

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Работа допущена к защите
Директор ВШТМиМФ,
д.ф.-м.н., чл.-корр. РАН
_____ А. М. Кривцов
« ____ » _____ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ДИНАМИКИ НЕСЖИМАЕМОЙ ВЯЗКОЙ ЖИДКОСТИ

по направлению подготовки

01.03.03 «Механика и математическое моделирование»

направленность

01.03.03_01 Механика и математическое моделирование сред с
микроструктурой

Выполнил

студент гр. 5030103/90101

Е.В. Михеев

Руководитель

Доцент ВШТМиМФ,

к.ф.-м.н.

К.П. Фролова

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

Физико-механический институт

Высшая школа теоретической механики и математической физики

УТВЕРЖДАЮ

Директор ВШТМиМФ

А. М. Кривцов

«__» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Михееву Евгению Викторовичу, гр. 5030103/90101

1. Тема работы: Численное моделирование динамики несжимаемой вязкой жидкости.
2. Срок сдачи студентом законченной работы: 05.06.2023
3. Исходные данные по работе: справочная литература, актуальные научные публикации по теме работы.
4. Содержание работы (перечень подлежащих разработке вопросов): разработка численного метода решения задачи, программная реализация алгоритма, внедрение графического пользовательского интерфейса, обработка и верификация результатов.
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено.
6. Консультанты по работе: отсутствуют.
7. Дата выдачи задания: 27.02.2023

Руководитель ВКР: К. П. Фролова, доцент ВШТМиМФ, к.ф.-м.н.

Задание принял к исполнению: 27.02.2023

Студент: Е. В. Михеев

РЕФЕРАТ

На 67 с., 27 рисунков, 0 таблиц, 16 источников

НЕСЖИМАЕМАЯ ВЯЗКАЯ ЖИДКОСТЬ, МЕТОД КОНЕЧНЫХ ОБЪЕМОВ, УРАВНЕНИЯ НАВЬЕ-СТОКСА, C++

В данной работе рассматривается задача разработки решателя системы уравнений Навье-Стокса несжимаемой вязкой жидкости в динамической двумерной постановке методом конечных объемов на основе неявной схемы интегрирования по времени. В работе разработан численный метод с применением итерационного процесса для удовлетворения уравнениям, описывающим задачу, на каждом шаге интегрирования. Для реализации разработанного алгоритма используется ЯП C++ 17-й генерации и инструменты разработки под ОС *Windows* от компании *Microsoft*. Произведена валидация решателя на существующие аналитические решения, пакет численного моделирования *Ansys Fluent* и результаты, изложенные в научных статьях. Для взаимодействия с решателем разработан современный графический пользовательский интерфейс, содержащий все необходимые настройки решателя и возможности отслеживания процесса решения задачи. Таким образом, разработанная в ходе работы программа представляет собой прототип расчетного ПО (что актуально в контексте потребности отечественного рынка расчетов в современном ПО), которое может применяться для решения задач нестационарной двумерной гидродинамики и при необходимости может быть расширено на более широкий класс задач.

THE ABSTRACT

67 pages, 27 figures, 0 tables, 16 sources

INCOMPRESSIBLE VISCID FLUID, FINITE VOLUME METHOD, NAVIER-STOKES EQUATIONS, C++

This paper considers the problem of solver development for Navier-Stokes system of equations in transient 2-dimensional case for incompressible viscid fluid by the finite volume method with implicit time integration. This paper describes the developed numerical method at each step, including the detailed description of iterative process on each time step, which is provided in order to satisfy the equations with prescribed accuracy. C++17 in couple with Microsoft development tools were used to implement the developed algorithm. Final program is validated on existing analytical solutions, numerical solutions in CAE Ansys Fluent and results, described in science articles. Also, modern graphical user interface with all necessary options and residuals control was developed. Thus, the developed software represents the prototype of engineering software, which can be applied to solution of problems of transient 2-dimensional hydrodynamics and can be extended for wider number of problems.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ СПРАВКА.....	8
1.1 Вывод уравнений гидродинамики вязкой жидкости	8
1.1.1 Уравнение баланса количества движения	8
1.1.2 Закон сохранения массы.....	9
1.1.3 Итоговая система интегро-дифференциальных уравнений Навье- Стокса	10
1.2 Алгоритм интегрирования системы уравнений Навье-Стокса	10
1.2.1 Постановка задачи	11
1.2.2 Общий вид численного алгоритма решения задачи.....	12
1.2.3 Доказательство сходимости метода	14
1.2.4 Пример применения алгоритма.....	15
1.3 Численная аппроксимация системы уравнений Навье-Стокса в интегральной форме.....	17
1.3.1 Явный шаг схемы, часть 1.....	19
1.3.2 Явный шаг схемы, часть 2.....	20
1.3.3 Неявный шаг по X, часть 1	25
1.3.4 Неявный шаг по X, часть 2	29
1.3.5 Неявный шаг по Y.....	31
1.3.6 Инкрементальный шаг	35
1.4 Граничные условия	35
1.4.1 Граничные условия I рода	36
1.4.2 Граничные условия II рода.....	37
1.4.3 Граничные условия по давлению	38
ГЛАВА 2. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ.....	41
2.1 Структура.....	41
2.1.1 Класс <i>Node</i>	41
2.1.2 Класс <i>Cell</i>	42

2.1.3 Класс <i>NavierStokes</i>	43
2.2 Особенности	45
2.2.1 Параллельное выполнение задач.....	45
2.2.2 Асинхронность	46
ГЛАВА 3. РЕЗУЛЬТАТЫ.....	48
3.1 Течение в плоском канале	48
3.2 Обтекание препятствий в канале.....	55
3.2.1 Обтекание цилиндра круглого сечения.....	55
3.2.2 Обтекание цилиндра квадратного сечения.....	58
ЗАКЛЮЧЕНИЕ	61
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	63
Приложение 1. Сравнение динамики обтекания цилиндра квадратного сечения в разработанном ПО и <i>Ansys</i>	65

ВВЕДЕНИЕ

Решение задач гидродинамики является одной из областей механики, где применение аналитических методов решения редко представляется возможным в силу необходимости решения нелинейных по пространству дифференциальных уравнений в частных производных. Несмотря на то, что уравнения динамики жидкости часто сводятся к системе уравнений Навье-Стокса, описывающим поведение несжимаемых жидкостей (к которой в инженерных задачах гидродинамики считается вода), даже они имеют аналитическое решение лишь для крайне узкого класса случаев, таких как течение жидкости в трубе или в емкости с движущимися стенками. Но даже эти задачи решаются аналитически лишь в статической постановке, то есть методами математической физики можно рассчитать лишь устоявшееся движение жидкости, но не проследить его развитие.[6]

Тем не менее, развитие вычислительного оборудования и численных методов сделало возможным решение задач гидродинамики методами численного моделирования, ведущим из которых для задач, подразумевающих эйлерову постановку, является метод конечных объемов (МКО).[10] МКО на сегодняшний день поддерживается всеми ведущими вычислительными пакетами, способными решать задачи гидродинамики. Идея метода заключается в разбиении расчетной области на фиксированные (речь не идет о вариациях метода, таких как ALE) объемы, для каждого из которых решается система уравнений в интегральной форме, из чего в итоге складывается картина течения во всей расчетной области.

Целями данной выпускной квалификационной работы является разработка решателя системы уравнений Навье-Стокса в динамической постановке, а также создание для него графического пользовательского интерфейса, позволяющего производить настройки решателя без манипуляций с кодом, то есть, в конечном итоге, создание прототипа расчетной программы для решения двумерных задач нестационарной гидродинамики.

В результате работы решены следующие задачи:

а) Разработан решатель системы уравнений Навье-Стокса. Для разработки выбран язык программирования C++ в силу высокой вычислительной сложности решения нестационарного нелинейного двумерного дифференциального уравнения в частных переменных.

б) Произведена верификация работы решателя на существующие аналитические решения, результаты экспериментов и расчеты в программе вычислительной гидродинамики *Ansys Fluent*.

в) Разработан графический пользовательский интерфейс (UI) для комфортного взаимодействия пользователя с решателем. Также на основе разработки UI описаны некоторые особенности написания пользовательских программ, задействующих большие вычислительные ресурсы.

Глава 1. Теоретическая справка

1.1 Вывод уравнений гидродинамики вязкой жидкости

Система уравнений Навье-Стокса несжимаемой вязкой жидкости состоит из уравнения баланса количества движения и уравнения баланса. Общепринятый вид данной системы приведен в (1.1). [2]

$$\begin{cases} \frac{\partial \underline{v}}{\partial t} + (\underline{v} \cdot \nabla) \underline{v} = -\frac{1}{\rho} \nabla p + \nu \Delta \underline{v} + \underline{f} \\ \nabla \cdot \underline{v} = 0 \end{cases} \quad (1.1)$$

где \underline{v} – вектор скорости;

p – давление;

ρ – плотность;

ν – кинематическая вязкость;

\underline{f} – вектор массовых сил.

Однако в данной работе используется несколько измененная, консервативная форма системы (1.1), вывод которой приведен ниже.

1.1.1 Уравнение баланса количества движения

Рассмотрим некоторый произвольный объем жидкости. Для него закон баланса количества энергии в интегральной форме будет иметь вид (1.2) [5].

$$\frac{d}{dt} \int_V \rho \underline{v} dV = \int_V \rho \underline{f} dV + \int_{\partial V} \underline{n} \cdot \underline{T} dS - \int_{\partial V} (\underline{n} \cdot \rho \underline{v}) \underline{v} dS \quad (1.2)$$

где $\underline{T} = -p\underline{E} + \mu \nabla \underline{v}$ – тензор напряжений;

\underline{n} – нормаль к поверхности рассматриваемого объема;

V – непосредственно рассматриваемый объем;

∂V – граница рассматриваемого объема;

μ – динамическая вязкость;

$\underline{f} = 0$ – массовые силы в дальнейшем приняты несущественными.

Решение выполняется в эйлеровой постановке – все объемы постоянны во времени, их узлы не перемещаются, исследуется поле скоростей во времени на постоянной сетке.

Для дальнейших преобразований использованы следующие предположения:

– Жидкость несжимаема, то есть $\rho = const$ в пространстве и времени.

– Рассматриваемый объем мал, поэтому скорости во всех его точках можно принять равными в данный момент времени, таким образом $\frac{d}{dt} \int_V \rho \underline{v} dV \approx \rho V \dot{\underline{v}}$.

Используя данные предположения, (1.2) может быть переписано как (1.3). Проекция на оси двумерной декартовой системы координат приведены в (1.4).

$$0 \approx -\frac{d\underline{v}}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p \underline{n} dS + \underline{v} \int_{\partial V} \underline{n} \cdot \nabla \underline{v} dS - \int_{\partial V} \underline{n} \cdot \underline{v} \underline{v} dS \right] \quad (1.3)$$

$$0 =$$

$$= -\frac{du}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_1 dS + \underline{v} \int_{\partial V} n_1 \frac{\partial u}{\partial x} + n_2 \frac{\partial u}{\partial y} dS - \int_{\partial V} (n_1 u u + n_2 v u) dS \right] \quad (1.4)$$

$$0 =$$

$$= -\frac{dv}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_2 dS + \underline{v} \int_{\partial V} n_1 \frac{\partial v}{\partial x} + n_2 \frac{\partial v}{\partial y} dS - \int_{\partial V} (n_1 u v + n_2 v v) dS \right]$$

где u – компонента скорости, направленная по оси X декартовой СК;

v – компонента скорости, направленная по оси Y декартовой СК;

p – гидростатическое давление;

n_1 – компонента нормали к поверхности объема, направленная по оси X;

n_2 – компонента нормали к поверхности объема, направленная по оси Y.

1.1.2 Закон сохранения массы

Локальная форма закона сохранения массы приведена в выражении (1.5).

$$\frac{d\rho}{dt} + \nabla \cdot (\rho \underline{v}) = 0 \quad (1.5)$$

В предположении постоянства плотности в пространстве и времени, выражение выше примет вид (1.6), называемый уравнением неразрывности.

$$\nabla \cdot \underline{v} = 0 \quad (1.6)$$

Данное выражение является локальным, то есть должно выполняться в каждой точке пространства. Численный алгоритм, основываясь на принципах численной аппроксимации производных, будет требовать выполнение данного уравнения в каждом конечном объеме пространства.

1.1.3 Итоговая система интегро-дифференциальных уравнений Навье-Стокса

Объединяя (1.4) и (1.6) получена система (1.7) из трех скалярных интегро-дифференциальных уравнений.

$$\begin{cases} 0 = -\frac{du}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_1 dS + \nu \int_{\partial V} n_1 \frac{\partial u}{\partial x} + n_2 \frac{\partial u}{\partial y} dS - \int_{\partial V} (n_1 u u + n_2 \nu u) dS \right] \\ 0 = -\frac{dv}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_2 dS + \nu \int_{\partial V} n_1 \frac{\partial v}{\partial x} + n_2 \frac{\partial v}{\partial y} dS - \int_{\partial V} (n_1 u v + n_2 \nu v) dS \right] \\ 0 = -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \end{cases} \quad (1.7)$$

Для решения данной системы применен алгоритм с расщеплением на основе неявной схемы интегрирования.

1.2 Алгоритм интегрирования системы уравнений Навье-Стокса

В силу нелинейности задачи, используется итерационный алгоритм, то есть на каждом шаге по времени необходимо некоторое количество итераций, чтоб удовлетворить уравнению.

В приведенном ниже алгоритме используется следующая индексация:

- n – индекс шага по времени.
- k – индекс итерации в рамках текущего шага по времени.

1.2.1 Постановка задачи

Задача – решить дифференциальное уравнение (или систему дифференциальных уравнений) с заданными ГУ и НУ – (1.8).

$$\begin{aligned}\underline{\Delta}(f(t, \underline{r})) &= \underline{g}(t, \underline{r}) \\ f|_{t=0} &= \underline{a}(\underline{r}) \\ f|_{\partial V} &= \underline{b}(t, \underline{r})\end{aligned}\tag{1.8}$$

где $\underline{\Delta}$ – некоторый линейный (дифференциальный, интегральный или дифференциально-интегральный) оператор, в общем случае – векторный;

f – искомая функция, в общем случае – векторная;

$\underline{g}(t, x, y, z)$ – некоторая известная функция;

\underline{a} – вектор начальных условий;

\underline{b} – вектор граничных условий.

Решение задачи выполняется на двумерной ортогональной сетке с постоянным шагом по X и по Y . Пример расчетной сетки из финальной версии разработанной программы приведен на рисунке 1.1. Расчетная сетка с препятствиями на пути течения жидкости приведена на рисунке 1.2. В задаче массовые силы приняты несущественными, то есть $\underline{g}(t, \underline{r}) = 0$ и соответственно $\underline{\Delta}(f(t, \underline{r})) = 0$.

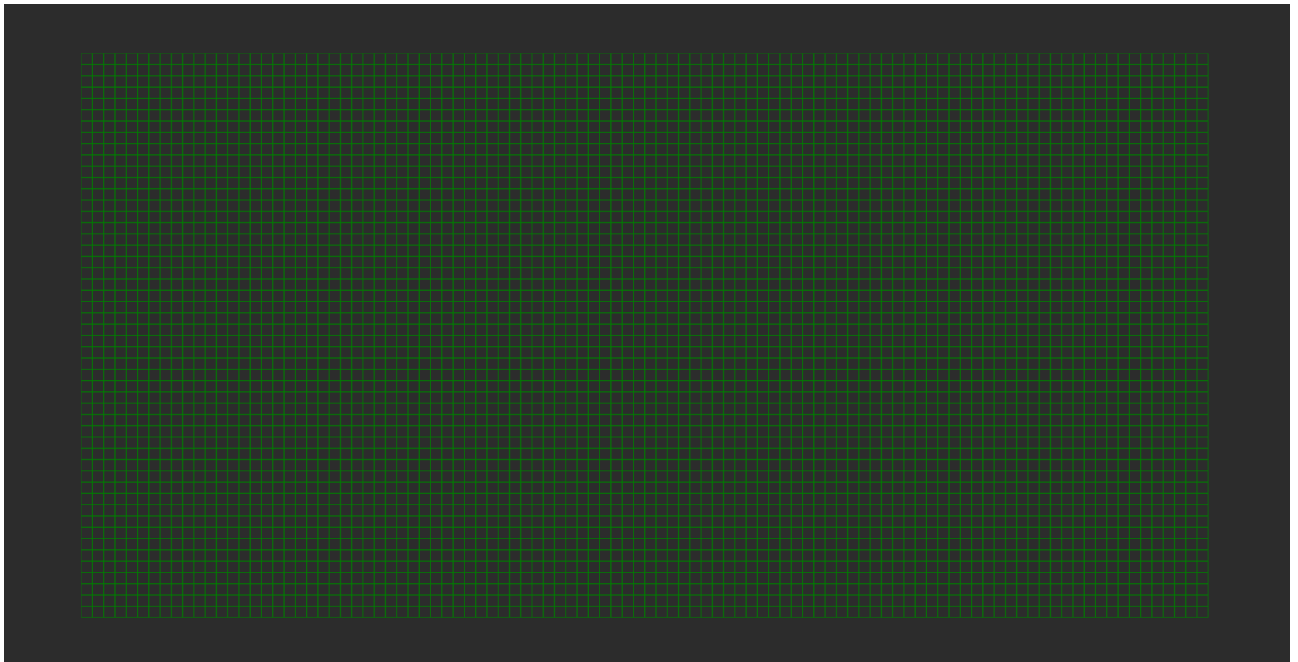


Рисунок 1.1 – Расчетная сетка

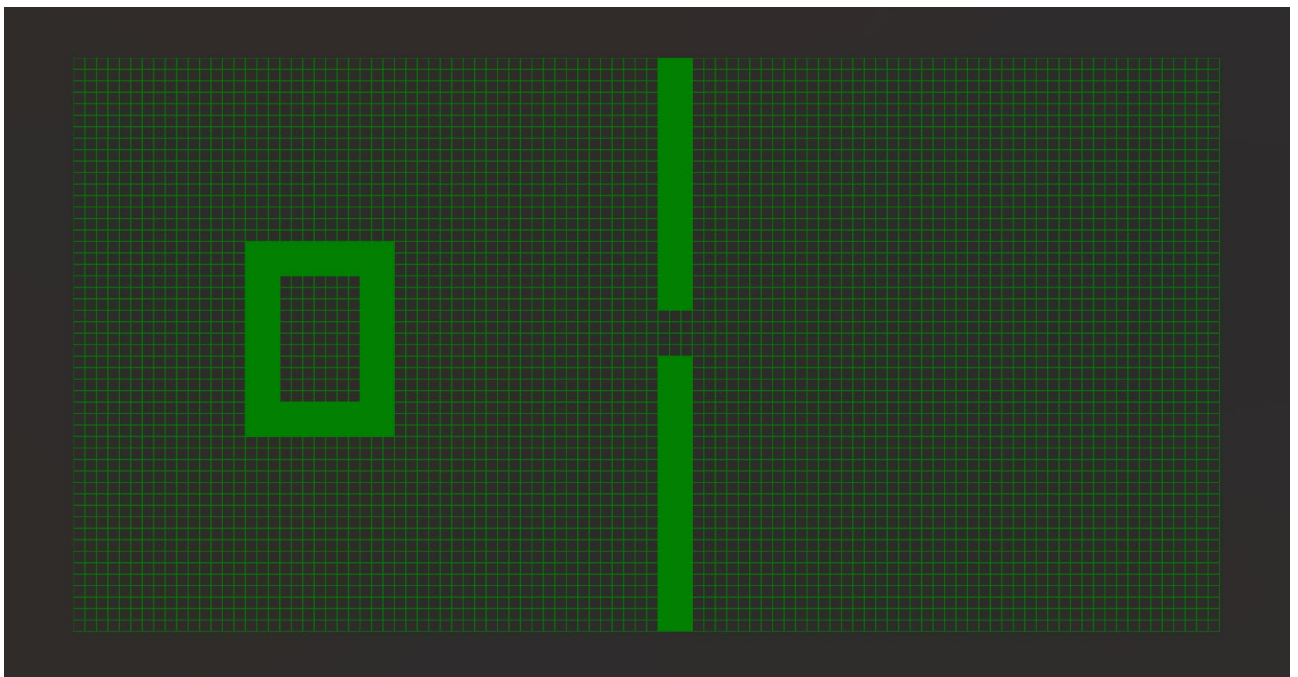


Рисунок 1.2 – Расчетная сетка с препятствиями

1.2.2 Общий вид численного алгоритма решения задачи

Алгоритм, используемый для интегрирования системы уравнений, является неявной схемой интегрирования с расщеплением по пространственным координатам. Общая схема одной итерации алгоритма на $(n+1)$ -ом шаге по времени приведена в системе (1.9) – (1.12).[4][9]

$$\underline{\xi}^k = \tau \underline{\Lambda}(f^k) \quad (1.9)$$

$$\underline{\hat{\xi}} - \underline{\xi}^k = \tau \underline{\Lambda}_1(\underline{\hat{\xi}}) \quad (1.10)$$

$$\underline{\xi}^{k+1} - \underline{\hat{\xi}} = \tau \underline{\Lambda}_2(\underline{\xi}^{k+1}) \quad (1.11)$$

$$\underline{f}^{k+1} = \underline{f}^k + \underline{\xi}^{k+1} \quad (1.12)$$

где τ – итерационный параметр;

$\underline{\Lambda}_1$ – часть оператора $\underline{\Lambda}$, содержащая частные производные по x и компоненты интегралов второго рода на поверхностях с нормальными, ориентированными вдоль оси X ;

$\underline{\Lambda}_2$ – часть оператора $\underline{\Lambda}$, содержащая частные производные по y и компоненты интегралов второго рода на поверхностях с нормальными, ориентированными вдоль оси X ;

\underline{f}_k – искомая функция в начале k -ой итерации;

$\underline{\xi}, \underline{\hat{\xi}}$ – невязки.

