Московский Государственный Университет имени
М.В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчёт по заданию в рамках курса «Суперкомпьютерное моделирование и технологии» Численное решение задачи Дирихле для уравнения Пуассона в криволинейной области

Выполнил:
Де Ен Де
608 группа
Вариант 3

Москва 2023

Содержание

1 Математическая постановка задачи

В области $D \subset R^2$, ограниченной контуром γ, рассматривается дифференциальное уравнение Пуассона:

$$-\Delta u = 1,$$

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 граничным условием Дирихле:

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

Для данной работы мне был предложен **вариант 3**, который соответствует следующим точкам: $A(3, 0), B(0, 2), C(-3, 0)$.

2 Численный метод решения уравнения

Для решения был выбран предложенный метод наименьших невязок. Этот метод позволяет получить последовательность сеточных функций $\omega^{(k)} \in H, k = 1, 2, ...$, сходящуюся по норме пространства H к решению разностной схемы, т.е.

$$||\omega - \omega^{(k)}||_E \to 0, k \to \infty.$$

Начальное приближение $\omega^{(0)}$ равно нулю во всех точках расчетной сетки.

Метод является одношаговым. Итерация $\omega^{(k+1)}$ вычисляется по итерации $\omega^{(k)}$ согласно равенствам:

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

где невязка $r^{(k)} = A\omega^{(k)} - B$, итерационный параметр

$$\tau_{k+1} = \frac{(Ar^{(k)}, r^{(k)})}{||Ar^{(k)}||_E^2}.$$

В качестве условия остановки итерационного процесса следу ет использовать неравенство

$$||\omega^{(k+1)} - \omega^{(k)}||_E < \delta,$$

где δ – положительное число, определяющее точность итерационного метода.

3 Краткое описание проделанной работы по созданию *OpenMP*-программы

Для построения *OpenMP*-программы использовались следующие директивы:

```
#pragma omp parallel for default(shared) private(i, j) schedule(dynamic)
```

для арифметических операций сеточных функций,

```
#pragma omp parallel for default(shared) private(i) reduction(+: res) schedule(dynamic)
```

для вычисления суммы(скалярное произведение, интегрирование).

4 Результаты расчетов для разных размеров задач и на разном числе нитей

Выполнение последовательной программы решающей данное задание при M = 80, N = 80 заняло 270.971 секунды. А для M=160, N=160 заняло 913.675 секунд. Значение δ решено было взять равным 10^{-6}. Ускорение считалось как отношение времени выполнения последовательной программы к времени выполнения параллельной программы на той же сетке $Boost = \frac{time(sequential)}{time(parallel)}$.

Число *OpenMP*-нитей	Число точек сетки M × N	Время решения (с)	Ускорение
2	80 × 80	176.46	1.53
4	80 × 80	141.96	1.90
8	80 × 80	103.24	2.62
16	80 × 80	74.81	3.62
4	160 × 160	439.11	1.23
8	160 × 160	449.07	2.03
16	160 × 160	190.53	4.79
32	160 × 160	139.22	6.56

