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

Отчёт по суперкомпьютерному моделированию

Задание №2

Выполнила:
студентка 611 группы
Макарова Алёна

Москва, 2016

1 Математическая постановка задачи

В прямоугольной области:

$$\Pi = [A_1, A_2] \times [B_1, B_2]$$

требуется найти дважды гладкую функцию $u = u(x, y)$, удовлетворяющую дифференциальному уравнению:

$$-\Delta u = F(x, y), \quad A_1 < x < A_2, \quad B_1 < y < B_2 \tag{1}$$

и дополнительному условию

$$u(x, y) = \phi(x, y) \tag{2}$$

во всех граничных точках (x, y) прямоугольника Π.

2 Численный метод для решения задачи

Для решения задачи используется разностная схема.
В расчетной области Π определяется прямоугольная сетка

$$\omega_h = (x_i, y_j), i = 0, 1, \ldots, N_1, j = 0, 1, \ldots, N_2, \tag{3}$$

Параметры N_1, N_2 определяют размеры решетки.

Приближенным решением задачи (1), (2) называется функция $p = p(x_i, y_j)$, удовлетворяющая уравнениям

$$\begin{gathered} -\Delta_h p_{ij} = F(x_i, y_j), \quad (x_i, y_j) \in \omega_h, \\ p_{ij} = \phi(x_i, y_j), \quad (x_i, y_j) \in \gamma_h \end{gathered} \tag{4}$$

Эти соотношения представляют собой систему линейных алгебраических уравнений с числом уравнений равным числу неизвестных и определяют единственным образом неизвестные значения p_{ij}. Совокупность уравнений (4) называется разностной схемой для задачи (1), (2).

Приближенное решение этой системы может быть получено методом скорейшего спуска. Начальное приближение

$$p_{ij}^{(0)} = \phi(x_i, y_j), \quad (x_i, y_j) \in \gamma_h,$$

во внутренних узлах сетки $p_{ij}^{(0)}$ - любые числа. Метод является одношаговым. Последующие итерации вычисляются согласно равенствам:

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

где невязка

$$\begin{gathered} r_{ij}^{(k)} = -\Delta_h p_{ij}^{(k)} - F(x_i, y_j), \quad (x_i, y_j) \in \omega_h, \\ r_{ij}^{(k)} = 0, \quad (x_i, y_j) \in \gamma_h. \end{gathered} \tag{5}$$

Итерационный параметр

$$\tau_{k+1} = \frac{(r^{(k)}, r^{(k)})}{(-\Delta_h r^{(k)}, r^{(k)})}$$

Известно, что с увеличением номера итерации k последовательность сеточных функций $p^{(k)}$ сходится к точному решению p задачи (4) по норме пространства H, то есть

$$\| p - p^k \| \to +\infty, \quad k \to +\infty.$$

Существенно большей скоростью сходимости обладает метод сопряженных градиентов. Начальное приближение $p^{(0)}$ и первая итерация $p^{(1)}$ вычисляются так же, как и в методе скорейшего спуска. Последующие итерации осуществляются по формулам:

$$p_{ij}^{(k+1)} = p_{ij}^{(k)} - \tau_{k+1} g_{ij}^{(k)}, \quad k = 1, 2, \ldots$$

Здесь

$$\tau_{k+1} = \frac{(r^{(k)}, g^{(k)})}{(-\Delta_h g^{(k)}, g^{(k)})}$$

вектор

$$g_{ij}^{(k)} = r_{ij}^{(k)} - \alpha_k g_{ij}^{(k-1)}, \quad k = 1, 2, \ldots$$
$$g_{ij}^{(0)} = r_{ij}^{(0)},$$

коэффициент

$$\alpha_{k+1} = \frac{(-\Delta_h r^{(k)}, g^{(k-1)})}{(-\Delta_h g^{(k-1)}, g^{(k-1)})}.$$

Вектор невязки $r^{(k)}$ вычисляется согласно равенствам (5). Итерационный процесс останавливается, как только

$$\| p^{(n)} - p^{(n-1)} \| < \epsilon, \tag{6}$$

где ϵ - заранее выбранное положительное число.

3 Вариант

Номер варианта: $11 mod 18 + 18(11 mod 2) = 29$

$$F(x, y) = \frac{x^2 + y^2}{4(4 + xy)^{\frac{3}{2}}}, \phi(x, y) = \sqrt{4 + xy}$$
$$\Pi = [0, 4] \times [0, 4]$$

Задача решается для равномерной решетки и евклидовой нормы.