Так как решение производится на регулярной ортогональной сетке, из нее можно выделить набор вертикальных и горизонтальных «осей», которые пересекаются в узлах сетки.

Разделение оператора Λ на компоненты Λ_1 и Λ_2 , содержащие только производные по X и только производные по Y соответственно позволяет решать конечно-разностное уравнение

$$\underline{\hat{\xi}} - \underline{\xi}^k = \tau \underline{\Lambda}_1(\underline{\hat{\xi}}) \quad \text{на каждой из горизонтальных «осей»}, \text{ а потом использовать}$$

расчитанные значения $\underline{\hat{\xi}}$ для решения конечно-разностного уравнения

$$\underline{\xi}^{k+1} - \underline{\hat{\xi}} = \tau \underline{\Lambda}_2(\underline{\xi}^{k+1}) \quad \text{на каждой из вертикальных «осей»}. \text{ По итогам этих двух}$$

операций в каждом из узлов будет храниться обновленное значение невязки ξ^{k+1} .

Рассмотрим алгоритм пошагово:

– Шаг (1.9): производится вычисление невязки $\underline{\xi}_k$ уравнения в начале итерации. Данный шаг в дальнейшем будем называть «явным».

– Шаг (1.10): решается уравнение относительно невязки $\underline{\xi}$ на основании невязок, определенных на явном шаге. Данный шаг будем называть неявным по X .

– Шаг (1.11): решается уравнение относительно невязки $\underline{\xi}_{k+1}$ на основании невязок, определенных на неявном шаге по X . Данный шаг будем называть неявным по Y .

– Шаг (1.12): рассчитанные невязки $\underline{\xi}_{k+1}$ прибавляются к значению искомой функции известной на начало итерации. Данный шаг будем называть инкрементальным.

Таким образом итерационный процесс длится до тех пор, пока невязки, вычисленные в начале шага, деланные на итерационный параметр $(\frac{\xi_k}{\tau})$, не будут малы с точностью до заранее заданного $\varepsilon > 0$. В результате итерационного процесса на данном шаге по времени уравнение станет тождеством с точностью до некоторого сколь угодно малого числа: $\frac{\xi_k}{\tau} = \underline{\Lambda}(f_k) < \varepsilon$

1.2.3 Доказательство сходимости метода

Покажем, что итерационный процесс приводит к выполнению тождества $\underline{\Lambda}(f(t, \underline{r})) = 0$ на каждом шаге по времени.

Доказательство для упрощения записи приведено для случая, когда оператор $\underline{\Lambda}$ и неизвестная функция f являются скалярами. Доказательство для случаев векторных и тензорных функций производится аналогичным образом.

$$\begin{aligned}
 (1.10) \rightarrow \quad & \hat{\xi} - \tau \underline{\Lambda}_1(\hat{\xi}) = \xi^k \\
 & (I - \tau \underline{\Lambda}_1)\hat{\xi} = \xi^k \\
 & \hat{\xi} = (I - \tau \underline{\Lambda}_1)^{-1} \xi^k
 \end{aligned} \tag{1.13}$$

где I – единичный оператор: $I(f) = f$.

$$\begin{aligned}
(1.11) \rightarrow \quad & \xi^{k+1} - \tau\Lambda_2(\xi^{k+1}) = \hat{\xi} \\
& (I - \tau\Lambda_2)\xi^{k+1} = \hat{\xi} \\
& \xi^{k+1} = (I - \tau\Lambda_2)^{-1}\hat{\xi}
\end{aligned} \tag{1.14}$$

$$\begin{aligned}
(1.13)+(1.14) \rightarrow \quad & \xi^{k+1} = (I - \tau\Lambda_2)^{-1}(I - \tau\Lambda_1)^{-1}\xi^k \\
& \xi^k = (I - \tau\Lambda_1)(I - \tau\Lambda_2)\xi^{k+1} \\
& (I - \tau\Lambda_1 - \tau\Lambda_2 + \tau^2\Lambda_1\Lambda_2)\xi^{k+1} = \xi^k \\
& \xi^{k+1} - \tau \underbrace{(\Lambda_1 + \Lambda_2)(\xi^{k+1})}_{\Lambda(\xi^{k+1})} + \tau^2\Lambda_1\Lambda_2(\xi^{k+1}) = \underbrace{\xi^k}_{\tau\Lambda(f^k)}
\end{aligned} \tag{1.15}$$

Учитывая, что:

$$\xi^{k+1} = f^{k+1} - f^k$$

$\Lambda(\xi^{k+1}) = \Lambda(f^{k+1} - f^k) = \Lambda(f^{k+1}) - \Lambda(f^k)$ – в силу линейности оператора.

Выражение (1.15) можно переписать как:

$$\begin{aligned}
f^{k+1} - f^k - \tau\Lambda(f^{k+1}) + \tau\Lambda(f^k) + \tau^2\Lambda_1\Lambda_2(\xi^{k+1}) &= \tau\Lambda(f^k) \\
\frac{f^{k+1} - f^k}{\tau} &= \Lambda(f^{k+1}) - \tau\Lambda_1\Lambda_2(f^{k+1} - f^k) \\
\underbrace{\frac{f^{k+1} - f^k}{\tau}}_{\xi^{k+1}} &
\end{aligned}$$

При сходящемся итерационном процессе с какого-то номера итерации m :

$$f^{m+1} - f^m = \xi^{m+1} < \varepsilon \rightarrow 0, \text{ тогда } \frac{\xi^{m+1}}{\tau} \rightarrow 0 \text{ и } \Lambda(f^{m+1}) = 0$$

Что и требовалось доказать – исходное ДУ на данном шаге по времени выполнено.

1.2.4 Пример применения алгоритма

Приведем данный алгоритм для системы уравнений в локальной форме

(1.1). Запись системы сразу приведена в координатной форме.

Здесь и далее:

– i – индекс узла интегрирования вдоль оси X .

– j – индекс узла интегрирования вдоль оси Y.

– m – индекс рассматриваемой ячейки (конечного объема).

Ниже записаны оператор Λ и его расщепления по X и Y, используемые для применения в алгоритме интегрирования. Так как операцию дифференцирования по времени нельзя отнести ни к оператору по X, ни к оператору по Y, частные производные $\frac{\partial}{\partial t}$ разделены между операторами Λ_1 и Λ_2 поровну.

$$\Lambda(u, v, p, t) = \begin{cases} -\frac{du}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_1 dS + v \int_{\partial V} n_1 \frac{\partial u}{\partial x} + n_2 \frac{\partial u}{\partial y} dS - \int_{\partial V} (n_1 u u + n_2 v u) dS \right] \\ -\frac{dv}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_2 dS + v \int_{\partial V} n_1 \frac{\partial v}{\partial x} + n_2 \frac{\partial v}{\partial y} dS - \int_{\partial V} (n_1 u v + n_2 v v) dS \right] \\ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \end{cases}$$

$$\Lambda_1(u, v, p, t) = \begin{cases} -\frac{1}{2} \frac{du}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_1 dS + v \int_{\partial V} n_1 \frac{\partial u}{\partial x} dS - \int_{\partial V} n_1 u u dS \right] \\ -\frac{1}{2} \frac{dv}{dt} + \frac{1}{V} \left[v \int_{\partial V} n_1 \frac{\partial v}{\partial x} dS - \int_{\partial V} n_1 u v dS \right] \\ -\frac{\partial u}{\partial x} \end{cases}$$

$$\Lambda_2(u, v, p, t) = \begin{cases} -\frac{1}{2} \frac{du}{dt} + \frac{1}{V} \left[v \int_{\partial V} n_2 \frac{\partial u}{\partial y} dS - \int_{\partial V} n_2 v u dS \right] \\ -\frac{1}{2} \frac{dv}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p n_2 dS + v \int_{\partial V} n_2 \frac{\partial v}{\partial y} dS - \int_{\partial V} n_2 v v dS \right] \\ -\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \end{cases}$$

– Явный шаг:

$$\begin{cases} \frac{\xi_u^k}{\tau} = -\frac{du^k}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p^k n_1 dS + v \int_{\partial V} n_1 \frac{\partial u^k}{\partial x} + n_2 \frac{\partial u^k}{\partial y} dS - \int_{\partial V} (n_1 u^k u^k + n_2 v^k u^k) dS \right] \\ \frac{\xi_v^k}{\tau} = -\frac{dv^k}{dt} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} p^k n_2 dS + v \int_{\partial V} n_1 \frac{\partial v^k}{\partial x} + n_2 \frac{\partial v^k}{\partial y} dS - \int_{\partial V} (n_1 u^k v^k + n_2 v^k v^k) dS \right] \\ \frac{\xi_p^k}{\tau} = -\frac{\partial u^k}{\partial x} - \frac{\partial v^k}{\partial y} \end{cases}$$

– Неявный шаг по X:

$$\begin{cases} \frac{\hat{\xi}_u - \xi_u^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_u}{\Delta t} \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} \hat{\xi}_p n_1 dS + v \int_{\partial V} n_1 \frac{\partial \hat{\xi}_u}{\partial x} dS - \int_{\partial V} n_1 u_k \hat{\xi}_u dS \right] \\ \frac{\hat{\xi}_v - \xi_v^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_v}{\Delta t} + \frac{1}{V} \left[v \int_{\partial V} n_1 \frac{\partial \hat{\xi}_v}{\partial x} dS - \int_{\partial V} n_1 u_k \hat{\xi}_v dS \right] \\ \frac{\hat{\xi}_p - \xi_p^k}{\tau} = -\frac{\partial \hat{\xi}_u}{\partial x} \end{cases}$$

– Неявный шаг по Y :

$$\left\{ \begin{array}{l} \frac{\xi_u^{k+1} - \hat{\xi}_u}{\tau} = -\frac{1}{2} \frac{\xi_u^{k+1}}{\Delta t} + \frac{1}{V} \left[\nu \int_{\partial V} n_2 \frac{\partial \xi_u^{k+1}}{\partial y} dS - \int_{\partial V} n_2 v_k \xi_u^{k+1} dS \right] \\ \frac{\xi_v^{k+1} - \hat{\xi}_v}{\tau} = -\frac{1}{2} \frac{\xi_v^{k+1}}{\Delta t} + \frac{1}{V} \left[-\frac{1}{\rho} \int_{\partial V} \xi_p^{k+1} n_2 dS + \nu \int_{\partial V} n_2 \frac{\partial \xi_v^{k+1}}{\partial y} dS - \int_{\partial V} n_2 v_k \xi_v^{k+1} dS \right] \\ \frac{\xi_p^{k+1} - \hat{\xi}_p}{\tau} = -\frac{\partial \xi_v^{k+1}}{\partial y} \end{array} \right.$$

– Инкрементальный шаг:

$$\left\{ \begin{array}{l} u^{k+1} = u^k + \xi_u^{k+1} \\ v^{k+1} = v^k + \xi_v^{k+1} \\ p^{k+1} = p^k + \xi_p^{k+1} \end{array} \right.$$

– Конец итераций на $(n+1)$ -ом шаге по времени. Когда $\xi_u^{k+1}, \xi_v^{k+1}, \xi_p^{k+1} < \varepsilon$:

$$\left\{ \begin{array}{l} u^{n+1} = u^{k+1} \\ v^{n+1} = v^{k+1} \\ p^{n+1} = p^{k+1} \end{array} \right.$$

1.3 Численная аппроксимация системы уравнений Навье-Стокса в интегральной форме

Так как в работе система уравнений Навье-Стокса решается в интегральной форме, преобразуем выражения (1.4) и (1.6) для использования в алгоритме, описанном в п. 1.2.2.

На рисунке 1.3 приведена элементарная ячейка расчетной сетки, для которой выполняется аппроксимация системы уравнений Навье-Стокса.

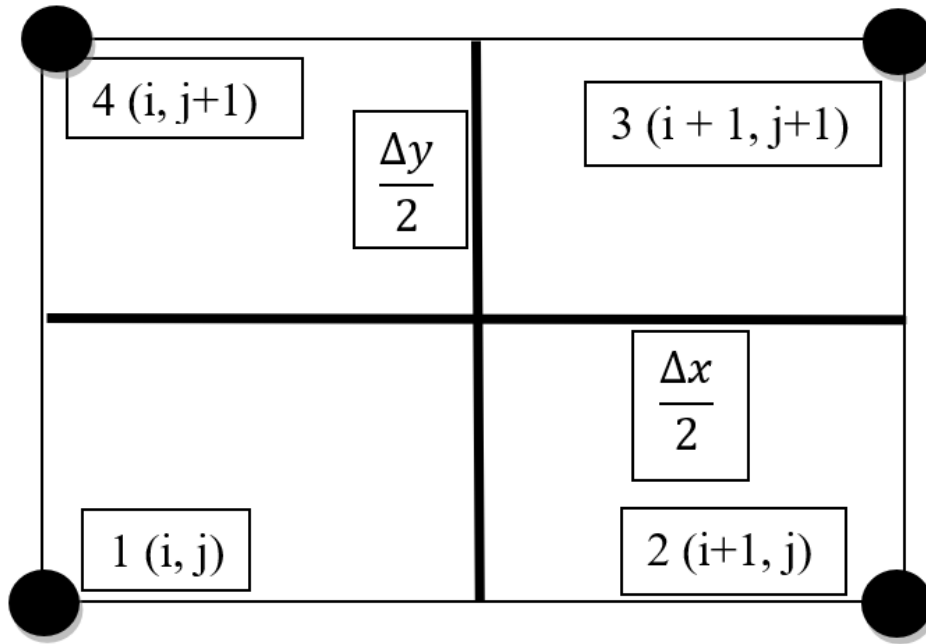


Рисунок 1.3 – Элементарная ячейка

Введена нумерация узлов от 1 до 4. Нумерация ведется против часовой стрелки. Такая нумерация в дальнейшем применена для всех выражений. Δx и Δy – линейные размеры ячейки вдоль осей X и Y соответственно.

Пусть ячейка обладает номером m . Порядок нумерации ячеек не важен. Вдоль оси X сетка имеет n_x ячеек, вдоль оси Y – n_y ячеек. Соответственно, сетка содержит $(n_x + 1)(n_y + 1)$ узлов.

Элементарная ячейка в свою очередь может быть разделена на 4 меньших объема, каждый из которых относится к соответствующему узлу. Разделение на рисунке 1.3 обозначено черными прямыми.

В алгоритме некоторые значения расчетных величин относятся к узлам, некоторые к ячейкам.

Узловые величины:

- u – горизонтальная компонента скорости.
- u_k – горизонтальная компонента скорости на k -й итерации.
- ξ_u – невязка первого уравнения системы (1.7).
- v – вертикальная компонента скорости.
- v_k – вертикальная компонента скорости на k -й итерации.
- ξ_v – невязка второго уравнения системы (1.7).

– V – величина объема «вокруг» узла.

Величины, относящиеся к ячейке:

– p – давление в ячейке.

– p_k – давление в ячейке на k -й итерации.

– ξ_p – невязка третьего уравнения системы (1.7).

Таким образом, в алгоритме скорости и сопутствующие им величины определяются в узлах, а давление определяется для всего конечного объема (элементарной ячейки).

1.3.1 Явный шаг схемы, часть 1

В выражениях (1.16) – (1.18) первый шаг схемы интегрирования применен к каждому из уравнений системы (1.7). В результате получены невязки в начале для каждого из уравнений, на основе которых производятся вычисления в последующих шагах.

$$\xi_u^k = \tau \left[-\frac{\partial u}{\partial t} + \frac{1}{V} F_u \right] \quad (1.16)$$

$$\xi_v^k = \tau \left[-\frac{\partial v}{\partial t} + \frac{1}{V} F_v \right] \quad (1.17)$$

$$\xi_p^k = \tau_p \left[-\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right] \quad (1.18)$$

где $F_u = -\frac{1}{\rho} \int_{\partial V} p n_1 dS + \nu \int_{\partial V} n_1 \frac{\partial u}{\partial x} + n_2 \frac{\partial u}{\partial y} dS - \int_{\partial V} (n_1 u u + n_2 v u) dS$;

$F_v = -\frac{1}{\rho} \int_{\partial V} p n_2 dS + \nu \int_{\partial V} n_1 \frac{\partial v}{\partial x} + n_2 \frac{\partial v}{\partial y} dS - \int_{\partial V} (n_1 u v + n_2 v v) dS$;

1.3.1.1 Аппроксимация нелинейных слагаемых

Рассмотрим функцию следующего вида:

$$f(u, v) = uv$$

Такие слагаемые содержатся в конвективных слагаемых уравнений (1.4). Их можно трактовать как «перенос скорости v скоростью u ». Из этого определения следует метод сеточной аппроксимации подобных слагаемых:

$$uv = \begin{cases} \bar{u}_{i,i+1} v_{i,j}, & \text{если } \bar{u}_{i,i+1} > 0 \\ \bar{u}_{i,i+1} v_{i+1,j}, & \text{если } \bar{u}_{i,i+1} < 0 \end{cases} \quad (1.19)$$

где $\bar{u}_{i,i+1} = \frac{u_{i+1,j} + u_{i,j}}{2}$.

То есть, если в центре ячейки горизонтальная скорость направлена вправо, то переноситься будет вертикальная скорость из левого узла и наоборот.

Тем не менее, выражение (1.19) неудобно для программной реализации, так как требует дополнительного ветвления кода, вследствие чего (1.19) в коде используется в форме (1.20).

$$uv = \frac{\bar{u}_{i,i+1} + |\bar{u}_{i,i+1}|}{2} v_{i,j} + \frac{\bar{u}_{i,i+1} - |\bar{u}_{i,i+1}|}{2} v_{i+1,j} \quad (1.20)$$

Математическая эквивалентность (1.19) и (1.20) очевидна и доказывается путем раскрытия знаков модуля.

1.3.2 Явный шаг схемы, часть 2

Выражения F_u и F_v в (1.16) и (1.17) представляют собой поверхностные (в двумерной постановке – контурные) интегралы второго рода, требующие аппроксимации для применения в численном алгоритме. Физический смысл такого интеграла – поток через границу объема.

Приведем способ численной аппроксимации контурного интеграла 2-го рода на примере прямоугольной ячейки в декартовой системе координат с указанными нормальными к сторонам ячейки, приведенной на рисунке 1.4. Нормали для каждой ячейки в расчете приняты внешними к контуру ячейки.

