



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра исследования операций

Численное решение двумерной задачи Дирихле для уравнения Пуассона в криволинейной области

Выполнил:
студент 611 группы
Жуков А.С.
Вариант 10

Москва, 2024

Содержание

| | | |
|----------|--|----------|
| 1 | Постановка задачи | 3 |
| 2 | Метод решения задачи | 4 |
| 2.1 | Метод фиктивных областей | 4 |
| 2.2 | Разностная схема | 4 |
| 2.3 | Метод скорейшего спуска для решения СЛАУ | 5 |
| 3 | Результаты | 6 |
| 3.1 | Последовательная реализация + OpenMP | 6 |
| 3.2 | Реализация с использованием MPI | 7 |
| 3.3 | Реализация гибридной программы с использованием MPI и OpenMP | 8 |
| 4 | Литература. | 9 |

1 Постановка задачи

В области $D \subset R^2$ рассматривается дифференциальное уравнение Пуассона вида

$$-\Delta u = f(x, y),$$

в котором оператор Лапласа

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

Для выделения единственного решения уравнение дополняется граничными условиями Дирихле:

$$u(x, y) = 0, \quad (x, y) \in \gamma$$

В конкретном варианте 6, необходимо решить задачу при следующих условиях:

- Область D ограничена дугой гиперболы и отрезком прямой:
 $\{(x, y) : x^2 - 4y^2 > 1, 1 < x < 3\},$
- $f(x, y) = 1$ для всех $(x, y) \in D$.

2 Метод решения задачи

Для решения данной задачи предлагается воспользоваться методом фиктивных областей, который преобразует исходную криволинейную задачу в задачу на прямоугольнике. Такую задачу можно численно решить методом конечных разностей. Данный метод сводит задачу к решению разностной схемы, которую можно приближенно решить итерационным методом скорейшего спуска.

2.1 Метод фиктивных областей

Основная идея метода состоит в том, чтобы расширить первоначальную криволинейную область D до более простой прямоугольной Π . Это позволяет использовать более традиционные численные методы, которые легче реализовать и анализировать. Для этого введем функцию $v(x, y)$, для которой внутри и на границе заданной области D остаются изначальные ограничения, а в введенной фиктивной области $\Pi \setminus \bar{D}$ ограничивается равенством:

$$-\frac{\partial}{\partial x} \left(\frac{1}{\epsilon} \frac{\partial v}{\partial x} \right) - \frac{\partial}{\partial y} \left(\frac{1}{\epsilon} \frac{\partial v}{\partial y} \right) = 0.$$

Требуется найти функцию $v(x, y)$, которая должна быть непрерывной в области Π , и чтобы вектор потока $-1/\epsilon \left(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \right)$ имел непрерывную нормальную компоненту на общей части криволинейной границы области D и прямоугольника Π . Из [2] функция $v(x, y)$ равномерно приближает решение $u(x, y)$ в области D при $\epsilon \rightarrow \infty$.

2.2 Разностная схема

В данной работе полученная краевая задача решалась численно методом конечных разностей [3]. В замыкании прямоугольника $\bar{\Pi}$ определяется равномерная прямоугольная сетка $\bar{\omega}_h = \bar{\omega}_1 \times \bar{\omega}_2$, где $\bar{\omega}_1 = \{x_i = A_1 + i * h_1, i = \overline{0, M}\}$, $\bar{\omega}_2 = \{y_j = A_2 + j * h_2, j = \overline{0, N}\}$, $h_1 = (B_1 - A_1)/M$, $h_2 = (B_2 - A_2)/N$. Через ω_h обозначим множество узлов сетки прямоугольника, не лежащих на границе Γ . Рассмотрим линейное пространство функций, заданных на сетке ω_h . Обозначим через w_{ij} значение сеточной функции $w \in H$ в узле сетки $(x_i, y_j) \in \omega_h$. Будем считать, что в пространстве H задано скалярное произведение и евклидова норма

