4. Лабораторная работа №2

4.1. Задание №1

Используя средства OpenMP для редукции действий с массивами, реализуйте параллельные функции по нахождению суммы, минимума и максимума некоторого массива. Код заготовки модуля `reduction` находится в файле `reduction.f90` в директории `src`. Вам необходимо дополнить функции `omp_sum`, `omp_max` и `omp_min`.

После того, как функции будут реализованы, их надо протестировать с помощью программы `test_reduction.f90` из каталога `test`.

Ознакомьтесь с кодом программы и выясните как её следует запускать.

Данная программа для нахождения среднего времени работы функции использует алгоритм реального времени (on-line average), который реализован в функции `online_average` из модуля `stats` из каталога `src`. Сам алгоритм нахождения среднего основывается на постоянном обновлении среднего значения выборки по мере поступления новых значений. Его преимущество заключается в том, что не требуется накапливать результаты замеров в массив, что экономит память в случае большого количества замеров.

Протестируйте быстродействие параллельных функций с помощью скрипта для построения графиков `plot.py`.

Как им пользоваться показано в лабораторной №0. Вы должны получить по три графика на каждую из созданных функций.

4.2. Задание №2

Используйте формулу трапеции для реализации функции `trapezoidal`, которая аппроксимирует значение интеграла от скалярной функции $f(x)$.

$$\int_a^b f(x)\mathrm{d}x \approx \frac{h}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} f(x_i) \cdot h,\ x_i = a + ih.$$

Код заготовки модуля `trapezoidal_rule` вам необходимо дополнить, он находится в одноименном файле все в стой же директории `src`.

Функция должна уметь распараллеливать свою работу на заданное число потоков с помощью технологии OpenMP. После реализации функции проверьте корректность ее работы и быстродействие с помощью тестирующей программы `test_trapezoidal.f90` из каталога `test`.

Исходный код модуля `integrands`, где находятся подынтегральная функция смотрите в файле `integrands.f90` в каталоге `src`.

Для нахождения среднего также используется модуль `stats`, кратко описанный в предыдущем задании. Протестировать быстродействие параллельного вычисления в зависимости от числа потоков можно с помощью скрипта для построения графиков `plot.py`.