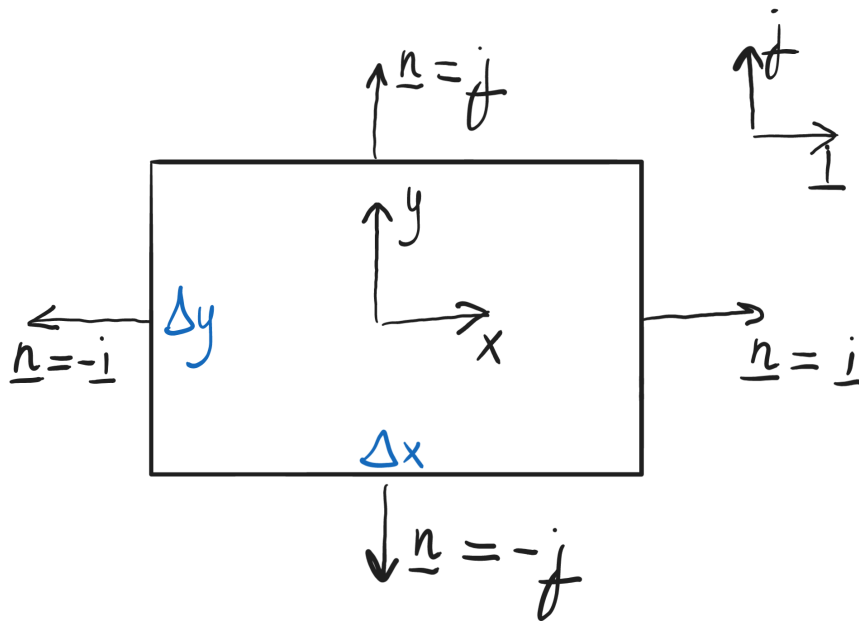


Рисунок 1.4 – Элементарная ячейка сетки с указанием нормалей

Пусть в ячейке определена некоторая векторная функция $\underline{f}(x, y)$. Тогда аппроксимация контурного интеграла 2-го рода для этой функции по контуру данной ячейки будет иметь вид (1.21).

$$\begin{aligned}
 \int_{\partial V} \underline{f}(x, y) \cdot \underline{n} dS &= \\
 &= \int_{\partial V} (f_x \underline{i} + f_y \underline{j}) \cdot (n_1 \underline{i} + n_2 \underline{j}) dS = \\
 &= \int_{\partial V} (f_x n_1 + f_y n_2) dS = f_y(x, y) \Delta x \Big|_{y=\frac{\Delta y}{2}} - f_y(x, y) \Delta x \Big|_{y=-\frac{\Delta y}{2}} + \\
 &+ f_x(x, y) \Delta y \Big|_{x=\frac{\Delta x}{2}} - f_x(x, y) \Delta y \Big|_{x=-\frac{\Delta x}{2}}
 \end{aligned} \tag{1.21}$$

Таким образом, для прямоугольного контура в декартовой системе координат контурный интеграл распадается на 4 слагаемых, относящихся к каждой из стенок. Аппроксимация производится в предположении, что длины сторон ячейки достаточно малы, чтобы изменением функции на них можно было бы пренебречь.

На рисунке 1.5 приведен расширенный фрагмент расчетной сетки, состоящий из 9 ячеек, в центре которого находится ячейка с рисунка 1.3. Желтым, зеленым, красным и голубым обозначены конечные объемы, относящиеся к каждому из узлов.

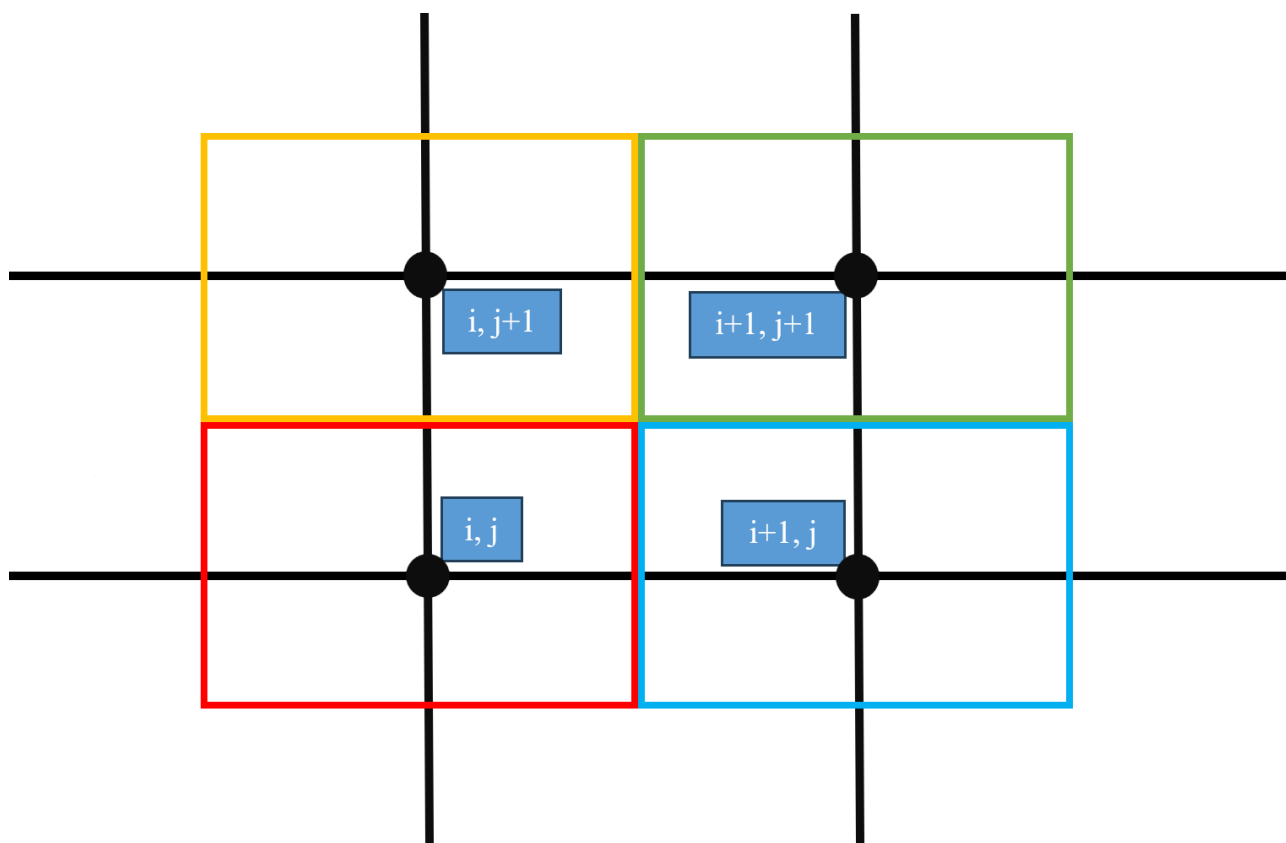


Рисунок 1.5 – Расширенный фрагмент расчетной сетки

Для вычисления невязок уравнений на явном шаге по формулам (1.16)-(1.18) для каждого узла необходимо раскрытие контурных интегралов F_u и F_v .

В программной реализации используется свойство аддитивности определенных интегралов. Программный алгоритм реализован так, что в начале итерации на явном шаге для каждого узла вводятся переменные, содержащие значение интегралов F_u и F_v , а затем идет цикл, перебирающий все ячейки, и для каждого из 4 узлов ячейки рассчитываются интегралы на стенках конечного объема, относящегося к данной ячейке. То есть, находясь в m -ой ячейке, в значение интеграла, относящегося к узлу (i, j) добавляются значения со стенок, отмеченных пунктиром на рисунке 1.6.

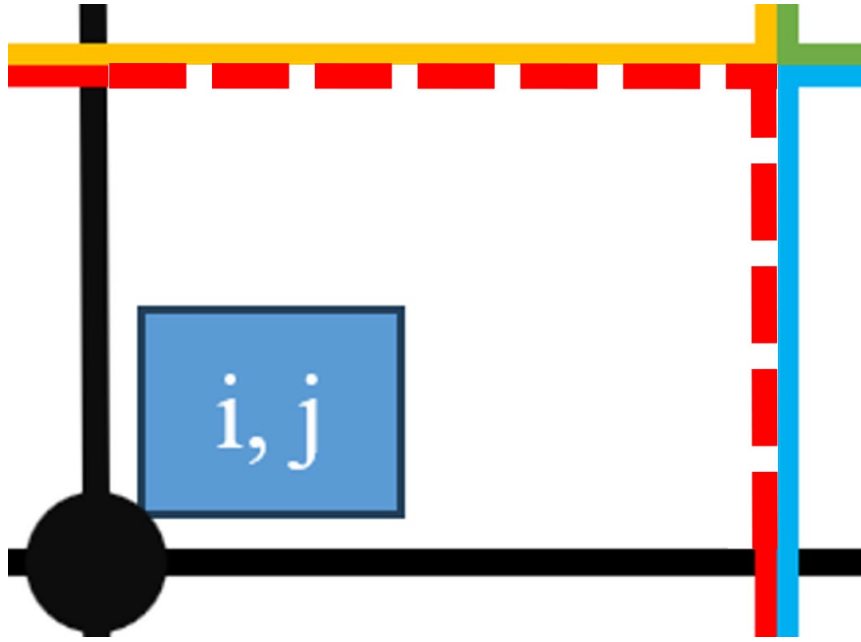


Рисунок 1.6 – К раскрытию контурных интегралов для узла (i, j)

Стоит также отметить, что рассчитанные значения интеграла на красных пунктирных границах сверху и справа также добавляются с изменением знака в значения интеграла для узлов $(i, j + 1)$ и $(i + 1, j)$ соответственно, так как поток подынтегрального вектора через эти стенки для соседних ячеек одинаков по модулю и отличается знаком в силу противоположных ориентаций нормали.

По итогам прохода циклом по всем ячейкам, во всех узлах собираются соответствующие контурные интегралы, причем при этом учтены узлы у границ области, которые имеют меньшее число краевых поверхностей.

Таким образом, получены следующие составляющие интегралов в ячейке:

$$\begin{aligned}
 F_{u,x_{12}} &= \frac{\Delta y}{2} \left[-\frac{p_m^k}{\rho} + v \frac{u_2^k - u_1^k}{\Delta x} - \frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} u_1^k - \frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} u_2^k \right] \\
 F_{u,x_{34}} &= \frac{\Delta y}{2} \left[-\frac{p_m^k}{\rho} + v \frac{u_3^k - u_4^k}{\Delta x} - \frac{\bar{u}_{34} + |\bar{u}_{34}|}{2} u_4^k - \frac{\bar{u}_{34} - |\bar{u}_{34}|}{2} u_3^k \right] \\
 F_{u,y_{14}} &= \frac{\Delta x}{2} \left[v \frac{u_4^k - u_1^k}{\Delta y} - \frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} u_1^k - \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} u_4^k \right] \\
 F_{u,y_{23}} &= \frac{\Delta x}{2} \left[v \frac{u_3^k - u_2^k}{\Delta y} - \frac{\bar{v}_{23} + |\bar{v}_{23}|}{2} u_2^k - \frac{\bar{v}_{23} - |\bar{v}_{23}|}{2} u_3^k \right] \\
 F_{v,x_{12}} &= \frac{\Delta y}{2} \left[v \frac{v_2^k - v_1^k}{\Delta x} - \frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} v_1^k - \frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} v_2^k \right] \\
 F_{v,x_{34}} &= \frac{\Delta y}{2} \left[v \frac{v_3^k - v_4^k}{\Delta x} - \frac{\bar{u}_{34} + |\bar{u}_{34}|}{2} v_4^k - \frac{\bar{u}_{34} - |\bar{u}_{34}|}{2} v_3^k \right]
 \end{aligned}$$

$$F_{v,y_{14}} = \frac{\Delta x}{2} \left[-\frac{p_m^k}{\rho} + v \frac{v_4^k - v_1^k}{\Delta y} - \frac{v_{14} + |\bar{v}_{14}|}{2} v_1^k - \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} v_4^k \right]$$

$$F_{v,y_{23}} = \frac{\Delta x}{2} \left[-\frac{p_m^k}{\rho} + v \frac{v_3^k - v_2^k}{\Delta y} - \frac{v_{23} + |\bar{v}_{23}|}{2} v_2^k - \frac{\bar{v}_{23} - |\bar{v}_{23}|}{2} v_3^k \right]$$

где $F_{u,x_{12}}$ – составляющая интеграла F_u на вертикальной стенке между узлами 1 и 2 (рисунок 1.3).

$F_{u,y_{14}}$ – составляющая интеграла F_u на горизонтальной стенке между узлами 1 и 4 (рисунок 1.3).

Остальные наименования произведены аналогичным образом. Здесь и далее, нижними индексами 1,2,3,4 будет обозначаться принадлежность величины к узлу в соответствии с индексацией, введенной на рисунке 1.3.

Затем вычисленные интегралы в соответствии с вышеописанным правилом прибавляются к значению невязок в узлах текущей ячейки:

$$\xi_{u,1}^k += \tau \frac{1}{V} (F u_{x_{12}} + F u_{y_{14}})$$

$$\xi_{u,2}^k += \tau \frac{1}{V} (-F u_{x_{12}} + F u_{y_{23}})$$

$$\xi_{u,3}^k += \tau \frac{1}{V} (-F u_{x_{34}} - F u_{y_{23}})$$

$$\xi_{u,4}^k += \tau \frac{1}{V} (F u_{x_{34}} - F u_{y_{14}})$$

$$\xi_{v,1}^k += \tau \frac{1}{V} (F v_{x_{12}} + F v_{y_{14}})$$

$$\xi_{v,2}^k += \tau \frac{1}{V} (-F v_{x_{12}} + F v_{y_{23}})$$

$$\xi_{v,3}^k = \tau \frac{1}{V} (-F v_{x_{34}} - F v_{y_{23}})$$

$$\xi_{v,4}^k = \tau \frac{1}{V} (F v_{x_{34}} - F v_{y_{14}})$$

После прохода циклом по всем ячейкам, в каждом узле сформированы контурные интегралы F_u и F_v .

В конце явного шага производится проход циклом по всем узлам и к значениям невязок для каждого узла добавляются неинтегральные слагаемые:

$$\xi_{u,1}^k += -\frac{u_1^k - u_1^n}{\Delta t} \tau$$

$$\xi_{u,2}^k += -\frac{u_2^k - u_2^n}{\Delta t} \tau$$

$$\begin{aligned}\xi_{u,3}^k &+ = -\frac{u_3^k - u_3^n}{\Delta t} \tau \\ \xi_{u,4}^k &+ = -\frac{u_4^k - u_4^n}{\Delta t} \tau \\ \xi_{v,1}^k &+ = -\frac{v_1^k - v_1^n}{\Delta t} \tau \\ \xi_{v,2}^k &+ = -\frac{v_2^k - v_2^n}{\Delta t} \tau \\ \xi_{v,3}^k &+ = -\frac{v_3^k - v_3^n}{\Delta t} \tau \\ \xi_{v,4}^k &+ = -\frac{v_4^k - v_4^n}{\Delta t} \tau\end{aligned}$$

Таким образом, во всех узлах сформированы невязки для первого и второго уравнений системы (1.7).

Наконец, проходом по всем ячейкам формируются невязки для третьего уравнения системы (1.7):

$$\xi_{p,m}^k = \tau_p \left[-\frac{\bar{u}_{23} - \bar{u}_{14}}{\Delta x} - \frac{\bar{v}_{34} - \bar{v}_{12}}{\Delta y} \right]$$

Для расчета частных производных взяты средние значения скоростей на соответствующих стенках, таким образом, выражение в скобке представляет собой дивергенцию вектора скорости в центре ячейки.

Явный шаг итерации завершен.

После явного шага производится проверка малости невязок. Если $\xi_u^{k+1}, \xi_v^{k+1}, \xi_p^{k+1} < \varepsilon$, то:

$$\begin{cases} u^{n+1} = u^{k+1} \\ v^{n+1} = v^{k+1} \\ p^{n+1} = p^{k+1} \end{cases}$$

Происходит выход из итерационного процесса и начинается следующий шаг по времени.

1.3.3 Неявный шаг по X, часть 1

Выделим из первого и второго уравнений системы (1.7) только те слагаемые, в которых содержится вертикальная (сонаправленная с осью X) компонента нормали – n_1 . Из третьего уравнения системы выделим частную производную по X. Эти слагаемые составляют часть оператора по X.

$$\left\{ \begin{array}{l} \frac{\hat{\xi}_u - \xi_u^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_u}{\Delta t} + \frac{1}{V} \int_{\partial V} n_1 \left(-\frac{\hat{\xi}_p}{\rho} - u^k \hat{\xi}_u + v \frac{\partial \hat{\xi}_u}{\partial x} \right) dS \\ \frac{\hat{\xi}_v - \xi_v^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_v}{\Delta t} + \frac{1}{V} \int_{\partial V} n_1 \left(-u^k \hat{\xi}_v + v \frac{\partial \hat{\xi}_v}{\partial x} \right) dS \\ \frac{\hat{\xi}_p - \xi_p^k}{\tau_p} = -\frac{\partial \hat{\xi}_u}{\partial x} \end{array} \right. \quad (1.22)$$

Из третьего уравнения системы (1.22) можно выразить $\hat{\xi}_p$ и подставить в первое. Тогда система примет вид:

$$\left\{ \begin{array}{l} \frac{\hat{\xi}_u - \xi_u^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_u}{\Delta t} + \frac{1}{V} \int_{\partial V} n_1 \left(-\frac{\xi_p^k}{\rho} - u^k \hat{\xi}_u + \left(v + \frac{\tau_p}{\rho} \right) \frac{\partial \hat{\xi}_u}{\partial x} \right) dS \\ \frac{\hat{\xi}_v - \xi_v^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_v}{\Delta t} + \frac{1}{V} \int_{\partial V} n_1 \left(-u^k \hat{\xi}_v + v \frac{\partial \hat{\xi}_v}{\partial x} \right) dS \\ \hat{\xi}_p = -\tau_p \frac{\partial \hat{\xi}_u}{\partial x} + \xi_p^k \end{array} \right. \quad (1.23)$$

Для i -го узла j -ой строки данную систему можно аппроксимировать, используя описанный в п. 1.3.2 метод раскрытия контурных интегралов. Использованная в выражениях ниже индексация ячеек приведена на рисунке 1.7.

Первое выражение системы (1.23) после аппроксимации примет вид (1.24). Второе выражение этой системы примет вид (1.25). В выражениях индекс j зафиксирован.

$$\begin{aligned}
& \frac{\hat{\xi}_{u,i} - \xi_{u,i}^k}{\tau} = \\
& = -\frac{1}{2} \frac{\hat{\xi}_{u,i}}{\Delta t} + \\
& + \frac{\Delta y}{V} \left(-\frac{\bar{u}_{i,i+1} + |\bar{u}_{i,i+1}|}{2} \hat{\xi}_{u,i} - \frac{\bar{u}_{i,i+1} - |\bar{u}_{i,i+1}|}{2} \hat{\xi}_{u,i+1} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{\hat{\xi}_{u,i+1} - \hat{\xi}_{u,i}}{\Delta x} \right) - \\
& - \frac{\Delta y}{V} \left(-\frac{\bar{u}_{i-1,i} + |\bar{u}_{i-1,i}|}{2} \hat{\xi}_{u,i-1} - \frac{\bar{u}_{i-1,i} - |\bar{u}_{i-1,i}|}{2} \hat{\xi}_{u,i} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{\hat{\xi}_{u,i} - \hat{\xi}_{u,i-1}}{\Delta x} \right) + \\
& + \frac{\Delta y}{2V} \left(-\frac{\xi_{p,m_1}^k}{\rho} - \frac{\xi_{p,m_4}^k}{\rho} + \frac{\xi_{p,m_2}^k}{\rho} + \frac{\xi_{p,m_3}^k}{\rho} \right)
\end{aligned} \tag{1.24}$$

$$\begin{aligned}
& \frac{\hat{\xi}_{v,i} - \xi_{v,i}^k}{\tau} = \\
& = -\frac{1}{2} \frac{\hat{\xi}_{v,i}}{\Delta t} + \frac{\Delta y}{V} \left[-\frac{\bar{u}_{i,i+1} + |\bar{u}_{i,i+1}|}{2} \hat{\xi}_{v,i}^k - \frac{\bar{u}_{i,i+1} - |\bar{u}_{i,i+1}|}{2} \hat{\xi}_{v,i+1}^k + \nu \frac{\hat{\xi}_{v,i+1} - \hat{\xi}_{v,i}}{\Delta x} \right] - \\
& - \frac{\Delta y}{V} \left[-\frac{\bar{u}_{i-1,i} + |\bar{u}_{i-1,i}|}{2} \hat{\xi}_{v,i-1}^k - \frac{\bar{u}_{i-1,i} - |\bar{u}_{i-1,i}|}{2} \hat{\xi}_{v,i}^k + \nu \frac{\hat{\xi}_{v,i} - \hat{\xi}_{v,i-1}}{\Delta x} \right]
\end{aligned} \tag{1.25}$$

