



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

Павлишин Кирилл Юрьевич

608 группа

15 вариант

**ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МНОГОМЕРНЫХ ФУНКЦИЙ МЕТОДОМ
МОНТЕ-КАРЛО**

Москва
2022

1 Введение

В качестве модельной задачи предлагается задача вычисления многомерного интеграла методом Монте-Карло.

Программная реализация должна быть выполнена на языке C или C++ с использованием библиотеки параллельного программирования MPI.

Требуется исследовать масштабируемость параллельной MPI-программы на параллельной вычислительной системе ВМК МГУ IBM Polus.

2 Математическая постановка задачи

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

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

где функция $f(x, y, z) = \sqrt{y^2 + z^2}$,

область $G = \{(x, y, z) : 0 \leq x \leq 2, y^2 + z^2 \leq 1\}$

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

4 Нахождение точного значения интеграла аналитически

$$I = \iiint_G \sqrt{y^2 + z^2} dx dy dz = \{x = x, y = r \cos \phi, z = r \sin \phi\} = \int_0^2 dx \int_0^{2\pi} d\phi \int_0^1 r^2 dr = \frac{4\pi}{3}$$

5 Описание 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 ϵ 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.

В моём варианте реализована парадигма «мастер–рабочие»: один из процессов («мастер») генерирует по очереди для каждого из остальных процессов («рабочих») $n / (\text{количество процессов} - 1)$ случайных точек и передаёт их. Все процессы–рабочие получают свой набор точек и вычисляют свою часть суммы. Затем в процессе-мастере вычисляется общая сумма с помощью операции редукции. После чего в «мастере» вычисляется ошибка (разность между посчитанным значением и точным значением, вычисленным аналитически). В случае если ошибка выше требуемой точности, которую подали на вход программе, то генерируются дополнительные $n / (\text{количество процессов} - 1)$ точек для каждого процесса-рабочего и расчёт продолжается.

6 Исследование масштабируемости программы на системе Polus

Для корректного исследования масштабируемости программы было выбрано $n = 16740$ и зерно для генератора случайных чисел, равное 1, одинаковые для всевозможных комбинаций ϵ и числа MPI-процессов.

Таблица 1: Таблица с результатами расчётов для системы Polus

Точность ϵ	Число MPI-процессов	Время работы программы (с)	Ускорение	Ошибка
$3.0 \cdot 10^{-5}$	2	0.0309413	1	1.99304e-05
	4	0.0282031	1.0970886	1.99304e-05
	16	0.0273628	1.1307797	1.99304e-05
	32	0.0329068	0.94027	1.99304e-05
$5.0 \cdot 10^{-6}$	2	0.244562	1	4.1148e-06
	4	0.221029	1.10647	4.1148e-06
	16	0.207898	1.1763557	4.1148e-06
	32	0.32678	0.7483995	4.1148e-06
$1.5 \cdot 10^{-6}$	2	11.2653	1	1.38048e-06
	4	10.2372	1.10042785	1.38048e-06
	16	9.38869	1.19987985	1.38048e-06
	32	12.0932	0.93154	1.38048e-06

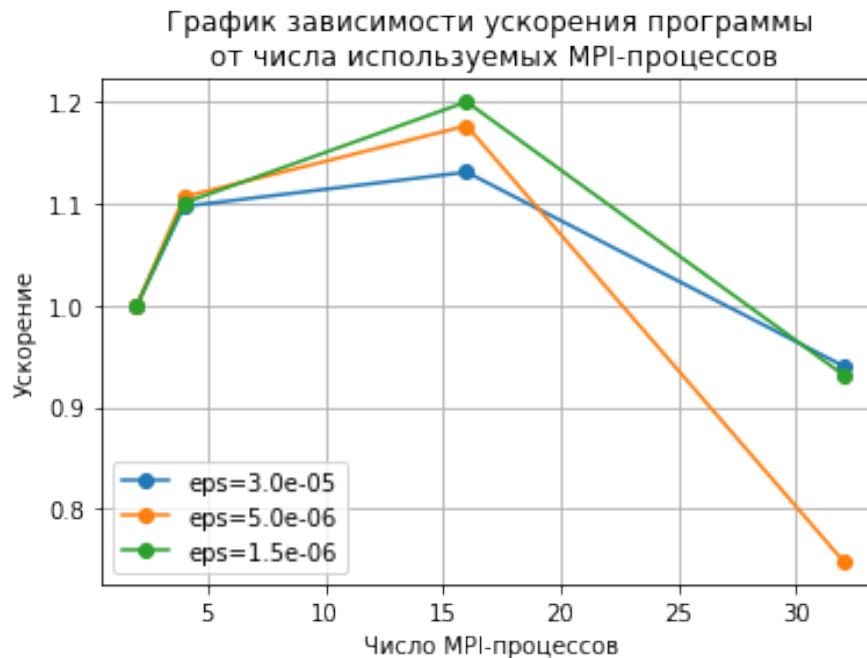


Рис. 1: График зависимости ускорения программы от числа используемых MPI-процессов для каждого значения ϵ