

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Компьютерная графика»
Тема: Кубические сплайны

Студент гр. 0304

Максименко Е.М.

Преподаватель

Герасимова Т.В.

Санкт-Петербург

2023

Цель работы.

- изучение сплайнов и области их применения
- изучение способов построения сплайнов

Задание.

Реализовать интерактивное приложение, отображающее заданные полиномиальные кривые.

При этом для кривых, состоящих из нескольких сегментов, должно быть обеспечено свойство непрерывной кривизны. Программа должна позволять пользователю: интерактивно менять положение контрольных точек, касательных, натяжений.

Вариант 12.

Кубический сплайн, состоящий из 2-х сегментов.

Выполнение работы.

Работа была выполнена с использованием языка программирования C++ и фреймворка Qt 6. Каркасом программы послужила программа из работы 1.

В работе необходимо построить кубический сплайн, состоящий из нескольких сегментов. Для вычисления функции сплайна был написан класс *Spline* (см. рис. 1).

```

1  #ifndef SPLINE_H
2  #define SPLINE_H
3
4  #include <QVector>
5  #include <QPointF>
6
7  class Spline
8  {
9  public:
10     Spline(const QVector<QPointF>& points);
11     void build();
12     void flush();
13     double getValue(double x) const;
14 private:
15     double _h(int i) const;
16     double _s(int i) const;
17     double _r(int i) const;
18
19     void _forward_traverse();
20     void _backward_traverse();
21     void _calculate_abd_coefficients();
22
23     struct Coefficients
24     {
25         double a;
26         double b;
27         double c;
28         double d;
29     };
30
31     const QVector<QPointF>& _points;
32     QVector<Spline::Coefficients> _coefficients;
33     QVector<QPointF> _forward_traverse_coefficients;
34 };
35
36 #endif // SPLINE_H
37

```

Рисунок 1. Класс *Spline*

Класс *Spline* принимает набор точек, по которым он будет строиться и сохраняет их в поле *_points*. Класс имеет 3 публичных метода:

- *void build()* — метод вычисления коэффициентов кубического полинома для каждого сегмента сплайна по заданным точкам.
- *void flush()* — метод очистки вектора коэффициентов полиномов и вектора для промежуточных вычислений.
- *double getValue(double x)* — метод получения значения функции сплайна в точке *x*.

Для вычисления функции сплайна используется метод *void build()*, который последовательно вызывает методы *void _forward_traverse()*, *void _backward_traverse()*, *void _calculate_abd_coefficients()*. При расчете коэффициентов использовались выкладки с рис. 2-3.

$$\begin{aligned}
 \varphi_i(x) &= a_i + b_i(x-x_{i-1}) + c_i(x-x_{i-1})^2 + d_i(x-x_{i-1})^3 \quad (2) \\
 \varphi_i(x_{i-1}) &= y_{i-1}, \quad \varphi_i(x_i) = y_i \quad (3) \\
 \varphi'_i(x_i) &= \varphi'_{i+1}(x_i) \quad (4) \\
 \varphi''_i(x_i) &= \varphi''_{i+1}(x_i) \quad (5) \\
 \varphi''_1(x_0) &= 0 \quad (6) \\
 \varphi''_n(x_n) &= 0 \quad (7) \\
 \text{из (3)-(7)} &\Rightarrow \\
 a_i &= y_{i-1} - \varphi(x_{i-1}) \quad (8) \\
 a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 &= y_i \quad (9) \quad \text{где } h_i = x_i - x_{i-1}, i \in [2; n] \\
 \varphi'_i(x) &= b_i + 2c_i(x-x_{i-1}) + 3d_i(x-x_{i-1})^2 \quad (10) \\
 \varphi''_i(x) &= 2c_i + 6d_i(x-x_{i-1}) \quad (11) \\
 \text{из (4), (5) и (10), (11)} &\Rightarrow \\
 b_i + 2c_i h_i + 3d_i h_i^2 &= b_{i+1} \quad (12) \\
 c_i + 3d_i h_i &= c_{i+1} \quad (13) \\
 \text{из (6), (7) и (13)} &\Rightarrow \\
 c_1 &= 0 \quad (14) \\
 c_n + 3d_n h_n &= 0 \quad (15) \\
 \text{из (13)} &\Rightarrow \\
 d_i &= \frac{c_{i+1} - c_i}{3h_i} \quad (16) \\
 \text{из (8), (9) и (16)} &\Rightarrow \\
 b_i &= \frac{y_i - y_{i-1}}{h_i} - \frac{(c_i h_i + \frac{c_{i+1} - c_i}{3h_i} h_i^2)}{1} = \frac{y_i - y_{i-1}}{h_i} - \frac{(c_i + \frac{2c_i}{3}) h_i}{3} \quad (17) \\
 \text{из (16), (17) и (12)} &\Rightarrow \\
 \frac{y_i - y_{i-1}}{h_i} - \frac{(c_i + \frac{2c_i}{3}) h_i}{3} + h_i(c_i + c_{i+1}) &= \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{(c_{i+1} + \frac{2c_{i+1}}{3}) h_{i+1}}{3} \\
 3h_i(c_i + c_{i+1}) - h_i(c_i + \frac{2c_i}{3}) + h_{i+1}(c_{i+1} + \frac{2c_{i+1}}{3}) &= 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right) \\
 \left[3h_i c_i + 3h_i c_{i+1} - \frac{2}{3} h_i c_i + h_{i+1} c_{i+1} + \frac{2}{3} h_{i+1} c_{i+1} \right] &= \\
 = h_i c_i + 2h_i c_{i+1} + 2h_{i+1} c_{i+1} + h_{i+1} c_{i+2} &= h_i c_i + 2c_{i+1}(h_i + h_{i+1}) + h_{i+1} c_{i+2} \\
 h_i c_i + 2c_{i+1}(h_i + h_{i+1}) + h_{i+1} c_{i+2} &= 3\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right) \quad i \in [2; n-1] \\
 h_{i-1} c_{i-1} + 2c_i(h_{i-1} + h_i) + h_i c_{i+1} &= 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right) \quad i \in [2; n] \quad (18) \\
 \text{При } i=n &\Rightarrow \\
 c_{n+1} &= 0 \quad (19)
 \end{aligned}$$

