации текущего шага по времени \mathbf{U}^* .

Явная схема накладывает ограничение на используемый шаг по времени: число Куранта

$$Co = \max \left[\frac{\Delta\tau \sum_k \max(V_{n,k}, 0)}{\Delta V_P} \right] < 1. \quad (2.2)$$

Как показали тестовые расчеты, наилучшие результаты дает $Co = 0.5$. Заметим, что условие (2.2) является естественным для задач с поверхностью раздела: граница на шаге по времени не должна «перескакивать» через ячейку при переходе из одной ячейки в другую.

❗ Описание в проекте прикладной задачи :

Предельное значение числа Куранта задается оператором [Special Data]:

R("CriCourant") = 0.5

Если оператор не указан, то по умолчанию используется $Co = 0.5$.

2.1.1 Расчет объемной доли на грани КО

Для замыкания уравнений (2.1) необходимо рассчитать значения объемной доли на грани $\phi_{f,k}$. В алгоритме CISIT для этого используется «простой» алгоритм реконструкции

межфазной поверхности в ячейке: $\varphi_{f,k}$ равен площади грани, занятой f-фазой. Для расчета этой площади используется подход, позаимствованный из алгоритмов компьютерной графики: межфазная поверхность - это поверхность постоянного уровня функции с $\varphi = 0.5$.

Ячейка в общем случае представляет собой объем, ограниченный замкнутым набором плоских граней. Ячейка и грани описываются набором точек - вершин. Если значения φ в вершинах известны, то площадь грани, занятой f-фазой можно рассчитать с помощью следующего алгоритма:

- 1) перебираем все ребра грани (ребро – это отрезок, соединяющий две вершины) и находим на ребре точки с $\varphi = 0.5$; эти точки либо существуют на ребре, либо их нет;
- 2) просматриваем последовательно все вершины грани и формируем точки многоугольника с $\varphi = 0.5$; если вершина имеет значение $\varphi > 0.5$, добавляем ее в многоугольник; если между двумя вершинами есть точка с $\varphi = 0.5$, то добавляем и ее;
- 3) рассчитываем площадь полученного многоугольника;
- 4) делим площадь на площадь грани и получаем «точную» долю жидкой фазы на грани $\varphi_{f,k}$.

Для расчета значений в вершинах ячейки в коде используется алгоритм обратного - объемного осреднения:

$$\varphi_n = \frac{\sum_m \frac{\varphi_{c,m}}{\Delta V_{c,m}}}{\sum_m \frac{1}{\Delta V_{c,m}}} \quad (2.3)$$

Здесь n - это индекс вершины, «с,м» – индекс ячейки-соседа вершины, а суммирование проводится по всем ячейкам, которые содержат эту вершину.

Заметим, что этот алгоритм можно использовать и для восстановления межфазной поверхности в виде набора ее многоугольников в ячейках.

Ниже значения на грани, рассчитанные по такому «геометрическому» алгоритму, будут обозначаться как $\varphi_{s,k}$.

① Описание в проекте прикладной задачи :

В текущей версии VOF-алгоритма можно визуализировать поверхность раздела, построенную по алгоритму, описанному выше. Для этого необходимо в секцию [Special Data] добавить логический оператор `L("IsSurfVOF") = .True.`

В этом случае будут созданы vtk-файлы с именем (XX – номер шага по времени)

<префикс результатов>_VOFS_XX.vtk,

содержащие поверхность межфазной границы в формате VTK. Эту поверхность можно просмотреть в пост-процессоре Anes и ParaView.

2.1.2 Предиктор-шаг алгоритма CISIT

Схема CISIT на шаге по времени разбивается на два шага: предиктор и корректор. На предиктор-шаге поле φ на новом шаге по времени рассчитывается с использованием явной формы уравнения (2.1):

$$\varphi_P = \varphi_P^0 + \frac{\Delta \tau}{\Delta V_P} \left[-\sum_k \varphi_{s,k}^0 V_{n,k} + V_{gf}^0 \Delta V_P \right] \quad (2.4)$$

где $\varphi_{s,k}^0$ - значения на гранях, рассчитанные с помощью описанного выше алгоритма по полям с предыдущего шага по времени.

На предиктор-шаге могут возникнуть четыре ситуации, связанные с переходом фазовой поверхности из одной ячейки в другую и выходом значения φ_p из интервала $0 \dots 1$. Таких ситуаций четыре:

- 1) *Overfilling* – перелив ячейки ($\varphi_p > 1$).
- 2) *Overdepleting* – переочистка ячейки ($\varphi_p < 0$).
- 3) *Underfilling* – недолив ячейки без фронта ($\varphi_p < 1$).
- 4) *Underdepleting* – недоочистка ячейки без фронта ($\varphi_p > 0$).

Причина появления подобных ситуаций связана с алгоритмом расчета φ на грани. Рассмотрим одномерную задачу без массообмена (рисунок 2.1) и ячейку "P" с фронтом. Если сетка по оси x равномерная, то значение на грани в одномерной задаче рассчитывается по соотношению:

$$\varphi_{s,e}^0 = \begin{cases} 0, & \frac{\varphi_P^0 + \varphi_E^0}{2} < 0.5 \\ 1, & \frac{\varphi_P^0 + \varphi_E^0}{2} \geq 0.5 \end{cases}$$

Поскольку в одномерном случае скорость потока постоянна и равна U_0 (см. уравнение (1.12)), то значение на новом шаге по времени равно:

$$\varphi_P = \varphi_P^0 + \frac{\Delta\tau}{\Delta x_P} [\varphi_{s,w}^0 U_0 - \varphi_{s,e}^0 U_0] = \varphi_P^0 + \frac{\Delta\tau U_0}{\Delta x_P} = \varphi_P^0 + Co_P$$

Из этого уравнения следует, что при числе Куранта $Co_P > 1 - \varphi_P^0$ значение в ячейке станет больше единицы.

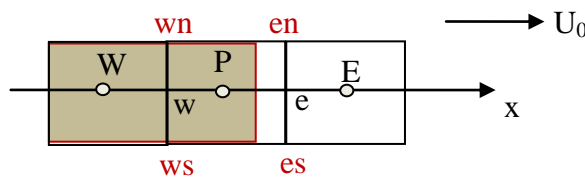


Рисунок 2.1 - Ячейки одномерной задачи

2.1.3 Корректор-шаг алгоритма CISIT

В коде Apes используется модифицированный алгоритм CISIT. Главная модификация касается расчета новых значений объемных долей на гранях, необходимых для удовлетворения дискретного уравнения (2.1) на новом шаге по времени.

Overfilling – перелив ячейки

Этот случай возникает, когда в ячейке значение $\varphi_p > 1$ после предиктор-шага. Для исправления этой «ошибки» (связанной с «неправильностью» $\varphi_{s,k}$ на гранях с *вытекающими* объемными потоками) нужно просто перераспределить излишек φ_p в ячейки вниз по потоку, а в КО "P" положить $\varphi_p = 1$. В оригинальном алгоритме CISIT это делается из условия баланса массы и предположения, что масса распределяется пропорционально вытекающим потокам в ячейки-соседи (nb - индекс ячейки-соседа):

$$\varphi_{nb} = \varphi_{nb} + (\varphi_P - 1) \cdot w_{nb} \cdot \frac{\Delta V_P}{\Delta V_{nb}}, \quad w_{nb} = \frac{\max(0, V_{n,nb})}{\sum_{nb} \max(0, V_{n,nb})} \quad (2.5)$$

После коррекции значение в ячейке устанавливается равной $\varphi_p = 1$.

Это соотношение можно получить из других соображений. Изменим значения φ на «вытекающих» гранях в уравнении (2.4) так, чтобы на шаге по времени значение в ячейке на новом шаге было равно единице:

$$\frac{1-\varphi_p^0}{\Delta\tau} \Delta V_p = \sum_k \varphi_{s,k}^0 \max(0, -V_{n,k}) - \sum_k (\varphi_{s,k}^0 + \delta\varphi_k) \max(0, V_{n,k}) + V_{gf,p}^0 \Delta V_p$$

Если вычесть это уравнение из (2.4), то можно получить уравнение для коррекции объемной доли на гранях:

$$\frac{1-\varphi_p}{\Delta\tau} \Delta V_p = -\sum_k \max(0, V_{n,k}) \cdot \delta\varphi_k$$

Если предположить, что коррекция на гранях одинакова, то (φ_p^1 -значение в ячейке, полученное на предиктор-шаге):

$$\delta\varphi = \frac{\varphi_p^1 - 1}{\Delta\tau} \frac{\Delta V_p}{\sum_k \max(0, V_{n,k})}, \quad \varphi_{t,k} = \varphi_{s,k}^0 + \delta\varphi \quad (2.6)$$

Хорошо видно, что из (2.6) следует «перераспределительный» алгоритм оригинального CISIT (2.5). Новые значения на грани $\varphi_{t,k}$ позволяют включить корректор-шаг в уравнения (2.4) предиктор-шага и ввести ограничение в соотношение (2.5) - перелив необходимо осуществлять только в неполные ячейки с $\varphi_{nb} < 1$.

Overdepleting – переочистка ячейки

В этой ситуации в ячейке "P" после первого шага значение φ_p становится отрицательной. Решение этой проблемы реализуется по аналогии с переливом. Единственное отличие заключается в том, что в ячейки вниз по потоку переносятся отрицательные значения φ_p :

$$\varphi_{nb} = \varphi_{nb} + \varphi_p w_{nb} \frac{\Delta V_p}{\Delta V_{nb}} \quad (2.7)$$

После коррекции значение в ячейке устанавливается равной $\varphi_p = 0$. Одновременно с коррекцией значений в ячейках производится и коррекция значений на гранях $\varphi_{t,k}$, аналогично первому случаю (в этом случае значения на гранях уменьшаются). При применении соотношений (2.7) в них не включаются ячейки вниз по потоку, в которых $\varphi_{nb} = 0$ (чтобы их не переочищать).

Underfilling – недолив в ячейку

Третий и четвертый шаги связаны с технологией реконструкции поверхности раздела, основанной на использовании значений φ в вершинах ячеек и понятии поверхности раздела как поверхности с $\varphi = 0.5$.

Третий шаг связан с ситуацией, когда в ячейке $\varphi_p < 1$, а значения φ , рассчитанные во всех вершинах ячейки $\varphi_{n,k} > 0.5$. Последнее означает, что на следующем шаге по времени значения $\varphi_{f,k}^0$ на всех гранях будут равны единице. Это в свою очередь означает, что из этой ячейки граница раздела уже ушла. Поскольку при решении уравнения неразрывности (1.12) будет выполнен баланс объемных потоков, то значение φ_p больше меняться не будет.

Для решения этой проблемы нужно дозаполнить ячейку до 1. Для этого используется алгоритм первого шага (2.5). При этом в отличие от первого шага величина $(\varphi_p - 1)$ будет отрицательной, и мы фактически будем доливать ячейку из соседей по потоку.

***Underdepleting* – недоочистка ячейки**

Четвертый шаг аналогичен третьему и связан с ситуацией, когда в ячейке $\varphi_p > 0$, а значения φ , рассчитанные во всех вершинах ячейки $\varphi_{n,k} < 0.5$. Последнее означает, что на следующем шаге по времени значения $\varphi_{f,k}^0$ на всех гранях станет равным нулю и граница из этой ячейки ушла.

Для решения этой проблемы используется скорректированный алгоритм второго шага (2.7). При переборе ячеек-соседей проверяется заполненность ($\varphi = 1$) ячеек *вверх* по потоку (соседи-доноры). Если такие ячейки есть, то коррекция не выполняется. Причина очень простая – на следующем шаге по времени донорские ячейки будут перелиты и данная ячейка будет дозаполнена первым шагом. В противном случае этот алгоритм вызывает появление большого числа итераций корректор-шага.

