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:

«Процедуры и функции в качестве параметров»

Выполнил: Мигалев Р. П.

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

Оценка:

Подпись:

Дата:

2016

Введение

Численные алгоритмы используются для работы с числами. Они располагают величины в случайном порядке, раскладывают числа на простые множители, находят НОД и НОК, определяют геометрические площади и т. д. И хотя подобные алгоритмы применимы лишь в некоторых случаях, лежащие в их основе методы очень полезны. Одними из таких методов являются методы поиска нулей функции.

Методы поиска корня уравнения на отрезке

Рассматривается уравнение вида F(x) = 0. Предполагается, что функция F(x) достаточно гладкая, монотонная на этом отрезке, и существует единственный корень уравнения x\* на отрезке [a, b]. На отрезке [a, b] ищется приближенное решение x с точностью ε, т. е. такое, что |x - x\*| < ε.

*Метод дихотомии:*

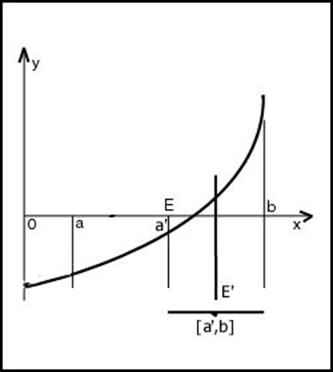
Очевидно, что если на отрезке [a, b] существует корень уравнения, то значения функции на концах отрезка имеют разные знаки: F(a) \* F(b) < 0. Метод заключается в делении отрезка пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака функции в середине отрезка.

- Начальное приближение: границы исходного отрезка a0 = a, b0 = b.

- Итерационный процесс: ak+1 = (ak+bk) / 2, bk+1 = bk, если F(ak) \* F((ak + bk) / 2) > 0 ИЛИ ak+1 = ak, bk+1 = (ak+bk) / 2, если F(bk) \* F((ak + bk) / 2) > 0

- Условие окончания: |ak-bk| < ε.

- Приближенное значение корня: x\* ≈ (ak+bk) / 2.



*Метод итераций:*

Идея метода заключается в замене исходного уравнения F(x) = 0 уравнением вида x = f(x).

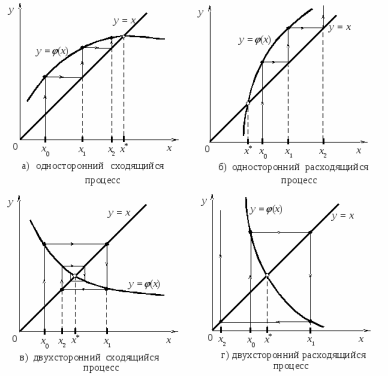
- Достаточное условие сходимости метода: |f`(x)|<1, x принадлежит [a, b].

- Начальное приближение корня: x0 = (a+b)/2 (середина исходного отрезка).

- Итерационный процесс: xk+1 = f(xk).

- Условие окончания: |xk-xk-1| < ε.

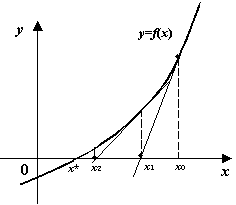
- Приближенное значение корня: x\* ≈ xконечное.



*Метод Ньютона:*

Метод Ньютона является частным случаем метода итераций.

- Условие сходимости: |F(x)\*F``(x)|<(F`(x))2 на отрезке [a, b].

- Итерационный процесс: xk+1 = xk-F(xk)/F`(xk).

Алгоритм работы программы

В качестве ε взята величина 1E-8.

1) Считать левую и правую границы рассматриваемого отрезка, взять начальное приближение mx как их среднее арифметическое.

2) Рассчитать корень методом *дихотомии*:

2.0) Передать в функцию *dichotomy* параметры *lx* и *rx —* заданные границы.

2.1) Рассчитать длину отрезка *dx = |lx - rx|* и его середину *mx = (lx - rx) / 2*.

2.2) Если произведение функций от левой границы и середины отрезка меньше нуля, т.е. *f(lx) \* f(mx) < 0*, то сдвигаем левую границу *(lx = mx)*, иначе — правую *(rx = mx)*. Повторяем процесс до тех пор, пока длина отрезка больше ε — *dx > ε*.

2.3) Вывести *mx* — искомое приближение нуля функции.

3) Рассчитать корень методом *итераций*:

3.0) Передать в функцию *iterations* начальное приближение, взятое как *mx = (lx + rx) / 2* — середина заданного отрезка, а также адрес указателя на логическую переменную *isConvergent*, в которой хранится информация о сходимости функции.

