



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

Фазлетдинов Рамиль Рустамович, 608

**ЗАДАНИЕ ПО КУРСУ «СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И  
ТЕХНОЛОГИИ» ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МНОГОМЕРНЫХ  
ФУНКЦИЙ МЕТОДОМ МОНТЕ-КАРЛО**

ВАРИАНТ 11

Москва  
2022

# Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>3</b>
1.1	Введение . . . . .	3
1.2	Математическая постановка . . . . .	3
1.3	Численный метод решения задачи . . . . .	3
1.3.1	Описание требований к программной реализации . . . . .	4
1.4	Спецификация варианта . . . . .	4
<b>2</b>	<b>Ход работы</b>	<b>4</b>
2.1	Аналитическое решение . . . . .	4
2.2	Краткое описание программной реализации . . . . .	4
2.3	Исследование масштабируемости . . . . .	5

# 1 Постановка задачи

## 1.1 Введение

В качестве модельной задачи предлагается задача вычисления многомерного интеграла методом Монте-Карло. Программная реализация должна быть выполнена на языке С или С++ с использованием библиотеки параллельного программирования MPI. Требуется исследовать масштабируемость параллельной MPI-программы на следующих параллельных вычислительных системах ВМК МГУ:

1. IBM Polus

## 1.2 Математическая постановка

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

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

## 1.3 Численный метод решения задачи

Пусть область  $G$  ограничена параллелепипедом

$$\Pi : \begin{cases} a_1 \leq x \leq b_1, \\ a_2 \leq y \leq b_2, \\ a_3 \leq z \leq b_3 \end{cases}$$

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  $\Pi$ . 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  $|\Pi|$  - объём параллелепипеда  $\Pi$ .  $|\Pi| = (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)$

### 1.3.1 Описание требований к программной реализации

Параллельная MPI-программа принимает на вход требуемую точность и генерирует случайные точки до тех пор, пока требуемая точность не будет достигнута. Программа вычисляет точность как модуль разности между приближённым значением, полученным методом Монте-Карло, и точным значением, вычисленным аналитически.

Программа считывает в качестве аргумента командной строки требуемую точность  $\epsilon$  и выводит четыре числа:

- Посчитанное приближённое значение интеграла
- Ошибка посчитанного значения: модуль разности между приближённым и точным значениями интеграла
- Количество сгенерированных случайных точек
- Время работы программы в секундах

Время работы программы измеряется следующим образом. Каждый MPI-процесс измеряет своё время выполнения, затем среди полученных значений берётся максимум.

## 1.4 Спецификация варианта

Необходимо выполнить задачу в парадигме «мастер–рабочие»: один из процессов («мастер») генерирует случайные точки и передаёт каждому из остальных процессов («рабочих») отдельный, предназначенный для него, набор сгенерированных случайных точек.

Вариант интеграла:

$$\iiint_G \sqrt{x^2 + y^2} \, dx dy dz,$$

где область  $G$  ограничена поверхностями  $x^2 + y^2 = z^2$ ,  $z = 1$ .

## 2 Ход работы

### 2.1 Аналитическое решение

$$\begin{aligned} \iiint_G \sqrt{x^2 + y^2} \, dx dy dz &= \iint_Q dx dy \int_{\sqrt{x^2 + y^2}}^1 \sqrt{x^2 + y^2} \, dz = \\ \iint_{x^2 + y^2 \leq 1} (\sqrt{x^2 + y^2} - (x^2 + y^2)) \, dx dy &= \int_0^{2\pi} d\varphi \int_0^1 (r - r^2) r \, dr = 2\pi \left( \frac{r^3}{3} - \frac{r^4}{4} \right) \Big|_0^1 = 2\pi \left( \frac{1}{3} - \frac{1}{4} \right) = \frac{\pi}{6} \end{aligned}$$

### 2.2 Краткое описание программной реализации

В данной реализации мастер на каждом шаге генерирует  $(\text{comm\_size} - 1) * \text{block}$  точек, где  $\text{block} = \text{dots\_each\_iter} / \text{comm\_size}$ ,  $\text{dots\_each\_iter}$  - количество троек  $(x, y, z)$  равное 1024,

comm\_size - количество процессов. Затем используется функция Scatter для того, чтобы каждый процесс забрал в свой буфер  $3 * \text{block}$  точек. После подсчёта каждым процессом своей суммы происходит операция редукции, результат которой анализирует мастер для сравнения с заданной точностью. Нахождение времени работы основывается на поиске максимума затраченного времени с момента получения точек до окончания подсчётов среди slave-процессов.

