Report _ OpenMP
Тан Цзычэнь 604 edu-cmc-skmodel24-604-04
31/10/2024

1. Математическая постановка задачи

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

$$-\Delta u = f(x,y) \quad (1)$$

в котором оператор Лапласа

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

функция $f(x,y)$ считается известной. Для выделения единственного решения уравнение дополняется граничными условием Дирихле (см. [1]):

$$u(x,y) = 0,\ (x,y) \in \gamma \qquad (2)$$

Требуется найти функцию $u(x,y)$, удовлетворяющую уравнению (1) в области D и краевому условию (2) на ее границе.

Требуется приближенно найти решение задачи (1),(2) для случая, когда $f(x,y) = 1$ при всех $(x,y) \in D$. Конкретное задание определяется геометрией области D.

5. трапеция с вершинами в точках $A(-3,0), B(3,0), C(2,3), D(-2,3)$;

$\Pi = \{(x,y): A_1 < x < B_1, A_2 < y < B_2\}$.

$$\hat{D} = \Pi \setminus \bar{D}$$

Выберем и зафиксируем малое $\varepsilon > 0$.(Выполняя расчеты, считать константу ε метода фиктивных областей равной h^2, где $h = \max(h_1, h_2)$ - наибольший шаг сетки $\bar{\omega}_h$.)

$$-\frac{\partial}{\partial x}\left(k(x,y)\frac{\partial v}{\partial x}\right) - \frac{\partial}{\partial y}\left(k(x,y)\frac{\partial v}{\partial y}\right) = F(x,y), (x,y) \in \mathrm{D}$$
$$v(x,y) = 0,\ when(x,y) \text{on the boundary of the trapezoid} \qquad (3)$$

$$k(x,y) = \begin{cases} 1,\ (x,y) \in D, \\ \dfrac{1}{\varepsilon},\ (x,y) \in \hat{D} \end{cases} \qquad (4)$$

$$F(x,y) = \begin{cases} f(x,y), & (x,y) \in D \\ 0, & (x,y) \in \hat{D}. \end{cases} \qquad (5)$$

Последнее означает, что в каждой точке $(x_0, y_0) \in \gamma \cap \Pi$ должно выполняться равенство:

$$\lim_{(x,y)\to(x_0,y_0),(x,y)\in D} \left(W(x,y), n(x_0,y_0)\right) = \lim_{(x,y)\to(x_0,y_0),(x,y)\in \hat{D}} \left(W(x,y), n(x_0,y_0)\right) \qquad (6)$$

где $n(x,y)$ - вектор единичной нормали к границе γ в точке (x,y), определенный всюду или почти всюду на кривой.

Известно [2], что функция $v(x,y)$ равномерно приближает решение $u(x,y)$ задачи (1),(2) в области D, а именно,

$$\max_{P\in\bar{D}}|v(x,y)-u(x,y)|<C\varepsilon,\ C>0 \qquad (7)$$

2 . Разностная схема решения задачи.

Краевые задачу (3),(6) предлагается решать 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

$$\bar{\omega}_1 = \{x_i = A_1 + ih_1, i = \overline{0,M}\}, \bar{\omega}_2 = \{y_j = A_2 + jh_2, j = \overline{0,N}\}.$$

Здесь $h_1 = (B_1 - A_1)/M, h_2 = (B_2 - A_2)/N$.

$$(u,v) = \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} h_1 h_2 u_{ij} v_{ij},\ \ \| u \|_E = \sqrt{(u,u)} \qquad (8)$$

В методе конечных разностей дифференциальная задача математической физики заменяется конечно-разностной операторной задачей вида

$$Aw = B, \qquad (9)$$

Дифференциальное уравнение задачи (3) во всех внутренних точках сетки аппроксимируется разностным уравнением:

$$-\frac{1}{h_1}\left(a_{i+1j}\frac{w_{i+1j}-w_{ij}}{h_1} - a_{ij}\frac{w_{ij}-w_{i-1j}}{h_1}\right) - \frac{1}{h_2}\left(b_{ij+1}\frac{w_{ij+1}-w_{ij}}{h_2} - b_{ij}\frac{w_{ij}-w_{ij-1}}{h_2}\right) = F_{ij},$$
$$i = \overline{1,M-1}, j = \overline{1,N-1} \qquad (10)$$

в котором коэффициенты:

$$a_{ij} = \frac{1}{h_2}\int_{y_{j-1/2}}^{y_{j+1/2}} k\left(x_{i-1/2}, t\right)dt,\ b_{ij} = \frac{1}{h_1}\int_{x_{i-1/2}}^{x_{i+1/2}} k\left(t, y_{j-1/2}\right)dt \qquad (11)$$

