**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по практической работе №2**

**по дисциплине «Вычислительная математика»**

**Тема: Изучение понятия обусловленности задачи нахождения корня.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Преподаватель |  | Попова Е.В. |

Санкт-Петербург

2022

## Цель работы.

## Исследование обусловленности задачи нахождения корня уравнения на примере линейной функции.

## Основные теоретические положения.

Под обусловленностью вычислительной задачи понимают чувствительность ее решения к малым погрешностям входных данных. Задачу называют хорошо обусловленной, если малым погрешностям входных данных отвечают малые погрешности решения, и плохо обусловленной, если возможны сильные изменения решения. Количественной мерой степени обусловленности вычислительной задачи является число обусловленности, которое можно интерпретировать как коэффициент возможного возрастания погрешностей в решении по отношению к вызвавшим их погрешностям входных данных. Пусть между абсолютными погрешностями входных данных x и решения y установлено неравенство:

,

где x\* и y\* - приближённые входные данные и приближённое решение соответственно. Тогда величина называется абсолютным числом обусловленности. Если же установлено неравенство

между относительными ошибками данных и решения, то величину называют относительным числом обусловленности. Для плохо обусловленной задачи 𝜈 ≫ 1. Грубо говоря, если , где 𝜈 − относительное число обусловленности, то порядок N показывает число верных цифр, которое может быть утеряно в результате по сравнению с числом верных цифр входных данных. Ответ на вопрос о том, при каком значении 𝜈 задачу следует признать плохо обусловленной, зависит, с одной стороны, от предъявляемых требований 3 к точности решения и, с другой, – от уровня обеспечиваемой точности исходных данных. Например, если требуется найти решение с точностью 0.1%, а входная информация задается с точностью 0.02%, то уже значение 𝜈 = 10 сигнализирует о плохой обусловленности. Однако, при тех же требованиях к точности результата, гарантия, что исходные данные задаются с точностью не ниже 0.0001%, означает, что при задача хорошо обусловлена. Если рассматривать задачу вычисления корня уравнения 𝑦 = 𝑓(𝑥), то роль числа обусловленности будет играть величина

,

где – корень уравнения.

## Задание.

Используя программы-функции BISECT и Round, исследовать обусловленность задачи нахождения корня уравнения 𝑓(𝑥) = 0. Значения функции 𝑓(𝑥) следует вычислить приближенно с точностью Δ, варьируемой в пределах от 0.1 до 0.000001. Порядок выполнения работы следующий:

1) Отделение корня уравнения 𝑓(𝑥) = 0.

2) Составление подпрограммы вычисления функции 𝑓(𝑥) для параметра с вводимая с клавиатуры.

3) Составление головной программы, вычисляющей корень уравнения с заданной точностью ε, и содержащую обращение к подпрограмме F, программам-функциям BISECT, Round и представление результатов.

4) Проведение вычислений по программе, варьируя значения параметров.

5) Анализ результатов.

1. Параметр с варьируется от а до в (допустимых). Параметры eps и delta постоянны и равны значению 0.01.

2. Параметр с постоянен и равен 5, eps постоянен и равен 0.01, delta варьируется от 0.00001 до 0.1.

3. Параметр с постоянен, delta постоянна и равна 0.01, eps варьируется от 0.000001 до 10.

4. Параметр с постоянен, delta и eps одновременно варьируются от 0.000001 до 1. Построить график зависимости eps от количества итераций.

5. Параметр eps постоянен и равен 0.01, c и delta варьируются независимым друг от друга образом.

## Выполнение работы.

*Вариант 28.*

1. Функция f(x) = 1/cos(cx) + 1 тригонометрическая и, следовательно,

является периодической, то есть значения функции через регулярные промежутки времени повторяются.

В область допустимых значений данной функции не входят значения,

принадлежащие интервалу (π/2c + πn/c), где n – целое число, так как функция cos(cx) принимает значение равное 0.

При с, принимающем значения от 3 до 3.5, корень находится на отрезке [3; 4].

