

РАЗРАБОТКА АЛГОРИТМА ДЛЯ СОВМЕСТНОГО РЕШЕНИЯ МНОГОМЕРНЫХ И СЕТЕВЫХ ЗАДАЧ

С.А. Филимонов
ООО ТИРР, г. Красноярск
e-mail: bdk@inbox.ru

Д.В. Бойков
e-mail: dimkadam@yandex.ru

This paper presents an algorithm of combined solving of the multidimensional and the network problems. An example of a hybrid testing problems with two-dimensional and network parts are presented.

Введение

На практике встречаются конструкции, различные части которых требуют различного уровня моделирования. В таких задачах существует один (или несколько) элементов значительно более сложных, чем остальные. Зачастую простые элементы – это различные подводящие трубы, шланги и т.д. Примером такой задачи являются различные системы охлаждения, где теплообменник – это сложный элемент, подводящие к нему патрубки – простые элементы. При моделировании этих задач можно пойти несколькими путями:

- 1 – построить полностью многомерную модель;
- 2 – построить полностью сетевую модель, в которой сложный элемент представлен в виде одной ветви (или набора ветвей). Для определения гидравлического сопротивления элемента необходимо использовать многомерное моделирование;
- 3 – построить гибридную модель, включающую в себя как элементы сети, так и многомерные элементы.

Первый вариант может оказаться очень сложным и трудоемким в реализации. Вторым вариантом более прост при построении геометрии, но при различных режимах течения необходимо каждый раз определять гидравлическое сопротивление. В некоторых случаях наиболее приемлемым можно считать третий вариант, где в одной задаче разрешается как многомерная, так и сетевая область. Однако для такого варианта необходимо разработать соответствующую математическую модель.

Математическая модель

Для разрешения гибридной задачи необходимо было создать алгоритм, одинаково разрешающий как CFD (Computational Fluid Dynamics), так и сетевые задачи. За основу был взят алгоритм SIMPLE [1]. Основные уравнения представлены ниже. Для разработки алгоритма был рассмотрен стационарный случай – изотермическая задача с несжимаемой жидкостью без учета действия гравитационных сил.

Уравнение сохранения массы:

$$\oint_S \rho (\vec{U} \cdot \mathbf{n}) dS = \int_V q dV, \quad (1)$$

где ρ – плотность, \mathbf{U} – вектор скорости в контрольном объёме, S – площадь поверхности контрольного объема, q – источник массы в контрольном объеме, V – контрольный объем.

Уравнение сохранения импульса:

$$\oint_S \rho \vec{U} (\mathbf{U} \cdot \mathbf{n}) dS = - \oint_S P \mathbf{n} dS - \oint_S (\boldsymbol{\tau} \cdot \mathbf{n}) dS, \quad (2)$$

где P – давление, $\boldsymbol{\tau}$ – вязкостное напряжение, которое для CFD задачи равно:

$$\boldsymbol{\tau} = -\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad (3)$$

а для сетевой задачи:

$$\tau = \left(\frac{\lambda \cdot l}{d} + \xi \right) \cdot \frac{\rho \cdot u_t^2}{2}. \quad (4)$$

Здесь μ – динамическая вязкость, u и v – компоненты скорости, λ – коэффициент линейного трения, d – гидравлический диаметр ветви, l – длина ветви, ξ – коэффициент местного сопротивления, u_t средняя скорость на ветви.

Дискретизация двумерной части проводилась на разнесенной (шахматной) сетке. Последнее связано с тем, что дискретизация сетевой задачи на графе, по сути, осуществляется так же. На рисунке 1 представлена схема дискретизации на примере u -компоненты скорости.

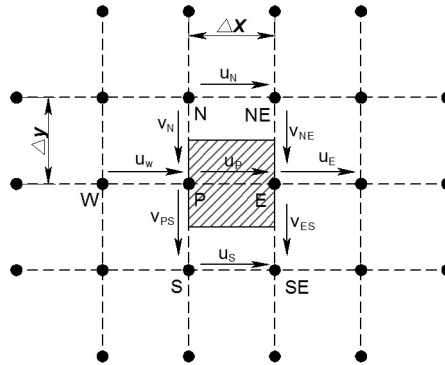


Рис. 1. Дискретизация двумерной части задачи на примере u -компоненты скорости.

После дискретизации получаем следующую систему уравнений.

Уравнение сохранения массы относительно точки P:

$$\rho u_w \cdot \Delta y - \rho u_p \cdot \Delta y + \rho v_n \cdot \Delta x - \rho v_{ps} \cdot \Delta x = 0, \quad (5)$$

Уравнение сохранения импульса для u – компоненты скорости.

$$\begin{aligned} \rho u_w u_W \cdot \Delta y - \rho u_p u_E \cdot \Delta y + \rho u_n (v_{np} + v_{ne})/2 \cdot \Delta x - \rho u_s (v_{sp} + v_{se})/2 \cdot \Delta x - \\ - \mu \frac{\Delta u}{\Delta y} \cdot \Delta x - \mu \frac{\Delta v}{\Delta x} \cdot \Delta y = (P_P - P_E) \cdot \Delta y \end{aligned} \quad (6)$$

Преобразуем полученное уравнение следующий вид

$$A_p \cdot u_p - A_w \cdot u_w - A_e \cdot u_e - A_n \cdot u_n - A_s \cdot u_s = (P_P - P_E) \cdot \Delta y, \quad (7)$$

где коэффициенты A_p , A_w , A_e , A_n и A_s зависят от схемы аппроксимации. Аналогичным образом получаем уравнение для v – компоненты скорости.

