Отчёт по заданию: Решение двумерного уравнения Пуассона методом фиктивных областей с использованием OpenMP

Вариант: №9
Студент: Ма Синьюэ 617

Цель работы

Целью данной работы является численное решение двумерного уравнения Пуассона с использованием метода фиктивных областей и реализация параллельного алгоритма на основе технологии OpenMP.

Постановка задачи

Для успешного выполнения задания требуется:

1. Разработать последовательный код программы, вычисляющий приближённое решение разностной схемы методом скорейшего спуска, выполнить расчёты на сгущающихся сетках:
$$(M, N) = (10, 10), (20, 20), (40, 40).$$

2. Используя средства OpenMP, разработать параллельный код программы, вычисляющий приближённое решение разностной схемы, проверить качество работы алгоритма, выполнив расчёты на сетке
$$(M, N) = (40, 40)$$
при числе потоков 1, 4, и 16, провести сравнение с последовательным вариантом алгоритма.

3. Заполнить таблицу с результатами расчётов OpenMP-программы.

Метод и реализация

Область, сетка и правая часть

Рассматривается эллиптическая область
$$D = \{(x, y) : \ x^2 + 4y^2 < 1\},$$

вложенная в прямоугольный контейнер

$$A_1 = -1,\ B_1 = 1, \qquad A_2 = -0.6,\ B_2 = 0.6.$$

Сетка равномерная:

$$x_i = A_1 + i\,h_x, \quad i = 0, \ldots, M; \qquad y_j = A_2 + j\,h_y, \quad j = 0, \ldots, N,$$

где $h_x = \dfrac{B_1 - A_1}{M},\ h_y = \dfrac{B_2 - A_2}{N}$. Правая часть строится как индикатор области:

$$B_{ij} = \begin{cases} F_{\mathrm{VAL}} = 1, & (x_i, y_j) \in D, \\ 0, & (x_i, y_j) \notin D. \end{cases}$$

Метод фиктивных областей и выбор ε

Коэффициент теплопроводности задаётся по правилу

$$k(x,y) = \begin{cases} 1, & (x,y) \in D, \\ \dfrac{1}{\varepsilon}, & (x,y) \notin D, \end{cases} \qquad \boxed{\varepsilon = \big(\max\{h_x, h_y\}\big)^2}.$$

(Именно такая формула ε используется в коде: `eps = max(h1,h2)*max(h1,h2);`).

Коэффициенты схемы a_{ij}, b_{ij}

Для ячейки (i,j) вычисляются доли пересечения её граней с областью D. Длины пересечения отрезков с эллипсом:

$$x = \mathrm{const} = x_0: \quad y \in \left[-\sqrt{\tfrac{1-x_0^2}{4}},\ \sqrt{\tfrac{1-x_0^2}{4}}\right], \qquad y = \mathrm{const} = y_0: \quad x \in \left[-\sqrt{1-4y_0^2},\ \sqrt{1-4y_0^2}\right].$$

Пусть $\ell_{ij}^{(v)}$ — часть вертикального ребра длиной h_y, лежащая в D, а $\ell_{ij}^{(h)}$ — часть горизонтального ребра длиной h_x, лежащая в D. Тогда (ровно как в функции `fictitious_regions_setup`):

$$a_{ij} = \begin{cases} 1, & \ell_{ij}^{(v)} = h_y, \\ \dfrac{1}{\varepsilon}, & \ell_{ij}^{(v)} = 0, \\ \dfrac{\ell_{ij}^{(v)}}{h_y} + \dfrac{1 - \frac{\ell_{ij}^{(v)}}{h_y}}{\varepsilon}, & \text{иначе}, \end{cases} \qquad b_{ij} = \begin{cases} 1, & \ell_{ij}^{(h)} = h_x, \\ \dfrac{1}{\varepsilon}, & \ell_{ij}^{(h)} = 0, \\ \dfrac{\ell_{ij}^{(h)}}{h_x} + \dfrac{1 - \frac{\ell_{ij}^{(h)}}{h_x}}{\varepsilon}, & \text{иначе}. \end{cases}$$

Действие оператора A

Разностный оператор для внутренних узлов ($i = 1..M-1,\ j = 1..N-1$) реализован как

$$(Aw)_{ij} = -\frac{1}{h_x}\left(a_{i+1,j}\frac{w_{i+1,j}-w_{i,j}}{h_x} - a_{i,j}\frac{w_{i,j}-w_{i-1,j}}{h_x}\right) - \frac{1}{h_y}\left(b_{i,j+1}\frac{w_{i,j+1}-w_{i,j}}{h_y} - b_{i,j}\frac{w_{i,j}-w_{i,j-1}}{h_y}\right).$$

(См. функцию `apply_A`).

