



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра исследования операций

Численное интегрирование многомерных функций методом Монте-Карло

Второе задание по курсу «Суперкомпьютерное моделирование и
технологии»

Выполнил:
студент 614 группы
Грузицкий М.А.
Вариант 5

Москва 2021

1 Математическая постановка задачи

Функция $f(x, y, z)$ — непрерывна в ограниченной замкнутой области $G \subset \mathbb{R}^3$. Требуется вычислить определённый интеграл:

$$I = \iiint_G f(x, y, z) dx dy dz.$$

2 Численный метод решения задачи

Пусть область G ограничена параллелепипедом Π :
$$\begin{cases} a_1 \leq x \leq b_1 \\ a_2 \leq y \leq b_2 \\ a_3 \leq z \leq b_3 \end{cases}$$

Рассмотрим функцию:
$$F(x, y, z) = \begin{cases} f(x, y, z), & (x, y, z) \in G \\ 0, & (x, y, z) \notin G \end{cases}$$

Преобразуем искомый интеграл:

$$I = \iiint_G f(x, y, z) dx dy dz = \iiint_{\Pi} F(x, y, z) dx dy dz.$$

Пусть $p_1(x_1, y_1, z_1), p_2(x_2, y_2, z_2), \dots$ — случайные точки, равномерно распределённые в Π . Возьмём n таких случайных точек. В качестве приближённого значения интеграла предлагается использовать выражение:

$$I \approx |\Pi| \cdot \frac{1}{n} \sum_{i=1}^n F(p_i),$$

где $|\Pi|$ — объем параллелепипеда Π . $|\Pi| = (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)$.

3 Нахождение точного значения интеграла аналитически

$$I = \iiint_G x^3 y^2 z dx dy dz,$$

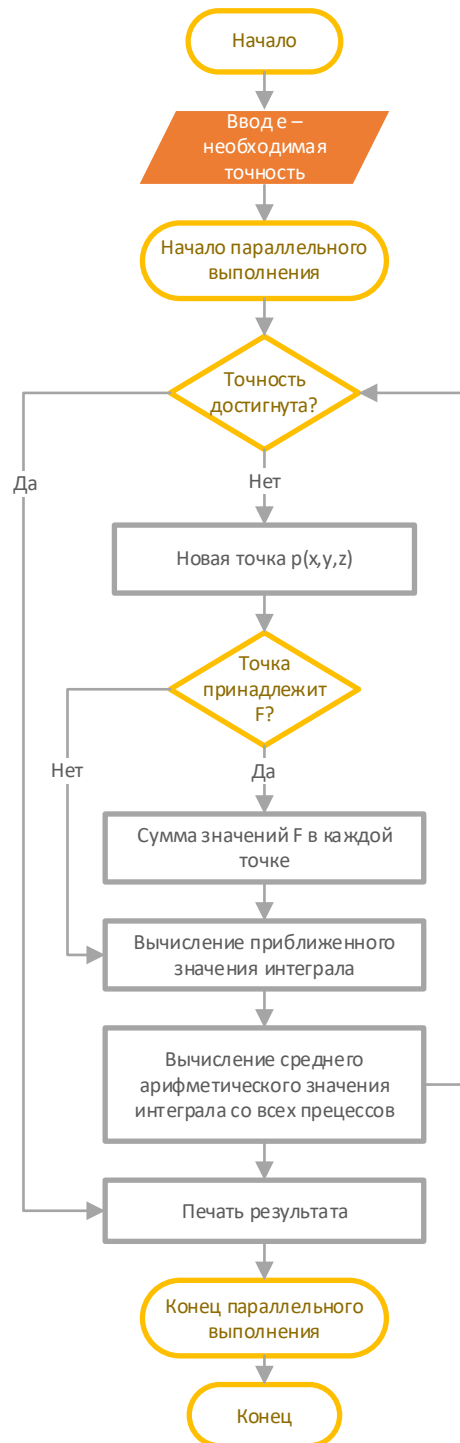
где область $G = \{(x, y, z) : -1 \leq x \leq 0, -1 \leq y \leq 0, -1 \leq z \leq 0\}$.

$$\begin{aligned}
I &= \iiint_G x^3 y^2 z \, dx dy dz \\
&= \int_{-1}^0 x^3 dx \int_{-1}^0 y^2 dy \int_{-1}^0 z dz \\
&= \frac{z^2}{2} \Big|_{-1}^0 \cdot \int_{-1}^0 x^3 dx \int_{-1}^0 y^2 dy = \left(0 - \frac{1}{2}\right) \int_{-1}^0 x^3 dx \int_{-1}^0 y^2 dy = \\
&= -\frac{1}{2} \int_{-1}^0 x^3 dx \int_{-1}^0 y^2 dy = -\frac{1}{6} \cdot y^3 \Big|_{-1}^0 \\
&= -\frac{1}{6} \int_{-1}^0 x^3 dx = -\frac{1}{6} (0 + 1) \int_{-1}^0 x^3 dx = -\frac{1}{24} x^4 \Big|_{-1}^0 \\
&= -\frac{1}{24} (0 - 1) = \frac{1}{24} .
\end{aligned}$$

Точное значение интеграла — $\frac{1}{24} \approx 0.041666667$.

