

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 вычислительной
задачи

Студент гр. 0304

Преподаватель

Крицын Д. Р.

Попова Е. В.

Санкт-Петербург

2021

Вариант 12.

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:

$$\Delta(y^*) \leq v_{\Delta} \cdot \Delta(x^*),$$

где x^* и y^* - приближённые входные данные и приближённое решение соответственно. Тогда величина v_{Δ} называется абсолютным числом обусловленности. Если же установлено неравенство

$$\delta(y^*) \leq v_{\delta} \cdot \delta(x^*)$$

между относительными ошибками данных и решения, то величину v_{δ} называют относительным числом обусловленности. Для плохо обусловленной задачи $v_{\Delta} \gg 1$. Грубо говоря, если $v_{\delta} = 10^N$, где v_{δ} —

относительное число обусловленности, то порядок v 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 v задачу следует признать плохо обусловленной, зависит, с одной стороны, от предъявляемых требований 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 $v = 10$ 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 $v=10^3$ задача хорошо обусловлена. Если рассматривать задачу вычисления корня уравнения $y = f(x)$, то роль числа обусловленности будет играть величина

$$v_{\Delta} = \frac{1}{|f'(x^0)|},$$

где x^0 – корень уравнения.

Постановка задачи.

Используя программы-функции BISECT и Round, 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 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 $f(x) = c(x - d)$ для параметров c и d вводимых 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.

Выполнение работы.

1. Была разработана и реализована программа для произведения требуемых вычислений. Были созданы функции *double f(double)* для вычисления значения исследуемой функции $f(x)$ и *double f_conditioning()* для вычисления числа обусловленности функции $f(x)$. *f_conditioning()* не принимает на вход параметров, т.к. производная линейной функции $f(x)$ постоянна.
2. Была реализована головная функция, вызывающая *bisect()*, *_round()* и *f_conditioning()* для вычисления корня, округления полученного корня для заданного Δ и вычисления обусловленности для $f(x)$.
3. Проведём вычисления при помощи программы для всех вариаций входных значений c , ε и Δ :

1.

c	eps	delta	x0	NΔ	NΔ max
0.010000	0.010000	0.010000	15.000000	100.000000	1.000000
0.100000	0.010000	0.010000	15.000000	10.000000	1.000000
1.000000	0.010000	0.010000	15.000000	1.000000	1.000000
10.000000	0.010000	0.010000	15.000000	0.100000	1.000000
100.000000	0.010000	0.010000	15.000000	0.010000	1.000000
1000.000000	0.010000	0.010000	15.000000	0.001000	1.000000

Рисунок 1. Параметр c варьируется от 0.01 до 1000. Параметры ϵ и δ постоянны и равны значению 0.01.

Вывод: обусловленность вычислительно задачи обратно пропорционально зависит от значения производной исследуемой функции в точке корня функции, в данном случае это значение всегда равно параметру c . При довольно большом значении c задача считается хорошо обусловленной ($c > 1$) с данными параметрами ϵ , Δ .

2.

c	eps	delta	x0	NΔ	NΔ max
1.000000	0.010000	0.000010	15.000000	1.000000	1000.000000
1.000000	0.010000	0.000100	15.000000	1.000000	100.000000
1.000000	0.010000	0.001000	15.000000	1.000000	10.000000
1.000000	0.010000	0.010000	15.000000	1.000000	1.000000
1.000000	0.010000	0.100000	15.000000	1.000000	0.100000

Рисунок 2. Параметр c постоянен и равен 1, ϵ постоянен и равен 0.01, δ варьируется от 0.00001 до 0.1.

Вывод: обусловленность задачи зависит от параметра Δ (т.е. от точности вычисления x_0). Чем меньше Δ , тем лучше обусловлена задача — $v_{\Delta \max}$ обратно пропорционально зависит от Δ , а из-за постоянного параметра c v_{Δ} не меняется.