Диагональный предобуславливатель D^{-1}

Используется диагональное предобуславливание (см. `apply_Dinv`):

$$D_{ij} = \frac{a_{i+1,j}+a_{i,j}}{h_x^2} + \frac{b_{i,j+1}+b_{i,j}}{h_y^2}, \qquad z_{ij} = (D^{-1}r)_{ij} = \frac{r_{ij}}{D_{ij}}.$$

Метод сопряжённых градиентов (PCG)

Итерационный процесс в `solve`:

$$r^{(0)} = B, \quad z^{(0)} = D^{-1}r^{(0)}, \quad p^{(1)} = z^{(0)}, \quad \langle u, v\rangle = \sum_{i,j} u_{ij}v_{ij}\,h_x h_y;$$

$$\text{на шаге } k: \quad \alpha_k = \frac{\langle z^{(k-1)}, r^{(k-1)}\rangle}{\langle Ap^{(k)}, p^{(k)}\rangle}, \qquad w^{(k)} = w^{(k-1)} + \alpha_k p^{(k)},$$

$$r^{(k)} = r^{(k-1)} - \alpha_k A p^{(k)}, \qquad z^{(k)} = D^{-1}r^{(k)}, \qquad \beta_{k+1} = \frac{\langle z^{(k)}, r^{(k)}\rangle}{\langle z^{(k-1)}, r^{(k-1)}\rangle},$$

$$p^{(k+1)} = z^{(k)} + \beta_{k+1}p^{(k)}.$$

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

где $\|\cdot\|_E$ — евклидова норма, а δ задаётся пользователем (в тестах $\delta = 10^{-6}$).

Параллельная реализация OpenMP

Параллелизация выполняется с помощью директив:

```
#pragma omp parallel for collapse(2)
```

Они применяются в циклах:

- вычисление коэффициентов a_{ij}, b_{ij}, F_{ij};
- операция Aw;
- обновление векторов r, z, p, w.

Время измеряется с помощью `omp_get_wtime()`.

Результаты расчётов

1. Последовательная версия

Таблица 1: Результаты для последовательной программы (метод скорейшего спуска).

Размер сетки $(M \times N)$	**Число итераций**	**Время решения**
10×10	17	0.0001
20×20	31	0.0005
40×40	61	0.0034

2. Параллельная версия (OpenMP)

Размер сетки $(M \times N)$	**Число нитей**	**Число итераций**	**Время решения**
40×40	1	60	0.005
40×40	4	60	0.004
40×40	16	60	0.009

Размер сетки $(M \times N)$	**Число нитей**	**Число итераций**	**Время решения**	**Ускорение**
400×600	2	546	3.098	1.59
400×600	4	546	1.776	2.78
400×600	8	546	1.170	4.21
400×600	16	546	0.977	5.05
800×1200	4	989	12.214	3.19
800×1200	8	989	7.554	5.15
800×1200	16	989	6.034	6.45
800×1200	32	989	6.216	6.27

Анализ результатов, представленных в таблице 2, показывает, что параллельная реализация с использованием OpenMP демонстрирует существенное ускорение по сравнению с последовательным вариантом. Для сетки 400×600 ускорение возрастает почти линейно до 16 нитей, достигая значения $S \approx 5.0$. Для более крупной сетки 800×1200 наблюдается схожая тенденция: при увеличении числа нитей до 16 ускорение достигает $S \approx 5.5$, однако дальнейшее увеличение числа потоков до 32 не приводит к заметному улучшению производительности из-за возрастания накладных расходов на синхронизацию и ограничений пропускной способности памяти.

Таким образом, оптимальное количество потоков для данной задачи составляет 8–16, что обеспечивает баланс между временем решения и эффективностью параллельного использования ресурсов.

Выводы

- Метод фиктивных областей позволяет корректно учесть сложную форму области без явного задания граничных условий.

- Метод сопряжённых градиентов с диагональным предобуславливанием обеспечивает устойчивую и быструю сходимость.
- Реализация OpenMP демонстрирует значительное ускорение при увеличении числа потоков.
- При больших размерах сетки параллельный алгоритм показывает почти линейное ускорение до 16 потоков.