Коррекция границ изменения VOF-переменной

В коде Anes в корректор шаг добавляется еще один непринципиальный шаг: коррекция маленьких значений и значений, близких к единице:

- 1) если в ячейке $\varphi_p < 10^{-5}$ и значение во всех вершинах $\varphi_{n,k} < 0.5$, то без всяких перераспределений мы полагаем $\varphi_p = 0$;
- 2) если в ячейке $1 < \varphi_p < 1.001$ и значение во всех вершинах $\varphi_{n,k} > 0.5$, то без всяких перераспределений мы полагаем $\varphi_p = 1$.

Это позволяет уменьшить число «бессмысленных» итераций для первого и второго шага, когда последовательно на границы РО будут переноситься маленькие значения.

Алгоритм корректор-шага

Корректор-шаг выполняется в виде внутреннего итерационного цикла внутри решения дискретных уравнений для VOF-переменной:

- 1) на итерации последовательно выполняются указанные пять шагов;
- 2) после выполнения первых двух шагов производится пересчет значений φ в вершинах всех ячеек;
- 3) итерационный процесс прекращается, если ячейки для коррекции отсутствуют.
- 4) на каждом шаге независимо перебираются все ячейки (P2P алгоритм) и обрабатываются алгоритмами описанными выше.

Как показали расчеты, для корректор-шага число внутренних итераций не превышает 2 (в худшем случае 3-7 итераций).

❶ Реализация в Anes

При проведении расчета в консоль Решателя выводится информация о количестве внутренних итераций корректор-шага и безразмерные значения интегральных безразмерных балансов объемов и массы на каждом шаге по времени:

VOF: NoCorIter = 2 TotVol = 0.00E+00 TotMass = 1.22E-02

2.1.4 Граничные условия для VOF-переменной

Поскольку в уравнении для VOF-переменной нет диффузии, то граничные условия необходимо задавать только на входной границе. На непроницаемых стенках и выходных границах используются граничные условия по умолчанию. На входной границе необходимо использовать граничное условие BC_MASSONLY (см. раздел 3.5.5 документа [8]).

2.1.5 Сглаживание VOF-переменной и свойств

Алгоритм CISIT приводит к тому, что область с фронтом ($0 < \varphi < 1$) на новом шаге по времени занимает один слой ячеек. Фактически это означает, что все свойства смеси являются «разрывными». Далее такое распределение объемной доли будет называться точным распределением φ . Для расчета массовых потоков на гранях ячеек и источников членов могут понадобиться более гладкие распределения φ . Для этого в VOF-алгоритмах используются различные алгоритмы сглаживания [4].

В коде Anes используется циклический алгоритм, основанный на циклическом перерасчете значений в центре ячейке и в ее вершинах [6]. В цикле этого алгоритма:

- 1) по соотношениям (2.3) рассчитываются значения в вершинах ячеек:

$$\varphi_n = \frac{\sum_m \frac{\varphi_{c,m}}{\Delta V_{c,m}}}{\sum_m \frac{1}{\Delta V_{c,m}}} \quad (2.8)$$

- 2) по этим значениям в вершинах рассчитываются новые сглаженные значения в центрах КО

$$\tilde{\varphi}_p = \frac{\sum_{iV=1}^{N_{\text{Vert}}} \Phi_{iV}}{N_{\text{Vert}}} \quad (2.9)$$

- 3) шаги (1) – (2) повторяются NoSmooth раз.

На рисунке 2.2 показан эффект сглаживания VOF- переменной.

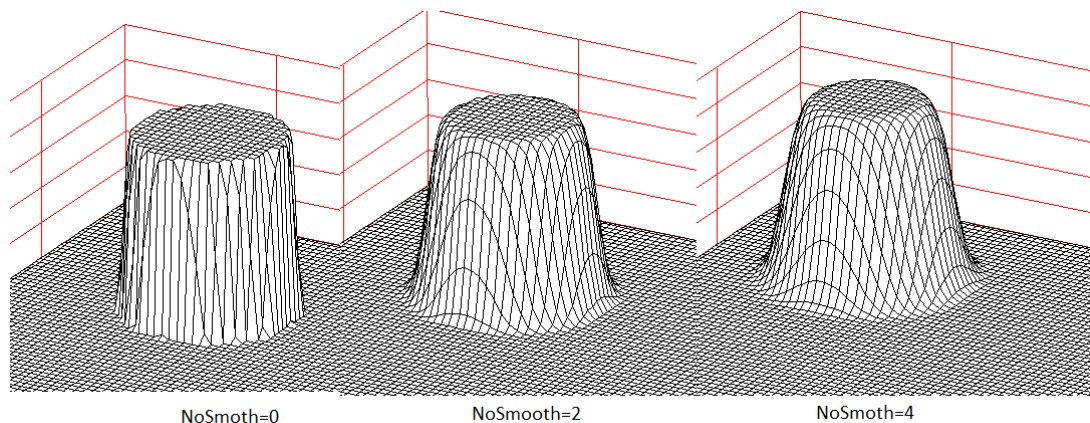


Рисунок 2.2 – Сглаживание VOF-функции

Отметим важное свойство такого алгоритма сглаживания. Алгоритм CISIT обладает свойством консервативности в том смысле, что он сохраняет объем жидкой фазы. Если нет массового источника и границы расчетной области (РО) непроницаемые, то сохраняется интеграл:

$$\int_{\text{PO}} \varphi dV = \text{const} \quad (2.10)$$

Как показывают тестовые расчеты, аналогичный интеграл от сглаженной доли $\tilde{\varphi}$ так же сохраняется, но приближенно (в пределах 1 - 2%).

В коде Anes для расчета свойств смеси в *центрах ячеек* на текущем шаге по времени используются соотношения:

$$\begin{aligned} \rho &= \rho_f \phi + (1 - \phi) \rho_g, \quad \mu = \mu_f \phi + (1 - \phi) \mu_g, \quad \lambda = \lambda_f \phi + (1 - \phi) \lambda_g, \\ \rho c_p &= \phi \rho_f c_{pf} + (1 - \phi) \rho_g c_{pg} \end{aligned} \quad (2.11)$$

и два поля: точное поле ϕ и сглаженное поле $\tilde{\phi}$. Выбор поля определяется прикладной задачей. Лучше всего использовать точные свойства, рассчитанные по точному полю. В некоторых задачах использование точных свойств приводит к появлению нефизических колебаний межфазной поверхности. В этом случае необходимо перейти на использование сглаженных свойств.

При использовании VOF-алгоритма смесь рассматривается как однофазная G-фаза когда Anes. Для расчета свойств смеси задаются отдельно свойства f-фаз и g-фаз VOF-модели. Напомним, что эти свойства должны быть постоянными.

Описание в проекте прикладной задачи :

Число циклов сглаживания NoSmooth задается оператором секции [Special Data] :

I("NoSMO_VOF") = 2

По умолчанию используется значение NoSmooth = 2.

Если оператор той же секции

L("IsSMO_VOFPROP") = .FALSE.

то для расчета свойств в центрах КО используется точное поле VOF-переменной, если значение оператора .TRUE., то используется сглаженное поле VOF-переменной.

Свойства f- и g- фаз задаются операторами секции [Properties]. Для этого используется постфиксы с именами FLU и GAS:

PROP("<Имя G-материала PO>.Dens.FLU") = <плотность f-фазы>

PROP("<Имя G-материала PO>.Dens.GAS") = <плотность g-фазы>

PROP("<Имя G-материала PO>.Visc.FLU") = <кинематическая вязкость f-фазы>

PROP("<Имя G-материала PO>. Visc.GAS") = < кинематическая вязкость g-фазы>

PROP("<Имя G-материала PO>. SigmsFG") = <коэффициент поверхностного натяжения>

PROP("<Имя G-материала PO>.Cp.FLU") = <теплоемкость f-фазы>

PROP("<Имя G-материала PO>. Cp.GAS") = < теплоемкость g-фазы>

PROP("<Имя G-материала PO>.Cond.FLU") = <теплопроводность f-фазы>

PROP("<Имя G-материала PO>. Cond.GAS") = < теплопроводность g-фазы>

2.1.6 Расчет кривизны межфазной поверхности

Для расчета кривизны межфазной поверхности K используется соотношение (1.2) и сглаженное поле объемной доли жидкой фазы:

$$K = -\text{div}(\mathbf{n}_{\text{int}}), \quad \mathbf{n}_{\text{int}} = \frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|} \quad (2.12)$$

На дискретном уровне используется теорема Гаусса-Остроградского:

$$K_P = -\frac{1}{\Delta V_P} \sum_k \mathbf{n}_{\text{int},k} \Delta A_k, \quad \mathbf{n}_{\text{int},k} = (\mathbf{n}_{\text{int},k} \cdot \mathbf{n}_k), \quad (2.13)$$

$$\mathbf{n}_{\text{int},k} = \left(\frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|} \right)_P \cdot f_P + \left(\frac{\nabla \tilde{\phi}}{|\nabla \tilde{\phi}|} \right)_{\text{nb}} \cdot (1 - f_P)$$

Значение нормали $\mathbf{n}_{\text{int},k}$ на грани рассчитывается с помощью линейной интерполяции (см. раздел 4.2 документа [7]):

Уравнение (2.13) позволяет легко включить эффекты смачивания на границе раздела трех фаз: «жидкость» - «пар» - «твердая стенка».

При решении уравнений (2.4) значение φ на граничной грани - твердой стенке не используется, поскольку объемный поток на такой грани равен нулю. Значения на граничной грани могут быть любые, по умолчанию в коде Anes используется «адиабатическое» условие:

$$\frac{\partial \varphi}{\partial n} = (\nabla \varphi \cdot \mathbf{n}_w) = 0, \quad (2.14)$$

где \mathbf{n}_w - нормаль к стенке, направленная из расчетной области (см. рисунок 2.3). Это означает, что вектор нормали \mathbf{n}_{int} направлен параллельно поверхности стенки, что в свою очередь означает отсутствие смачиваемости (точнее говоря «нейтральную» смачиваемость).

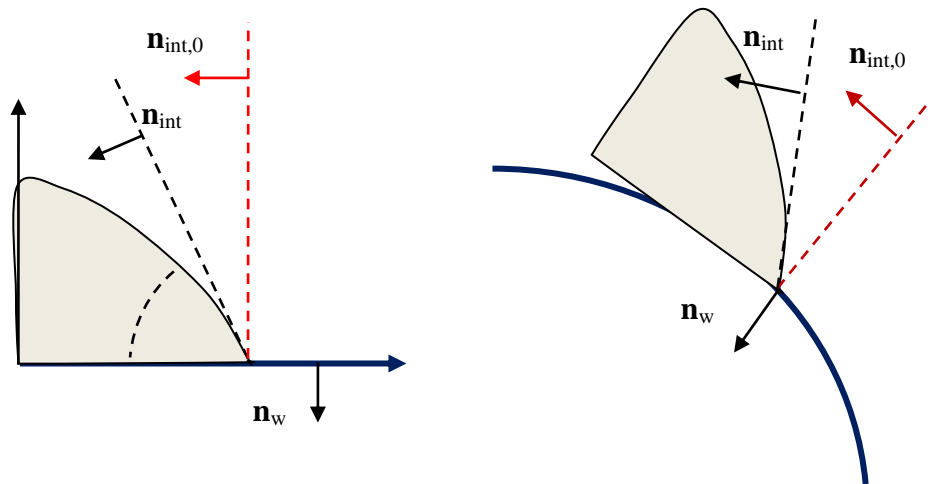


Рисунок 2.3 - Учет эффектов смачивания

Эффект смачивания описывается краевым углом смачивания θ_c - углом между плоскостью стенки и плоскостью, касательной к межфазной поверхности, отсчитываемый от поверхности стенки в жидкой фазе. Плохая смачиваемость соответствует углам смачивания больше 90° , хорошая - углам смачивания меньше 90° , отсутствие смачиваемости - $\theta_c = 90^\circ$.

