Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа прикладной математики и вычислительной физики

Курсовая работа
«Расчет пространственных дифференциальных операторов по методу конечных объемов»
по дисциплине:
«Вычислительная гидродинамика»

Работу выполнил
студент гр.5040301/40501 Д.Ю. Кучиев

Преподаватель А.А. Пожилов

16 января 2025 г.

Содержание

1 Задание к курсовой работе.

Необходимо разработать программу для численного решения уравнения конвективно-диффузионного переноса температуры в заданном поле скоростей по методу контрольного объема (МКО), применительно к задаче о течении в каверне с движущейся крышкой и с разнонагретыми стенками. Для выполнения расчета необходимо написать программу которая будет численно вычислять пространственные операторы по МКО и вследствии позволит разрешить уравнения конвективно-диффузионного переноса. Программа должна быть выполнена на языке программирования Fortran. Индивидуальный вариант: 3.

2 Порядок выполнения работы.

1. Провести расчет градиента скалярного поля $p(x, y)$ в центрах ячеек с помощью метода Грина-Гаусса и Грина-Гаусса с итерациями.

2. Провести расчет дивергенции произведения скалярной величины на вектор, $div(p \cdot \vec{V})$. Расчет скалярной величины на грани необходимо провести с помощью центральной (Central), противопоточной схемы первого и второго порядка (FOU и SOU соответственно).

3. Провести расчет лапласиана скалярного поля $p(x, y)$.

4. Провести расчет ротора векторного поля $\vec{V}(x, y)$.

5. Решить уравнение конвективно-диффузионного переноса скалярной величины (температуры) для заданного скалярного поля скоростей.

3 Описание метода конечных объемов (МКО)

3.1 Дискретизация уравнений.

Метод конечных объемов (МКО) — это численный метод, который широко применяется для решения дифференциальных уравнений в частных производных. Основной особенностью метода является его способность сохранять физические законы (массу, импульс, энергию) на уровне дискретной формы уравнений, в отличие от метода конечных разностей, который работает с узловыми точками. Можно это интерпритировать так, что МКР работает с узлами и точченными значениями, а МКО с интервалами.

Для применения МКО расчетную область делят на некоторые объемы (ячейки), которые в совокупности составляют всю исходную область. Вершины формируют узлы расчетной сетки.

Для построения контрольного объема используют один из двух методов: объемно-центрированный (Cell-centered), при котором решение ищется в центрах конечного

объема, и вершино-центрированный (Vertex-based), при котором решение ищется в узлах расчетной сетки и конечные объемы строятся вокруг них (рисунок 3.1).

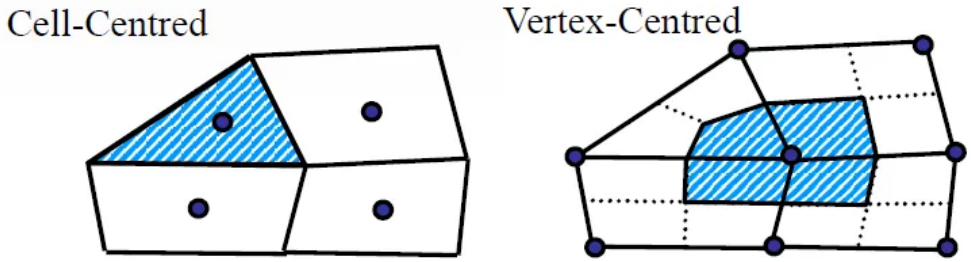


Рис. 3.1: Схематическое представление построения КО с помощью различных подходов.

Для потсроения КО рассмотрим неподвижный объем Ω, изображенный на рисунке 3.2. Поверхность КО обозначается σ, соответственно элемент поверхности и объема обозначаются как $\delta\sigma$ и $\delta\Omega$. Так для определения поток проходящих сквозь поверхность объема введем нормаль к элементарной поверхност таким образом, чтобы она была направлена наружу, тогда получим так называемый вектор элемента площадки $\delta\vec{\sigma} = \vec{n} \cdot \delta\sigma$.

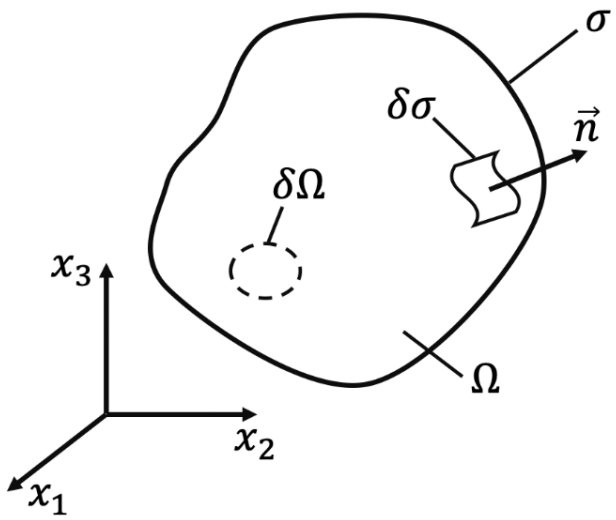


Рис. 3.2: Иллюстрация КО.

Рассмотрим задачу конвективно-диффузионного переноса скалярной величины ϕ. Запишем уравнение описывающее этот процесс сразу в интегральной формулировке (3.1). Переход от интегральной формулировки к дифференциальной осуществляется путем применения теоремы Остроградского-Гаусса.

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\phi\delta\Omega + \oint_{\sigma}\vec{F}\cdot\delta\vec{\sigma} = \int_{\Omega}\rho S_{\phi}\delta\Omega \tag{3.1}$$

В свою очередь вектор плотности потока $\vec{F}$ можно разложить на конвективную и диффузионную составляющие:

$$\vec{F} = \vec{F}_{\text{конв}} + \vec{F}_{\text{диф}} = \rho\vec{V}\phi + \vec{q}_\phi \tag{3.2}$$

Приведем континуальную форму получившихся уравнений к дискретной. Для этого введем следующие обозначения: $(...)_c$ - в центре ячейки, $(...)_f$ - в центре грани. Применяя теорему о среднем для объемных и поверхностных интегралов, получим:

$$\frac{\partial}{\partial t}\int_\Omega \rho\phi\delta\Omega \simeq \frac{\partial(\rho\phi)_c}{\partial t}\Omega \tag{3.3}$$

$$\int_\Omega \rho S_\phi \delta\Omega \simeq (\rho S_\phi)_c \cdot \Omega \tag{3.4}$$

Для поверхностного интеграла ввиду замкнутости мы не можем напрямую применить теорему о среднем, потому разобьем интеграл на сумму интегралов:

$$\oint_\sigma \vec{F}\cdot\delta\vec{\sigma} = \sum_f \int_{\sigma_f} \vec{F}\cdot\vec{\sigma}_f \simeq \sum_f \vec{F}_f\cdot\vec{\sigma}_f \tag{3.5}$$

В результате получаем полудискретный (ввиду того, что осталась недискретная производная по времени) аналог для уравнения конвективно-диффузионного переноса:

$$\frac{\partial(\rho\phi)_c}{\partial t}\Omega + \sum_f \vec{F}_f\cdot\vec{\sigma}_f = (\rho S_\phi)_c\cdot\Omega \tag{3.6}$$

Тогда для проведения дальнейшего расчета мы уже можем определить некоторый алгоритм или порядок действий:

1. Расчет центров КО.
2. Расчет центров граней.
3. Расчет объем ячеек.
4. Нахождение векторов площади грани $\vec{\sigma}_f$.
5. Выбор способа расчет потоков $\vec{F}_f$ на грани, используя значения в центрах КО.
6. Схема сведения задачи.

3.2 Описание расчета геометрических характеристик.

Для использования МКО необходимо определить геометрические характеристики КО. Для этого рассмотрим ячейку полиэдральной формы. На рисунке 3.3 представлена ее грань. Как известно, любой многоугольник можно представь как свокоупность треугольников, таким образом разобьем грань на треугольники с вершиной в центре грани.

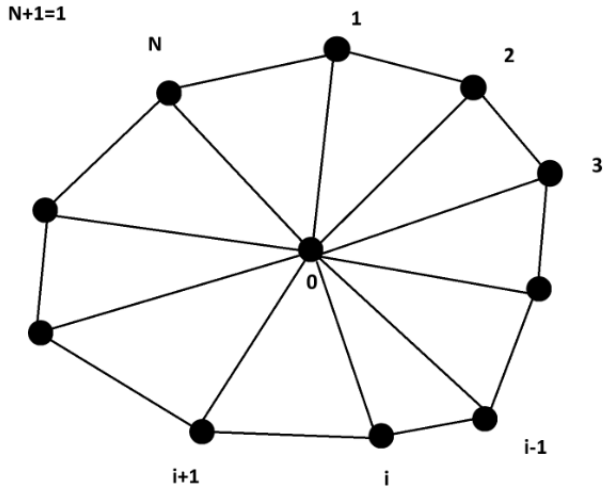


Рис. 3.3: Грань ячейки произвольной формы.

Вешины данной грани пронумерованы от 1 до N. При этом номер вершины N+1 соответствует вершине 1.

Вектор площади грани определяются таким образом, что при разбиении на треугольники с одной из вершин в точке 0 определяется как сумма площадей данных треугольников.

$$\vec{\sigma}_f = \sum_i \vec{\sigma}_{0,i,i+1} = \frac{1}{2}\sum_i (\vec{r}_0 - \vec{r}_i) \times (\vec{r}_0 - \vec{r}_{i+1}) = \frac{1}{2}\sum_i \vec{r}_i \times \vec{r}_{i+1} \tag{3.7}$$

Таким образом, учитывая что индексы обходятся по кругу и тем самым уничтожают одно из слагаемых, получаем, что вектор поверхности грани не зависит от выбора точки 0. Положим, что $\vec{r}_0 = \vec{r}_1$, тогда можно получить итоговую формулу для вычислений $\vec{\sigma}_f$:

$$\vec{sigma}_f = \frac{1}{2}\sum_{i=2}^{N-1} (\vec{r}_1 - \vec{r}_i) \times (\vec{r}_1 - \vec{r}_{i+1}) \tag{3.8}$$

Для вычисления объема ячейка произвольной формы разбивается на пирамиды с основанием σ_f и вершиной в точке 0. ТОгда можно записать:

$$\Omega = \frac{1}{3}\sum_f (\vec{r}_f - \vec{r}_0) \cdot \vec{\sigma}_f$$

Применяя теорему Остроградского-Гаусса перейдя к дискретной форме получим, что

$$\sum_f \vec{\sigma}_f = 0$$

Тогда выражение примет вид:

$$\Omega = \frac{1}{3}\sum_f \vec{r}_f \cdot \vec{\sigma}_f = \frac{1}{3}\sum_f (\vec{r}_f - \vec{r}_{f_1}) \cdot \vec{\sigma}_f \tag{3.9}$$

Последнее равенство реализует достижение вычислений одного порядка, выбрав центр произвольной грани как начало отсчета. Это позволительно ввиду отсутствия зависимости объема от точки отсчета.

Центр грани $\vec{r}_f$ определяется несколькими способами. Первый способ подразумевает вычисление путем среднего арифметического. Существуют еще способ установления центра грани в центре тяжести поверхности или центра тяжести контура. Лучшим остается последний, так как в нем отсутствует зависимость от $\vec{r}_0$. Запишем выажение для центра тяжести контура:

$$\vec{r}_f = \sum_i \frac{1}{2}(\vec{r}_i + \vec{r}_{i+1}) \frac{|\vec{r}_{i+1} - \vec{r}_i|}{\sum_k |\vec{r}_{k+1} - \vec{r}_k|} \tag{3.10}$$

Последний способ определения имеет преимущество над способом определения с помощью центра тяжести в случае вытянутой ячейки, так как определяет более точно нормаль к поверхности, что приводит к меньшим погрешностям при вычислении, например, диффузионных потоков.

Аналогично для расчета центра объема: среднее арифметическое, центр тяжести, центр тяжести контура или центр тяжести каркаса. Лучшим вариантом является определения при помощи центра тяжести контура:

$$\vec{r}_c = \sum_f \vec{r}_f \cdot \frac{|\vec{\sigma}_f|}{\sum_k |\vec{\sigma}_k|} \tag{3.11}$$

Так как в данной работ рассматривается структурированная расчетная сетка с четырехугольными ячейками, вследствии чего выражения для расчетов геометрических характеристик преобразуются и сильно упрощаются.

Рассматривая двумерные расчетные сетки, каждая грань ячейки представляет собой отрезок, а ее площадь равна длине ребра. Ввиду специфики сетки можно разбить нормали на две большие группы: грань расположенная между узлами i,j и i,j+1 ($\vec{n_{i,j}^I}$) и грань расположенная между узлами i,j и i+1,j ($\vec{n_{i,j}^J}$). Найдем общий вид формулы для этих двух вариантов.

$$\vec{r}_I = (r_{I,x}, r_{I,y}) = (x_{i,j+1} - x_{i,j}, y_{i,j+1} - y_{i,j}) \tag{3.12}$$

$$\vec{n_{i,j}^I} = (r_{I,y}, -r_{I,x}) \tag{3.13}$$

Аналогично для другой группы:

$$\vec{r}_J = (r_{J,x}, r_{J,y}) = (x_{i+1,j} - x_{i,j}, y_{i,j+1} - y_{i+1,j}) \tag{3.14}$$

$$\vec{n_{i,j}^{J}} = (r_{J,y}, r_{J,x}) \tag{3.15}$$

То есть эти веткора получаются путем развороте вектора грани на 90 градусов. Тогда получим, что все нормали $\vec{n_{i,j}^{I}}$ будут направлены вправо, а $\vec{n_{i,j}^{J}}$ будут направлены вверх (на скошенных сетках нормали направлены не строго по горизонтали а как-то иначен но в направлении увеличения индекса). Схематично это представлено на рисунке 3.4. В процессе вычисления пространственных операторов направление нормали будет выбираться таким образом, чтобы она была внешней, а это достигается путем изменения знаков в координатах вектора нормали.

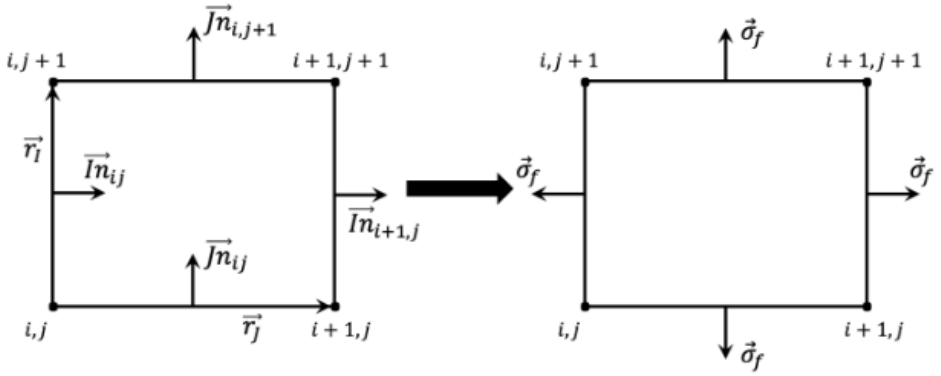


Рис. 3.4: Схема расположение векторов нормалей к граням.