4 Программная реализация

Описание работы программы. Реализована независимая генерация точек MPI-процессами.



5 Исследование масштабируемости программы

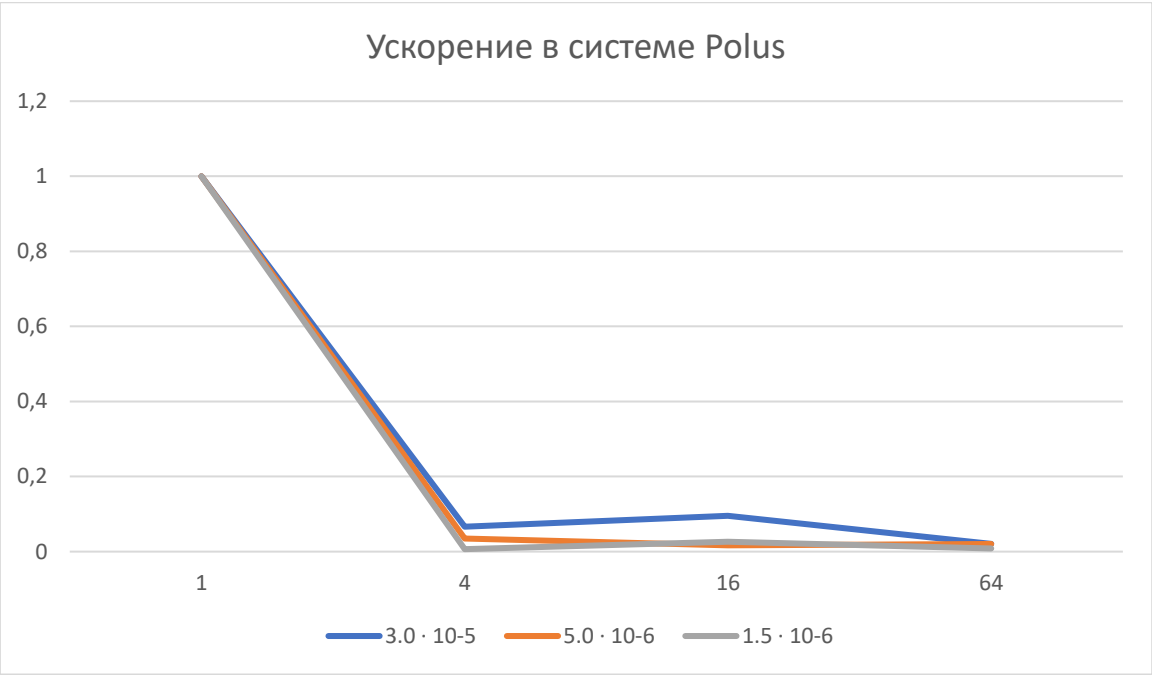
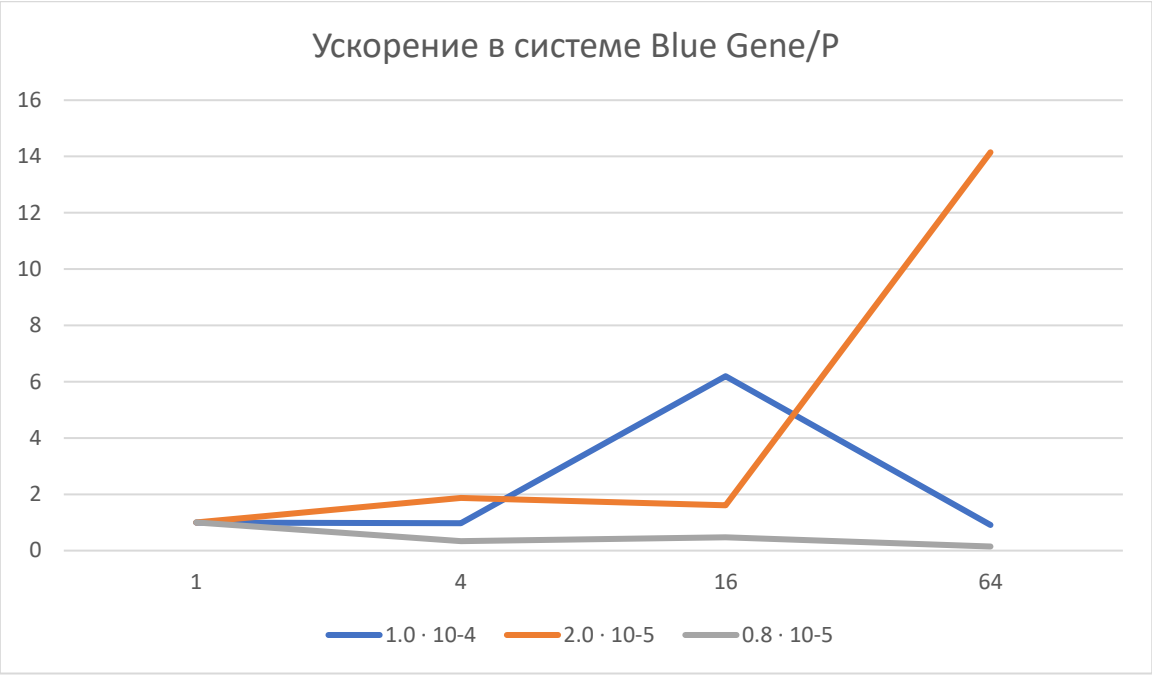
Исследование масштабируемости программы на системах Blue Gene/P и Polus для различного числа MPI-процессов и различных значений входного параметра ε .