Для учета смачиваемости необходимо подменить «параллельную» нормаль к фазовой поверхности в соотношении (2.13). Если задача двумерная и стенка горизонтальная (рисунок 2.3), то положение вектора нормали *на стенке* \mathbf{n}_{int} можно получить из геометрических соображений:

$$\mathbf{n}_{int} = (n_{int,x}, n_{int,y}) = [-\sin(\theta_c), -\cos(\theta_c)],$$

которое легко обобщается на произвольную геометрию:

$$\mathbf{n}_{int} = \mathbf{n}_w \cos(\theta_c) + \mathbf{n}_{int,0} \sin(\theta_c) \quad (2.15)$$

Здесь $\mathbf{n}_{int,0}$ - вектор нормали, полученный из соотношения (2.14).

Описание в проекте прикладной задачи :

Краевой угол смачивания для патча типа Wall задается с помощью оператора секции [Special Data] :

R("<Имя патча>.ContactAngle") = 90

Значение угла задается в градусах.

2.1.7 Баланс массы в алгоритме CISIT

Как уже отмечалось, алгоритм CISIT является консервативным и выполняет закон сохранения массы жидкой фазы на каждом шаге по времени:

$$\int_{\text{PO}} \varphi dV = \text{const} \quad (2.16)$$

Однако для получения дискретных уравнений движения и энергии необходимо выполнение более «строгого» баланса массы или объема жидкой фазы. Это связано со способом получения дискретных уравнений переноса в коде Anes (см. раздел 3.3 документа [7]).

При выводе уравнений в диагональном члене a_p появляется дискретный аналог уравнения неразрывности D_p (см. уравнение (3.23) документа [7]):

$$D_p = \left\{ \frac{\rho_p \varphi_p - \rho_p^0 \varphi_p^0}{\Delta \tau} \Delta V_p + \sum_k m_{n,k} \varphi_k \Delta A_c \right\} \quad (2.17)$$

При использовании алгоритма SIMPLE на каждой итерации выполняется условие $D_p = 0$, поэтому член D_p удаляется из диагонального члена дискретных уравнений кода Anes для любых Φ -переменных.

При использовании VOF-алгоритма в качестве уравнения неразрывности используется *объемная форма* этого уравнения (1.12), поэтому алгоритм SIMPLE на каждой итерации выполняет другое дискретное уравнение:

$$\sum_k V_{n,k} \Delta A_k - \left[M_{gf} \left(\frac{1}{\rho_f} - \frac{1}{\rho_g} \right) \right]_p \Delta V_p = 0, \quad (2.18)$$

а член D_p , который удаляется из дискретного уравнения, равен:

$$D_p = (\rho_f - \rho_g) \left\{ \frac{\varphi_p - \varphi_p^0}{\Delta \tau} \Delta V_p + \sum_k V_{n,k} \varphi_k \Delta A_k - V_{gf,p} \Delta V_p \right\} \quad (2.19)$$

Резонно возникает вопрос - какие значения объемной доли в центре КО и на гранях использовать, чтобы свести значение D_p к нулю? Здесь возможны следующие варианты:

- 1) VOF_face_smo: для значений в центрах ячеек используется *сглаженное* поле $\tilde{\varphi}$ на текущем шаге по времени, а значение на гранях рассчитывается с помощью линейной интерполяции этого сглаженного поля;
- 2) VOF_face_lin: для значений в центрах ячеек используется *точное* поле φ на текущем шаге по времени, а значение на гранях рассчитывается с помощью линейной интерполяции точного поля;
- 3) VOF_face_tru: для значений в центрах ячеек используется *точное* поле φ на текущем шаге по времени, а в качестве значений на грани используются значения $\varphi_{t,k}$, полученные на корректор-шаге CISIT (2.6).

На рисунках 2.4 - 2.7 показаны безразмерные поля объемного уравнения неразрывности (2.18) и члена D_p для трех вариантов расчета членов (2.19). В качестве масштаба обезразмеривания используется среднее значение модулей объемных потоков на всех гранях текущей ячейки. В качестве примера рассмотрена сложная задача о пленочном кипении.

Как следует из рисунков, минимальные значения D_p имеет модель VOF_face_tru. Немногие пики на рисунке связаны с трудностями включения шагов Underfilling и UnderDepleting в алгоритм расчета поправки φ на грани на корректор-шаге (2.6). В остальных ячейках (включая ячейки с межфазной границей) невязка D_p уменьшилась на порядки по сравнению с вариантами VOF_face_smo и VOF_face_lin.

В коде Anes для уравнений движения реализованы все три варианта. Вариант VOF_face_smo используется, если используются сглаженные свойства (флаг isSMO_VOFPROP = .TRUE.).

Если используются точные свойства, то выбор модели VOF_face_lin или VOF_face_tru определяется оператором VofForFMassID файла проекта. Для уравнения энергии всегда используется «точная» модель VOF_face_tru.

Многочисленные тестовые результаты показали следующее:

1. Несмотря на большие значения D_p для моделей VOF_face_smo и VOF_face_lin в ячейках с межфазной границей, поля скоростей, давления и положение границы раздела практически одинаковы для всех трех моделей.

2. При использовании моделей VOF_face_lin и VOF_face_tru в ряде задач возникают нефизичные колебания поверхности раздела фаз. Причина этого, по-видимому, связана с использованием «размазанной» кривизны с точным распределением ϕ .

В связи с этим для решения уравнений движения можно использовать любой вариант. По умолчанию в коде используется модель VOF_face_lin.

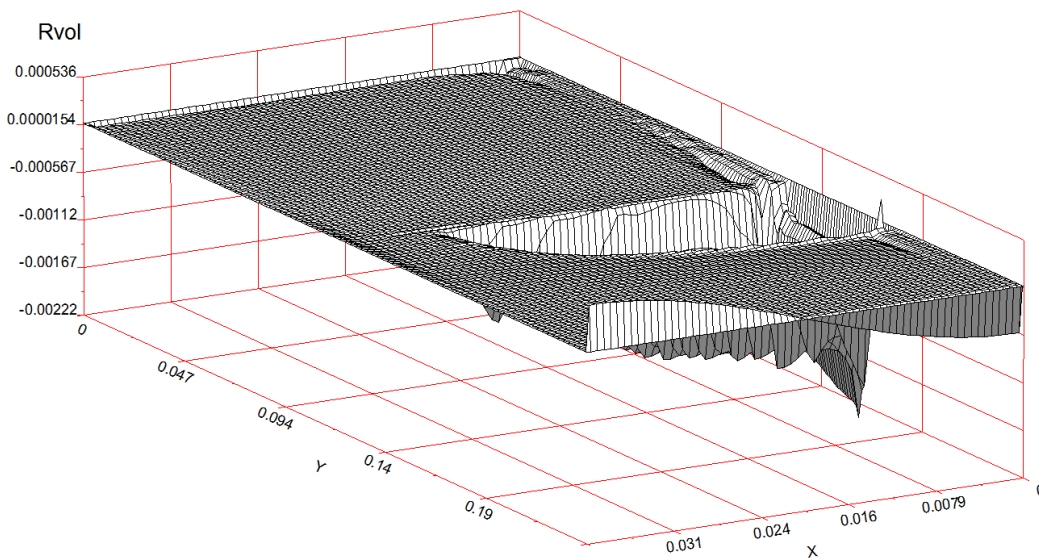


Рисунок 2.4 - Невязка объемного уравнения неразрывности (2.18)