4 Описание реализации

Задача нахождения приближённого решения реализована отдельной функцией **Iter**. Основной алгоритм:
1. Инициализация начального решения и вектора невязки;
2. Вычисление невязки;
3. Вычисление итерационных параметров;
4. Нахождение нового решения;
5. Проверка на выполение условия останова. Если не выполнено - п.2.
Далее, программа была доработана для реализации параллельного выполнения.

4.1 MPI

Решётка, на которой решается задача, разбивается на прямоугольники. Каждый прямоугольник - область расчета процесса. Таким образом, каждый процесс выполняет действия с частью всё решетки. Каждый процесс вычисляет смещение, соответствующее его области в "большой"матрице.

Основным преобразованем алгоритма является организация обмена информацией между процессами, так как часть данных одноо процесса используется при вычислении в другом процессе. Для этого, с помощью функции MPI_Cart_create, создаётся коммуникатор с декартовой топологией, для каждого процесса определяются его соседи. Каждый раз перед тем как вычислить 5-ти точечную аппроксимацию - процессы обмениваются данными со своими соседями, передавая им обновлённые данные(MPI_Send/MPI_Recv). Для расчета скалярного произведения используется MPI_Reduce (сбор из всех процессов - передача всем процессам).

Процесс с номером 0 является менеджером системы. Вначале он собирает от каждого процесса их смещения и размеры области. Затем отправляет соответвующие координаты, для которых должны быть проведены расчеты. А после - агрегирует полученный релультат в "большой"матрице решения.

4.2 OpenMP

В рамках вычислений также возможно производить независимые операции для каждого узла сетки. Основные используемые возможности openmp: распараллеливание циклов (pragma omp for) и использование reduce для агрегации результата между нитями.

Возможность распараллеливания между нитями присутствует в следующих операциях:

1. Вычисление невязки, новой фунции решения и подобные, так как вычисление каждой ячейки – это независимая операция.
2. Подготовка данных к отправке посредством MPI другим процессам, и обработка принятых данных)
3. Вычисление 5-ти точечной аппроксимации (аналогично п.1)
4. Вычисление скалярного произведения (aggregate sum)

5 Результаты расчетов

Исследование проводилось на суперкомпьютерах:

1. "Ломоносов"
2. IBM Blue Gene/P

Для сетки 1000x1000 программа совершает 1513 итераций.
Для сетки 2000x2000 программа совершает 2831 итерацию.

ПВС	Число процессоров	Размерность сетки	Время (с)	Ускорение
Lomonosov	1	1000 × 1000	477.30	1
Lomonosov	8	1000 × 1000	60.76	7.85
Lomonosov	16	1000 × 1000	30.23	15.79
Lomonosov	32	1000 × 1000	15.20	31.4
Lomonosov	128	1000 × 1000	4.35	109.72
Lomonosov	1	2000 × 2000	timeout	-
Lomonosov	8	2000 × 2000	453.64	1
Lomonosov	16	2000 × 2000	227.58	1.99
Lomonosov	32	2000 × 2000	114.40	3.97
Lomonosov	128	2000 × 2000	29.40	15.43
BlueGene(MPI)	1	1000 × 1000	1644.20	1
BlueGene(MPI)	128	1000 × 1000	67.74	24.27
BlueGene(MPI)	256	1000 × 1000	34.39	47.80
BlueGene(MPI)	512	1000 × 1000	19.29	85.23
BlueGene(MPI)	1	2000 × 2000	timeout	-
BlueGene(MPI)	128	2000 × 2000	496.15	1
BlueGene(MPI)	256	2000 × 2000	249.77	1.98
BlueGene(MPI)	512	2000 × 2000	129.40	3.83
BlueGene(OpenMP/MPI)	1	1000 × 1000	1317.3	1
BlueGene(OpenMP/MPI)	128	1000 × 1000	26.16	50.4
BlueGene(OpenMP/MPI)	256	1000 × 1000	17.85	73.8
BlueGene(OpenMP/MPI)	512	1000 × 1000	12.48	105.55
BlueGene(OpenMP/MPI)	1	2000 × 2000	timeout	-
BlueGene(OpenMP/MPI)	128	2000 × 2000	166.21	1
BlueGene(OpenMP/MPI)	256	2000 × 2000	91.91	1.8
BlueGene(OpenMP/MPI)	512	2000 × 2000	60.36	2.75

6 Графики приближенного решения и точного решения

На этом графике оба решения - точное и приближённое. Для того чтобы показать их различие, я приведу график разности.