Пример графика функции при с = 3 см. рис. 1. Корни , где n

целое число.

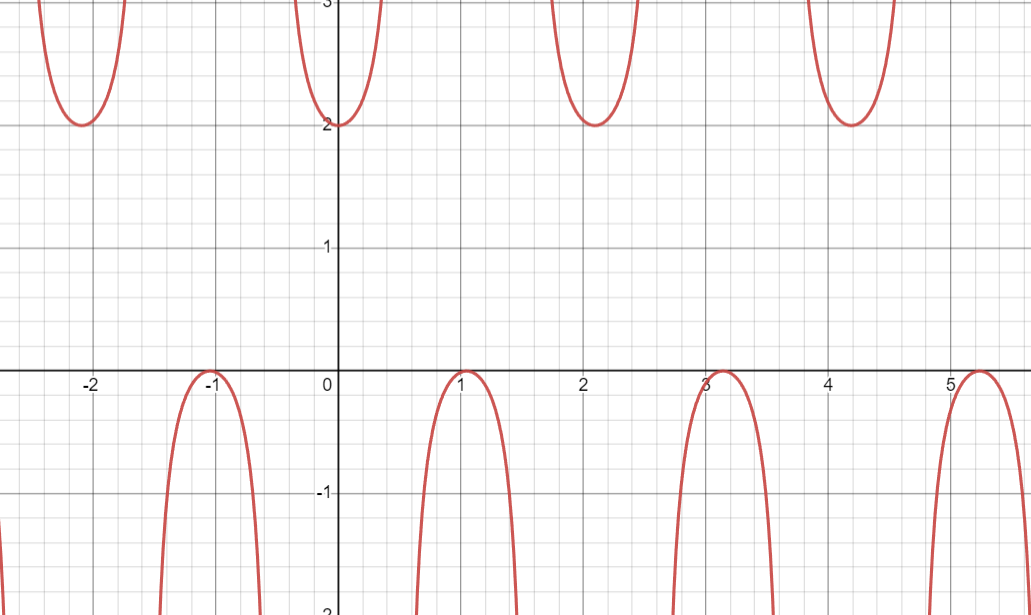


Рисунок 1 – график функции f(x) = 1/cos(cx) + 1 при с = 3.

1. Была составлена подпрограмма F для вычисления функции 𝑓(𝑥) для

параметра с вводимая с клавиатуры. Функция F принимает аргумент x и возвращает значение функции 1/cos(cx) + 1.

double F(double x){

return (1/(cos(c\*x)))+1;

}

1. Составлена головная программы - main, вычисляющей корень

уравнения с заданной точностью ε, считывание которой осуществляется с клавиатуры, и содержащую обращение к подпрограмме F и программам-функциям BISECT, Round и осуществляющая вывод результатов на экран. Разработанный код представлен в приложении.

1. Проведение вычислений и анализ.
2. Параметр с варьируется от 3 до 3.4. Параметры eps и delta

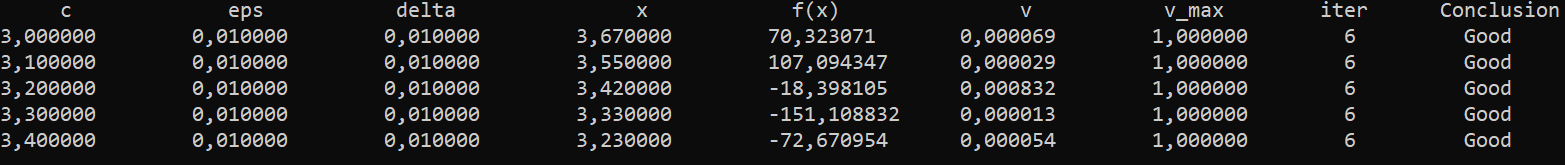
постоянны и равны 0.01.

Рисунок 2 – вычисления по пункту 1.

Вывод: При любом значении c от 3 до 3.5 и постоянных значениях eps и delta равных 0.01 функция плохо обусловлена.