Центр грани расчитывается как среднее арифметическое координат соответствующих узлов:

$$\vec{r}_f = (\frac{x_{i,j+1} + x_{i,j}}{2}, \frac{y_{i,j+1} + y_{i,j}}{2}) \tag{3.16}$$

или

$$\vec{r}_f = (\frac{x_{i+1,j} + x_{i,j}}{2}, \frac{y_{i+1,j} + y_{i,j}}{2}) \tag{3.17}$$

В зависимости от того какая это грань формула будет отличаться соответственно.

Центр объема в данном случае расчитывается следующим образом:

$$\vec{r}_c = \sum_f \vec{r}_f \frac{\sigma_f}{\sum_k \sigma_k} \tag{3.18}$$

В данной работе также введены фиктивные ячейки, которые имеют индексы: $i = \{0, NI\}$ при $j = \overline{1, NJ-1}$ и $j = \{0, NI\}$ при $i = \overline{1, NI-1}$. Они являются заграничными и введены для упрощения расчтов потоков для приграничных ячеек. Для рассматриваемых точек центры объемов совпадают с центрами граней с теми же индексами. Исключением является заграничные ячейки с i=0, j=0. Для них соответствующие грани имеют индексы i=1, j=1.

Объем или в данном случае площадь ячейки вычисляется, как сумма площадей двух треугольников создаваемых при проведение диоганали в четырехугольной ячейке:

$$\Omega_{i,j} = \frac{1}{2}(\vec{n^I_{i,j}} \cdot \vec{r_{i,j}} + \vec{n^j_{i,j}} \cdot \vec{r_{i,j}}) \tag{3.19}$$

$\vec{r_{i,j}} = (x_{i+1,j+1} - x_{i,j}, y_{i+1,j+1} - y_{i,j})$ – Вектор диагонали

4 Описание расчета пространственных дифференциальных операторов.

4.1 Расчет градиента скалярной величины.

Используя теорему Остроградского-Гаусса и теорему о среднем можем получить следующее выражение:

$$\sum_f \phi_f \vec{\sigma}_f = \oint_\sigma \vec{n}\phi d\sigma = \int_\Omega \nabla\phi d\Omega \simeq (\nabla\phi)_c \Omega \tag{4.1}$$

То есть получаем, что вычисление градиента в центре ячейки сводится к суммированию произведений значений величины на гранях и вектора грани. Ввиду того, что мы определили объем ячейки и вектор грани, то необходимо как-то найти значение ϕ на грани. Для этого используется линейная интерполяция на грань:

$$\phi_f = \phi_M = (1-\beta)\phi_L + \beta\phi_R \tag{4.2}$$

, где $\beta = \frac{|\vec{P_L M}|}{|\vec{P_L M}| + |\vec{P_R M}|}$ - весовой коэффициент.

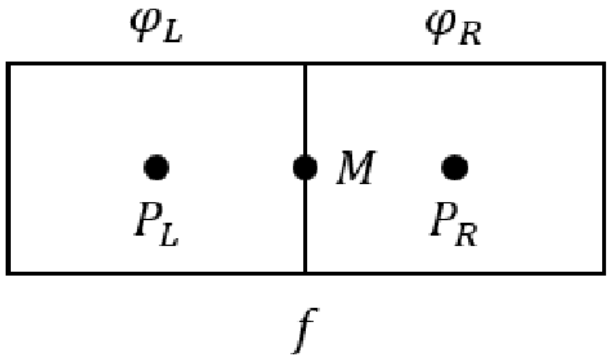


Рис. 4.1: Схема линейной интерполяции.

Для получения второго порядка точности в приграничных ячейках значение скалярной функции в центре граничной грани b вычислялось следующим образом:

$$\phi_b = \frac{1}{2}(\phi_p + \phi_0)$$

где ϕ_P– значение ϕ в центре приграничной ячейки, ϕ_0 – значение в центре фиктивной заграничной ячейки, центр которой определялся как $\vec{r}_0 = 2\vec{r}_b - \vec{r}_p$ поскольку заграничная ячейка может быть произвольной формы.

В случае скошенной расчетной сетки линейная интерполяция плохо работает и дает погрешность. Поэтому необходима применять так называемый метод Грина-Гаусса с итерациями:

$$\phi_M = \phi_E + E\vec{M} \cdot (\nabla\phi)_E \tag{4.3}$$

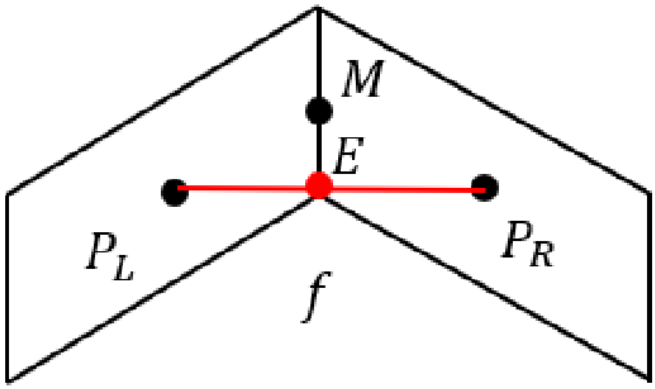


Рис. 4.2: Скошенная сетка.

Получаем итерационный процесс, где на первом шаге мы используем линейную интерполяцию на грань и поулчаем значение в точке E и считаем градиенты в центрах ячеек, а затем интерполируем линейно эти градиенты на грань и с их помощью уточняем значение в точке M пересчитывая градиенты. То есть $(\nabla\phi)_E = (1-\beta)(\nabla\phi)_L + \beta(\nabla\phi)_R$.

Таким образом итерационный процесс выглядит так, а применяя его на скошенной сетке можно улучшить значение градиентов:

$$\begin{cases} (\nabla\phi)_c^k = \frac{1}{\Omega}\sum_f \phi_f^k \vec{\sigma}_f \\ \phi_f^k = \phi_E + E\vec{M} \cdot (\nabla\phi)_E^{k-1} \end{cases} \tag{4.4}$$

4.1.1 Расчет дивергенции.

Запишем теорему Остроградского-Гаусса для дивергенции произведения скалярной величины на вектор $\vec{V}$:

$$\sum_f \phi_f \vec{V}_f \cdot \vec{\sigma}_f \simeq \oint_\sigma \vec{n} \cdot \phi \vec{V} d\sigma = \int_\Omega \nabla \cdot \phi \vec{V} d\Omega \simeq (\nabla \cdot \phi \vec{V})_c \cdot \Omega \tag{4.5}$$

То есть получим:

$$(\nabla \cdot \phi \vec{V})_c = \frac{1}{\Omega} \sum_f \phi_f \vec{V}_f \cdot \vec{\sigma}_f \tag{4.6}$$

Здесь для нахождения значения вектора на грани используется линеная интерполяция. Расчет скалярной величины на грани осуществаляется с помощью следующих схем:

1. Центрально разностная схема - вычисляем значения скалярной величины на грани путем линейной интерполяции:

2. Противопоточная схема первого порядка (FOU) - при использовании данной схемы учитывается знак массового расхода $G_f = \rho \vec{V} \cdot \vec{\sigma}_f$:

$$\phi_f = \begin{cases} \phi_L, \ G_f \geq 0 \\ \phi_R, \ G_f < 0 \end{cases} \tag{4.7}$$

3. Противопоточная схема второго порядка (SOU) также учитывает знак массового расхода

$$\phi_f = \begin{cases} \phi_L + (\nabla \phi)_L \cdot \vec{P_L M}, \ G_f \geq 0 \\ \phi_R + (\nabla \phi)_R \cdot \vec{P_R M}, \ G_f < 0 \end{cases} \tag{4.8}$$

Для расчета скалярной величины на границах расчетной области также используеются заграничные ячейки, но уже со вторым порядком точности:

$$\phi_{f,0} = 2\phi_b - \phi_{f,1} + 4(\phi_{f,1} - \phi_b) + 3(\nabla \phi)_{f,1} \cdot \vec{r}_{1,b} \tag{4.9}$$

4.1.2 Расчет лапласиана скалярного поля.

Аналогично получаем:

$$\int_\Omega \Delta \phi d\Omega = \oint_\sigma \frac{\partial \phi}{\partial n} d\sigma \simeq \sum_f (\frac{\partial \phi}{\partial n})_f \cdot \sigma_f = (\Delta \phi)_c \cdot \Omega \tag{4.10}$$

Здесь необходимо найти значение производной на грани вдоль нормали и необходимо учесть поправку на неортогональность. Тогда получим:

$$(\frac{\partial \phi}{\partial n})_f = \frac{\phi_R - \phi_L}{\vec{P_R P_L}} + (\vec{n}_f - \vec{i}_\xi) \cdot (\overline{\nabla \phi})_f \tag{4.11}$$

Градиент на грани определяется путем линейной интерполяции.

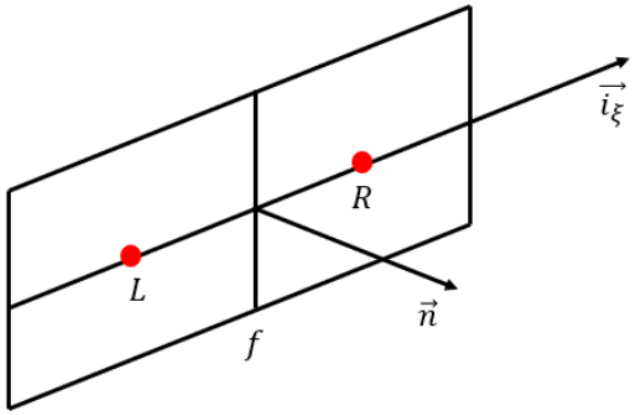


Рис. 4.3: Вычисление проивзодной по нормали для внутренних ячеек.

Для граничных ячеек с учетом поправки на неортогональность получим:

$$(\frac{\partial \phi}{\partial n})_b = \frac{5}{3}\frac{\phi_1 - \phi_b}{\vec{Pb}} - \frac{2}{3}\vec{n}^* \cdot (\nabla\phi)_1 + (\vec{n}^* - \vec{i_\xi}) \cdot (\nabla\phi)_1 \tag{4.12}$$

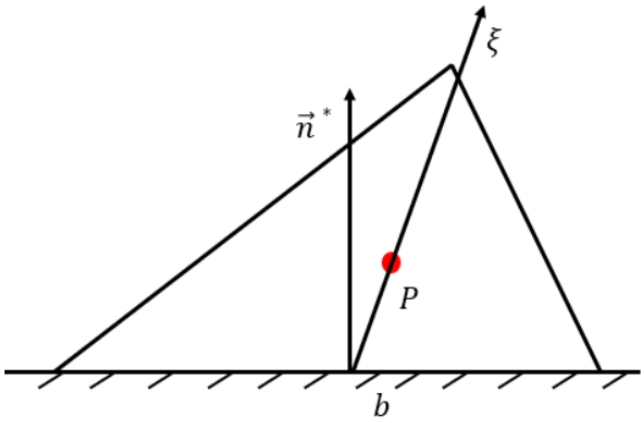


Рис. 4.4: Вычисление проивзодной по нормали для граничных ячеек.

4.1.3 Расчет ротора векторного поля.

Используя ту же самую теорему Гаусса-Остроградского и теорему о среднем можем получить следующее выражение аналогично:

$$(\nabla \times \vec{V})_c = \frac{1}{\Omega}\sum_f \vec{V}_f \times \vec{\sigma}_f \tag{4.13}$$

Поскольку рассматривается двумерная расчетная область, то останет только z компонента ротора, что мы можем интерпритировать как скалярную величину распределенной по расчетной

области.

5 Описание алгоритма расчета уравнения конвективно-диффузионного переноса температуры.

В данной работе рассматривается стационарное течение несжимаемой жидкости в отсутствии объемных источников. С учтом этого уравнение конвективно-диффузионного переноса преобразуется в вид:

$$\oint_{\sigma} (\rho \vec{V} \phi + \vec{q}_{\phi}) \cdot \delta \vec{\sigma} = 0 \tag{5.1}$$

В качестве скалярной функции рассмтривается внутренняя энергия, тогда обозначим удельную внутреннюю энергию как $\phi = e$. Учитывая, что диффузионный поток можно определить используя базовую модель Фурье-Фика, то есть:

$$\vec{q_e} = -\lambda \nabla T \tag{5.2}$$

, где λ - коэффициент теплопроводности, а T - температура. А также взять во внимание тот факт, что внутренняя энергия связан с температурой для несжимаемой жидкости следующим соотношением: $e = c_p T$. Итоговое выражение выглядит следующим образом:

$$\oint_{\sigma} (\rho c_p T \vec{V} - \lambda \nabla T) \cdot \delta \vec{\sigma} = 0 \tag{5.3}$$

Данное уравнение можем обезразмерить, введя масштабы скорости $\vec{V}_s$, температуры T_s и длины L_s. Тогда уравнение преобразуется в слудеющий вид:

$$\oint_{\sigma} (T \vec{V} - \frac{\lambda}{\mu c_p} \frac{\mu}{\rho V_s L_s} \nabla T) \cdot \vec{\sigma} = \oint_{\sigma} (T \vec{V} - \frac{1}{Pr \cdot Re} \nabla T) \cdot \vec{\sigma} = 0 \tag{5.4}$$

Получим уравнение внутренней энергии в безразмерной форме.

Полученное уравнение будем решать методом установления. Для применения данного метода вводится псевдовремя τ, а в уравнении добавляется соответствующая производная температуры по псевдовремени. Ищется стационарное решение, поэтому добавленное слагаемое проподает и получается, что исходное выражение верно.

$$\int_{\Omega} \frac{\partial T}{\partial \tau} \delta \Omega + \oint_{\sigma} (T \vec{V} - \frac{1}{Pr \cdot Re} \nabla T) \cdot \vec{\sigma} = 0 \tag{5.5}$$

В дискретной форме, используя явную схему Эйлера для дискретизации по вермени, обусловленной тем, что нам не важен порядок точности по этому времени, как и шаг, получим для M+1 шага по времени:

$$\frac{T_c^{M+1} - T_c^M}{\Delta\tau_c} = -\frac{1}{\Omega}\sum_f (T\vec{V} - \frac{1}{Pr \cdot Re}\nabla T)_f^M \cdot \vec{\sigma}_f = R_c^M \tag{5.6}$$

Получили формулу для невязки стационарного оператора на текущем временном слое в центре расматриваемой ячейки. Для определения невязки расчитываются конвективный и диффузионные потоки на гранях ячейки, расчет которых был представлен ранее. Шаг по псевдовремени для рассматриваемой ячейки определяется по локальному времени путем следующего соотношения:

$$\frac{1}{\Delta\tau_c} = \frac{1}{\Delta\tau_c^{\text{конв}}} + \frac{1}{\Delta\tau_c^{\text{дифф}}}$$

$$\Delta\tau_c^{\text{конв}} = \text{CFL}\frac{\Omega}{\sum_f |\mathbf{v}_f \cdot \sigma_f|}, \quad \Delta\tau_c^{\text{дифф}} = \text{VNM}\frac{1}{2\sum_f \frac{a_f}{\sigma_f}}$$

где CFL - число Куранта, которое в случае равномерной расчетной сетки равно $CFL = V_s\frac{\Delta t}{\Delta x}$, VNM - число фон Неймана, которое в случае равномерной сетки равно $VNM = D\frac{2\Delta t}{\delta x^2}$. Их значение задаются так, чтобы достичь максимальной скорости сходимости, а устойчивость при этом сохранялась. Обычно их выбирают такими, чтобы они были меньше или равны 1.

