Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новосибирский государственный технический университет»

Факультет автоматики и вычислительной техники

Кафедра АСУ



**Лабораторная работа №2**

по дисциплине: «Параллельное программирование»

«Метод численного интегрирования на языке С++ OpenMP»

Группа: АВТ-813

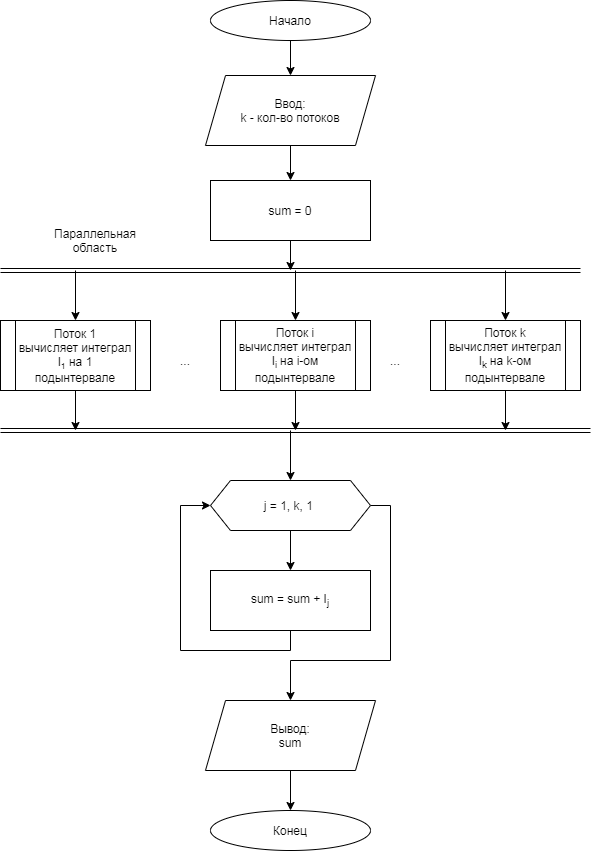
Преподаватель: Владимир Владимирович Ландовский

Студенты: Самсонов Николай, Смирнов Кирилл, Юксеев Михаил.

Новосибирск

2020

**Описание алгоритма**



**Текст программы**

#include <iostream>

#include <ctime>

#include <math.h>

#include "omp.h"

using namespace std;

double fun(double x);

int main() {

setlocale(0, "");

int k;

double a = 0, b = 0;

int n = 0;

double sum = 0;

cout << "Введите число потоков: ";

cin >> k;

cout << "a: ";

cin >> a;

cout << "b: ";

cin >> b;

cout << "n: ";

cin >> n;

double h = (b - a) / n;

unsigned int start\_time = clock(); // начальное время

#pragma omp parallel for num\_threads(k) reduction (+: sum)

for (int i = 1; i <= n; ++i)

sum += fun(a + i \* h) \* h;

unsigned int end\_time = clock(); // конечное время

cout << "result: " << sum << endl;

cout << "time: " << (end\_time - start\_time) << endl;

return 0;

}

double fun(double x) {

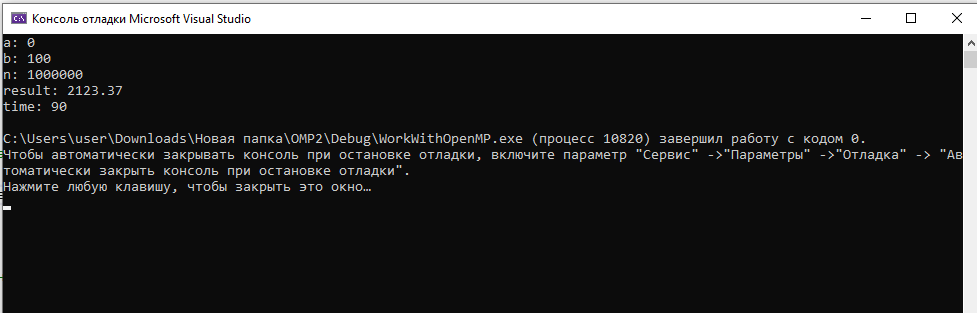
return (cos(2 \* x \* x) \* sin(x) + log(x \* x \* x)) - (cos(2 \* x \* x) / sin(x) - log(x \* x \* x));

}

**Пример работы программы**

Функция: (cos(2\*x\*x)\*sin(x)+log(x\*x\*x))- (cos(2 \* x \* x)/ sin(x) - log(x \* x \* x)).

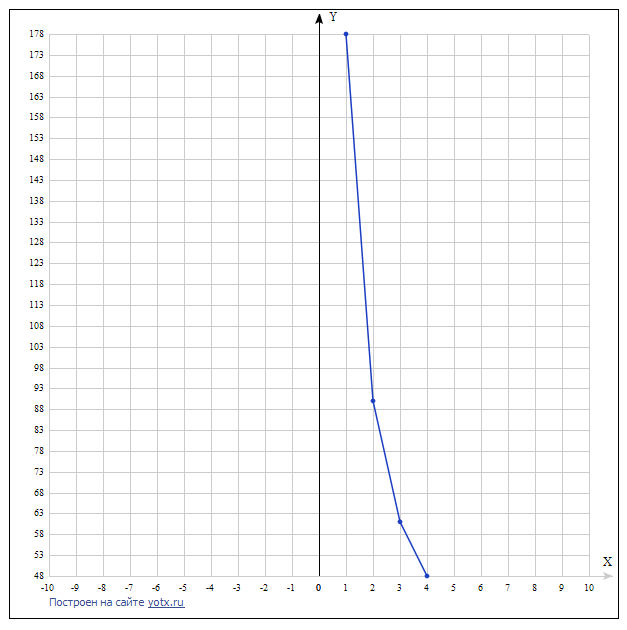
Задаем пределы интегрирования от 0 до 100, количество подынтервалов 1000000.



В результате получили интеграл равный 2123.37 и затраченное время 90 мс.

**Результаты экспериментов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | n | Число потоков | Затраченное время (мс) | Значение интеграла |
| 0 | 100 | 1000000 | 1 | 178 | 2123.37 |
| 0 | 100 | 1000000 | 2 | 90 | 2123.37 |
| 0 | 100 | 1000000 | 3 | 61 | 2123.37 |
| 0 | 100 | 1000000 | 4 | 48 | 2123.37 |



Комментарии к графику: X – число ядер, Y – время в миллисекундах.

**Вывод:**

Таким образом, мы увидели, что OpenMP – удобный инструмент для работы с потоками. С увеличением кол-ва потоков уменьшается время выполнения программы, однако уменьшается оно не равномерно.