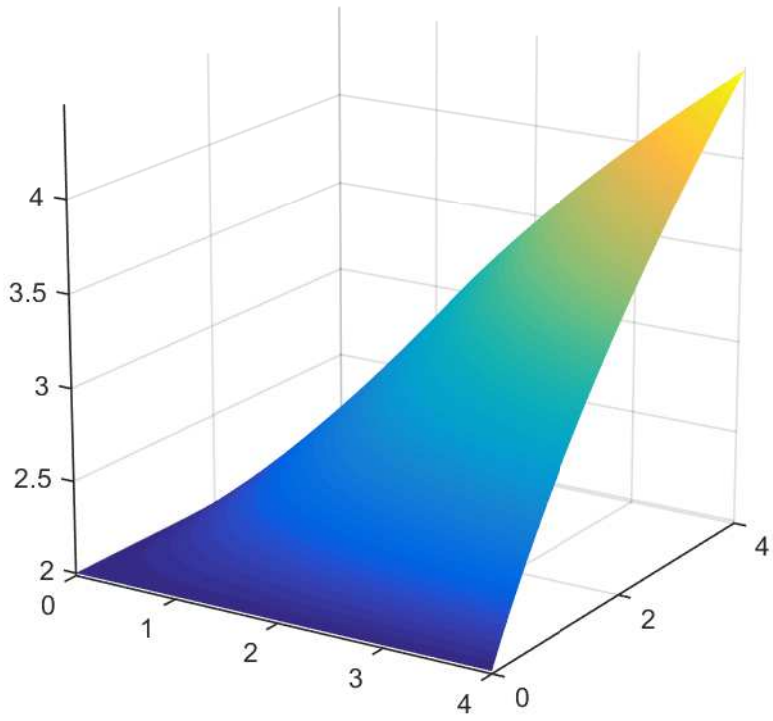

Рис. 1: Решение

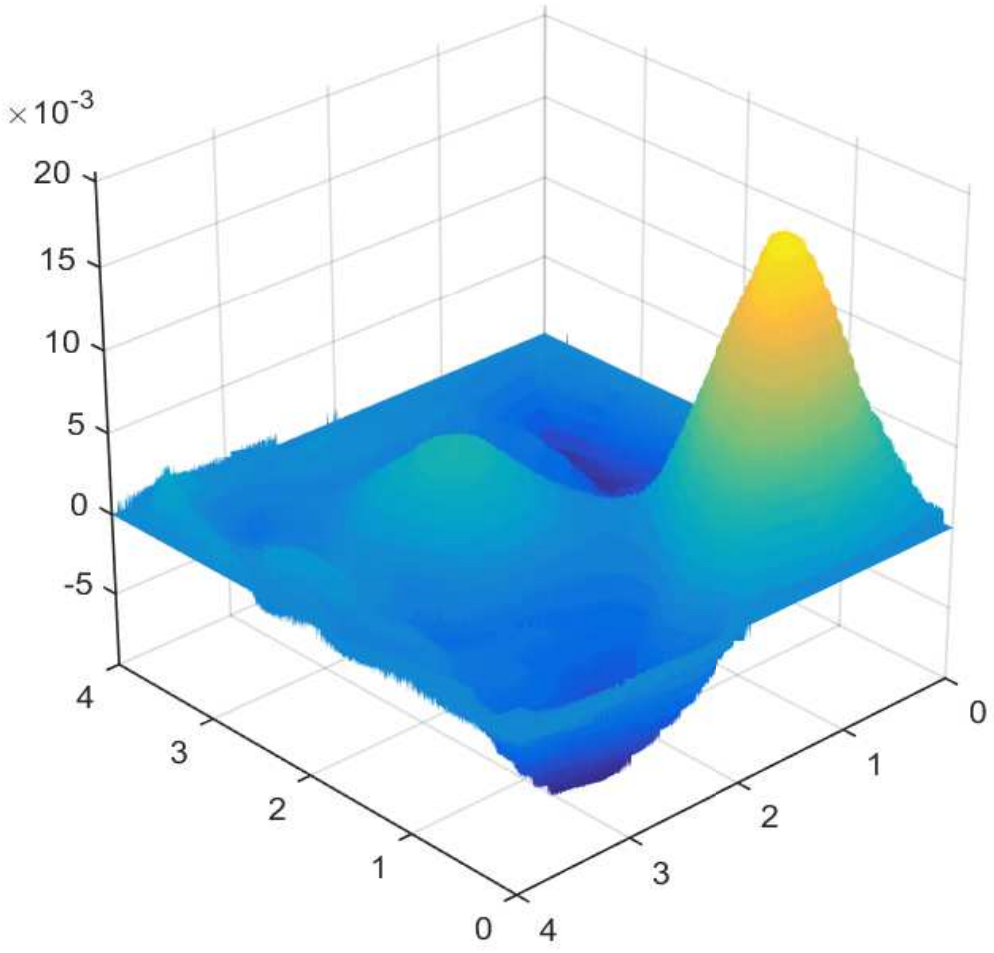

Рис. 2: Разность точного и аналитического решения