**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Вычислительная математика»**

Тема: **ИЗУЧЕНИЕ ПОНЯТИЯ ОБУСЛОВЛЕННОСТИ ВЫЧИСЛИТЕЛЬНОЙ ЗАДАЧИ**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Виноградов К.А. |
| Преподаватель |  | Лисс А.Р. |

Санкт-Петербург

2017

**Цель работы.**

Исследовать обусловленность задачи нахождения корня уравнения для 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. Если между относительными ошибками данных и решения установлено неравенство δ(y\*) ≤ νδ δ(x\*) - νδ называют относительным числом обусловленности. Для плохо обусловленной задачи ν>>1.

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 ν=103 задача хорошо обусловлена.

Если рассматривать задачу вычисления корня уравнения y=f(x), то роль числа обусловленности будет играть величина

, где x0 - корень уравнения.

**Постановка задачи.**

Используя подпрограммы-функции BISECT и Round из файла methods.cpp, исследовать обусловленность задачи нахождения корня уравнения f(x)=0 для линейной функции f(x)=c(x-d). Значения функции f(x) следует вычислить приближенно с точностью Delta, варьируемой в пределах от 0.1 до 0.000001.

**Выполнение работы.**

Допустим в линейной функции f(x)=c(x-d) d=5,3535.

1. Аналитически найдем отрезки [Left, Right], на которых функция f(x) удовлетворяет условиям применимости метода бисекции. Функция f(x) непрерывная, линейная, при c≠0 имеет единственный нуль, равный d. Поэтому f(x) удовлетворяет условиям применимости метода бисекции на любом отрезке [Left, Right], при условии что отрезок содержит d. Пусть отрезок [Left, Right] = [2,7].
2. Составим подпрограмму вычисления функции f(x)=c(x-d) для параметров c и d, вводимых с клавиатуры (см прил. А). Предусмотрено округление вычисленных значений функции f(x) с использованием подпрограммы-функции Round с точностью Delta, также вводимой с клавиатуры.
3. Составим головную программу, вычисляющую корень уравнения с заданной точностью Eps и содержащую обращение к подпрограмме f(x), подпрограммам-функциям BISECT, Round и представлению результатов(см прил. А). Реализован вывод результатов вычислений в файл result.txt.
4. Проведем вычисления по программе, варьируя значения параметров  
   c (тангенс угла наклона прямой, см. табл. 1 и рис. 1), Eps (точность вычисления корня, см. табл. 2 и рис. 2) и Delta (точность задания исходных данных, см. табл. 3 и рис. 3).

Формула для вычисления числа обусловленности 

Имеем, f(x)=c(x-d) т.е. . Для плохо обусловленной задачи ν>>1.

Таблица 1 – Значения с при постоянных значениях Eps и Delta

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Eps | Delta | c | x | k |  | Оценка обусловленности |
| 0.0001 | 0.00001 | 100 | 5.353424 | 15 | 0.01 | Хорошо |
| 0.0001 | 0.00001 | 10 | 5.353424 | 15 | 0.1 | Хорошо |
| 0.0001 | 0.00001 | 1 | 5.353424 | 15 | 1 | Хорошо |

Продолжение таблицы 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0.0001 | 0.00001 | 0.1 | 5.353424 | 15 | 10 | Хорошо |
| 0.0001 | 0.00001 | 0.01 | 5.353271 | 11 | 100 | Плохо |
| 0.0001 | 0.00001 | 0.001 | 5.349609 | 8 | 1000 | Плохо |
| 0.0001 | 0.00001 | 0.0001 | 5.359375 | 5 | 10000 | Плохо |
| 0.0001 | 0.00001 | 0.00001 | 5.750000 | 1 | 100000 | Плохо |
| 0.0001 | 0.00001 | 0.000001 | 2.000000 | 0 | 1000000 | Плохо |

Рисунок 1 – Зависимость x и k от c

Таблица 2 – Значения Delta при постоянных значениях Eps и с

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | Delta | c | x | k |
| 0.0001 | 0.1 | 10 | 5.349609 | 8 |
| 0.0001 | 0.01 | 10 | 5.353271 | 11 |
| 0.0001 | 0.001 | 10 | 5.353424 | 15 |
| 0.0001 | 0.0001 | 10 | 5.353424 | 15 |
| 0.0001 | 0.00001 | 10 | 5.353424 | 15 |
| 0.0001 | 0.000001 | 10 | 5.353424 | 15 |

Рисунок 2 – Зависимость x и k от delta

Таблица 3 *–* Значения Eps при постоянных значениях Delta и с

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eps | Delta | c | x | k |
| 0.1 | 0.001 | 10 | 5.281250 | 5 |
| 0.01 | 0.001 | 10 | 5.339844 | 8 |
| 0.001 | 0.001 | 10 | 5.353271 | 12 |
| 0.0001 | 0.001 | 10 | 5.353271 | 11 |
| 0.00001 | 0.001 | 10 | 5.353500 | 15 |
| 0.000001 | 0.001 | 10 | 5.353500 | 15 |

Рисунок 3 – Зависимость x и k от eps

**Выводы.**

Теоретическое значение корня равно d=5.3535. Были проанализированы таблицы и сравнены экспериментальные значения корня с теоретическим. Можно сделать следующий вывод: корень вычисляется с большей точностью, когда точность исходных данных и точность вычисления отличаются незначительно и при этом погрешность входных данных мала (в данном случае от 0,001 до 0,000001). Близкие значения точности входных и выходных данных и свидетельствуют о хорошей обусловленности задачи.

Приложение А

Код программы

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <iostream>

#include <conio.h>

#include "methods.h"

double Round (double X,double Delta)

{

if (Delta<=1E-9) {puts("Wrong rounding accuracy\n");exit(1);}

if (X>0.0) return (Delta\*(long((X/Delta)+0.5)));

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

}

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("False interval\n");exit(1);}

if (Eps<=0.0) {puts("Wrong accuracy\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); /\* вычисление середины отрезка \*/

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 delta,c,d;

int main()

{

int k;

long int s;

float a1,b1,c1,d1,eps1,delta1;

double a,b,eps,x;

double F(double);

FILE\* pFile;

pFile = fopen("result.txt", "w");

printf("Insert d:");

scanf("%f",&d1);

d = d1;

printf("Insert a:");

scanf("%f",&a1);

a = a1;

printf("Insert b:");

scanf("%f",&b1);

b = b1;

printf("Insert eps:");

scanf("%f",&eps1);

eps = eps1;

printf("Insert c:");

scanf("%f",&c1);

c = c1;

printf("Insert delta:");

scanf("%f",&delta1);

delta = delta1;

x = BISECT(a,b,eps,k);

fprintf(pFile, "x=%f k=%d\n",x,k);

printf("Results are sucsessfully written in the file");

}

double F(double x)

{

extern double c,d,delta;

double s;

long int S;

s = c\*(x - d);

if( s/delta < 0 )

S = s/delta - .5;

else

S = s/delta + .5;

s = S\*delta;

s = Round( s,delta );

return(s);

}