

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

Саратов 2024

## СОДЕРЖАНИЕ

1	Work 12.....	3
---	--------------	---

## 1 Work 12

### Задание

Реализовать параллельно один из методов приближенного вычисления двойных интегралов, а именно метод статистических испытаний.

### Метод статистических испытаний

Метод статистических испытаний основан на генерации случайных точек  $x, y$ , принадлежащих области интегрирования функции. Алгоритм вычисляет значения функции в каждой точке  $(x, y)$  и суммирует их. С помощью отношения полученной суммы к количеству всех точек можно определить величину двойного интеграла как площадь области интегрирования умноженная на полученный коэффициент.

$$J = \int_{a_1}^{b_1} \int_{a_2}^{b_2} f(x, y) dx dy \approx (b_1 - a_1)(b_2 - a_2) \cdot \frac{1}{N} \sum_{i=0}^N f(x, y) \quad (1)$$

### Реализация

Фрагмент кода решения приведен ниже:

```
1  #include <iostream>
2  #include "mpi.h"
3  #include <time.h>
4  #include <cmath>
5  #define PI 3.1415926535897932384626433832795
6
7  using namespace std;
8
9  int NProc, ProcId;
10 MPI_Status st;
11
12 double f1(const double x, const double y, const double a1, const double b1, const double a2, const double b2) {
13     return (1 + exp(sin(PI * x) * cos(PI * y))) / (b1 - a1) / (b2 - a2);
14 }
15
16 void integral(const double a1, const double b1, const double a2, const double b2, const double h1, const
↵ double h2, double* res) {
17     double delta = (b2 - a2) * (b1 - a1);
18
19     double sum = 0;
20     int n = (int)((b1 - a1) / h1);
21
```

```

22     for (int i = ProcId; i < n; i += NProc) {
23         double x = a1 + (b1 - a1) * rand() / (RAND_MAX + 1.0);
24         double y = a2 + (b2 - a2) * rand() / (RAND_MAX + 1.0);
25         sum += f1(x, y, a1, b1, a2, b2);
26     }
27
28     if (ProcId != 0)
29         MPI_Send(&sum, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
30     else {
31         for (int i = 1; i < NProc; ++i) {
32             double s;
33             MPI_Recv(&s, 1, MPI_DOUBLE, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD,
34                 ↪ &st);
35             sum += s;
36         }
37         *res = delta * sum / n;
38     }
39 }
40
41 double experiment(double* res, void f(const double a1, const double b1, const double a2, const double b2,
42     ↪ const double h1, const double h2, double* res)) {
43     double stime = 0, ftime = 0;
44     double a1 = 0, b1 = 16;
45     double a2 = 0, b2 = 16;
46     double h1 = 0.000001, h2 = 0.000001;
47     stime = clock();
48     f(a1, b1, a2, b2, h1, h2, res);
49     ftime = clock();
50     return (ftime - stime) / CLOCKS_PER_SEC;
51 }
52
53 void calculate(void f(const double a1, const double b1, const double a2, const double b2, const double h1,
54     ↪ const double h2, double* res)) {
55     double avg_time = 0;
56     double min_time = 0;
57     double max_time = 0;
58     double res = 0;
59     int numbExp = 10;
60
61     min_time = max_time = experiment(&res, f);
62     avg_time = min_time / numbExp;
63     for (int i = 0; i < numbExp - 1; ++i) {
64         double time = experiment(&res, f);
65         if (time > max_time)
66             max_time = time;
67         if (time < min_time)
68             min_time = time;

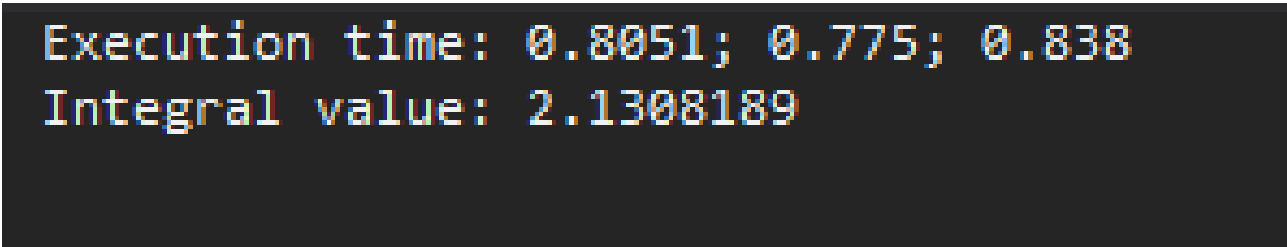
```

```

69         avg_time += time / numbExp;
70     }
71
72     if (ProcId == 0) {
73         cout << "Execution time: " << avg_time << "; " << min_time << "; " << max_time <<
74             << "\n";
75         cout.precision(8);
76         cout << "Integral value: " << res << "\n";
77     }
78 }
79
80 int main() {
81     MPI_Init(NULL, NULL);
82     MPI_Comm_size(MPI_COMM_WORLD, &NProc);
83     MPI_Comm_rank(MPI_COMM_WORLD, &ProcId);
84
85     calculate(integral);
86     cout << "\n";
87
88     MPI_Finalize();
89     return 0;
90 }

```

## Результат работы



```

Execution time: 0.8051; 0.775; 0.838
Integral value: 2.1308189

```

Рисунок 1 – Work-12

## Характеристики устройства

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

Ядер: 6

Оперативная память: 16 Гб