

Отчет о программе, разработанной на C++ Вариант 34 Демиденко Ксения БПИ208

Задание

34. *Задача для агронома.* Председатель дачного кооператива Сидоров В.И. получил указание, что в связи с составлением единого земельного кадастра, необходимо представить справку о площади занимаемых земель. Известно, что территория с запада и востока параллельна меридианам, на севере ограничена параллелью, а с юга выходят к реке, описываемой функцией $f(x)$. Требуется создать многопоточное приложение, вычисляющее площадь угодий методом адаптивной квадратуры. При решении использовать парадигму рекурсивного параллелизма. Замечание: кривизну Земли из-за малой занимаемой площади не учитывать.

Спецификация

Спецификация BC

OS: macOS 11.5.2

Спецификация средств разработки IDE

CLion 2021.2.1

Модель приложения

При вычислении интеграла используется парадигма «рекурсивного параллелизма».

Описание:

Рекурсивный параллелизм: процессы выполняют независимые рекурсивные вызовы процедуры (комбинаторные задачи, метод ветвей и границ и т. д.) ¹

Программа считается рекурсивной, если она содержит процедуры, которые вызывают сами себя — прямо или косвенно. Рекурсия дуальна итерации в том смысле, что рекурсивные программы можно преобразовать в итеративные и наоборот. Однако каждый стиль программирования имеет свое применение, поскольку одни задачи по своей природе рекурсивны, а другие — итеративны.

Рекурсивную программу можно реализовать с помощью параллелизма, если она содержит несколько независимых рекурсивных вызовов. Два вызова процедуры (или функции) являются независимыми, если их множества записи не пересекаются. Это условие выполняется, если: 1) процедура не обращается к глобальным переменным или только читает их; 2) аргументы и результирующие переменные (если есть) являются различными переменными. Например, если процедура не обращается к глобальным переменным и имеет только параметры значения, то любой ее вызов будет независимым. (Хорошо, если процедура читает и записывает только локальные переменные, тогда каждый экземпляр процедур имеет локальную копию переменных.) ²

Особенности реализации

При работе с программой пользователю представляется несколько способов организации входных данных (о способах см. ниже), таких как сама функция, а также границы вычисления интеграла. Далее интеграл вычисляется рекурсивным методом адаптивной квадратуры, рекурсия запускается в новых потоках, однако их количество ограничено, и после превышения лимита новые вызовы рекурсии уже не будут порождать новые потоки.

Все данные по работе программы или об ошибках, если они возникнут, будут выведены в консоль.

При работе с функциями, коэффициенты которых слишком велики при большом расстоянии между левой и правой границей вычисления могут длиться достаточно долго.

¹ [Парадигмы параллельного программирования](#)

² [Обзор области параллельных вычислений п.1.5](#)

Организация входных данных

Вычисляемая функция представляет собой многочлен с действительными коэффициентами вида $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, причем во избежание долгой работы программы количество коэффициентов ограничено до 7 (то есть максимальная степень многочлена функции равна 6).

- Функция может быть считана из файла, тогда файл должен представлять из себя список коэффициентов функции, записанных через пробел.

Например, входной файл с данными вида:

100 -3 54.5 2

Будет преобразован в функцию $100x^3 - 3x^2 + 54.5x + 2$

- Если же файл с функцией не предоставлен, она будет случайно сгенерирована программой.

Левая и правая границы вычисления занимаемой площади также могут быть представлены пользователем в качестве аргументов командной строки (каждая из границ представляет из себя действительное число, модуль которого меньше 100, причем левая граница должна быть меньше правой).

- Границы могут быть переданы пользователем как аргументы командной строки через пробел, причем сначала записывается левая граница, а потом правая.
- Если же границы не предоставлены, они будут случайно сгенерированы программой.

Ввод данных

Для удобства использования реализовано 4 способа ввода функции и границ вычисления:

1. Чтобы функция и границы были сгенерированы случайно, можно просто запустить исполняемый файл без аргументов:

```
./ABCtask5
```

2. Чтобы работать с заданной функцией, но случайно генерировать границы, первым аргументом нужно передать txt-файл с заданными коэффициентами (формат файла описан выше, для примера в основном каталоге с остальными исполняемыми будет помещен файл coefficients.txt, в который можно записывать свои коэффициенты):

```
./ABCtask5 <path_to_coefficients_file>
```

3. Чтобы работать с случайной функцией, но при этом задать границы, первым и вторым аргументами нужно передать левую и правую границы вычислений (формат ввода границ описан выше):

```
./ABCtask5 <west_bound> <east_bound>
```

4. Чтобы задать свой файл с функцией и свои границы вычислений, первым аргументом нужно передать txt-файл с заданными коэффициентами, а вторым и третьим аргументами передать левую и правую границы вычислений:

```
./ABCtask5 <path_to_coefficients_file> <west_bound> <east_bound>
```