

Задание (использование директивы task)

(Всего нужно набрать 11 баллов)

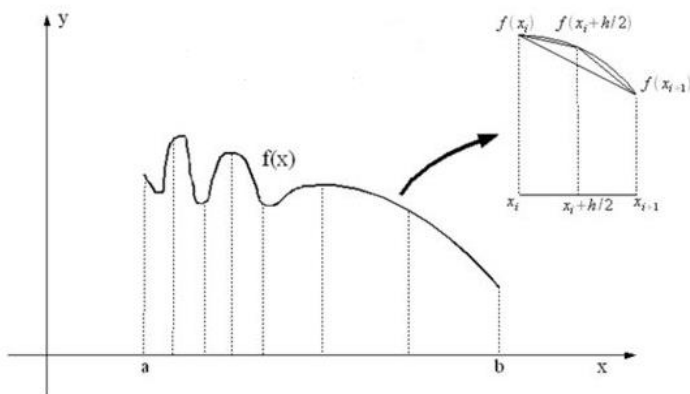
Адаптивная квадратура (5 баллов)

1) Напишите функцию f , которая вычисляет $1/(1+x*x)$.

2) Напишите функцию `integr`, которая принимает в качестве параметров концы отрезка и другую функцию, и вычисляет ее интеграл на указанном отрезке, пользуясь формулой трапеций по 10 отрезкам разбиения.

3) Напишите функцию `integral`, которая принимает в качестве параметров концы отрезка и другую функцию, и вычисляет ее интеграл на указанном отрезке, пользуясь **алгоритмом адаптивной квадратуры**: сначала вычисляется интеграл на интересующем нас отрезке, а также на половинках нашего отрезка, при помощи функции `integr`. Если значение интеграла отличается от суммы интегралов по половинкам нашего отрезка не более, чем на $1e-10$, ответ считается правильным и возвращается в качестве функции `integral`. Если же это не так, то в качестве результата функции `integral` возвращается сумма интегралов по половинкам нашего отрезка, вычисленных при помощи функции `integral`.

Адаптивная квадратура



Адаптивность заключается в том, что там, где подынтегральная функция изменяется быстро, рекурсивных вызовов функции `integral` будет больше. На тех участках, где подынтегральная функция изменяется медленно, не потребуется сильно измельчать соответствующий отрезок, поэтому рекурсивных вызовов будет меньше.

Частота разбиения выбирается в соответствии с плавностью изменения функции.

4) Распараллельте алгоритм адаптивной квадратуры при помощи директивы `task` из OpenMP. Можно написать сначала последовательный алгоритм с рекурсией, а потом параллельный алгоритм.

Быстрая сортировка (3 балла)

Используя директиву `task`, распараллельте алгоритм быстрой сортировки. Можно воспользоваться последовательным алгоритмом, который рассматривался в первом семестре.

Список (3 балла)

1) Напишите структурный тип данных `Pair`, состоящий из двух полей: целое число и логическое значение. Напишите также функцию `processNode`, принимающую по ссылке такую структуру и присваивающую логическому полю результат проверки целого поля на простоту.

2) Напишите функцию `processList`, принимающую два параметра: библиотечный список `list<Pair>` и указатель на функцию

`void (*f)(Pair &)`

принимающую элемент списка по ссылке, и применяющую функцию-параметр к данным, хранящимся в каждом элементе списка.

3) Распараллельте функцию из предыдущей задачи при помощи директивы `task` из OpenMP.

4) Исследуйте ускорение от распараллеливания, запуская функции из задач 2 и 3, на примере списка из структур задачи 1, при заполнении целых полей структур случайными положительными числами.

Можно заменить структуру `Pair` шаблонным классом `pair` стандартной библиотеки C++. Возможно также предложить другую функцию в пункте 1.