Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3
Задание №2 по курсу «Численные методы»
Вариант №12

Студент:	С. М. Власова
Преподаватель:	И. Э. Иванов
Группа:	М8О-306Б
Дата:	
Оценка:	
Подпись:	

Москва, 2020

Задание №3.2

Построить кубический сплайн для функции, заданной в узлах интерполяции, предполагая, что сплайн имеет нулевую кривизну при $x = x_0$ и $x = x_4$. Вычислить значение функции в точке $x = X^{\star}$.

Вариант: 12

$X^{\star} = 0.8$.

i	0	1	2	3	4
x_i	0.0	0.5	1.0	1.5	2.0
f_i	0.0	0.97943	1.8415	2.4975	2.9093

1 Описание метода решения

Использование одной интерполяционной формулы на большом числе узлов нецелесообразно. Интерполяционный многочлен может проявить свои колебательные свойства, его значения между узлами могут сильно отличаться от значений интерполируемой функции. Одна из возможностей преодоления этого недостатка заключается в применении **сплайн-интерполяции.** Суть сплайн-интерполяции заключается в определении интерполирующей функции по формулам одного типа для различных не пересекающихся промежутков и в стыковке значений функции и её производных на их границах.

Наиболее широко применяемым является случай, когда между любыми двумя точками разбиения исходного отрезка строится многочлен $n-$й степени:

$$S(x) = \sum_{k=0}^{n} a_{ik} \cdot x^k, \quad x_{i-1} \le x \le x_i, \quad i = 1, \dots, n,$$

который в узлах интерполяции принимает значения аппрокимируемой функции и непрерывен вместе со своими $(n-1)$ производными. Такой кусочно-непрерывный интерполяционный многочлен называется сплайном. Его коэффициенты находятся из условий равенства в узлах сетки значений сплайна и приближаемой функции, а также равенства $n-1$ производных соответствующих многочленов. На практике наиболее часто используется интерполяционный многочлен третьей степени, который удобно представить как

$$S(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad x_{i-1} \le x \le x_i, \quad i = 1, \dots, n.$$

Для построения кубического сплайна необходимо построить n многочленов третьей степени, т.е. определить $4n$ неизвестных a_i, b_i, c_i, d_i. Эти коэффициенты ищутся из условий в узлах сетки.

$$\begin{aligned}
&S(x_{i-1}) = a_i = a_{i-1} + b_{i-1}(x_{i-1} - x_{i-2}) + c_{i-1}(x_{i-1} - x_{i-2})^2 + d_{i-1}(x_{i-1} - x_{i-2})^3 = f_{i-1}\\
&S'(x_{i-1}) = b_i = b_{i-1} + 2 \cdot c_{i-1}(x_{i-1} - x_{i-2}) + 3 \cdot d_{i-1}(x_{i-1} - x_{i-2})^2,\\
&S''(x_{i-1}) = 2 \cdot c_i = 2 \cdot c_{i-1} + 6 \cdot d_{i-1}(x_{i-1} - x_{i-2}), \qquad i = 2, \dots, n\\
&S(x_0) = a_1 = f_0\\
&S''(x_0) = c_1 = 0\\
&S(x_n) = a_n + b_n(x_n - x_{n-1}) + c_n(x_n - x_{n-1})^2 + d_n(x_n - x_{n-1})^3 = f_n\\
&S''(x_n) = c_n + 3 \cdot d_n(x_n - x_{n-1}) = 0
\end{aligned}$$

Предполагается, что сплайны имеют нулевую кривизну на концах отрезка. В общем случае могут быть использованы и другие условия.

Если ввести обозначение $h_i = x_i - x_{i-1}$ и исключить из системы a_i, b_i, c_i, d_i, то можно получить систему из $n-1$ линейных алгебраических уравнений относительно

$c_i, \ \ i = 2, \ldots, n$ с трёхдиагональной матрицей:

$$2(h_1 + h_2)c_2 + h_2c_3 = 3[(f_2 - f_1)/h_2 - (f_1 - f_0)/h_1]$$
$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_ic_{i+1} = 3[(f_i - f_{i-1})/h_i - (f_{i-1} - f_{i-2})/h_{i-1}], \ \ i = 3, \ldots, n-1$$
$$h_{n-1}c_{n-1} + 2(h_{n-1} + h_n)c_n = 3[(f_n - f_{n-1})/h_n - (f_{n-1} - f_{n-2})/h_{n-1}].$$

