**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе №2**

**по дисциплине «Вычислительная математика»**

Тема: **Изучение понятия обусловленности вычислительной задачи**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7383 |  | Кирсанов А.Я. |
| Преподаватель |  | Сучков А.И. |

Санкт-Петербург

2018

**Цель работы.**

Используя программы-функции BISECT и Round исследовать обусловленность задачи нахождения корня уравнения (1)

 (1)

для линейной функции (2)

. (2)

Сравнить полученные данные с теоретическими.

**Основные теоретические положения.**

Задачу называют хорошо обусловленной, если малым погрешностям входных данных отвечают малые погрешности решения, и плохо обусловленной, если возможны сильные изменения решения.

Пусть между абсолютными погрешностями входных данных  и решения установлено неравенство (3):

 (3)

где и  – приближенное решение и приближенные входные данные. Тогда величина  называется абсолютным числом обусловленности.

Если же установлено неравенство (4):

 (4)

между относительными ошибками данных и решения, то величину  называют относительным числом обусловленности. Для плохо обусловленной задачи

>>1.

Если рассматривать задачу вычисления корня уравнения ,то роль числа обусловленности будет играть величина (5):

 (5)

где - корень уравнения.

**Постановка задачи.**

1) Аналитически отделить корень уравнения (1), то есть найти отрезки [*Left, Right*], на которых функция удовлетворяет условиям применимости метода бисекции.

2) Составить подпрограмму вычисления корня функции (2) для параметров *c* и *d*, вводимых с клавиатуры. Предусмотреть округление вычисленных значений функции (2) с использованием программы-функции Round с точностью Delta, также вводимой с клавиатуры.

3) Составить головную программу, вычисляющую корень уравнения с заданной точностью Eps и содержащую обращение к подпрограмме , программам-функциям BISECT, Round и представление результатов.

4) Провести вычисления по программе, варьируя значения параметров  (тангенс угла наклона прямой), Eps (точность вычисления корня) и Delta (точность задания исходных данных).

5) Проанализировать полученные результаты и обосновать выбор точности Eps вычисления корня. Сопоставить полученные теоретические результаты с экспериментальными данными.

**Выполнение работы.**

Листинги программы, используемых функций и заголовочного файла представлены в приложениях А, Б, В соответственно.

Результаты работы программы и сравнение их с теоретическими данными даны в табл. 1. В уравнении (2) выбрано *d* = 2.99999, левая граница *a* = -5, правая граница *b* = 5.

В качестве теоретических данных выступает число обусловленности -, рассчитываемое по формуле (5).

Обусловленность задачи определяется выражением (3), в котором  = Eps, а  = Delta.

Таблица 1 – Тестирование программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Значение Eps | Значение *c* | Значение Delta | Экспериментальные данные | | Теоретические данные |
| Значение  *x* | Обусловленность  задачи |  |
| 0.01 | 20 | 0.1 | 2.988281 | Плохо | 0.05 |
| 0.1 | 1 | 0.1 | 2.968750 | Хорошо | 1 |
| 0.0001 | 10 | 0.1 | 2.998047 | Плохо | 0.1 |
| 0.1 | 0.1 | 0.01 | 2.968750 | Хорошо | 10 |
| 0.001 | 0.1 | 0.01 | 2.968750 | Плохо | 10 |
| 0.1 | 200 | 0.1 | 2.968750 | Хорошо | 0.005 |
| 0.1 | 0.5 | 0.1 | 2.968750 | Хорошо | 2 |
| 0.1 | 0.5 | 0.001 | 2.968750 | Хорошо | 2 |
| 0.0001 | 200 | 0.1 | 2.999878 | Хорошо | 0.005 |
| 0.0001 | 0.1 | 0.1 | 2.500000 | Плохо | 10 |

**Выводы.**

Проанализированы результаты, полученные программой (см. табл. 1) и произведен анализ обусловленности задачи с помощью выражения (3). Для анализа было использовано абсолютное число обусловленности . По результатам анализа задача хорошо обусловлена, когда верно неравенство из выражения (3). Обнаружено, что если уравнение линейное (2), то на значение  влияет только коэффициент *с.*

Приложение А

Листинг программы

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include "methods.h"

double delta,c,d;

int main()

{

int k;

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

double a,b,eps,x;

double f(double);

printf("введите eps:");

scanf("%f",&eps1);

eps = eps1;

printf("введите c:");

scanf("%f",&c1);

c = c1;

printf("введите d:");

scanf("%f",&d1);

d = d1;

printf("введите a:");

scanf("%f",&a1);

a = a1;

printf("введите b:");

scanf("%f",&b1);

b = b1;

printf("введите delta:");

scanf("%f",&delta1);

delta = delta1;

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

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

}

Приложение Б

Используемые функции

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <iostream>

#include "METHODS.H"

using namespace std;

double Round (double X,double Delta)

{

if (Delta<=1E-9) {puts("Неверное задание точности округления\n");exit(1);}

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

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

}

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);

}

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("Неверное задание интервала\n");exit(1);}

if (Eps<=0.0) {puts("Неверное задание точности\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;

}

Приложение В

Листинг заголовочного файла

extern double F(double);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Функция F (X), задаваемая пользователем \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

double Round (double X,double Delta);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Функция Round (X,Delta) , предназначена для округления \*/

/\* X с точностью Delta \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Функция BISECT предназначена для решения уравнения F(X)=0 \*/

/\* методом деления отрезка пополам. Использованы обозначения: \*/

/\* Left - левый конец промежутка \*/

/\* Right - правый конец промежутка \*/

/\* Eps - погрешность вычисления корня уравнения; \*/

/\* N - число итераций \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/