3.

c	eps	delta	x0	NΔ	NΔ max
10.000000	0.000010	0.010000	15.000000	0.100000	0.001000
10.000000	0.000100	0.010000	15.000000	0.100000	0.010000
10.000000	0.001000	0.010000	15.000000	0.100000	0.100000
10.000000	0.010000	0.010000	15.000000	0.100000	1.000000
10.000000	0.100000	0.010000	15.000000	0.100000	10.000000
10.000000	1.000000	0.010000	15.000000	0.100000	100.000000
10.000000	10.000000	0.010000	15.000000	0.100000	1000.000000

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

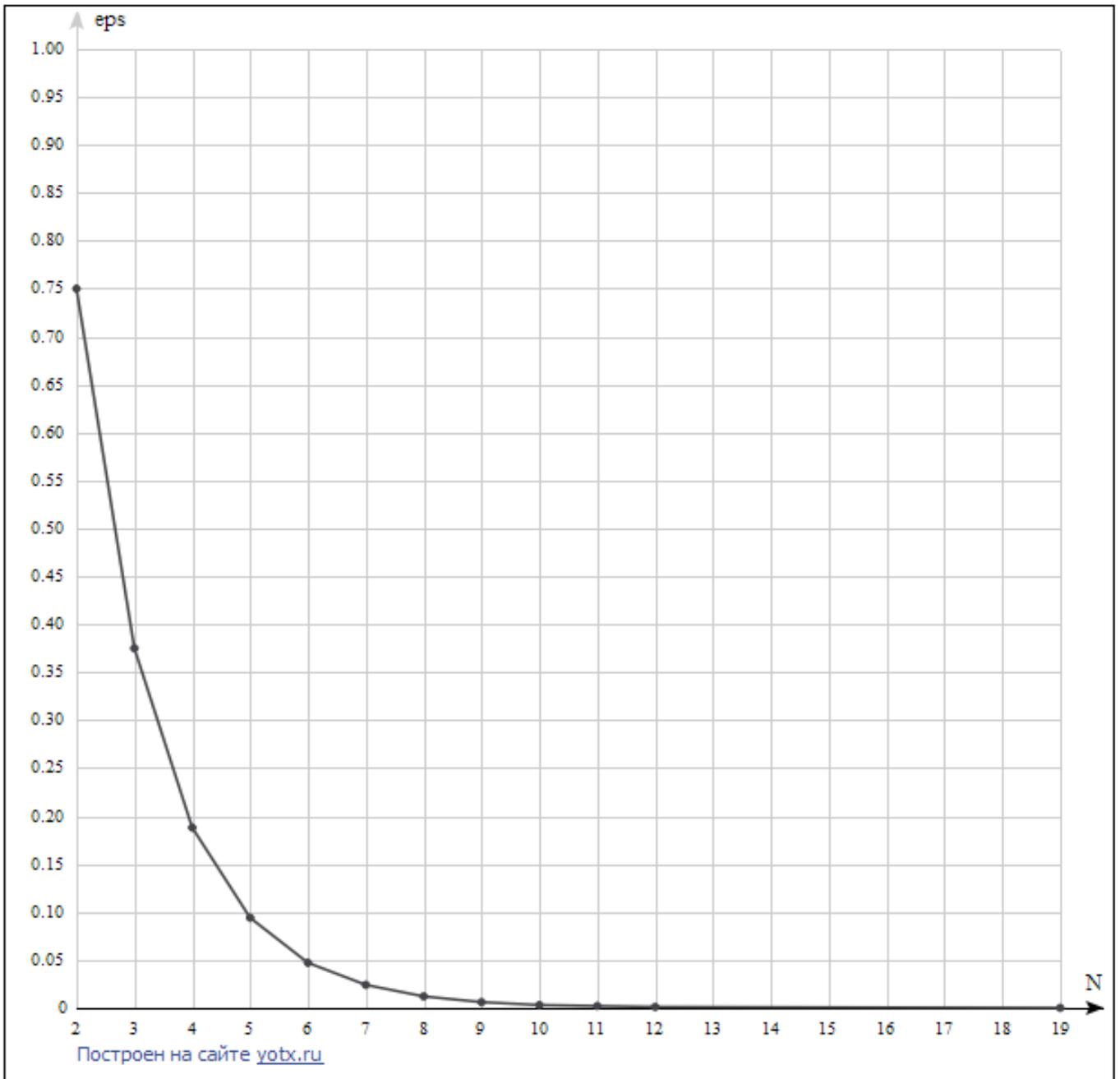
Вывод: обусловленность задачи зависит также от параметра ε — минимального интервала, на котором функция отсечения корня прерывает свою работу. Чем больше ε , тем лучше обусловлена задача - $v_{\Delta \max}$ прямо пропорционально ε , а v_{Δ} , опять же, не меняется ($c = 10$).