6 Описание алгоритма вычислительного кода.

Блок схема для вычисления пространственных операторов и разрешения уравнений конвективно-диффузионного переноса представлена на рисунке 6.1.

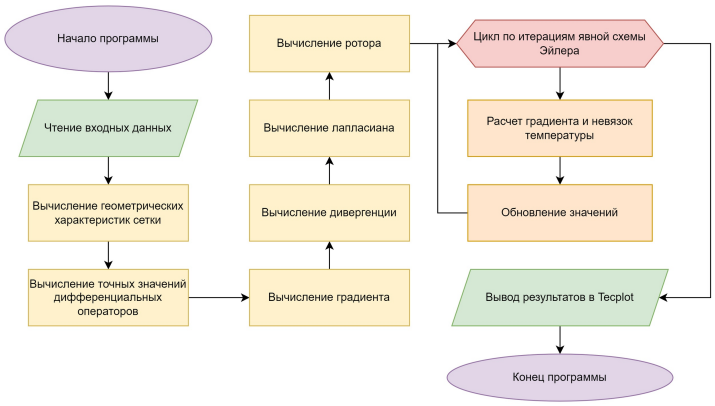


Рис. 6.1: Блок схема вычислительной программы.

Для структурирования кода программа была реализована в модульном варианте. Далее будет описана работа каждого из модулей:

1. Модуль A_Main.f90 Данный модуль является управляющим и основным. В нём производится чтение входного файла `input.txt`, который содержит следующие параметры:

1. Имя файла, содержащего необходимую расчетную сетку;
2. Номер метода расчета градиента давления $\nabla\phi$;
3. Номер схемы расчета скалярной величины при вычислении $\nabla \cdot (\phi \mathbf{V})$;
4. Номер схемы расчета конвективных потоков при решении уравнения переноса температуры T;
5. Масштаб скорости V;
6. Масштаб длины L;
7. Число Рейнольдса Re;
8. Число Прандтля Pr;
9. Число Куранта CFL;
10. Число фон Неймана VNM;
11. Количество итераций при решении уравнения переноса T;

Далее производится чтение количества узлов расчетной сетки N_I, N_J, выделение памяти под массивы и чтение самих узлов расчетной сетки из файла, указанного в `input.txt`. После этого, если решается задача о течении в каверне, заполняются массивы полей T_{flos} и $\mathbf{V}_{flos}$, полученные в ходе расчета в программе Flos. Следующим этапом является расчет геометрических характеристик, который выполняется в модуле `B_CalcMetric.f90`.

После инициализации полей p и $\mathbf{V}$ производится расчет точных значений дифференциальных операторов, поочередное вычисление пространственных дифференциальных операторов и вывод в консоль максимальных значений погрешности. Затем решается уравнение переноса T, а результаты записываются в файл с использованием модуля `B_OutputFields.f90`.

2. Модуль B_CalcMetric.f90 Этот модуль используется для вычисления геометрических характеристик расчетной сетки. В нём для каждой ячейки вычисляются:

- центры граней $\mathbf{r}_f$;
- векторы нормали к грани $\mathbf{I}_n$ или $\mathbf{J}_n$ в зависимости от направления (индекса i или j);
- объём каждой ячейки Ω;
- центр ячейки $\mathbf{r}_c$, с учётом особенностей приграничных ячеек.

3. Модуль B_Functions.f90 Этот модуль содержит функции для:

- вычисления полей p и $\mathbf{V}$;

- расчета точных значений пространственных дифференциальных операторов в виде полиномиальных функций от x и y;
- вычисления значения переменной по методу линейной интерполяции.

4. Модуль B_CalcGradient.f90 Модуль предназначен для численного расчета градиента скалярной функции $\nabla\phi$ методами Грина-Гаусса и Грина-Гаусса с итерациями. Для каждой грани определяются:

- значение ϕ_f;
- вектор площади грани σ_f;
- центр грани $\mathbf{r}_f$;
- индексы соседних ячеек.

5. Модуль B_CalcDivergence.f90 В этом модуле реализован численный расчет дивергенции произведения $\phi \cdot \mathbf{V}$ с использованием схем `Central`, `FOU`, `SOU`.

6. Модуль B_CalcLapl.f90 Реализован численный расчет лапласиана скалярной величины ϕ. Инициализируется массив для лапласиана, а затем вычисляются вектор площади грани σ_f, центр грани $\mathbf{r}_f$ и индексы соседних ячеек.

7. Модуль B_CalcRotor.f90 Модуль отвечает за численный расчет ротора векторного поля $\mathbf{V}$. Инициализация массива и определение геометрических характеристик проводятся аналогично другим модулям.

8. Модуль B_OutputFields.f90 Результаты расчетов записываются в файлы с расширением `.plt` для последующей обработки в `Tecplot`.

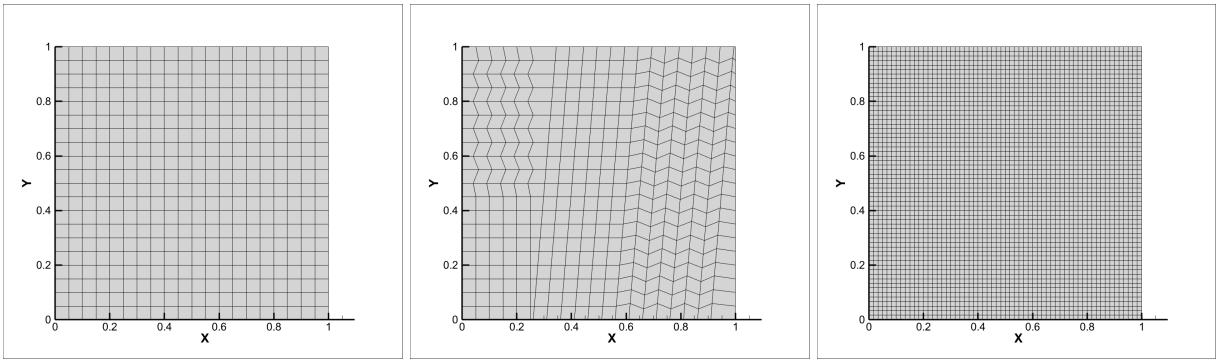
Решение уравнения конвективно-диффузионного переноса температуры Решение задачи о течении в каверне с движущейся крышкой осуществляется итерационно. Инициализируются поле температуры $T_1 = 1$ и граничные условия:

- $T = 1$ на левой стенке ($i = 0$ и $j = 1, N_J - 1$);
- $T = 2$ на правой стенке ($i = N_I$ и $j = 1, N_J - 1$).

В модуле `B_CalcResT.f90` рассчитываются невязки Res_T и значения температуры обновляются на каждой итерации.

Параллельные вычисления Для исследования эффективности распараллеливания процесса использовались конструкции из пакета OpenMP:

- `OMP DO` для автоматического распределения итераций цикла между потоками;
- `private()` для определения локальных копий переменных;



(a) Base mesh (b) Skew mesh (c) Fine mesh

Рис. 7.1: Различные виды сеток.

- `single` для выполнения блока кода одной нитью.

Время работы параллельной области определялось с помощью функции `OMP_GET_WTIME()`, а число потоков задавалось переменной окружения `OMP_NUM_THREADS`.

7 Результаты исследований реализованных методов расчета пространственных дифференциальных операторов.

на рисунке 7.1 представлены три различных расчетные сетки, используемые при вычислении пространственных операторов и отладки программы. Расчетная область представляет собой квадрат. Базовая и скошенная расчетная сетка имеет размеры 21 на 21 узел, имельченная 61 на 61.

При вычислении дифференциальных операторов вычислялась точность расчета с помощью относительной ошибки:

$$Error(...) = |\frac{(...)_{ex} - (...)_{num}}{(...)_{ex}}|,$$

где $(...)_{ex}$ - точное значение дифферениального оператора, $(...)_{num}$ - численное значение дифф. оператора.

Для определения порядка точности производились расчет при одинх и тех же паратрах на базовой сетке и измельченной, определялись ошибки и после порядок находился по формуле:

$$O(...) = log_{\frac{h_b}{h_f}}(\frac{Error(...)_{base}}{Error(...)_{fine}}) \tag{7.1}$$

Для проведения расчетов использовался первый порядок точности так как погрешности этих методов и схем значительно превышают порядок машинной погрешности.

В таблице 1 представлены полиномы различных степеней для скалярного поля p, а также

соответствующие им значения скалярных операторов. При расчете дивергенции необходимо использовать векторное поле, поэтому было определено оно равным: $\vec{V} = (1, 1)$.

Таблица 1: Виды полиномов используемых в дальнеших расчетах с точными операторами

	p	∇p_x	∇p_y	$\nabla \cdot (p\vec{V})$	Δp
P1	$x + 3y$	1	3	4	0
P2	$x^2 + y^2 + 2xy + x + y$	$1 + 2x + 2y$	$1 + 2x + 2y$	$2 + 4x + 4y$	4
P3	$x^3 + 5y^3 + x^2y + y^2x + xy + x + y$	$1 + 3x^2 + y + y^2 + 2xy$	$1 + x + x^2 + 2xy + 15y^2$	$2 + x + 4x^2 + y + 4xy + 16y^2$	$8(x + 4y)$
P4	$x^4 + y^4 + x^2 + y^2$	$4x^3 + 2x$	$4y^3 + 2y$	$4x^3 + 4y^3 + 2x + 2y$	$12(x^2 + y^2) + 4$

В таблице 2 представлены полиномы различных степеней для компонент векторного поля, а также соответствующие им значения z-компоненты оператора ротора.

Таблица 2: Виды полиномов используемых в дальнеших расчетах с точным ротором векторного поля

	u	v	$rot\vec{V}_z$
P1	y	$-3x$	-4
P2	$y^2 - 4xy + y$	$x^2 + xy + x$	$6x - y$
P3	$-3y^3 - xy - y$	$x^3 - xy + 4x$	$3x^2 + 9y^2 + x - y + 5$

7.1 Расчет градиента скалярного поля.

Для расчета градиента скалярного поля были использованы методы Грина-Гаусса и метод Грина-Гаусса с итерациями. В результате расчетов были определены максимальные среди двух компонент градиента ошибки наибольшие для всех ячеек. При расчетах на базовой и измельченной расчетных сетках также определялась ошибка в центре расчетной области. Полученные значение представлены в таблице 3.

		Без итераций		С итерациями	
Вид сетки	P	max(Error)	Error в центре	max(Error)	Error в центре
Base	P1	1.32e-5	3.66e-6	1.31e-5	3.65e-6
	P2	4.26e-6	6e-7	1.13e-2	8e-7
	P3	1.2e-2	8e-4	1.23e-2	7.9e-4
Fine	P3	1.37e-3	8.96e-5	3.87e-3	9.19e-5
Skew	P1	1.69		2.98e-5	
	P2	0.65		1.13e-2	

Таблица 3: Ошибка вычислений для различных вариантов

Оценим порядок точности для полинома третье степени. Получим, что для метода Грина Гаусса порядок точности для приграничных и внутренних ячеек соответственно равен:

$$O(\nabla p)_b = 1.975$$

$$O(\nabla p)_{inner} = 1.993$$

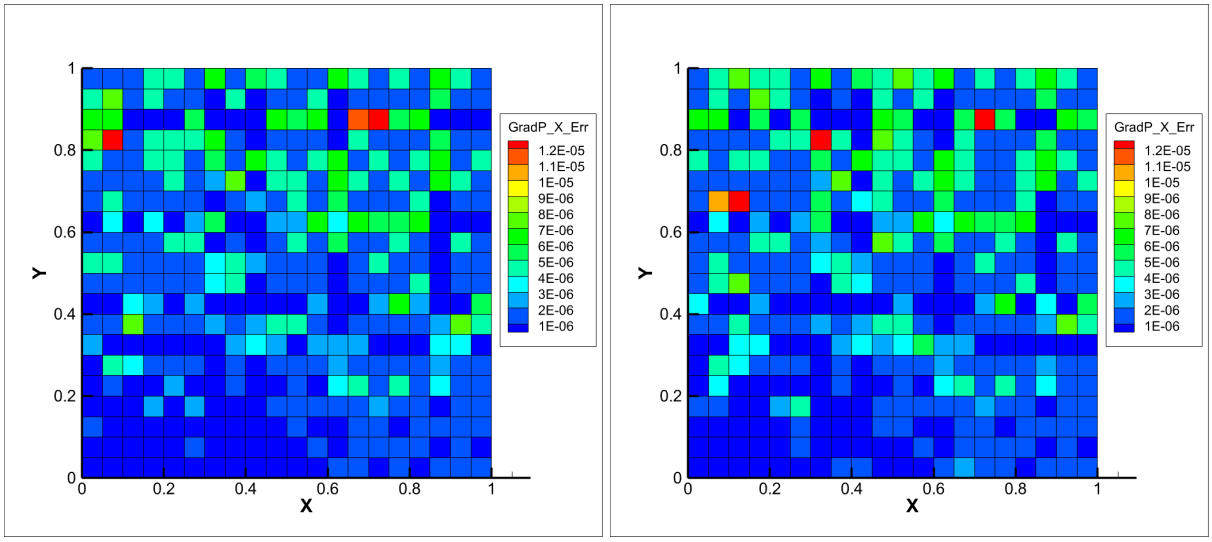
Для метода Грина Гаусса с итерациями соответственно:

$$O(\nabla p)_b = 1.053$$

$$O(\nabla p)_{inner} = 1.95$$

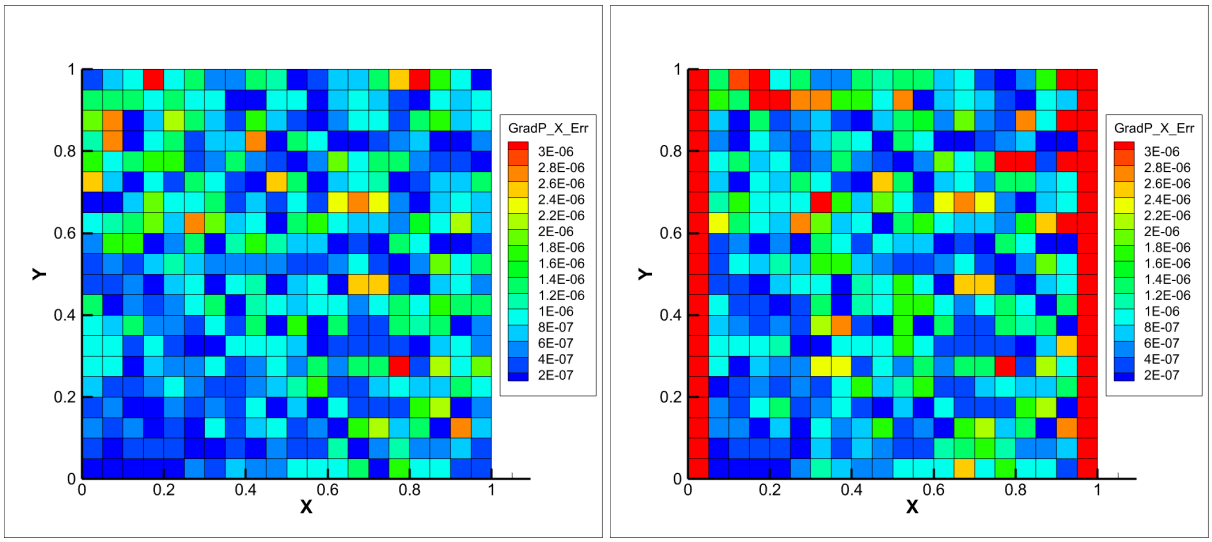
Видно, что для метода Грина-Гаусса с итерациями имеем первый порядок точности для пограничных ячеек, потому что в нем не вводили поправку второго порядка для приграничных ячеек.