## 2.3 Исследование масштабируемости

Таблица 1: Таблица с результатами расчётов для системы Polus, seed = 758631

Точность $\varepsilon$	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	2	0.0708848	1	2.84684e-05
	4	0.0247464	2.8644	2.62617e-05
	8	0.0124932	5.6738	2.97089e-05
	16	0.00992493	12.0936	2.97089e-05
$5.0 \cdot 10^{-6}$	2	0.073383	1	4.40623e-06
	4	0.0238563	3.076	3.79917e-06
	8	0.0122959	5.968	4.66277e-06
	16	0.00571301	12.5198	3.06944e-06
$1.5 \cdot 10^{-6}$	2	0.0746809	1	5.52963e-07
	4	0.026404	2.8283	5.91808e-07
	8	0.0124416	6.0025	6.53105e-07
	16	0.00586133	12.7412	8.00831e-07

Таблица 2: Таблица с результатами расчётов для системы Polus с разными зёрнами генерации при точности  $5.0 \cdot 10^{-6}$

Зерно	Количество точек	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
173	50176	2	0.000674322	1	2.94507e-07
		4	0.000600296	1.1233	3.26349e-06
		8	5.4298e-05	12.4189	8.98286e-07
		16	3.67229e-05	18.3624	8.98286e-07
1731	22016	2	0.000426901	1	2.94507e-07
		4	0.000559807	0.7625	3.26349e-06
		8	0.000104399	4.0891	8.98286e-07
		16	6.28071e-05	6.7970	4.29615e-06
17311	385024	2	0.00491566	1	2.94507e-07
		4	0.00188009	2.6145	3.26349e-06
		8	0.00102172	4.8111	8.98286e-07
		16	8.21953e-05	5.9804	4.29615e-06

Графики находятся на следующей странице.