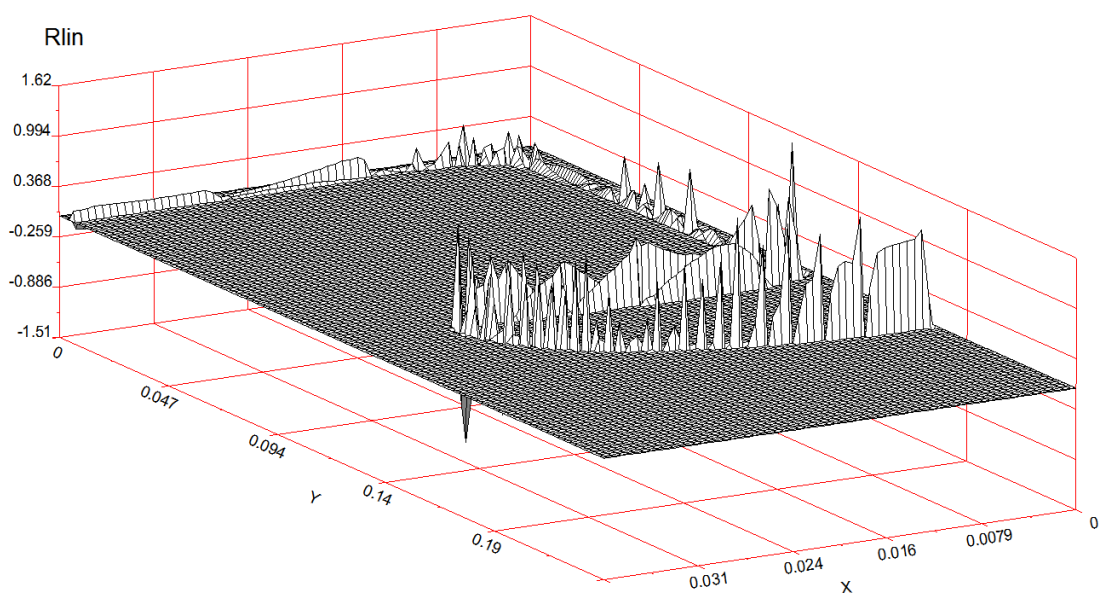


Рисунок 2.5 - Невязка уравнения баланса (2.19) для модели VOF_face_lin

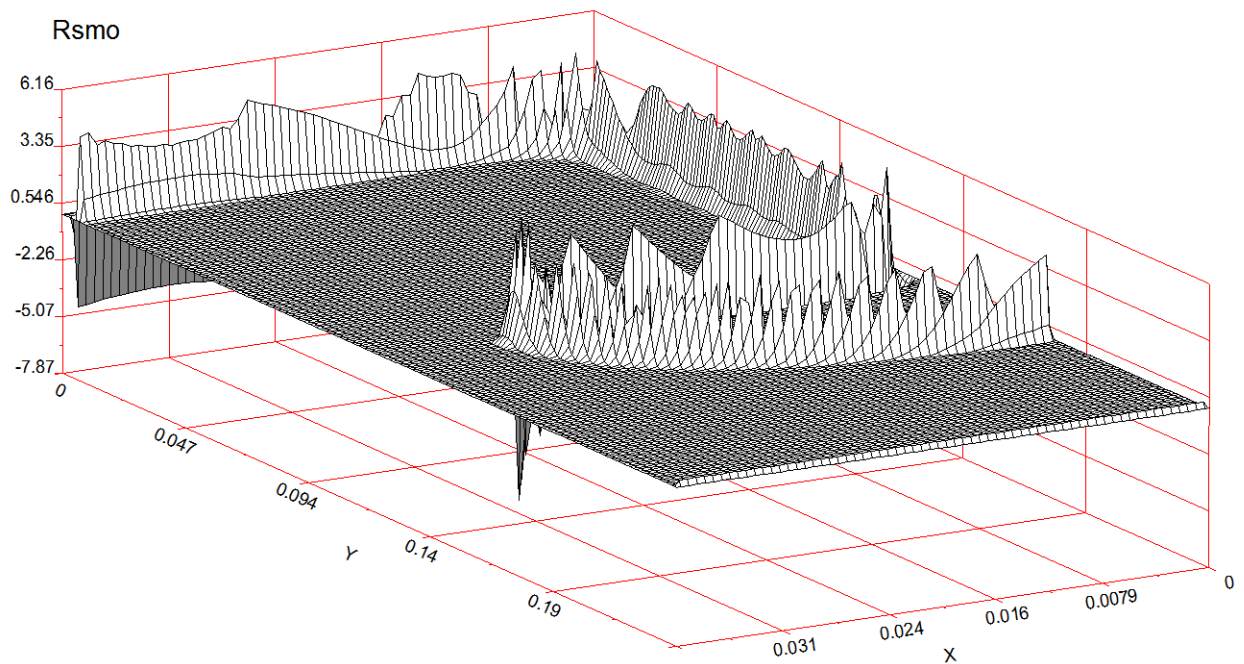


Рисунок 2.6 - Невязка уравнения баланса (2.19) для модели VOF_face_smo

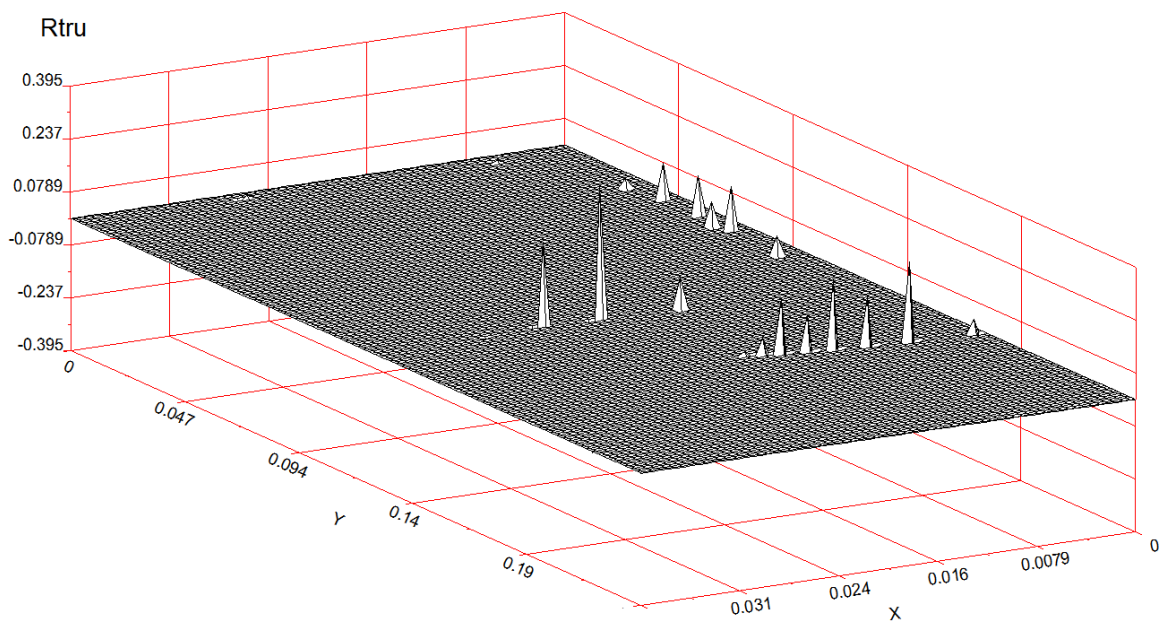


Рисунок 2.7 - Невязка уравнения баланса (2.19) для модели VOF_face_tru

Описание в проекте прикладной задачи :

Выбор модели VOF_face_lin и VOF_face_tru определяется оператором секции [Special Data] :
 C("VofForFMassID ") = LIN / FracTRU

2.2. VOF и алгоритм SIMPLE/PIMPLE

Уравнение для VOF-переменной (раздел 2.1) включается в итерационный SWEEP-цикл шага по времени как показано на рисунке 2.8. Поскольку в явное уравнение (2.1) входит скорость U^* , с предыдущей итерации, то решение для VOF-переменной необходи-

мо включить в итерационный SWEEP-цикл (красная линия, алгоритм "Explicit"). Как показали тестовые расчеты задач без массообмена, для большинства задач можно исключить решение VOF-уравнения из SWEEP-цикла и решать его только на первой итерации. Этот алгоритм в коде называется "OneSWEEP" (синяя линия). Использование этого алгоритма позволяет сократить время расчета одного шага на 20 - 30 %. Выбор того или иного варианта можно осуществить, проведя предварительные расчеты.

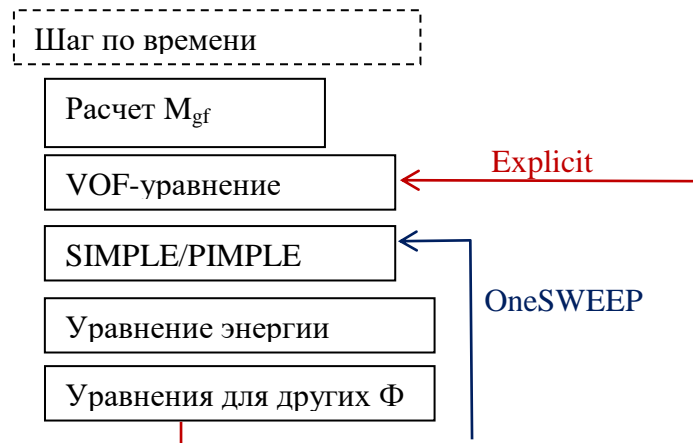


Рисунок 2.8 - Расчет шага по времени

Для решения VOF-задач можно использовать как алгоритм SIMPLE, так и алгоритм PIMPLE (SIMPLE + PISO). Для большинства задач рекомендуется использовать алгоритм PIMPLE, который позволяет уменьшить время расчета в несколько раз по сравнению с SIMPLE (PIMPLE в первую очередь и был реализован для решения VOF-задач (см. раздел 6.2 документа [7])).

При моделировании массообмена на межфазной поверхности дополнительно решается уравнение энергии. Его решение включается стандартным способом в SWEEP-итерации PIMPLE. При этом расчет межфазного массового источника рассчитывается на первой итерации по полям с предыдущего шага по времени.

Описание в проекте прикладной задачи :

Выбор алгоритма OneSWEEP или Explicit определяется оператором секции [Special Data] :

```
C("SweepVOFModelID") = "OneSweep" / "Explicit"
```

Если уравнение энергии не решается, то по умолчанию используется модель OneSWEEP. При решении уравнения энергии по умолчанию используется модель Explicit.

2.3. Уравнения движения

Дискретные уравнения движения получаются из системы (1.18) с использованием стандартных алгоритмов G-фазы кода Anes (см. разделы 3 и 4 документа [7]), особым образом обрабатываются только силовые источники (1.20):

$$\mathbf{F}_b = \left[-(\rho_f - \rho_g)(\mathbf{g} \cdot \mathbf{x}) + \sigma \mathbf{K} \right] \nabla \varphi \quad (2.20)$$

Как уже отмечалось, кривизна межфазной поверхности \mathbf{K} всегда рассчитывается по *сглаженному* полю φ . Для расчета градиента можно использовать как *сглаженную*, так и *точную* объемную долю. В коде Anes реализованы оба подхода, их активация определяется флагом `IsSMO_VOFPROP`.

Если используется модель *сглаженных* свойств (`IsSMO_VOFPROP = .TRUE.`), то градиент рассчитывается по *сглаженному* полю $\tilde{\varphi}$. Эта постановка соответствует CSF (Continuum Surface Force) модели, предложенной Бракбиллом [4]. Заметим, что эта модель используется практически во всех реализованных VOF-алгоритмах.

Если используются точные свойства (`IsSMO_VOFPROP = .FALSE.`), то для расчета градиента используется точное поле ϕ . Тестовые расчеты показали, что в большинстве задач она вполне работоспособна. Однако в некоторых задачах возникают нефизичные возмущения межфазной поверхности. По-видимому, причиной этого эффекта является использование сглаженной кривизны и точного градиента. Сглаженная модель CSF также имеет свои «недостатки». В частности, при моделировании течений с существенным влиянием поверхностного натяжения на границе раздела возникают паразитные вихри, которых быть не должно.

2.3.1 Структурный силовой источник

При работе со структурными сетками используется модель сдвинутых скоростей (шахматные сетки). В этой модели компоненты скорости определены на гранях КО, а сам контрольный объем для компоненты скорости состоит из двух «половинок» основных КО (см. рисунок 2.9).

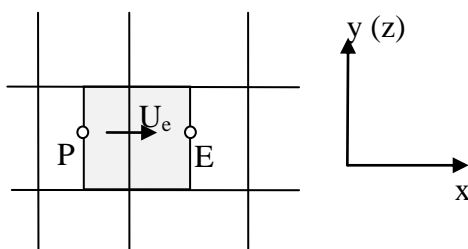


Рисунок 2.9 - КО для компоненты скорости U_x

Силовой источник для скорости U_e в коде записывается следующим образом:

$$F_e \Delta V_e = (C_{g,e} + C_{k,e}) \Delta A_e (\phi_E - \phi_P), \quad \Delta V_e = \frac{\Delta V_P + \Delta V_E}{2}, \quad (2.19)$$

$$C_{g,e} = -(\rho_f - \rho_g)(\mathbf{x}_e \cdot \mathbf{g}), \quad C_{k,e} = K_e \cdot \sigma, \quad K_e = K_P f_P + K_E (1 - f_P)$$

Здесь \mathbf{x}_e - радиус-вектор центра e-границы, K_e - значение кривизны межфазной поверхности на границе "e", рассчитанное с помощью линейной интерполяции.

При использовании сглаженных свойств значения ϕ_P и ϕ_E рассчитываются по сглаженному полю $\tilde{\phi}$. При использовании точных свойств значения ϕ_P и ϕ_E рассчитываются по точному полю ϕ . При этом для расчета кривизны всегда используется сглаженное поле.

2.3.2 Неструктурный силовой источник

При работе с неструктурными сетками используется модель совмещенных скоростей, в которой компоненты скорости заданы в центрах КО. Для расчета силового источника реализованы два алгоритма: стандартный алгоритм расчета источника в ячейке и алгоритм, основанный на использовании BodyForce модели (см. раздел 4.8 документа [7]).

Выбор алгоритма определяется оператором секции [Special Data] файла проекта:
`L("IsBodyForceVOF") = .TRUE.`

По умолчанию в коде используется BodyForce алгоритм, описанный в следующем разделе.

При использовании стандартного алгоритма (`IsBodyForceVOF = .FALSE.`) силовой источник рассчитывается в центре ячейки по соотношению:

$$\mathbf{F}_P \Delta V_P = (C_{g,P} + C_{k,P})(\nabla \varphi)_P \Delta V_P, \quad (2.20)$$

$$C_{g,P} = -(\rho_f - \rho_g)(\mathbf{x}_P \cdot \mathbf{g}), \quad C_{k,P} = K_P \cdot \sigma$$

Дополнительно в алгоритм Рие-Чоу расчета объемного и массового потока на грани ячейки вводятся коррекционные члены, аналогичные коррекционным членам для давления (см. раздел 4.2 документа [7]):

$$U_{n,k} = U_{n,k}^0 - \left(\frac{\Delta V}{a_p} \right)_f C_{s,f} \frac{(\varphi_{nb} - \varphi_P)}{\delta_k} + \left(\frac{\Delta V}{a_p} C_{s,P} \nabla \varphi \right)_f \cdot \frac{\mathbf{r}_{P-nb}}{\delta_k},$$

$$C_{s,P} = -(\rho_f - \rho_g)(\mathbf{x}_P \cdot \mathbf{g}) + K_P \cdot \sigma, \quad (2.21)$$

$$C_{s,f} = -(\rho_f - \rho_g)(\mathbf{x}_f \cdot \mathbf{g}) + \overline{(K_P)}_f \cdot \sigma$$

Здесь

$U_{n,k}^0$ - нормальная компонента скорости на грани, рассчитанная по модели

Рие-Чоу (соотношение (4.5) документа [7]),

$\mathbf{x}_P, \mathbf{x}_f$ - радиус-вектор центра ячейки и центра грани,

$\overline{(f)}_k$ - линейная интерполяция значения поля на k-грань.

Как и в случае структурных сеток, при использовании сглаженных свойств значения φ_P, φ_{nb} и градиент φ рассчитываются по сглаженному полю $\tilde{\varphi}$. При использовании точных свойств значения φ_P, φ_{nb} рассчитываются по точному полю φ .

2.3.3 BodyForce модель силового источника

В этой модели «сильно неоднородный» силовой источник (2.20) моделируется с помощью BodyForce модели. Подробно эта модель описана в разделе 4.8 документа [7]. В этой модели производится согласование значений источника в ячейке и значений источника на гранях ячейки. Применительно к источнику (2.20) это сводится к следующему:

- 1) точно рассчитываются значения источника на гранях ячеек;
- 2) по соотношениям BodyForce модели рассчитываются «сглаженные» значения источников в центрах ячеек, которые и используются в дискретных уравнениях движения;
- 3) в алгоритм расчета объемного и массового потока на грани ячейки вводятся коррекционные BodyForce члены.