На рисунках 7.2 - 7.7 представлены x-компоненты градиента скаляра при расчете на разных сетках с разным полиномом. Посколько для другой компоненты ситуация аналогичная оставим только эту.



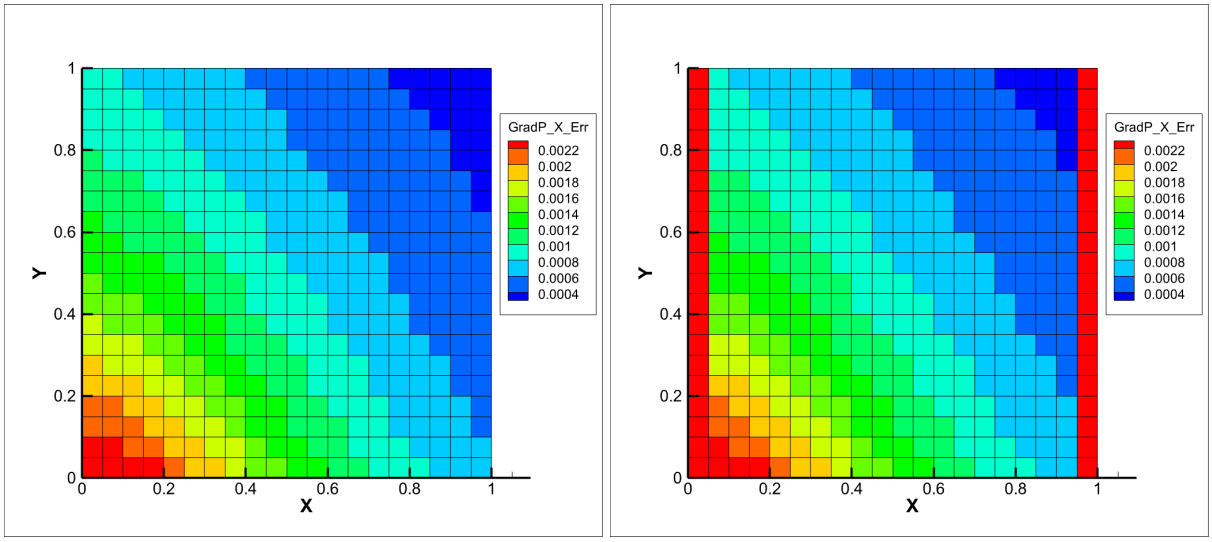
(a) GG (b) GG with iteration

Рис. 7.2: Распределение ошибки для полинома первой степени на base сетке.



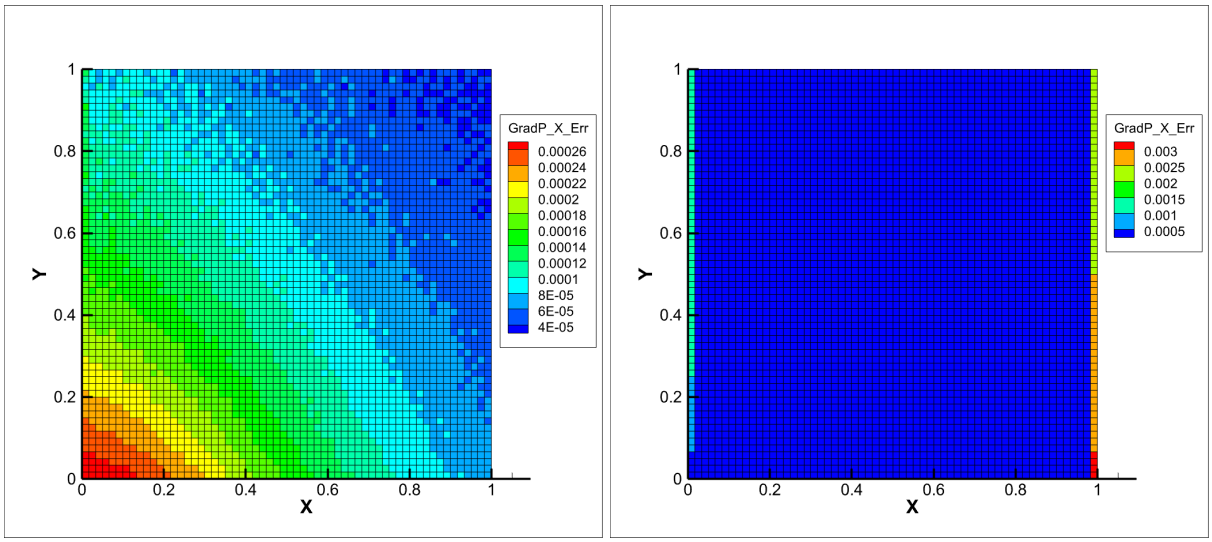
(a) GG (b) GG with iteration

Рис. 7.3: Распределение ошибки для полинома второй степени на base сетке.



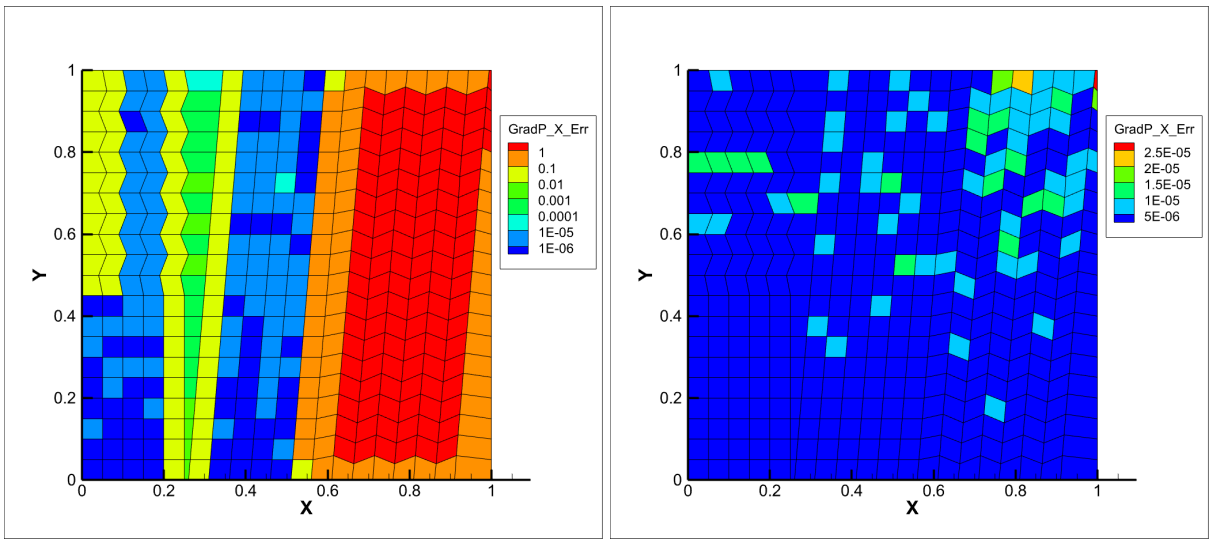
(a) GG (b) GG with iteration

Рис. 7.4: Распределение ошибки для полинома третьей степени на base сетке.



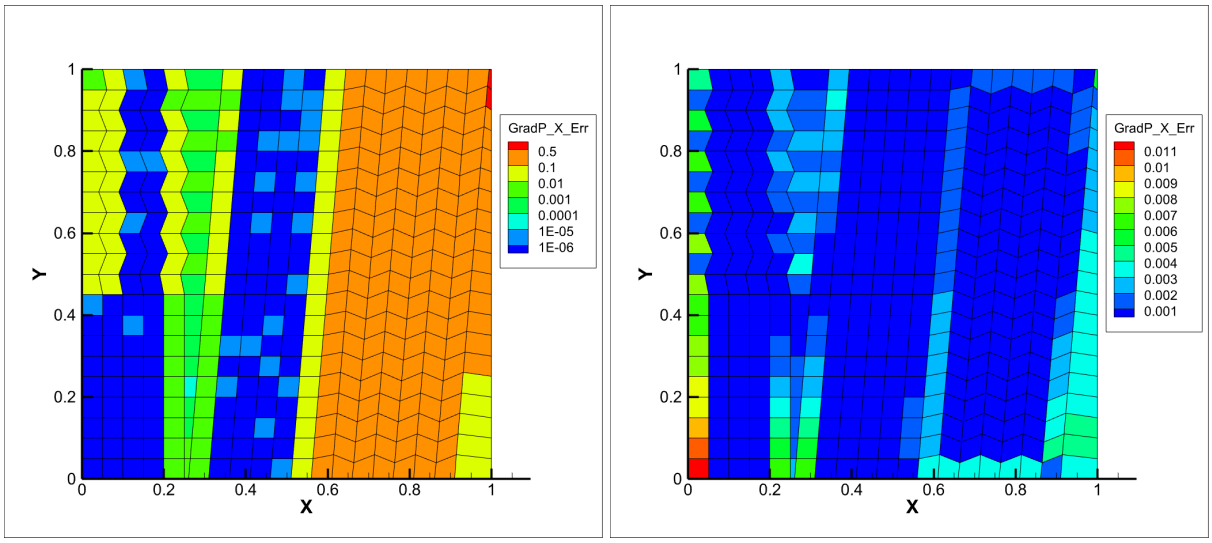
(a) GG (b) GG with iteration

Рис. 7.5: Распределение ошибки для полинома третьей степени на fine сетке.



(a) GG (b) GG with iteration

Рис. 7.6: Распределение ошибки для полинома первой степени на base сетке.



(a) GG (b) GG with iteration

Рис. 7.7: Распределение ошибки для полинома второй степени на base сетке.

Видно, что для полинома первой степени получаем одинаково точные результаты. В случае полинома второй степени во внутренней области также наблюдается машинная точность, однако для приграничных ячеек присутствуют отличия о которых говорилось ранее.

При рассмотрении ошибки градиента при расчетах на базовой и измельченной сетках видно, что по тойже причине метод Грина Гаусса с итерациями дает резкое повышение ошибки в пограничных ячейках. Поэтому получили там первый порядок точности, а для внутренних ячеек и там и там второй.

Также на примере рисунков 7.6 и 7.7 наблюдаем сильное улучшение расчета градиента при

внесении поправки и использовании итерационного метода Грина-Гаусса.

Метод Грина-Гаусса с итерациями для того, чтобы учесть скошенность, предполагает выполнение итераций. Если рассмотреть зависимость точности решения для двух полиномов на скошенной сетке от количества итераций на рисунке 7.8, то видно, что в случае P1 ошибка за 5 итераций падает на 5 порядков, а для P2 ошибка падает только на 2 порядка после 3 итераций и не изменяется. Такое поведение говорит, что поправка на скошенность не влияет на величине максимальной ошибки, вызванной нелинейностью P2, но при этом позволяет снизить ее до уровня сетки base.

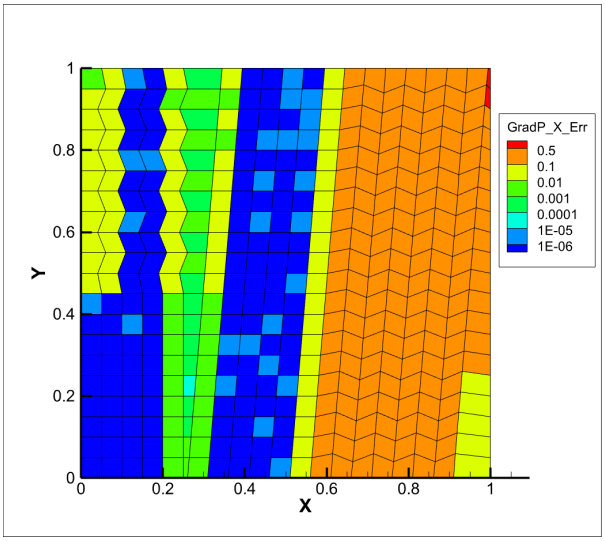


Рис. 7.8: Зависимость ошибки от количества итераций в методе GG с итерациями.

7.2 Расчет дивергенции произведения скалярной величины на вектор.

Для расчета дивергенции произведения скалярной величины на вектор были использованы три схемы: FOU, SOU и Central. Использовались полиномы 1-3 степени. Векторно поле было постоянным как и в прошлой задаче. Аналогично предыдущему пункту была построена таблица 4.

