

## Тема: Построение плоских полиномиальных кривых.

### Цель работы

Написать программу, строящую полиномиальную кривую по заданным точкам. Обеспечить возможность изменения позиций точек и, при необходимости, значений касательных векторов натяжения.

### Задача

Вариант 1. Интерполяционный многочлен лагранжа по 5 точкам.

### Оборудование

Ноутбук с предустановленным QT Creator, видеокарта ATI.

### Теоретическая часть

**Интерполяционный многочлен Лагранжа** — многочлен минимальной степени, принимающий данные значения в данном наборе точек. Для  $n + 1$  пар чисел  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ , где все  $x_i$  различны, существует единственный многочлен  $L(x)$  степени не более  $n$ , для которого  $L(x_i) = y_i$ .

**Лагранж** предложил способ вычисления таких многочленов:

$$L(x) = \sum_{j=0}^n y_j l_j(x)$$

где базисные полиномы определяются по формуле:

$$l_j(x) = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i} = \frac{x - x_0}{x_j - x_0} \dots \frac{x - x_{j-1}}{x_j - x_{j-1}} \frac{x - x_{j+1}}{x_j - x_{j+1}} \dots \frac{x - x_n}{x_j - x_n}$$

Полиномы Лагранжа используются для **интерполяции**, а также для **численного интегрирования**.

Пусть для функции  $f(x)$  известны значения  $y_j = f(x_j)$  в некоторых точках. Тогда мы можем

$$f(x) \approx \sum_{j=0}^n f(x_j) l_j(x)$$

интерполировать эту функцию как

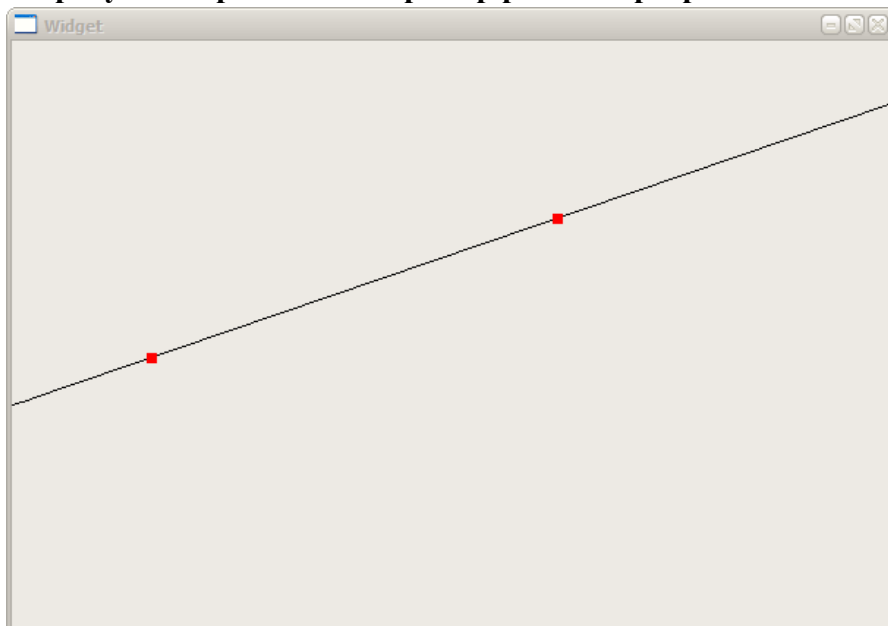
Этот подход используется при отрисовке графика функции.

### Ход решения

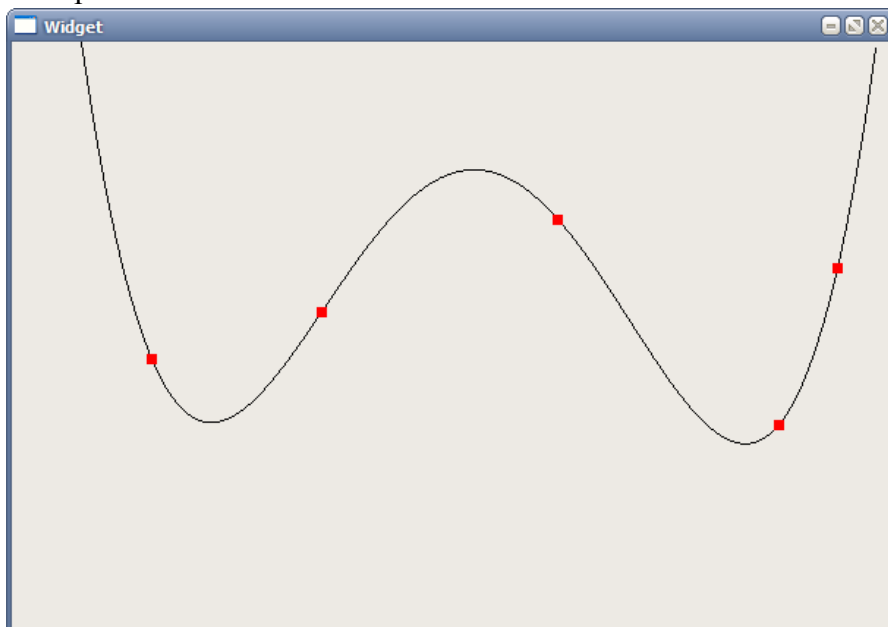
Многочлен Лагранжа строится на основе массива исходных точек специальной функцией(код из Википедии):

```
float lagrange_polynomial (float *x, float *y, int n, float argx)
{
    float c, s=0;
    for (int i=0; i<n; i++)
    {
        c=1;
        for (int j=0; j<n; j++)
            if (i!=j) c*=(argx-x[j])/(x[i]-x[j]);
        s+=c*y[i];
    }
    return s;
}
```