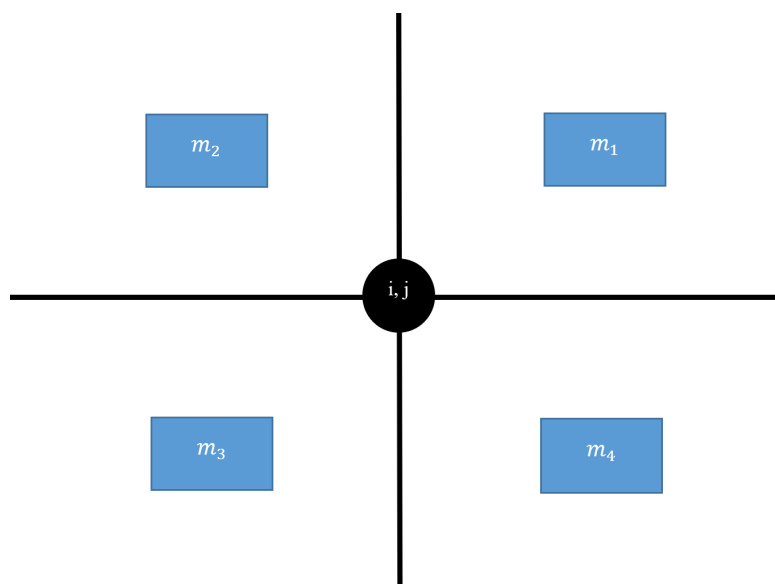


Рисунок 1.7 – Используемая индексация ячеек

Приводя подобные слагаемые в выражениях (1.24) и (1.25), эти выражения примут следующий вид:

$$\begin{aligned}
& \frac{\Delta y}{V} \left(\frac{\bar{u}_{i-1,i} + |\bar{u}_{i-1,i}|}{2} + \frac{1}{\Delta x} \left(v + \frac{\tau_p}{\rho} \right) \right) \hat{\xi}_{u,i-1} - \\
& - \left(\frac{1}{\tau} + \frac{1}{2\Delta t} + \frac{\Delta y}{V} \left[\frac{\bar{u}_{i,i+1} + |\bar{u}_{i,i+1}|}{2} - \frac{\bar{u}_{i-1,i} + |\bar{u}_{i-1,i}|}{2} + \frac{2}{\Delta x} \left(v + \frac{\tau_p}{\rho} \right) \right] \right) \hat{\xi}_{u,i} + \\
& + \frac{\Delta y}{V} \left(-\frac{\bar{u}_{i,i+1} - |\bar{u}_{i,i+1}|}{2} + \frac{1}{\Delta x} \left(v + \frac{\tau_p}{\rho} \right) \right) \hat{\xi}_{u,i+1} = \\
& = - \left(\frac{\xi_{u,i}^k}{\tau} + \frac{\Delta y}{2V} \left(-\frac{\xi_{p,m_1}^k}{\rho} - \frac{\xi_{p,m_4}^k}{\rho} + \frac{\xi_{p,m_2}^k}{\rho} + \frac{\xi_{p,m_3}^k}{\rho} \right) \right)
\end{aligned} \tag{1.26}$$

$$\begin{aligned}
& \frac{\Delta y}{V} \left(\frac{\bar{u}_{i-1,i} + |\bar{u}_{i-1,i}|}{2} + \frac{v}{\Delta x} \right) \hat{\xi}_{v,i-1} - \\
& - \left(\frac{1}{\tau} + \frac{1}{2\Delta t} + \frac{\Delta y}{V} \left[\frac{\bar{u}_{i,i+1} + |\bar{u}_{i,i+1}|}{2} - \frac{\bar{u}_{i-1,i} - |\bar{u}_{i-1,i}|}{2} + \frac{v}{\Delta x} \right] \right) \hat{\xi}_{v,i} + \\
& + \frac{\Delta y}{V} \left(-\frac{\bar{u}_{i,i+1} - |\bar{u}_{i,i+1}|}{2} + \frac{v}{\Delta x} \right) \hat{\xi}_{v,i+1} = -\frac{\xi_{v,i}^k}{\tau}
\end{aligned} \tag{1.27}$$

Выражения (1.26) и (1.27) для каждого из $j = 1..(n_y + 1)$ представляют собой системы линейных разностных уравнений вида:

$$a_i x_{i-1} - c_i x_i + b_i x_{i+1} = -f_i$$

Таким образом, на неявном шаге по X требуется решить $n_y + 1$ СЛАУ для нахождения невязок $\hat{\xi}_u$ и $\hat{\xi}_v$ в каждом узле.

1.3.3.1 Метод прогонки решения СЛАУ

Часто при решении дифференциальных уравнений в частных производных возникает необходимость решения трехдиагональной СЛАУ. В приведенном алгоритме в рамках одной итерации на неявных шагах по X и по Y необходимо решить $n_x + n_y + 2$ таких СЛАУ в соответствии с алгоритмом метода расщепления.

Трехдиагональную СЛАУ можно записать в виде:

$$a_i x_{i-1} - c_i x_i + b_i x_{i+1} = -f_i$$

где $i = 1..n$;

a_i, c_i, b_i, f_i – известные числа, в дальнейшем называемые прогоночными коэффициентами;

Наиболее эффективным алгоритмом решения таких СЛАУ является метод прогонки (*Thomas algorithm*), сложность которого составляет $O(n)$. Алгоритм изложен в [3].

1.3.4 Неявный шаг по X, часть 2

Для решения СЛАУ методом прогонки при фиксированном j необходимо в каждом узле сформировать набор прогоночных коэффициентов. Для этого производится проход циклом по всем ячейкам. Дальнейшие расчеты приведены на примере ячейки номер m с рисунка 1.3.

Аппроксимируем интегралы и производные в первом уравнении (1.23) системы для узлов 1 и 2:

а) Узел 1:

$$\frac{\hat{\xi}_{u1} - \xi_{u1}^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_{u1}}{\Delta t} + \frac{1}{V} \frac{\Delta y}{2} \left(-\frac{\xi_{pm}^k}{\rho} - \frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} \hat{\xi}_{u1} - \frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} \hat{\xi}_{u2} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{\hat{\xi}_{u2} - \hat{\xi}_{u1}}{\Delta x} \right)$$

Собирая коэффициенты при подобных слагаемых ($\hat{\xi}_{u1}, \hat{\xi}_{u2}$), получены следующие прогоночные коэффициенты для узла 1:

$$\begin{aligned} a_{1+} &= 0 \\ c_{1+} &= \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta y}{2} \left[\frac{|\bar{u}_{12}| + \bar{u}_{12}}{2} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta x} \right] \\ b_{1+} &= \frac{1}{V} \frac{\Delta y}{2} \left[-\frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta x} \right] \\ f_{1+} &= \frac{\xi_{u1}^k}{\tau} - \frac{1}{V} \frac{\Delta y}{2} \frac{\xi_{pm}^k}{\rho} \end{aligned}$$

б) Узел 2:

$$\frac{\hat{\xi}_{u2} - \xi_{u2}^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_{u2}}{\Delta t} + \frac{1}{V} \frac{\Delta y}{2} \left(+\frac{\xi_{pm}^k}{\rho} + \frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} \hat{\xi}_{u1} + \frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} \hat{\xi}_{u2} - \left(\nu + \frac{\tau_p}{\rho} \right) \frac{\hat{\xi}_{u2} - \hat{\xi}_{u1}}{\Delta x} \right)$$

Собирая коэффициенты при подобных слагаемых ($\hat{\xi}_{u1}, \hat{\xi}_{u2}$), получены следующие прогоночные коэффициенты для узла 2:

$$a_{2+} = \frac{1}{V} \frac{\Delta y}{2} \left[\frac{|\bar{u}_{12}| + \bar{u}_{12}}{2} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta x} \right]$$

$$c_{2+} = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta y}{2} \left[-\frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} + \left(\nu + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta x} \right]$$

$$b_{2+} = 0$$

$$f_{2+} = \frac{\xi_{u2}^k}{\tau} + \frac{1}{V} \frac{\Delta y}{2} \frac{\xi_{pm}^k}{\rho}$$

Аналогичная процедура производится для 3-го и 4-го узлов.

Раскроем интегралы и производные во втором уравнении системы (1.23)

для узлов 1 и 2:

а) Узел 1:

$$\frac{\hat{\xi}_{v1} - \xi_{v1}^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_{v1}}{\Delta t} + \frac{1}{V} \frac{\Delta y}{2} \left[-\frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} \xi_{v1}^k - \frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} \xi_{v2}^k + \nu \frac{\hat{\xi}_{v2} - \hat{\xi}_{v1}}{\Delta x} \right]$$

Прогоночные коэффициенты:

$$a_{1+} = 0$$

$$c_{1+} = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta y}{2} \left[\frac{\bar{u}_{12} + |\bar{u}_{12}|}{2} + \nu \frac{1}{\Delta x} \right]$$

$$b_{1+} = \frac{1}{V} \frac{\Delta y}{2} \left[-\frac{\bar{u}_{12} - |\bar{u}_{12}|}{2} + \nu \frac{1}{\Delta x} \right]$$

$$f_{1+} = \frac{\xi_{v1}^k}{\tau}$$

б) Узел 2:

$$в) \frac{\hat{\xi}_{v2} - \xi_{v2}^k}{\tau} = -\frac{1}{2} \frac{\hat{\xi}_{v2}}{\Delta t} + \frac{1}{V} \frac{\Delta y}{2} \left[+\frac{|\bar{u}_{12}| + \bar{u}_{12}}{2} \xi_{v1}^k + \frac{|\bar{u}_{12}| - \bar{u}_{12}}{2} \xi_{v2}^k - \nu \frac{\hat{\xi}_{v2} - \hat{\xi}_{v1}}{\Delta x} \right]$$

Прогоночные коэффициенты:

$$a_{2+} = \frac{1}{V} \frac{\Delta y}{2} \left[+\frac{|\bar{u}_{12}| + \bar{u}_{12}}{2} + \nu \frac{1}{\Delta x} \right]$$

$$c_{2+} = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta y}{2} \left[-\frac{|\bar{u}_{12}| - \bar{u}_{12}}{2} + \nu \frac{1}{\Delta x} \right]$$

$$b_{2+} = 0$$

$$f_{2+} = \frac{\xi_{v2}^k}{\tau}$$

Аналогичная процедура производится для 3-го и 4-го узлов.

После того, как прогоночные коэффициенты сформированы, решается $n_y + 1$ СЛАУ методом прогонки, в результате чего в узлах формируются невязки, используемые на неявном шаге по Y .

Наконец, согласно третьему выражению системы (1.23) рассчитываются невязки $\hat{\xi}_p$ для каждой ячейки:

$$\hat{\xi}_p = -\tau_p \frac{\bar{\xi}_{u,23} - \bar{\xi}_{u,14}}{\Delta x} + \xi_p^k$$

1.3.5 Неявный шаг по Y

Выделим из первого и второго уравнений системы (1.7) только те слагаемые, при которых содержится горизонтальная (сонаправленная с осью Y) компонента нормали – n_2 . Из третьего уравнения системы выделим частную производную по Y. Эти слагаемые составляют Λ_2 – часть оператора по Y.

$$\left\{ \begin{array}{l} \frac{\xi_u^{k+1} - \hat{\xi}_u}{\tau} = -\frac{1}{2} \frac{\xi_u^{k+1}}{\Delta t} + \frac{1}{V} \int_{\partial V} n_2 \left(-v^k \xi_u^{k+1} + v \frac{\partial \xi_u^{k+1}}{\partial y} \right) dS \\ \frac{\xi_v^{k+1} - \hat{\xi}_v}{\tau} = -\frac{1}{2} \frac{\xi_v^{k+1}}{\Delta t} + \frac{1}{V} \int_{\partial V} n_2 \left(-\frac{\xi_p^{k+1}}{\rho} - v^k \xi_v^{k+1} + v \frac{\partial \xi_v^{k+1}}{\partial y} \right) dS \\ \frac{\xi_p^{k+1} - \hat{\xi}_p}{\tau_p} = -\frac{\partial \xi_v^{k+1}}{\partial y} \end{array} \right. \quad (1.28)$$

Из третьего уравнения системы (1.28) можно выразить ξ_p^{k+1} и подставить в первое. Тогда система примет вид:

$$\left\{ \begin{array}{l} \frac{\xi_u^{k+1} - \hat{\xi}_u}{\tau} = -\frac{1}{2} \frac{\xi_u^{k+1}}{\Delta t} + \frac{1}{V} \int_{\partial V} n_2 \left(-v^k \xi_u^{k+1} + v \frac{\partial \xi_u^{k+1}}{\partial y} \right) dS \\ \frac{\xi_v^{k+1} - \hat{\xi}_v}{\tau} = -\frac{1}{2} \frac{\xi_v^{k+1}}{\Delta t} + \frac{1}{V} \int_{\partial V} n_2 \left(-\frac{\hat{\xi}_p}{\rho} - v^k \xi_v^{k+1} + \left(v + \frac{\tau_p}{\rho} \right) \frac{\partial \xi_v^{k+1}}{\partial y} \right) dS \\ \xi_p^{k+1} = -\tau_p \frac{\partial \xi_v^{k+1}}{\partial y} + \hat{\xi}_p \end{array} \right. \quad (1.29)$$

Для j -го узла i -го столбца данную систему можно аппроксимировать, используя описанный в п. 1.3.2 метод раскрытия контурных интегралов. Используемая в выражениях ниже индексация ячеек приведена на рисунке 1.7.

Первое выражение системы (1.29) после аппроксимации примет вид (1.30). Второе выражение этой системы примет вид (1.31). В выражениях индекс i зафиксирован.

$$\begin{aligned}
& \frac{\xi_{u,j}^{k+1} - \hat{\xi}_{u,j}}{\tau} = \\
& = -\frac{1}{2} \frac{\xi_{u,j}^{k+1}}{\Delta t} + \\
& + \frac{\Delta x}{V} \left[-\frac{\bar{v}_{j,j+1} + |\bar{v}_{j,j+1}|}{2} \xi_{u,j}^{k+1} - \frac{\bar{v}_{j,j+1} - |\bar{v}_{j,j+1}|}{2} \xi_{u,j+1}^{k+1} + v \frac{\xi_{u,j+1}^{k+1} - \xi_{u,j}^{k+1}}{\Delta y} \right] - \\
& - \frac{\Delta x}{V} \left[-\frac{\bar{v}_{j-1,j} + |\bar{v}_{j-1,j}|}{2} \xi_{u,j-1}^{k+1} - \frac{\bar{v}_{j-1,j} - |\bar{v}_{j-1,j}|}{2} \xi_{u,j}^{k+1} + v \frac{\xi_{u,j}^{k+1} - \xi_{u,j-1}^{k+1}}{\Delta y} \right]
\end{aligned} \tag{1.30}$$

$$\begin{aligned}
& \frac{\xi_{v,j}^{k+1} - \hat{\xi}_{v,j}}{\tau} = \\
& = -\frac{1}{2} \frac{\xi_{v,j}^{k+1}}{\Delta t} + \\
& + \frac{\Delta x}{V} \left(-\frac{\hat{\xi}_{pm}}{\rho} - \frac{\bar{v}_{j,j+1} + |\bar{v}_{j,j+1}|}{2} \xi_{v,j}^{k+1} - \frac{\bar{v}_{j,j+1} - |\bar{v}_{j,j+1}|}{2} \xi_{v,j+1}^{k+1} + \right. \\
& \left. + \left(v + \frac{\tau_p}{\rho} \right) \frac{\xi_{v,j+1}^{k+1} - \xi_{v,j}^{k+1}}{\Delta y} \right) - \\
& - \frac{\Delta x}{V} \left(-\frac{\hat{\xi}_{pm}}{\rho} - \frac{\bar{v}_{j-1,j} + |\bar{v}_{j-1,j}|}{2} \xi_{v,j-1}^{k+1} - \frac{\bar{v}_{j-1,j} - |\bar{v}_{j-1,j}|}{2} \xi_{v,j}^{k+1} + \right. \\
& \left. + \left(v + \frac{\tau_p}{\rho} \right) \frac{\xi_{v,j}^{k+1} - \xi_{v,j-1}^{k+1}}{\Delta y} \right) + \frac{\Delta x}{2V} \left(-\frac{\xi_{p,m_1}^k}{\rho} - \frac{\xi_{p,m_2}^k}{\rho} + \frac{\xi_{p,m_3}^k}{\rho} + \frac{\xi_{p,m_4}^k}{\rho} \right)
\end{aligned} \tag{1.31}$$

Приводя подобные слагаемые в выражениях (1.30) и (1.31), эти выражения примут следующий вид:

$$\begin{aligned}
& \frac{\Delta x}{V} \left(\frac{\bar{v}_{j-1,j} + |\bar{v}_{j-1,j}|}{2} + \frac{v}{\Delta y} \right) \xi_{u,j-1}^{k+1} - \\
& - \left(\frac{1}{2\Delta t} + \frac{1}{\tau} + \frac{\Delta x}{V} \left(\frac{\bar{v}_{j,j+1} + |\bar{v}_{j,j+1}|}{2} - \frac{\bar{v}_{j-1,j} - |\bar{v}_{j-1,j}|}{2} \right) \right) \xi_{u,j}^{k+1} + \\
& + \frac{\Delta x}{V} \left(-\frac{\bar{v}_{j,j+1} - |\bar{v}_{j,j+1}|}{2} + \frac{v}{\Delta y} \right) \xi_{u,j+1}^{k+1} = -\frac{\hat{\xi}_{u,j}}{\tau}
\end{aligned} \tag{1.32}$$

$$\begin{aligned}
& \frac{\Delta x}{V} \left(\frac{\bar{v}_{j-1,j} + |\bar{v}_{j-1,j}|}{2} + \left(v + \frac{\tau_p}{\rho} \right) \right) \xi_{v,j-1}^{k+1} - \\
& - \left(\frac{1}{2\Delta t} + \frac{1}{\tau} + \frac{\Delta x}{V} \left(\frac{\bar{v}_{j,j+1} + |\bar{v}_{j,j+1}|}{2} - \frac{\bar{v}_{j-1,j} - |\bar{v}_{j-1,j}|}{2} \right) \right) \xi_{v,j}^{k+1} + \\
& + \frac{\Delta x}{V} \left(-\frac{\bar{v}_{j,j+1} - |\bar{v}_{j,j+1}|}{2} + \left(v + \frac{\tau_p}{\rho} \right) \right) \xi_{v,j+1}^{k+1} = \\
& = - \left(\frac{\hat{\xi}_{v,j}}{\tau} + \frac{\Delta x}{2V} \left(-\frac{\xi_{p,m_1}^k}{\rho} - \frac{\xi_{p,m_2}^k}{\rho} + \frac{\xi_{p,m_3}^k}{\rho} + \frac{\xi_{p,m_4}^k}{\rho} \right) \right)
\end{aligned} \tag{1.33}$$

Выражения (1.32) и (1.33) для каждого из $i = 1..(n_x + 1)$ представляют собой системы линейных разностных уравнений вида:

$$a_j x_{j-1} - c_j x_j + b_j x_{j+1} = -f_j$$

Таким образом, на неявном шаге по Y требуется решить $n_x + 1$ СЛАУ для нахождения невязок ξ_u^{k+1} и ξ_v^{k+1} в каждом узле.

Для решения СЛАУ методом прогонки при фиксированном i необходимо в каждом узле сформировать набор прогоночных коэффициентов. Для этого производится проход циклом по всем ячейкам. Дальнейшие расчеты приведены на примере ячейки номер t с рисунка 1.3.

Раскроем интегралы и производные в первом уравнении системы (1.29) для узлов 1 и 4:

а) Узел 1:

$$\frac{\xi_{u1}^{k+1} - \hat{\xi}_{u1}}{\tau} = -\frac{1}{2} \frac{\xi_{u1}^{k+1}}{\Delta t} + \frac{1}{V} \frac{\Delta x}{2} \left[-\frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} \xi_{u1}^{k+1} - \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} \xi_{u4}^{k+1} + v \frac{\xi_{u4}^{k+1} - \xi_{u1}^{k+1}}{\Delta y} \right]$$

Прогоночные коэффициенты:

$$\begin{aligned}
a_{1+} & = 0 \\
c_{1+} & = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta x}{2} \left[\frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} + v \frac{1}{\Delta y} \right] \\
b_{1+} & = \frac{1}{V} \frac{\Delta x}{2} \left[-\frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} + v \frac{1}{\Delta y} \right]
\end{aligned}$$

$$f_{1+} = \frac{\hat{\xi}_{u1}}{\tau}$$

б) Узел 2:

$$\frac{\xi_{u4}^{k+1} - \hat{\xi}_{u4}}{\tau} = -\frac{1}{2} \frac{\xi_{u4}^{k+1}}{\Delta t} + \frac{1}{V} \frac{\Delta x}{2} \left[\frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} \xi_{u1}^{k+1} + \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} \xi_{u4}^{k+1} - v \frac{\xi_{u4}^{k+1} - \xi_{u1}^{k+1}}{\Delta y} \right]$$

Прогоночные коэффициенты:

$$a_{4+} = \frac{1}{V} \frac{\Delta x}{2} \left[+ \frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} + v \frac{1}{\Delta y} \right]$$

$$c_{4+} = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta x}{2} \left[- \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} + v \frac{1}{\Delta y} \right]$$

$$b_{4+} = 0$$

$$f_{4+} = \frac{\hat{\xi}_{u4}}{\tau}$$

Аналогичная процедура производится для 3 и 4 узлов.