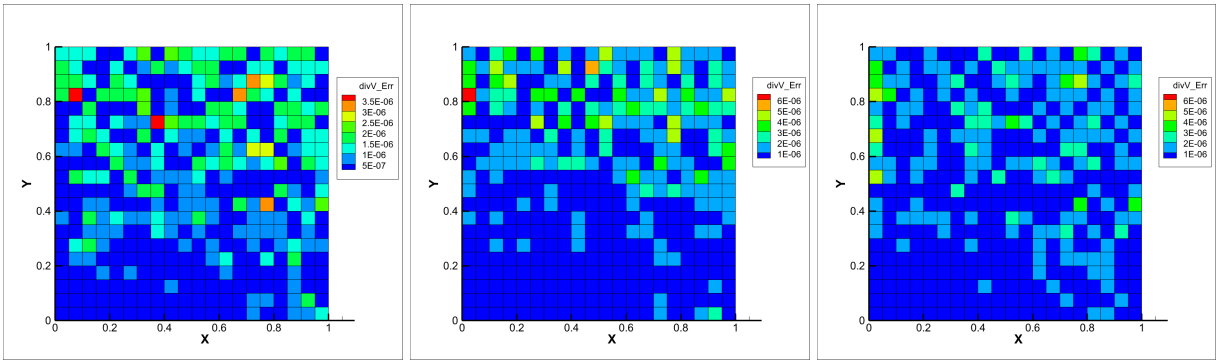
Таблица 4: Ошибка вычислений для различных вариантов схем интерполяции

		Central		FOU		SOU	
Вид сетки	P	max(Error)	Error в центре	max(Error)	Error в центре	max(Error)	Error в центре
Base	P1	3.63e-6	9.17e-7	6.2e-6	7e-7	4.78e-6	5.97e-7
	P2	1.13e-2	2.2e-7	3.84e-2	1.6e-2	2.72e-2	5.5e-7
	P3	9.3e-3	1.6e-3	6.2e-2	5.2e-2	1.87e-2	7.5e-4
Fine	P3	3.2e-3	1.7e-4	2.14e-2	1.7e-2	2.19e-3	8.4e-5
Skew	P1	0.38		1.81		0.68	

Получили следующие порядки точности для схем:

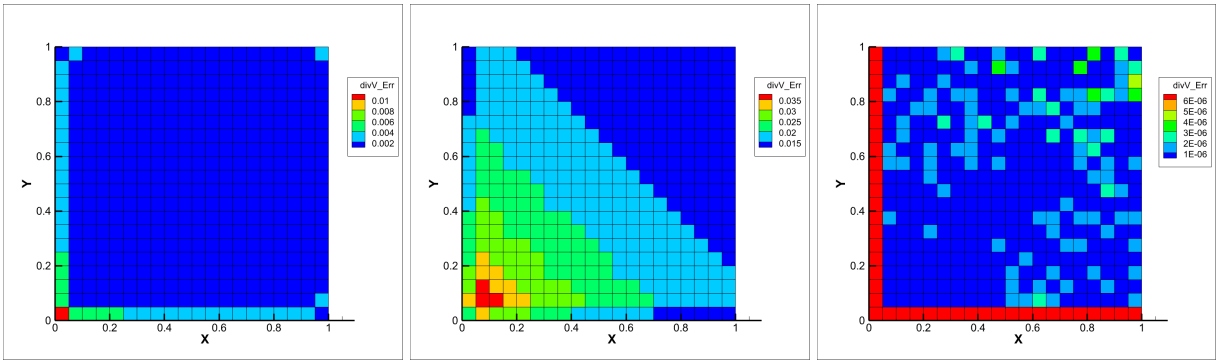
- Central: $O(div(p\vec{V}))_b = 0.97$, $O(div(p\vec{V}))_i = 2.04$
- FOU: $O(div(p\vec{V}))_b = 0.96$, $O(div(p\vec{V}))_i = 1.01$
- SOU: $O(div(p\vec{V}))_b = 1.95$, $O(div(p\vec{V}))_i = 1.99$

На рисункаъ 7.9 - 7.13 представлены распределения ошибки для дивергенции для различных схем и сеток.



(a) Central (b) FOU (c) SOU

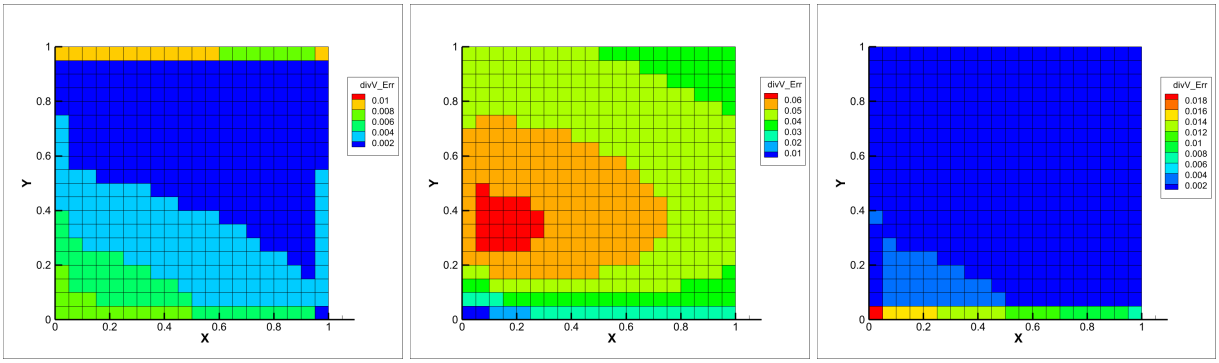
Рис. 7.9: Распределение ошибки для полинома первой степени на base сетке.



(a) Central (b) FOU (c) SOU

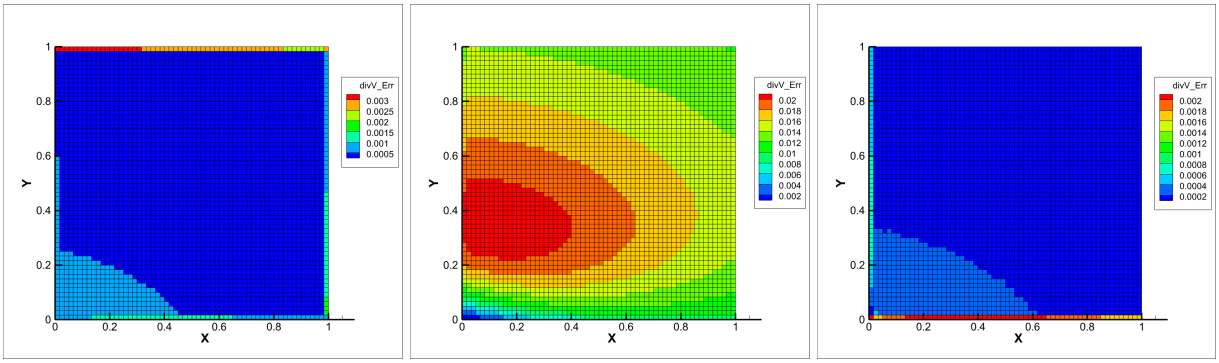
Рис. 7.10: Распределение ошибки для полинома второй степени на base сетке.

Исходя из представленных на рисунках распределений видно, что для полинома первой степени все рассмотренные схемы дают правильные с точностью до машинной погрешности результаты. В случае полинома второй степени для противопоточной схемы первого порядка возникает существенная ошибка во всей области, а для Central и SOU во внутренней области по-прежнему наблюдается машинная точность. Единственное, что схема SOU для второго ряда ячеек дает ошибку, превосходящую машинную, что обусловлено ошибкой при расчете градиента в приграничных ячейках. Рассмотрим распределения ошибки при расчетах на базовой и измельченной сетках для полинома третьей степени.



(a) Central (b) FOU (c) SOU

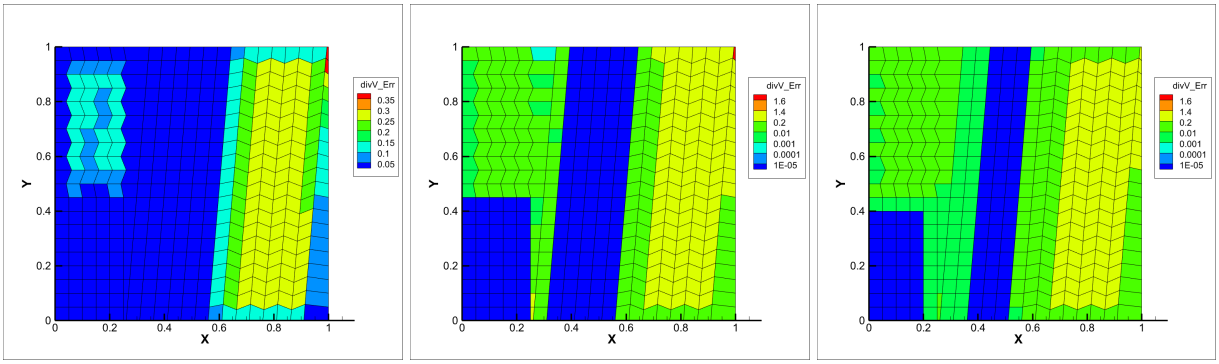
Рис. 7.11: Распределение ошибки для полинома третьей степени на base сетке.



(a) Central (b) FOU (c) SOU

Рис. 7.12: Распределение ошибки для полинома третьей степени на fine сетке.

По рисунку 7.13 можно заметить, что для схем Central, FOU и SOU (GG) наблюдается очень большая ошибка уже для полинома D1, так как в них отсутствует какая-либо поправка на скошенность.



(a) Central (b) FOU (c) SOU

Рис. 7.13: Распределение ошибки для полинома первой степени на skew сетке.

7.3 Лапласиан скалярного поля.

Рассчитаем лапласиан скалярного поля при помощи двух методов расчета градиента. В таблице 5 приведены значения относительной ошибки расчетов, как максимальная, так и в центре расчетной области.

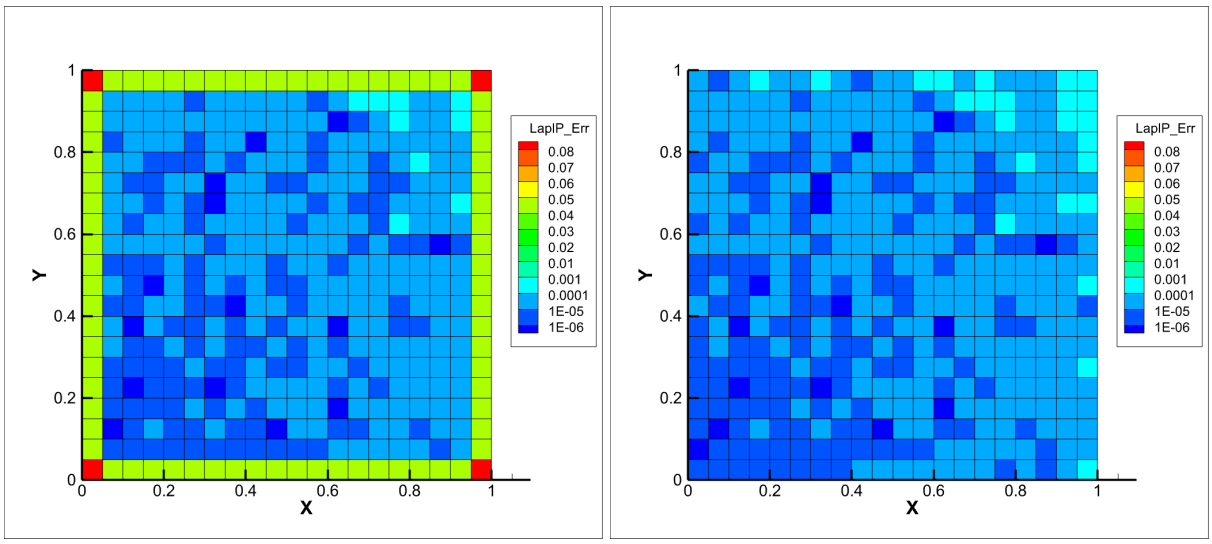
Таблица 5: Ошибка вычислений лаплсиана для различных методов и схем

		GG		GGI	
Вид сетки	P	max(Error)	Error в центре	max(Error)	Error в центре
Base	P2	8.32e-2	3.15e-5	2.78e-4	3.11e-5
	P3	0.29	5.92e-6	0.3	5.43e-6
	P4	0.1	9.6e-4	1.42e-2	9.9e-4
Fine	P4	8.9e-2	1.2e-4	4.58e-3	1.19e-4
Skew	P2	17.4		8.37	
	P3	38.7		16.7	

Определим порядок точности для приграничной и внутренней ячеек:

- GG: $O(\Delta p)_b = 0.1$, $O(\Delta p)_i = 1.89$
- GGI: $O(\Delta p)_b = 1.02$, $O(\Delta p)_i = 1.92$

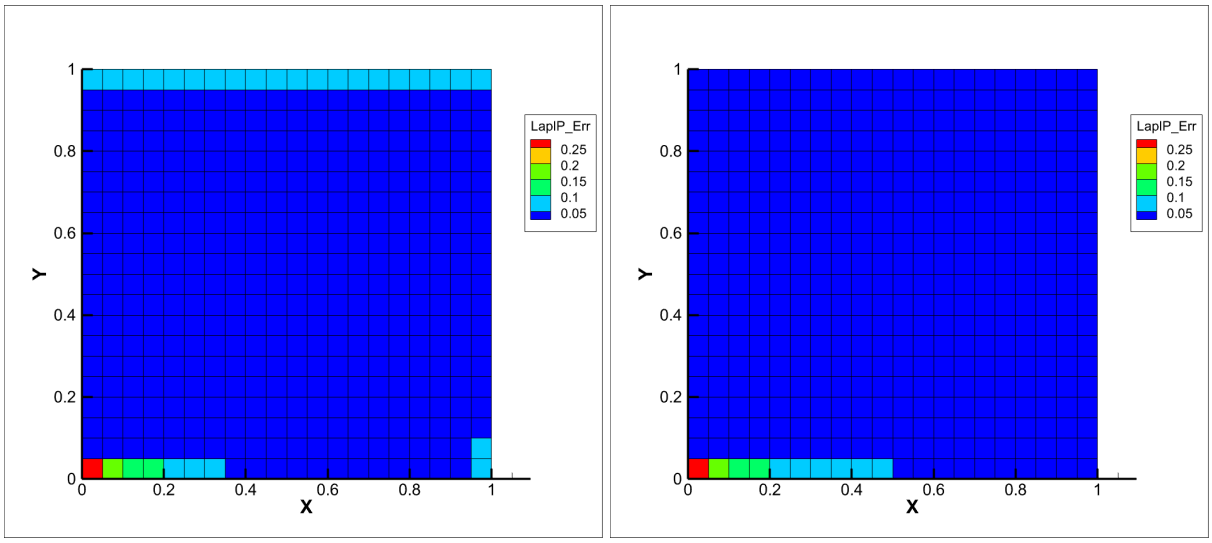
На рисунка 7.14 - 7.19 представлены распределение ошибки в расчетной области для разных методов и видов полиномов.



(a) GG (b) GG with iteration

Рис. 7.14: Распределение ошибки вычисления лапласиана для полинома второй степени на base сетке.

По рисунку 7.14 видно, что для метода GG в приграничных ячейках наблюдается сильное увеличение погрешности, связанное со специальным способом расчета градиента на границе. Во внутренней области для GG наблюдается хаотическое поведение ошибки, однако её значение превышает машинную точность. В случае метода GGI ошибка распределена хаотично уже во всей области.

