Министерство образования и науки РФ

ФГБОУ ВПО «Тамбовский государственный технический университет»

Кафедра « »

Лабораторная работа №5

по дисциплине «Геометрическое моделирование»

Исследование поверхности Безье

Выполнил: студент гр. −31 Д. Ю.

Проверил: С. А.

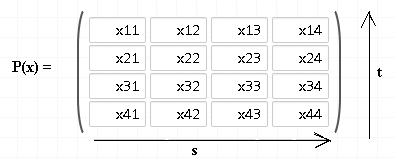
Тамбов,

***Цели и задачи****.*

Построение поверхности Безье по 16 точкам.

***Решение задачи****.*

Для построения необходимо задать поверхность состоящую из 16 точек:

Также и для направлений Y, Z.

Переменные s, t изменяются от 0 до 1.

Основная формула для расчетов имеет следующий вид:

где ,

Процесс нахождения координат сводится к перемножению матриц, S[1;4]\*Mb[4;4] даст вектор SMb[1;4], который умножим на матрицу точек P(x)[4;4], получив вектор P’[1;4]. Опять перемножим вектор на матрицу Mb[4;4], и наконец умножив полученный вектор на TT [4;1] имеем число, которое и необходимо выводить на холст.

Основная часть программы:

/// <summary>

/// Построить поверхность

/// </summary>

**private** void roundButton1\_Click**(object** sender**,** System**.**EventArgs e**)**

**{**

//Чистим холст

Clear**();**

\_list**.**Clear**();**

PointPairList two **=** **new** PointPairList**(),** vert **=** **new** PointPairList**();**

**for** **(**double t **=** 0.01**;** t **<** 1.0**;** t**+=**0.04**)**

**{**

**for** **(**double s **=** 0.01**;** s **<** 1.0**;** s**+=**0.04**)**

**{**

var smb **=** SMb**(new** PointPair4**(**s**\***s**\***s**,** s**\***s**,** s**,** 1.0**));**

var px **=** Px**(**smb**);**

var py **=** Py**(**smb**);**

var pz **=** Pz**(**smb**);**

var pmx **=** PM**(**px**);** var pmy **=** PM**(**py**);** var pmz **=** PM**(**pz**);**

var resX **=** MT**(**pmx**,** **new** PointPair4**(**t**\***t**\***t**,** t**\***t**,** t**,** 1.0**));**

var resY **=** MT**(**pmy**,** **new** PointPair4**(**t**\***t**\***t**,** t**\***t**,** t**,** 1.0**));**

var resZ **=** MT**(**pmz**,** **new** PointPair4**(**t**\***t**\***t**,** t**\***t**,** t**,** 1.0**));**

\_list**.**Add**(**resX**,** resY**);**

two**.**Add**(**resZ**,** resY**);**

vert**.**Add**(**resX**,** resZ**);**

**}**

//Добавить точки разрыва в кривые

\_list**.**Add**(**PointPairBase**.**Missing**,** PointPairBase**.**Missing**);**

two**.**Add**(**PointPairBase**.**Missing**,** PointPairBase**.**Missing**);**

vert**.**Add**(**PointPairBase**.**Missing**,** PointPairBase**.**Missing**);**

**}**

DrawGraph**(**zedGraphControl\_XY**,** \_list**,** 1**);**

DrawGraph**(**zedGraphControl\_YZ**,** two**,** 2**);**

DrawGraph**(**zedGraphControl\_XZ**,** vert**,** 3**);**

**}**

/// <summary>

/// S[1;4]\*Mb[4;4] = SMb[1;4]

/// </summary>

**private** PointPair4 SMb**(**PointPair4 S**)**

**{**

//S.x//S.y//S.z//S.t//S[x,y,z,t]

PointPair4 tmp **=** **new** PointPair4**(**

**(**S**.**X**\***Mb**[**0**].**X **+** S**.**Y**\***Mb**[**1**].**X **+** S**.**Z**\***Mb**[**2**].**X **+** S**.**T**\***Mb**[**3**].**X**),**

**(**S**.**X**\***Mb**[**0**].**Y **+** S**.**Y**\***Mb**[**1**].**Y **+** S**.**Z**\***Mb**[**2**].**Y **+** S**.**T**\***Mb**[**3**].**Y**),**

**(**S**.**X**\***Mb**[**0**].**Z **+** S**.**Y**\***Mb**[**1**].**Z **+** S**.**Z**\***Mb**[**2**].**Z **+** S**.**T**\***Mb**[**3**].**Z**),**

**(**S**.**X**\***Mb**[**0**].**T **+** S**.**Y**\***Mb**[**1**].**T **+** S**.**Z**\***Mb**[**2**].**T **+** S**.**T**\***Mb**[**3**].**T**)**

**);**

**return** tmp**;**

**}**

/// <summary>

/// Tmp[S\*Mb]\*P[4;4] = P[1;4]

