МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Компьютерная графика»

Тема: Кубический сплайны

Вариант 17

Студент гр. 8383	____________________	Киреев К.А.
Студент гр. 8383	____________________	Муковский Д.В.
Преподаватель	____________________	Герасимова Т.В.

Санкт-Петербург

2021

Цель работы.

Реализовать интерактивное приложение, отображающее заданные полиномиальные кривые.

Задание.

B-сплайн (n = 6, k = 4) с равномерным узловым вектором.

В отчете д.б. представлена реализуемая в программе формула, описан алгоритм построения и показаны основные характеристики кривой.

Основные теоретические положения.

Интерполяция B-сплайнами.

Чуть более сложный тип интерполяции – так называемая полиномиальная сплайн-интерполяция, или интерполяция B-сплайнами. В отличие от обычной сплайн-интерполяции, сшивка элементарных B-сплайнов производится не в точках (t_i, x_i), а в других точках, координаты которых обычно предлагается определить пользователю. Таким образом, отсутствует требование равномерного следования узлов при интерполяции B-сплайнами.

Сплайны могут быть полиномами первой, второй или третьей степени (линейные, квадратичные или кубические). Применяется интерполяция B-сплайнами точно так же, как и обычная сплайн-интерполяция, различие состоит только в определении вспомогательной функции коэффициентов сплайна.

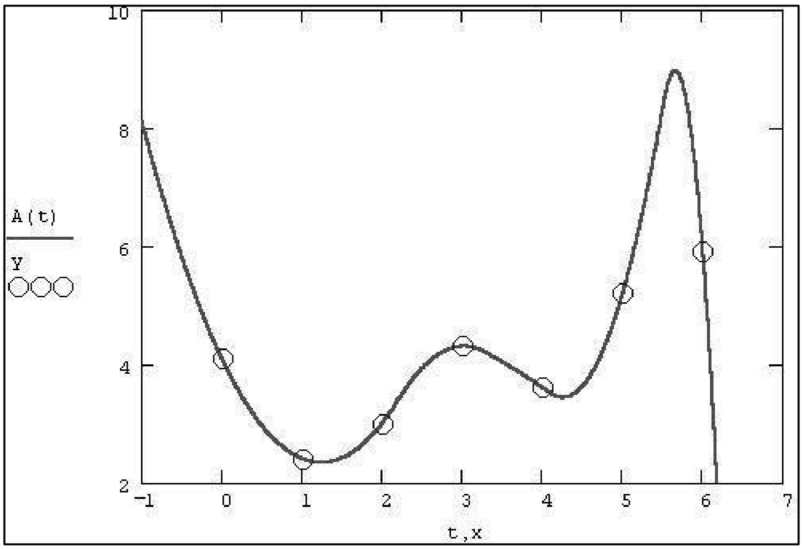


Рис. 1 - Интерполяция B-сплайнами

Наиболее приемлем способ, при котором кривая описывается многочленом 3-й степени:

x(t) =	A11 t3	+	A12 t2	+	A_{13} t	+	A_{14};
y(t) =	A21t3	+	A22t2	+	A_{23}t	+	A_{24};
z(t) =	A31 t3	+	A32t2	+	A_{33}t	+	A34,

$0<t<1$ (переход от точки i к i+1 точке)

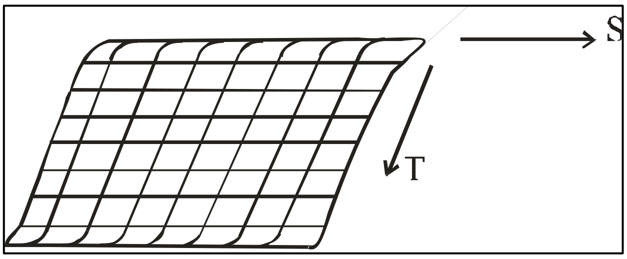
Кубические уравнения выбраны потому, что для сегментов произвольной кривой:

- -не существует представление более низкого порядка, которая обеспечивает сопряжение на границах связи

- -при более высоком порядке, появляются осцилляции и волнистость. Из ряда способов описания бикубических кривых (метод Эрмита, метод Безье и т.п.) наиболее применяем метод B-сплайнов, для которого характерно несовпадение кривой с аппроксимируемыми точками что, однако гарантирует равенство 1-й и 2-й производных при стыковке сегментов. B-сплайн описывается следующей формулой:

  x(t)=TMsGsx – обобщенная форма описания кривой для всех методов где: T=[t3,t2,t,1] – параметр, определяющий переход от точки Pi к Pi +1 M – матрица обобщения для B – сплайна.

Для трехмерных поверхностей определяется два параметра S и T, изменение которых дают координату любой точки на поверхности.



