МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ. MPI. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ПО МЕТОДУ ПРЯМОУГОЛЬНИКОВ ДЛЯ ДВУМЕРНОЙ ФУНКЦИИ.**

ЛАБОРАТОРНАЯ РАБОТА №5

студента 4 курса 451 группы

направления 09.03.04 — Программная инженерия

факультета КНиИТ

Голикова Артема Олеговича

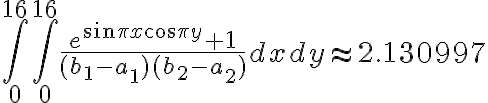
Проверил

канд. физ-мат. н., доцент \_\_\_\_\_\_\_\_\_\_\_\_ К. П. Савина

Саратов 2020

**Цель работы:** изучить использование потоков MPI приводящих к уменьшению вычисления интегралов для двумерной функции на примере метода прямоугольников; сравнить время выполнения с использованием разных режимов.

**Ход работы:**



Код программы реализующий метод прямоугольников для двумерной функции:

// Work11

#include <time.h>

#include <iostream>

#include <mpi.h>

#define PI 3.1415926535897932384626433832795

int NProc, ProcId;

using namespace std;

double f(double x, double y)

{

return (exp(sin(PI \* x) \* cos(PI \* y)) + 1);

}

void integral(const double a1, const double b1, const double a2, const double b2,

const double h, double\* res)

{

double sum; // локальная переменная для подсчета интеграла

sum = 0.0;

for (double x = a1 + h / 2.0; x < b1; x+= h)

{

for (double y = a2 + (ProcId + 1) \* h / 2.0; y < b2; y += NProc \* h)

{

sum += f(x,y) / ((b1 - a1) \* (b2 - a2));

}

}

MPI\_Reduce(&sum, res, 1, MPI\_DOUBLE, MPI\_SUM, 0, MPI\_COMM\_WORLD);

if (ProcId == 0)

{

\*res \*= h \* h;

}

}

double experiment(double\* res)

{

double stime; // время начала и конца расчета

double a1 = 0.0, a2 = 0.0; // левая граница интегрирования

double b1 = 16.0, b2 = 16.0; // правая граница интегрирования

double h = 0.001; // шаг интегрирования

if (ProcId == 0)

{

stime = MPI\_Wtime();

}

integral(a1, b1, a2, b2, h, res); // вызов функции интегрирования

if (ProcId == 0)

{

return MPI\_Wtime() - stime;

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

int i; // переменная цикла

double time; // время проведенного эксперимента

double res; // значение вычисленного интеграла

double min\_time; // минимальное время работы

// реализации алгоритма

double max\_time; // максимальное время работы

// реализации алгоритма

double avg\_time; // среднее время работы

// реализации алгоритма

int numbExp = 10; // количество запусков программы

MPI\_Init(NULL, NULL);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &NProc);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &ProcId);

min\_time = max\_time = avg\_time = experiment(&res);

// оставшиеся запуски

for (i = 0; i < numbExp - 1; i++)

{

time = experiment(&res);

if (ProcId == 0)

{

avg\_time += time;

if (max\_time < time) max\_time = time;

if (min\_time > time) min\_time = time;

}

}

// вывод результатов эксперимента

if (ProcId == 0)

{

cout << " Среднее время: " << avg\_time / numbExp << "\n Минимальное время: " <<

min\_time << "\n Максимальное время:" << max\_time << endl;

cout.precision(8);

cout << " Значение интеграла: " << res << endl;

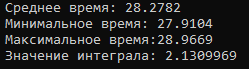
}

MPI\_Finalize();

return 0;

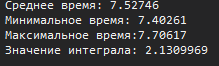
}

Результат выполнения в однопоточном режиме:



Код программы реализующий метод прямоугольников для параллельной реализации (1-ый способ):

Результат выполнения в многопоточном режие:



Сравнение результатов:

|  |  |  |
| --- | --- | --- |
| Последовательная  реализация | Параллельная  Реализация | Ускорение |
| Мин. время: 27,9104  Макс. время: 28,9669  Сред. время: 28,2782 | Мин. время: 7,40261  Макс. время: 7,70617  Сред. время: 7,52746 | 3,7567 |

Вычисления происходили на процессоре Intel Core i5-2310 (4 ядра, 4 потока). По результатам сравнения видно, что распараллеливание при вычислении интегралов для двумерной функции дает прирост по быстродействию в 3.7 раза.

**Вывод:** изучил использование потоков MPI приводящих к уменьшению вычисления интегралов для двумерной функции на примере метода прямоугольников; сравнил время выполнения с использованием разных режимов.