/// </summary>

**private** PointPair4 Px**(**PointPair4 SMb**)**

**{**//SMb[x;y;z;t]

PointPair4 tmp **=** **new** PointPair4**(**

**(**SMb**.**X**\***\_pList**[**0**].**X **+** SMb**.**Y**\***\_pList**[**4**].**X **+** SMb**.**Z**\***\_pList**[**8**].**X **+** SMb**.**T**\***\_pList**[**12**].**X**),**

**(**SMb**.**X**\***\_pList**[**1**].**X **+** SMb**.**Y**\***\_pList**[**5**].**X **+** SMb**.**Z**\***\_pList**[**9**].**X **+** SMb**.**T**\***\_pList**[**13**].**X**),**

**(**SMb**.**X**\***\_pList**[**2**].**X **+** SMb**.**Y**\***\_pList**[**6**].**X **+** SMb**.**Z**\***\_pList**[**10**].**X **+** SMb**.**T**\***\_pList**[**14**].**X**),**

**(**SMb**.**X**\***\_pList**[**3**].**X **+** SMb**.**Y**\***\_pList**[**7**].**X **+** SMb**.**Z**\***\_pList**[**11**].**X **+** SMb**.**T**\***\_pList**[**15**].**X**)**

**);**

**return** tmp**;**

**}**

**private** PointPair4 Py**(**PointPair4 SMb**)**

**{**

PointPair4 tmp **=** **new** PointPair4**(**

**(**SMb**.**X **\*** \_pList**[**0**].**Y **+** SMb**.**Y **\*** \_pList**[**4**].**Y **+** SMb**.**Z **\*** \_pList**[**8**].**Y **+** SMb**.**T **\*** \_pList**[**12**].**Y**),**

**(**SMb**.**X **\*** \_pList**[**1**].**Y **+** SMb**.**Y **\*** \_pList**[**5**].**Y **+** SMb**.**Z **\*** \_pList**[**9**].**Y **+** SMb**.**T **\*** \_pList**[**13**].**Y**),**

**(**SMb**.**X **\*** \_pList**[**2**].**Y **+** SMb**.**Y **\*** \_pList**[**6**].**Y **+** SMb**.**Z **\*** \_pList**[**10**].**Y **+** SMb**.**T **\*** \_pList**[**14**].**Y**),**

**(**SMb**.**X **\*** \_pList**[**3**].**Y **+** SMb**.**Y **\*** \_pList**[**7**].**Y **+** SMb**.**Z **\*** \_pList**[**11**].**Y **+** SMb**.**T **\*** \_pList**[**15**].**Y**)**

**);**

**return** tmp**;**

**}**

**private** PointPair4 Pz**(**PointPair4 SMb**)**

**{**

PointPair4 tmp **=** **new** PointPair4**(**

**(**SMb**.**X **\*** \_pList**[**0**].**Z **+** SMb**.**Y **\*** \_pList**[**4**].**Z **+** SMb**.**Z **\*** \_pList**[**8**].**Z **+** SMb**.**T **\*** \_pList**[**12**].**Z**),**

**(**SMb**.**X **\*** \_pList**[**1**].**Z **+** SMb**.**Y **\*** \_pList**[**5**].**Z **+** SMb**.**Z **\*** \_pList**[**9**].**Z **+** SMb**.**T **\*** \_pList**[**13**].**Z**),**

**(**SMb**.**X **\*** \_pList**[**2**].**Z **+** SMb**.**Y **\*** \_pList**[**6**].**Z **+** SMb**.**Z **\*** \_pList**[**10**].**Z **+** SMb**.**T **\*** \_pList**[**14**].**Z**),**

**(**SMb**.**X **\*** \_pList**[**3**].**Z **+** SMb**.**Y **\*** \_pList**[**7**].**Z **+** SMb**.**Z **\*** \_pList**[**11**].**Z **+** SMb**.**T **\*** \_pList**[**15**].**Z**)**

**);**

**return** tmp**;**

**}**

/// <summary>

/// P[1;4]\*MbT[4;4] = PM[1;4]

/// </summary>

**private** PointPair4 PM**(**PointPair4 p**)**

**{**

PointPair4 tmp **=** **new** PointPair4**(**

**(**p**.**X **\*** MbT**[**0**].**X **+** p**.**Y **\*** MbT**[**1**].**X **+** p**.**Z **\*** MbT**[**2**].**X **+** p**.**T **\*** MbT**[**3**].**X**),**

**(**p**.**X **\*** MbT**[**0**].**Y **+** p**.**Y **\*** MbT**[**1**].**Y **+** p**.**Z **\*** MbT**[**2**].**Y **+** p**.**T **\*** MbT**[**3**].**Y**),**

**(**p**.**X **\*** MbT**[**0**].**Z **+** p**.**Y **\*** MbT**[**1**].**Z **+** p**.**Z **\*** MbT**[**2**].**Z **+** p**.**T **\*** MbT**[**3**].**Z**),**

