Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Учебно-исследовательская работа»

Тема: Решение нелинейного уравнения методом итераций

Семестр 1

Выполнил работу

Студент группы РИС-22-1Б

Вышенская Екатерина Игоревна

Проверил

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Г. Пермь-2022

**Введение**

В вычислительной практике часто приходится решать нелинейные уравнения. Это может составлять самостоятельную задачу или являться составной частью более сложных вычислений. В подавляющем большинстве методы решения нелинейных уравнений являются итерационными, то есть алгоритм решения состоит в многократном повторении вычислительной процедуры. Полученное таким образом решение всегда является приближенным, но может быть сделано сколь угодно близким к точному.

Целью работы является разработка программы, которая решает нелинейные уравнения методом итераций.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

1. Анализ решения уравнения методом итераций.

2. Реализация технологии разработки программы.

3. Решение различных уравнений.

**1. Анализ решения уравнения итераций.**

*Формулировка задачи*

Необходимо разработать программу, которая решает

нелинейные уравнения методом итераций.

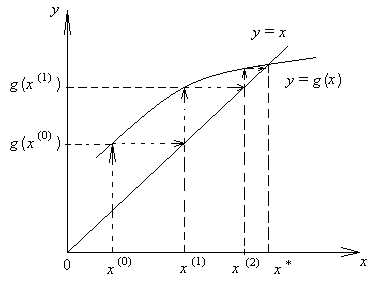
Методы приближенного нахождения корней - это технология, которая позволяет найти приближенное значение корня, но с очень высокой степенью точности.

Дано:

* f(x) = 0;
* Интервал [a; b], в котором находится корень;
* Эпсилон e – точность вычислений -> достижение точности, это окончание поиска приближенного значения корня обычно точность 10-6

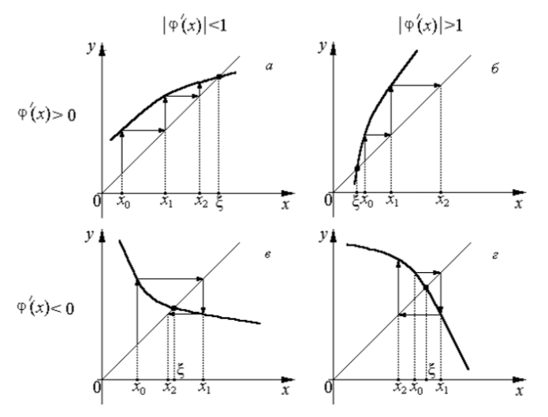
Метод итераций

1. Для применения метода простой итерации уравнение f(x)=0 представляется в виде x = φ(x), где φ(x)= f (x) + x.
2. Итерации проводятся по формуле xn+1 = φ(xn ).
3. В системе координат (x, y) построены графики функций y = φ(x) и y = x . Абсциссы точек пересечения этих линий соответствуют корням уравнения. Если имеется начальное приближение x0 , то на графике y = φ(x) легко находится точка (x0, φ(x0)). Проведя через нее прямую y = x1 до пересечения с прямой y = x , получим точку (x1, φ(x0)). Абсцисса этой точки x1 дает первое приближение к решению. На рисунке 1 представлена геометрическая интерпретация метода.



*Рис.1 - геометрическая интерпретация метода*

1. Итерации продолжаются до тех пор, пока |xn+1 − xn| >= e
2. На рисунке 2 изображено поведение последовательных приближений для случаев: |φ`(x)| < 1 и |φ′(x)| > 1. Как видно из рисунков, сходимость последовательности xn к истинному x\* имеет место лишь при выполнении условия |φ`(x)| < 1. Если же |φ′(x)| > 1, то итерационный процесс расходится.



*Рис.2 - условие сходимости метода*

**2. Реализация технологии разработки программы**

Поточный ввод-вывод в C++ выполняется с помощью функций сторонних библиотек. В С++ разработана новая библиотека ввода-вывода *iostream*, использующая концепцию объектно-ориентированного программирования. Библиотека *iostream* определяет три стандартных потока: *cin* стандартный входной поток, *cout* стандартный выходной поток, *cerr* стандартный поток вывода сообщений об ошибках (*stderr* в С). В задаче будут использованы потоки *cin* и *cout*.

В стандартную математическую библиотеку языка Си (а, значит, и C++) входит множество специальных математических функций, которые нужно знать и уметь использовать. Для того, чтобы использовать эти функции в своей программе, необходимо подключить *заголовочный* файл, содержащий описания этих функций, что делается строчкой в начале программы:

#include <math.h>

Для решения задачи понадобиться функция *pow* - возведение в степень, возвращает a^{b}. Использование: pow(a,b).

Также, для работы программы необходима функция abs(), которая возвращает абсолютную величину выражения, функция max() и min(), возвращающая максимальное и минимальное значение соовтветственно.

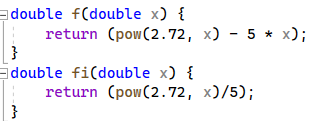
Функция setlocale() устанавливает или изменяет для текущей программы информацию о национальной специфике. В программе данная функция понадобиться для вывода сообщений на русском языке (Рис.2).



*Рис.3 - подключение русского языка функцией setlocale()*

В ходе решения задачи нам будут нужны переменные типа *double* a, b, x1, x2, e и переменная типа *int* i, отвечающая за кол-во итераций.

Также, нам понадобиться создать две функции типа *double* f(*double* x), первая будет отвечать за исходную функцию f(x), вторая - за функцию φ(x):



*Рис.4 - создание функции f(x) и функции φ(x)*

Для решения задачи нам понадобятся операторы ветвления *if* и *else* и оператор циклического процесса *while*.