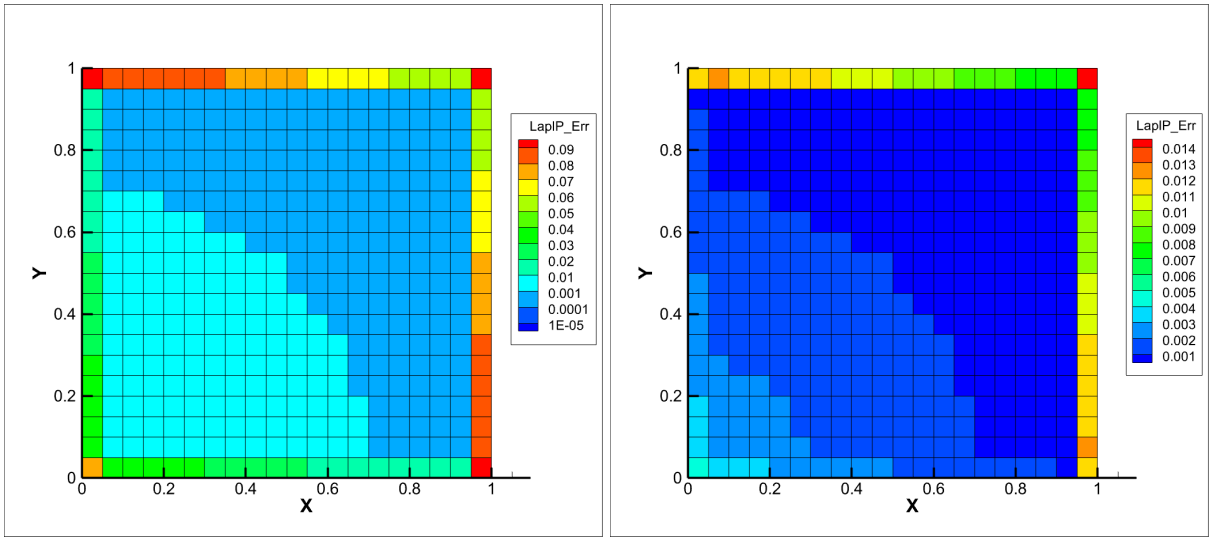


(a) GG (b) GG with iteration

Рис. 7.15: Распределение ошибки вычисления лапласиана для полинома третьей степени на base сетке.

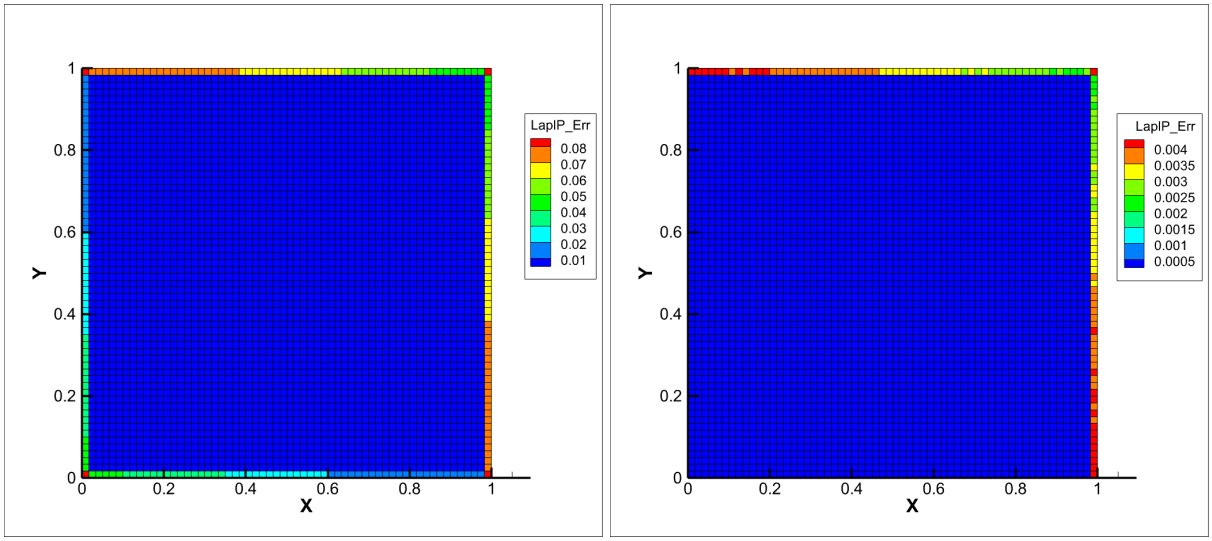
Из рисунков 7.16 и 7.17 наблюдаем аналогичное поведение как и в предыдущих расчетах при измельчении сетки, что ошибка в центре сильно падает, и остается только на границах

расчетной облсти.



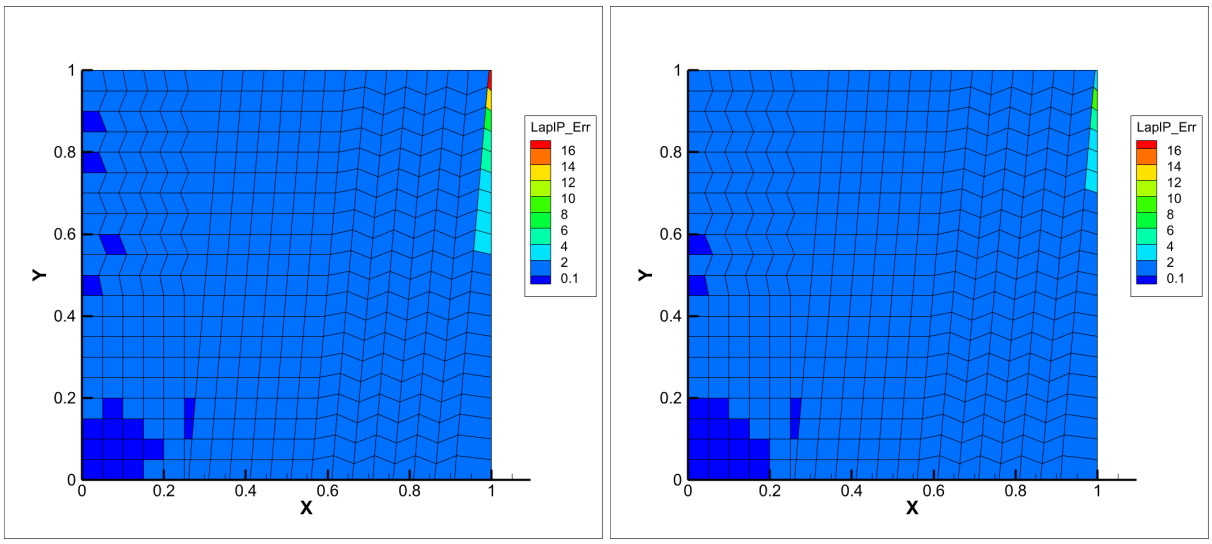
(a) GG (b) GG with iteration

Рис. 7.16: Распределение ошибки вычисления лапласиана для полинома четвертой степени на base сетке.



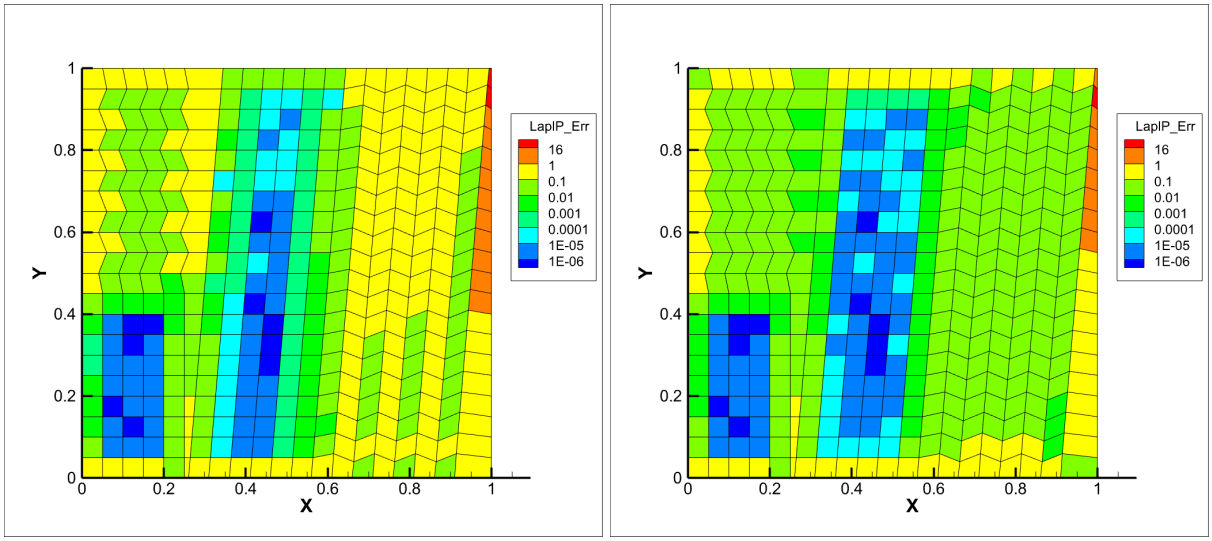
(a) GG (b) GG with iteration

Рис. 7.17: Распределение ошибки вычисления лапласиана для полинома второй степени на fine сетке.



(a) GG (b) GG with iteration

Рис. 7.18: Распределение ошибки вычисления лапласиана для полинома второй степени на skew сетке.



(a) GG (b) GG with iteration

Рис. 7.19: Распределение ошибки вычисления лапласиана для полинома третьей степени на skew сетке.

Для скошенных сеток наблюдаем огромные ошибки, которые уменьшаются при использовании итерационного метода.

7.4 Расчет ротора векторного поля.

Для расчета ротора векторного поля использовались полиномы третьей степени. Аналогично предыдущим трем пунктам. В таблице 6 представлены значения ошибки ротора как

максимальной так и в центре расчетной области.

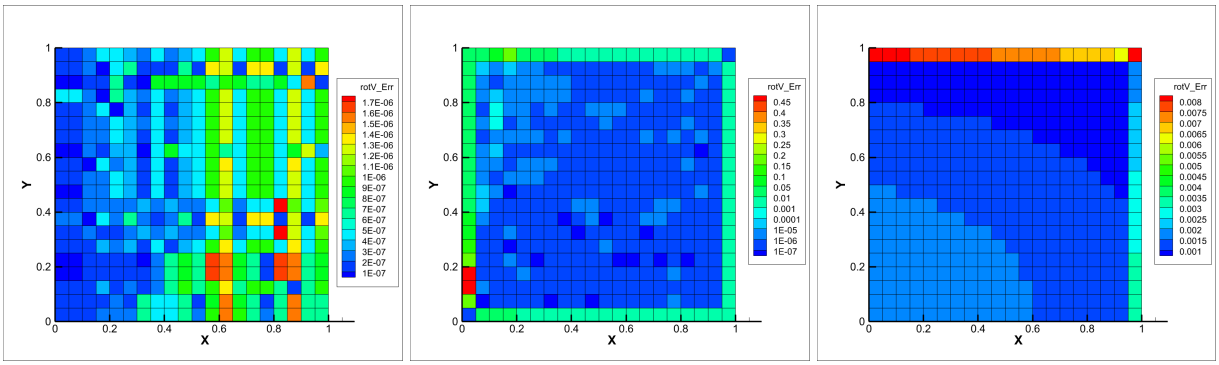
Таблица 6: Ошибка вычислений ротора для различных сеток и полиномов

	Base		Fine		Skew	
P	max(Error)	Error в центре	max(Error)	Error в центре	max(Error)	Error в центре
P1	1.78e-6	4e-7	7.39e-6	1e-7	1e-1	
P2	0.5	5.25e-7	0.5	6e-7	2.13	
P3	8.5e-3	1.2e-3	2.9e-3	1.3e-4	7.69e-2	

Оценим порядок точности для приграничной и внутренней ячеек соответственно:

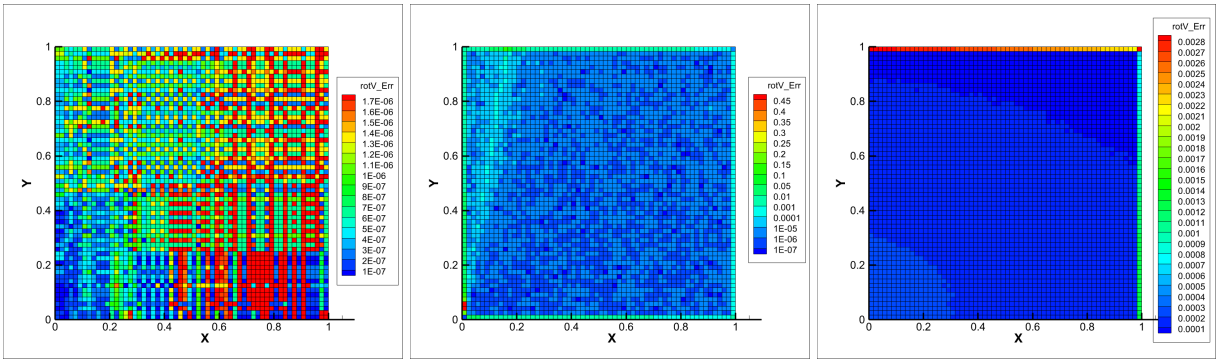
- $O(\Delta p)_b = 0.978$, $O(\Delta p)_i = 2.02$

На рисунка 7.20 - 7.22 представлены ошибки поля ротора. Можно заметить, что для полинома второй степени максимальная ошибка сильно выросла, это может быть связано с самим полиномом, а не сколько с методом. Для оплиномов первой степени для исходной и измельченной сеток наблюдаем машинную точность. Также и для полинома второй степени но внутри области. Для полинома третьей степени ошибка внутри стала больше.



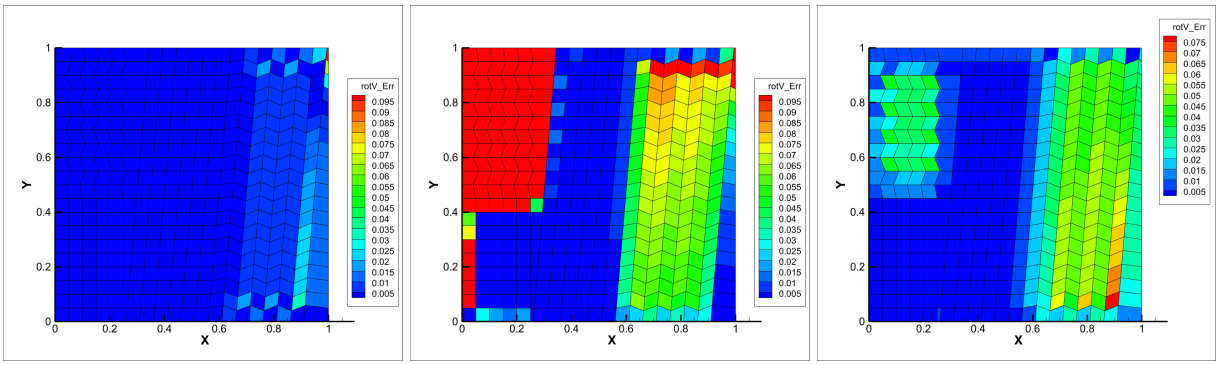
(a) P1 (b) P2 (c) P3

Рис. 7.20: Распределение ошибки ротора на base сетке.