$$(u, v) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} h_1 h_2 u_{ij} v_{ij}, \|u\| = \sqrt{(u, u)}.$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вида $Aw = B$, где $A : H \rightarrow H$ – оператор, действующий в пространстве сеточных функций, $B \in H$ – известная правая часть. При построении разностной схемы все уравнения краевой задачи аппроксимируются их разностными аналогами сеточными уравнениями, связывающими значения искомой сеточной функции в узлах сетки. Полученные таким образом уравнения должны быть функционально независимыми, а их общее количество совпадать с числом неизвестных, т.е. с количеством узлов сетки. Дифференциальное уравнение задачи во всех внутренних точках сетки аппроксимируются разностным уравнением

$$-\frac{1}{h_1} \left(a_{i+1j} \frac{w_{i+1j} - w_{ij}}{h_1} - a_{ij} \frac{w_{ij} - w_{i-1j}}{h_1} \right) - \frac{1}{h_2} \left(b_{ij+1} \frac{w_{ij+1} - w_{ij}}{h_2} - b_{ij} \frac{w_{ij} - w_{ij-1}}{h_2} \right) = F_{ij}, \quad (*)$$

$i = \overline{1, M-1}, j = \overline{1, N-1}$. Где коэффициенты равны

$$a_{ij} = \frac{1}{h_2} \int_{y_{j-1/2}}^{y_{j+1/2}} k(x_{i-1/2}, t) dt, \quad i = \overline{1, M}, j = \overline{1, N},$$

$$b_{ij} = \frac{1}{h_1} \int_{x_{j-1/2}}^{x_{j+1/2}} k(t, y_{i-1/2}) dt, \quad i = \overline{1, M}, j = \overline{1, N},$$

$$F_{ij} = \frac{1}{h_1 h_2} \iint_{\Pi_{ij}} F(x, y) dx dy, \quad i = \overline{1, M-1}, j = \overline{1, N-1},$$

$$\Pi_{ij} = \{(x, y) : x_{i-1/2} \leq x \leq x_{i+1/2}, y_{i-1/2} \leq y \leq y_{i+1/2}\}.$$

По сути, интегралы в коэффициентах a_{ij} равны длине части отрезка $\left[(x_{i-1/2}, y_{j-1/2}), (x_{i-1/2}, y_{j+1/2}) \right]$, которая лежит в области D . Для b_{ij} аналогично. Интегралы в коэффициентах F_{ij} равны площади пересечения $\Pi_{ij} \cap D$. Все они вычисляются аналитически.

2.3 Метод скорейшего спуска для решения СЛАУ

Полученная разностная схема приближенно решается итерационным методом скорейшего спуска [4]. Начальное приближение $w^{(0)}$ выбран равным нулю во всех точках расчетной сетки. Метод является одношаговым. Итерация $w^{(k+1)}$ вычисляется по итерации $w^{(k)}$ согласно следующему равенству:

$$w_{ij}^{(k+1)} = w_{ij}^k - \tau_{k+1} r_{ij}^{(k)},$$

где невязка $r^{(k)} = Aw^{(k)} - F$, A – самосопряженный и положительноопределенный оператор соответствующий $(*)$, итерационный параметр равен