Теперь необходимо проделать аналогичную процедуру для сетевой задачи. Для представления сети используется ориентированный граф. Дискретизация сетевой части представлена в следующем виде



Рис. 2. Схема дискретизации сетевой части задачи.

Уравнение сохранения массы для начального узла:

$$\sum_i \nabla_{il} \cdot xl_in_i - xl = Q_{In} \quad (8)$$

Уравнение сохранения массы для конечного узла:

$$-\sum_i \nabla_{il} \cdot xl_out_i + xl = Q_{Out} \quad (9)$$

Уравнение сохранения импульса на ветви:

$$\rho u_{in} \cdot u_t \cdot f - \rho u_{out} \cdot u_t \cdot f - \left(\frac{\lambda \cdot l}{d} + \xi \right) \cdot \frac{\rho \cdot u_t^2}{2} \cdot f = (P_{out} - P_{in}) \cdot f, \quad (10)$$

где xl – расход на ветви, xl_in_i – расход на ветви, инцидентной начальному узлу (исключая рассматриваемую ветвь), xl_out_i – то же самое относительно конечного узла, u_t – средняя скорость на ветви. В зависимости от направления течения жидкости u_{out} или u_{in} принимаются равными u_t , а другое слагаемое можно представить в виде соответствующего расхода.

$$\begin{cases} \text{если } u_t > 0, \text{ то } u_{out} = u_t \text{ и } \rho u_{in} \cdot f = xl_in = \sum_i \nabla_{il} \cdot xl_in_i \\ \text{если } u_t < 0, \text{ то } u_{in} = u_t \text{ и } \rho u_{out} \cdot f = xl_out = \sum_i \nabla_{il} \cdot xl_out_i \end{cases} \quad (11)$$

Далее представляем уравнение в виде:

$$A_p \cdot u_t + A_w \cdot u_{in} - A_e \cdot u_{out} = (P_{out} - P_{in}) \cdot f, \quad (12)$$

где A_p , A_w , и A_e – зависят от направления и режима течения.

Приходим к выводу, что в общем виде уравнения можно записать как:

$$A_p \cdot u_p + \sum_{i=1}^N A_i \cdot u_i = (P_{out} - P_{in}) \cdot f + b, \quad (13)$$

где для двухмерной части задачи $N = 4$, а для сетевой части $N = 2$, b – правая часть и уравнение сохранения массы:

$$\sum_i x l_i = \sum_i \rho u_i \cdot f_i = Q, \quad (14)$$

где для двухмерной части задачи $x l_i$ – расходы через грани контрольного объема, а для сетевой расходы на ветвях инцидентных узлу, взятых с соответствующим знаком.

Связывая между собой сетевую и двухмерную части задачи необходимо увязать между собой давления в стыкующихся узлах и расходы в стыкующихся ветвях. Был предложен алгоритм стыковки с использованием стыковочных ветвей с нулевым сопротивлением (Рис. 3).

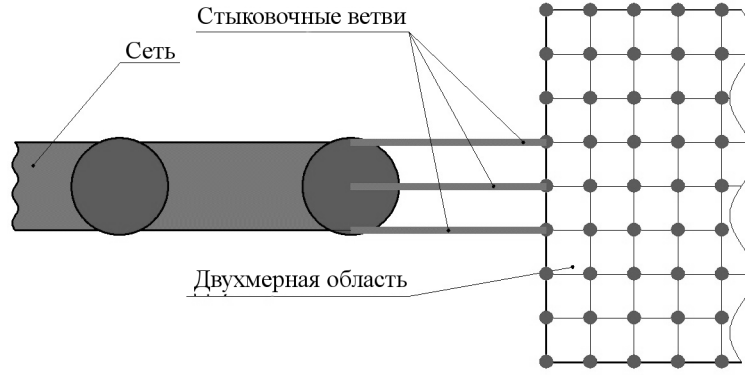


Рис. 3. Вариант стыковки с использованием стыковочных ветвей.

В этом случае мы задаем набор одномерных ветвей, с нулевым сопротивлением и задаем на них усреднённый расход. Такой вариант стыковки удобен тем, что отпадёт необходимость выравнивать давление, так как оно рассчитывается.

$$x l_i = \frac{x l_t + \sum_i x l_i^{2D}}{2 \cdot N} \cdot \alpha_i, \quad u_i = \frac{x l_i}{\rho_i \cdot f_i} \quad (15)$$

Здесь: i – номер стыковочной ветви, $x l_i$ – расход в i -й ветви, $x l_t$ – расход в трубе, $x l_i^{2D}$ – расход в ветвях стыковочной грани 2D области, N – количество узлов, α_i – коэффициент необходимый для формирования профиля течения на выходе из трубы. Теперь, получив все необходимые уравнения, переходим к алгоритму счета.