Раскроем интегралы и производные во втором уравнении системы (1.29)

для узлов 1 и 4:

а) Узел 1:

$$\frac{\xi_{v1}^{k+1} - \hat{\xi}_{v1}}{\tau} =$$

$$= -\frac{1}{2} \frac{\xi_{v1}^{k+1}}{\Delta t} +$$

$$+ \frac{1}{V} \frac{\Delta x}{2} \left(-\frac{\hat{\xi}_{pm}}{\rho} - \frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} \xi_{v1}^{k+1} - \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} \xi_{v4}^{k+1} + \left(v + \frac{\tau_p}{\rho} \right) \frac{\xi_{v4}^{k+1} - \xi_{v1}^{k+1}}{\Delta y} \right)$$

Прогоночные коэффициенты:

$$a_{1+} = 0$$

$$c_{1+} = \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta x}{2} \left[\frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} + \left(v + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta y} \right]$$

$$b_{1+} = \frac{1}{V} \frac{\Delta x}{2} \left[- \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} + \left(v + \frac{\tau_p}{\rho} \right) \frac{1}{\Delta y} \right]$$

$$f_{1+} = \frac{\hat{\xi}_{v1}}{\tau} - \frac{1}{V} \frac{\Delta x}{2} \frac{\hat{\xi}_{pm}}{\rho}$$

б) Узел 2:

$$\begin{aligned} \frac{\xi_{v4}^{k+1} - \hat{\xi}_{v4}}{\tau} &= \\ &= -\frac{1}{2} \frac{\xi_{v4}^{k+1}}{\Delta t} + \\ &+ \frac{1}{V} \frac{\Delta x}{2} \left(+ \frac{\hat{\xi}_{pm}}{\rho} + \frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} \xi_{v1}^{k+1} + \frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} \xi_{v4}^{k+1} - \left(v + \frac{\tau}{\rho} \right) \frac{\xi_{v4}^{k+1} - \xi_{v1}^{k+1}}{\Delta y} \right) \end{aligned}$$

Прогоночные коэффициенты:

$$\begin{aligned} a_{4+} &= \frac{1}{V} \frac{\Delta x}{2} \left[\frac{\bar{v}_{14} + |\bar{v}_{14}|}{2} + \left(v + \frac{\tau}{\rho} \right) \frac{1}{\Delta y} \right] \\ c_{4+} &= \frac{1}{2} \frac{1}{\Delta t} + \frac{1}{\tau} + \frac{1}{V} \frac{\Delta x}{2} \left[-\frac{\bar{v}_{14} - |\bar{v}_{14}|}{2} + \left(v + \frac{\tau}{\rho} \right) \frac{1}{\Delta y} \right] \\ b_{4+} &= 0 \\ f_{4+} &= \frac{\hat{\xi}_{v4}}{\tau} + \frac{1}{V} \frac{\Delta x}{2} \frac{\hat{\xi}_{pm}}{\rho} \end{aligned}$$

Аналогичная процедура производится для 3 и 4 узлов.

После того, как прогоночные коэффициенты сформированы, решается $n_x + 1$ СЛАУ методом прогонки, в результате чего в узлах формируются невязки, используемые на инкрементальном шаге.

Наконец, согласно третьему выражению системы (1.29) рассчитываются невязки $\hat{\xi}_p$ для каждой ячейки:

$$\xi_p^{k+1} = -\tau_p \frac{\bar{\xi}_{v,34} - \bar{\xi}_{v,12}}{\Delta y} + \hat{\xi}_p$$

1.3.6 Инкрементальный шаг

В конце итерации вычисленные невязки прибавляются к текущим значениями скоростей и давления:

$$\begin{cases} u^{k+1} = u^k + \xi_u^{k+1} \\ v^{k+1} = v^k + \xi_v^{k+1} \\ p^{k+1} = p^k + \xi_p^{k+1} \end{cases}$$

1.4 Граничные условия

В разработанной на основе вышеописанного алгоритма программе реализованы несколько видов граничных условий, а именно:

- Задание вектора скорости на границе.
- Стена с прилипанием.

- Условие свободного вытекания.
- Задание давления на границе.

Далее каждое из ГУ рассмотрено по отдельности и изложены особенности реализации.

Блок граничных условий в разработанной программе выглядит как показано на рисунке 1.8.

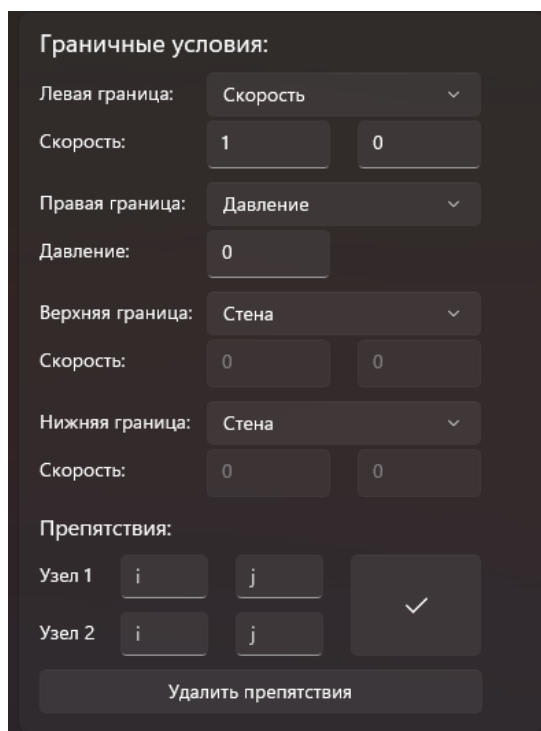


Рисунок 1.8 – Блок ГУ

1.4.1 Граничные условия I рода

К граничным условиям I относится жесткое задание на границе расчетной области значений искомых величин – вектора скорости и давления. В данном разделе рассмотрено ГУ I относительно скоростей. Задание давления рассматривается в п. , так как реализовано в связке с ГУ II рода относительно скоростей.

Общий вид ГУ первого рода для скорости имеет вид:

$$\underline{v}|_{\partial V} = (\tilde{u}, \tilde{v})$$

В разработанной программе представлены 2 ГУ первого рода относительно скорости вида «Скорость» и «Стена».

ГУ «Скорость» (рисунок 1.9) позволяет установить на границе расчетной области постоянное значение вектора скорости покомпонентно, в результате чего в области начнется движение флюида.

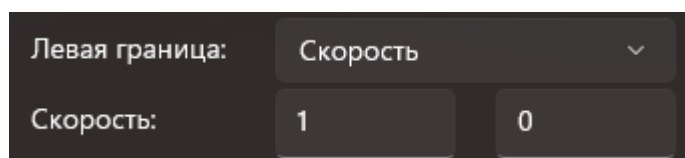


Рисунок 1.9 – ГУ скорости

ГУ реализовано следующим образом:

а) В каждый момент времени граничным узлам присваивается заданный в начале расчета известный вектор скоростей (u, v) .

б) В ходе расчета для граничных узлов невязки первого и второго уравнений системы (1.7) на каждом шаге по времени тождественно равны нулю, таким образом, скорость в граничных узлах не меняется.

ГУ «Стена» (рисунок 1.10) реализовано аналогично ГУ «Скорость» и физически означает прилипание жидкости к поверхности (вектор скорости тождественно равен 0).

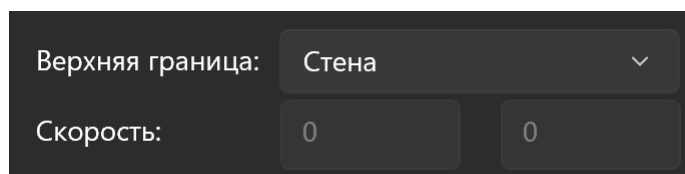


Рисунок 1.10 – ГУ стены

1.4.2 Граничные условия II рода

Граничные условия II для скорости реализуют свободное вытекание жидкости из рассматриваемого объема. Общий вид ГУ второго рода для скорости имеет вид:

$$\frac{\partial v_n}{\partial n} |_{\partial V} = \nabla(\underline{v} \cdot \underline{n}) \cdot \underline{n} = 0$$

ГУ ставится в предположении, что к выходу из области течение устанавливается и вектор скорости направлен по нормали к поверхности, поэтому $v_n = |\underline{v}|$. Физический смысл ГУ: на выходе из области течение

установилось и его скорость в каждой точке не меняется вдоль нормали к границе.

Способ численной реализации для ортогональной сетки на примере границы $x = l$ (или $x = 0$):

$$а) \frac{\partial v_n}{\partial n} |_{\partial V} = \nabla(\underline{v} \cdot \underline{n}) \cdot \underline{n} = \frac{\partial u}{\partial x} = 0.$$

б) На неявном шаге по X для каждой из горизонтальных «осей», по которым осуществляется расчет невязок в крайних узлах осуществляется приравнивание прогоночных коэффициентов (для левой и правой границ области):

$$a_{u,n_x} = c_{u,n_x} \text{ ИЛИ } b_{u,0} = c_{u,0}$$

$$a_{v,n_x} = c_{v,n_x} \text{ ИЛИ } b_{v,0} = c_{v,0}$$

Таким образом, после расчета невязок методом прогонки, невязки в крайних узлах каждой «оси» будут равны.

в) После неявного шага по Y для каждой из вертикальных «осей», по которым осуществляется расчет невязок в крайних узлах осуществляется приравнивание прогоночных невязок:

$$\xi_{u,n_x} = \xi_{u,n_x-1} \text{ ИЛИ } \xi_{u,0} = \xi_{u,1}$$

$$\xi_{v,n_x} = \xi_{v,n_x-1} \text{ ИЛИ } \xi_{v,0} = \xi_{v,1}$$

Таким образом по итогам итерации невязки в краевых узлах будут равны и будет выполняться $\frac{\partial u}{\partial x} = \frac{u_{n_x} - u_{n_x-1}}{\Delta x} = 0$ ИЛИ $\frac{u_1 - u_0}{\Delta x} = 0$.

Данное ГУ для верхней и нижней границ расчетной области ($y = 0$ или $y = h$) реализуется аналогичным образом.

ГУ можно использовать как самостоятельное, так и в сочетании с ГУ на давление, которое рассмотрено в п. 1.4.3.2.

1.4.3 Граничные условия по давлению

Несмотря на то, что общепринятыми ГУ для системы уравнений Навье-Стокса является задание скоростей или их производных на границах расчетной области, также возможно решение задач с применением ГУ относительно давления. Тем не менее, задание давления на входе или выходе из области

возможно только в совокупности с условиями на скорость. Далее рассмотрены два метода задания давления.

1.4.3.1 Давление и ГУ I рода по скорости

Такое ГУ может понадобиться, например, при моделировании вытекания жидкости из большой емкости через узкий слив. В таком случае известно лишь давление на входе в слив.

ГУ ставится следующим образом:

$$\begin{aligned} \underline{v}|_{\partial V} &= \underline{0} \\ p|_{\partial V} &= \tilde{p} \end{aligned}$$

Равенство нулю скорости на границе области является вынужденным условием, без которого система не является замкнутой. Тем не менее, как показано далее, такое грубое условие не влияет на развитие течение сразу за границей и служит лишь для сходимости задачи.

ГУ реализуется следующим образом:

а) В начале каждого шага по времени, вектор скорости (u, v) и промежуточные значения (u_k, v_k) на рассматриваемой области приравниваются к 0.

б) Невязки первых двух уравнений системы (1.7) в приграничных узлах также поддерживаются равными нулю в течение итерационного процесса в рамках шага по времени.

в) В приграничных ячейках устанавливается заданное значение давления \tilde{p} . Невязки ξ_p поддерживаются равными нулю.

Результаты расчета течения в узком канале с таким ГУ приведены в приложении.

1.4.3.2 Давление и ГУ II рода по скорости

Рассмотренное в п. 1.4.2 ГУ второго рода для скорости можно использовать в связке с заданием давления, такое ГУ может быть установлено на границе вытекания жидкости из расчетной области, если известно давление на выходе. Тем не менее, данное ГУ можно также использовать и на входе в области, например для замены граничным условием отброшенной части трубопровода.

ГУ ставится следующим образом:

$$\frac{\partial v_n}{\partial \underline{n}} \Big|_{\partial V} = 0$$

$$p \Big|_{\partial V} = \tilde{p}$$

Предполагается, что к выходу из области вектор скорости перпендикулярен границе и не изменится вдоль нормали.

Реализация:

а) Для узлов, принадлежащим краевым ячейкам, применяются алгоритм установки ГУ второго рода согласно п. 1.4.2.

б) В этих ячейках устанавливается необходимое значение давления \tilde{p} , а невязки ξ_p поддерживаются равными 0.

Глава 2. Программная реализация

Изложенный в разделе 1.3 алгоритм реализован программно на основе языка программирования *C++/WinRT* [15] на основе фреймворка *Universal Windows Platform (UWP)* и графических инструментов *Windows User Interface (WinUI)* [13].

В данном разделе изложены особенности реализации программы и взаимодействия ее логической части с графическим интерфейсом.

2.1 Структура

Программа разработана с применением принципов объектно-ориентированного программирования. Так, реализация алгоритма из п. 1.3 основана на трех классах:

– *Node* – класс описывает узел сетки, его поля хранят параметры узла, а также все узловые переменные – скорости, невязки и пригоночные коэффициенты.

– *Cell* – класс описывает ячейку сетки, его поля хранят указатели на узлы, принадлежащие данной ячейке, а также переменные, относящиеся к ячейке – давление и невязки по третьему уравнению системы (1.7).

– *NavierStokes* – класс содержит все необходимые для расчета параметры, такие как геометрические размеры области, свойства флюида, параметры сходимости, граничные условия и массивы узлов и ячеек.

Ниже каждый из классов рассмотрен подробнее и описаны методы, фигурирующие в процессе решения задачи.

2.1.1 Класс *Node*

Поля класса хранят следующие переменные:

- Компоненты вектора скорости.
- Координаты узла.
- Величину объема, относящегося к данному узлу.
- Булеву переменную, определяющую, является ли узел граничным.

– Промежуточные значения компонент вектора скорости в процессе итерации.

– Невязки первого и второго уравнений системы (1.7).

– 8 прогоночных коэффициентов: 4 для неявного шага по X и 4 для неявного шага по Y .

Класс имеет методы:

– *void UpdateVelocities* – метод обновляет значения полей, связанных со скоростями.

– *void EvaluateXiuv* – метод рассчитывает невязки в конце явного шага.

2.1.2 Класс *Cell*

Поля класса содержат следующие переменные:

– Линейные размеры ячейки.

– Давление в ячейке.

– Промежуточное значение давления в процессе итерации.

– Невязку третьего уравнения системы (1.7).

– 4 указателя на узлы, принадлежащие ячейке.

Класс имеет методы:

– *void EvaluateIntegralsForCell* – метод «собирает» контурные интегралы для каждого из узлов на явном шаге итерационного процесса согласно алгоритму, изложенному в пп. 1.3.1 и 1.3.2.

– *void XStepForCell* – в методе происходит расчет прогоночных коэффициентов для каждого из узлов ячейке на неявном шаге по X (пп. 1.3.3 и 1.3.4).

– *void YStepForCell* – в методе происходит расчет прогоночных коэффициентов для каждого из узлов ячейке на неявном шаге по Y (п. 1.3.5).

– *void EvaluateXp* – в методе обновляются невязки по давлению в конце неявных шагов по X и Y .

– *void UpdatePressure* – метод обновляет значения полей, связанных с давлением.

2.1.3 Класс *NavierStokes*

Основной логический класс программы. Содержит следующие поля:

- Массивы указателей на объекты узлов и ячеек расчетной сетки.
- Массивы препятствий – узлов, принадлежащих стенам на пути движения флюида.
- Плотность и вязкость жидкости.
- Линейные размеры расчетной области и количество узлов по каждой из осей.
- Время расчета и шаг по времени.
- Параметры сходимости τ и τ_p .
- Граничные условия для каждой из сторон.

Класс имеет методы:

- *void SetInitialConditions* – метод осуществляет стартовую инициализацию решателя – создает массивы узлов и ячеек с нулевыми значениями скоростей, задает граничные условия и генерирует массив препятствий, если такие имеются.
- *void SetBoundaryConditions* – метод в начале каждого шага по времени устанавливает необходимые ГУ.
- *void ExplicitStep* – метод выполняет явный шаг в рамках итерации: циклом по всем ячейкам рассчитываются контурные интегралы, затем циклом по узлам на основе интегралов и неинтегральных слагаемых вычисляются невязки.
- *void ThomasAlg* – метод реализует алгоритм прогонки решения СЛАУ: на вход он получает ссылки на массивы прогоночных коэффициентов и ссылку на массив решения, в который записываются результаты решения СЛАУ.
- *void XStep* – метод выполняет неявный шаг по X: для каждой горизонтальной «оси» формируются массивы прогоночных коэффициентов и методом прогонки рассчитываются невязки. В конце метода для каждой ячейки обновляет значение невязки по давлению.
- *void YStep* – метод выполняет неявный шаг по Y, выполняя аналогичные операции.

– *Solve* – основной метод решателя. Производит последовательное решение задачи. В основе метода лежит цикл *for* по всем шагам по времени. На каждом шаге запускается итерационный процесс, производящий последовательное выполнение шагов алгоритма. Итерационный процесс длится до тех пор, пока модули невязок не станут меньше некоторого заранее определенного $\varepsilon > 0$. После расчета всех шагов по времени, каждый узел сетки хранит в себе массивы скоростей во все моменты времени, а каждая ячейка – массив значений давления. Эти массивы в дальнейшем используются для отрисовки соответствующих полей.

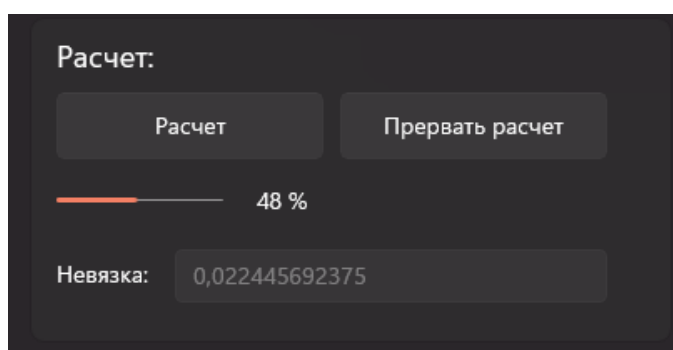


Рисунок 2.1 – Прогресс решения

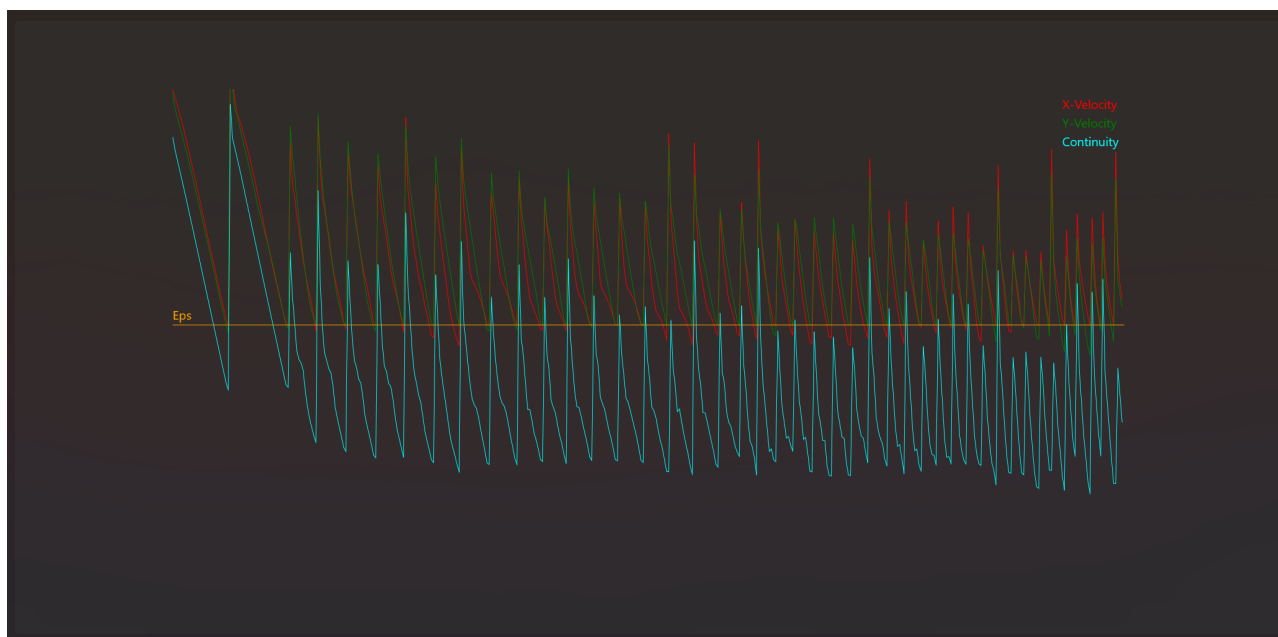


Рисунок 2.2 – Графики невязок в процессе решения

в) В случае неудовлетворительной сходимости решения (вызванной неоптимальным выбором параметров сходимости или некорректным заданием ГУ) пользователь имеет возможность прервать решение без перезапуска программы. Это также возможно благодаря асинхронности – если возвращаемый тип имеет вид *IAsyncAction* (или *IAsyncOpertaionWithProgress*), то такая функция имеет метод *Cancel()*, позволяющий приостановить ее выполнение в любой момент и начать заново с измененными параметрами.