При увеличении значения параметра c nu тоже увеличивается.

1. Параметр с постоянен и равен 5, eps постоянен и равен 0.01, delta

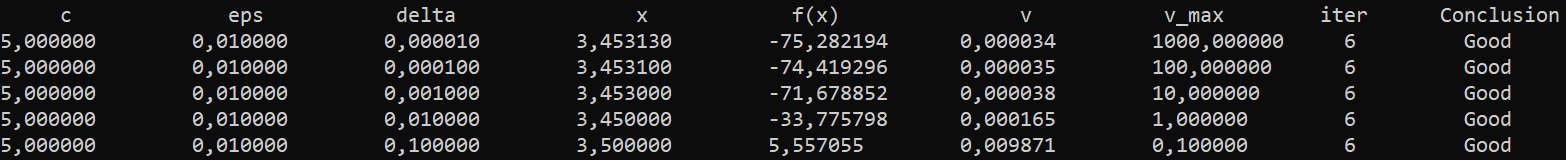
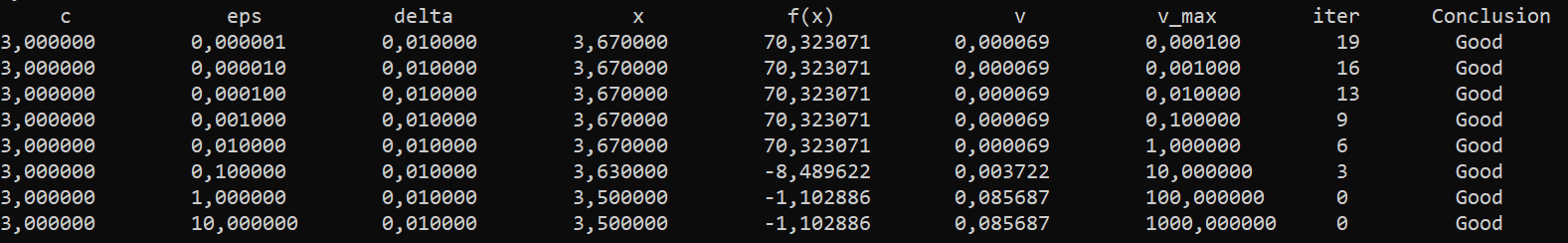
варьируется от 0.00001 до 0.1.

Рисунок 3 – вычисления по условиям из пункта 2.

Вывод: При постоянных параметрах c и delta равных 5 и 0.01 соответственно и увеличении значения параметра delta от 0.00001 до 0.1 точность функции v уменьшается.

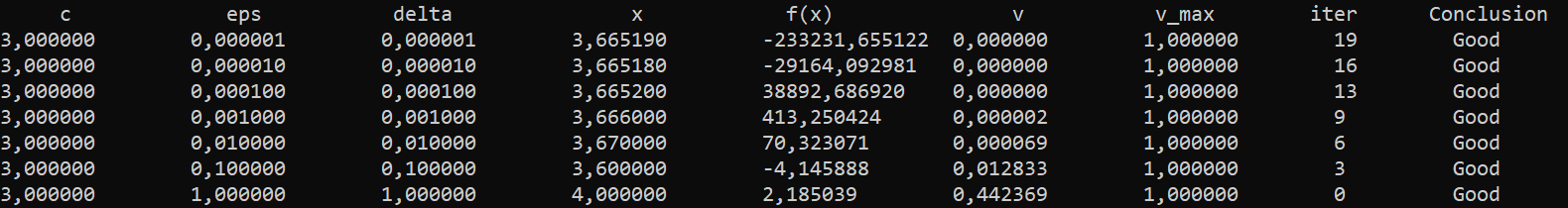
1. Параметр с постоянен и равен 3, delta постоянна и равна 0.01, eps

варьируется от 0.000001 до 10.

Рисунок 4 – вычисления по условиям из пункта 3.

Вывод: При значении с равном 3 и постоянном delta = 0.01 и eps варьирующемся от 0.000001 до 10 функция хорошо обусловлена.