В BodyForce модели значение в центре ячейки "P" рассчитывается по соотношению:

$$\mathbf{F}_P = \frac{1}{\Delta V_P} \sum_{nb} (1 - f_{p,nb}) \cdot (\mathbf{F}_f \cdot \mathbf{d}_{p,nb}) \mathbf{n}_f \Delta A_f,$$

где $f_{p,nb}$ - коэффициент линейной интерполяции на грань между ячейками "P" и "nb"

$$\mathbf{F}_f = F_P f_{p,nb} + F_{nb} (1 - f_{p,nb}),$$

$\mathbf{d}_{p,nb}$ - вектор из центра ячейки P в центр ячейки nb, \mathbf{n}_f - нормаль грани, направленная в nb-ячейку. Скалярное произведение силы \mathbf{F}_f на грани и вектора $\mathbf{d}_{p,nb}$ рассчитывается по соотношению:

$$\mathbf{F}_f \cdot \mathbf{d}_{p,nb} = C_{s,f} (\varphi_{nb} - \varphi_P), \quad C_{s,f} = -(\rho_f - \rho_g)(\mathbf{x}_f \cdot \mathbf{g}) + \overline{(K_P)}_f \cdot \sigma$$

Здесь, как и в стандартной модели, при использовании сглаженных свойств значения φ_P, φ_{nb} рассчитываются по сглаженному полю $\tilde{\varphi}$. При использовании точных свойств значения φ_P, φ_{nb} рассчитываются по точному полю φ .

Для расчета нормальной скорости на грани используется соотношение Рие-Чоу, аналогичное (2.21):

$$U_{n,k} = U_{n,k}^0 - \left(\frac{\Delta V}{a_p} \right)_f C_{s,f} \frac{(\varphi_{nb} - \varphi_p)}{\delta_k} + \left(\frac{\Delta V}{a_p} \mathbf{F}_p \right)_f \cdot \frac{\mathbf{r}_{p-nb}}{\delta_k}$$

2.3.4 Пример использования силовых источников

В качестве демонстрации описанных выше моделей можно привести двумерную задачу о превращении квадратной капли воды в цилиндрическую каплю за счет сил поверхностного натяжения в невесомости. Исходный размер квадрата соответствует цилиндрической капле радиуса $R_0 = 50$ мкм. Среда - воздух. В начальный момент времени давление и скорость среды равны нулю во всей РО. Для расчета использовались структурные сетки.

На рисунке 2.10 показаны поля объемной доли жидкости для четырех характерных моментов времени: Время 400 мкс соответствует стационарному состоянию сформировавшейся цилиндрической капли. На рисунке 2.11 показана эволюция относительного поля давления во времени. В качестве характерного давления p_0 используется лапласовский скачок давления для цилиндрической капли $p_0 = \sigma/R_0$.

Как следует из рисунков 2.10, 2.11 и 2.13b после нескольких циклов колебаний поверхности раздела капля принимает форму цилиндра (с большой степенью точности), а давление внутри капли совпадает с лапласовским скачком давления.

Однако после формирования цилиндрической капли поле скорости не равно нулю. Это поле в стационарном режиме показано на рисунке 2.12, а изменение модуля максимальной скорости во времени - на рисунке 2.13а. Заметим, что аналогичные «паразитные» вихревые структуры были обнаружены и другими авторами.

Для моделирования использовалась три модели расчета свойств в центре ячейки и на ее гранях. На рисунке 2.13а показаны кривые максимальной скорости для этих трех моделей. Для моделей с точной величиной ϕ (VOF_face_lin и VOF_face_tru) амплитуда «паразитной» скорости уменьшилась почти в два раза, но сами вихри все равно остались.

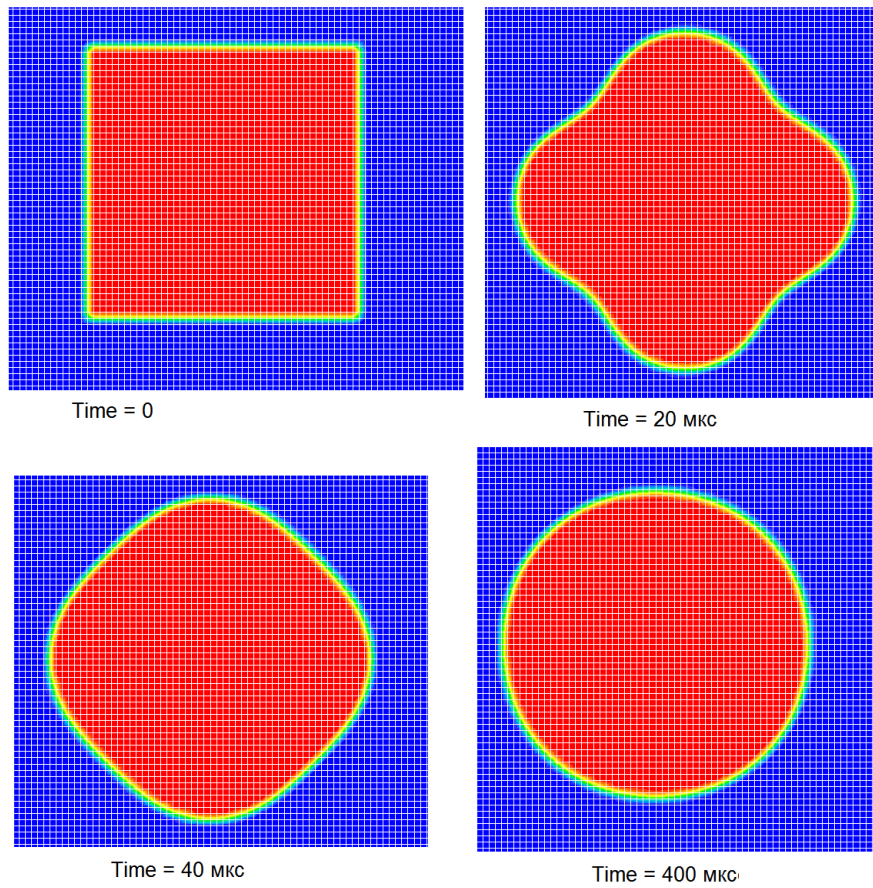


Рисунок 2.10 - Эволюция формы капли жидкости

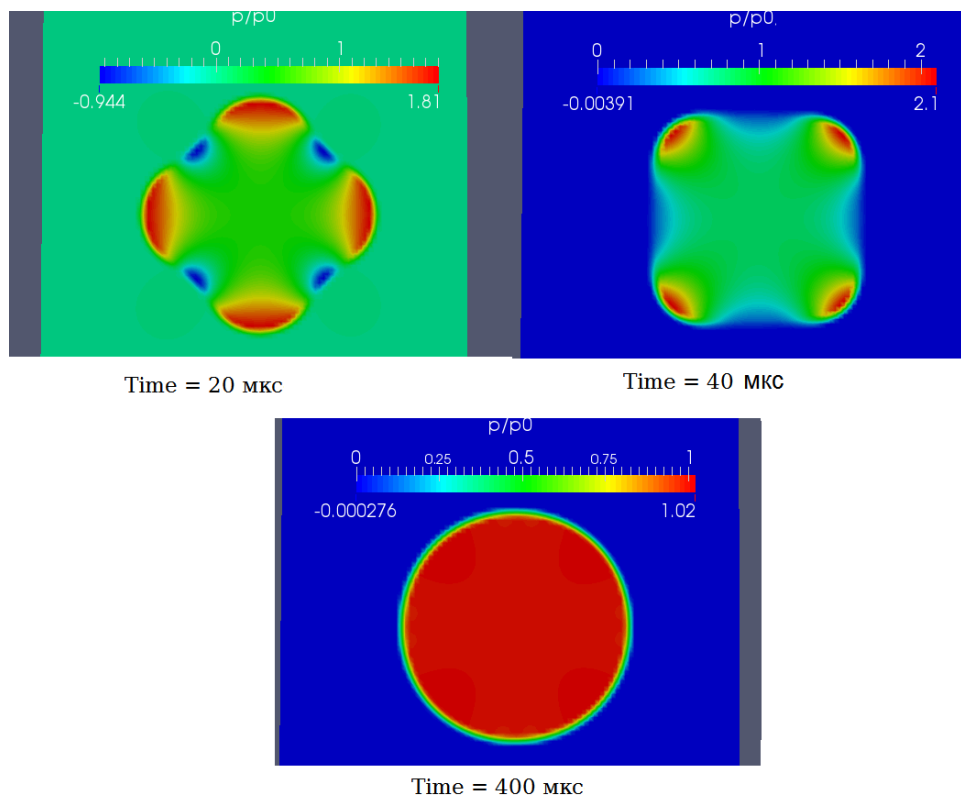


Рисунок 2.11 - Эволюция относительного поля давления

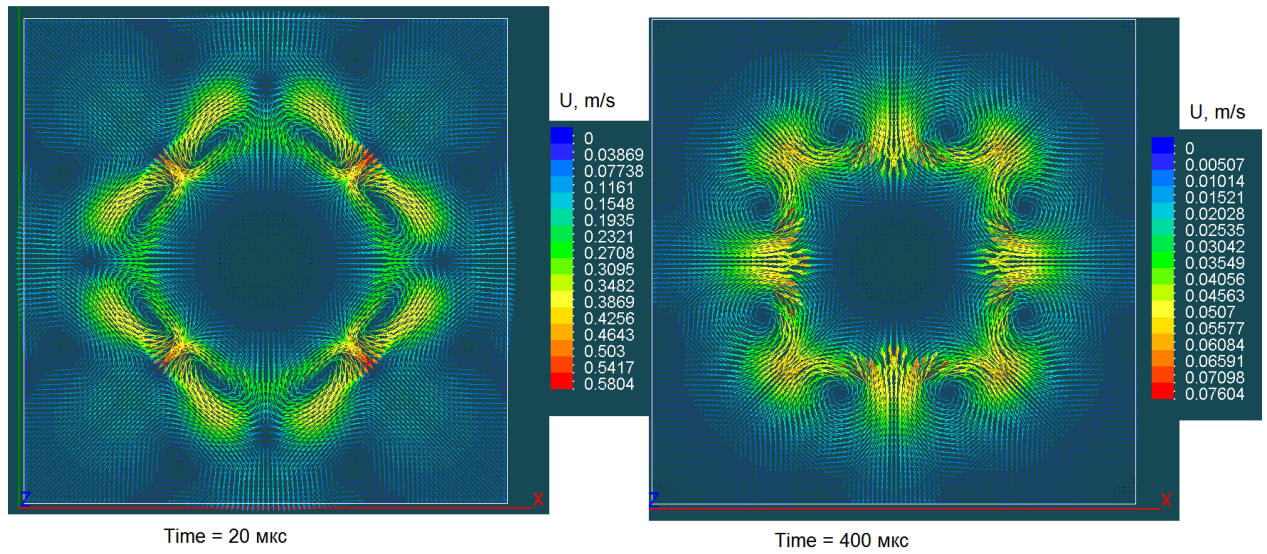


Рисунок 2.12 - Поле вектора скорости для начального и конечного момента формирования цилиндрической капли