4.

c	eps	delta	x0	NΔ	NΔ max
10.000000	0.000010	0.000010	15.000000	0.100000	1.000000
10.000000	0.000100	0.000100	15.000000	0.100000	1.000000
10.000000	0.001000	0.001000	15.000000	0.100000	1.000000
10.000000	0.010000	0.010000	15.000000	0.100000	1.000000
10.000000	0.100000	0.100000	15.000000	0.100000	1.000000
10.000000	1.000000	1.000000	15.000000	0.100000	1.000000

Рисунок 4. Параметр c постоянен и равен 10, delta и eps одновременно варьируются от 0.000001 до 1.

Вывод: при одновременном изменении параметров ε , Δ $v_{\Delta \max}$ остаётся постоянным, т.е. зависимость максимального числа обусловленности $v_{\Delta \max}$ от ε , Δ отличается лишь «направлением» зависимости ($\varepsilon \setminus \Delta$ прямо\обратно пропорциональны $v_{\Delta \max}$ соответственно).



Видимо, зависимость $N(\epsilon)$ приблизительно выражена функцией $\log(1/x)$. Это происходит из-за того, что с увеличением ϵ увеличивается минимальная длина отрезка функции bisect, и вследствие уменьшается количество делений отрезка (итераций алгоритма отсечения корня).

5.

c	eps	delta	x0	NA	NA max
0.010000	0.010000	0.000010	15.000000	100.000000	1000.000000
0.010000	0.010000	0.000100	15.000000	100.000000	100.000000
0.010000	0.010000	0.001000	15.000000	100.000000	10.000000
0.010000	0.010000	0.010000	15.000000	100.000000	1.000000
0.010000	0.010000	0.100000	15.000000	100.000000	0.100000
0.100000	0.010000	0.000010	15.000000	10.000000	1000.000000
0.100000	0.010000	0.000100	15.000000	10.000000	100.000000
0.100000	0.010000	0.001000	15.000000	10.000000	10.000000
0.100000	0.010000	0.010000	15.000000	10.000000	1.000000
0.100000	0.010000	0.100000	15.000000	10.000000	0.100000
1.000000	0.010000	0.000010	15.000000	1.000000	1000.000000
1.000000	0.010000	0.000100	15.000000	1.000000	100.000000
1.000000	0.010000	0.001000	15.000000	1.000000	10.000000
1.000000	0.010000	0.010000	15.000000	1.000000	1.000000
1.000000	0.010000	0.100000	15.000000	1.000000	0.100000
10.000000	0.010000	0.000010	15.000000	0.100000	1000.000000
10.000000	0.010000	0.000100	15.000000	0.100000	100.000000
10.000000	0.010000	0.001000	15.000000	0.100000	10.000000
10.000000	0.010000	0.010000	15.000000	0.100000	1.000000
10.000000	0.010000	0.100000	15.000000	0.100000	0.100000
100.000000	0.010000	0.000010	15.000000	0.010000	1000.000000
100.000000	0.010000	0.000100	15.000000	0.010000	100.000000
100.000000	0.010000	0.001000	15.000000	0.010000	10.000000
100.000000	0.010000	0.010000	15.000000	0.010000	1.000000
100.000000	0.010000	0.100000	15.000000	0.010000	0.100000
1000.000000	0.010000	0.000010	15.000000	0.001000	1000.000000
1000.000000	0.010000	0.000100	15.000000	0.001000	100.000000
1000.000000	0.010000	0.001000	15.000000	0.001000	10.000000
1000.000000	0.010000	0.010000	15.000000	0.001000	1.000000
1000.000000	0.010000	0.100000	15.000000	0.001000	0.100000

Рисунок 6. Параметр eps постоянен и равен 0.01, c и delta варьируются независимым друг от друга образом.