1. Параметр с постоянен и равен 3, delta и eps одновременно

варьируются от 0.000001 до 1.

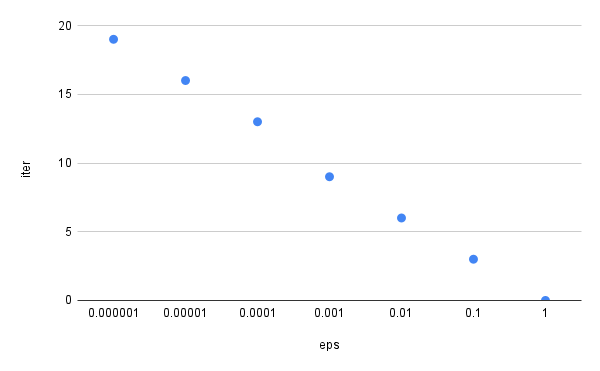
Рисунок 5 – вычисления по условиям из пункта 4.

Рисунок 6 – график зависимости eps от количества итераций.

Вывод: При постоянном значение с равном 3 и одновременном изменение параметров eps и delta от 0.000001 до 1 функция плохо обусловлена. Чем меньше значение eps, тем больше число итераций.

1. Параметр eps постоянен и равен 0.01, c и delta варьируются

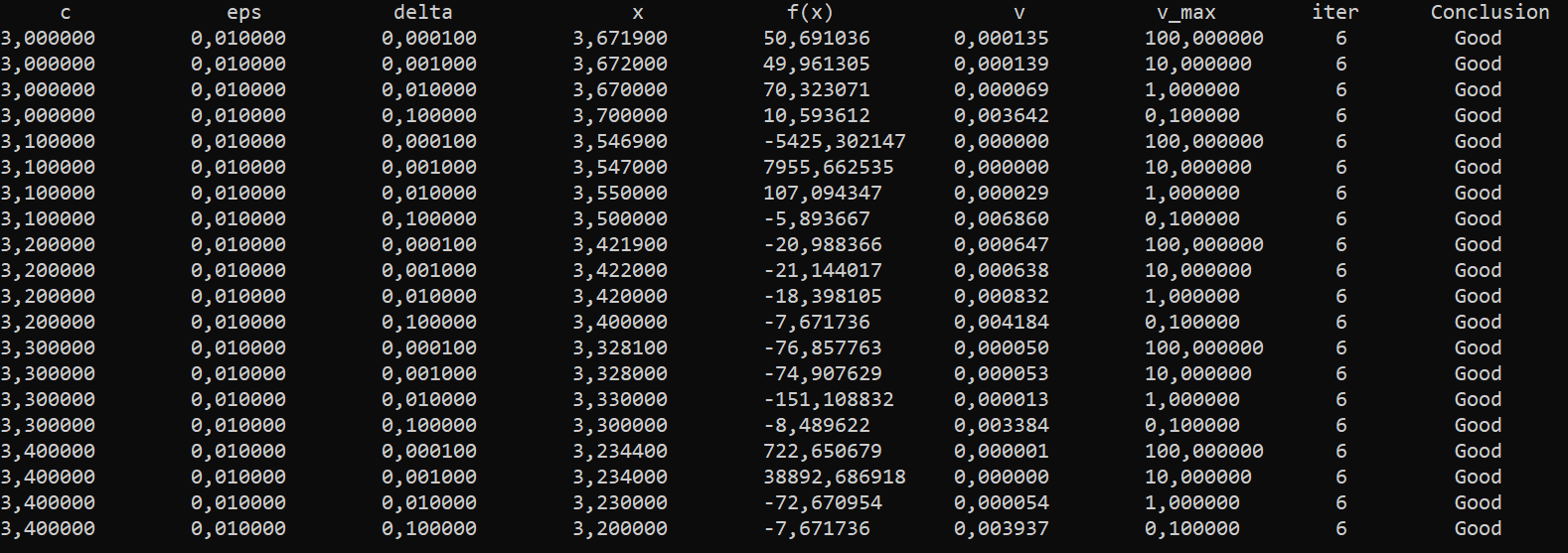
независимым друг от друга образом.

