

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ, ФОРМУЛА СИМПСОНА,
ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННЫХ И КРАТНЫХ ИНТЕГРАЛОВ**

ОТЧЕТ О ПРАКТИКЕ

Студента 3 курса 311 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Аношкина Андрея Алексеевича

Проверил

Старший преподаватель

М. С. Портенко

СОДЕРЖАНИЕ

1 Work 01 3

1 Work 01

Задание

Реализуйте параллельные алгоритмы, использующие метод прямоугольников и формулу Симпсона для подсчета интегралов. Точные значения интегралов указаны для проверки численных вычислений. В случае, если в верхнем пределе интегрирования указан знак бесконечности, то в расчете необходимо заменить его на 10^6 . Сравните время численного интегрирования для последовательной и параллельной реализации. Какое ускорение выполнения программы предоставляет переход к многопоточной версии?

Вариант задания 2:

$$\int_0^{\infty} \frac{dx}{1+x^2} = \frac{\pi}{2}$$

Метод прямоугольников, формула Симпсона

Метод прямоугольников геометрически заключается в том, что интеграл приближенно представляется в виде суммы площадей элементарных прямоугольников.

Для случая деления отрезка интегрирования на равные части и вычисления функции в центре отрезков:

$$\begin{cases} J = \int_a^b f(x) dx \approx \sum_{i=0}^{N-1} (h \cdot f(x_i)) = h \cdot \sum_{i=0}^{N-1} f(x_i) \\ x_i = a + i \cdot h + \frac{h}{2} \end{cases} \quad (1)$$

где N — количество отрезков интегрирования, а $h = (b - a)/N$

Ступенчатая (stair-case) аппроксимация гладких изогнутых поверхностей, возникает в результате дискретизации модели прямоугольной сеткой. Для борьбы со ступенчатой аппроксимацией может использоваться, например, квадратурная формула Симпсона:

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(a) + f(b) + 4 \sum_{k=1}^N f(a + (2k-1)h) + 2 \sum_{k=1}^{N-1} f(a + 2kh)] \quad (2)$$

где $h = \frac{b-a}{2N}$, $N \gg 1$.

Формула Симпсона геометрически заключается в том, что через три ординаты, отвечающие трем последовательным узлам сетки, проводится парабола

и затем складываются получившиеся при этом площади элементарных криволинейных трапеций.