при всех $i = \overline{1,M}, j = \overline{1,N}$. Здесь полуцелые узлы

$$x_{i\pm1/2} = x_i \pm 0.5h_1,\ y_{j\pm1/2} = y_j \pm 0.5h_2$$

Правая часть разностного уравнения

$$F_{ij} = \frac{1}{h_1 h_2}\iint_{\Pi_{ij}} F(x,y)dxdy,\ \Pi_{ij} = \left\{(x,y): x_{i-1/2} \leqslant x \leqslant x_{i+1/2}, y_{j-1/2} \leqslant y \leqslant y_{j+1/2}\right\} \qquad (12)$$

при всех $i = \overline{1, M-1}, j = \overline{1, N-1}$.
Введем обозначения правой и левой разностных производных по переменным x, y соответственно:

$$w_{x,ij} = \frac{w_{i+1j} - w_{ij}}{h_1}, \quad w_{\bar{x},ij} = w_{x,i-1j} = \frac{w_{ij} - w_{i-1j}}{h_1}$$
$$w_{y,ij} = \frac{w_{ij+1} - w_{ij}}{h_2}, \quad w_{\bar{y},ij} = w_{y,ij-1} = \frac{w_{ij} - w_{ij-1}}{h_2}$$

С учетом принятых обозначений разностное уравнение (10) можно представить в более компактном и удобном виде:

$$-(\boldsymbol{a w_{\bar{x}}})_{\boldsymbol{x,ij}} - \left(\boldsymbol{b w_{\bar{y}}}\right)_{\boldsymbol{y,ij}} = \boldsymbol{F_{ij}},\ \boldsymbol{i = \overline{1, M-1}, j = \overline{1, N-1}} \qquad \textbf{(13)}$$

Краевые условия Дирихле задачи (3),(6) аппроксимируются точно равенством:

$$\boldsymbol{w_{ij} = w(x_i, y_j) = 0},\ \boldsymbol{(x_i, y_j) \in \Gamma} \qquad \textbf{(14)}$$

3. Метод решения системы линейных алгебраических уравнений.

Приближенное решение системы уравнений (2) для сформулированных выше краевых задач может быть получено итерационным методом наименьших невязок. Этот метод позволяет получить последовательность сеточных функций $w^{(k)} \in H, k = 1, 2, \ldots$, сходящуюся по норме пространства H к решению разностной схемы, т.е.

$$\left\| w - w^{(k)} \right\|_E \to 0, \quad k \to +\infty$$

Начальное приближение $w^{(0)}$ можно выбрать любым способом, например, равным нулю во всех точках расчетной сетки.
Метод является одношаговым. Итерация $w^{(k+1)}$ вычисляется по итерации $w^{(k)}$ согласно равенствам:

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

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

$$\tau_{k+1} = \frac{\left[Ar^{(k)},\ r^{(k)}\right]}{\left\|Ar^{(k)}\right\|_E^2}$$

В качестве условия остановки итерационного процесса следует использовать неравенство

$$\left\| w^{(k+1)} - w^{(k)} \right\|_{\boldsymbol{E}} < \boldsymbol{\varepsilon}$$

где ε - положительное число, определяющее точность итерационного метода. Оценку точности приближенного решения сеточных уравнений (2) можно проводить в других нормах пространства сеточных функций, например, в максимум норме

$$\|w\|_c = \max_{x \in \overline{w}_h} |w(x)|$$

Константу ε для данной задачи предлагается взять равной 10^{-6} .

4. краткое описание проделанной работы по созданию OpenMP-программы.

Размер матрицы A в уравнении (2) равен $\big((M-1)\times(N-1)\big)\times\big((M-1)\times(N-1)\big)$.

Описание переменных:
MatrA_index - индекс столбца элемента в матрице, который не равен 0.
MatrA_val - Значение элемента в матрице, который не равен 0.
Matr_map - матрица результатов.
VecB - вектор B.
VecRes - результирующий вектор.
struct CSR - разреженная матрица

const double A1 = -3, B1 = 3; // x
const double A2 = 0, B2 = 3; // y

Описание функции:
Функция func_k возвращает значение функции $k(x,y)$
Функция func_F возвращает значение функции $F(x,y)$
Функция calculate_aij вспомогательная функция для расчета a_ij
Функция calculate_bij вспомогательная функция для расчета b_ij
Функция calculate_Fij вспомогательная функция для расчета F_ij
Функция Filling_MatrA_VecB() заполняет матрицу A и вектор B по формуле из главы 2.
Функция Change_Matr_to_CSR() преобразует MatrA_index и MatrA_val в разреженную матрицу CSR.
Функция SpMV() вычисляет произведение разреженных матриц и векторов.
Функция Axpy() представляет собой линейное вычисление вектора.
Функция Vcopy() скопирует вектор.
Функция Vdiff() вычисляет разность двух векторов.
Функция Dot() вычисляет скалярное произведение.
Функция Norm_Vec() вычисляет норму вектора.
Функция Slove_SOLE_sequential() использует метод наименьших невязок для решения уравнения.
Функция Get_map() переводит результирующий вектор в плоскость.