Рисунок – вычисления по условиям из пункта 5.

Вывод: При постоянном значение eps равном 0.01 и при различных значениях с и delta функция является хорошо обусловленной.

## Выводы.

В ходе выполнения практической работы была изучена обусловленность задачи нахождения корня уравнения 1/cos(cx) + 1 = 0.

При различных значениях eps и delta функция хорошо обусловлена.

# приложение

#include <iostream>

#include <cmath>

double delta, c;

double F(double x){

return (1/(cos(c\*x)))+1;

}

double BISECT(double Left, double Right, double Eps, int &N){

double E = fabs(Eps)\*2.0;

double FLeft = F(Left);

double FRight = F(Right);

double X = (Left + Right) / 2.0;

double Y;

if (FLeft\*FRight>0.0) {

puts("neverno zadan interval\n");

exit(1);

}

if (Eps <= 0.0) {

puts("neverno zadana tochnost\n");

exit(1);

}

N = 0;

if (FLeft == 0.0)

return Left;

if (FRight == 0.0)

return Right;

while ((Right - Left) >= E) {

X = 0.5\*(Right + Left);

Y = F(X);

if (Y == 0.0)

return (X);

if (Y\*FLeft < 0.0)

Right = X;

else {

Left = X;

FLeft = Y;

}

N++;

};

return(X);

}

double Round(double X, double Delta) {

if (Delta <= 1E-9){

puts("Неверно задана точность округления\n");

exit(1);

}

if (X>0.0)

return (Delta\*(long((X / Delta) + 0.5)));

else

return (Delta\*(long((X / Delta) - 0.5)));

}

int main(){

setlocale(LC\_ALL, "Russian");

printf("Пункт 1:\n");

double eps = 0.01;

double delta = 0.01;

int n = 0;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v", "v\_max", "iter", "Conclusion");

for(c = 3; c <=3.5 ; c += 0.1){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs((c\*sin(c\*x))/(pow(cos(c\*x), 2)));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

printf("\n\nПункт 2:\n");

c = 5;

eps = 0.01;

n = 0;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v", "v\_max", "iter", "Conclusion");

for(delta = 0.00001; delta <= 0.1; delta \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs((c\*sin(c\*x))/(pow(cos(c\*x), 2)));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

printf("\n\nПункт 3:\n");

c = 3;

delta = 0.01;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v", "v\_max", "iter", "Conclusion");

for(eps = 0.000001; eps <= 10; eps \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs((c\*sin(c\*x))/(pow(cos(c\*x), 2)));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

printf("\n\nПункт 4:\n");

c = 3;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v", "v\_max", "iter", "Conclusion");

for(eps = 0.000001, delta = 0.000001; eps <= 1 && delta <= 1; eps \*= 10, delta \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs((c\*sin(c\*x))/(pow(cos(c\*x), 2)));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

printf("\n\nПункт 5:\n");

eps = 0.01;

printf("%6s\t%14s\t%14s\t%14s\t%14s\t%14s\t%14s\t%10s\t%10s\n", "c", "eps", "delta", "x", "f(x)", "v", "v\_max", "iter", "Conclusion");

for(c = 3; c <= 3.5; c += 0.1){

for(delta = 0.0001; delta <= 0.1; delta \*= 10){

double x = Round(BISECT(3, 4, eps, n), delta);

double derivative = fabs((c\*sin(c\*x))/(pow(cos(c\*x), 2)));//вычисление абсолютного значения

double v = 1/derivative;

printf("%6f\t%6f\t%6f\t%6f\t%6f\t%6lf\t%6f\t%d\t", c, eps, delta, x, F(x), v, eps/delta, n);

if(v < eps/delta)

printf("%6s\n", "Good");

else

printf("%6s\n", "Bad");

}

}

return 0;

}