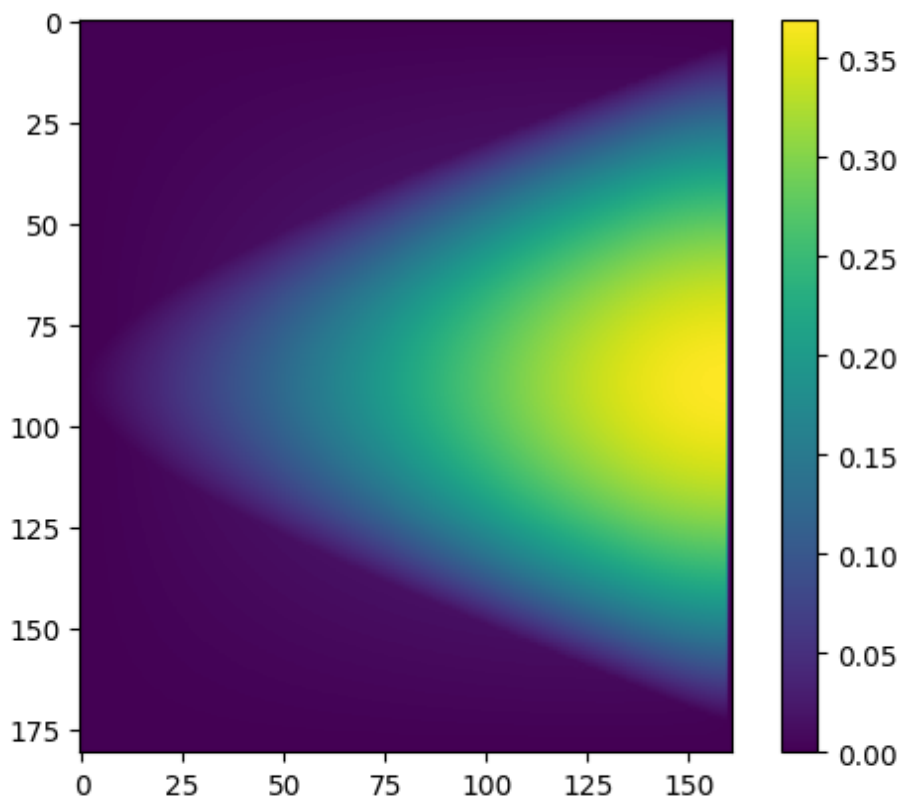
$$\tau_{k+1} = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})}.$$

В качестве условия остановки итерационного процесса используется неравенство

$$\|w^{(k+1)} - w^{(k)}\| < \delta.$$

3 Результаты

Рис. 1: Непрерывная тепловая карта, построенная на решении, полученном на сетке 160×180



3.1 Последовательная реализация + OpenMP

В результате первого этапа работы была написана последовательная программа, реализующая численное решение заданной двумерной задачи Дирихле для уравнения Пуассона в криволинейной области. Основой программы является реализация метода скорейшего спуска. На каждой итерации метода выполняются три цикла: первый вычисляет невязку, второй, используя полученные значения, вычисляет шаг τ , третий меняет значения матрицы w и вычисляет норму разности с предыдущим значением матрицы $\|w^{(k+1)} - w^{(k)}\|$.

Затем в данную программу интегрирована технология OpenMP, позволяющая использовать несколько потоков вычислений в рамках одного процесса. Это было сделано посредством применения директивы `#pragma omp for` к каждому из трёх циклов, составляющих итерацию.

Время работы последовательной программы на сетке 80×90 – 40 секунд, на сетке 160×180 – 541 секунда. Из таблицы 1 видно, что наибольшее ускорение в первом случае достигается на 8 потоках, а на большой сетке на 32 потоках.

Таблица 1: Таблица с результатами расчетов на ПВС IBM Polus (OpenMP код).

| Количество OpenMP-нитей | Число точек сетки ($M \times N$) | Число итераций | Время решения (с) | Ускорение |
|-------------------------|------------------------------------|----------------|-------------------|-----------|
| 2 | 80×90 | 88589 | 6.79 | 5.89 |
| 4 | 80×90 | 88589 | 3.78 | 10.58 |
| 8 | 80×90 | 88589 | 2.36 | 16.95 |
| 16 | 80×90 | 88589 | 2.51 | 15.93 |
| 4 | 160×180 | 328567 | 49.03 | 11.03 |
| 8 | 160×180 | 328567 | 28.56 | 18.94 |
| 16 | 160×180 | 328567 | 24.09 | 22.46 |
| 32 | 160×180 | 328567 | 22.78 | 23.74 |

