Санкт-Петербургский Политехнический университет Петра Великого
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчет по лабораторной работе №1

Дисциплина: Вычислительная математика

Вариант: 18

Выполнил

студент гр. 3530901/90003 ____________________________ Руднев А.К.

(подпись)

Преподаватель ____________________________ Цыган В.Н.

(подпись)

«___» _______________ 2021 г.

Санкт-Петербург
2021

Входные данные: вариант 18

Таблица 1 – Таблично заданная функция

x	0.0	0.2	0.4	0.7	0.9	1.0
f(x)	1.0000	1.2214	1.4918	2.0138	2.4596	2.7183

Необходимо построить а) сплайн-функцию; б) полином Лагранжа 5-й степени и использовать обе аппроксимирующие функции нахождения корня уравнения $f(x) + x^2 = 2$ на промежутке [0.0, 1.0] методом бисекции. Результаты двух вариантов нахождения корня сравнить для одинаковых задаваемых значений погрешности метода бисекции.

Решение:

1) Создам Spline – проект на языке C++, в котором будет реализовано решение данной задачи.

2) Для нахождения корня уравнения на промежутке [0.0, 1.0] будет использоваться метод бисекции, который предпологает:

2.1 Вычисление функции на концах, а также в середине промежутка.

2.2 Нахождение значения функции в середине промежутка.

2.3 Сравниваем знаки на концах.

2.4 A заменяем на C, если знак A = знаку C, иначе B меняем C.

2.5 Если значение в точке C меньше, чем предполагаемая погрешность, то заканчиваем вычисления.

2.6 Если значение в точке C больше, чем погрешность, то повторяем алгоритм.

3) Метод бисекции будет находить корень, используя две аппроксимирующие функции:

3.1 Полином Лагранжа. Значения каждой новой точки функции будет высчитываться с построением нового полинома Лагранжа 5 степени. Для этого в проект был дабавлен соответствующий метод lagrange и модернизирована функции bisection.

3.2 Аппроксимирующая функция Spline. В начале создается Spline-функция, и при дальнейшем значение функции в точках будет получено с помощью функции seval.