5 Дополнительные графики

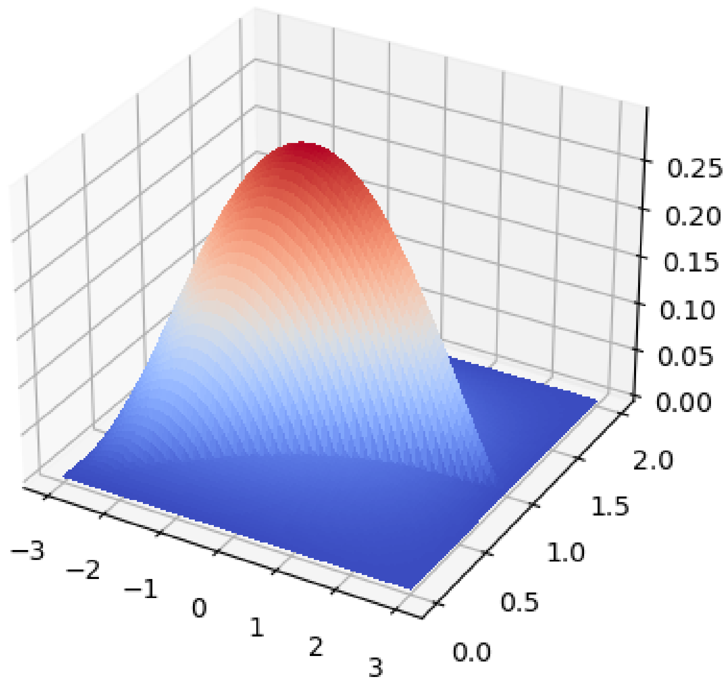

Рисунок 1. Численное решение при разбиении 80 * 80

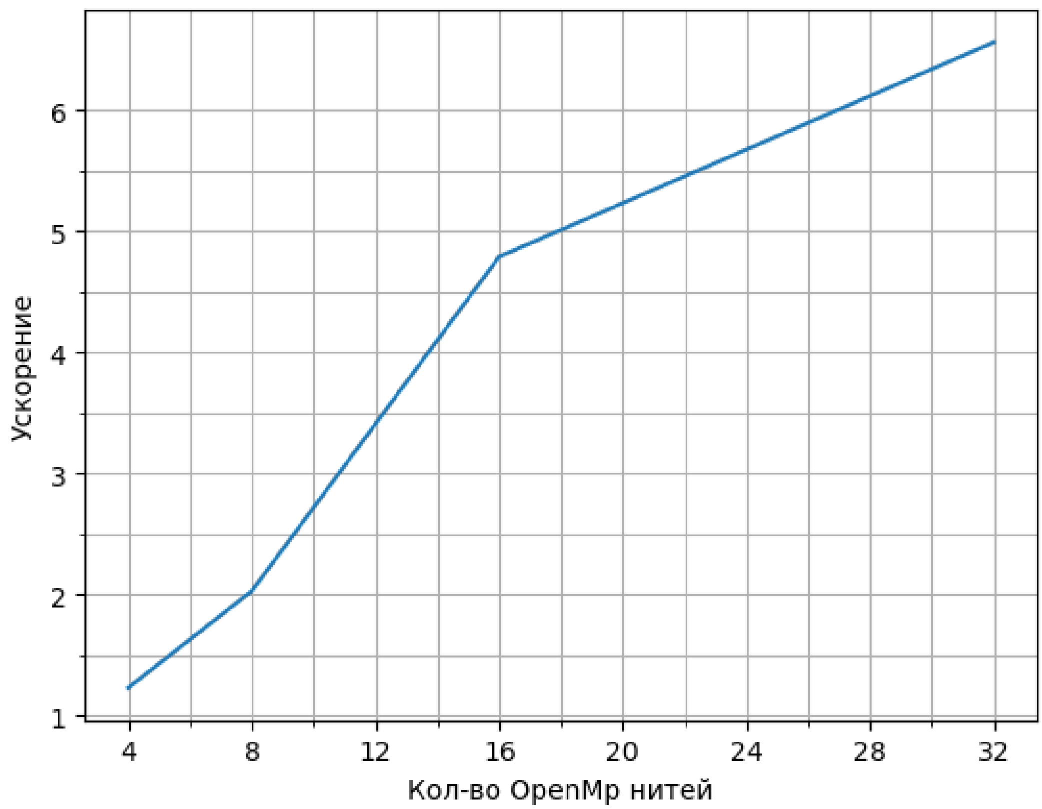

Рисунок 2. Зависимость выполнения задачи на сетке 160 * 160 от количества нитей OpenMP

