Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева

институт компьютерных технологий и защиты информации

кафедра Прикладной математики и Информатики имени Ю.В.Кожевникова

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

«Вычислительная математика»

Тема: «Итерационные методы решения нелинейных уравнений»

Выполнил

студент группы 4317

Мохамед М.О

Казань 2023

Цель работы: научиться решать нелинейные уравнения методом простых итераций, методом Ньютона и модифицированным методом Ньютона с помощью ЭВМ.

Содержание работы:

1. Изучить метод простых итераций, метод Ньютона и модифицированный метод Ньютона для решения нелинейных уравнений.
2. На конкретном примере усвоить порядок решения нелинейных уравнений с помощью ЭВМ указанными методами.
3. Составить программу (программы) на любом языке программирования и с её помощью решить уравнение с точностью и . Сделать вывод о скорости сходимости всех трёх методов.
4. Изменить , и снова решить задачу. Сделать выводы о: скорости сходимости рассматриваемых методов; влиянии точности на скорость сходимости, влиянии выбора начального приближения в методе простых итераций на скорость сходимости.
5. Составить отчёт о проделанной работе.

*ВЫПОЛНЕНИЕ РАБОТЫ*

Задание.

1. Графическим и аналитическим методами найти отрезок, на котором существует единственный корень нелинейного уравнения:

(1)

1. Построить итерационные формулы метода простых итераций, метода Ньютона и модифицированного метода Ньютона, реализующие процесс поиска корня нелинейного уравнения (1) на найденном отрезке.
2. Составить программу на любом языке программирования, реализующие поиск приближенного решения нелинейного уравнения на выбранном отрезке с помощью построенных итерационных процессов.

Решение.

*Графическим методом* найдем отрезок, на котором уравнение (1) имеет хотя бы один корень. Данная функция имеет сложный аналитический вид, поэтому преобразуем уравнение (1) к виду и построим два графика  и , имеющих более простой аналитический вид (рис.1). Абсцисса точки пересечения графиков принадлежит отрезку и является приближенным решением уравнения.