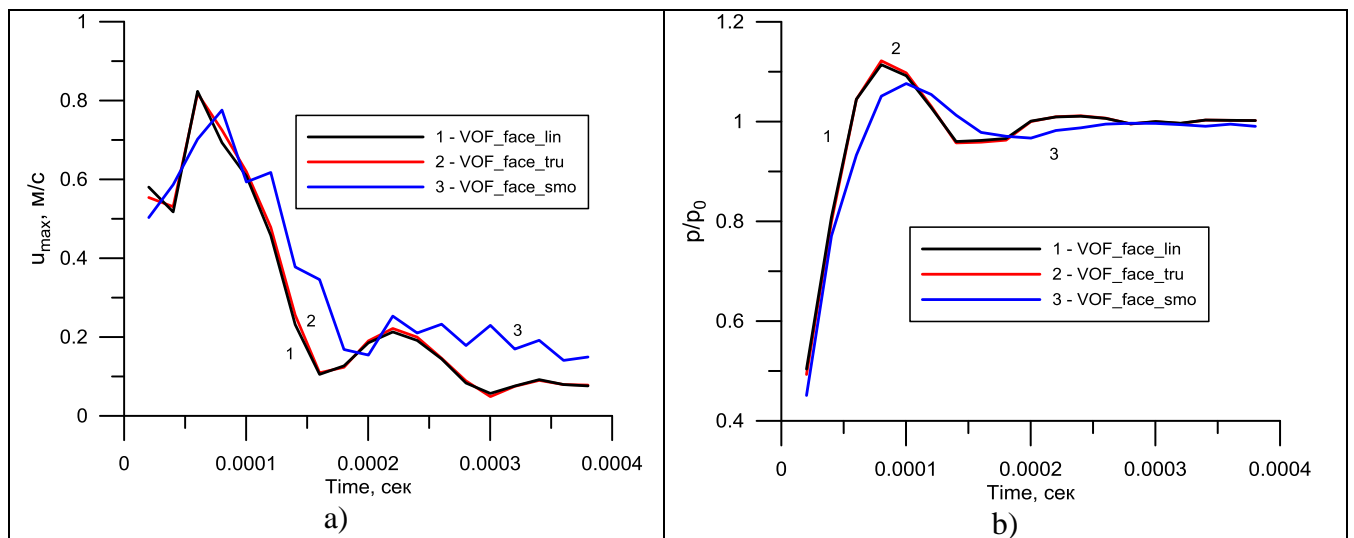


Рисунок 2.13 - Зависимость модуля вектора максимальной скорости (а) и относительного давления в центре капли жидкости (б)

2.4. Уравнение энергии

Основная цель включения уравнения энергии в систему VOF-уравнений связана с необходимостью моделирования массообмена на межфазной границе. Однако его можно использовать и для моделирования теплообмена в VOF-задачах без массообмена (например, задача о падении термопары в горячую воду).

Дискретное уравнение энергии можно получить с помощью стандартного алгоритма МКО кода Anes (см. документ [7]). Для простоты рассмотрим случай противопоточной численной схемы. Интегрируя уравнение(1.21) по КО, получим:

$$\begin{aligned} \frac{(\rho c_p)_P^0 (T_P - T_P^0)}{\Delta \tau} \Delta V_P + \sum_{k, nb} \max(0, -V_{n,k}) (\rho c_p)_k \cdot (T_P - T_{nb}) + \sum_{k, nb} \frac{\lambda_k}{\delta_k} \cdot (T_P - T_{nb}) \Delta A_k = \\ = M_{gf,P} \Delta V_P \left[\Delta h_{gs} + (T_{sat} - T_P) (c_{pf} - c_{pg}) \right] - T_P \cdot D_{T,P}, \end{aligned} \quad (2.22)$$

$$D_{T,P} = (\rho_f c_{pf} - \rho_g c_{pg}) \left\{ \frac{\Phi_P - \Phi_P^0}{\Delta \tau} \Delta V_P + \sum_k V_{n,k} \Phi_k \Delta A_k - \frac{M_{gf,P}}{\rho_f} \Delta V_P \right\},$$

$$(\rho c_p)_P = \rho_f c_{pf} \Phi_P + \rho_g c_{pg} (1 - \Phi_P), \quad (\rho c_p)_k = \rho_f c_{pf} \Phi_{t,k} + \rho_g c_{pg} (1 - \Phi_{t,k})$$

Здесь $\delta_k = |\mathbf{d}_{p,nb}|$ - расстояние между центрами ячеек P и nb, λ_k - коэффициент теплопроводности на k-ой грани. Член D_T является аналогом невязки (2.19) в уравнениях движения.

2.4.1 Аппроксимация конвективных членов

При решении уравнения энергии для расчета конвективных потоков энергии используется модель `VOF_face_tru`. В этой модели используются «точные» значения объемной доли в центрах ячеек и значения на гранях $\Phi_{t,k}$, полученные на корректор-шаге `CISIT`. Как показано в разделе 2.1.7, в этом случае член $D_{T,P}$ мал и его можно удалить из дискретного уравнения.

2.4.2 Аппроксимация диффузионных членов

В принципе для расчета коэффициентов теплопроводности на грани λ_k можно использовать тот же подход, что и для конвективных членов:

$$\lambda_k = \lambda_f \Phi_{t,k} + \lambda_g (1 - \Phi_{t,k}), \quad (2.23)$$

а для расчета δ_k использовать геометрическое расстояние между центрами ячеек. Однако в коде для ячеек, содержащих межфазную границу, используется другой алгоритм.

Суть этого алгоритма заключается в следующем: при расчете диффузионных потоков будем считать, что температура в данной ячейке T_P , входящей в дискретное уравнение (2.22), равна температуре межфазной границы T_{int} . Для задач без массообмена на межфазной границе это утверждение не имеет большого смысла, поскольку T_{int} не определена. Однако для задач с массообменом T_{int} известна и равна температуре насыщения T_{sat} .

Если одна из ячеек *внутренней* грани содержит межфазную поверхность, то коэффициент теплопроводности на грани λ_k и расстояние между центрами ячеек δ_k рассчитываются по следующему алгоритму (рисунок 2.14):

$$\text{Вариант а: } \lambda_e = \lambda_f \varphi_e^t + \lambda_g (1 - \varphi_{t,e}), \quad \delta_e = \frac{\Delta x_P + \Delta x_E}{2}, \quad (2.24)$$

$$\text{Вариант б: } \lambda_e = \lambda_g, \quad \delta_e = (1 - \varphi_P) \Delta x_P + \frac{\Delta x_E}{2},$$

$$\text{Вариант с: } \lambda_e = \lambda_f, \quad \delta_e = \varphi_P \Delta x_P + \frac{\Delta x_E}{2},$$

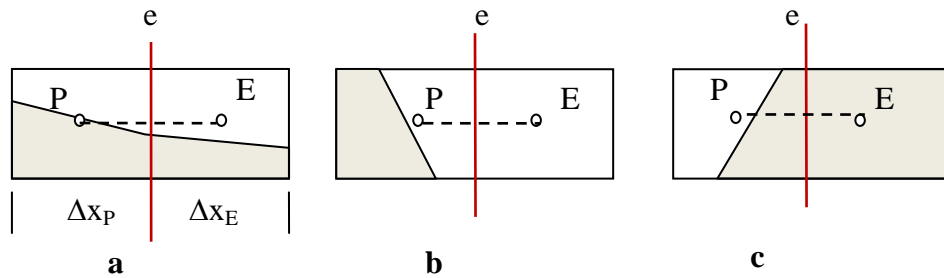


Рисунок 2.14 - Варианты расположения внутренней грани КО

Идея соотношений (2.24) близка к идеям других VOF-алгоритмам, в которых для расчета подвода тепла к фазовой поверхности используются различные *конечно-разностные* аппроксимации второго порядка точности. Алгоритм Anes основан на *конечно-объемном* представлении, что в большей степени согласуется с численной схемой МКО. Соотношения (2.24) имеют первый порядок точности (направление нормали \mathbf{n}_{int} фазовой границы не принимается во внимание), который обычно численно более устойчив, чем алгоритмы более высоких порядков. Именно поэтому он был выбран в качестве первой версии модели массообмена. В следующих версиях кода модель будет усовершенствована.

2.4.3 Граничные условия на твердой стенке

В Anes граничные условия рассматриваются как поверхностные источники. Если граница непроницаема, то плотность теплового потока на граничной грани q_w рассчитывается по соотношению (раздел 3.5 документа [7]):

$$q_w = \frac{1}{\frac{1}{C_b} + \frac{1}{\alpha_w}} (T_b - T_P), \quad \alpha_w = \frac{\lambda_w}{\delta_w}$$

где C_b и T_b - коэффициент и значение граничного условия. Для граничного условия первого рода ($C_b = BC_FIXVAL$) $C_b = \infty$, а $T_b = T_{wall}$. Для расчета λ_w и δ_w используются соотношения, аналогичные (2.24) (см. рисунок 2.15):

$$\text{Вариант а: } \lambda_w = \lambda_g, \quad \delta_w = (1 - \varphi_P) \Delta x_P, \quad (2.25)$$

$$\text{Вариант б: } \lambda_w = \lambda_f, \quad \delta_w = \varphi_P \Delta x_P,$$

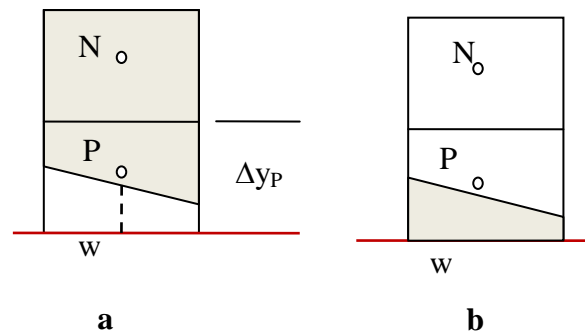


Рисунок 2.15 - Варианты расположения граничной грани КО

Соотношения (2.25) являются фактически первой версией модели для учета пристенных микрослоев. Они учитывают увеличение подвода тепла к фазовой поверхности через пленку пара при пленочном кипении и увеличение отвода тепла от пленки жидкости при конденсации при уменьшении толщин пленок. Однако эта модель не может описать, например, следующие ситуации:

- 1) начало процесса конденсации на холодной стенке, когда вблизи нее нет жидкой пленки,
- 2) начало процесса возникновения паровой фазы при перегреве поверхности в чистой жидкости,
- 3) теплоперенос в жидком микрослое при росте парового пузыря на перегретой стенке.

Эти эффекты будут учтены в следующих версиях кода

2.4.4 Модель массообмена на межфазной границе

В коде Anes используется простая модель массообмена, суть которой заключается в следующем.

Будем считать, что в ячейках с межфазной границей температура равна температуре насыщения T_{sat} . Это можно сделать, если заменить дискретное уравнение (2.22) на «вырожденное» уравнение

$$10^{20} T_p = 10^{20} T_{sat}.$$

В этом случае оригинальное уравнение для этой ячейки можно использовать для вычисления притока или оттока массы f-фазы в ячейке:

$$M_{gf,p} \Delta V_p = \frac{1}{\Delta h_{gf}} \left\{ \sum_{k,nb} \max(0, -V_{n,k}) (\rho c_p)_k \cdot (T_{sat} - T_{nb}) + \sum_{k,nb} \frac{\lambda_k}{\delta_k} \cdot (T_{sat} - T_{nb}) \right\} \quad (2.26)$$

Заметим, что в соотношении (2.26) входит не только теплоприток за счет теплопроводности, но и часть *втекающего* теплового потока за счет конвекции.

Описание в проекте прикладной задачи :

Для активации модели массообмена необходимо определить в секции [Properties] два новых постоянных свойства:

PROP("<Имя G-материала PO>.VOF_TSAT") = <температура насыщения T_{sat} >
 PROP("<Имя G-материала PO>.VOF_DHSAT ") = <теплота кипения Δh_{gf} >

2.5. Верификация VOF-алгоритма

Для верификации VOF-алгоритма использовалось большое число тестовых задач. В данном разделе приводятся результаты решения классической задачи об обрушении водяной колонны и сравнение их с результатами кода OpenFOAM (версия 2.1 для Windows).

