



Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Вершков Станислав Александрович

Задание по курсу «Суперкомпьютерное  
моделирование и технологии»

Численное интегрирование многомерных  
функций методом Монте-Карло

ОТЧЕТ

Москва, 2022

# Оглавление

1	Математическая постановка задачи	2
2	Численный метод решения задачи	2
3	Аналитическое решение	3
4	Описание алгоритма	3
5	Исследование масштабируемости программы	4

# 1 Математическая постановка задачи

Функция  $f(y, z) = \sqrt{y^2 + z^2}$  – непрерывна в ограниченной области замкнутой области  $G \subset \mathbb{R}^3$ , где  $G = \{(x, y, z) : 0 \leq x \leq 2, y^2 + z^2 \leq 1\}$ . Требуется вычислить определённый интеграл:

$$I = \iiint_G \sqrt{y^2 + z^2} dx dy dz.$$

# 2 Численный метод решения задачи

Преобразуем искомый интеграл:

$$I = \iiint_G \sqrt{y^2 + z^2} dx dy dz = 2 \iint_{\{(y,z) : y^2 + z^2 \leq 1\}} \sqrt{y^2 + z^2} dy dz = \\ 2 \cdot 4 \iint_{\{(y,z) : y^2 + z^2 \leq 1, y \geq 0, z \geq 0\}} \sqrt{y^2 + z^2} dy dz = 8 \iint_{\{(y,z) : 0 \leq y \leq 1, 0 \leq z \leq 1\}} F(y, z) dy dz,$$

где

$$F(y, z) = \begin{cases} \sqrt{y^2 + z^2}, & (y, z) \in \{(y, z) : y^2 + z^2 \leq 1\}, \\ 0, & (y, z) \notin \{(y, z) : y^2 + z^2 \leq 1\}. \end{cases}$$

Пусть  $p_1(y_1, z_1), p_2(y_2, z_2), \dots$ , – случайные точки, равномерно распределенные в  $\{(y, z) : 0 \leq y \leq 1, 0 \leq z \leq 1\}$ . Возьмем  $n$  случайных точек. В качестве приближенного значения интеграла используется выражение:

$$I \approx \frac{8}{n} \sum_{i=1}^n F(p_i). \quad (1)$$

### 3 Аналитическое решение

Найдем точное решение интеграла аналитически:

$$\begin{aligned} I &= \iiint_G \sqrt{y^2 + z^2} \, dx \, dy \, dz = 2 \iint_{\{(y,z) : y^2+z^2 \leq 1\}} \sqrt{y^2 + z^2} \, dy \, dz = \\ &= 2 \iint_{\{(r,\phi) : 0 \leq r \leq 1, 0 \leq \phi \leq 2\pi\}} r \sqrt{(r \cos \phi)^2 + (r \sin \phi)^2} \, dr \, d\phi = \\ &= 2 \iint_{\{(r,\phi) : 0 \leq r \leq 1, 0 \leq \phi \leq 2\pi\}} r^2 \, dr \, d\phi = 2 \int_0^{2\pi} d\phi \int_0^1 r^2 dr = \frac{4\pi}{3}. \end{aligned}$$

### 4 Описание алгоритма

На вход алгоритма подается точность  $\epsilon$ . На первом шаге генерируется  $n$  случайных точек, равномерно распределенных в  $\{(y, z) : 0 \leq y \leq 1, 0 \leq z \leq 1\}$ . На втором шаге вычисляется приближенное значение интеграла по формуле 1. На третьем шаге вычисляется величина  $\delta = |\frac{4\pi}{3} - I|$ . На четвертом шаге, если  $\delta > \epsilon$ , то переходим к первому шагу, генерируя при этом дополнительно  $n$  точек (не обнуляя старое значение суммы, а уточняя значение интеграла); иначе завершаем работу алгоритма.

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. Все процессы-рабочие вычисляют свою часть суммы по формуле 1. Затем, сумма с помощью операции редукции вычисляется на процессе-мастере. После чего вычисляется ошибка (разность между посчитанным значением и точным значением, вычисленным аналитически). В случае если ошибка вы-

ше требуемой точности, которую подали на вход программе, то генерируются дополнительные точки и расчёт продолжается.

## **5 Исследование масштабируемости программы**

На момент написания отчета система Blue Gene/P не работает, а система Polus не поддерживает 64 MPI-процесса.

Как видно из таблицы 1, время последовательной программы достаточно мало, поэтому добавление распараллеливания с помощью технологии MPI только замедлит работу программы из-за накладных расходов.

Точность $\epsilon$	Число MPI-процессов	Время (с)	Ошибка
$3.0 \cdot 10^{-5}$	0	0.014	0.000018
	2	2.8234	0.000019
	4	4.8362	0.000022
	16	7.5734	0.000020
	32	0.1208	0.000016
$5.0 \cdot 10^{-6}$	0	0.521	0.000002
	2	0.948	0.000002
	4	5.3836	0.000003
	16	0.5686	0.000003
	32	0.6402	0.000003
$1.5 \cdot 10^{-6}$	0	0.8484	0.000001
	2	5.2354	0.000001
	4	4.8646	0.000001
	16	1.5236	0.000001
	32	2.7864	0.000001

**Таблица 1.**

Мы видим замедление программы из-за накладных расходов. Параллельная программа для 2 MPI-процессов в парадигме "мастер-рабочий" представляет собой аналог последовательной программы в том смысле, что только один процесс вычисляет значение интеграла.

Как видно из результатов работы программы о масштабируемости и ускорении не приходится говорить. Эта программа отлично справляется в сво-

ей последовательной версии и параллелизация её только замедляет, поэтому здесь не приведено никаких графиков.