Анализ решения:

0. Первая корень x1 в точке a. Второй корень x2 = fi(x1).

1. Проверка на сходимость (abs(x2 - x1) > abs(fi(x2)) - x2), если разница между найденными корнями уменьшается, то процесс схождения происходит, в противном случае программа выводит сообщение "Условие на сходимость не выполнено!" и заканчивает работу.
2. Цикл while, условием выхода из которого является проверка на точность abs(x2 - x1) >= e.
3. Внутри цикла: присваивание переменной x1 старого значения x2, а переменной x2 - нового значения x2 = fi(x2). Также - увеличение счётчика i на 1 за каждую итерацию цикла.
4. После цикла - проверка, входит ли найденный корень в интервал (x2 <= max(a,b) && x2 >= min(a,b)). Если условие не выполняется, программа выводит сообщение "В заданном интервале нет корня!" и заканчивает работу.
5. Вывод корня и кол-ва итераций при выполнении условия из предыдущего пункта.

*Полный код программы представлен в приложении А.*

*В приложении Б представлена графическая модель решения задачи.*

**3. Решения различных уравнений.**

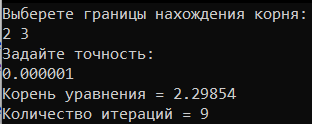
1. 3 \* sin(pow(x, 0.5)) + 0.35 \* x - 3.8

Границы: [2;3]

Переход от функции f(x) к функции fi(x):

fi(x) = (3 \* sin(pow(x, 0.5)) - 3.8) / (- 0.35)

Работа программы:

****

*Рис.5 - решение уравнения (1)*

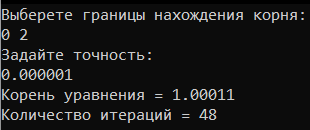
1. 0.25 \* pow(x, 3) + x - 1.2502

Границы: [0;2]

Переход от функции f(x) к функции fi(x):

fi(x) = (0.25 \* pow(x, 3) - 1.2502)/(- 1)

Работа программы:

****

*Рис.6 - решение уравнения (2)*

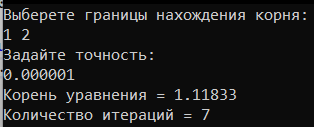
1. 0.1 \* pow(x,2) - x \* log(x)

Границы: [1;2]

Переход от функции f(x) к функции fi(x):

fi(x) = 0.1 \* pow(x, 2) - x \* log(x) + x

Работа программы:

****

*Рис.7 - решение уравнения (3)*

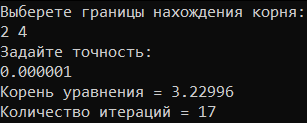
1. 3 \* x - 4 \* log(x) - 5

Границы: [2;4]

Переход от функции f(x) к функции fi(x):

fi(x) = (- 4 \* log(x) - 5) / (- 3)

Работа программы:

****

*Рис.8 - решение уравнения (4)*

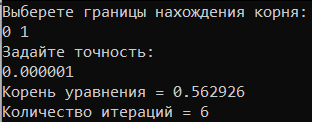
1. acos(x) - pow(1 - 0.3 \* pow(x,3), 0.5)

Границы: [0;1]

Переход от функции f(x) к функции fi(x):

fi(x) = 0.1 \* pow(x, 2) - x \* log(x) + x

Работа программы:

****

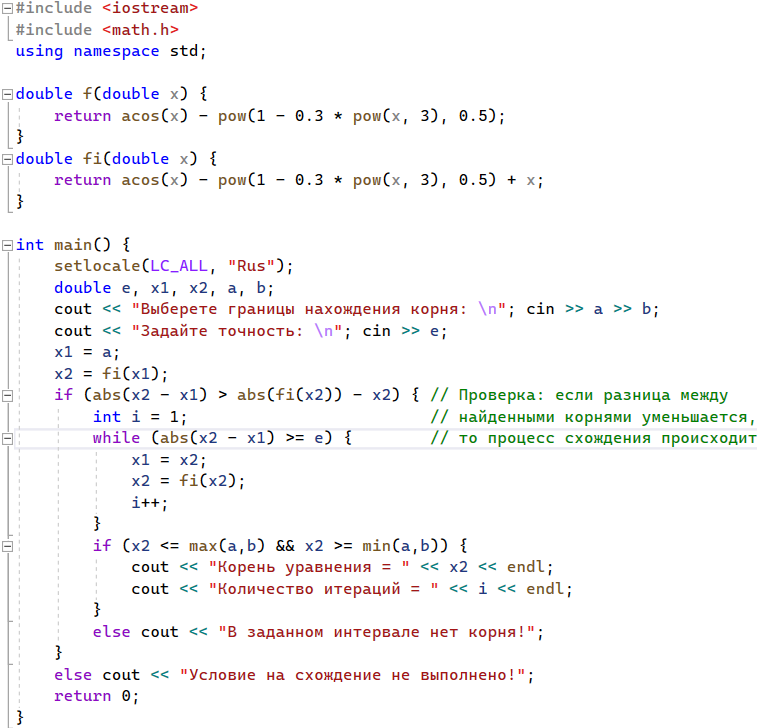
*Рис.9 - решение уравнения (5)*

**Заключение**

Была разработана программа, находящая приближенное значение корня нелинейного уравнения на заданном интервале с заданной точностью методом итераций. Нахождение корня данным методом значительно легче с точки зрения написания кода, чем методом дихотомии и метод Ньютона, т.к. в этом методе всего два условия, и мы пользуемся только одной функцией fi(x). Однако кол-во итераций варьируется от 6 до 48. Это можно отнести к минусам данного метода

**Приложения**

Приложение А – Код программы



Приложение Б – Графическая модель решения задачи.