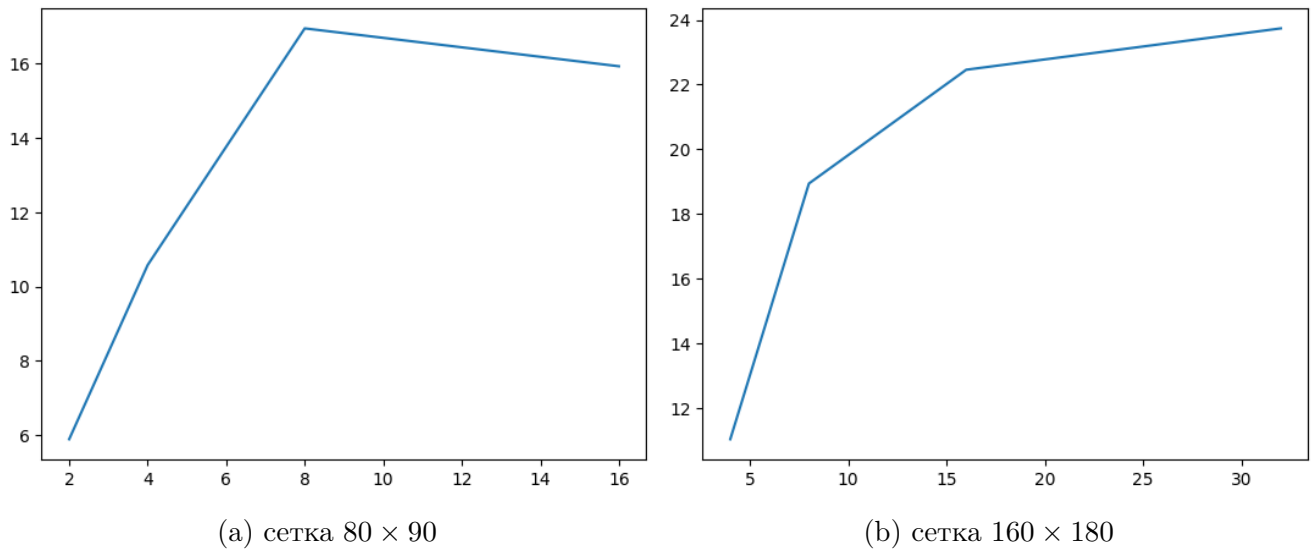


Рис. 2: Графики ускорений

3.2 Реализация с использованием MPI

В результате следующего этапа данная программа была изменена с учетом особенностей технологии MPI, которая позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. Сетка, приближающая исходную задачу, была разделена на области оптимальным образом с помощью функции `MPI_Dims_create`. Затем все процессы на каждой итерации считали значение невязки в своей области, а потом с помощью сообщений обменивались значениями на границах с соседними процессами. Далее, с помощью редуцирования значений по всем областям считается итерационный параметр τ и значение нормы разности решений на двух итерациях $\|w^{(k+1)} - w^{(k)}\|$. Затем вычисляются значения искомой функции в каждой области и процессы так же обмениваются значениями на границе. Результаты тестирования приведены в таблице 2, из которой видно значительное ускорение при увеличении числа используемых вычислительных узлов.

Таблица 2: Таблица с результатами расчетов на ПВС IBM Polus (MPI код).

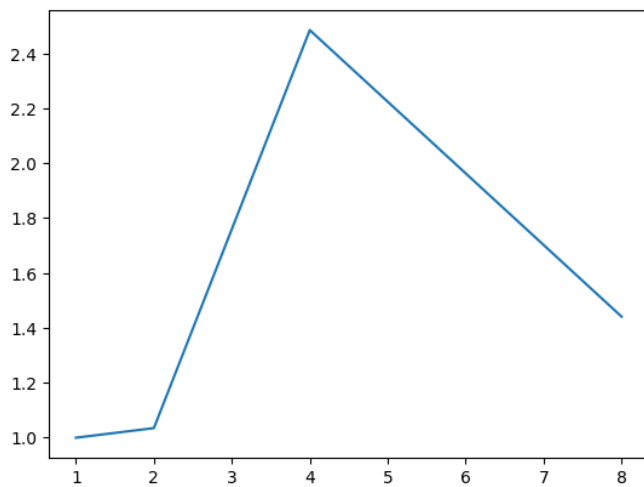
| Количество процессов MPI | Число точек сетки ($M \times N$) | Число итераций | Время решения (с) | Ускорение |
|--------------------------|------------------------------------|----------------|-------------------|-----------|
| 1 | 40×40 | 18989 | 1.897 | 1 |
| 2 | 40×40 | 18989 | 1.001 | 1.895 |
| 4 | 40×40 | 18989 | 0.564 | 3.363 |