3.1) Если достаточное условие сходимости не выполняется — *|phi`(x)| >= 1* — записать в *isConvergent* значение *false* и завершить работу функции.

3.2) Найти новое значение *phi\_x = phi(x)*.

3.3) Если модуль разности между ним и старым значением меньше эпсилон — *|phi\_x - x| < ε*  — записать в *isConvergent* значение *true* и вернуть *phi\_x*.

3.4) Запустить рекурсивно функцию *iterations* с новым значением *phi\_x* и адресом указателя *isConvergent*.

4) Рассчитать корень методом *Ньютона*:

4.0) Передать в функцию *newton* начальное приближение, взятое как *mx = (lx + rx) / 2* — середина заданного отрезка, а также адрес указателя на логическую переменную *isConvergent*, в которой хранится информация о сходимости функции.

4.1) Если достаточное условие сходимости не выполняется — *|f(x) \* f``(x)| >= (f`(x))2* — записать в *isConvergent* значение *false* и завершить работу функции.

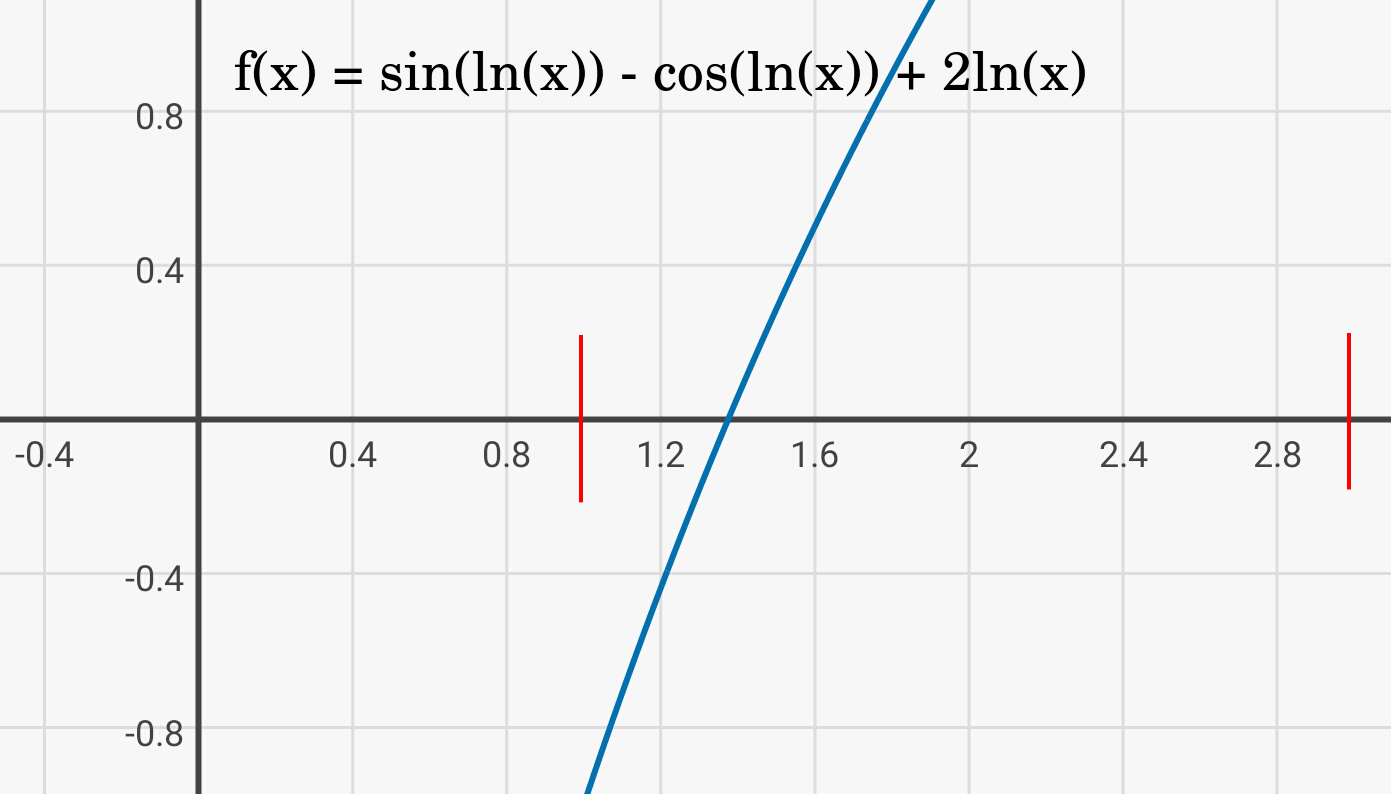
4.2) Рассчитать значение *k = f(x) / f`(x)*.