Вывод: для того, чтобы задача была хорошо обусловлена, нужно, чтобы c и Δ лежали в определённых пределах — c должно быть достаточно велико, и Δ достаточно мало.

Выводы.

Таким образом, обусловленность задачи в основном зависит от соотношения трёх параметров — c , Δ и ε . Параметр c напрямую влияет на значение производной рассматриваемой линейной функции во всех её точках, а значит и на число обусловленности, а параметры Δ и ε — на максимальное значение числа обусловленности. При этом все эти параметры должны лежать в определённых пределах относительно друг друга, иначе значение одного из чисел обусловленности будет слишком большим для того, чтобы назвать задачу обусловленной.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <stdio.h>
#include <math.h>
#include <errno.h>

#define ERRNO_BISECT_INTERVAL 1
#define ERRNO_BISECT_PRECISION 2

#define ERRNO_ROUND_PRECISION 2

double bisect(double (*f)(double),
              double left, double right,
              double eps, int* N)
{
    double e = fabs(eps) * 2;
    double fleft = f(left);
    double fright = f(right);
    double x = (left + right) / 2, y;

    if(fleft*fright > 0){
        errno = ERRNO_BISECT_INTERVAL;
        return NAN;
    }
    if(eps <= 0){
        errno = ERRNO_BISECT_PRECISION;
        return NAN;
    }

    *N = 0;

    if(fleft == 0) return left;
    if(fright == 0) return right;

    while((right - left) >= e)
    {
        x = (left + right) / 2;
        y = f(x);
        if(y == 0)
            return x;
    }
}
```

```

        if(y*fleft < 0)
            right = x;
        else {
            left = x;
            fleft = y;
        }

        (*N)++;
    }

    return x;
}

double _round(double x, double delta)
{
    if(delta <= 1e-9){
        errno = ERRNO_ROUND_PRECISION;
        return NAN;
    }
    return delta * floor((x / delta) + (x > 0 ? 0.5 : -0.5));
}

#define d 15

double f_c;
double f(double x)
{
    return f_c*(x - d);
}

double f_conditioning() // функция для вычисления обусловленности корня x0 функции f
{
    return 1 / fabs(f_c);
}

int main()
{
    int N;

    puts("1. Параметр c варьируется от 0.01 до 1000. Параметры eps и delta
    постоянны и равны значению 0.01.\n");
}

```



```

{
    double x0 = bisect(&f, d - 2, d + 2, eps, &N);
    x0 = _round(x0, delta);
    double cond = f_conditioning();
    printf("%lf\t\t%lf\t\t%lf\t\t%lf\t\t%lf\t\t%lf\n", f_c, eps, delta, x0,
cond, eps/delta);
}

puts("\n5. Параметр eps постоянен и равен 0.01, c и delta варьируются
независимым друг от друга образом.\n");
puts("c\t\t\teps\t\t\tdelta\t\t\tx0\t\t\tNΔ\t\t\tNΔ max");
f_c = 0.01;
for(f_c = 0.01; f_c <= 1000; f_c *= 10)
for(double delta = 0.00001; delta <= 0.1; delta *= 10)
{
    double x0 = bisect(&f, d - 2, d + 2, 0.01, &N);
    x0 = _round(x0, delta);
    double cond = f_conditioning();
    printf("%lf\t\t%lf\t\t%lf\t\t%lf\t\t%lf\t\t%lf\n", f_c, 0.01, delta, x0,
cond, 0.01/delta);
}
}

```