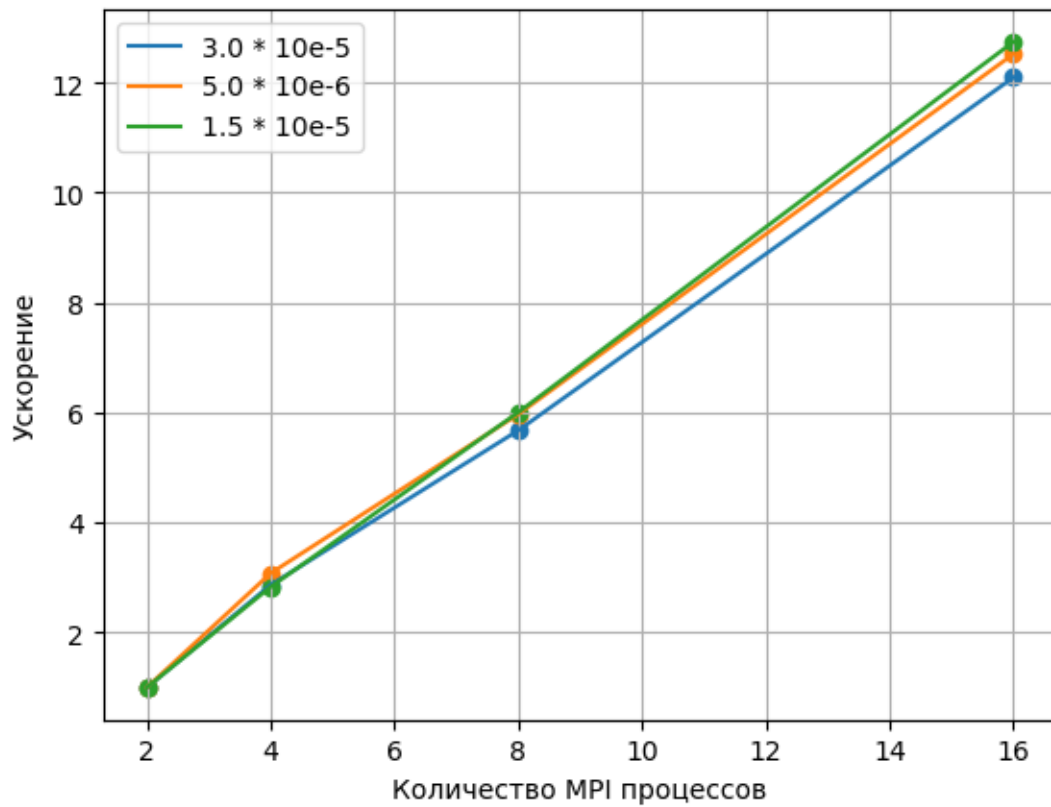


Рис. 1: Зависимость ускорения от количества процессов (seed = 758631)

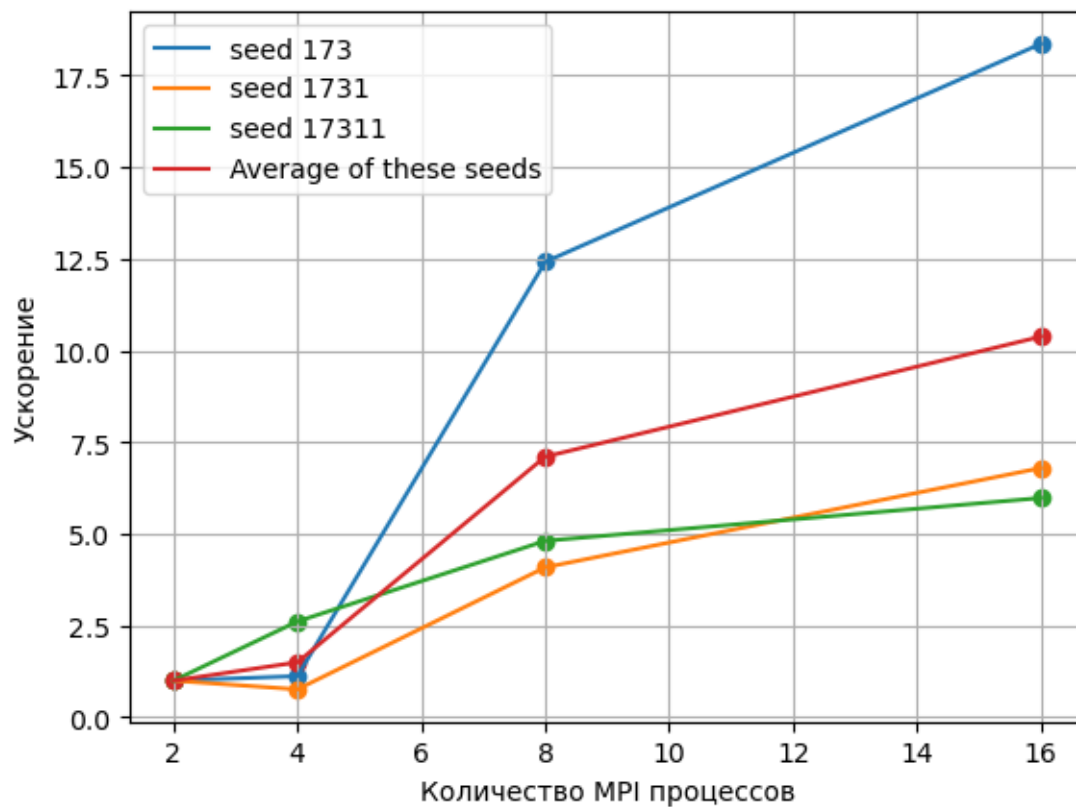


Рис. 2: Зависимость ускорения при различных зернах и фиксированной точности  $5.0 \cdot 10^{-6}$