(a) P1 (b) P2 (c) P3

Рис. 7.21: Распределение ошибки ротора на fine сетке.



(a) P1 (b) P2 (c) P3

Рис. 7.22: Распределение ошибки ротора на skew сетке.

Поскольку при расчете ротора не использовались никакие поправки на скошенность, наблюдается существенная погрешность для большинства ячеек, а для полинома второй степени погрешность больше, чем для первой степени.

8 Перенос температуры в каверне.

8.1 Постановка задачи.

В данной работе проводится численное решение уравнения конвективно-диффузионного переноса температуры в заданном поле скоростей по МКО, применительно к задаче о течении в каверне с движущейся крышкой и с разнонагретыми стенками. Каверна имеет две изотермические стенки левая - горячая, правая - холодная, температура холодной стенки T_cold равна 1.0, температура горячей стенки T_hot равна 2.0. Две другие стенки каверны – адиабатические. Задача решается в безразмерной постановке, масштабы для обезразмеривания задачи: H – высота каверны, $U_w = 1$ – скорость движения крышки (верхней стенки) каверны. Течение и теплообмен в каверне определяются следующими безразмерными параметрами: K = L/H = 0.5 – геометрический фактор

(отношение ширины L каверны к ее высоте H), Re=90 – число Рейнольдса, Pr=1 – число Прандтля. Расчетная область и сетка представлены на рисунке 8.1.

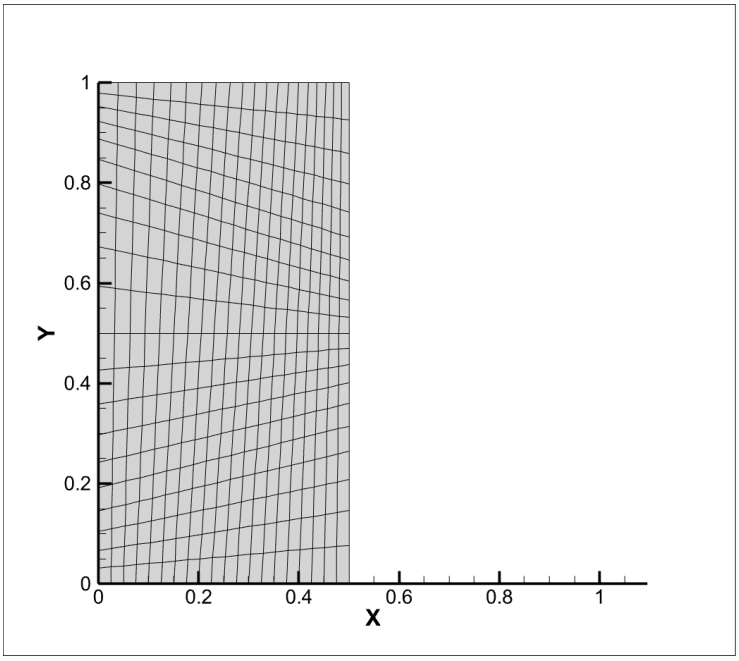


Рис. 8.1: Вид расчетной области.

В данном случае используется скошенная регулярная сетка, состоящая из прямоугольников. Граничные условия представлены на рисунке 8.2.

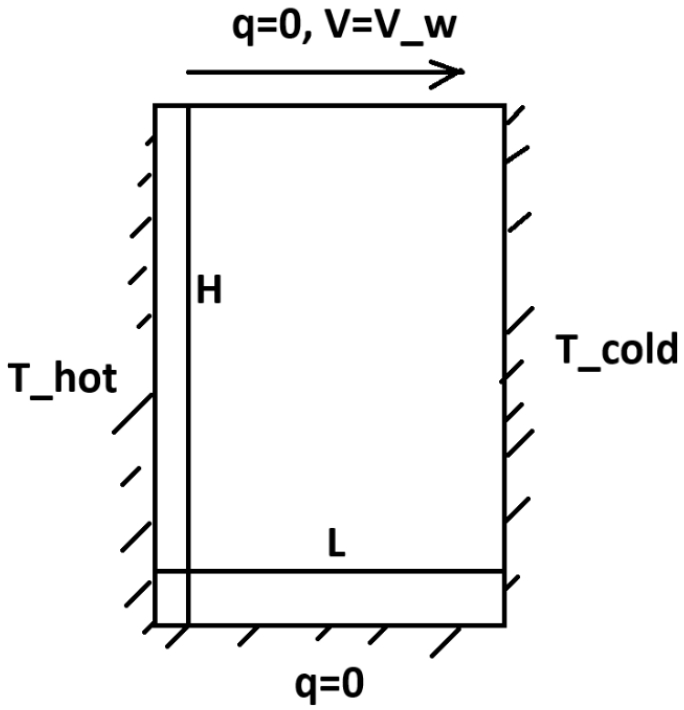


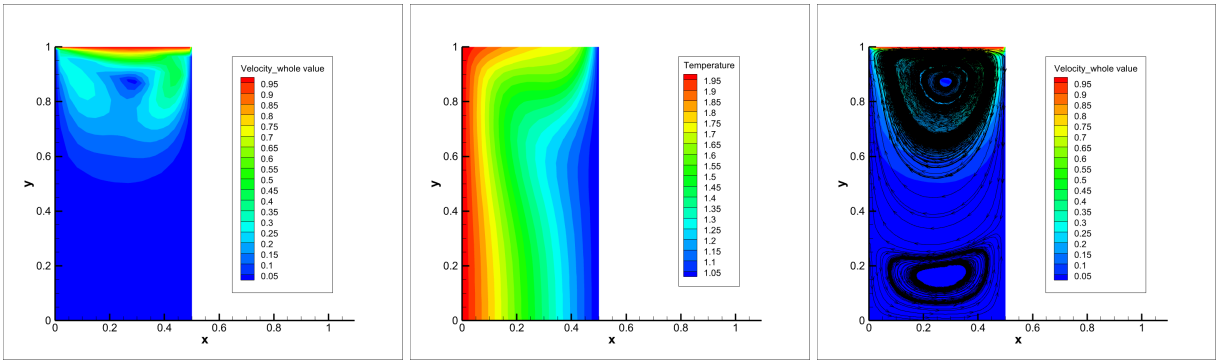
Рис. 8.2: Граничные условия.

Для нахождения численного решения был написан дополнительный модуль под названием B_CalcResT. Он представляет из себя возможность проведения расчетов конвективных и диффузионных потоков на гранях вследствии чего находятся невязки и значения шагов по псевдовремени. Также он учитывается нулевой тепловой поток на нижней и верхней стенке. Для достижения эффекта адиабатической стенки на каждой итерации значение температуры в фиктивных ячейках меняется таким образом, чтобы на грани градиент температуры вдоль нормали к грани был равен нулю.

Также в работе необходимо выполнить расчет в ПО Flos и сравнить полученные результаты с вычисленными.

8.2 Результаты полученны с помощью программы FLOS.

Сетка строилась в программе FLOS и имеет размеры 21x21 ячейки. На рисунке 8.3 изображено поле модуля скорости, линии тока, а также поле температуры.



(a) Поле модуля скорости. (b) Поле температуры. (c) Линии тока.

Рис. 8.3: Поля распределений величин.

Видно, что большие скорости достигаются в верхней части каверны в частности в правом верхнем углу. Как видно, так как верхняя крышка каверны находится в движении, образуется круговое движение жидкости по часовой стрелке в верхней половине каверны, что подтверждает картина распределения линий тока, где можно заметить, что в верхней половине каверны образуется один большой вихрь, а в нижней появляется еще один но менее интенсивный с меньшими скоростями.

В свою очередь в можно увидеть, что наличие конвективного переноса приводит к искажению изолиний температуры. Наибольшее искажение изолиний наблюдается в области основного вихря. При этом на нижней стенке изолинии практически вертикальные, что согласуется с поставленным условием адиабатичности на ней. На верхней стенке также поставлено условие адиабатичности, однако из-за движения данной границы изолинии температуры искажаются в сторону движения стенки.

В таблице 7 представлены входные параметры, задаваемые для получения решения. Значения чисел Куранта и фон Неймана задавались таким образом, чтобы добиться максимальной устойчивости.

Таблица 7: Входные параметры

Входной параметр	Значение
Метод расчета ∇T	GGI
Схема расчета конв. потоков	Central UGB
V_s, м/с	1
L_s, м	0.5
Re	90
Pr	1
CFL	1
VNM	0.03
Количество итераций (niter)	15000

Как говорилось ранее, наличие конвективных поток приводит к искажению изолиний температуры. Для наглядности в ходе исследования с помощью вычислительного кода был проведен расчет в случае отсутствия конвективных потоков на расчетной сетке, полученной в программе Flos. Поле температуры в случае такой постановки задачи представлено на рисунке 8.4.

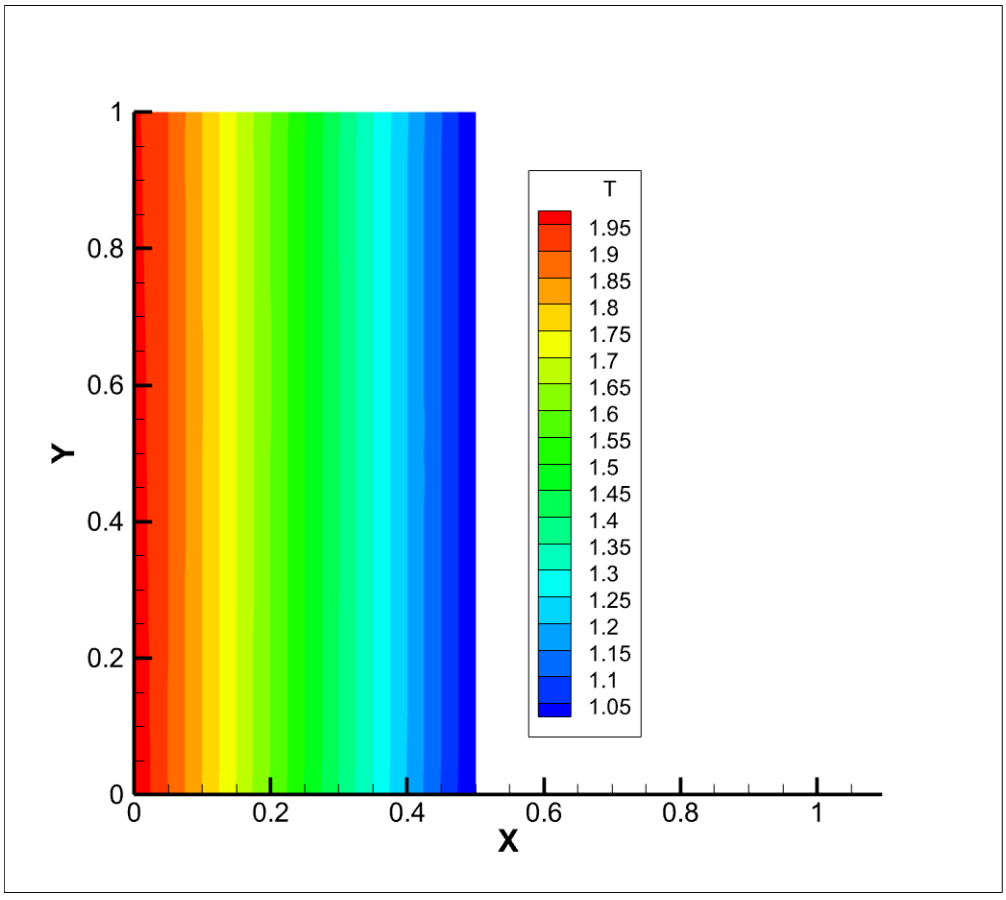


Рис. 8.4: Поле температуры в отсутствие конвекции.

Видно, что в данном случае температура меняется линейно от значения 1 до 2.

При решении задачи в случае наличия конвективных потоков были получены графики истории сходимости, представленные на рисунке 8.5.

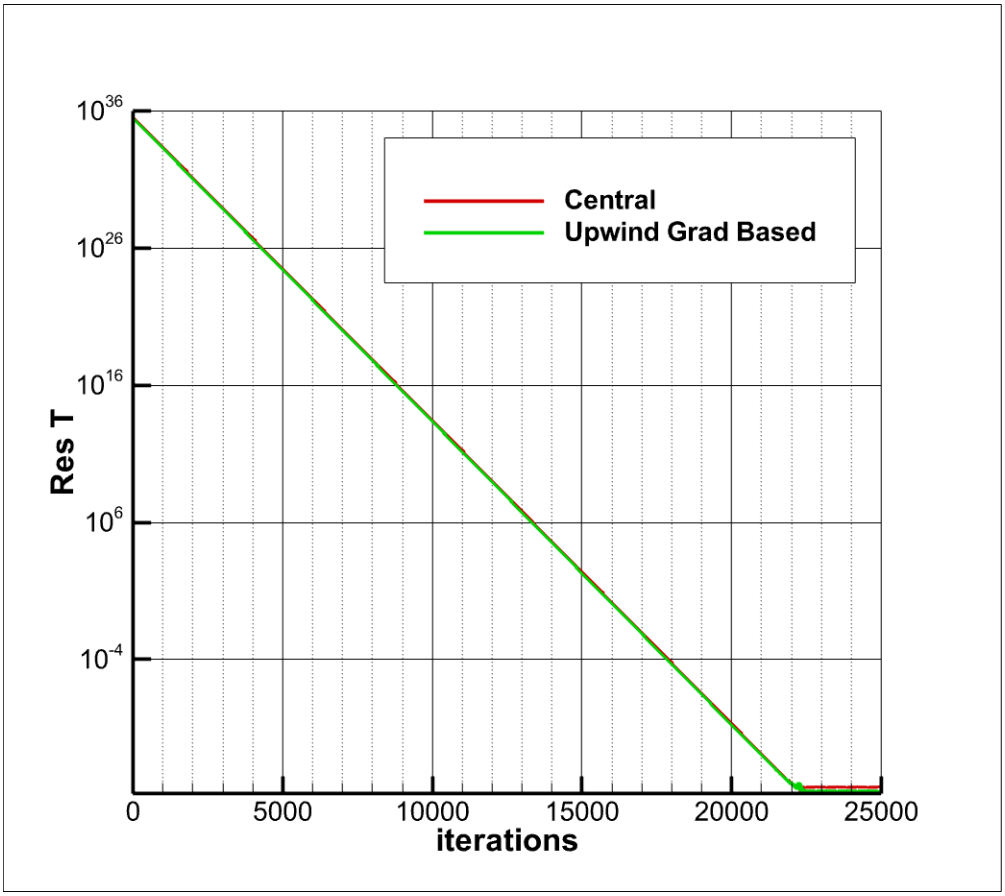
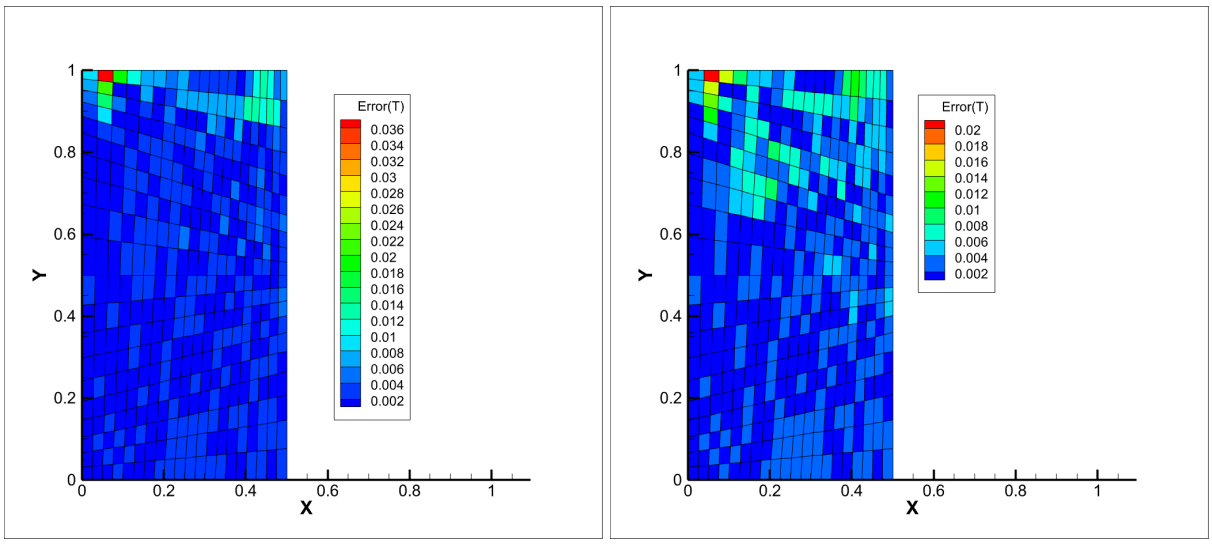