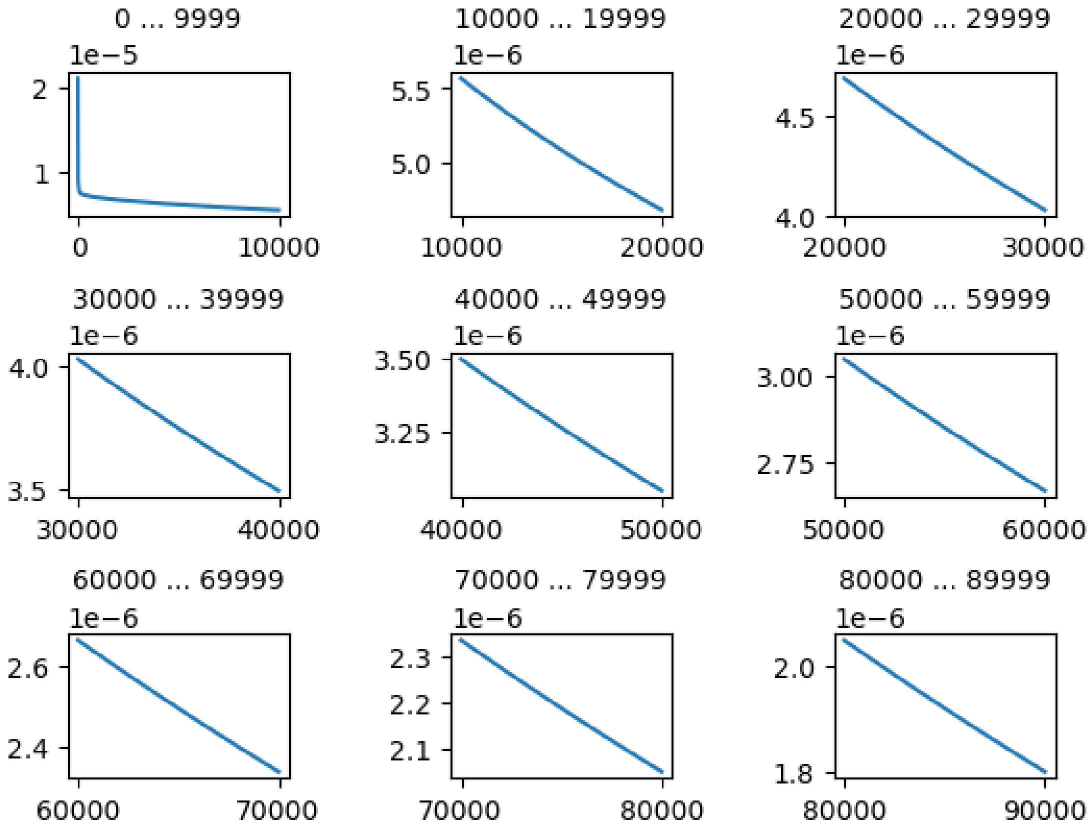

Рисунок 3. График сходимости на каждые 10000 итераций

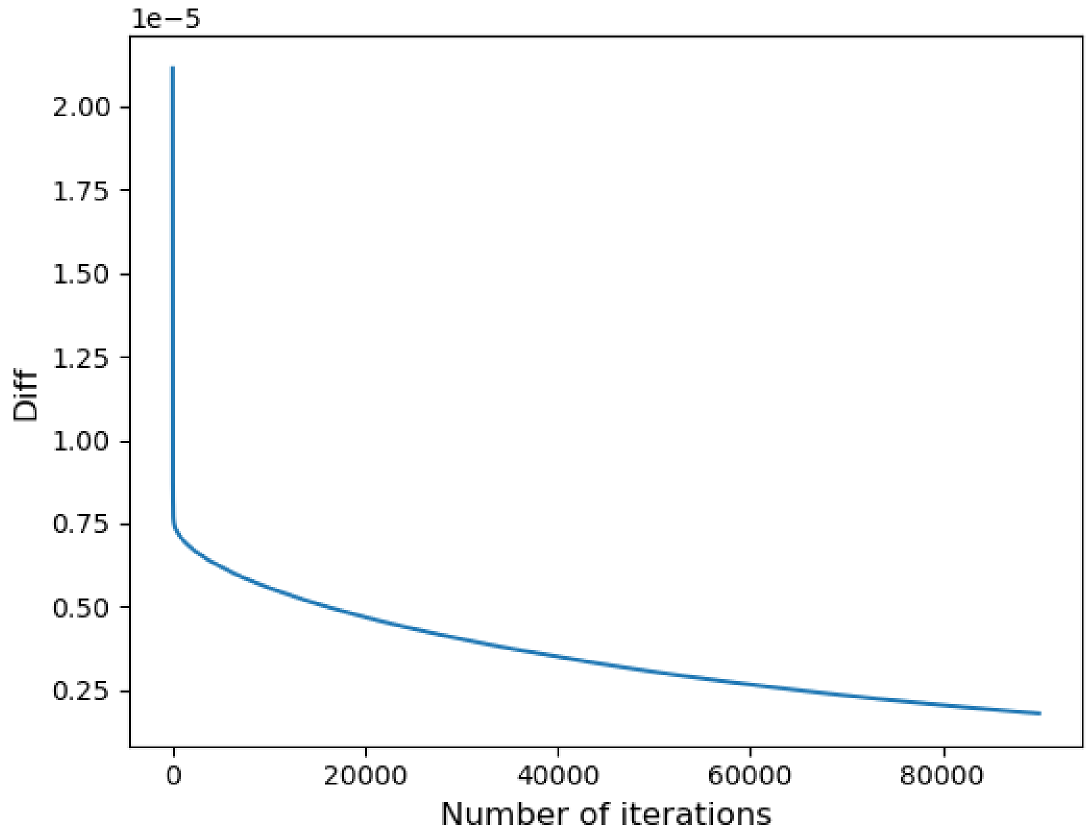

Рисунок 4. Общий график сходимости