3.3 Реализация гибридной программы с использованием MPI и OpenMP

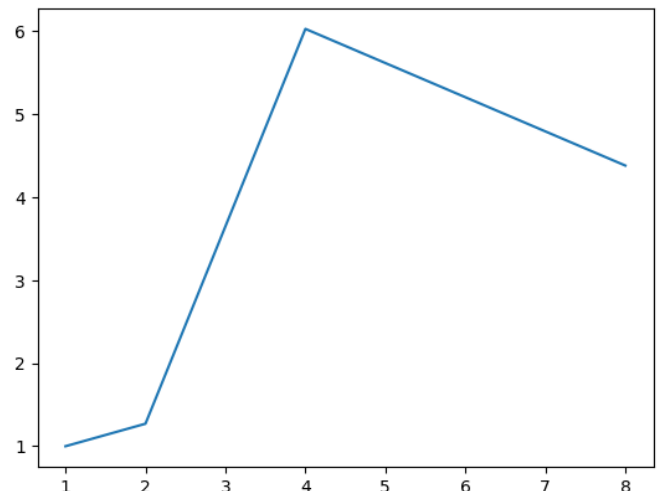
Для выполнения задания была модифицирована программа использующая технологию MPI. Как и в случае с последовательной программой, каждый из трёх циклов внутри итерации был разделён на несколько параллельных OpenMP процессов директивой `#pragma omp for`. Из таблицы 3, содержащей результаты экспериментов, видно, что наибольшее ускорение в обоих случаях достигается на 4 OpenMP потоках.

Таблица 3: Таблица с результатами расчетов на ПВС IBM Polus (MPI+OpenMP код).

| Количество процессов MPI | Количество OpenMP-нитей в процессе | Число точек сетки ($M \times N$) | Число итераций | Время решения (с) | Ускорение |
|--------------------------|------------------------------------|------------------------------------|----------------|-------------------|-----------|
| 2 | 1 | 80×90 | 85243 | 18.31 | 1 |
| 2 | 2 | 80×90 | 85243 | 17.69 | 1.035 |
| 2 | 4 | 80×90 | 85243 | 7.45 | 2.458 |
| 2 | 8 | 80×90 | 85243 | 12.71 | 1.441 |
| 4 | 1 | 160×180 | 323945 | 188.45 | 1 |
| 4 | 2 | 160×180 | 323945 | 148.24 | 1.271 |
| 4 | 4 | 160×180 | 323945 | 30.38 | 6.203 |
| 4 | 8 | 160×180 | 323945 | 43.04 | 4.378 |



(а) сетка 80×90



(б) сетка 160×180

Рис. 3: Графики ускорений

4 Литература.

1. А.Н. Тихонов, А.А. Самарский. Уравнения математической физики. М. Изд. "Наука". 1977.
2. Г.И. Марчук. Методы вычислительной математики. М. Изд. "Наука". 1989.
3. А.А. Самарский. Теория разностных схем. М. Изд. "Наука". 1989.
4. А.А. Самарский, А.В. Гулин. Численные методы. М. Изд. "Наука". 1989.
5. В.А. Ильин, Г.Д. Ким. Линейная алгебра и аналитическая геометрия. Изд. Московского университета. 2002.
6. IBM Polus - <http://hpc.cmc.msu.ru>