Листинг:

```cpp
#ifndef SPLINE_FUNCTION_H
#define SPLINE_FUNCTION_H


#include <iomanip>

class Function {

    // Переменные для построения функции Spline
public:
    double bS[7], cS[7], dS[7];
    //Таблично заданная функция
    double
            y[7] = {0, 1.0000, 1.2214, 1.4918, 2.0138, 2.4596, 2.7183};
    //Таблично заданная функция
    double x[7] = {0, 0.0, 0.2, 0.4, 0.7, 0.9, 1.0};
    int n = 6; //степень полинома

    //Сплайн
    //Построение функции, используя Spline
    Function() {
        spline(n, x, y, bS, cS, dS);
    };
    //Стандартная подпрограмма Spline
    void spline(int n,double * x,double * y,double * b,double * c,double * d) {...}
    //Стандартная подпрограмма Seval
    static double seval(int n, double u, double * x, double * y, double * b, const double * c, double * d) {...}
    //Стандартная подпрограмма Seval
    static double seval(int n, double u, double * x, double * y, double * b, const double * c, double * d) {...}
    //Получение значения функции при помощи Seval
    double getYSpline(double u) {
        return seval(n, u, x, y, bS, cS, dS) + u * u - 2;
    }
    //Метод бисекции для сплайн
    static double bisectionSpline(Function function, double a, double d, double e) {
        double aX, bX, cX, // [A, B] - интервал, C - середнина
        aF, bF, cF; // Значение spline-функции в этих точках

        aX = 0.0; //Начало промежутка
        bX = 1.0; //Конец промежутка
        cX = 0.0; //Середина промежутка

        aF = Function().getYSpline(aX); //Значение spline-функции в точке a
        bF = Function().getYSpline(bX); //Значение spline-функции в точке b
        int i = 0;
        while (i < 300000) {
            i++;

            if (aF * bF > 0) {
            cX = (aX + bX) / 2; //Середина промежутка
            cF = Function().getYSpline(cX);
            //Присвоение точки середины одному из концов промежутка (середина становится концом)
            if (aF * cF < 0) {
                bF = cF;
                bX = cX;
            } else {
                aF = cF;
                aX = cX;
            }
            //Завершение работы, если данные удовлетворяют погрешности
            if (std::abs( _Xx: aX - bX) < e) {
                break;
            } else {

            }
            //Вывод подробных результатов в терминал
            printf( _Format: "Bisection Spline, interval [%lf, %lf]:\n", aX, bX);
            std::cout << "F(a)=" << std::setprecision(10) << aF << std::endl;
            std::cout << "F(b)=" << std::setprecision(10) << bF << std::endl;
            std::cout << "F(c)=" << std::setprecision(10) << cF << std::endl;
            printf( _Format: "\n");

        }
        // Подробные результаты решения

        printf( _Format: "Need %i iterations with precision %lf\n", i, e);
```

```
        printf( _Format: "Value of bisection %lf\n\n\n", cX);

        return cX;
    }

    //Лагранж
    //Стандартная подпрограмма Lagrange
    double lagrange(double xc, double *x,double *y) {...}
    //Получение значения в точке
    double getYL(double u) {
        return lagrange(u, x, y) + u * u - 2;
    }
    //Метод бисекции для лагранж
    double bisectionLagrange(Function function, double a, double d, double e) {
        printf( _Format: "\n\n\n");
        double aX, bX, cX, // [A, B] - интервал, C - середнина
        aF, bF, cF; // Значение spline-функции в этих точках

        aX = 0.0; //Начало промежутка
        bX = 1.0; //Конец промежутка
        cX = 0.0; //Середина промежутка

        aF = Function().getYL(aX); //Значение spline-функции в точке a
        bF = Function().getYL(bX); //Значение spline-функции в точке b
        int i = 0;
        while (i < 300000) {
            i++;
            cX = (aX + bX) / 2; //Середина промежутка
            cF = Function().getYL(cX);
            //Присвоение точки середины одному из концов промежутка (середина становится концом)
            if (aF * cF < 0) {
                bF = cF;
                bX = cX;
            } else {
                aF = cF;
                aX = cX;
            }
            //Завершение работы, если данные удовлетворяют погрешности
            if (std::abs( _Xx: aX - bX) < e) {
                break;
            } else {

            }
            //Вывод подробных результатов в терминал
            printf( _Format: "Bisection Lagrange, interval [%lf, %lf]:\n", aX, bX);
            std::cout << "F(a)=" << std::setprecision(10) << aF << std::endl;
            std::cout << "F(b)=" << std::setprecision(10) << bF << std::endl;
            std::cout << "F(c)=" << std::setprecision(10) << cF << std::endl;
            printf( _Format: "\n");

        }
        // Подробные результаты решения
        printf( _Format: "Need %i iterations with precision %lf\n", i, e);
        printf( _Format: "Value of bisection %lf\n\n\n", cX);
        return cX;
    }
#include ...


int main() {
    Function spline = Function(); //Построение сплайн-функции

    double
    a = 0.0, //Начало промежутка
    b = 1.0, //Конец промежутка
    e = 0.1; //Погрешность
    double bis = Function().bisectionSpline(spline, a, b, e); //Результат метода бисекции
    double bisl = Function().bisectionLagrange(spline, a, b, e); //Результат метода бисекции
    //Выведение результатов в консоль
    std::cout << "\nResult bisection = " << std::setprecision(15) << bis << std::endl;
    std::cout << "\nResult bisection = " << std::setprecision(15) << bisl << std::endl;
    return 0;
```

Результаты вычислений:

```
Bisection Spline, interval [0.500000, 1.000000]:
F(a)=-0.1013643779
F(b)=1.7183
F(c)=-0.1013643779

Bisection Spline, interval [0.500000, 0.750000]:
F(a)=-0.1013643779
F(b)=0.6805417194
F(c)=0.6805417194

Bisection Spline, interval [0.500000, 0.625000]:
F(a)=-0.1013643779
F(b)=0.2582974576
F(c)=0.2582974576

Need 4 iterations with precision 0.100000
Value of bisection 0.562500
```

Рис.1

```
Bisection Lagrange, interval [0.500000, 1.000000]:
F(a)=-0.3879582389
F(b)=-1.436213152
F(c)=-0.3879582389

Bisection Lagrange, interval [0.500000, 0.750000]:
F(a)=-0.3879582389
F(b)=0.9239063275
F(c)=0.9239063275

Bisection Lagrange, interval [0.625000, 0.750000]:
F(a)=-0.02579267333
F(b)=0.9239063275
F(c)=-0.02579267333

Need 4 iterations with precision 0.100000
Value of bisection 0.687500
```