Рисунок 2. Вывод формул для коэффициентов сплайна

Из $n-1$ уравнений (18) и уравнений (14) и (19) \Rightarrow
 СЛАУ для определения коэффициентов c_i

Метод прогонки:

$$h_{i-1}c_{i-1} + s_i c_i + h_i c_{i+1} = r_i \quad (20)$$

$$\begin{cases} s_i = 2(h_{i-1} + h_i) \\ r_i = 3 \left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right) \end{cases} \quad (21)$$

При $i=2$ с учетом (14)

$$c_2 = k_2 - l_2 c_3 \quad (22), \quad k_2, l_2 - \text{прогонные коэффициенты}$$

$$(h_2 - l_2 c_3) \cdot \frac{1}{h_2} + h_2 c_3 = r_2,$$

где $k_1 = \frac{r_2}{s_2}, l_2 = \frac{h_2}{s_2}$

Используем $c_{i-1} = k_{i-1} - l_{i-1} c_i$ в (20)

$$h_{i-1}(k_{i-1} - l_{i-1} c_i) + s_i c_i + h_i c_{i+1} = r_i$$

$$c_i(s_i - h_{i-1}l_{i-1}) = r_i - h_{i-1}k_{i-1} - h_i c_{i+1}$$

$$c_i = \frac{r_i - h_{i-1}k_{i-1}}{s_i - h_{i-1}l_{i-1}} - \frac{h_i}{s_i - h_{i-1}l_{i-1}} c_{i+1}$$

$$\Rightarrow k_i = \frac{r_i - h_{i-1}k_{i-1}}{s_i - h_{i-1}l_{i-1}}, \quad l_i = \frac{h_i}{s_i - h_{i-1}l_{i-1}} \quad (24)$$

$$k_1 = 0, l_1 = 0$$

$$c_n = k_n \quad (25)$$

Последовательно применяется (24) для $i=2, \dots, n$ — прямой ход прогонки
 Последовательно применяется (23) для $i=n-1, n-2, \dots, 2$ — обратный ход прогонки

Рисунок 3. Вывод формул для коэффициентов сплайна

Согласно данным выкладкам, набор коэффициентов $\{c_i\}$ вычисляется методом прогонки, состоящим из прямого прохода (метод `void _forward_traverse()`) и обратного прохода (метод `void _backward_traverse()`). После вычисления коэффициентов $\{c_i\}$ можно вычислить оставшиеся коэффициенты полиномов (метод `void _calculate_abd_coefficients()`). Код данных методов приведен в листингах 1-3.

