

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки
Кафедра мікроелектроніки

Лабораторна робота №4

Варіант №21

Виконав: студент групи ДП-82
Мнацаканов Антон
Перевірив: Домбругов М.Р.

Київ-2020

**Розв'язання нелінійних рівнянь
з одним невідомим.
Методи поділу навпіл (бісекції) та хорд**

Мета роботи: вивчення алгоритмів і налаштування програм для розв'язання нелінійних рівнянь методом поділу навпіл (бісекції) і методом хорд.

Що зробити: знайти корені рівняння $f(x) = 0$ методом бісекції. Впевнитись, що їх значення узгоджуються з результатами аналітичного дослідження функції $f(x)$. Визначити порядок збіжності методу бісекції. Додатково провести аналогічні дослідження методу хорд.

1. Уясніть призначення окремих блоків схеми алгоритму для розв'язання рівняння виду $f(x) = 0$ методом бісекції. Складіть програму, що реалізує цей алгоритм. Фрагмент програми, що власне розв'язує рівняння, оформте у вигляді окремої процедури. Передбачте в ній додатковий вихідний параметр – код помилки. Йому буде присвоєно одне певне значення (скажімо, 0), якщо процедура розв'язання пройшла успішно і помилок немає, і інше (скажімо, -1) якщо розв'язок не здобуто.
2. Після блоку 2 введіть додаткову перевірку правильності вибору початкового інтервалу пошуку – прийнятого припущення про те, що функція $f(x)$ на кінцях початкового інтервалу має різні знаки.
3. Візьміть ту ж саму функцію $f(x)$, яку ви досліджували при виконанні лабораторної роботи № 3. За допомогою вашої програми знайдіть її найменший за модулем ненульовий корінь. Початковий інтервал пошуку кореня виберіть самостійно.
4. З метою налагодження програми і усвідомлення деталей роботи алгоритму введіть в програму після блоку 3 проміжний друк номера ітерацій i , а також значень a , x , b , $|b - a|$, $f(x)$ на кожній ітерації. Потурбуйтеся, щоб результати, що виводяться, мали вигляд охайної таблиці.
5. Дослідіть, як похибки поточного наближення до кореня $e(i) = |b - a|$ залежать від номера ітерації i . Побудуйте графік залежності $\lg e(i)$ від i . На основі цих даних впевніться, що порядок збіжності методу бісекції дорівнює 1.
6. Задавайте $\varepsilon = 10^{-5}$, 10^{-6} , 10^{-7} , ... Зменшуйте ε доти, доки програма не почне зациклюватися. Порівняйте цю величину з величиною машинного епсілона.
7. Знайдіть решту коренів рівняння $f(x) = 0$.

Хід роботи

Для методу бісекції $e^{(i+1)} = \frac{1}{2}e^{(i)}$ і його порядок збіжності дорівнює 1.

Будем користуватися цим правилом для перевірки виведених значень.

Фрагмент коду на С для методу бісекції:

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

int main()
{
double c = 0.00000001;
double a = 0.5;
double b = 2;
double x;
double y1 = 10*a*a*exp(-a)-3*a;
double y;
double e;
int i=0;
double e1;
printf("i\t\tte\tte1\t\ttx\t\tproverka\n");
double proverka;
do
{
x = a + (b-a)/2;
y = 10*x*x*exp(-x)-3*x;
if ((y1>0 && y>0)||(y1<0 && y<0))
{
a=x;
}
else
{
b=x;
}
e = fabs(b-a);
i++;
e1=0.5*e;
printf ("%d\t\tt0%e\t0%e", i, e, x);
```

