Задания по OpenMP

1. Две произвольные функции f(x) и g(x), гладкие на отрезке [a, b].

Требуется найти min[f'(x)] и max[g'(x)]. Для численных расчётов использовать N равных интервалов. (4 балла)

Требования:

* расчёт минимума и максимума должен быть обёрнут в директиву section;
* распараллелить внутренние циклы.

1. Дана произвольная непрерывная функция h(x). Вычислить интеграл при помощи квадратуры четвёртого порядка погрешности. Для численных расчётов использовать N равных интервалов. Использовать OpenMP для параллелизации вычислений. (4 балла)
2. Исследовать время выполнения расчётов задач 1 и 2 при разном количестве исполняющих потоков. Увеличьте N, если это необходимо. (2 балла)

## *Небольшая справка*:

Большинство вычислительных методов работает по следующей схеме.

Исходный интервал [a, b], на котором идёт работа, разбивается на N отрезков точками xi (i=1, N-1). Возьмём простейшее — равномерное — разбиение. Тогда взяв x0=a, xN=b, получим xi = a + (b-a)\*i/N. Или, взяв обозначение ∆x = (b-a)/N, получим xi = a + i\*∆x.

И далее вместо всех x из [a, b] будем рассматривать только xi.

Вспомним, что по определению производная определяется по формуле y'(x) = lim(ε⟶0)[y(x+ε) - y(x)]/ε.

Введём обозначение yi = y(xi)

Так как в нашем дискретном случае минимальное расстояние между x равно ∆x, получим разделённую разность y'R(xi) = [y(xi+∆x) - y(xi)]/∆x = (yi+1 - yi)/∆x.

На самом деле, никто не запрещает поменять знак при ∆x и получить симметричную формулу y'L(xi) = (yi-1 - yi)/(-∆x) = (yi - yi-1)/∆x.

По факту y'Ri определяет правостороннюю производную, y'Li — левостороннюю в данной точке.

Так как значения производных не зависят друг от друга, мы можем вычислять их независимо на каждом интервале.

Теперь займёмся интегралом.

Вспомним, что интеграл является всего лишь суммой, то есть представи́м в виде: ∫[a, b]y(x)dx = ∑[0 ≤ i ≤ N] ciy(xi) + o(∆xM).

ci здесь — это коэффициенты метода, а M — порядок погрешности.

Сумма в правой части и называется квадратурой.

Один из способов получить квадратуру — это взять интерполяционный многочлен, подставить в левую часть вместо y(x) и вычислить интеграл.

Всем известный интерполяционный многочлен Лагранжа

P(x) = ∑[0 ≤ i ≤ N] { ∏[0 ≤ j ≤ N, j != i] (x-xj) / ∏[0 ≤ j ≤ N, j != i] (xi-xj) }

на нашей равномерной сетке можно упростить.

Основная идея заключается в том, чтобы сделать замену x = x0 + k∆x, и заметить, что (xi-xj) = (i-j)∆x. Тогда наш интеграл по [a, b] превращается в интеграл по [0, N]. Отмечу лишь, что в интерполяционный многочлен на равномерной сетке называется интерполяционным многочленом Ньютона.

Пропустим страшные вычисления, в которых это делается, и оставим это на совести численных методов анализа :)

В итоге, подставив P(x) в левую часть и взяв интеграл, получим так называемые формулы Котеса.

Например, для трёх точек мы получим коэффициенты (b-a)/6, 2(b-a)/3, (b-a)/6, что можно записать в виде формулы Симпсона:

∫[a, b]y(x)dx ≈ (b-a)(y0 + 4y1 + y2)/6.

Для одной точки имеем всем известный [метод прямоугольников](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2), а для двух — [метод трапеций](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D1%82%D1%80%D0%B0%D0%BF%D0%B5%D1%86%D0%B8%D0%B9).

Но что делать, если мы хотим использовать что-то более точное, ведь мы разбивали интервал на N частей, где хотим взять большое N!? Всё просто: используем аддитивное свойство интеграла: ∫[a, b]y(x)dx = ∑[0 ≤ i < N] ∫[xi, xi+1]y(x)dx. Каждый интеграл по кусочку интервала получается независим от других.