Фиксация одной переменной позволяет перейти к построению кривой на поверхности. Общая форма записи (для направления x):

$x(S,t)=SCxT^T$

где: Cx – коэффициенты кубического многочлена (для определения коэффициентов y,z соответственно Cy,Cz)

Для B-сплайна:
$X(S,t)=SMsPxMs^TT^T$

$Y(S,t)=SMsPyMs^TT^T$

$Z(S,t)=SMsPzMs^TT^T$

P – управляющие точки (16 точек) (4 по S и 4 по T).

Выполнение работы.

Работа была выполнена в среде разработке Qt Creator.

Пусть P(t) определяет кривую как функцию от параметра t, тогда B-сплайн имеет вид:

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t), \qquad t_{\min} \le t < t_{\max}, \qquad 2 \le k \le n+1,$$

где B_i есть n+1 вершина многоугольника, а $N_{i,\ k}$ — нормализованные функции базиса B-сплайна.

Для i-й нормализованной функции базиса порядка k (степени k - 1) функции базиса $N_{i,\,k}(t)$ определяются рекурсивными формулами Кокса—де Бура:

$$N_{i,1}(t) = \begin{cases} 1 & \text{если} \quad x_i \le t < x_{i+1} \\ 0 & \text{иначе} \end{cases}$$

и

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}.$$

Величины x_i — это элементы узлового вектора, удовлетворяющие отношению $x_i < x_{i+1}$. Параметр t изменяется от t_{min} до t_{max} вдоль кривой P(t)[1].

Реализация представлена в листинге 1:

```
float MyWidget::B(float x, int n, int k)
{
    if(k == 0)
    {
        if(knots[n] <= x && x < knots[n+1])
        {
            return 1.0f;
        }
        return 0.0f;
    }

    float a = B(x, n, k-1);
    float b = B(x, n+1, k-1);
    float c = 0.0f, e = 0.0f;

    if(a != 0.0f)
    {
        c = (x - knots[n]) / (knots[n+k] - knots[n]);
    }

    if(b != 0)
    {
        e = (knots[n+k+1] - x) / (knots[n+k+1] - knots[n+1]);
    }

    return (a*c + b*e);
}
```

Узловой вектор имеет длину равную количеству контрольных точек + степень сплайна + 1 (n=6 + k=4 + 1). Пример открытого равномерного узлового вектора для степени 4.

```
[0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]
```

Реализация расчета узлового вектора представлена в листинге 2.

```
void MyWidget::CrKnotVector(){
    QVector <float> knots;
    for(int i = 0; i < k; i++)
    {
        knots.append(0.0f);
    }
    for(int i=0; i < points.length()-k+1; i++)
    {
        knots.append((float)i);
    }
    for(int i=0; i < k;i++)
    {
        knots.append((float)(points.length()-k));
    }
    this->knots=knots;
}
```

Формула для вычисления B-сплайна:

$$X(t) = \sum_i x_i B_{i,n}(t),$$
$$Y(t) = \sum_i y_i B_{i,n}(t),$$

Реализация показана в листинге 3:

```
void MyWidget::B_spline(){
    if(points.length() < 6)
    {
        return;
    }
    glColor3d(0,1,1);
    glBegin(GL_LINE_STRIP);
    CrKnotVector();
    float xmin=knots[0];
    float xmax=knots.last();
    float delta = xmax - xmin;
    float step = delta/300;
    for(float t = xmin; t < xmax; t += step){
        float x = 0.0f, y = 0.0f;
        for(int i = 0; i < points.length(); i++){
            x+=B(t,i,d) * points[i].x() * g_Weight[i];
            y+=B(t,i,d) * points[i].y() * g_Weight[i];
        }
        glVertex2f(x,y);
    }
    glVertex2f(points.last().x(),points.last().y());
    glEnd();
}
```

Степень сплайна и количество контрольных точек выбирается пользователем. Нажатие правой кнопки мыши добавляет контрольную точку, также их можно перемещать удержанием левой кнопки мыши.

Кривая обладает непрерывностью в точках стыковки сегментов, кроме того, непрерывны первые две производные. Кривая обладает гладкостью. Примеры работы программы приведены на рис. 2 и 3.