Рис. 2

```
F(a)=-0.1013643779
F(b)=1.7183
F(c)=-0.1013643779

Bisection Spline, interval [0.500000, 0.750000]:
F(a)=-0.1013643779
F(b)=0.6805417194
F(c)=0.6805417194

Bisection Spline, interval [0.500000, 0.625000]:
F(a)=-0.1013643779
F(b)=0.2582974576
F(c)=0.2582974576

Bisection Spline, interval [0.500000, 0.562500]:
F(a)=-0.1013643779
F(b)=0.07102232387
F(c)=0.07102232387

Bisection Spline, interval [0.531250, 0.562500]:
F(a)=-0.01698238056
F(b)=0.07102232387
F(c)=-0.01698238056

Bisection Spline, interval [0.531250, 0.546875]:
F(a)=-0.01698238056
F(b)=0.02656092072
F(c)=0.02656092072

Need 7 iterations with precision 0.010000
Value of bisection 0.539063
```

Рис. 3

```
Bisection Lagrange, interval [0.500000, 1.000000]:
F(a)=-0.3879582389
F(b)=-1.436213152
F(c)=-0.3879582389

Bisection Lagrange, interval [0.500000, 0.750000]:
F(a)=-0.3879582389
F(b)=0.9239063275
F(c)=0.9239063275

Bisection Lagrange, interval [0.625000, 0.750000]:
F(a)=-0.02579267333
F(b)=0.9239063275
F(c)=-0.02579267333

Bisection Lagrange, interval [0.625000, 0.687500]:
F(a)=-0.02579267333
F(b)=0.4030640476
F(c)=0.4030640476

Bisection Lagrange, interval [0.625000, 0.656250]:
F(a)=-0.02579267333
F(b)=0.1705204816
F(c)=0.1705204816

Bisection Lagrange, interval [0.625000, 0.640625]:
F(a)=-0.02579267333
F(b)=0.06732180482
F(c)=0.06732180482

Need 7 iterations with precision 0.010000
Value of bisection 0.632813
```

Рис. 4

На рисунках 1-4 представлены результаты работы программы для двух аппроксимирующих функций, и соответственно для погрешности 0.1 и 0.01.

Сравнение результатов:

Сравню ответы, которые были получены методом бисекции, используя различные аппроксимирующие функции. Количество итераций метода бисекции занесу в таблицу 1. Значение корня уравнения в таблицу 2.

Таблица 1 – Количество итераций

Погрешность	Количество итераций
0.1	4
0.01	7
0.001	10
0.0001	14
0.00001	17
0.000001	20

Таблица 2 – Значение корня

Погрешность	Spline	Lagrange
	Ответ	Ответ
0.1	0.5625	0.5625
0.01	0.5390625	0.5390625
0.001	0.5380859375	0.5380859375
0.0001	0.53741455078125	0.53729248046875
0.00001	0.537376403808594	0.537269592285156
0.000001	0.537383079528809	0.537266731262207

Из полученных результатов можно сделать вывод, что методы интерполяции Spline и Лагранжа дают почти идентичные результаты ха исключением результатов с слишком маленькой погрещностью. Различия наблюдаются, начиная с погрешности = 0.0001. В методе бисекции была задана необходимая погрешность и исходя из этой погрешности вычислялся корень уравнения, удовлетворяющий ей. Графики приведены на рисунке 5 и 6.