Рассматривается двумерная прямоугольная расчетная область шириной $L_x = 4 \cdot a$ и высотой $L_y = 3 \cdot a$, где $a = 0.146$ м (это параметры эксперимента [9]). Область заполнена воздухом. В начальный момент времени в левой нижней части области расположена прямоугольная колонна воды размером $L_{cx} = a$, $L_{cy} = 2a$. Сила тяжести направлена вертикально вниз и равна $g_y = -9,8$ м/с². На левой, правой и нижней границе расчетной области задаются условия прилипания. На верхней свободной границе задается постоянное *статическое* давление равное нулю, что эквивалентно заданию на границе давления p_g :

$$p_g = \rho \cdot L_y \cdot g_y \quad (3.1)$$

В расчете учитывались как силы плавучести, так и силы поверхностного натяжения. Как показали расчеты, влияние последних пренебрежимо мало.

Результаты моделирования приведены на рисунках 2.16 - 2.18. Время проведения расчета:

Код OpenFOAM (PISO)	= 123 сек,
Код Anes (SIMPLE)	= 443 сек,
Anes (PIMPLE, OneSweep)	= 135 сек,
Anes (PIMPLE, Explicit)	= 173 сек

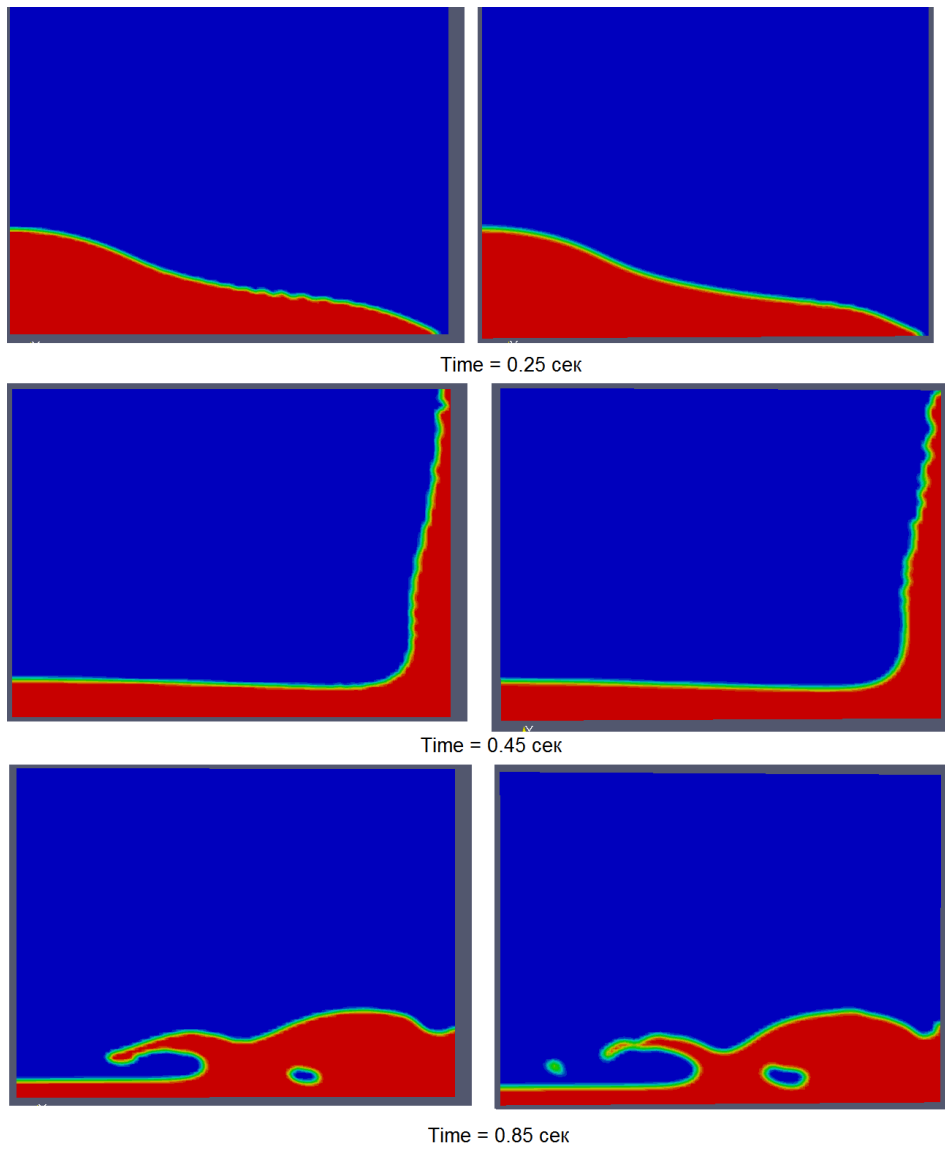


Рисунок 2.16 - Поле VOF-переменной для разных моментов времени: слева - результаты с использованием Apes, справа - OpenFOAM

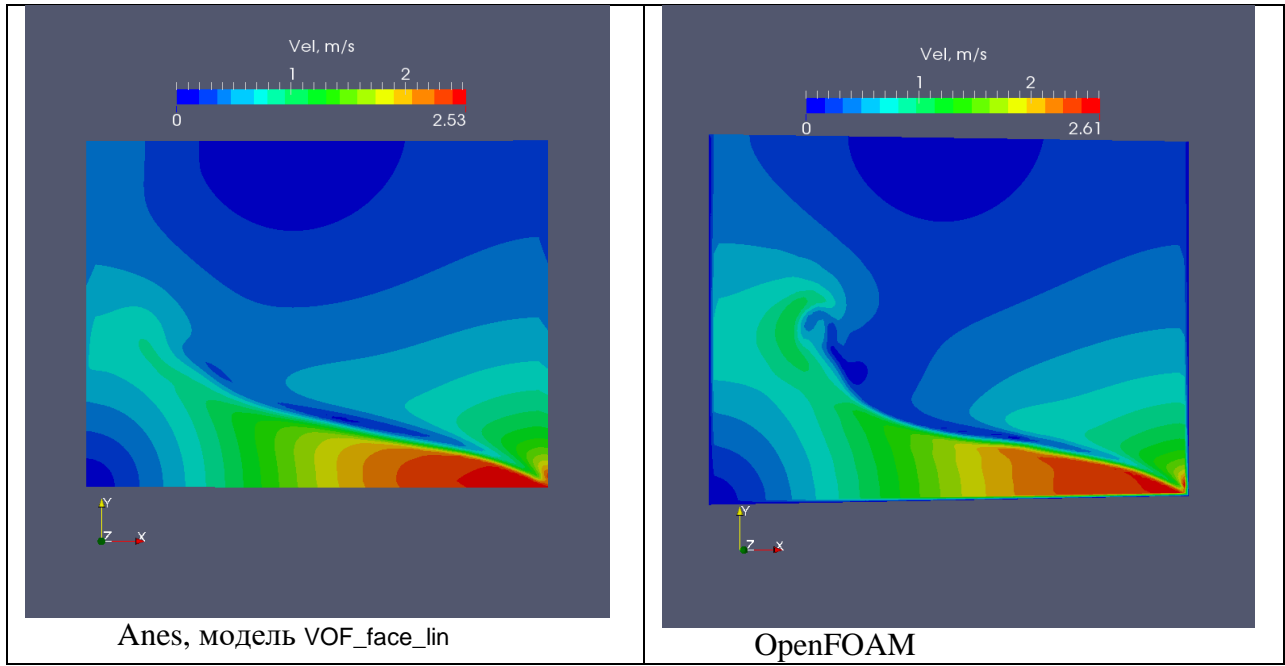


Рисунок 2.17 - Поле модуля скорости для момента времени 0.25 сек.

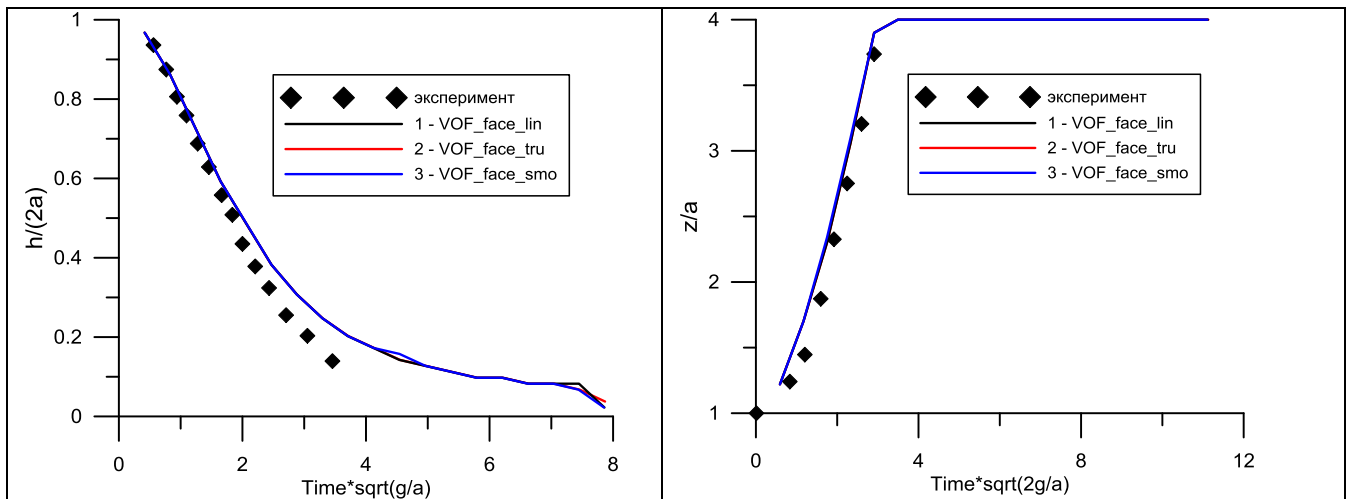


Рисунок 2.18 - Безразмерная высота и ширина колонны, точки - эксперимент [9]

3. Работа с VOF-алгоритмом ANES

3.1. Активация модели

Для активации VOF-модели необходимо включить решение Φ -переменной с именем "VOF". Двухфазная смесь моделируется G-фазой кода Anes. Постоянные свойства f- и g-фазы двухфазной смеси задаются операторами секции [Properties]:

```
PROP("<Имя G-материала PO>.Dens.FLU") = <плотность f-фазы>
PROP("<Имя G-материала PO>.Dens.GAS") = <плотность g-фазы>

PROP("<Имя G-материала PO>.Visc.FLU") = <кинематическая вязкость f-фазы>
PROP("<Имя G-материала PO>.Visc.GAS") = <кинематическая вязкость g-фазы>
```

Для моделирования сил поверхностного натяжения необходимо определить свойство:

```
PROP("<Имя G-материала PO>.SigmsFG") = <коэффициент поверхностного натяжения>
```

Если решается уравнение энергии для Φ -переменной "TG", то необходимо добавить еще четыре свойства:

```
PROP("<Имя G-материала PO>.Cp.FLU") = <теплоемкость f-фазы>
PROP("<Имя G-материала PO>.Cp.GAS") = <теплоемкость g-фазы>

PROP("<Имя G-материала PO>.Cond.FLU") = <теплопроводность f-фазы>
PROP("<Имя G-материала PO>.Cond.GAS") = <теплопроводность g-фазы>
```

Для моделирования массообмена на межфазной границе нужно добавить свойства:

```
PROP("<Имя G-материала PO>.VOF_TSAT") = <температура насыщения  $T_{sat}$ >
PROP("<Имя G-материала PO>.VOF_DHSAT ") = <теплота кипения  $\Delta h_{gf}$ >
```

Краевой угол смачивания для Wall-патча задается с помощью оператора секции [Special Data]:

```
R("<Имя патча>.ContactAngle") = 90
```

Значение угла задается в градусах. Для включения сил плавучести необходимо в секции [Internal Sources] активировать расчет этих сил через плотность и задать ненулевые компоненты вектора ускорения свободного падения, например:

```
ArxForce = ARX_RHO
AgravY = -9.8
```

3.2. Настройка численной схемы

Для решения VOF-задач лучше всего использовать алгоритм PIMPLE, который устанавливается оператором секции [Main]:

```
WayOfSolve = WS_PIMPLE
```

Коэффициент релаксации для давления необходимо установить в 1, коэффициенты релаксации для скоростей ≤ 1 . Для настройки других параметров используются операторы секции [Special Data]. Ниже первым красным (или единственным) значением указывается значение по умолчанию.