Рис. 8.5: График невязок.

Для обеих схем были определены максимальные значеня ошибки:

- Central : $Error(T)_{max} = 3.78E - 02$
- UGB: $Error(T)_{max} = 4.26E - 02$

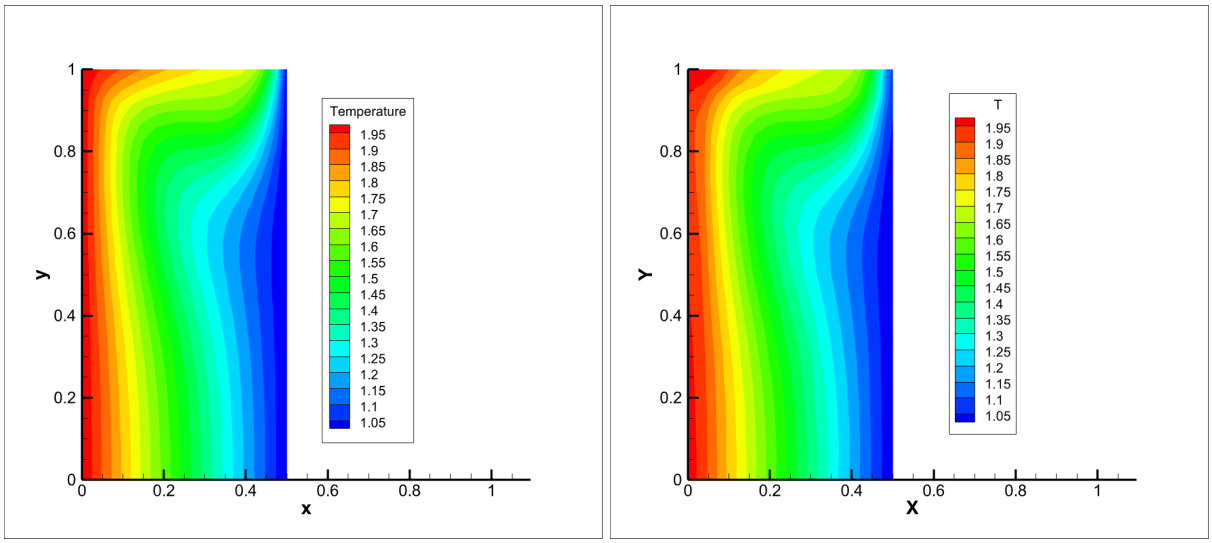
На рисунке 8.6 представлены распределения ошибки при исользовании различных схем.



(a) Central (b) UGB

Рис. 8.6: Распределение ошибки температуры.

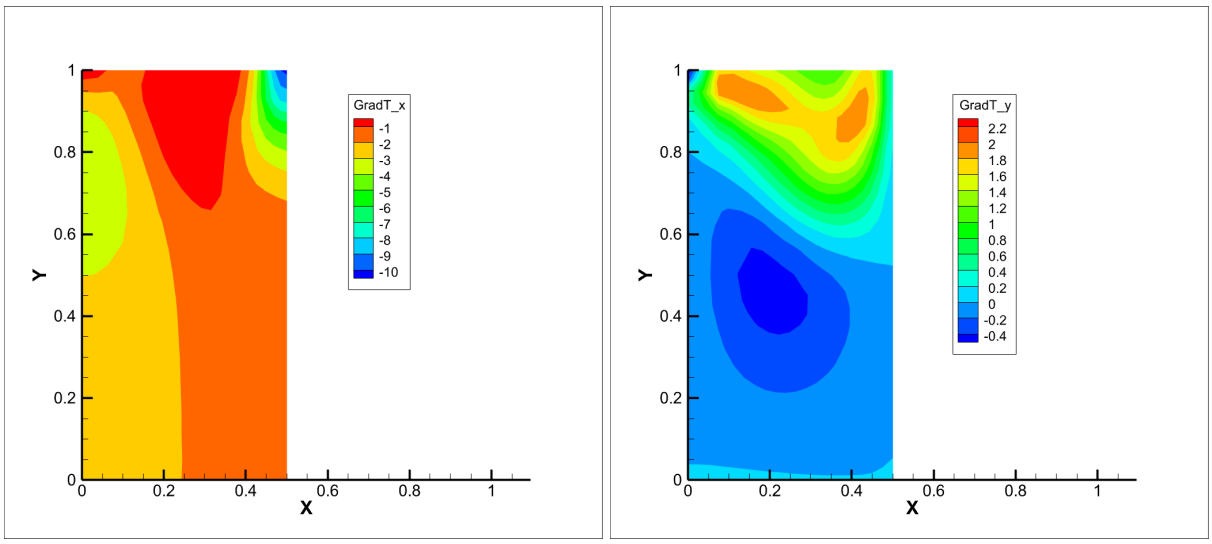
Видно, что для схемы Upwind Gradine Based ошибка чуть больше чем для центрально-разностной схемы, однако отличия качественные в температурном поле не велики, потому будут представлены пол только для Central схемы.



(a) FLOS (b) Fortran

Рис. 8.7: Поле температуры.

Видно по рисунку 8.7 что поля совподают, а отклонения малы в существенной части расчетной области.



(a) x - компонента (b) y - компонента

Рис. 8.8: Поля компонент градиента температуры.

Из рисунка 8.8 делаем вывод о том, что там где скорость максимально тепловой поток тоже максимален в тоже самое время поперечный тепловой поток изменяется не так существенно. Напомним что тепловой поток пропорционален и обратен по знаку градиенту температуры.

8.3 Результаты тестирования эффективности параллелилзации.

В данном пункте проводится исследование эффективности параллелизации, реализованной с помощью технологии OpenMP. Запуск программы проводился при схеме расчета конвективных слагаемых UGB. Для проведения исследования эффективности параллелизации часть программы, задействованная при решении уравнения конвективно-диффузионного переноса температуры, была параллелизована средствами OpenMP. Для этого была добавлена директива OMP DO перед вложенным циклом по всем ячейкам в подпрограммах для расчета градиента температуры и её невязки, которая является директивой распараллеливания циклов, а именно она осуществляет автоматическое распределение итераций циклу между нитями. Для оценки эффективности параллелизации задавалось различное количество нитей от 1 до 4 с помощью переменной окружения OMP_NUM_THREADS, а использование функции OMP_GET_WTIME позволяло определять время работы параллельной области программы.

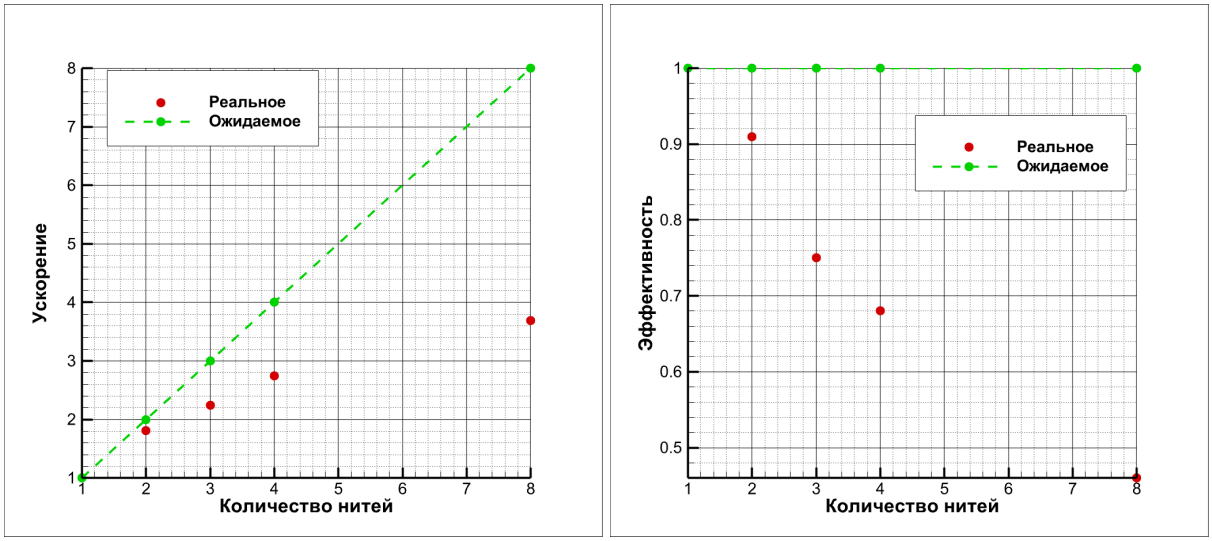
В таблице **??** приведена зависимость времени работы от количества используемых нитей и расчет ускорения и эффективности по следующим формулам:

$$S_p = T_1/T_p; \qquad E_p = S_p/p$$

, где p - число нитей, а T_1 - время последоваельного выполнения программы. На рисунке 8.9 изображены графики зависимости ускорения и эффективности от количества нитей.

Таблица 8: Зависимость времени, ускорения и эффективности от количества нитей.

p	T, c	S_p	E_p
1	75.8	1	1
2	41.4	1.81	0.91
3	33.4	2.24	0.75
4	27.4	2.74	0.68
8	20.3	3.69	0.46



(a) Ускорение (b) Эффективность

Рис. 8.9: Поля компонент градиента температуры.

Исходя из значений, приведенных в таблице 8, и графиков видно, что наблюдается довольно сильное отклонение от идеальных зависимостей, как для ускорения, так и для эффективности, уже при p = 3 эффективность параллелизации отклонилась примерно на 20%, и с дальнейшим увеличением числа нитей отклоняется все сильнее. На графике эффективности можно наблюдать аналогичные результаты: при трех ядрах E_p падает практически на 25%, после чего эффективность также продолжает падать. Таким образом, в случае данной постановки задачи распараллеливание неэффективно.

Проверим как влияют встроенные опции оптимизации в компиляторе gfortran. Расчет производились на 4 нитях.

- O0: 27.62 c
- O1: 12.86 c
- O2: 12.56 c

Получим, что использование любой встроенной опции оптимизации компилятара автоматически ускоряет работу программу почти в два раза.

9 Выводы.

- Проведено исследование численных методов для расчёта дифференциальных операторов на двумерных сетках с использованием метода конечных объёмов. Особое внимание уделено работе с различными типами структурированных сеток, включая скошенные. Для вычислений использовалась программа, реализованная на языке Fortran.

- В рамках исследования были изучены два метода расчёта градиента скалярного поля: Грина-Гаусса (GG) и его модификация GGI. Метод GG продемонстрировал второй порядок точности как внутри расчётной области, так и на границах, благодаря специальным алгоритмам обработки. Однако метод GGI, сохраняя второй порядок точности внутри области, обладает первым порядком точности на границах. Его основное преимущество заключается в возможности учитывать скошенность сеток, что позволило значительно снизить ошибки на таких геометриях.

- Расчёты дивергенции произведения скалярной функции и векторного поля выполнены с использованием схем Central, FOU и SOU. Схема Central показала второй порядок точности внутри расчётной области и первый на её границах. Схема FOU обеспечила второй порядок точности во всех ячейках. Схема SOU, аналогичная Central по точности, проявила себя лучше на скошенных сетках благодаря сочетанию с методом GGI.

- Рассчитан лапласиан скалярного поля, для которого градиент вычислялся методами GG и GGI. Лапласиан показал второй порядок точности во внутренних ячейках независимо от метода расчёта градиента. Однако в приграничных ячейках метод GG продемонстрировал точность ниже первого порядка, тогда как GGI обеспечил первый порядок и снизил ошибки на скошенных сетках.

- Также был рассчитан ротор векторного поля методом линейной интерполяции. Результаты показали второй порядок точности во внутренних ячейках и первый порядок на границах.

- Решено уравнение конвективно-диффузионного переноса температуры в задаче течения в каверне с движущейся крышкой и разнонагретыми стенками. Конвективные члены уравнения рассчитывались схемами Central и Upwind Gradient Based. Движение крышки вызвало формирование крупного вихря в верхней части каверны, что привело к искажению изолиний температуры. Схема UGB продемонстрировала менее точные результаты по сравнению с Central.

- Кроме того, проведено исследование эффективности параллелизации кода с использованием технологии OpenMP. Установлено, что ускорение при увеличении числа ядер отстаёт от линейного: при использовании двух ядер эффективность распараллеливания снизилась на 30%, что указывает на ограничение возможностей параллельного выполнения для данной задачи.
- Наконец, проведён анализ времени выполнения программы при использовании различных флагов оптимизации. Оптимизации -O1 и -O2 позволили сократить время работы программы почти в два раза.

10 Приложение.

10.1 Приложение A

Здесь находится код программы и отчет. `https://github.com/MrDionisio/FVM_CFD`