1. Задаем начальное поле давления.
2. Решаем уравнение сохранения импульса для получения полей скоростей.
3. Получив приблизительное поле скоростей, решаем уравнение для определения поправки давления.
4. Находим новые значения поля давления добавлением поправки давления к полученному ранее.

5. Вычисляем поправку скорости.
6. Проводим расчет значений расходов и скоростей на стыковочных ветвях.
7. Принимаем новые значения давления, возвращаемся к шагу 2 и повторяем процедуру до сходимости решения.

В итоге получаем модификацию SIMPLE алгоритма, в которой разрешаются двухмерная и сетевая задача. К особенностям этого алгоритма относится то, что уравнения для расчета скоростей в двухмерной области и сетевой части решаются разными методами. В то же время получаем общую систему уравнений для расчета давления.

Тестирование алгоритма.

Для того чтобы проверить алгоритм, в программе σ Flow [2] была построена модель, геометрия которой представлена на Рис 4.

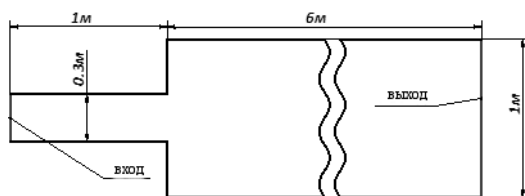


Рис. 4. Геометрия тестовой задачи.

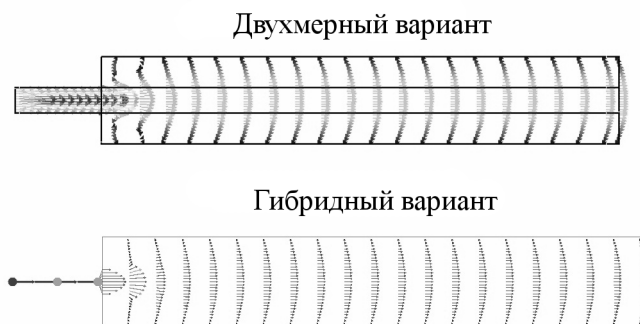


Рис. 5. Постановка задачи для стыковки и получившееся поле скоростей.

На входе задавался ударный профиль скорости с расходом равным 0.0314 кг/с. Физические свойства жидкости были приняты следующими: плотность – 1 кг/м³, динамическая вязкость 0.004 Па·с. Для гибридного варианта была построена аналогичная геометрия, в которой узкая часть была взята в виде сети Рис 5.

На Рис. 6 приведены графики сравнения профилей скорости на различном расстоянии от входа в широкий канал. Как видно из этих графиков профили скорости практически совпадают. Отсюда можно сделать вывод, что алгоритм стыковки работает корректно.

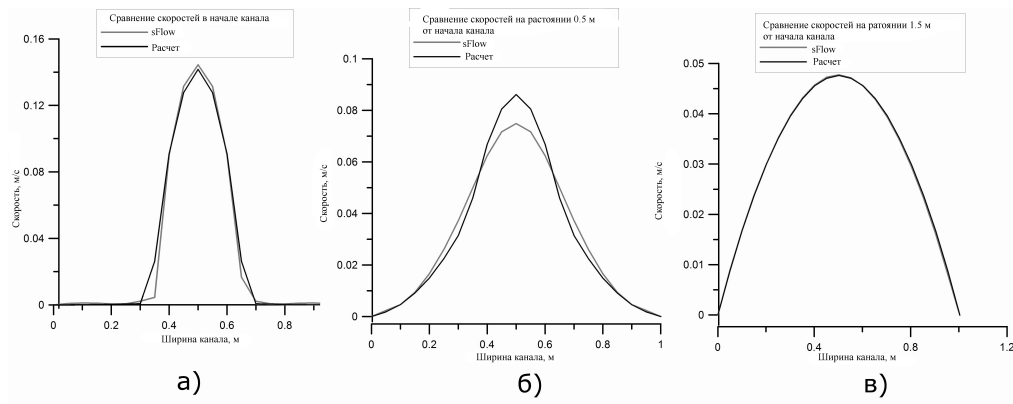


Рис. 6. Сравнение скорости на удалении от входа в широкую часть канала: а) на входе в широкую часть канала, б) на расстоянии 0.5м, в) на расстоянии 1.5м.

Заключение

Был разработан алгоритм, позволяющий разрешать как сетевые, так и двухмерные задачи. Проведена проверка работоспособности алгоритма на разрешении гибридной задачи.

Список литературы

- [1] ПАТАНКАР С. Численные методы решения задач теплообмена и динамики жидкости. // С. Патанкар М.: Энергоатомиздат, 1984 – 124с.
- [2] ДЕКТЕРЕВ А.А., ГАВРИЛОВ А.А., ХАРЛАМОВ Е.Б., ЛИТВИНЦЕВ К.Ю. ‘Использование программы σ Flow для численного исследования технологических объектов’ // Вычислительные технологии. 2003. Т. 8, Ч. 1, С. 250-255.