Реализация

Код решения приведен ниже:

```
1  #include <iostream>
2  #include <omp.h>
3  #include <time.h>
4  #define PI 3.1415926535897932384626433832795
5
6  using namespace std;
7
8  double f1(double x) {
9      return 1.0 / (1 + x * x);
10 }
11
12 void integral_posl(const double a, const double b, const double h, double* res) {
13     double sum = 0;
14     int n = (int)((b - a) / h);
15
16     for (int i = 0; i < n; ++i) {
17         double x = a + i * h + h / 2;
18         sum += f1(x) * h;
19     }
20
21     *res = sum;
22 }
23
24 void integral_parallel(const double a, const double b, const double h, double* res) {
25     double sum = 0;
26     int n = (int)((b - a) / h);
27
28     #pragma omp parallel for reduction(+: sum)
29     for (int i = 0; i < n; ++i) {
30         double x = a + i * h + h / 2;
31         sum += f1(x) * h;
32     }
33
34     *res = sum;
35 }
36
37 void integral_Simpson(const double a, const double b, const double h, double* res) {
38     double sum = f1(a) + f1(b);
39     int n = (int)((b - a) / 2 * h);
40
41     #pragma omp parallel for reduction(+: sum)
42     for (int i = 1; i <= n; ++i) {
43         double x = a + h * (2 * i - 1);
```

```

44         sum += f1(x) * 4;
45     }
46
47     #pragma omp parallel for reduction(+: sum)
48     for (int i = 1; i < n; ++i) {
49         double x = a + 2 * i * h;
50         sum += 2 * f1(x);
51     }
52
53     *res = h / 3 * sum;
54 }
55
56 double experiment(double* res, void f(const double a, const double b, const double h, double* res)) {
57     double stime = 0, ftime = 0;
58     double a = 0;
59     double b = 1e+6;
60     double h = 0.01;
61     stime = clock();
62     f(a, b, h, res);
63     ftime = clock();
64
65     return (ftime - stime) / CLOCKS_PER_SEC;
66 }
67
68 void calculate(void f(const double a, const double b, const double h, double* res)) {
69     double avg_time = 0;
70     double min_time = 0;
71     double max_time = 0;
72     double res = 0;
73     int numbExp = 10;
74
75     min_time = max_time = experiment(&res, f);
76     avg_time = min_time / numbExp;
77     for (int i = 0; i < numbExp - 1; ++i) {
78         double time = experiment(&res, f);
79         if (time > max_time)
80             max_time = time;
81         if (time < min_time)
82             min_time = time;
83
84         avg_time += time / numbExp;
85     }
86
87     cout << "Execution time: " << avg_time << "; " << min_time << "; " << max_time << "\n";
88     cout.precision(8);
89     cout << "Integral value: " << res << "\n";
90 }
91
92 int main() {
93

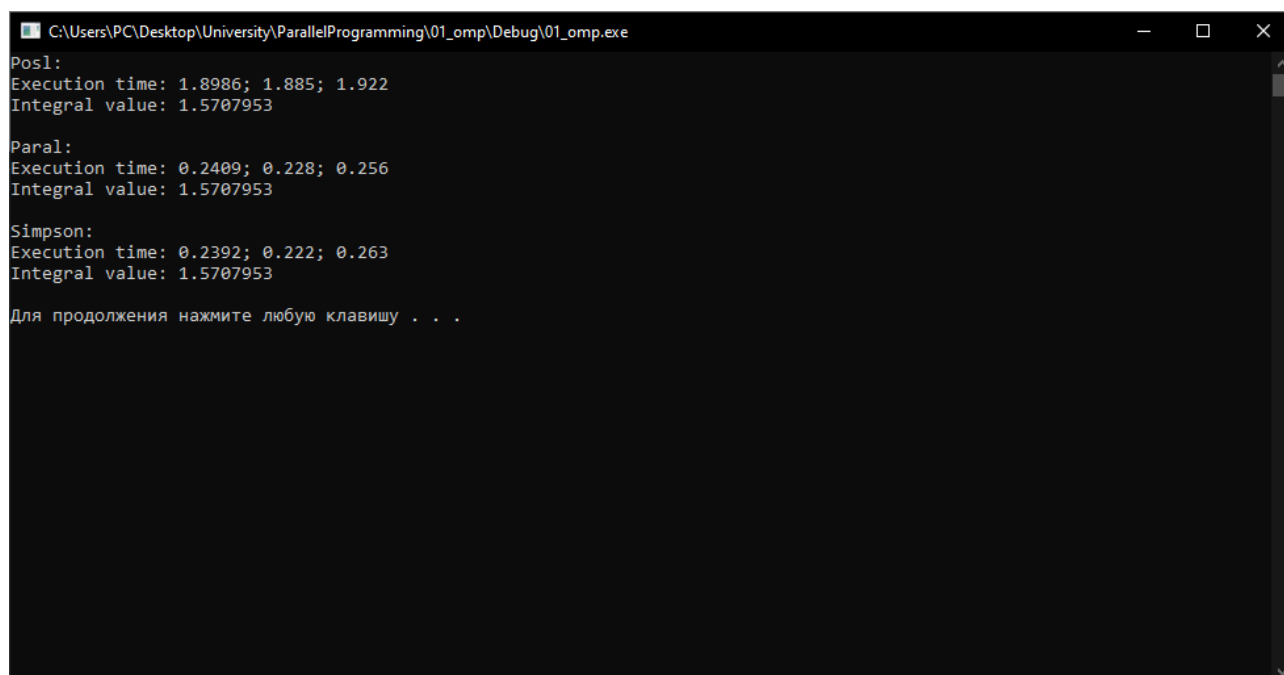
```

```

94     bool calcPosl = true, calcParal = true, calcSimpson = true;
95
96     // Последовательное вычисление
97
98     if (calcPosl) {
99         cout << "Posl:\n";
100        calculate(integral_posl);
101        cout << "\n";
102    }
103
104    // Параллельное вычисление
105
106    if (calcParal) {
107        cout << "Paral:\n";
108        calculate(integral_paral);
109        cout << "\n";
110    }
111
112    // Параллельное вычисление по формуле Симпсона
113
114    if (calcSimpson) {
115        cout << "Simpson:\n";
116        calculate(integral_Simpson);
117        cout << "\n";
118    }
119
120    system("pause");
121    return 0;
122 }

```

Результаты работы



```
C:\Users\PC\Desktop\University\ParallelProgramming\01_omp\Debug\01_omp.exe
Posl:
Execution time: 1.8986; 1.885; 1.922
Integral value: 1.5707953

Paral:
Execution time: 0.2409; 0.228; 0.256
Integral value: 1.5707953

Simpson:
Execution time: 0.2392; 0.222; 0.263
Integral value: 1.5707953

Для продолжения нажмите любую клавишу . . .
```

Рисунок 1 – Work-1

Значение ускорения выполнения программы при переходе к многопоточной версии:

$$a = \frac{timeSeq}{timePar} = \frac{1.8986}{0.2409} \approx 7.8812$$

Значение ускорения выполнения программы при переходе к многопоточной версии по формуле Симпсона:

$$a = \frac{timeSeq}{timePar} = \frac{1.8986}{0.2392} \approx 7.9373$$

Характеристики устройства

Процессор: Intel(R) Core(TM) i5-10400F

Ядер: 6

Оперативная память: 16 Гб