Листинг 1. Описание метода `void _forward_traverse()`

```
void Spline::_forward_traverse()
{
    double k = 0.f;
    double l = 0.f;

    _forward_traverse_coefficients.emplace_back(0, 0);
    _forward_traverse_coefficients.emplace_back(k, l);
    for (int i = 2; i < _points.size(); ++i)
    {
        double denominator = _s(i) - _h(i - 1) * l;
        double new_k = (_r(i) - _h(i - 1) * k) / denominator;
        double new_l = _h(i) / denominator;
        _forward_traverse_coefficients.emplace_back(new_k, new_l);
        k = new_k;
    }
}
```

```

        l = new_l;
    }
}

```

Листинг 2. Описание метода *void _backward_traverse()*

```

void Spline::_backward_traverse()
{
    int n = _forward_traverse_coefficients.size() - 1;
    _coefficients[n].c = _forward_traverse_coefficients.at(n).x();
    for (int i = n - 1; i > 1; --i)
    {
        _coefficients[i].c =
            _forward_traverse_coefficients.at(i).x() -
            _forward_traverse_coefficients.at(i).y() * _coefficients[i + 1].c;
    }
}

```

Листинг 3. Описание метода *void _calculate_abd_coefficients()*

```

void Spline::_calculate_abd_coefficients()
{
    for (int i = 1; i < _points.size(); ++i)
    {
        _coefficients[i].a = _points.at(i - 1).y();

        if (i < _points.size() - 1)
        {
            _coefficients[i].b = (
                _points.at(i).y() - _points.at(i - 1).y()
            ) / _h(i) - (
                _coefficients[i + 1].c + 2 * _coefficients[i].c
            ) * _h(i) / 3;
        } else
        {
            _coefficients[i].b = (
                _points.at(i).y() - _points.at(i - 1).y()
            ) / _h(i) - (2 * _coefficients[i].c) * _h(i) / 3;
        }

        if (i < _points.size() - 1)
        {
            _coefficients[i].d = (
                _coefficients[i + 1].c - _coefficients[i].c
            ) / (3 * _h(i));
        } else
        {
            _coefficients[i].d = -_coefficients[i].c / (3 * _h(i));
        }
    }
}

```

Отрисовка сплайна происходит путем разбиения отрезка между первой и последней опорной точкой на равные отрезки небольшой длины. Для краев данных отрезков вычисляется значение сплайна в данных точках, после чего строится линия, соединяющая эти две краевые точки.

Для настройки сплайна была добавлена возможность интерактивно добавлять новые точки, перемещать и удалять их. При изменении набора опорных точек сплайн перестраивается в реальном времени.

Описание свойств сплайна.

Кубический сплайн состоит из набора сегментов, которые задаются полиномом степени не выше 3-й. Сплайн обладает дефектом 1, т. к. степень интерполирующей сегмент функции составляет 3, а количество непрерывных производных — 2. Для каждого сегмента соблюдается условие гладкости (в краевых точках (с одинаковым x) соседних сегментов совпадают между собой значения функции, первой и второй производных).

В компьютерной графике кубические сплайны используются для определения контуров различных изображений: задается некоторое количество опорных точек, определяющих контур, после чего они объединяются плавной кривой — сплайном.

Тестирование.

Результаты тестирования см. на рис. 4-6.