2.2 Особенности

2.2.1 Параллельное выполнение задач

Как следует из изложенного алгоритма, в процессе решения используется некоторое количество циклов по всем узлам и ячейкам. Для сетки с большим количеством узлов последовательное выполнение каждого из этих циклов на современном вычислительном оборудовании является неэффективным.

Актуальные центральные процессоры компьютеров в большинстве своем содержат больше одного вычислительного блока – ядра. Популярные модели процессоров компаний *Intel* и *AMD* имеют от 4 до 8 ядер, причем каждое из них способно производить вычисления в 2 потока параллельно [7].

Разработка решателя системы уравнений Навье-Стокса велась на компьютере, оборудованном процессором с 6 ядрами и 12 потоками, поэтому произведена оптимизация программы под многопоточные вычисления. По умолчанию итерации цикла *for* выполняется последовательно в одном потоке, при этом остальные ядра процессора остаются незадействованными.

Рассмотрим пример оптимизации на неявном шаге по X. На данном шаге для каждой горизонтальной «оси» сетки производится вычисление невязок в узлах методом прогонки. Алгоритм сначала формирует в узлах прогоночные коэффициенты, а затем решает полученную СЛАУ. Таким образом, имеется цикл *for(int j = 0; j < n_y; j++)*, проходящий по всем горизонтальным «осям». Причем для каждой итерации цикла производится большое количество арифметических действий и решение СЛАУ, причем ничто не мешает выполнять

итерации цикла параллельно, так как расчеты, связанные с разными «осями», друг от друга не зависят.

Для этих целей применяется программная библиотека *Concurrency Runtime* от компании *Microsoft*, в которой реализованы параллельные циклы [16]. С ее помощью цикл `for(int j = 0; j < ny; j++)` заменяется циклом `concurrency::parallel_for(0, ny, int j[&])` [2], решающий задачу расчета невязок для каждой из «осей» независимо. Так как расчетная машина обладает возможностью независимого исполнения 12 потоков, то в один момент могут рассчитываться невязки в узлах 12-и «осей» одновременно.

В расчетной программе возможности библиотеки *Concurrency Runtime* применяются для всех циклов, содержащих большое количество арифметических операций в рамках итерации цикла. Таким образом, программа в полной мере задействует возможности современных ПК в параллельных вычислениях.

2.2.2 Асинхронность

При разработке программ с графическим пользовательским интерфейсом необходимо обеспечить независимое функционирование *UI* и логики программы. При реализации по умолчанию, при запуске вычислительно-затратной операции из графического интерфейса (например, по нажатию кнопки), главный поток прекращает обрабатывать *UI* и начинает выполнение вычислений, при этом *UI* перестает отвечать на команды пользователя до тех пор, пока не завершится расчет. И при разработке программы, ориентированной на использование конечным пользователем (такой, как расчетное ПО), такое поведение является нежелательным.

Для исключения такого поведения по многие современные языки программирования встроена поддержка асинхронности, пример использования рассмотрен приведен в программе, разработанной в рамках данной работы [3].

По нажатию на кнопку «Расчет» происходит запуск функции *Solve* класса *NavierStokes*, причем выполнение решения может занять длительное время в силу большого количества необходимых арифметических действий. Для

параллельного выполнения функции решателя и графического интерфейса реализован механизм асинхронности. Он изложен ниже:

г) Пользователь запускает расчет. Происходит вызов функции *Solve*.

д) При помощи операторов *co_await*, внедренных в тело функции, для нее выделяется отдельный поток, в нем запускается выполнение решателя.

е) Продолжается выполнение потока, обрабатывающего *UI*, благодаря чему пользователь может продолжать с ним взаимодействовать, в частности, наблюдать за процессом решения.

ж) Для информирования пользователя о ходе выполнения возвращаемый тип функции *Solve* установлен как *IAsyncOperationWithProgress*, что позволяет в нужные моменты возвращать из функции промежуточную информацию о расчете, не дожидаясь его завершения и не прерывая выполнения самой функции [14]. Так, в конце каждой 2-й итерации, из функции *Solve* в графический интерфейс возвращаются значения невязок каждого из трех решаемых уравнений, а также номер текущего шага по времени. В результате, пользователь имеет исчерпывающую информацию о ходе решения и его сходимости: процент выполнения и логарифмические графики невязок. Также в специальное поле выводится значение максимальной невязки в данный момент. Анимация изменения во времени графиков невязок приведена в приложении.

Глава 3. Результаты

По завершении разработки ПО, описанного выше, необходимо произвести его валидацию на примере уже исследованных задач, решенных как аналитически, так и численно. Ниже приведены несколько расчетных кейсов, показывающих что разработанное ПО адекватно решает поставленные перед ним задачи двумерной гидродинамики.

3.1 Течение в плоском канале

Течение жидкости в канале L между двумя стенками, разнесенными между собой на расстояние $2H$ является классической задачей гидродинамики, имеющей аналитическое решение и называемое течением Пуазейля.

Постановка задачи: несжимаемая вязкая жидкость совершает движение с устоявшимся полем скоростей в канале между плоскими стенками. Движение жидкости происходит по причине существования перепада давления Δp между началом и концом канала. На стенках канала выполняется условие прилипания, т.е. $\underline{v}|_{\partial V} = \underline{0}$. В таком случае, поле скоростей имеет следующий вид [1]:

$$\begin{cases} u = \frac{\Delta p H^2}{2\mu L} \left(1 - \left(\frac{y}{H} \right)^2 \right) \\ v = 0 \end{cases}$$

где u – горизонтальная компонента скорости;

v – вертикальная компонента скорости;

μ – динамическая вязкость = 0,001 Па · с;

H – расстояние от центральной горизонтальной оси канала до стенки = 0,25 м;

L – длина канала = 2 м;

Δp – перепад давления в канале = 500 Па.

Таким образом, в сечении канала профиль горизонтальной компоненты скорости имеет форму параболы, а вертикальная скорость тождественно равна 0.

В силу того, что аналитическое решение опирается на градиент давления, а не на скорость потока на входе расчетную область, в разработанном ПО задача решалась с использованием ГУ по давлению, изложенным в п. 1.4.3.2. На входе в расчетную область принято:

$$\begin{cases} \frac{du}{dx} = 0 \\ p = 500 \text{ Па} \end{cases}$$

На выходе:

$$\begin{cases} \frac{du}{dx} = 0 \\ p = 0 \text{ Па} \end{cases}$$

Таким образом, в расчетной области сразу формируется параболический профиль скорости без разгонного участка.

Поле горизонтальной компоненты скорости в начальный момент времени приведено на рисунке 3.1.

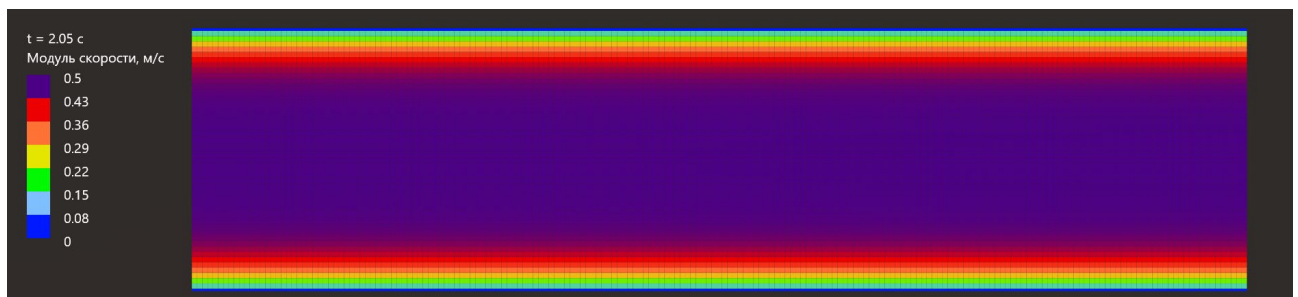


Рисунок 3.1 - Поле горизонтальной компоненты скорости в плоском канале в начальный момент времени

На рисунке 3.2 приведено устоявшееся поле вертикальной компоненты скорости.

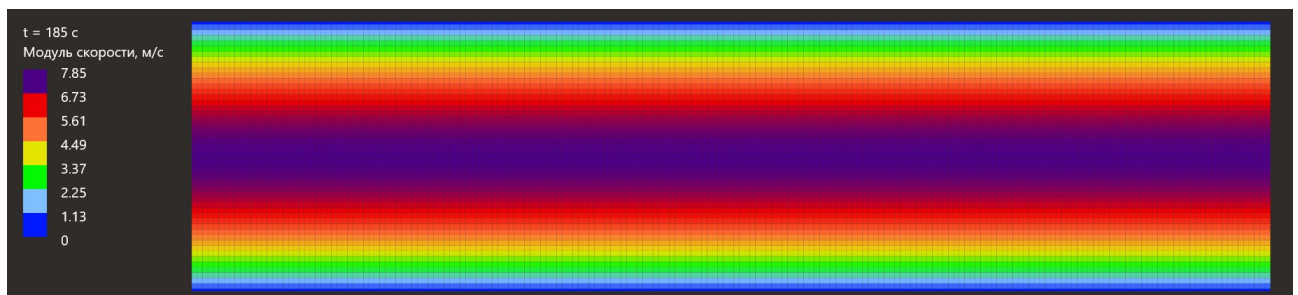


Рисунок 3.2 – Поле горизонтальной компоненты скорости в плоском канале

На графике 3.3 приведено сравнение профиля скорости в сечении канала для аналитического решения и численного решения в разработанном ПО.

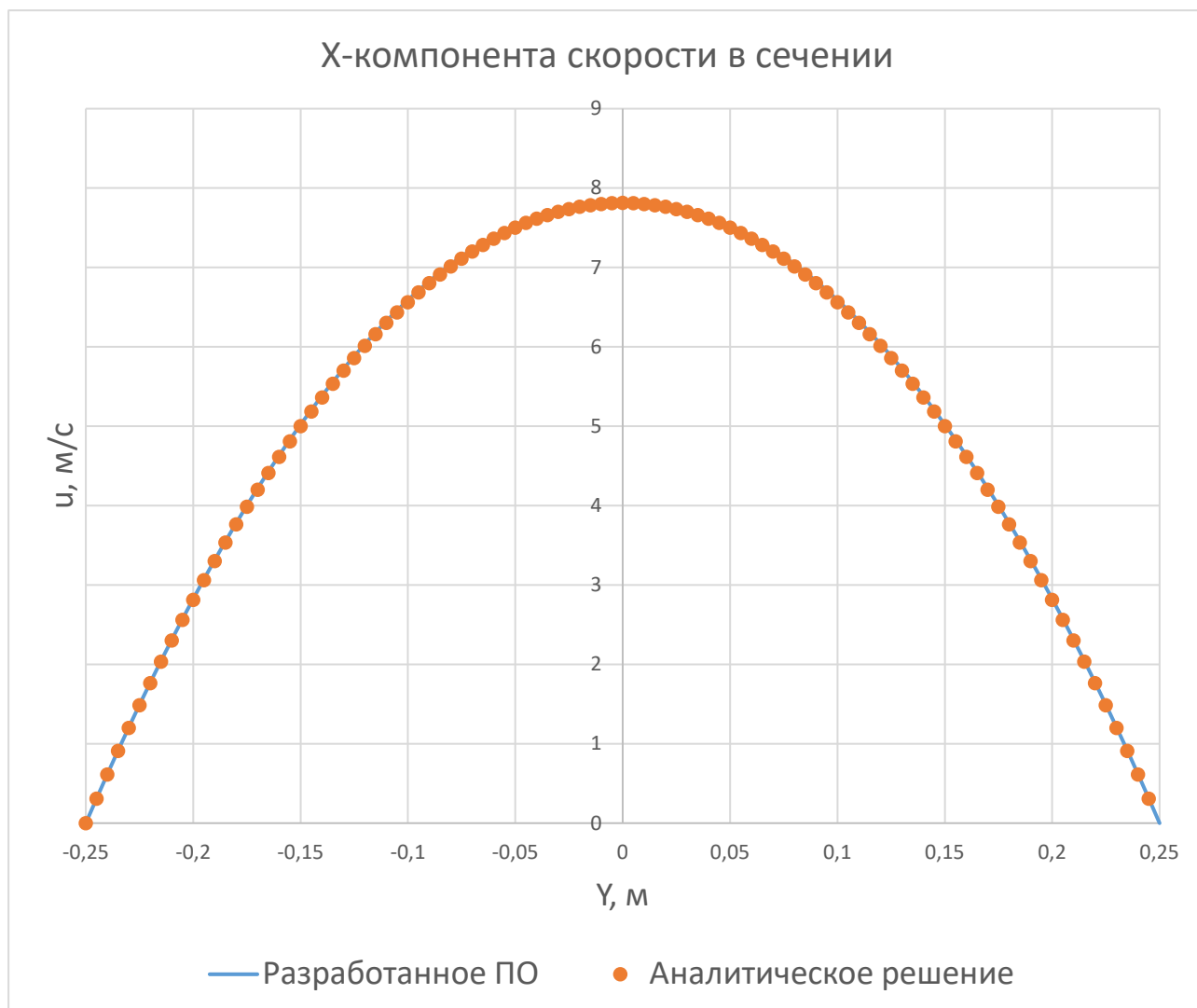


Рисунок 3.3 – Сравнение аналитического и численного расчета течения в плоском канале

Ошибка максимальной скорости составила:

$$\Delta u_{max} = \frac{7,8094 - 7,8431}{7,8094} * 100\% = -0,43\%$$

Таким образом отличие численного решения от аналитического составляет менее 1%.

На рисунке 3.4 приведено поле вертикальных компонент скорости.

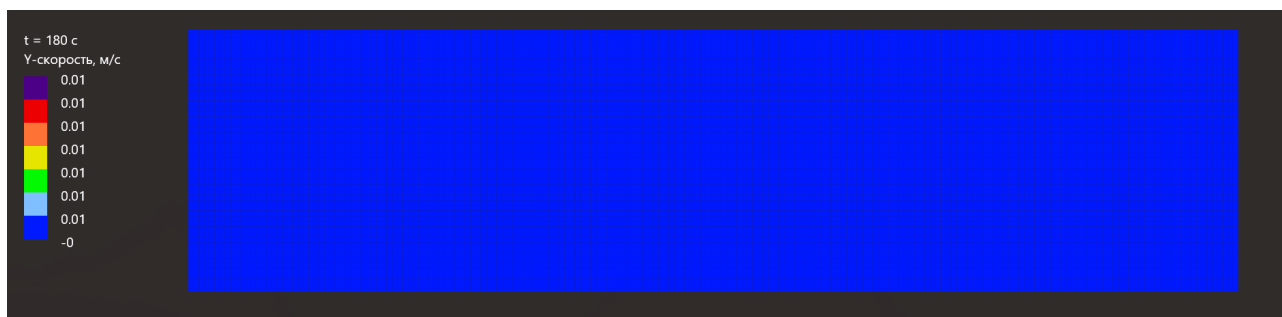


Рисунок 3.4 – Поле вертикальной компоненты скорости в плоском канале

Из приведенного поля видно, что вертикальная компонента скорости во всем канале стремится к 0, что согласуется с аналитическим решением.

На рисунке приведено поле давления.

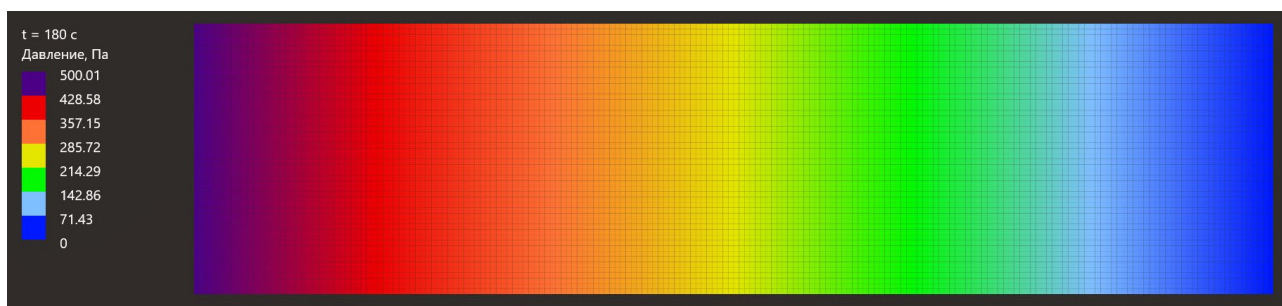


Рисунок 3.5 – Поле давления в плоском канале

Из приведенного поля давления видно выполнение граничных условий и линейность градиента давления в канале.

Дополнительно произведен расчет течения в плоском канале, в котором на входе в канал заданное ГУ по скорости $u = 1$ м/с и произведена валидация решения на расчет в гидрогазодинамическом пакете *Ansys Fluent*.

На рисунке 3.6 приведены рассчитанные поля скоростей в разработанном ПО и *Ansys Fluent*. В обоих случаях использована одинаковая сетка из прямоугольников размером 201X51 узел.

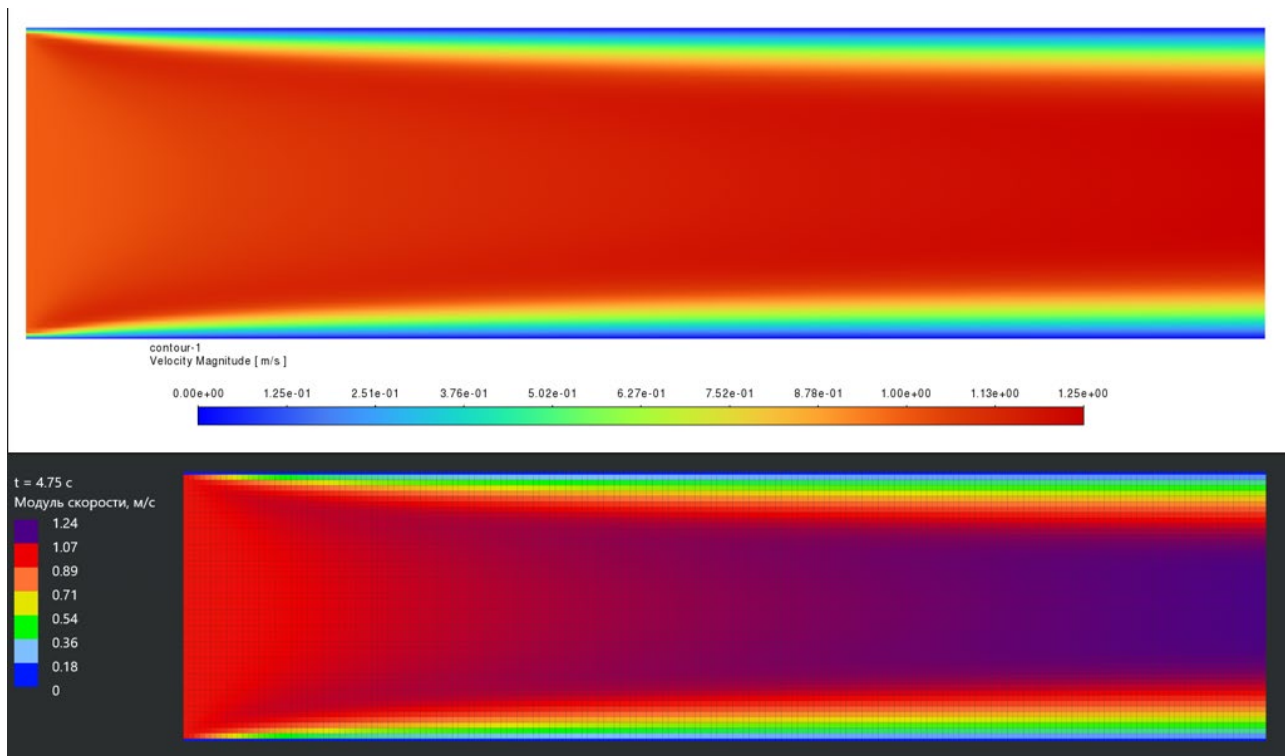


Рисунок 3.6 – Горизонтальная компонента скорости в сравнении разработанного ПО и Ansys Fluent.

