Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Компьютерная графика»

Студент: Я. С. Поскряков
Преподаватель: Г. С. Филиппов
Группа: М8О-306Б

Дата:
Оценка:
Подпись:

Москва, 2019

Построение плоских полиномиальных кривых..

Задача: Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможностьизменения позиции точек и, при необходимости, значений касательных векторов и натяжения.

Вариант №14:
Интерполяционный многочлен Лагранжа по шести точкам

1 Описание

В коде лабораторной номер 7 нужно реализовать построение и вывод функции Лагранжа по шести точкам

2 Исходный код

```
#include <iostream>
#include <GL/glut.h>
using namespace std;

double x1[6] = {0,0,0,0,0,0};
double y1[6] = {0,0,0,0,0,0};
double Lagrange(double var)
{
    double s, L;
   // double x1[6] = {-1.0, -0.6, -0.3, 0.3, 1.0, 0.5};
   // double y1[6] = {0.5, 1.0, 0.4, 0.1, 0.5, 0.2};
    L = 0;

    for(int i = 0;i < 6;i++){
        s = 1.0;

        for(int j = 0;j < 6;j++)
            if(j != i)
                s *= ((var - x1[j]) / (x1[i] - x1[j]));

        L += y1[i] * s;
    }

    return L;
}

```

```
27 void Initialize()
28 {
29     glClearColor(1.0, 1.0, 1.0, 1.0);
30     glMatrixMode(GL_PROJECTION);
31     glLoadIdentity();
32     gluOrtho2D(-5, 5, -5, 5);
33     glMatrixMode(GL_MODELVIEW);
34     glLoadIdentity();
35 }
36
37 void Draw()
38 {
39     double x;
40
41     glClear(GL_COLOR_BUFFER_BIT);
42     glColor3f(0.0, 0.0, 0.0);
43     glBegin(GL_LINES);
44
45     glVertex2d(-5, 0);
46     glVertex2d(5, 0);
47     glVertex2d(0, -5);
48     glVertex2d(0, 5);
49     glVertex2d(5, 0);
50     glVertex2d(4.7, 0.2);
51     glVertex2d(5, 0);
52     glVertex2d(4.7, -0.2);
53     glVertex2d(0, 5);
54     glVertex2d(-0.1, 4.7);
55     glVertex2d(0, 5);
56     glVertex2d(0.1, 4.7);
57
58     for(int i = -5; i < 5; i += 1) {
59         glVertex2d(i ,0);
60         glVertex2d(i ,0.1);
61     }
62
63     for(int i = -5; i < 5; i += 1) {
64         glVertex2d(0 ,i);
65         glVertex2d(0.1 ,i);
66     }
67     glEnd();
68
69     glColor3f(1, 0, 0);
70     glPointSize(2.0);
71     glBegin(GL_POINTS);
72
73     for(x = -6.0;x < 6.0;x += 0.001)
74     {
75         glVertex2d(x, Lagrange(x));
```

```
76      }
77
78      glEnd();
79      glFlush();
80  }
81
82  int main(int argc, char**argv)
83  {
84     cout << "Please, enter 6 points in <x> <y>.\n";
85      for(int i = 0; i < 6;++i) {
86          cin >> x1[i] >> y1[i];
87      }
88      glutInit(&argc, argv);
89      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
90      glutInitWindowSize(800, 500);
91      glutCreateWindow("Lagrange");
92      glutDisplayFunc(Draw);
93      Initialize();
94      glutMainLoop();
95      return 0;
96  }
```

3 Консоль

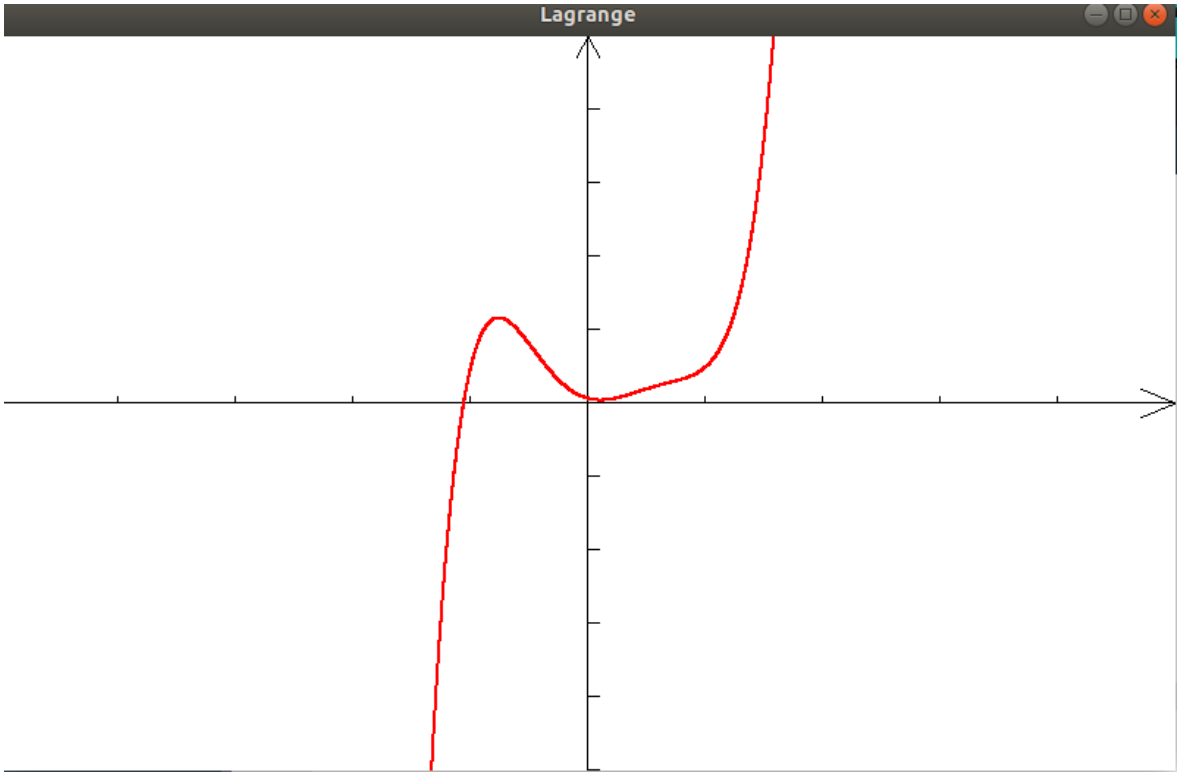
В консоли необходимо скомпилировать исходный код и запустить. Согласно заданию в окне необходимо будет ввести параметры освещения и точность апроксимации.

```
(base) yar@yarmachine:~/CG/already/LAB7$ g++ main.cpp -o test -lGL -lglut -lGLU
(base) yar@yarmachine:~/CG/already/LAB7$ ./test
Please,enter 6 points in <x><y>.
-1 0.5
-0.6 1.0
-0.3 0.4
0.3 0.1
1.0 0.5
0.5 0.2
```

После откроется изображение фигуры в окне.

Это окно можно изменять по размерам и перемещать по экрану без всяких побочных эффектов, фигура подстраивается под изменение размеров экрана и масштабируется соответствующим образом.

С помощью нажатий клавиатуры можно вращать и масштабировать фигуру произвольным образом:



4 Выводы

Выполнив данную лабораторную работу по курсу «Компьютерная графика», я не столкнулся с определенными сложностями, однако узнал, что такое интерполяционный многочлен Лагранжа и его построение.