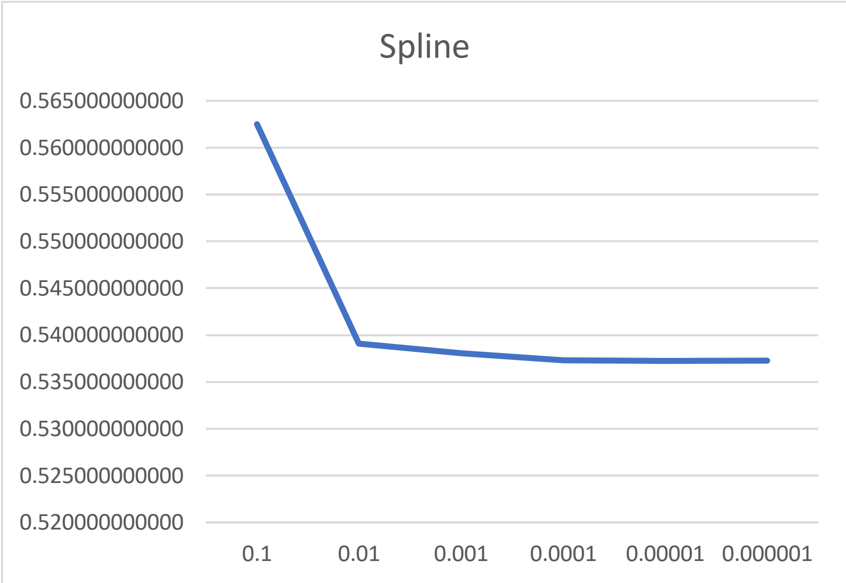

Рисунок 5 – Зависимость значения корня от погрешности

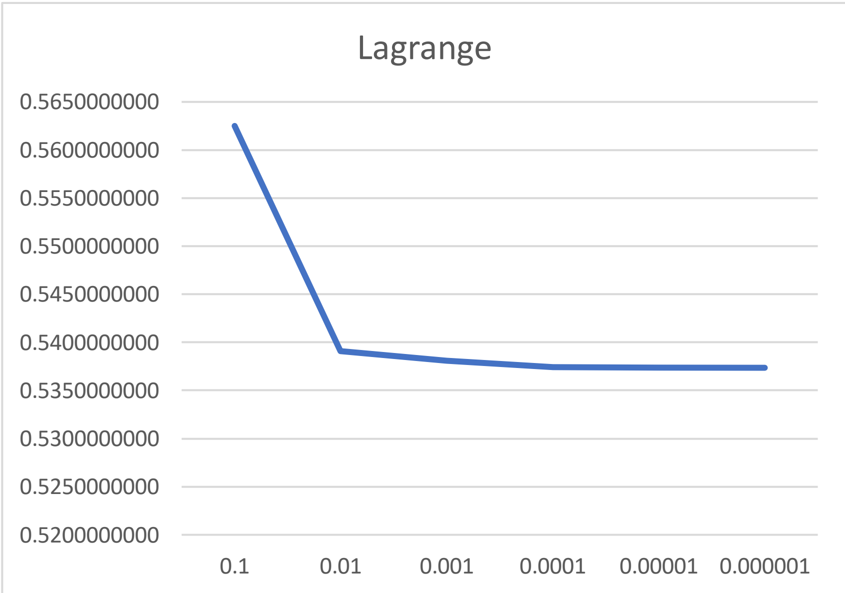

Рисунок 6 – Зависимость значения корня от погрешности

На рисунках 7 и 8 приведены полиномы, которые были построены с помощью Spline и Lagrange.

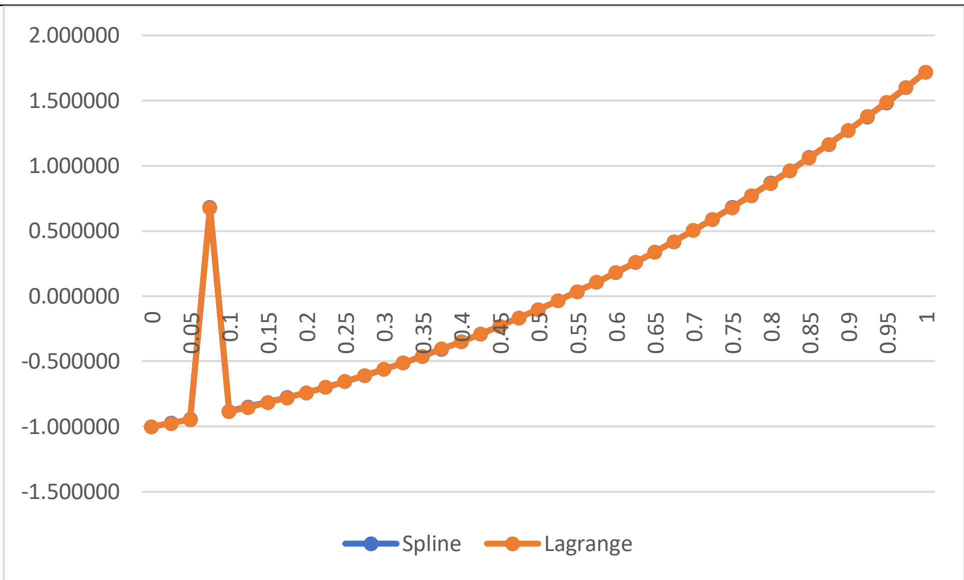

Рисунок 7 – Аппроксимация с помощью Spline и Lagrange

Как видно из рисунка 7, Spline и Lagrange аппроксимируют функцию почти идентично на промежутка [0; 1.0] с шагом 0.025, за исключением некоторых моментов (видно на рисунке небольшие выделения синего цвета)

Вывод: в ходе выполнения работы было проведено ознакомление с работой подпрограмм Spline, Seval, методом аппроксимации, используя полином Лагранжа, а также была самостоятельно написан метод бисекции, и вычислены корни для уравнения $f(x) + x^2 = 2$. Исходя из полученных данных было определено, что корень для аппроксимирующей функции spline почти совпадал с корнем аппроксимирующей функции Лагранжа. Было определено необходимое количество итераций метода бисекции для удовлетворения погрешности искомого значения: чем меньше погрешность – тем больше итераций метода необходимо.