```

    printf ("\t%e", e1);
    proverka=e/e1;
    printf ("\t%e\n", proverka);
}
while (fabs(e)>=c);
x = a + (b-a)/2;
printf ("\n\n\n%e\t\t", x );
return 0;
}

```

i	e	x	e1	proverka
1	7.500000e-01	1.250000e+00	3.750000e-01	2.000000e+00
2	3.750000e-01	1.625000e+00	1.875000e-01	2.000000e+00
3	1.875000e-01	1.812500e+00	9.375000e-02	2.000000e+00
4	9.375000e-02	1.718750e+00	4.687500e-02	2.000000e+00
5	4.687500e-02	1.765625e+00	2.343750e-02	2.000000e+00
6	2.343750e-02	1.789062e+00	1.171875e-02	2.000000e+00
7	1.171875e-02	1.777344e+00	5.859375e-03	2.000000e+00
8	5.859375e-03	1.783203e+00	2.929688e-03	2.000000e+00
9	2.929688e-03	1.780273e+00	1.464844e-03	2.000000e+00
10	1.464844e-03	1.781738e+00	7.324219e-04	2.000000e+00
11	7.324219e-04	1.781006e+00	3.662109e-04	2.000000e+00
12	3.662109e-04	1.781372e+00	1.831055e-04	2.000000e+00
13	1.831055e-04	1.781189e+00	9.155273e-05	2.000000e+00
14	9.155273e-05	1.781281e+00	4.577637e-05	2.000000e+00
15	4.577637e-05	1.781326e+00	2.288818e-05	2.000000e+00
16	2.288818e-05	1.781349e+00	1.144409e-05	2.000000e+00
17	1.144409e-05	1.781338e+00	5.722046e-06	2.000000e+00
18	5.722046e-06	1.781332e+00	2.861023e-06	2.000000e+00
19	2.861023e-06	1.781335e+00	1.430511e-06	2.000000e+00
20	1.430511e-06	1.781336e+00	7.152557e-07	2.000000e+00
21	7.152557e-07	1.781337e+00	3.576279e-07	2.000000e+00
22	3.576279e-07	1.781337e+00	1.788139e-07	2.000000e+00
23	1.788139e-07	1.781337e+00	8.940697e-08	2.000000e+00
24	8.940697e-08	1.781337e+00	4.470348e-08	2.000000e+00
25	4.470348e-08	1.781337e+00	2.235174e-08	2.000000e+00
26	2.235174e-08	1.781337e+00	1.117587e-08	2.000000e+00
27	1.117587e-08	1.781337e+00	5.587935e-09	2.000000e+00
28	5.587935e-09	1.781337e+00	2.793968e-09	2.000000e+00

i—> номер итерации

e—> |(b-a)|

e1—> $= \frac{1}{2}e^{(i)}$

x—> то как мы с каждой итерацией всё ближе и ближе приходим к нашему результату

Фрагмент коду на C для методу хорд:

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

int main()
{
    double c = 0.00000001;
    double a = 0.5;
    double b = 2;
    double y;
    double x;
    double ya = 10*a*a*exp(-a)-3*a;
    double yb = 10*b*b*exp(-b)-3*b;
    double xpooper;
x=a;
int i=1;
do
    {
        xpooper=x;
        x=a-((ya*b-ya*a)/(yb-ya));
        y = 10*x*x*exp(-x)-3*x;

        if ((ya>0 && y>0)|| (ya<0 && y<0))
        {
            a=x;
            ya=y;
        }
        else
        {
            b=x;
            yb=y;
        }
        printf("%d\t%f\n", i++, x);
    }
    while(fabs(xpooper-x)>=c);
    printf("%e\n", x);
return 0;
```

}

Результат отриманий методом хорд:

1	0.540619
2	0.716307
3	1.202680
4	1.646614
5	1.762151
6	1.778875
7	1.781026
8	1.781298
9	1.781332
10	1.781336
11	1.781337
12	1.781337
13	1.781337
1.781337e+00	

Висновок:

В цій лабораторній роботі я отримав розв'язки для рівняння $f(x) = 10x^2e^{-x} - 3x$ методом бісекції та методом хорд. Також цікавим є те що в цьому випадку для отримання результату методом бісекції знадобилося 28 ітерацій, а методом хорд всього 13.