Остальные коэффициенты сплайнов могут быть восстановлены по формулам:

$$a_i = f_{i-1}, \ \ i = 1, \ldots, n; \ \ b_i = (f_i - f_{i-1})/h_i - \frac{1}{3}h_i(c_{i+1} + 2c_i), \ \ d_i = \frac{c_{i+1} - c_i}{3h_i}, \ \ i = 1, \ldots, n-1$$

$$c_1 = 0, \ \ b_n = (f_n - f_{n-1})/h_n - \tfrac{2}{3}h_nc_n, \ \ d_n = -\tfrac{c_n}{3h_n}$$

2 Моя задача

Построить кубический сплайн для функции, заданной в узлах интерполяции, предполагая, что сплайн имеет нулевую кривизну при $x = x_0$ и $x = x_4$.

i	0	1	2	3	4
x_i	0.0	0.5	1.0	1.5	2.0
f_i	0.0	0.97943	1.8415	2.4975	2.9093

Вычислить значение функции в точке $x = X^{\star} = 0.8$.

Решение:
Введем обозначение $h_i = x_i - x_{i-1}$. Тогда $h_1 = h_2 = h_3 = h_4 = 0.5$.
Найдем коэффициенты сплайнов, решив СЛАУ относительно $c_i, \quad i = 2, \ldots, n:$

$$\begin{cases} 2(0.5+0.5)c_2 + 0.5c_3 = 3[\frac{1.8415-0.97943}{0.5} - \frac{0.97943-0}{0.5}] \\ 0.5c_2 + 2(0.5+0.5)c_3 + 0.5c_4 = 3[\frac{2.4975-1.8415}{0.5} - \frac{1.8415-0.97943}{0.5}] \\ 0.5c_3 + 2(0.5+0.5)c_4 = 3[\frac{2.9093-2.4975}{0.5} - \frac{2.4975-1.8415}{0.5}] \end{cases}$$

$$\begin{cases} 2c_2 + 0.5c_3 = -0.70416 \\ 0.5c_2 + 2c_3 + 0.5c_4 = -1.23642 \\ 0.5c_3 + 2c_4 = -1.4652 \end{cases}$$

Эта СЛАУ соответствует трёхдиагональной матрице и решается методом прогонки.
Решение СЛАУ методом прогонки:

$$c_1 = 0.0, \quad c_2 = -0.252926, \quad c_3 = -0.396617, \quad c_4 = -0.633446.$$

Тогда могут быть найдены остальные коэффициенты согласно формулам.

i	$[x_{i-1}, x_i]$	a_i	b_i	c_i	d_i
1	$[0.0, 0.5]$	0.0	2.00101	0.0	-0.168617
2	$[0.5, 1.0]$	0.97943	1.87455	-0.252926	-0.0957943
3	$[1.0, 1.5]$	1.8415	1.54978	-0.396617	-0.157886
4	$[1.5, 2.0]$	2.4975	1.03475	-0.633446	0.422297

3 Протокол

Входные данные я храню в файле *test3.2.t*: первая строка — число интерполяционных узлов, вторая строка — их значения, третья строка — значения функции в этих узлах, четвертая строка — точка, в которой необходимо вычислить значение функции.
Выходные данные я вывожу на экран.
Скриншот консоли:

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab3$ g++ 3.2.cpp -o 3.2
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab3$ cat test3.2.t
5
0.0 0.5 1.0 1.5 2.0
0.0 0.97943 1.8415 2.4975 2.9093
0.8
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab3$ ./3.2 < test3.2.t
Кубический сплайн на интервале [0, 0.5] : f(x) = 0 + 2.00101x^1 - 0.168617x^3
Кубический сплайн на интервале [0.5, 1] : f(x) = 0.97943 + 1.87455(x - 0.5)^1 - 0.252926(x - 0.5)^2 - 0.0957943(x - 0.5)^3
Кубический сплайн на интервале [1, 1.5] : f(x) = 1.8415 + 1.54978(x - 1)^1 - 0.396617(x - 1)^2 - 0.157886(x - 1)^3
Кубический сплайн на интервале [1.5, 2] : f(x) = 2.4975 + 1.03475(x - 1.5)^1 - 0.633446(x - 1.5)^2 + 0.422297(x - 1.5)^3
Точка 0.8 принадледит отрезку [0.5, 1], на этом отрезке функция представляется сплайном:
f(x) = 0.97943 + 1.87455(x - 0.5)^1 - 0.252926(x - 0.5)^2 - 0.0957943(x - 0.5)^3
f(0.8) = 1.51645
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab3$
```

Кубический сплайн на интервале $[0, 0.5]$:

$$f(x) = 2.00101x - 0.168617x^3$$

Кубический сплайн на интервале $[0.5, 1]$:

$$f(x) = 0.97943 + 1.87455(x - 0.5) - 0.252926(x - 0.5)^2 - 0.0957943(x - 0.5)^3$$

Кубический сплайн на интервале $[1, 1.5]$:

$$f(x) = 1.8415 + 1.54978(x - 1) - 0.396617(x - 1)^2 - 0.157886(x - 1)^3$$

Кубический сплайн на интервале $[1.5, 2]$:

$$f(x) = 2.4975 + 1.03475(x - 1.5) - 0.633446(x - 1.5)^2 + 0.422297(x - 1.5)^3$$

Точка $X^\star = 0.8$ принадлежит отрезку $[0.5, 1]$, на этом отрезке функция представляется сплайном:

$$f(x) = 0.97943 + 1.87455(x - 0.5) - 0.252926(x - 0.5)^2 - 0.0957943(x - 0.5)^3$$

$$f(X^\star) = f(0.8) = 1.51645$$

4 Исходный код

Листинг 1: Сплайн-интерполяция

```cpp
#include <iostream>
#include <math.h>
#include <vector>

void FindC(std::vector<double>& c, std::vector<double>& f, std::vector<double>& h, int n)
{
    std::vector<double> P(n);
    std::vector<double> Q(n);

    for( int i = 1; i < n; i++) //straight way
    {
        if( i == 1)
        {
            P[i] = -h[i]/(2*(h[i-1]+h[i])); //-c/b;
            Q[i] = 3*((f[i+1]-f[i])/h[i] - (f[i]-f[i-1])/h[i-1])/(2*(h[i-1]+h[i])); //d/b;
        }
        else if( i != n-1)
        {
            P[i] = -h[i]/( 2*(h[i-1]+h[i]) + h[i-1]*P[i-1]); //-c/(b + a * P[i-1]);
            Q[i] = ( 3*((f[i+1]-f[i])/h[i] - (f[i]-f[i-1])/h[i-1]) - h[i-1]*Q[i-1] )/( 2*(h[i-1]+h[i]) + h[i-1]*P[i-1]); //(d - a*Q[i-1])/(b + a*P[i-1]);
        }
        else
        {
            P[i] = 0;
            Q[i] = ( 3*((f[i+1]-f[i])/h[i] - (f[i]-f[i-1])/h[i-1]) - h[i-1]*Q[i-1] )/( 2*(h[i-1]+h[i]) + h[i-1]*P[i-1]); //(d - a*Q[i-1])/(b + a*P[i-1]);
        }
    }
    for( int i = n-1; i >= 1; i--) //backway
    {
        if( i == n-1)
            c[i] = Q[i];
        else
            c[i] = P[i]*c[i + 1] + Q[i];
    }

}

```

```
double SplainValue(double x, double a, double b, double c, double d, double x_star)
{
    double value = a + b*(x_star - x) + c*pow(x_star-x, 2) + d*pow(x_star-x, 3);
    return value;
}

void PrintSplain(double x, double a, double b, double c, double d, int n)
{
    double kaf;
    std::cout << "f(x) = " << a;
    for(int j = 0; j < 3; j++)
    {
        if(j == 0)
            kaf = b;
        else if(j == 1)
            kaf = c;
        else kaf = d;

        if(kaf < 0)
        {
            std::cout << " - " << -kaf;
            if(x < 0)
                std::cout << "(x - " << -x << ")^" << j + 1;
            else if(x > 0)
                std::cout << "(x - " << x << ")^" << j + 1;
            else
                std::cout << "x^" << j + 1;
        }
        else if(kaf > 0)
        {
            std::cout << " + " << kaf;
            if(x < 0)
                std::cout << "(x - " << -x << ")^" << j + 1;
            else if(x > 0)
                std::cout << "(x - " << x << ")^" << j + 1;
            else
                std::cout << "x^" << j + 1;
        }
    }
    std::cout << std::endl;
}

void PrintSplains(std::vector<double>& x, std::vector<double>& a, std::vector<double>& b,
    std::vector<double>& c, std::vector<double>& d, int n)
```