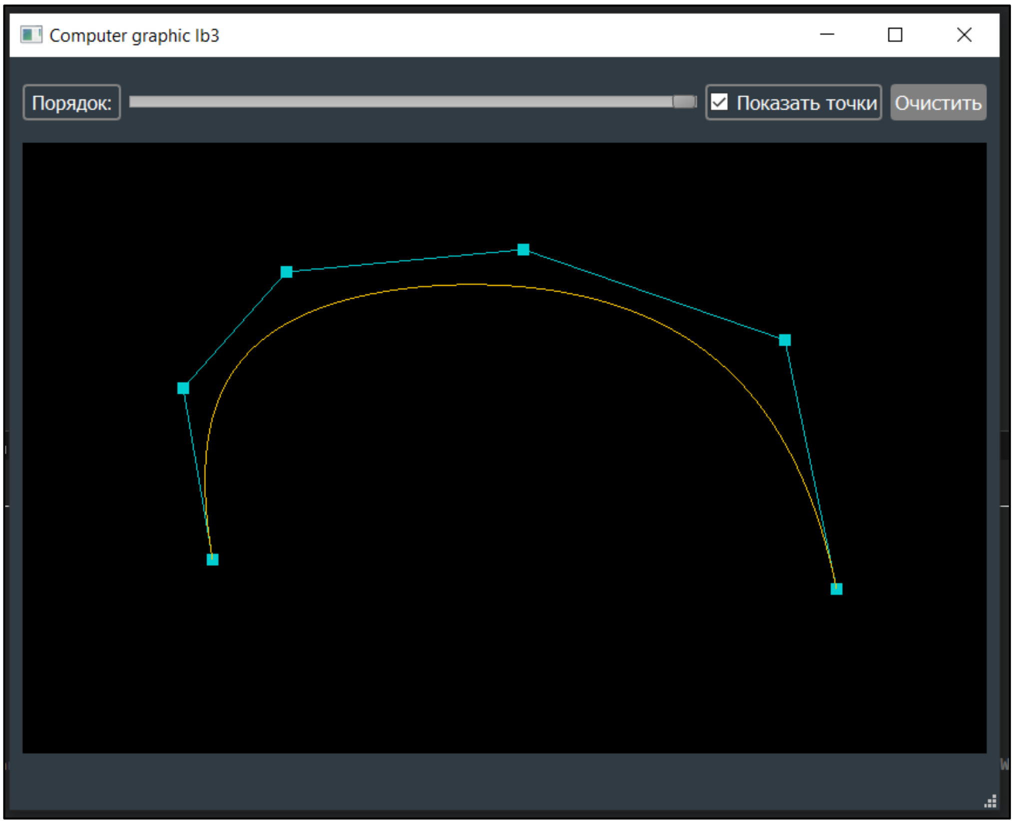


Рис. 2 – Пример работы программы

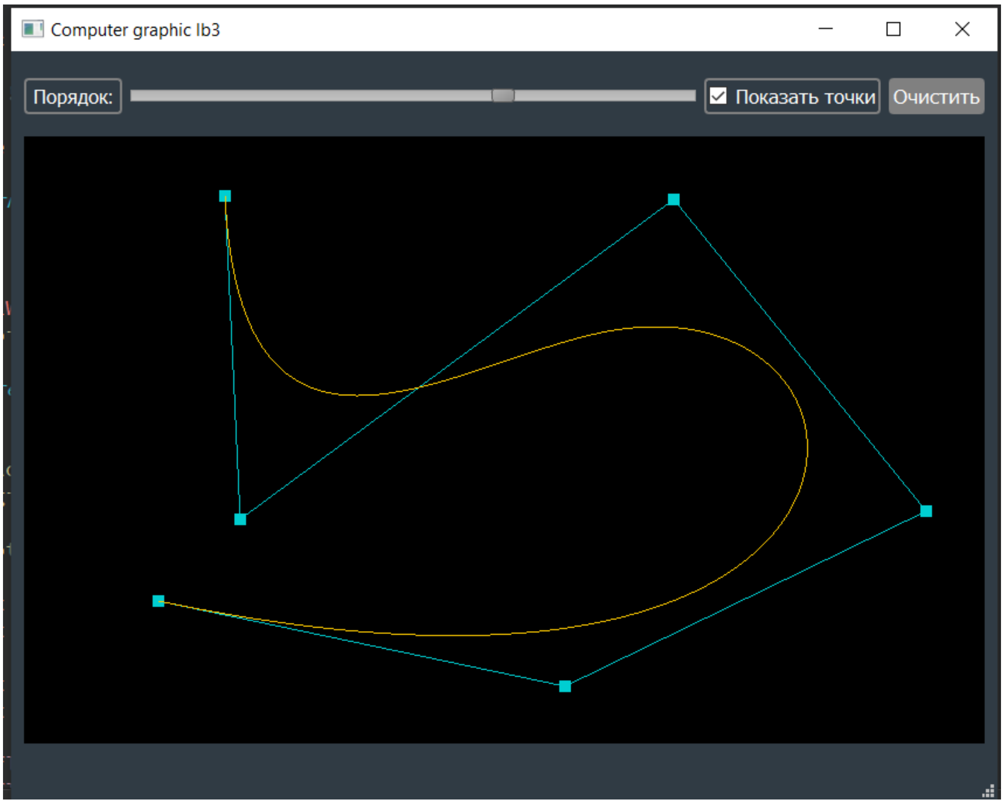


Рис. 3 – Пример работы программы

Выводы.

В итоге лабораторной работы разработано приложение отрисовки B-сплайнов, поддерживающее интерактивное взаимодействие с пользователем, улучшены навыки владения с OpenGL.