Московский авиационный институт

(государственный технический университет)

Факультет «Прикладная математика и физика»

Кафедра «Вычислительная математика и информатика»

Курсовой проект по

информатике и вычислительной технике

по теме:

«Вещественный тип. Приближенные вычисления. Табулирование функций»

Выполнил: Мигалев Р. П.

Студент группы М8О-106Б Преподаватель: Дубинин А. В.

Оценка:

Подпись:

Дата:

2016

Введение

Вычислительные машины обладают своим набором арифметических и логических операций, с помощью которых выполняют все задачи, переданные процессору. Чем меньше таких операций и чем они проще, тем производительнее данный компьютер. Таким образом, мы имеем необходимость в создании композиции простых команд, которые заменят собой другие, более сложные. В данном проекте мы работаем с примером такой задачи: требуется вычислить значение трансцендентной функции. Будем делать это двумя способами: используя функции из библиотеки maht.h или раскладывая функцию по формуле Тейлора.

Формула Тейлора

Рядом Тейлора функции f(x), дифференцируемой в окрестности точки «а» k раз, называется функциональный формальный ряд:

Данная формула позволяет привести функцию, дифференцируемую достаточное количество раз, к алгебраическому многочлену. Таким образом, компьютер, который способен складывать и умножать числа, сможет вычислить значение данной трансцендентной функции, используя элементарные операции. Этот алгоритм не является наиболее точным, быстрым и эффективным, но служит хорошим примером использования простых операций вместо сложных.

Вычисление значения функции по формуле Тейлора

Разложение функции  по формуле Тейлора:



где x — точка, n+1 — количество элементов в ряду.

Погрешность измерений равна разнице между значениями, вычисленными с помощью стандартных функций библиотеки math.h и с помощью ряда Тейлора.

Тип double ограничивает точность значения функции, поэтому нет смысла использовать слишком большое n. Количество итераций определяет, после какого значения n значение суммы ряда Тейлора нашей функции в точке перестает меняться.

Машинное эпсилон — минимальное число, отличное от нуля, при прибавлении которого к единице получается число, отличное от единицы.

Алгоритм работы программы

1) Считать требуемое число разбиений *n*.

2) Рассчитать длину шага *dx = (rx - lx) / 2*, где *rx* и *lx* — границы заданного отрезка.

3) Для каждой *i*-ой итерации цикла на *(n + 1)* повторение:

3.1) Рассчитать текущую точку на отрезке *x = lx + dx \* i*.

3.2) Рассчитать значение функции стандартными стандартными средствами языка: *f\_x = f(x)*, где *f* — это функция, принимающая на вход аргумент и возвращающая значение функции при нем.

3.3) Рассчитать значение *ts* функции по Тейлору:

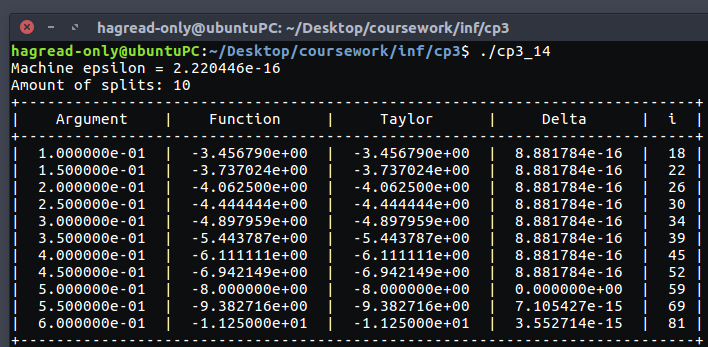
3.3.1) Для первого и текущего элемента ряда сказать, что он равен -3, т.е. *summand = -3*. Обозначить также переменную *tsr = summand* — разложение функции по Тейлору при порядке n = 0. Хранить в переменной *i* количество проделанных итераций (фактически число слагаемых в совершённом разложении).

3.3.2) Пока *|summand| > eps* (машинный эпсилон, рассчитанный ранее) и *i < 100* рассчитывать текущее слагамое разложения *summand* по формуле *-(i + 3) \* pow(x, i)*, прибавлять его к разложению *tsr i*-ого порядка, инкрементировать *i*.

3.3.3) Записать во внешнюю переменную *iterations* полученное число *i*, вернуть результат *tsr*.

3.4) Вывести новую строку таблицы, содержащую в себе x, f\_x, ts, |f\_x - ts|, iterations.

Результат работы программы



Заключение

Таким образом, на примере данной программы, мы видим, что разложение функции по формуле Тейлора в алгебраический многочлен позволяет вычислить достаточно точное значение данной функции. Этот метод можно использовать для любой функции, дифференцируемой достаточное число раз. Данная программа также является отличным примером того, как можно использовать композицию простых функций вместо одной сложной.