На графике 3.7 приведено сравнение профилей скорости на выходе из расчетной области.

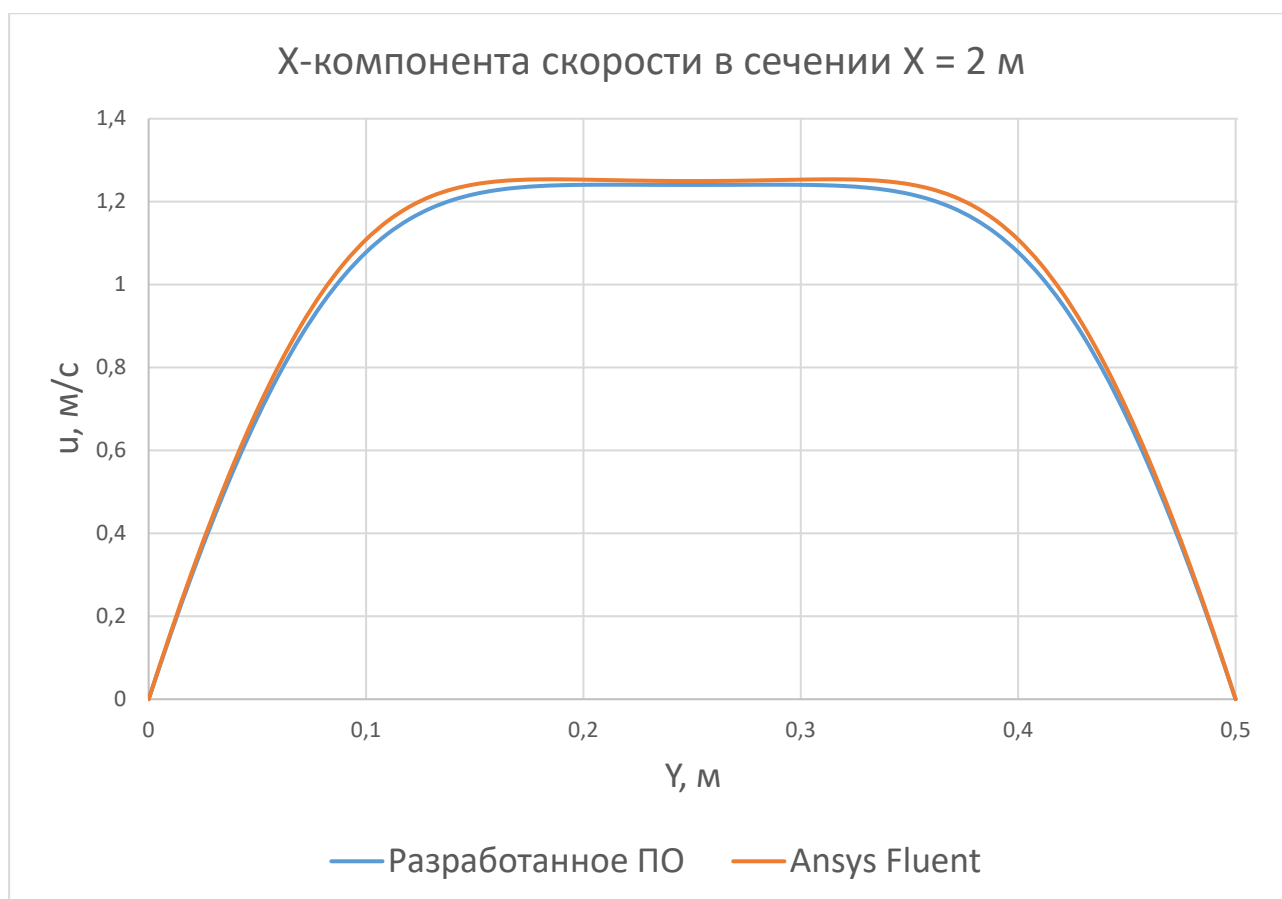


Рисунок 3.7 – Сравнение профиля горизонтальной компоненты скорости на выходе из расчетной области между разработанным ПО и Ansys Fluent

Обе кривые на графике не представляют собой ожидаемый параболический профиль, что объясняется тем, что при выбранных параметрах расчетной области и свойств жидкости на выходе из области течение еще не является развитым и для получения параболического профиля необходимо рассматривать область большей длины, что скажется на наглядности приводимых полей результатов. Задачей приведенного сравнения является прямое сравнение двух расчетных кодов и оценка результатов при идентичных расчетных параметрах, поэтому в данном случае валидация на аналитическое решение не производится.

Максимальное различие решений составило:

$$\Delta u_{max} = \frac{0,9826 - 0,9547}{0,9826} * 100\% = 2,83\%$$

Анализируя полученные результаты, можно сделать выводы о качественном совпадении полей горизонтальной компоненты скорости между

расчетами в разработанном ПО и *Ansys Fluent*, а также о количественной близости скоростей в одном и том же сечении. Количественное отличие приведенных решений мало и может объясняться различием в используемых численных методах.

На рисунках 3.8 и 3.9 приведены сравнения для вертикальной компоненты скорости и поля давления в рассматриваемом расчетном случае.

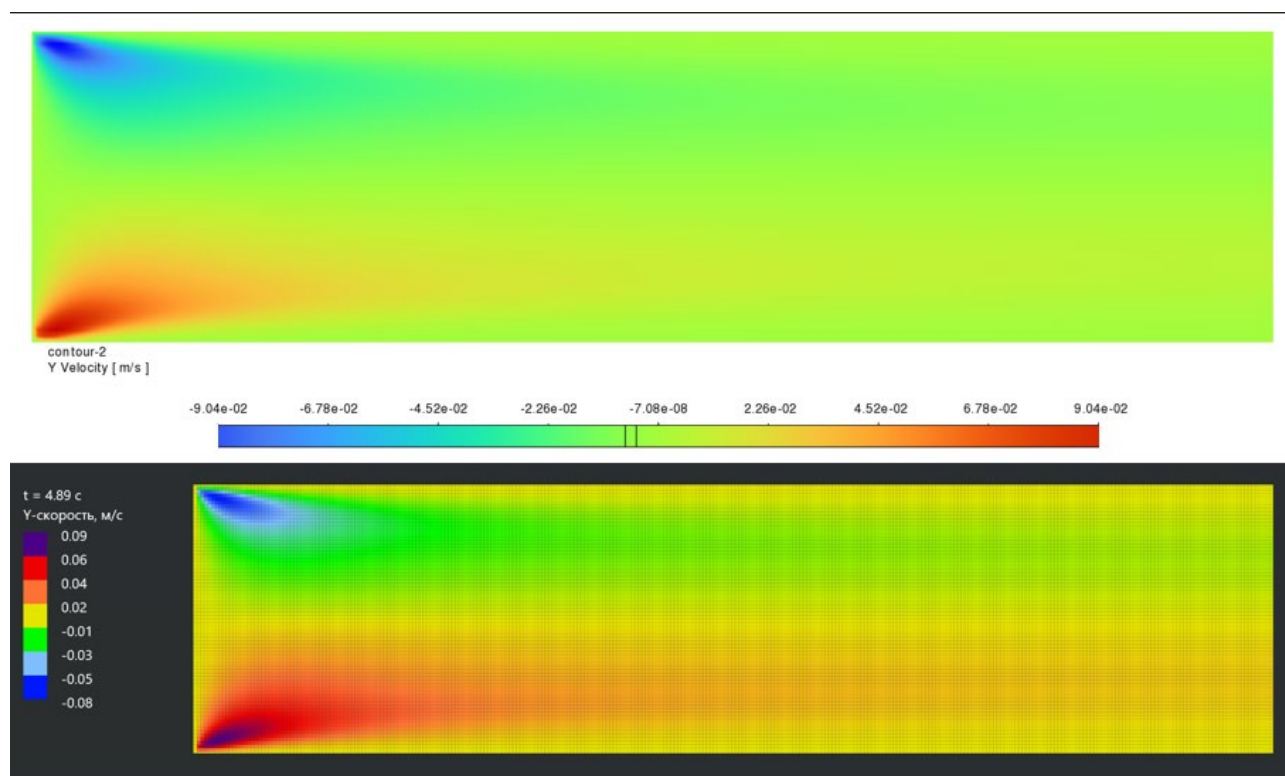


Рисунок 3.8 – Вертикальная компонента скорости в сравнении разработанного ПО с *Ansys Fluent*.

Из приведенных полей вертикальной компоненты скорости можно сделать вывод о качественном совпадении результатов и количественной близости как минимум до 2-го знака после запятой.

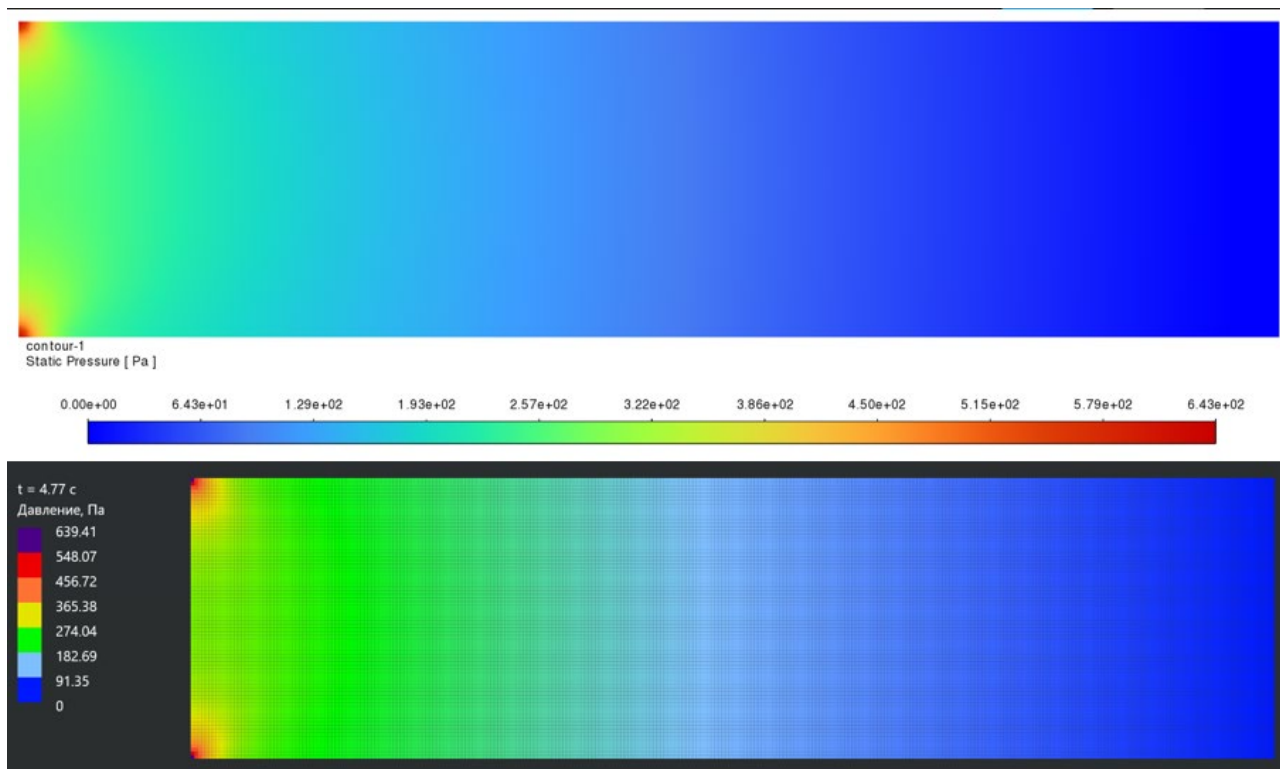


Рисунок 3.9 – Поле давления в сравнении разработанного ПО и Ansys Fluent.

Из приведенных полей давления также делаются выводы о качественном совпадении результатов. Максимальное давление фиксируется в левых верхнем и правом углу и различие между разработанным ПО и Ansys Fluent составляет:

$$\Delta p_{max} = \frac{643 - 639}{643} * 100\% = 0,6\%$$

3.2 Обтекание препятствий в канале

Другой классической задачей гидродинамики является исследование обтекания потоком жидкости тел различной формы. Ниже приведены поля линий тока при обтекании тел круглой формы (сечения цилиндра) и квадратной формы.

3.2.1 Обтекание цилиндра круглого сечения

На рисунках 3.10 и 3.11 приведено сравнение линий тока жидкости между разработанным ПО и результатами из статьи [11]. Каждой линии тока (линии одинакового расхода) в разработанном ПО соответствует оттенок цвета от синего до фиолетового в порядке «синий – голубой – зеленый – желтый – оранжевый – красный – фиолетовый».

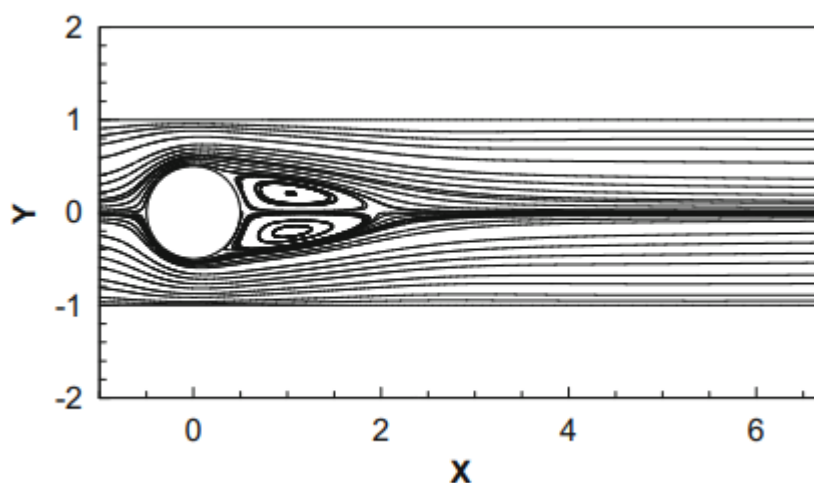


Рисунок 3.10 – Линии тока при обтекании круглого цилиндра несжимаемой жидкостью (статья) при $Re = 100$

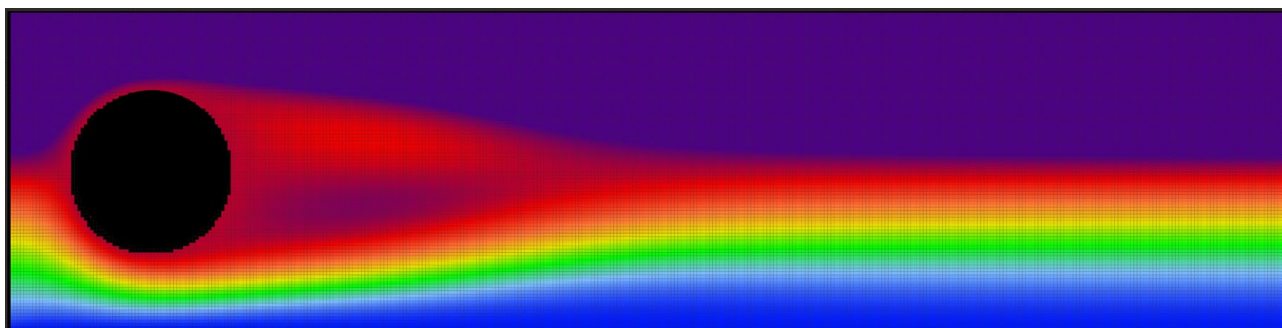


Рисунок 3.11 – Линии тока при обтекании круглого цилиндра несжимаемой жидкостью (разработанное ПО) при $Re = 100$

Оба результата получены при числе Рейнольдса $Re = 100$ и отношении диаметра цилиндра к ширине канала $\frac{D}{H} = \frac{1}{2}$. Скорость на входе в область u_0 .

Параметры расчета:

– $H = 2$ м; $L = 8$ м; $D = 1$ м

– $\rho = 1 \frac{\text{кг}}{\text{м}^3}$; $\mu = 0,01$ Па · с; $u_0 = 1 \frac{\text{м}}{\text{с}}$

Анализируя полученные результаты, можно сделать выводы о качественной близости полученных результатов:

– Огибающие линии тока в обоих случаях образуют «факел», сужающийся после прохождения жидкостью высшей точки цилиндра.

– За цилиндром образуются 2 статичных (не перемещающихся в пространстве) вихря, ниже и выше центральной оси канала.

На рисунках 3.12 и 3.13 приведены линии тока обтекания круглого цилиндра при числе Рейнольдса $Re = 150$.

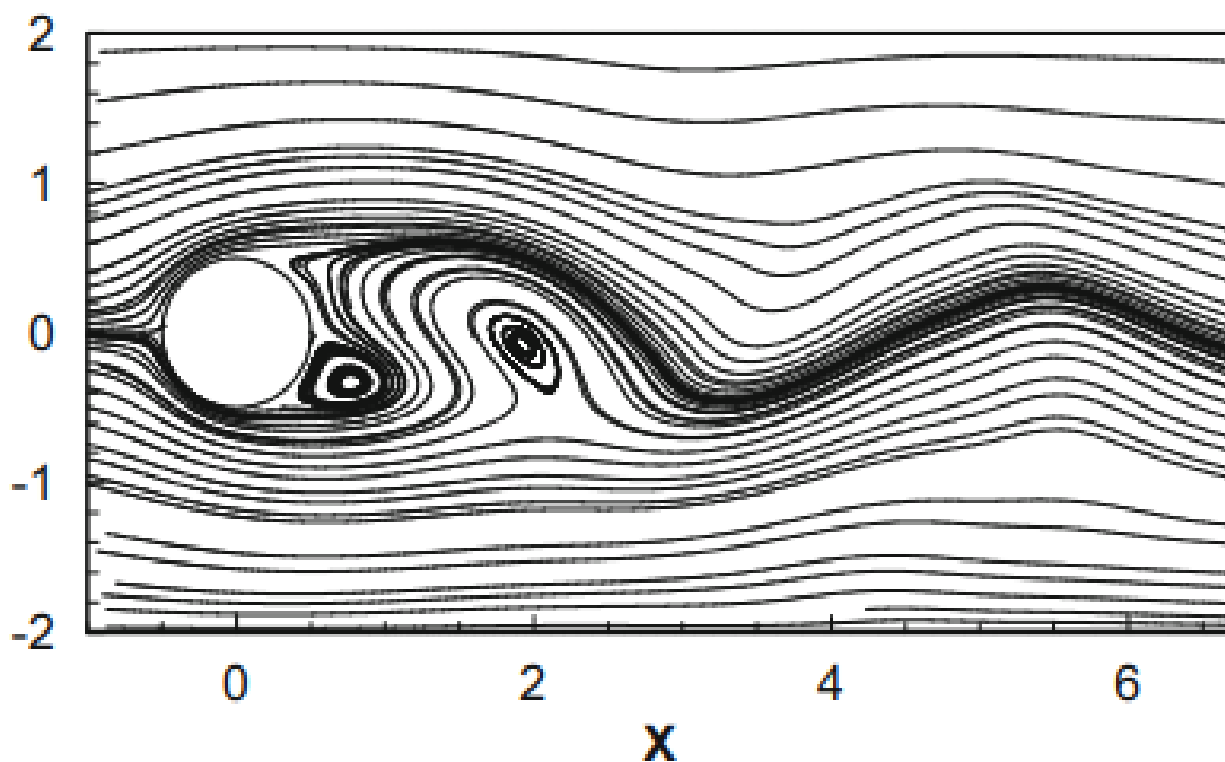


Рисунок 3.12 – Линии тока при обтекании круглого цилиндра несжимаемой жидкостью (статья) при $Re = 150$

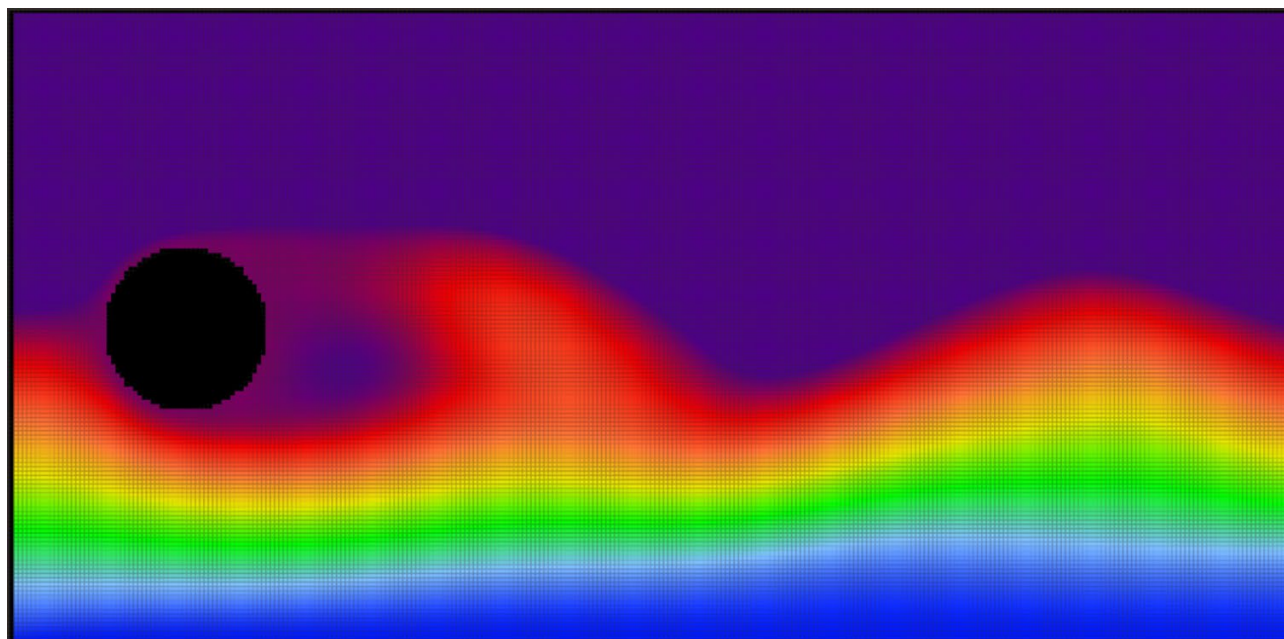


Рисунок 3.13 – Линии тока при обтекании круглого цилиндра несжимаемой жидкостью (разработанное ПО) при $Re = 150$

Параметры расчета:

– $H = 4 \text{ м}; L = 8 \text{ м}; D = 1 \text{ м}$

– $\rho = 1 \frac{\text{кг}}{\text{м}^3}; \mu = 0,0066 \text{ Па} \cdot \text{с}; u_0 = 1 \frac{\text{м}}{\text{с}}$

Анализируя полученные результаты, можно сделать выводы о качественной близости полученных результатов:

– В сравнении с предыдущим расчетным случаем, теперь за цилиндром образуется вихревая дорожка Кармана, то есть с постоянной периодичностью от задней поверхности цилиндра отделяются вихри жидкости.

