Отчёт по лабораторной работе №1

по дисциплине «Вычислительная математика»

Студент гр. 3530202/90002 Потапова А.М.

Преподаватель Леонтьева Т.В.

2021г

Вариант №13

Постановка задачи:

Для таблично заданной функции f(x)

x	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5
f(x)	0.5440	-0.4121	-0.9894	-0.6570	0.2794	0.9589

построить сплайн-функцию и использовать ее для нахождения корня уравнения
$f(x) = 1.8x^2$ на промежутке [-1, -0.5] методом бисекции.

Ход работы

1. В первую очередь была получена SPLINE-функция;
2. Затем для теста была вызвана функция SEVAL в точках исходной таблицы, результаты которой полностью совпали с табличными.
3. После этого стало возможным реализовать метод бисекции, цель которого — найти нуль функции $f(x) = 1.8x^2 – SEVAL(x)$. Алгоритм данного метода основывается на следствии из теоремы Больцано-Коши, гласящем, что у непрерывной функции, принимающей на концах некоторого отрезка разные знаки, найдется такая точка внутри этого отрезка, что в ней функция будет обращаться в нуль. Разделив отрезок пополам, продолжим работу с той его частью, где функция по-прежнему принимает значения разных знаков. Данную процедуру следует повторять до нахождения точки, являющейся искомым нулем нашей функции.

Текст программы:

```
#include <iostream>
#include "spline.c"
#include "cmathmsg.c"

#define N 6 // количество известных значений полинома
#define EPS 0.00001

double x[N] = { -1.0, -0.9, -0.8, -0.7, -0.6, -0.5 };
double y[N] = { 0.5440, -0.4121, -0.9894, -0.6570, 0.2794, 0.9589 };

double b[N], c[N], d[N];
double xx, S;
int last, flag;

double bisection(double xx)
{
    return 1.8 * xx * xx - seval(N, xx, x, y, b, c, d, &last);
}
```

```
int sign(double xx)
{
    if (bisection(xx) < 0.0)
        return -1;
    else if (bisection(xx) > 0.0)
        return 1;
    else
        return 0;
}

int main()
{
    double x1 = -1;
    double x2 = -0.5;
    double xi;
    double dx;

    spline(N, (int)x1, (int)x2, -0.5440, 0.9589, x, y, b, c, d, &flag);

    if (flag == 0)
    {
        printf("\n      X:    |    SPLINE:  |    Y[X]:\n");
        printf("-----------|-------------|------------\n");


        int i = 5;
        xx = x[i];
        last = 0;

        while (i >= 0)
        {
            S = seval(N, xx, x, y, b, c, d, &last);
            printf("%10.4f |%10.4f   |%10.4f\n", xx, S, y[i]);
            i--;
            xx = x[i];
        }
    }

    xi = (x1 + x2) / 2;
    dx = x2 - x1;

    printf("\n\n     X:     |   Bisection:\n");
    printf("------------|---------------\n");

    while (bisection(xi) > EPS || bisection(xi) * (-1) > EPS)
    {
        dx = dx / 2;
        xi = x1 + dx;
        printf("%10.5f  | %10.5f\n", xi, bisection(xi));

        if (sign(x1) != sign(xi))
            x2 = xi;
        else
            x1 = xi;
    }

    printf("\nBisection method root of f(x) = 1.8 * x^2:");
    printf("%10.5f\n\n", xi);

    return 0;
}
```

Результаты:

```
     X:    |    SPLINE:   |    Y[X]:
-----------|--------------|------------
   -0.5000 |    0.9589    |    0.9589
   -0.6000 |    0.2794    |    0.2794
   -0.7000 |   -0.6570    |   -0.6570
   -0.8000 |   -0.9894    |   -0.9894
   -0.9000 |   -0.4121    |   -0.4121
   -1.0000 |    0.5440    |    0.5440


     X:      |   Bisection:
------------|---------------
  -0.75000  |    1.94843
  -0.62500  |    0.67003
  -0.56250  |   -0.04325
  -0.59375  |    0.29569
  -0.57812  |    0.12011
  -0.57031  |    0.03670
  -0.56641  |   -0.00373
  -0.56836  |    0.01637
  -0.56738  |    0.00629
  -0.56689  |    0.00127
  -0.56665  |   -0.00123
  -0.56677  |    0.00002
  -0.56671  |   -0.00060
  -0.56674  |   -0.00029
  -0.56676  |   -0.00013
  -0.56676  |   -0.00006
  -0.56677  |   -0.00002
  -0.56677  |    0.00000

Bisection method root of f(x) = 1.8 * x^2:  -0.56677
```

Вывод

В результате работы программы мы получили значение $x = -0.56677$, что является вполне подходящим решением уравнения на заданном промежутке. Решение методом биссекции было получено на 18 шаге. В качестве погрешности было взято значение 10^{-5}.