```
{
    double kaf;
    for(int i = 0; i < n-1; i++)
    {
        std::cout << "ubic spline on the interval [" << x[i] << ", " << x[i+1] << "] : " << "f(
            x) = " << a[i];
        for(int j = 0; j < 3; j++)
        {
            if(j == 0)
                kaf = b[i];
            else if(j == 1)
                kaf = c[i];
            else kaf = d[i];

            if(kaf < 0)
            {
                std::cout << " - " << -kaf;
                if(x[i] < 0)
                    std::cout << "(x - " << -x[i] << ")^" << j + 1;
                else if(x[i] > 0)
                    std::cout << "(x - " << x[i] << ")^" << j + 1;
                else
                    std::cout << "x^" << j + 1;
            }
            else if(kaf > 0)
            {
                std::cout << " + " << kaf;
                if(x[i] < 0)
                    std::cout << "(x - " << -x[i] << ")^" << j + 1;
                else if(x[i] > 0)
                    std::cout << "(x - " << x[i] << ")^" << j + 1;
                else
                    std::cout << "x^" << j + 1;
            }
        }
        std::cout << std::endl;
    }
}

int main()
{
    double x_star;
    int n;
    std::cin >> n;
```

```

    std::vector<double> x(n);
    std::vector<double> f(n);
    std::vector<double> h(n-1, 0);
    std::vector<double> a(n-1, 0);
    std::vector<double> b(n-1, 0);
    std::vector<double> c(n-1, 0);
    std::vector<double> d(n-1, 0);

    for(int i = 0; i < n-1; i++)
    {
        if(i == 0)
            std::cin >> x[i];
        std::cin >> x[i+1];
        h[i] = x[i+1] - x[i];
    }
    for(int i = 0; i < n; i++)
        std::cin >> f[i];
    std::cin >> x_star;
    FindC(c, f, h, n-1);

    for(int i = 0; i < n-1; i++)
    {
        a[i] = f[i];

        if(i != n-2)
        {
            b[i] = (f[i+1]-f[i])/h[i] - h[i]*(c[i+1]+2*c[i])/3;
            d[i] = (c[i+1]-c[i])/(3*h[i]);
        }
        else
        {
            b[i] = (f[i+1]-f[i])/h[i] -h[i]*c[i]*2/3;
            d[i] = -c[i]/(3*h[i]);
        }
    }
    PrintSplains(x, a, b, c, d, n);

    for(int i = 0; i < n; i++)
    {
        if( x[i] <= x_star && x_star <= x[i+1])
        {
            std::cout << "The point " << x_star << " belongs to the interval [" << x[i] << ",
                " << x[i+1] << "], where the function is represented be the splain: " << std::
```

```
            endl;
        PrintSplain(x[i], a[i], b[i], c[i], d[i], n);
        std::cout << "f(" << x_star << ") = " << SplainValue(x[i], a[i], b[i], c[i], d[i],
            x_star) << std::endl;
        break;
      }
    }
    return 0;
}
```

5 Выводы

Выполнив второе задание третье лабораторной работы, я изучила еще один метод интерполяции функции — сплайн-интерполяцию. Этот подход позволяет преодолеть недостаток неопределенности на интервалах между интерполяционными узлами. Так как многочлены Лагранжа и Ньютона, определяются одинаковым образом на всей области точек, они могут проявить свои колебательные свойства —их значения между узлами могут сильно отличаться от значений интерполируемой функции. Сплайн-интерполяция заключается в построении интерполирующей функции по формулам одного типа для различных не пересекающихся интервалов и в стыковке значений функции, ее производных на их границах. Строится кусочно-непрерывный интерполяционный многочлен, называемый сплайном. Сплайн учитывает характер интерполируемой функции на каждом интервале, таким образом преодолевая неопределенность в точках, находящихся между интерполяцонными узлами.