– Момент образования вихрей приведен на рисунках выше. Качественно линии тока, приведенные в статье и полученные в разработанном ПО близки: в выбранный момент времени можно наблюдать два вихря – один уже оторвавшийся от цилиндра и один образующийся внизу цилиндра.

3.2.2 Обтекание цилиндра квадратного сечения

Аналогичные результаты получены для обтекания цилиндра квадратного сечения. Валидация результатов произведена на статью [8]. Линии тока приведены на рисунках 3.14 и 3.15.

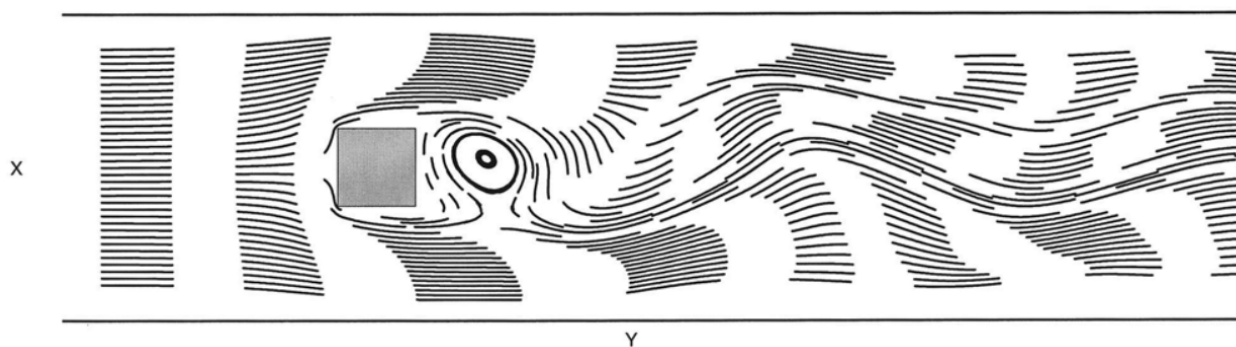


Рисунок 3.14 – Линии тока при обтекании квадратного цилиндра несжимаемой жидкостью (статья)

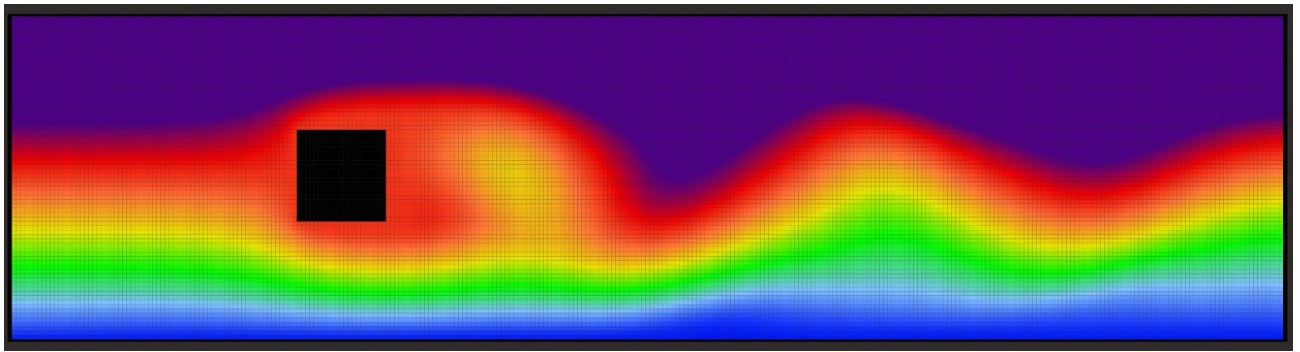


Рисунок 3.15 – Линии тока при обтекании квадратного цилиндра несжимаемой жидкостью (разработанное ПО)

Оба результата получены при числе Рейнольдса $Re = 1000$. Скорость на входе в область u_0 .

Параметры расчета:

$$- H = 3,8 \text{ м}; L = 15 \text{ м}; D = 1 \text{ м}$$

$$- \rho = 1 \frac{\text{кг}}{\text{м}^3}; \mu = 0,0001 \text{ Па} \cdot \text{с}; u_0 = 1 \frac{\text{м}}{\text{с}}$$

Для сравнения выбран момент времени, в который качественная картина течения соответствует результату, приведенному в статье:

– В обоих результатах за цилиндром находится вихрь, поднимающийся из нижних слоев жидкости.

– Линии тока за вихрем имеют схожую форму.

Также произведено сравнение обтекания цилиндра квадратного сечения в разработанном ПО и *Ansys Fluent* [12].

Параметры расчета:

$$- H = 0,5 \text{ м}; L = 4 \text{ м}; D = 0,18 \text{ м}$$

$$- \rho = 1000 \frac{\text{кг}}{\text{м}^3}; \mu = 0,001 \text{ Па} \cdot \text{с}; u_0 = 1 \frac{\text{м}}{\text{с}}$$

$$- \text{Итоговое число Рейнольдса } Re = 180000$$

Результаты сравнения, представленные в виде анимации изменения во времени скалярного поля скоростей в расчетной области приведены в Приложении 1.

Анализ результатов позволяет сделать следующие выводы:

– Оба расчета показывают качественно близкий установившийся режим вихревой дорожки: образующиеся вихри близки по форме и периоду их образования.

– Расчеты отличаются временем переходного процесса от начала течения в канале к устоявшемуся режиму вихревой дорожки: в расчете, произведенном в разработанном ПО, переходный процесс занимает большее время, что может объясняться различными использованными численными методами, лежащими в основе ПО.

ЗАКЛЮЧЕНИЕ

В данный момент среди компаний, занимающимися инженерными расчетами существует запрос на создание современного ПО, качество и комфорт использования которого было бы сопоставимо с зарубежными аналогами, более не представленными на рынке. В рамках данной работы разработан прототип ПО, решающий двумерную нестационарную задачу динамики несжимаемой вязкой жидкости методами численного моделирования.

В ходе работы разработан соответствующий численный метод, основанный на методе конечных объемов и осуществляющий интегрирование уравнений динамики по времени неявной схемой интегрирования путем минимизации невязок уравнений с наперед заданной точностью. Решение производится в эйлеровой постановке на статичной ортогональной сетке с поддержкой изменения геометрии.

Итоговое ПО имеет все необходимые настройки решателя, такие как задание ГУ из представленных 3-х типов, формирование препятствий в расчетной области, параметры жидкости и параметры расчета, включая точность и итерационные параметры. В процессе решения пользователь имеет доступ к невязкам всех трех решаемых уравнений для отслеживания сходимости итерационного процесса. После расчета к выводу доступны такие основные результаты гидродинамического расчета, как компоненты скорости, скалярное и векторное поле скоростей, поле давления и линии тока.

Программная реализация задействует наиболее современные инструменты разработки под ОС *Microsoft Windows* 10/11. Решатель программы написан на ЯП C++/WinRT, а графический пользовательский интерфейс задействует современные графические инструменты *WinUI* 2.8. Программа поддерживает многопоточные вычисления и асинхронную работу компонентов.

Произведена валидация разработанного ПО на существующие аналитические решения, расчетное ПО *Ansys Fluent* и результаты статей. Показано качественное и количественное совпадение результатов разработанного ПО с аналитическим и численными во *Fluent*'е решениями для

задачи течения жидкости в плоском канале. Показано качественное совпадение линий тока для низкорейнольдсовых обтеканий цилиндра с результатами, приведенными в научных статьях. Показана качественная близость расчета вихревой дорожки в разработанном ПО с аналогичным расчетом в *Ansys Fluent*.

Таким образом в работе создан прототип расчетного ПО для задач гидродинамики, который может быть применен к решению нестационарных двумерных задач, допускающих использование ортогональной сетки.

Среди возможных путей развития ПО можно отметить следующие:

- Поддержка моделей турбулентности;
- Поддержка тепловых эффектов течения;
- Экспорт результатов для дальнейшего использования в постпроцессорах по типу *ParaView*;
- Импорт сеток из сторонних программ и адаптация программы под произвольные сетки;
- Развитие модели поведения пропанта в жидкости. На данный момент в ПО существует поддержка запуска частиц пропанта в расчетную область, однако движение этих частиц определяется лишь рассчитанным заранее полем скоростей, и они не влияют на его формирование. Внедрив в программу решение связанной задачи распространения пропанта (с учетом влияния прочих сил, таких как сила тяжести) и задачи гидродинамики, она может применяться как ПО для решения задач в нефтегазовой области.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Лойцянский Л.Г. Механика жидкости и газа. Москва: Издательство "Наука", 1970. 469-472 с.
2. Седов Л.И. Механика сплошной среды. Наука, 1970. 166-170 с.
3. Федоренко Р.П. Введение в вычислительную физику. Москва: Издательство Московского физико-технического института, 1994. 92-99 с.
4. Яненко Н. Метод дробных шагов решения многомерных задач математической физики. Новосибирск: Академия наук СССР, Сибирское отделение, Вычислительный центр, 1967.
5. Batchelor G.K. An Introduction to Fluid Dynamics. Cambridge: Cambridge University Press, 2000. 142-148 pp.
6. Holman J.P., Lloyd J.R. Computational Fluid Dynamics. The basics with applications. 2nd ed. New York: McGraw-Hill, Inc., 1995. 4-10 pp.
7. Williams A. C++ Concurrency in Action. New York: Manning Publications Co., 2012. 15-20 pp.
8. Brandon D.J., Aggarwal S.K. A Numerical Investigation of Particle Deposition on a Square Cylinder Placed in a Channel Flow // Aerosol Science & Technology, Vol. 34, No. 4, Nov 2001. pp. 340-352.
9. Chorin A.J. A numerical method for solving incompressible viscous flow problems // Journal of Computational Physics, Vol. 2, No. 1, August 1967. pp. 12-26.
10. Eymard R., Gallouet T., Herbin R. Finite Volume Methods // Handbook of Numerical Analysis, Vol. 7, September 1997. pp. 713-1020.
11. Singha S., Sinhamahapatra K.P. Flow past a circular cylinder between parallel walls at low Reynolds numbers // Ocean Engineering, Vol. 37, No. 8-9, June 2010. pp. 757-769.
12. Kapeilsh K. To study the flow over a cylinder and Von Karman vortex street in Ansys Fluent. [Электронный ресурс] // Skill-Lync: [сайт]. [2021]. URL: <https://skill-lync.com/student-projects/to-study-the-flow-over-a-cylinder-and-von-karman-vortex-street-in-ansys-fluent> (дата обращения: 29.05.2023).

13. Kumaran J. Getting started with the Windows UI 2 Library [Электронный ресурс] // Microsoft Learn: [сайт]. [2023]. URL: <https://learn.microsoft.com/en-us/windows/apps/winui/winui2/getting-started> (дата обращения: 13.04.2023).
14. White S., Wojciakowski M. Concurrency and asynchronous operations with C++/WinRT [Электронный ресурс] // Microsoft Learn: [сайт]. [2023]. URL: <https://learn.microsoft.com/en-us/windows/uwp/cpp-and-winrt-apis/concurrency> (дата обращения: 11.05.2023).
15. White S., Wojciakowski M. C++/WinRT [Электронный ресурс] // Microsoft Learn: [сайт]. [2023]. URL: <https://learn.microsoft.com/en-us/windows/uwp/cpp-and-winrt-apis/> (дата обращения: 14.04.2023).
16. Whitney T., Sharkey K. Concurrency Runtime [Электронный ресурс] // Microsoft Learn: [сайт]. [2023]. URL: <https://learn.microsoft.com/en-us/cpp/parallel/concrt/concurrency-runtime?view=msvc-170> (дата обращения: 11.05.2023).

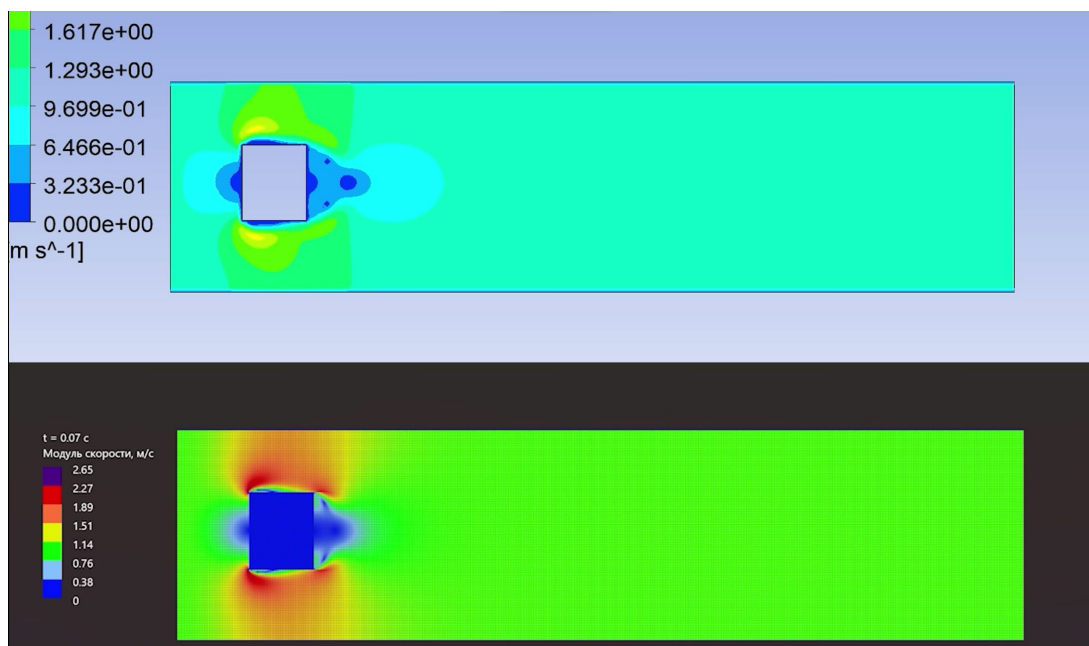
Приложение 1. Сравнение динамики обтекания цилиндра квадратного сечения в разработанном ПО и *Ansys*

Графический анимированный материал сравнения доступен по ссылке:

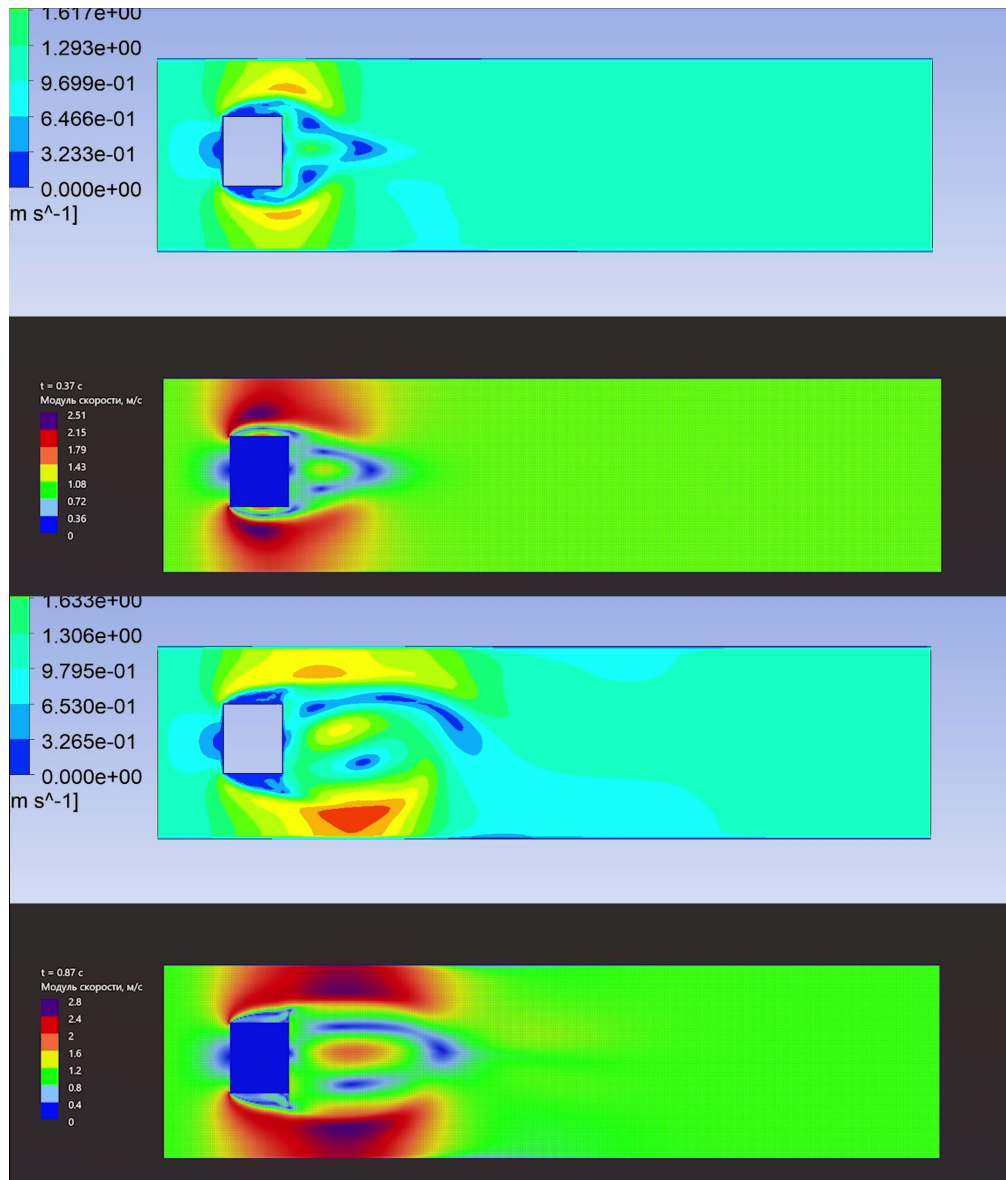
<https://1drv.ms/v/s!AmFFcwoPNwweqPNNO6yOcrQyW9g0oA?e=IWRkV7>

Ниже представлены ключевые моменты.

Начало движения:



Развитие движения:



Сформированная вихревая дорожка:

