Оглавление

1. Введение

В самых разных сферах науки и технологий неотъемлемую часть занимают дифференциальные уравнения в частных производных. Анализ результатов, осуществленных на основе дифференциальных уравнении, обеспечивается при помощи приближенных численных методов решения. Одним из самых популярных дифференциальных уравнений является уравнение Пуассона (эллиптическое уравнение).

Для решения этих эллиптических уравнений в случае малых измерений используются приближенные численные методы, которые позволяют преобразовать дифференциальное уравнение и их системы в системы алгебраических уравнений. Важным фактором при вычислении является внушительный объем вычисляемых данных. Здесь в помощь приходит использование высокопроизводительных вычислительных систем. Область проблем численного решения дифференциальных уравнений в частных производных является сферой интенсивных исследований.

Для ускорения сходимости итераций и более быстрого уменьшения низкочастотной составляющей невязки применяется многосеточный итерационный метод, предложенный Р.П. Федоренко. Суть его заключена в следующем. После нескольких итераций на подробной сетке проводится ограничение решения на сетку менее подробную, для которой число обусловленности значительно меньше. После нескольких итераций, позволяющих погасить высокочастотные компоненты невязки, проводится ограничение на менее подробную сетку и так далее. Затем следует обратный ход итераций – переход от последовательности грубых сеток ко все более и более подробным.

Известно, что главную трудность при решении систем сеточных уравнений представляет собой итерационное решение СЛАУ большой размерности – такая система плохо обусловлена. Как следствие, скорость сходимости итераций медленная.

Чтобы ускорить сходимость предлагается разделить задачу на «четную» и «нечетную» части (подробнее об этом в п.2). Получаем две независимых задачи на сетках с меньшим числом узлов. Стало быть, для

каждой из этих систем оператор получается лучше обусловленным, скорость сходимости становится выше.

В отличие от классического многосеточного метода, на сетках с меньшим числом узлов получается не одна задача, а две. Но при обратном ходе алгоритма нам не потребуется строить оператор интерполяции – достаточно нужную функцию просто продолжить на большую область четным или нечетным образом соответственно.

Основной целью и задачей в данной работе является разработка программного обеспечения, которое позволит получить решение уравнения Пуассона, а также сравнить полученные результаты с использованием технологии Open Multi-Processing (OpenMP).

2. Постановка задачи

Рассмотрим постановку эллиптической задачи с граничными условиями первого рода:

Пусть уравнение Пуассона

$$\Delta u = f(x, y) \tag{2.0.1}$$

решается в области:

$$\Omega : \{-1 \leq x \leq 1, -1 \leq y \leq 1\} \tag{2.0.2}$$

с условиями:

$$u(-1, y) = f_1(y);\ u(1, y) = f_2(y);\ u(x, -1) = f_3(x);\ u(x, 1) = f_4(x). \tag{2.0.3}$$

Произведем замену x на –x в (2.0.1) и поставим еще одну задачу

$$\Delta v = g(x, y) = f(-x, y) \tag{2.0.4}$$

в той же области с условиями:

$$v(-1, y) = f_2(y);\ v(1, y) = f_1(y);\ v(x, -1) = f_3(-x);\ v(x, 1) = f_4(-x). \tag{2.0.5}$$

Тогда, вполне очевидно, выполнено верное равенство:

$$v(x, y) = u(-x, y). \tag{2.0.6}$$

Введем две новые функции:

$$w_0 = \frac{1}{2}(u + v), \quad w_1 = \frac{1}{2}(u - v). \tag{2.0.7}$$

К этим двум функциям применим оператор Лапласа и получим :

$$\Delta w_0 = \frac{1}{2}\Delta(u + v) = \frac{1}{2}(\Delta u + \Delta v) = \frac{f(x, y) + f(-x, y)}{2} = \varphi(x, y). \tag{2.0.8}$$

На границах области для функции выполняются условия:

$$\begin{aligned} &w_0(-1, y) = \tfrac{1}{2}\left(f_1(y) + f_2(y)\right) = \varphi_0(y) \\ &w_0(1, y) = \tfrac{1}{2}\left(f_1(y) + f_3(y)\right) = \varphi_0(y) \\ &w_0(x, -1) = \tfrac{1}{2}\left(f_3(x) + f_3(-x)\right) = \varphi_3(x) \\ &w_0(x, 1) = \tfrac{1}{2}\left(f_4(x) + f_4(-x)\right) = \varphi_4(x). \end{aligned} \tag{2.0.9}$$

В силу того, что функция четная, выполнено:

$$\frac{\partial w_0(0,y)}{\partial x} = 0. \tag{2.0.10}$$

Аналогично,

$$\Delta w_1 = \frac{1}{2}\Delta(u-v) = \frac{1}{2}(\Delta u - \Delta v) = \frac{f(x,y) - f(-x,y)}{2} = \psi(x,y). \tag{2.0.11}$$

На границах области для функции выполняются условия:

$$\begin{array}{l} w_1(-1,y) = \frac{1}{2}\left(f_1(y) - f_2(y)\right) = \psi_{-1}(y) \\ w_1(1,y) = \frac{1}{2}\left(-f_1(y) + f_2(y)\right) = -\psi_{-1}(y) = \psi_1(y) \\ w_1(x,-1) = \frac{1}{2}\left(f_3(x) - f_3(-x)\right) = \psi_3(x) \\ w_1(x,1) = \frac{1}{2}\left(f_4(x) - f_4(-x)\right) = \psi_4(x). \end{array} \tag{2.0.12}$$

В силу того, что функция нечетная, выполнено:

$$w_1(0,y) = 0. \tag{2.0.13}$$

Каждую из этих двух функций можно искать на той же сетке (с той же точностью , определяемой аппроксимацией, или с точностью $\varepsilon/2$), что и исходную, но в области

$$\Omega_1 : \{0 \le x \le 1, -1 \le y \le 1\}. \tag{2.0.14}$$

Заметим, что каждую из получившихся задач можно разбить еще на две подзадачи в области

$$\Omega_2 : \{0 \le x \le 1, 0 \le y \le 1\}. \tag{2.0.15}$$

На этом этапе используется уже разделение на четную и нечетную части по направлению y.

Пусть уравнение Пуассона

$$\Delta w_0 = \varphi(x,y) \tag{2.0.16}$$

решается в области:

$$\Omega_{1/2} : \{0 \le x \le 1, -1 \le y \le 1\} \tag{2.0.17}$$

с условиями:

$$\frac{\partial}{\partial x}w_0(0,y)=0;\ w_0(1,y)=\varphi_1(y);\ w_0(x,-1)=\varphi_3(x);\ w_0(x,1)=\varphi_4(x). \tag{2.0.18}$$

Данная задача получена из Задачи (2.1.16) заменой у на –у. Снова введем в рассмотрение две новые функции:

$$w_{00}=\frac{1}{2}\left(w_0+w_{0,-1}\right),\quad w_{01}=\frac{1}{2}\left(w_0-w_{0,-1}.\right) \tag{2.0.19}$$

Так как первая из этих функция четная по у, то

$$\frac{\partial w_{00}}{\partial y}(x,0)=0. \tag{2.0.20}$$

Для второй функции верно

$$w_{01}(x,0)=0. \tag{2.0.21}$$

Для каждой из этих функций может быть поставлена задача в области, например,

$$\Omega_{1/4}:\{0\le x\le 1,0\le y\le 1\}. \tag{2.0.22}$$

Аналогично задача для функции ω может быть разделена на задачи для двух функций в области (2.1.22)

$$w_{01}=\frac{1}{2}\left(w_1+w_{1,-1}\right),\quad w_{11}=\frac{1}{2}\left(w_1-w_{1,-1}\right). \tag{2.0.23}$$

Каждая из этих задач может независимо решаться на своем исполнителе.

Может быть, для однопроцессорного вычислительного комплекса с четырьмя ядрами такого расщепления будет достаточно для решения не очень объемных задач. Как правило, для решения уравнений и систем эллиптического типа в настоящее время применяются сетки, включающие в себя миллионы и десятки миллионов узлов, и достигнутого на данном пути ускорения будет недостаточно.

3. Реализация решений эллиптических уравнений

3.1. Последовательная реализация

Рассмотрим реализацию метода верхней релаксации (SOR) для разностной аппроксимации уравнения Пуассона.

Самый распространенный метод решения дифференциальных уравнений является метод сеток. Область решения представляется в виде набора сетки узлов. Выбираем простейший пятиточечный шаблон разностной схемы "крест".

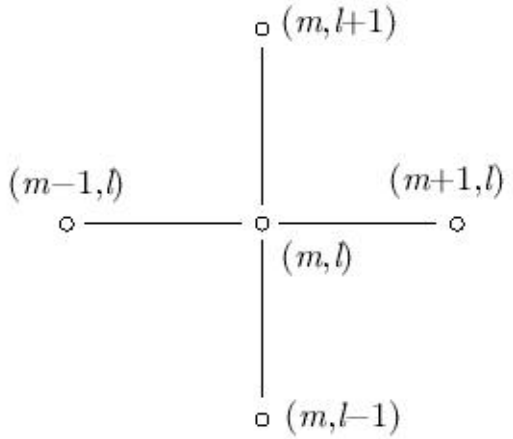

Рис. 1. Шаблон "крест"

Выпишем аппроксимирующее разностное уравнение для этого шаблона. Заменим производные вторыми разностями

$$\frac{u_{m-1,l}+u_{m,l-1}}{h^2}+\frac{u_{m,l+1}+u_{m+1,l}}{h^2}-\frac{4u_{ml}}{h^2}=f_{ml}, u_{ml}=U_{ml}, \qquad (3.1.1)$$

где h — шаг по сетке. Запишем формулу для расчета в методе релаксации:

$$\frac{u^{k+1}_{m-1,l}+u^{k+1}_{m,l-1}}{h^2}+\frac{u^{k}_{m,l+1}+u^{k}_{m+1,l}}{h^2}-\frac{4}{h^2}\left[\frac{u^{k+1}_{ml}}{\omega}+\left(1-\frac{1}{\omega}\right)u^{k}_{ml}\right]=f_{ml}, \qquad (3.1.2)$$

$u^{k+1}_{ml}=U_{m,l}$

Последовательность вычислений в методе релаксации - бегущий счет.

Это же разностное уравнение можно представить в виде:

$$u^{k+1}_{m-1,l} + u^{k+1}_{m,l-1} - \frac{4}{\omega}u^{k+1}_{ml} = -\left(u^{k}_{m,l+1} + u^{k}_{m+1,l}\right) + 4\left(1 - \frac{1}{\omega}\right)u^{k}_{ml} - h^2 f_{ml}, \tag{3.1.3}$$

где решение u^{k+1}_{ml}вычисляется начиная с левого нижнего угла прямоугольной области.

Чтобы реализовать данный метод знание спектра задачи не требуется, а значение параметра ω_O задается опытным путем.

Также в формулировках этот метод встречается как "шахматное упорядочение узлов"(рис. 2)

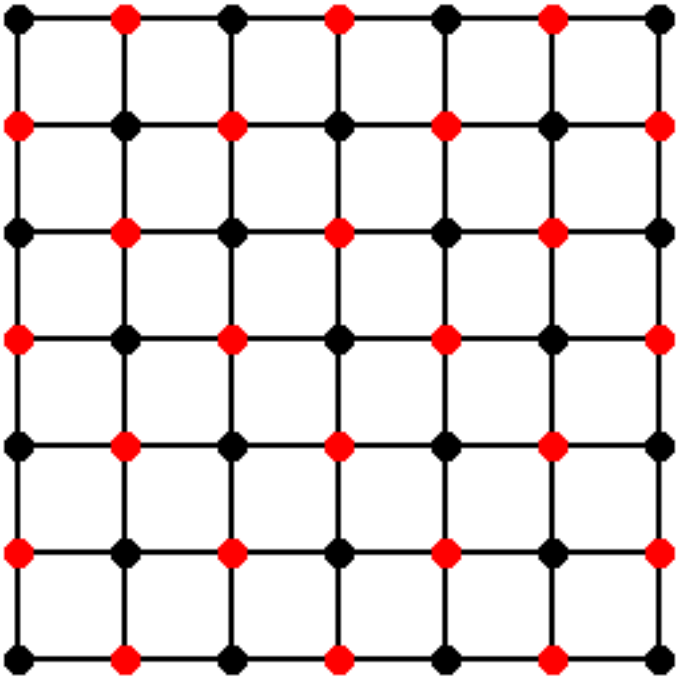

Рис. 2. Шахматное упорядочивание узлов

Все внутренние узлы - черные, если для них сумма значений индексов четная. Остальные внутренние узлы - красные. Тогда можно получить расчетные формулы по пятиточечному шаблону "крест". Для всех красных узлов расчетная формула выглядит также, как и формула (3.1.3) (Значение в красном узле ищется по значениям в черных с предыдущей итерации). Аналогично для значений в черных узлах.

3.2. Параллельная реализация

В современном мире актуальным методом построения многопроцессорных систем является использование процессоров с распределенной памятью. С каждым днем развитие высокопроизводительных кластерных вычислительных систем набирает обороты. Таким образом, актуальность таких систем также растет.

Основной проблемой в использовании данного метода является способ обрабатывания данных между процессорами с распределенной памятью. Возможны следующие способы передачи данных – ленточная схема (вертикальная или горизонтальная) (рис. 3) или блочное разбиение сетки.

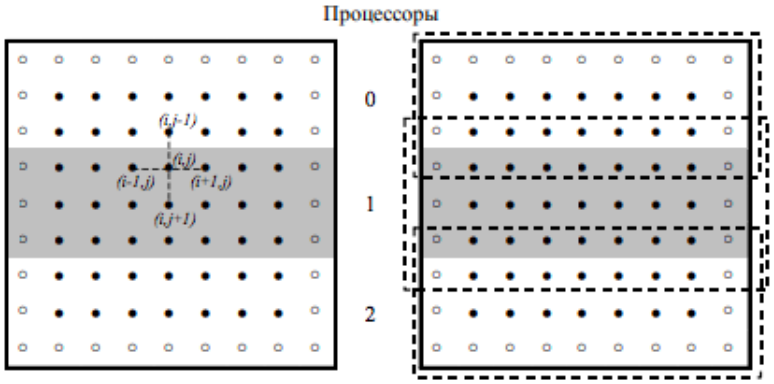

Рис. 3. Ленточное разбиение

Хоть и ленточное разбиение чуть проще в реализации, но не составляет труда перейти от разделения данных с помощью ленточной схемы к блочному способу. Оно не сложно обобщается на нужный нам метод. К тому же изменение способа разбиения сетки не потребует серьезных корректировок рассмотренной схемы параллельных вычислений.

В данной работе использовался метод параллельных вычислений для на многопроцессорных вычислительных системах с общей памятью Open Multi-Processing (OpenMP). Подход заключается в следующем: создание параллельной программы на основе последовательной путем добавления специальных директив. Они задают в программе параллельные области, в которых последовательный код разделяется на раздельные

потоки. Эти потоки могут выполняться на разных процессорах. В итоге программа выглядит в виде набора однопотоковых и многопотоковых участков кода(рис. 4).

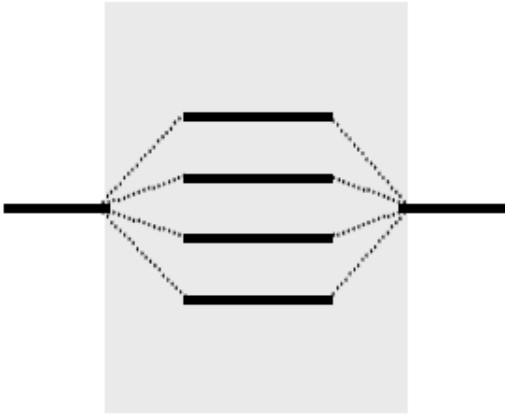

Рис. 4. Параллельные области в OpenMP

4. Результаты расчетов

4.1. Тестовые задачи

В качестве демонстрации работы алгоритма были рассмотрены следующие задачи:

Тест №1 - легкий случай для решения

$$\begin{aligned} &\tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} = 0 \\ &u(x,0) = 0 \\ &u(1,y) = y \\ &u(x,1) = x \\ &u(0,y) = 0 \end{aligned} \tag{4.1.1}$$

Тест №2 - легкий случай для решения + идеально бьется на "четное"/"нечетное"

$$\begin{aligned} &\tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} = -4\pi^2 \left(x^2 + y^2\right) \sin(2\pi xy) \\ &u(x,0) = 0 \\ &u(1,y) = \sin(2\pi y) \\ &u(x,1) = \sin(2\pi x) \\ &u(0,y) = 0 \end{aligned} \tag{4.1.2}$$

Тест №3 - похож на предыдущий, но уже не идеально бьется на "четное"/"нечетное"

$$\begin{aligned} &\tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} = -\tfrac{9}{4}\pi^2 \left(x^2 + y^2\right) \sin(2\pi xy) \\ &u(x,0) = 0 \\ &u(1,y) = \sin(1.5 \cdot \pi y) \\ &u(x,1) = \sin(1.5 \cdot \pi x) \\ &u(0,y) = 0 \end{aligned} \tag{4.1.3}$$

Тест №4

$$\begin{aligned} &\tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} = -\tfrac{5\pi^2}{8}\sin(\pi x)\cdot\cos\left(\tfrac{\pi y}{2}\right) \\ &u(x,0) = \sqrt{\sin(\pi x)} \\ &u(1,y) = 0 \\ &u(x,1) = 0 \\ &u(0,y) = 0 \end{aligned} \tag{4.1.4}$$

Тест №5 -сложный случай для решения

$$\begin{aligned} &\tfrac{\partial^2 u}{\partial x^2} + \tfrac{\partial^2 u}{\partial y^2} = f(x,y) \\ &u(x,0) = 0 \\ &u(1,y) = y \\ &u(x,1) = x \\ &u(0,y) = 0 \end{aligned} \tag{4.1.5}$$

, где

$$f(x,y) = \begin{cases} -10, \text{ if } \left(x-\frac{1}{3}\right)^2 + \left(y-\frac{1}{4}\right)^2 - \frac{1}{25} = 0 \\ 0, \text{ if } \left(x-\frac{1}{3}\right)^2 + \left(y-\frac{1}{4}\right)^2 - \frac{1}{25} \neq 0 \end{cases} \tag{4.1.6}$$

4.2. Результаты

При обработке результатов использовалась первая норма матрицы

$$\|A\|_1 = \max_j \sum_{i=1}^{m} |a_{ij}| \tag{4.2.1}$$

Результаты для Теста №1

	Последовательная реализация	Параллельная реализация
N	50x50	
t, sec	0.680975	0.055420
Error	7.223107e-07	6.610437-e07
Number of Iteration	4352	244
N	100x100	
t, sec	13.797844	0.887352
Error	1.657255e-06	1.625211e-06
Number of Iteration	17152	978
N	500x500	
t, sec	1182.326412	195.57125
Error	7.623471e-06	7.646981e-06
Number of Iteration	97356	11308

Рис. 5. Результаты для Теста №1

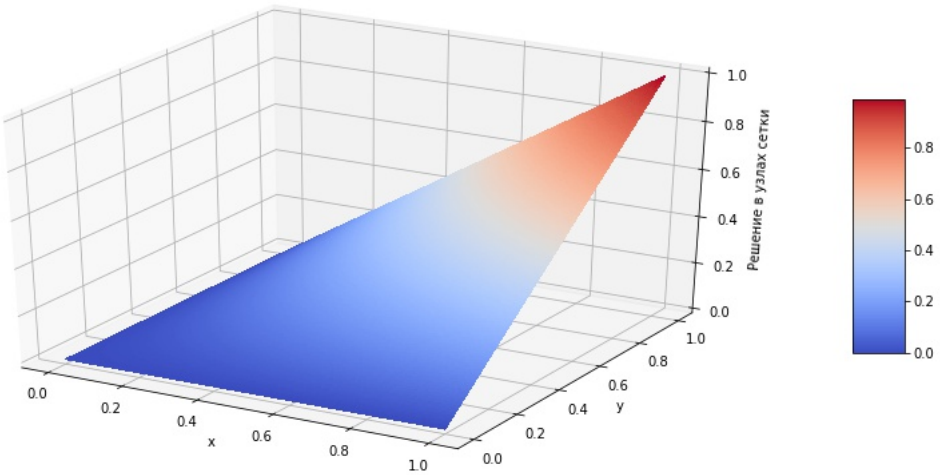

Рис. 6. Решение Теста №1

Результаты для Теста №2

	Последовательная реализация	Параллельная реализация
N	10x10	
t, sec	0.005279	0.001785
Error	5.744081e-04	5.746299e-04
Number of Iteration	238	64
N	50x50	
t, sec	1.834205	0.097715
Error	4.050247e-06	4.211044e-06
Number of Iteration	4365	222
N	100x100	
t, sec	26.89375	1.616918
Error	3.437591e-06	3.363198e-06
Number of Iteration	17240	938

Рис. 7. Результаты для Теста №2

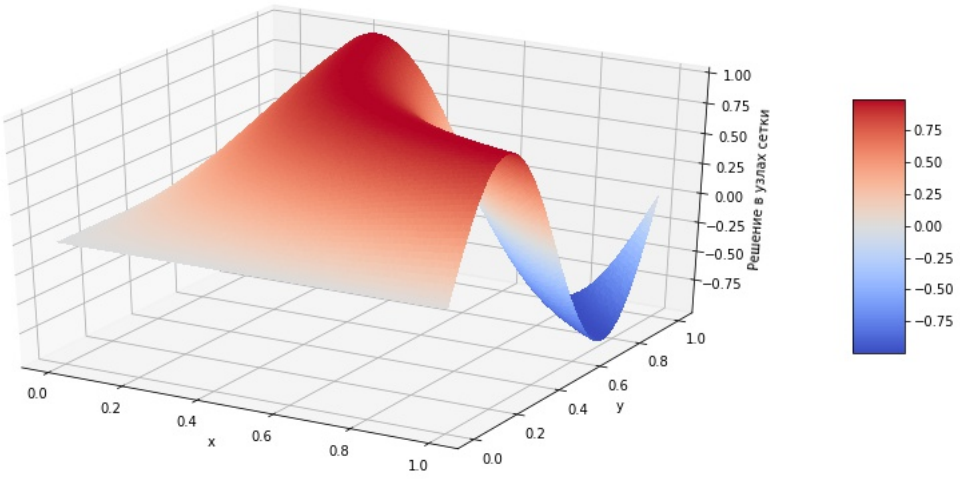

Рис. 8. Последовательное решение Теста №2

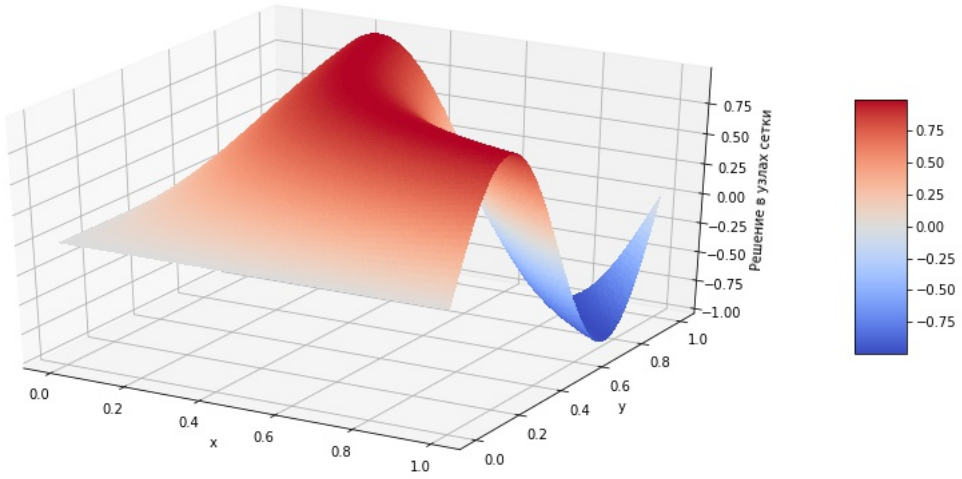

Рис. 9. Параллельное решение Теста №2

Результаты для Теста №3

	Последовательная реализация	Параллельная реализация
N	10x10	
t, sec	0.003962	0.001056
Error	1.294385e-02	1.29437e-02
Number of Iteration	240	64
N	50x50	
t, sec	1.352178	0.090461
Error	3.098873e-03	3.098532e-03
Number of Iteration	4407	232
N	100x100	
t, sec	25.704501	1.661092
Error	1.442211e-03	1.441912e-03
Number of Iteration	17440	963

Рис. 10. Результаты для Теста №3

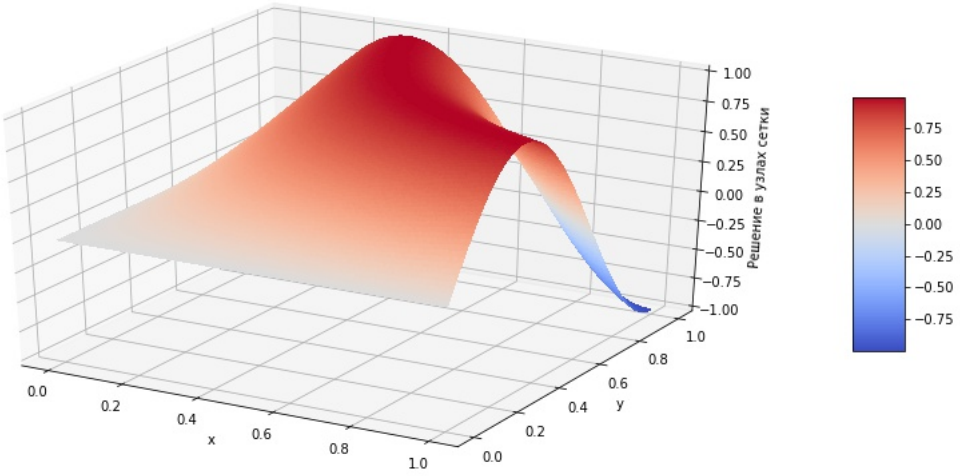

Рис. 11. Последовательное решение Теста №3

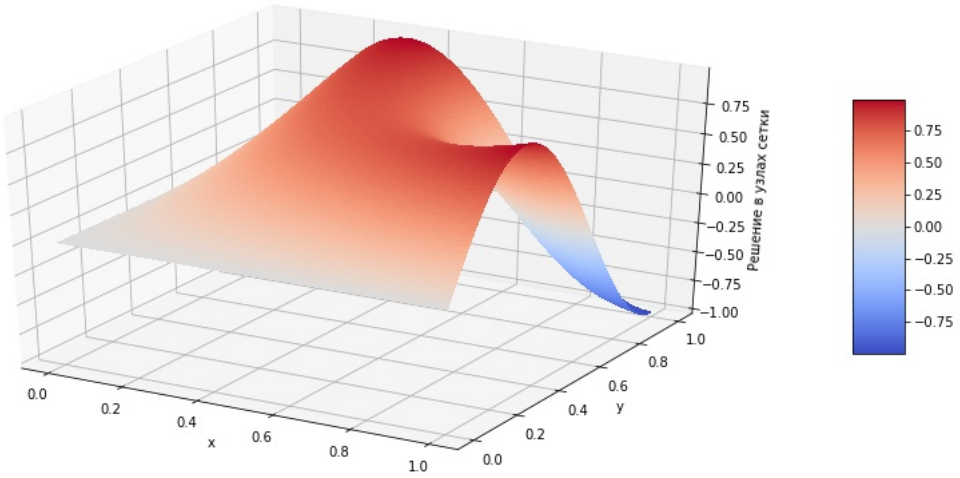

Рис. 12. Параллельное решение Теста №3

Результаты для Теста №4

	Последовательная реализация	Параллельная реализация
N	10x10	
t, sec	0.005323	0.001465
Error	4.5469758e-02	2.257165e-02
Number of Iteration	240	62
N	50x50	
t, sec	1.989258	0.119635
Error	5.379602e-03	5.379498e-03
Number of Iteration	4432	226
N	100x100	
t, sec	39.75971	2.279600
Error	2.502321e-03	5.502001e-03
Number of Iteration	17561	951

Рис. 13. Результаты для Теста №4

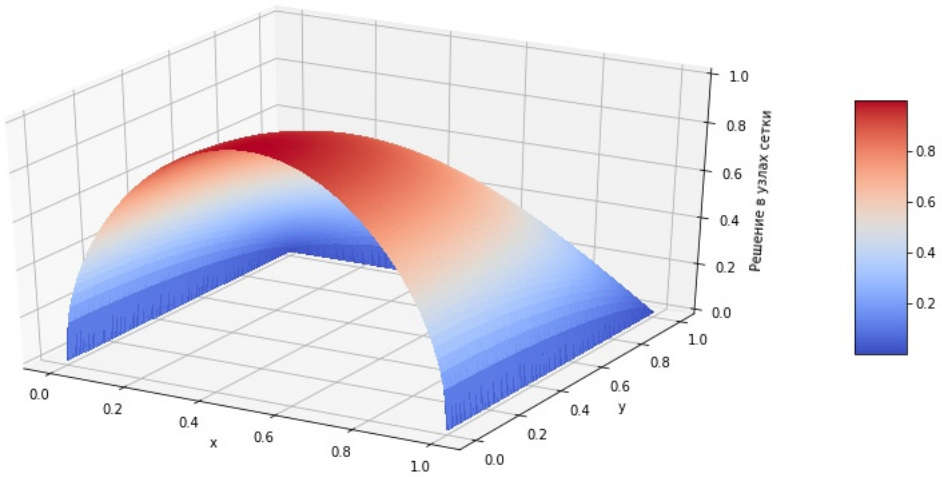

Рис. 14. Последовательное решение Теста №4

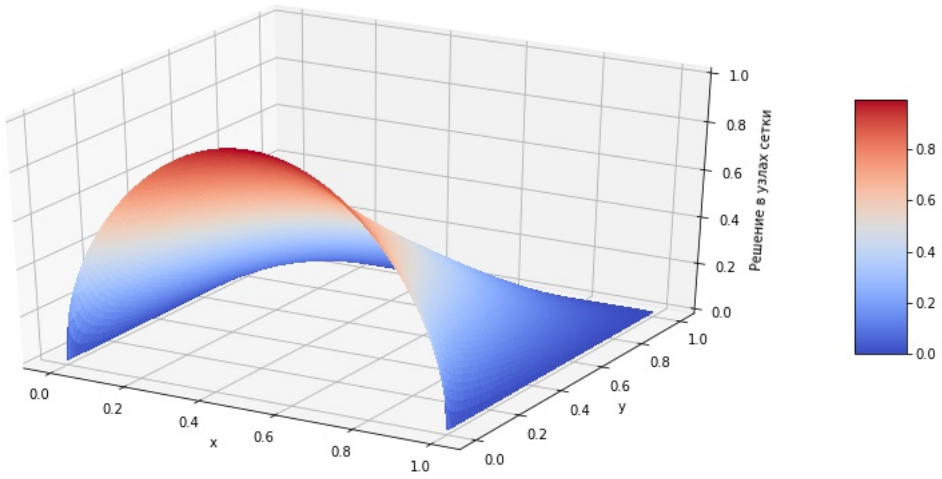

Рис. 15. Параллельное решение Теста №4

Результаты для Теста №5 - сложный случай

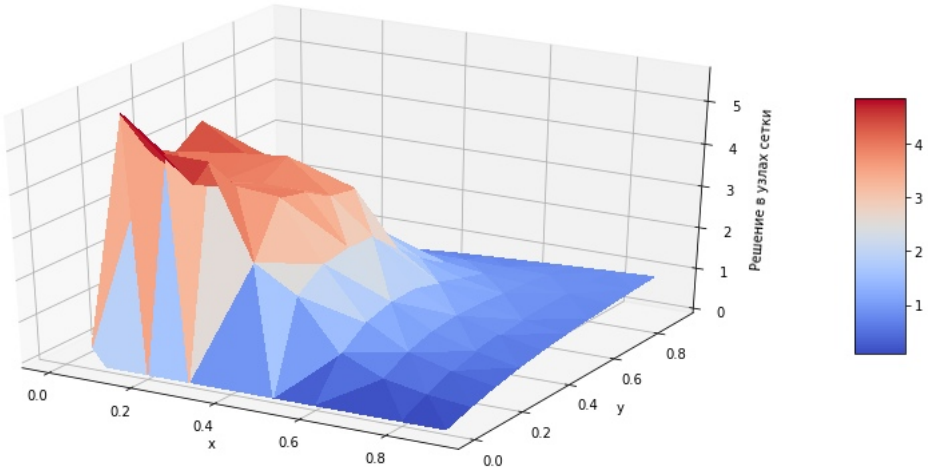

Рис. 16. Решение Теста №5 при N = 10

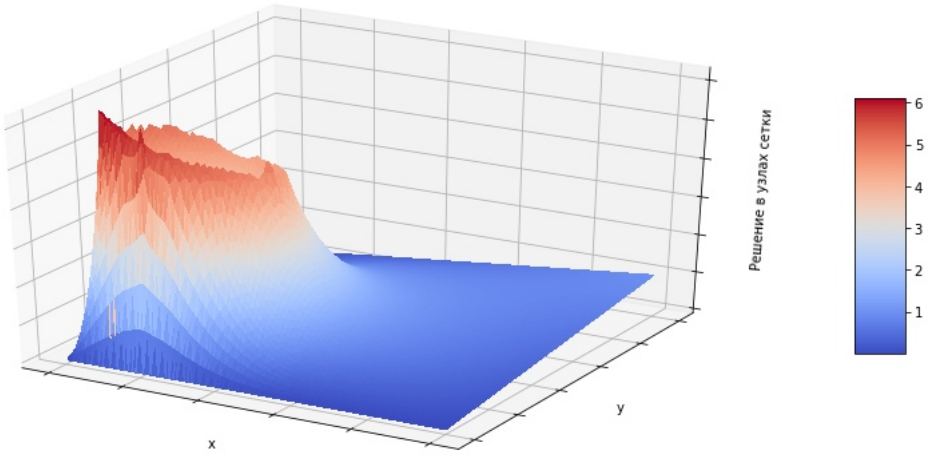

Рис. 17. Решение Теста №5 при N = 100

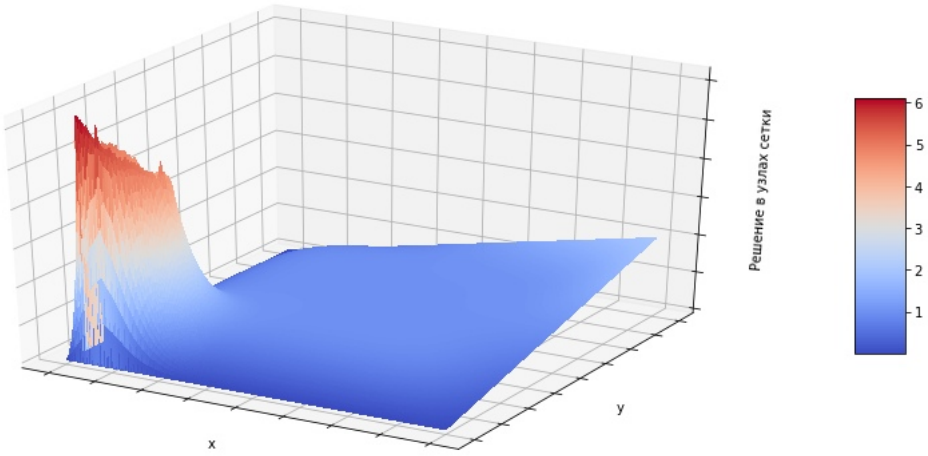

Рис. 18. Решение Теста №5 при N = 200

5. Вывод

В этой работе были реализованы последовательный и параллельный алгоритмы решения дифференциальных уравнений в частных производных. По результатам выполненной работы видно, что использование параллельных вычислений выигрышно на более крупных сетках. На мелких сетках для решения уравнения Пуассона выгоднее использовать последовательный алгоритм, посколько если использовать параллельный метод, то будет тратится время на пересылки граничных условий между блоками. В наших тестовых задачах это не сказалось, но если решать сложные задачи на большом количестве узлов, то очевидно самым рациональным решением будет использовать параллельный метод со способом передачи данных "Блочное разбиение".

6. Список литературы

1. Практические занятия по вычислительной математике в МФТИ / Аристова, Е. Н., Лобанов, А. И./Д:МФТИ, 2015 с.243.

2. А. А. Самарский / Разностные методы для эллиптических уравнений / Самарский А.А, Андреев В. Б. М., 1976 г., с.108

3. Основы параллельных вычислений для многопроцессорных вычислительных систем / Гергель В.П., Стронгин, Р.Г. / Н.Н, 2013 с.136

4. Численные методы /Самарский А.А., Гулин А.В/М.: Наука, 1989с.291

5. Вычислительная линейная алгебра/Дж.Деммель/Пер. с англ. - М.:Мир, 2001 с.347

6. Лекции и упражнения по многосеточным методам/ Ольшанский М.А/М.:ЦПИ,2003 с.68

7. Приложение 1

Программную реализацию, результаты и исходники можно найти на GitHub репозитории

https://github.com/jmacgyve/Solving-Elliptic-Equations