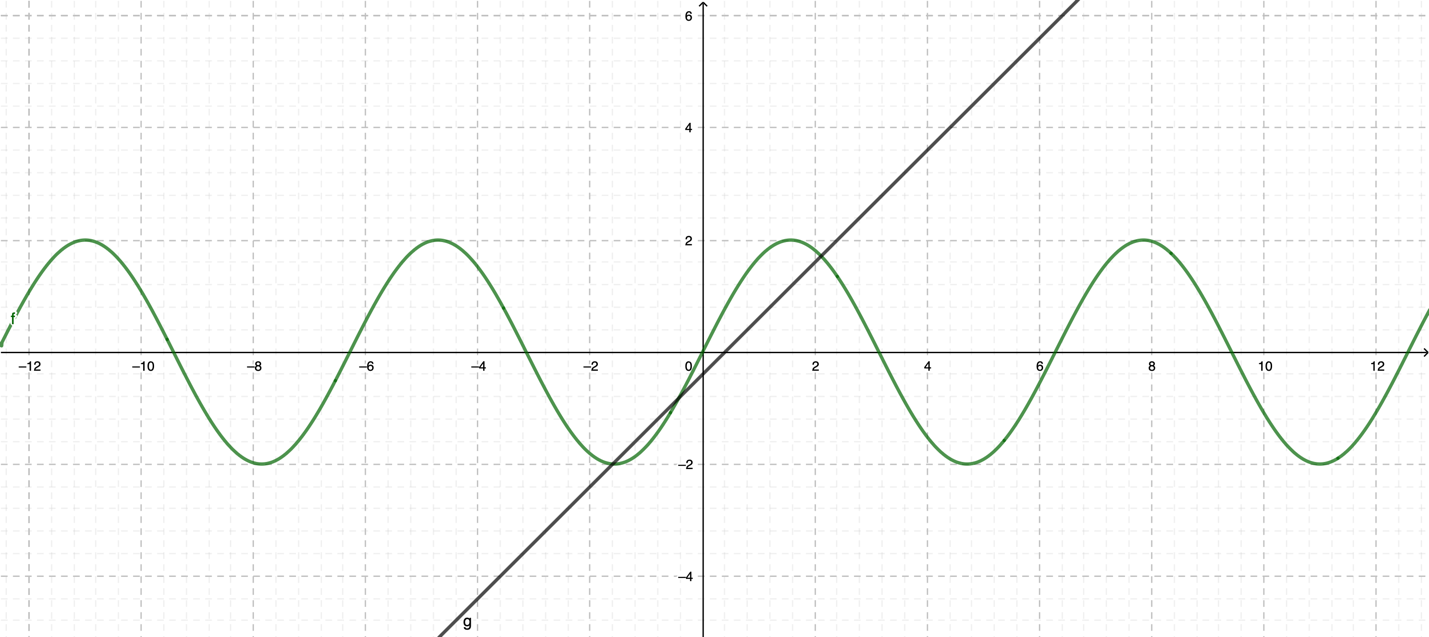


Рис. 1. График функций и

*Аналитический метод.* Докажем, что на отрезке уравнение (1) имеет единственное решение. Для этого необходимо выполнить проверку следующих трех условий:



Итак,



Итак, можно сделать вывод, что уравнение (1) имеет единственное решение на выбранном отрезке.

Метод простых итераций.

Пусть дано нелинейное уравнение (2), и оно имеет на отрезке единственный вещественный корень . Требуется найти корень с заданной точностью .

Умножаем обе части уравнения (1) на константу с и прибавляем *х,* получаем  (3).

Возьмем начальное приближение 



 (4) - итерационная формула метода простых итераций.

*Теорема.* Если функция определена и дифференцируема на отрезке , и выполняется условие , то итерационный процесс метода простых итераций, заданный формулой , сходится к единственному решению уравнения с любой степенью точности независимо от выбора начального приближения .

Из условия теоремы следует



Следовательно, если 1), то 

2) , то 

Пусть .

*Условие окончания итерационного процесса:*

Метод простых итераций для рассматриваемого примера

Так как производная , то значение выбирается из интервала

Выбрав наименьшее по модулю значение, получим

Выбираем произвольное значение из этого интервала. Пусть

. Тогда итерационная формула метода простых итераций будет иметь вид:

Итерационный процесс можно начать, задав произвольное начальное приближение .

Итерационный процесс заканчивается при одновременном выполнении двух условий:

и

Метод Ньютона.

Дано нелинейное уравнение (1’). На отрезке существует один корень . Нужно найти этот корень с точностью , т.е. .  и  сохраняют знак на  или .

Пусть - какое-либо приближение . Тогда  (2’), где - правка.

 (3’)

Подставляя (3’) в (2’) получим

 - итерационная формула метода Ньютона.

*Теорема.* Если функция  принимает значения разных знаков на концах отрезка ,  и  сохраняют определенный знак на , то исходя из начального приближения , удовлетворяющего условию, по методу Ньютона, заданного итерационной формулой , можно найти единственный корень уравнения  с любой степенью точности .

В качестве начального приближения выбираем правый или левый конец отрезка, в зависимости от того, в котором выполняется достаточное условие сходимости метода Ньютона вида:

*Условие окончания итерационного процесса:*

Метод Ньютона для рассматриваемого примера

Т.к. , то , из этого следует, что в качестве начального приближения выбираем

Итерационная формула метода Ньютона для данного уравнения запишется:

Итерационный процесс заканчивается при одновременном выполнении двух условий:

и

В этом случае значение  является приближенным значением корня нелинейного уравнения (1) на отрезке [].

*Модифицированный метод Ньютона.*

Если  мало меняется на отрезке , то можно положить, что, тогда мы получаем:

- итерационная формула модифицированного метода Ньютона.

В качестве начального приближения выбираем правый или левый конец отрезка, в зависимости от того, в котором выполняется достаточное условие сходимости метода Ньютона вида:

*Условие окончания итерационного процесса:*

Модифицированный метод Ньютона для рассматриваемого примера

Т.к. , то , из этого следует, что в качестве начального приближения выбираем

Итерационная формула модифицированного метода Ньютона для данного уравнения запишется так:

Итерационный процесс заканчивается при одновременном выполнении двух условий:

и

В этом случае значение  является приближенным значением корня нелинейного уравнения (1) на отрезке [].

