НАГЛЯДНЫЙ АНАЛИЗ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ ОПРЕДЕЛЁННЫХ ИНТЕГРАЛОВ

Звягинцева В. А., Найдер П. М.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Стройникова Е. Д. – ст. преп. кафедры информатики

В данной работе было проведено сравнение алгоритмов приближённого вычисления определённых интегралов. Опытным путём было установлено, что лучшим алгоритмом является метод трапеций. Реализованы учебные программы на языке программирования С++ для ознакомления и изучения студентами. Преимуществами программных кодов являются их понятность и доступность.

В математическом анализе интегралом функции называют предел суммы для бесконечного числа бесконечно малых слагаемых. Интегрирование – процесс отыскания множества первообразных. Функция $F(x)$ называется **первообразной** для функции $f(x)$ на некотором промежутке, если для всех x из этого промежутка выполняется равенство $F'(x) = f(x)$, или, что то же самое, $d(F(x)) = f(x)dx$. Конечный предел интегральной суммы

$$\sum_{i=0}^{n-1} f(\varepsilon_i)\Delta x_i \tag{1}$$

при $\Delta x_i \to 0$, не зависящий ни от способа дробления отрезка $[a; b]$, ни от выбора точек ε_i, называется **определённым интегралом** функции $f(x)$ по промежутку $[a; b]$ и обозначается символом

$$\int_a^b f(x)dx. \tag{2}$$

Существует несколько методов вычисления определённых интегралов:

1) метод прямоугольников:
   - левых;
   - правых;
2) метод трапеций;
3) по формуле Симпсона.

Метод прямоугольников – метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на многочлен нулевой степени, т. е. константу, на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота – значением подынтегральной функции в этих узлах.

Если отрезок $[a; b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по **формуле левых прямоугольников**:

$$\int_a^b f(x)dx \approx f(a)(b-a), \tag{3}$$

иначе по **формуле правых прямоугольников**:

$$\int_a^b f(x)dx \approx f(b)(b-a). \tag{4}$$

Метод трапеций – метод, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, т. е. линейную функцию. Площадь под графиком функции аппроксимируется прямоугольными трапециями:

$$\int_a^b f(x)dx \approx \frac{f(a)+f(b)}{2}(b-a). \quad (5)$$

Формула Симпсона. Суть метода заключается в приближении подынтегральной функции на отрезке [a; b] интерполяционным многочленом второй степени, т. е. приближением графика функции на отрезке параболой:

$$\int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right), \quad (6)$$

где f(a), f(b), $f\left(\frac{a+b}{2}\right)$ – значения функции в соответствующих точках (на концах отрезка и в его середине).

Имея реализованные алгоритмы, несложно определить их временную сложность. Получив данные, можно сравнить их и с лёгкостью узнать, какой же из методов работает лучше остальных. Для этого была составлена программа на языке программирования C++ с использованием фреймворка Qt.

Для оценки производительности алгоритмов применяется асимптотическая нотация.

Асимптотические обозначения. Намного легче работать с верхней и нижней границами функций временной сложности, используя для этого обозначение *O*-большое. Асимптотические обозначения позволяют упростить анализ, поскольку игнорируют детали, которые не влияют на сравнение эффективности алгоритмов. В частности, в асимптотических обозначениях игнорируется разница между постоянными множителями.

Чтобы программа могла обработать подынтегральное выражение, которое компьютер получает в качестве строки понятных ему математических выражений, было принято решение создать свой синтаксический анализатор.

Синтаксический анализ – процесс сопоставления линейной последовательности лексем естественного или формального языка с его формальной грамматикой.

Для создания анализатора был использован алгоритм нисходящего синтаксического анализа, реализуемый путём взаимного вызова процедур, где каждая процедура соответствует одному из правил формальной грамматики.

Полученная оценка сложности для каждого из алгоритмов равна $O(n \times 2^k)$, где n – количество разбиений, k – длина строки подынтегрального выражения.

Для начала было выбрано количество разбиений, равное 1000. Быстрее всего отработал метод трапеций, однако на таком количестве входных данных сложно выявить лучший алгоритм, т. к. разница между временем работы алгоритмов ничтожна мала. Также было установлено отличие между ответом, полученным методом правых прямоугольников, и ответами, полученными прочими методами.

При бо́льшем количестве разбиений результат стал уже заметнее. Теперь ответы везде совпали, а это значит, что можно полностью сосредоточиться именно на времени выполнения программы. Самым быстрым на этот раз снова оказался метод трапеций, отработав всего чуть более 5-ти минут. Но стоит отметить, что отрыв опять же не столь велик. Однако он есть и далее растёт с увеличением количества разбиений, а это значит, что лучший алгоритм приближённого вычисления определённых интегралов – метод трапеций.

Список использованных источников:

1. Сборник задач по высшей математике : учеб. пособие. В 10 ч. Ч. 6: Интегральное исчисление функций одной переменной / А. А. Карпук [и др.]. – Минск : БГУИР, 2006. – 148 с.

2. Карпук, А. А. Сборник задач по высшей математике : учеб. пособие. В 10 ч. Ч. 7: Интегральное исчисление функций многих переменных / А. А. Карпук, В. В. Цегельник, Е. А. Баркова. – Минск : БГУИР, 2007. – 119 с.

3. Скиена, С. Алгоритмы. Руководство по разработке / Стивен С. Скиена. – 2-е изд. ; пер. с англ. – СПб. : БХВ-Петербург, 2011. – 720 с.