Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет: “Информационные технологии и прикладная математика”
Кафедра: 806 “Вычислительная математика и программирование”

Лабораторная работа №7.

Построение плоских полиномиальных кривых.

Студент:	Ефимов А. В.
Группа:	М8О-307Б-18
Преподаватель:	Филиппов Г. С.
Оценка:	____________
Дата:	____________
Подпись:	____________

Москва, 2020

1. Постановка задачи

Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиции точек и, при необходимости, значений касательных векторов и натяжения.
Вариант: Интерполяционный многочлен Лагранжа по шести точкам.

2. Решение задачи

Для данного множества точек $(x_0,y_0),(x_1,y_1),\dots,(x_n,y_n)$будет иметь вид:

$$f(x)=\sum_{i=1}^{n} y_i l_i(x)$$

Где $l_i(x)$ – Лагранжевый базисный полином:

$$l_i(x)=\prod_{\substack{m=0\\ m\neq i}}^{n}\frac{x-x_m}{x_i-x_m}$$

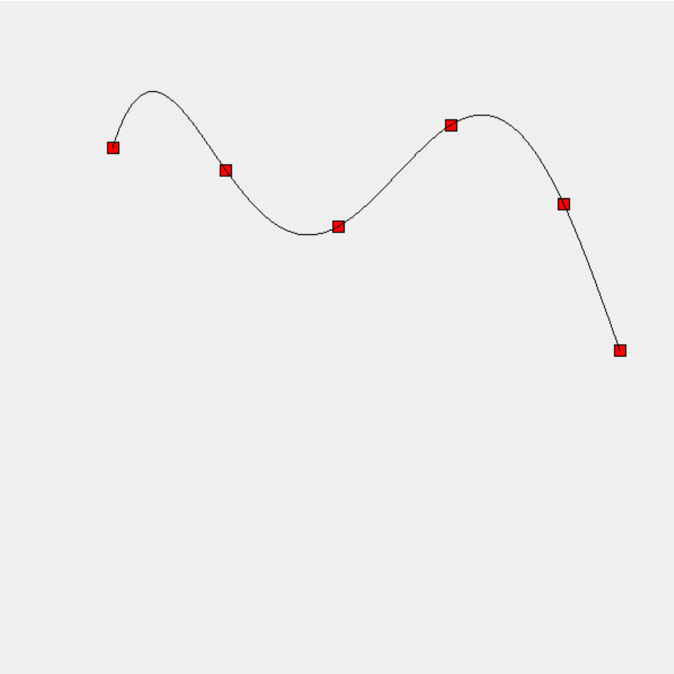
Причем порядок точек никак не влияет на подсчет функции. Пользуясь этим, все точке можно отсортировать по координате X. Это необходимо для:

- Проверки, что ни одна точка не совпадает с другой по координате X – если же это происходит, то базисные полиномы посчитать невозможно из-за деления на 0 (изгибы уходят в бесконечность);
- Нахождения минимального и максимального пределов подсчета кривой – это необходимо, чтобы сузить интервал от $(-\infty,+\infty)$ до $[x_{(0)},x_{(n)}]$, где $x_{(i)}$ – i-ое число в отсортированном множестве.

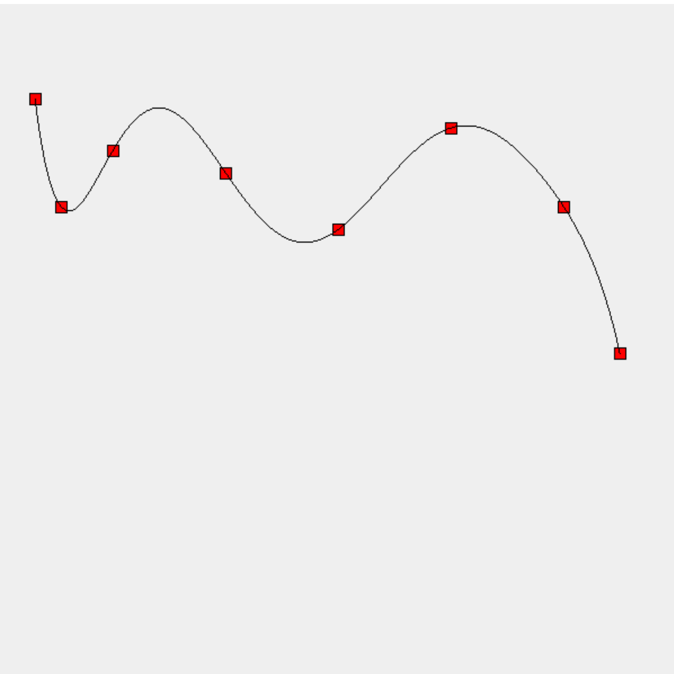
Задав значительно маленький шаг, кривую можно посчитать в виде множества близко расположенных точек, соединенных кривыми.