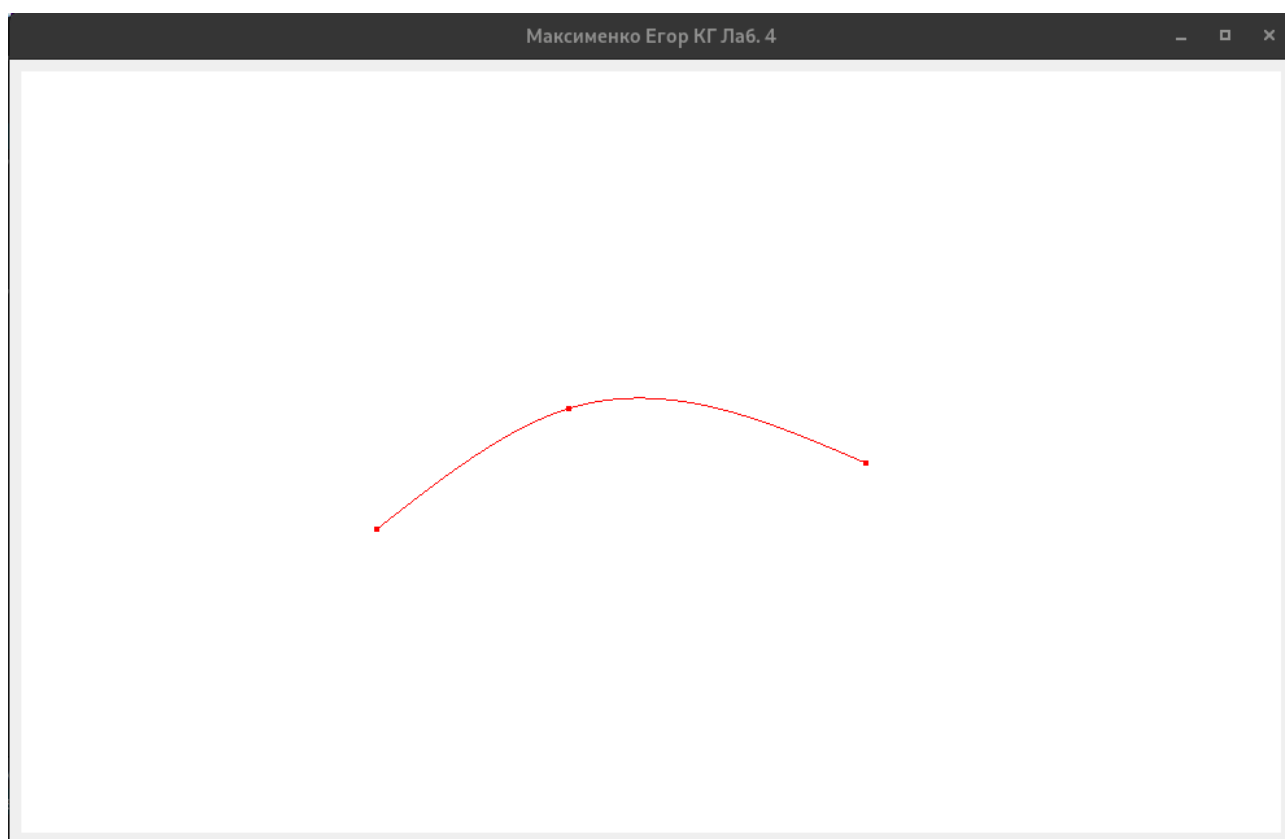


Рисунок 4. Сплайн с 3-мя опорными точками

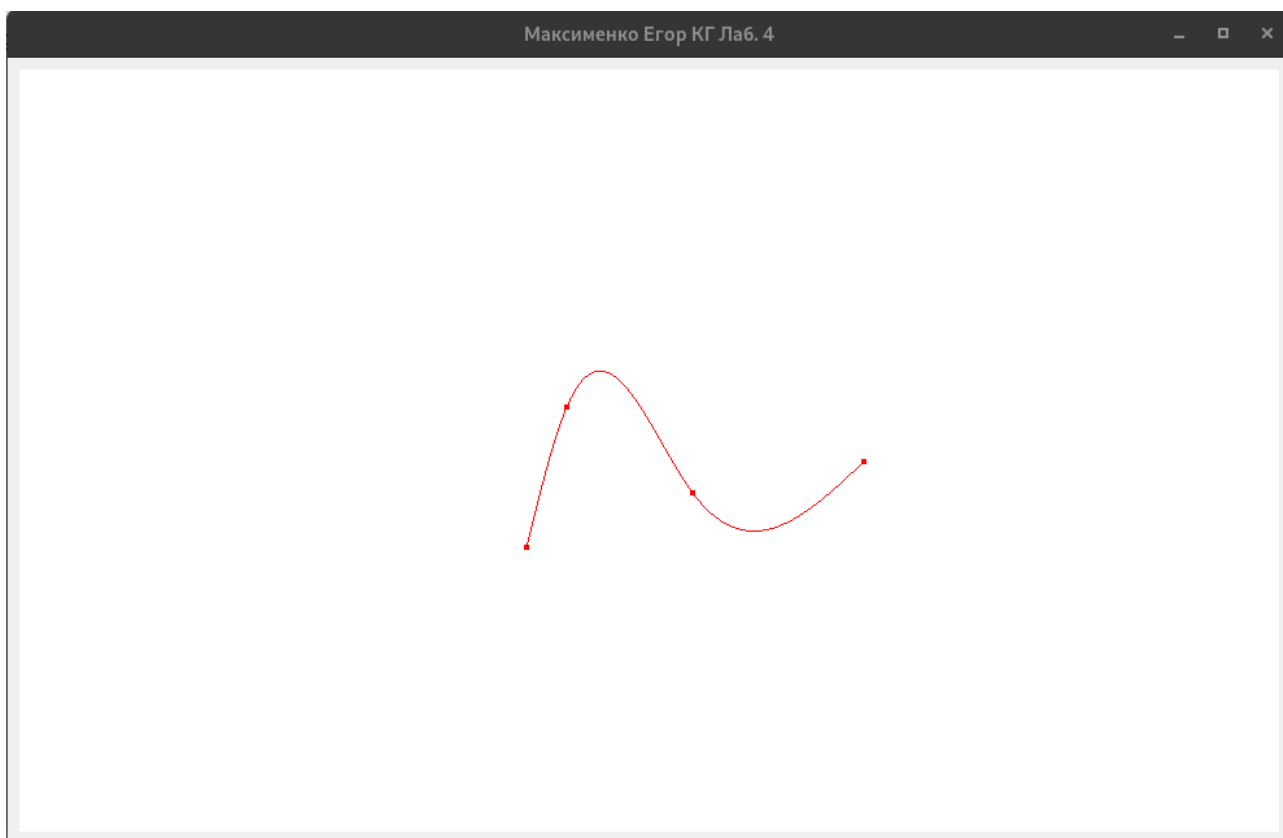


Рисунок 5. Сплайн с 4-мя опорными точками

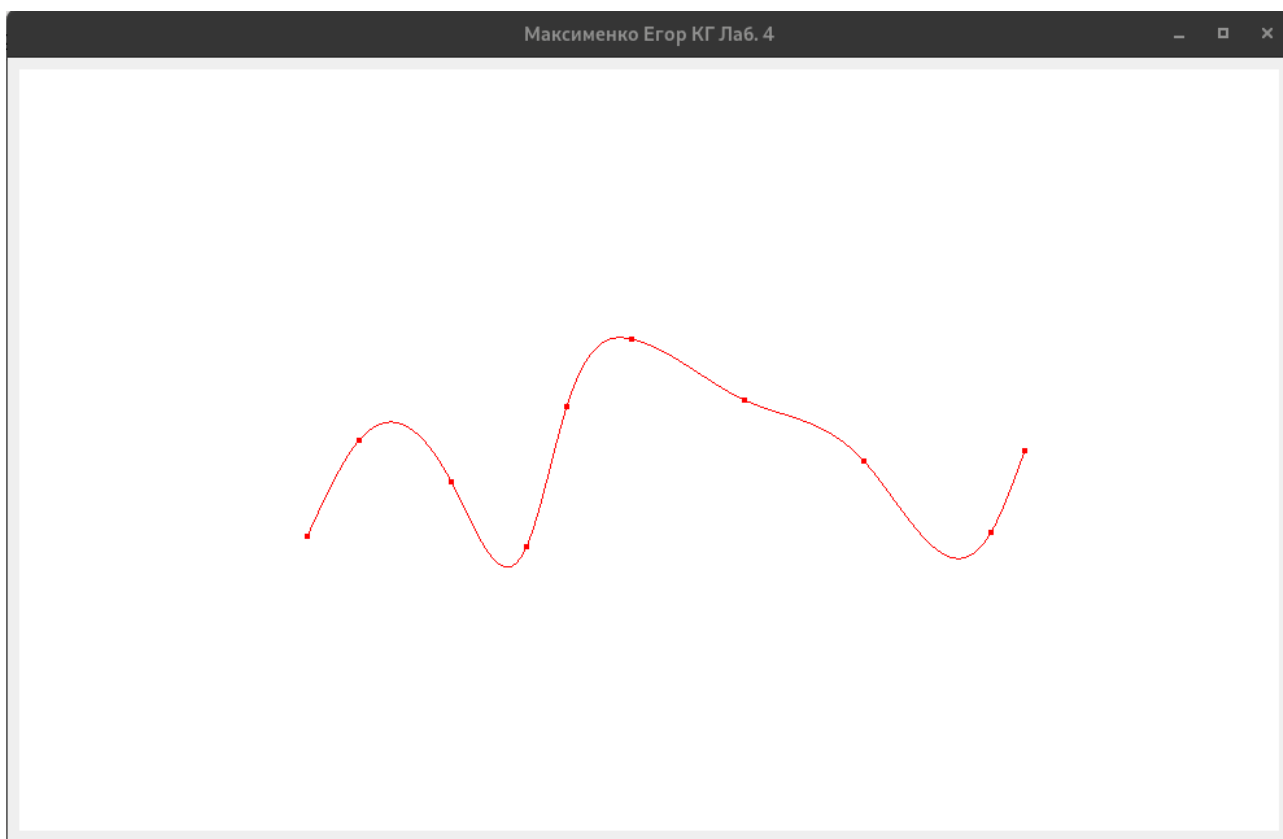


Рисунок 6. Сплайн с 10-ю опорными точками

Вывод

В ходе данной работы были изучены кубические сплайны и область их применения в компьютерной графике. Была разработана программа, позволяющая интерактивно управлять набором опорных точек для визуализации кубического сплайна.