



**Нижегородский государственный университет
им. Н.И. Лобачевского**

Факультет Вычислительной математики и кибернетики

Реализация вычисления определенного интеграла методом прямоугольников с использованием CUDA

Горшков А.В., Бастратов С.И.

ВМК ННГУ

anton.v.gorshkov@gmail.com

Постановка задачи

- Рассматривается вычисление интеграла:

$$I = \int_a^b f(x) dx$$

- Для численного вычисления интеграла используется метод прямоугольников:

$$I \approx h \sum_{i=0}^{n-1} f\left(a + i * h + \frac{h}{2}\right)$$

где n – размерность равномерной сетки на отрезке $[a, b]$ (число узлов сетки равно $n + 1$), h – шаг сетки.



Задание

- ❑ Ознакомиться с предоставленным проектом `CUDA_Integral`, убедиться в том, что он компилируется, запускается и выдает “Test FAILED.”.
- ❑ Реализовать функцию `integral_gpu` в файле `integral.cu`. Не превращать ее в ядро и не менять интерфейс!
- ❑ Число потоков в блоке является фиксированным и равным `THREADS_PER_BLOCK` (макрос уже определен в файле `integral.cu`), число блоков передается в функцию `integral_gpu` в качестве параметра и также не зависит от n .
- ❑ Таким образом, каждый поток вычисляет несколько слагаемых суммы.



Задание

- ❑ Для хранения частичных сумм для каждого потока использовать массив в разделяемой памяти (количество элементов массива равно количеству потоков, т.е. это будет статический массив длины `THREADS_PER_BLOCK`).
- ❑ После вычисления всех элементов данного массива просуммировать их и записать результат в массив частичных сумм блоков.
- ❑ После завершения работы ядра скопировать частичные суммы блоков в память CPU, просуммировать их и получить окончательный результат. Память для частичных сумм на CPU и GPU выделена заранее и указатели передаются в функцию `integral_gpu` в качестве параметров.
- ❑ Дополнительное задание: сохранив данную схему работы, сделайте код ядра не зависящим от `THREADS_PER_BLOCK`.