Для расчета VOF-уравнений только на одной первой итерации шага по времени используется оператор:

```
C("SweepVOFModelID") = "OneSweep" / "Explicit"
```

Максимальное число Куранта задается оператором:

```
R("CriCourant") = 0.5
```

Число циклов алгоритма сглаживания объемной доли жидкости задается оператором:

```
I("NoSMO_VOF") = 2
```

Использование сглаженной или точной объемной доли в силовых источниках и свойствах определяется оператором:

```
L("IsSMO_VOFPROP ") = .TRUE. / .FALSE.
```

Выбор модели расчета ϕ на гранях ячейки VOF_face_lin или VOF_face_tru определяется оператором:

```
C("VofForFMassID ") = LIN / FracTRU
```

Заметим что этот оператор используется только для модели точной объемной доли

(IsSMO_VOFPROP = .false.). При использовании сглаженных свойств всегда используется модель VOF_face_lin.

Для выбора модели расчета силовых источников при работе с неструктурными сетками используется оператор

```
L("IsBodyForceVOF") = .TRUE. / .FALSE.
```

По умолчанию используется BodyForce модель.

3.3. Визуализация межфазной поверхности

Для визуализации межфазной границы в виде 3d поверхности необходимо в секцию [Special Data] добавить логический оператор:

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

В этом случае для каждого «сохраняемого» шага по времени будут созданы vtk-файлы с именем (XX – номер шага по времени)

```
<префикс результатов>_VOFS_XX.vtk,
```

содержащие поверхность межфазной границы в формате VTK. Эту поверхность можно просмотреть как в постпроцессоре ParaView, так и в постпроцессоре Anes. На рисунке 3.1 показана поверхность раздела для задачи о всплытии трехмерного сферического пузырька в постпроцессоре Anes.

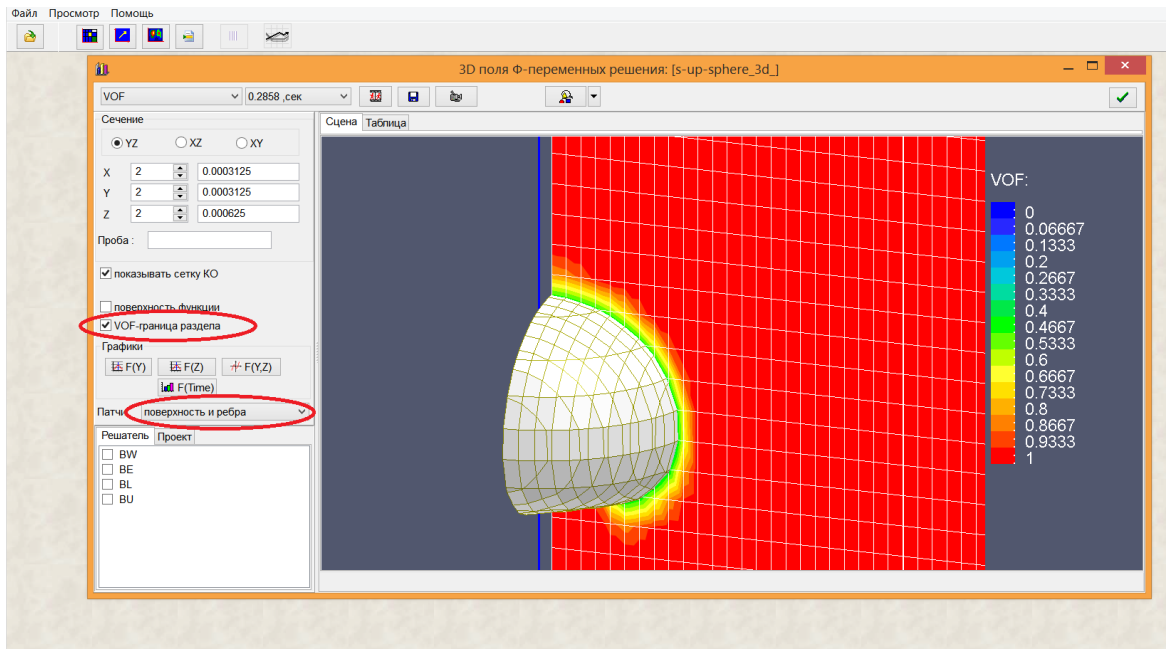


Рисунок 3.1 - Визуализация межфазной поверхности в постпроцессоре Anes

3.4. Начальное распределение VOF-переменной

Начальное распределение объемной доли можно задать в событии пользователя `onInit`. Для облегчения задания распределений ϕ в состав Решателя добавлены специальные подпрограммы пользователя.

Принцип работы этих подпрограмм очень простой: значение VOF в ячейке - это объемная доля ячейки, занятая f -фазой. Для работы с 2D задачами можно использовать три подпрограммы пользователя:

```
subroutine userVOFCircle(PV,PCIR,RCIR,FracVOF)
  real PV(2,4),PCIR(2)
end subroutine

SUBROUTINE userVOF_RECT(PV,RECT_C,RECT_L,FracVOF)
  real PV(2,4)
  real RECT_C(2),RECT_L(2)
end subroutine

SUBROUTINE userVOF_HX(PV,FH,FracVOF)
  real PV(2,4)
  external FH
end subroutine
```

Первая подпрограмма предназначена для расчета объемной доли *окружности*, «попавшей» внутрь плоского прямоугольника PV. Для описания координат окружности и прямоугольника используется двумерная система координат (x_1, x_2) . Координаты точек прямоугольника передаются в массиве `PV(1:2,iPoint)`. Индексы точек показаны на рисунке 3.2. Окружность задается координатами ее центра `PCIR(2)` и радиусом `RCIR`. Подпрограмма возвращает долю заданного прямоугольника, лежащую внутри окружности - `FracVOF`.

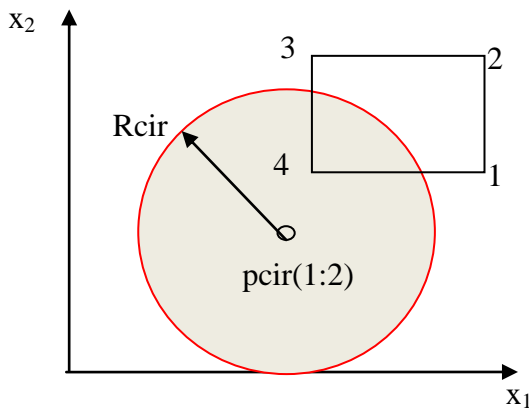


Рисунок 3.2 – Геометрические объекты для подпрограммы userVOFcircle

Вторая подпрограмма аналогична первой, но сам объект представляет собой не окружность, а *прямоугольник*, заданный своим центром $RECT_C(2)$ и размерами вдоль осей x_1 и x_2 - $RECT_L(2)$.

Третья подпрограмма рассчитывает долю прямоугольника PV , занятого границей функции FH . Эта функция задается пользователем, ее формат:

```
real function FH(x1)
end function
```

Функция возвращает координату x_2 поверхности раздела для значения координаты x_1 . При использовании этих сервисных функций следует помнить о правилах «отбора» двумерной СК (x_1, x_2) из трехмерной системы кода Anes (x, y, z) :

$(x_1, x_2) \leq (x, y), (z, x), (y, z)$

При работе с ячейками сетки PV - это 2D сечение параллелепипеда KO . Точки этого сечения можно либо заполнить самостоятельно, либо воспользоваться функцией:

```
function userCellRect2D(iSectDIR) result(PV)
  real PV(2,4)
  .....
end function
```

Эта функция возвращает значение прямоугольника двумерного сечения *текущей* ячейки с индексами (lx, ly, lz) для структурной сетки и $idCV$ для неструктурной. Направление сечения определяется аргументом $iSectDIR$:

$iSectDIR = 1 : (y, z); iSectDIR = 2 : (z, x); iSectDIR = 3 : (x, y)$

Пример использования первой подпрограммы без функции `userCellRect2D`:

```
[vf iLSINIT]
subroutine iLSINIT(RetVal)
  real PV(2,4), PCIR(2)
!-----
  RCIR = RUDat(1)/2
  PCIR(1:2) = (/0.0,4.0*RCIR /)
  PV(1:2,1) = (/ xFACE(lxC), zFACE(lzC) /)
  PV(1:2,2) = (/ xFACE(lxC+1), zFACE(lzC) /)
  PV(1:2,3) = (/ xFACE(lxC+1), zFACE(lzC+1) /)
  PV(1:2,4) = (/ xFACE(lxC), zFACE(lzC+1) /)
! -----
```

```

    call userVOFCircle(PV,PCIR,RCIR,retval)
! ----- У нас в цилиндре - ГАЗ -----
    RetVal = 1.0 - RetVal
end subroutine

```

Пример использования первой подпрограммы с функцией userCellRect2D:

```

[vf iLSINIT]
subroutine iLSINIT(RetVal)

    call userVOFCircle(userCellRect2D(3),[0.0 ,0.0 ], 5.5,retval)
! ----- У нас в цилиндре - ГАЗ -----
    RetVal = 1.0 - RetVal
end subroutine

```

Пример использования третьей подпрограммы для задачи о неустойчивости Релея-Тейлора:

```

[USER Fortran Subroutines]
real function HGX(XX)
    X00 = 2 * 3.1416 * XX
    HGX = 2 + 0.05 * COS(X00)
end function

.....
[vf iLSINIT]
subroutine iLSINIT(RetVal)
    real PV(2,4)
    external HGX
!-----
    call userVOF_HX(userCellRect2D(3),HGX,retval)
! У нас внизу - ГАЗ.
    RetVal = 1.0 - RetVal
end subroutine

```

Для работы с 3D задачами в декартовой СК имеется одна подпрограмма - userVOFSphere3D, аналогичная подпрограмме userVOFCircle. Подпрограмма предназначена для расчета объемной доли *сферы*, «попавшей» внутрь параллелепипеда PV в декартовой СК. Вызов подпрограммы:

```

subroutine userVOFSphere3D(PV,PSPE,RSPE,FracVOF)
    real PV(3,2),PSPE(3)

end subroutine

```

Здесь массив PV описывает параллелепипед. Вектор PV(1:3,1) - это точка «нижней» вершины параллелепипеда, вектор PV(1:3,2) - точка «верхней» вершины. При работе с текущими КО (с индексами idCV или (lxC,lxC,lzC)) для расчета этих векторов можно использовать функцию

```

function userCell3D() result(PV)
    real PV(3,2)
.....
end function

```

Вектор PSPE(1:3) описывает положение центра сферы, а переменная RSPE - радиус сферы.

Литература

1. C.W. Hirt, B.G. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries// J. Comput. Phys., 1981, Vol. 39, pp. 201-225.
2. Д.А. Лабунцов, В.В. Ягов. Механика двухфазных систем // М.: Издательство МЭИ, 2000, 374 с.
3. Р.И. Нигматулин. Динамика многофазных сред. Часть I// М.: Наука, Гл. ред. физ. - мат. лит., 1987, 464 с.
4. J. U. Brackbill, D. B. Kothe, and C. Zemach, A Continuum Method for Modeling Surface Tension// J. Comput. Phys., vol. 100, pp. 335–354, 1992.
5. O. Ubbink. Numerical prediction of two fluid systems with sharp interfaces// Thesis submitted for the Degree of Doctor of Philosophy of the University of London and Diploma of Imperial College, 1997.
6. Yeng-Yung Tsui & Shi-Wen Lin. A VOF-Based Conservative Interpolation Scheme for Interface Tracking (CISIT) of Two-Fluid Flows// Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology, 2013, V. 63, N. 4, pp. 263-283
7. Код Anes20хе. «Численные алгоритмы кода Anes», версия 2.24, 2019.
8. Код Anes20хе. «Описание математических моделей кода», версия 2.24, 2019.
9. J. C. Martin and W. J. Moyce, An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane// Phil. Trans. Roy. Soc. Lond., 1952, Ser. A, vol. 244, pp. 312–324.