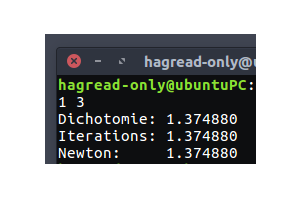
4.3) Получить новое значение x путем вычитания из старого значения *x* результат вычисления *k*.

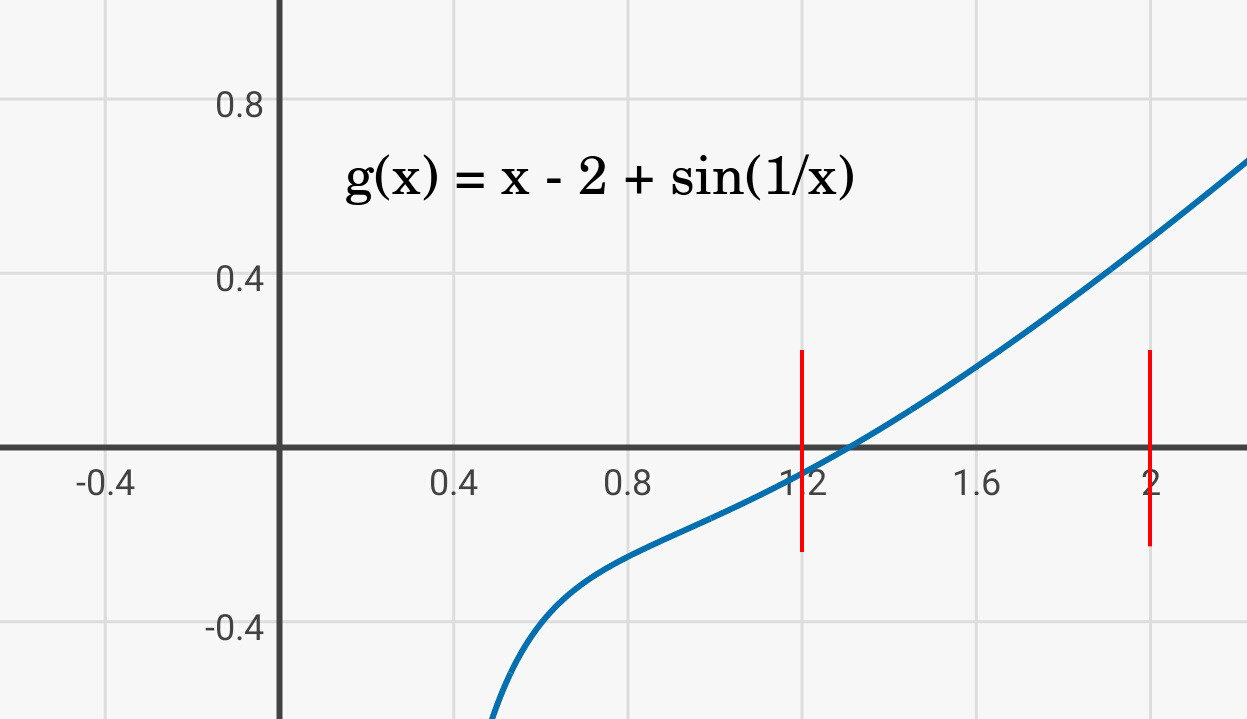
4.4) Если модуль разности между текущим *x* и старым значением меньше эпсилон — *|k| < ε*  — записать в *isConvergent* значение *true* и вернуть *x*.

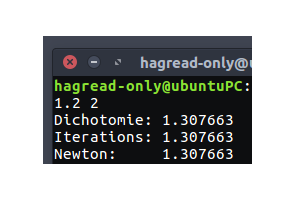
4.4) Запустить рекурсивно функцию *newton* с новым значением *x* и адресом указателя *isConvergent*.

Результат работы программы









Заключение

Таким образом мы смогли тремя различными способами найти корень уравнения на заданном отрезке и смогли сделать это довольно точно.

Данные методы в чистом виде имеют довольно ограниченную сферу применения. Тем не менее используемые в них идеи в отдельных ситуациях оказываются довольно ценными. К тому же данные методы не нуждаются в сложных структурах данных.