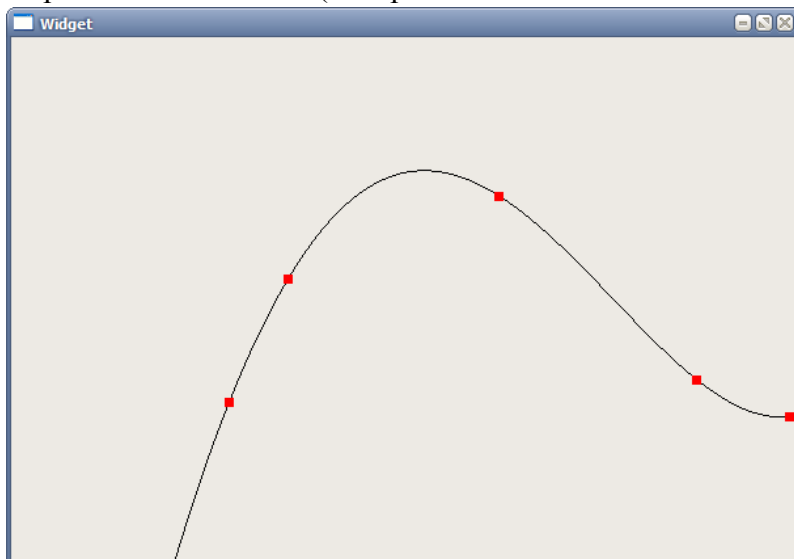
На рисунках представлен пример работы программы:



Построение по пяти точкам...



Перетаскивание точек (центральные остались на своих местах)



## Исходный код

В программе используются следующие исходные файлы:

```
const int NUM = 5;

QPointF Pts[NUM];
float sortX[NUM];
float sortY[NUM];

int pt_num = 0;
int act_point = -1;

void Widget::init()
{
    //    color = Qt::black;
    //    color = QColorDialog::getColor();

    color = Qt::red;

    Pts[0].setX(100.0);
    Pts[0].setY(100.0);

    penP.setWidth(7);
    penP.setColor(color);

    penL.setWidth(1);
    penL.setColor(Qt::black);

}

float lagrange_polynomial (float *x, float *y, int n, float argx);

void Widget::paintEvent(QPaintEvent* event)
{
    QPainter p(this);
    int i,j,t;
    int num_sort;

    int    ox=this->width();
    int    oy=this->height();

    // Begin Sorting
    QPointF z;
    int puz = 0;
    for (i=0; i<pt_num; i++)
    {
        sortX[i] = Pts[i].x();
        sortY[i] = Pts[i].y();
    }
}
```

```

while (puz = 0)
{
    puz = 0;
    for (i=1; i<pt_num; i++)
    {

        if (sortX[i] < sortX[i-1])
        {
            z = QPointF(sortX[i],sortY[i]);
            sortX[i] = sortX[i-1];
            sortY[i] = sortY[i-1];
            sortX[i-1] = z.x();
            sortY[i-1] = z.y();
            puz++;
        }
    }
}

// End Sorting
float zy, zyl, zx, zx1;
p.setPen(penL);
zx1 = 0;
zyl = lagrange_polynomial( sortX, sortY, pt_num, 0 );
for (i = 13; i < ox; i++)
{
    zx = i;
    zy = lagrange_polynomial( sortX, sortY, pt_num, zx );
    if ((zy>0)&&(zy<oy))
    {
        p.drawLine(zx1, zyl, zx, zy);
    }
    zyl = zy;
    zx1 = zx;
}
p.setPen(penP);
for (i = 0; i < pt_num; i++)
{
    p.drawPoint(Pts[i]);
}
}

```

### Замечания

Кривая перестраивается в реальном времени при перетаскивании точек. Поменяв кол-во опорных точек можно константой в коде. При отрисовке точки сортируются.

### Выводы

Многочлен Лагранжа позволяет достаточно легко достроить функцию по опорным точкам что может быть полезно как в математикам так и в компьютерной графике. В ходе выполнения лабораторной работы я познакомился с различными видами кривых и их способами отображения. Считаю тему крйне важной, тк она затрагивает обширную область рисования, когда модель не задана жесткой сеткой полигонов. Приближенное задание поверхностей нужно для трассировки лучей.