3. Программа

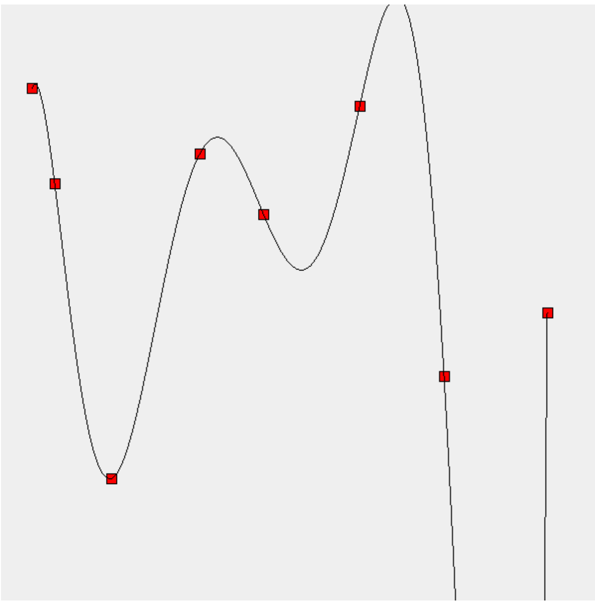
При запуске программы открывается окно с точками шестью точками, соединенные кривой Лагранжа:



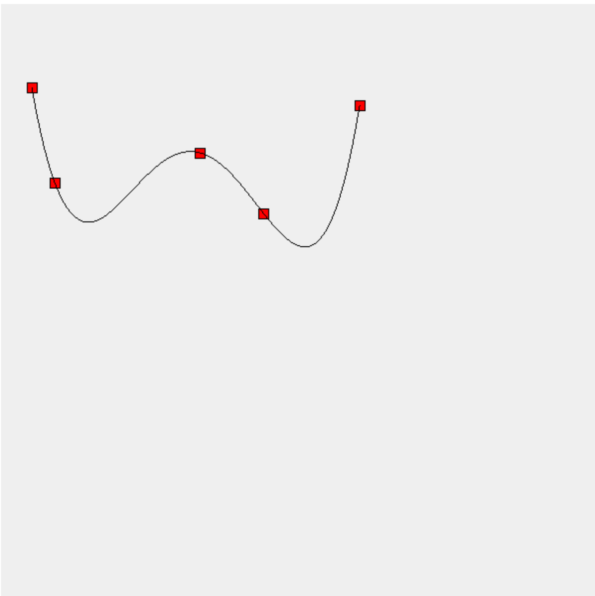
Новые точки можно добавлять двойным щелчком мыши:



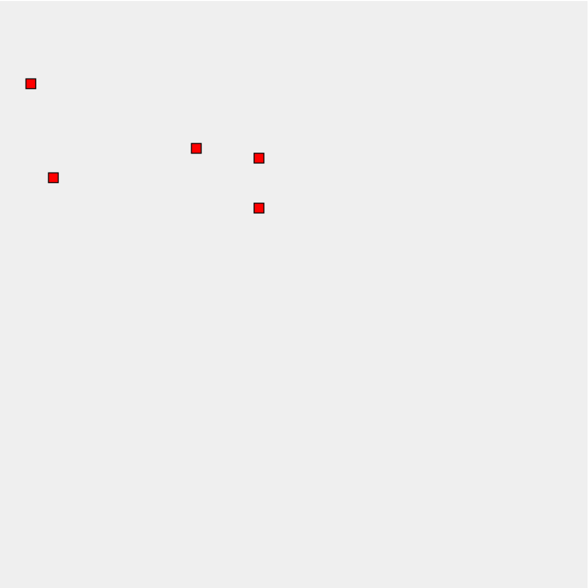
Существующие точки можно двигать:



Удаление происходит правым щелчком мыши:



Если какие-либо две точки имеют одинаковые координаты по X, то линия не считается:



4. Выводы

Полиномиальные кривые можно использовать создания сглаживания поверхностей, посчитав промежуточные точки между заданными точками полигональной сетки (за это, кстати, и отвечает новейший этап в OpenGL – этап Tessellation). Однако, многочлен Лагранжа нельзя использовать для сглаживания в реальных условиях. Несмотря на то, что в лабораторной работе он работал на вид быстро, он имеет квадратичную сложность (и с огромным скрытым коэффициентом, если учитывать маленький шаг при подсчете каждой точки) и для полигональных сеток в режиме реального времени пригодным не является.