# Практическая работа №2

**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.

**Постановка задачи.** 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.

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)));

**Варианты**

|  |  |
| --- | --- |
| № | f(x) |
| 32 | sin(cx) |
| 31 | c\*sin(x) |
| 30 | 1/(sin(cx))+1 |
| 29 | 1/(c\*sin(x))+1 |
| 28 | 1/(cos(cx))+1 |
| 27 | 1/(c\*cos(x))+1 |
| 26 | cos(cx) |
| 25 | c\*cos(x) |
| 24 | tg(cx) |
| 23 | с\*tg(x) |
| 22 | ctg(x) |
| 21 | c\*ctg(x) |
| 20 | arcsin((1/c)\*x)+0.5 |
| 19 | 1/c\*arcsin(x)+0.1 |
| 18 | arccos(x/c)-3 |
| 17 | 1/c\*arccos(x)-0.5 |
| 16 | arctg(cx+1) |
| 15 | c\*arctg(x)+1 |
| 14 | arcctg(cx)-3 |
| 13 | c\*arcctg(x)-3 |
| 12 | sh(cx)+1 |
| 11 | c\*sh(x+1) |
| 10 | ch(cx)-6 |
| 9 | c\*ch(x)-6 |
| 8 | th(cx-2) |
| 7 | c\*th(x)-2 |
| 6 | cth(cx)-2 |
| 5 | c\*cth(x)-6 |
| 4 | sch(cx)-0.5 |
| 3 | c\*sch(x)-0.5 |
| 2 | csch(cx)-0.5 |
| 1 | c\*csch(x)-0.5 |