Для последовательного программного кода я использую вышеуказанные функции, подробности см. в «trymain_seq.c».

log	sequential	M N	iteration	time/s	initial_value	eps
log10_seq	try.1184839	10 10	390	0.000000s	1.44	0.000005
log20_seq	try.1184860	20 20	6020	0.340000s	1.44	0.000005
log40_seq	try.1184749	40 40	73862	17.350000s	1.44	0.000005

try.1184916	80 90	906554	1008.18	1.44	0.000005
My_PC	160 180	6111612	5234.344	1.44	0.000005

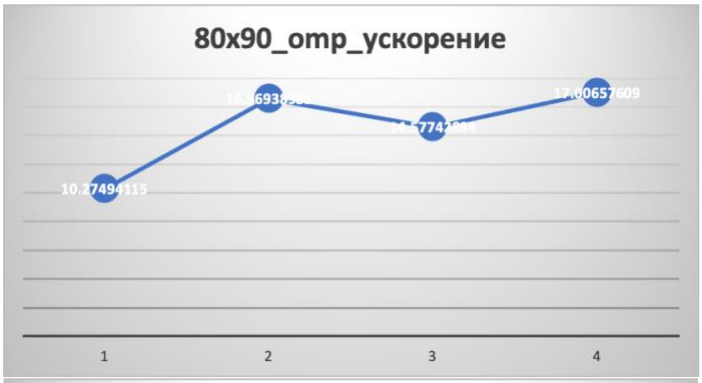
< task1_omp.c>

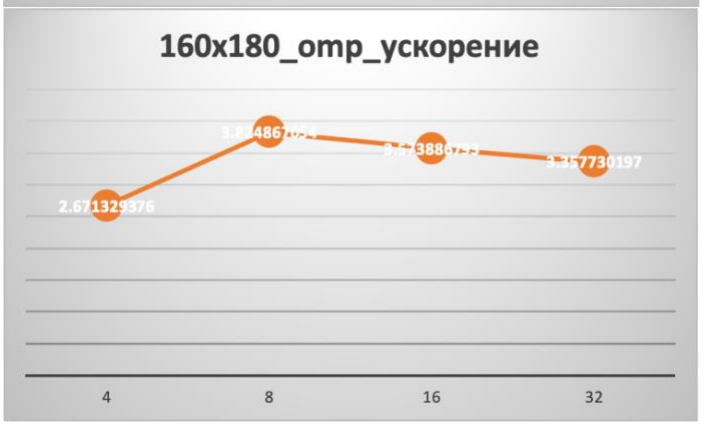
Реализация программы с использованием OpenMP:

OpenMP	thread	queue	M N	iteration	time/s
omp_t.1185684	1	normal	40 40	73862	3.708259s
omp_t.1185678	2	normal	40 40	73862	3.059998s
omp_t.1185678	4	normal	40 40	73862	2.339716s
omp_t.1185678	16	normal	40 40	73862	2.729556s

ускорение = T(sequential) / T(Parallel)

	OpenMP	thread	queue	M N	iteration	time/s	ускорение
1	omp_t.1185637	2	normal	80 90	906554	98.12027	10.2749411513034
2	omp_t.1185637	4	normal	80 90	906554	60.845934	16.5693898297296
3	omp_t.1185637	8	normal	80 90	906554	69.160344	14.5774289381788
4	omp_t.1185637	16	normal	80 90	906554	59.281774	17.0065760852568
1	run.1192366.out	4	normal	160 180	6111612	1959.452865	2.67132937642774
2	run.1192365.out	8	normal	160 180	6111612	1368.503513	3.8248670538853
3	run.1192001.out	16	normal	160 180	6111612	1464.608227	3.57388679341346
4	run.1191991.out	32	normal	160 180	6111612	1558.893566	3.35773019670029





Get the Solution image using Python :
Result_file in Polus : edu-cmc-skmodel24-604-04@polus.hpc.cs.msu.ru:/home_edu/edu-cmc-skmodel24-604/edu-cmc-skmodel24-604-04/

< MN40x40_omp_n8.out >< MN80x90_omp_n16.out > < Big_omp_n16.out>