*Листинг программы.*

#include <stdio.h>

#include <locale.h>

#include <math.h>

#include <iostream>

void MPI(float x0, float eps)

{

int n = 1;

float x, xn,dx, c = -1;

x = x0;

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf(" n+1 | X\_n | X\_n+1 | |X\_n+1 - X\_n| | |f(x\_n+1)| |\n"

"-----------+-------------+-------------+-----------------+---------------|\n");

do

{

xn = x + c \* (2\*sin(x)-x+0.5);

dx = xn-x;

printf("%10d | %12.7f|%13.8f|%17.8f|%15.8f|\n", n++, x, xn, fabs(dx), fabs(2\*sin(x)-x+0.5));

x = xn;

} while (fabs(dx) > eps || fabs(2\*sin(x)-x+0.5) > eps);

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf("\n");

}

void MN(float x0, float eps) {

int n = 1;

float x, xn, dx;

x = x0;

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf(" n+1 | X\_n | X\_n+1 | |X\_n+1 - X\_n| | |f(x\_n+1)| |\n"

"-----------+-------------+-------------+-----------------+---------------|\n");

if (((2\*sin(x)-x+0.5) \* (-2\*sin(x)) > 0)) {

do

{

if ((2\*cos(x)-1) != 0)

{

xn = x - ((2\*sin(x)-x+0.5) / (2\*cos(x)-1));

dx = xn-x;

printf("%10d | %12.7f|%13.8f|%17.8f|%15.8f|\n", n++, x, xn, fabs(dx), fabs(2\*sin(x)-x+0.5));

x = xn;

}

else {

printf("Значение производной в данной точке равно нулю\n");

}

} while (fabs(dx) > eps || fabs(2\*sin(x)-x+0.5) > eps);

}

else {

printf("Не выполняется достаточное условие сходимости метода Ньютона\n");

}

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf("\n");

}

void MMN(float x0, float eps) {

int n = 1;

float x, xn, dx;

x = x0;

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf(" n+1 | X\_n | X\_n+1 | |X\_n+1 - X\_n| | |f(x\_n+1)| |\n"

"-----------+-------------+-------------+-----------------+---------------|\n");

if (((2\*sin(x)-x+0.5) \* (-2\*sin(x)) > 0))

{

do

{

if ((2\*cos(x)-1) != 0) {

xn = x - ((2\*sin(x)-x+0.5) / (2\*cos(x)-1));

dx = xn-x;

printf("%10d | %12.7f|%13.8f|%17.8f|%15.8f|\n", n++, x, xn, fabs(dx), fabs(2\*sin(x)-x+0.5));

x = xn;

}

else {

printf("Значение производной в данной точке равно нулю\n");

}

} while (fabs(dx) > eps || (2\*sin(x)-x+0.5) > eps);

}

else {

printf("Не выполняется достаточное условие сходимости модифицированного метода Ньютона\n");

}

printf("-----------+-------------+-------------+-----------------+---------------|\n");

printf("\n");

}

int main() {

setlocale(LC\_CTYPE, "Russian");

float x0;

do {

printf("Введите начальное приближение X0 из отрезка [-2.5; -1.5]: ");

std::cin >> x0;

printf("\nНачальное приближение X0: %f Точность: 0.001\n\n", x0);

printf("\nМетод простых итераций\n\n");

MPI(x0, 0.001);

printf("\nМетод Ньютона\n\n");

MN(x0, 0.001);

printf("\nМодифицированный метод Ньютона\n\n");

MMN(x0, 0.001);

printf("\n\nНачальное приближение X0: %f Точность: 0.00001\n\n", x0);

printf("\nМетод простых итераций\n\n");

MPI(x0, 0.00001);

printf("\nМетод Ньютона\n\n");

MN(x0, 0.00001);

printf("\nМодифицированный метод Ньютона\n\n");

MMN(x0, 0.00001);

printf("\n\n");

} while (true);

return 0;

}

Вывод:

1. Наибольшую скорость сходимости имеет метод Ньютона.
2. Чем больше точность, тем меньше скорость сходимости.
3. Чем ближе к точному решению нелинейного уравнения выбрано начальное приближение, тем больше скорость сходимости