Таблица 1. Результаты расчетов для системы Blue Gene/P

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$1.0 \cdot 10^{-4}$	1	0.003395	1	0.000050
	4	0.003480	0.975575	0.000031
	16	0.000548	6.195255	0.000036
	64	0.003731	0.909944	0.000096
$2.0 \cdot 10^{-5}$	1	0.009692	1	0.000018
	4	0.005186	1.868878	0.000016
	16	0.006028	1.607830	0.000016
	64	0.000685	14.148905	0.000002
$0.8 \cdot 10^{-5}$	1	0.001409	1	0.000006
	4	0.004125	0.341575	0.000001
	16	0.002953	0.477141	0.000006
	64	0.009673	0.145663	0.000004

Таблица 2. Результаты расчетов для системы Polus

Точность ε	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	1	0.000248	1	0.000010
	4	0.003742	0.066275	0.000026
	16	0.002604	0.095238	0.000011
	64	0.012016	0.020639	0.000004
$5.0 \cdot 10^{-6}$	1	0.000063	1	0.000002
	4	0.001810	0.034807	0.000005
	16	0.003765	0.016733	0.000002
	64	0.003162	0.019924	0.000002
$1.5 \cdot 10^{-6}$	1	0.000107	1	0.000001
	4	0.016426	0.006514	0.000000
	16	0.004074	0.026264	0.000001
	64	0.012400	0.008629	0.000001



Приложение

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#include <time.h>
#include <mpi.h>
#include <stdlib.h>

/*
-1 <= x <= 0
-1 <= y <= 0
-1 <= z <= 0
*/

int n_points = 1;

double F(double x,double y,double z)
{
    return x*x*x*y*y*z;
}

double double_rand( double a, double b)
{
    double scale = ((double)rand()) / ( (double) RAND_MAX);
    return a + scale * (b - a);
}

double modul(int a1, int b1, int a2, int b2, int a3, int b3)
{
    return (b1 - a1) * (b2 - a2) * (b3 - a3);
}

int main(int argc, char** argv)
{
    /* ID процесса */
    int rank;
    /* Общее число процессов */
    int size;
    double a1 = -1;
    double b1 = 0;
    double a2 = -1;
    double b2 = 0;
    double a3 = -2;
```

```

double b3 = 0;
double x, y, z;
double res = 0;
double res1 = 0.041666667; //точное значение вычисленное аналитически
double sum = 0;
double eTime, sTime, pTime, maxTime;
bool check = true;
double localSum = 0 ;
double local_ans = 0;

double e = atof(argv[1]);

MPI_Init(&argc, &argv); // начало параллельного выполнения
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);

sTime = MPI_Wtime();

srand(time(NULL)+rank);

while(check)
{
    //printf("I am %d of %d\n", rank, size-1);
    x = double_rand(a1, b1);
    y = double_rand(a2, b2);
    z = double_rand(a3, b3);

    //printf("n = %d\n", n_points);
    //printf("x = %f, y = %f, z = %f\n", x, y, z);

    if(z>-1 && z<0)
    {
        localSum += F(x, y, z);
        //printf("sum = %f\n", localSum);
    }
    local_ans = localSum / n_points * modul(a1, b1, a2, b2, a3, b3);
    MPI_Reduce(&local_ans,&sum,1,MPI_DOUBLE,
MPI_SUM,0,MPI_COMM_WORLD);

    if(rank == 0){
        res = sum/size;
        check = fabs(res - res1) > e;
        //printf("sum = %f, res = %f, e = %f\n", sum, res, e);
    }
    n_points++;
}

```



```
//printf("n = %d\n", n_points);
MPI_Bcast(&check, 1, MPI_INT, 0, MPI_COMM_WORLD);
}

eTime = MPI_Wtime();
MPI_Reduce(&eTime, &maxTime, 1, MPI_DOUBLE, MPI_MAX, 0,
MPI_COMM_WORLD);
pTime = maxTime - sTime;

if (rank == 0)
{
    printf("Result = %f\n", res);
    printf("Mistake = %f\n", fabs(res - res1));
    printf("Points = %d\n", (n_points-1));
    printf("Time = %f\n", pTime);
}

MPI_Finalize();
return 0;
}
```