**(**p**.**X **\*** MbT**[**0**].**T **+** p**.**Y **\*** MbT**[**1**].**T **+** p**.**Z **\*** MbT**[**2**].**T **+** p**.**T **\*** MbT**[**3**].**T**)**

**);**

**return** tmp**;**

**}**

/// <summary>

/// Конечн число

/// </summary>

**private** double MT**(**PointPair4 Pm**,** PointPair4 t**)**

**{**

double tmp **=** **(**Pm**.**X**\***t**.**X **+** Pm**.**Y**\***t**.**Y **+** Pm**.**Z**\***t**.**Z **+** Pm**.**T**\***t**.**T**);**

**return** tmp**;**

**}**

Здесь поставлены два цикла for для назначения значений переменным s, t, внутри по отдельности высчитываются значения перемножаемых матриц. Полученные итоговые числа вносятся в массивы точек с последующим выводом на холст.

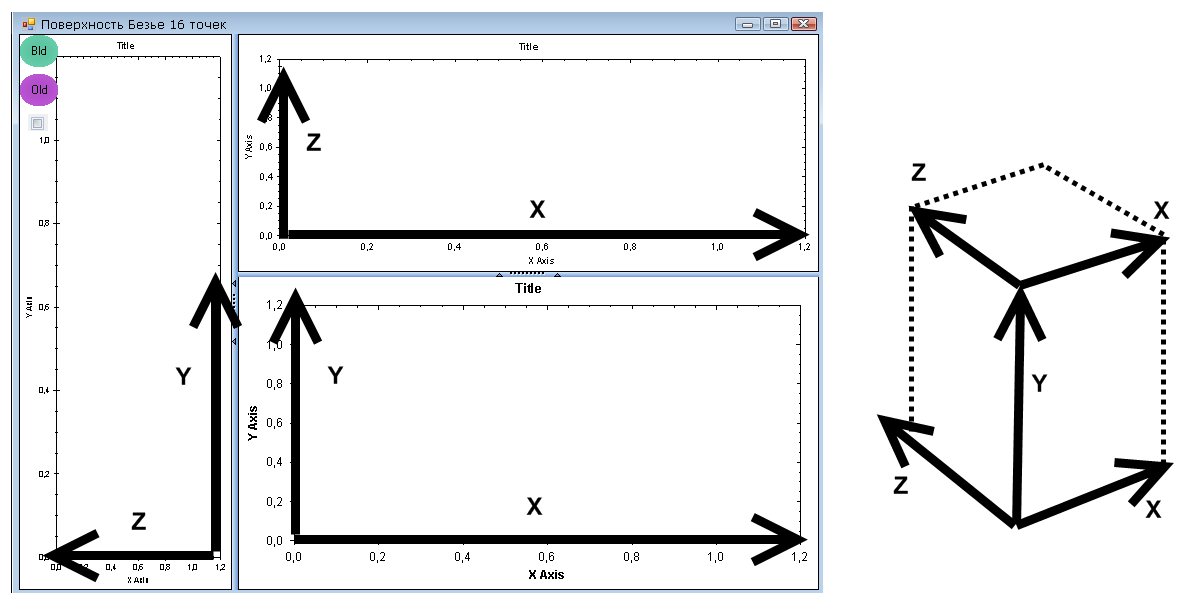
Так как задача требует трехмерного визуального вывода, то разбиваю плоскость на отдельные части (Рисунок 1) и вывожу изображение в каждую плоскость.

Рисунок Система координат

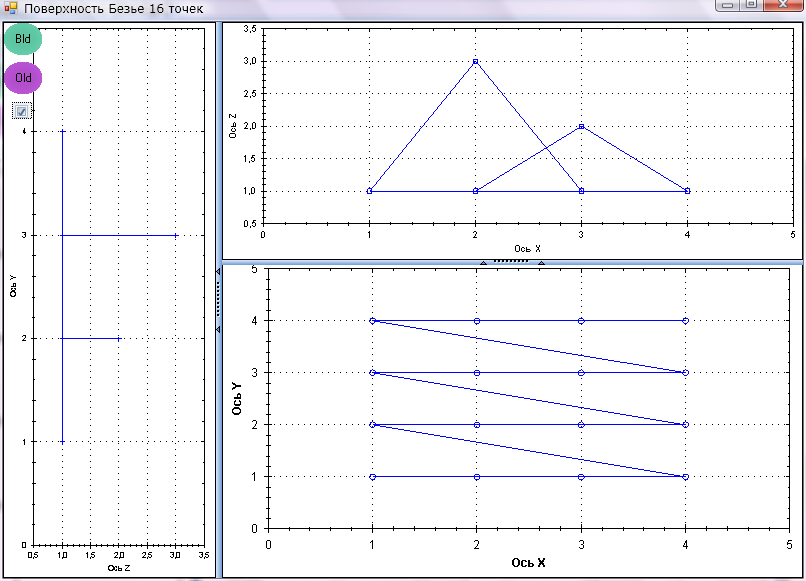
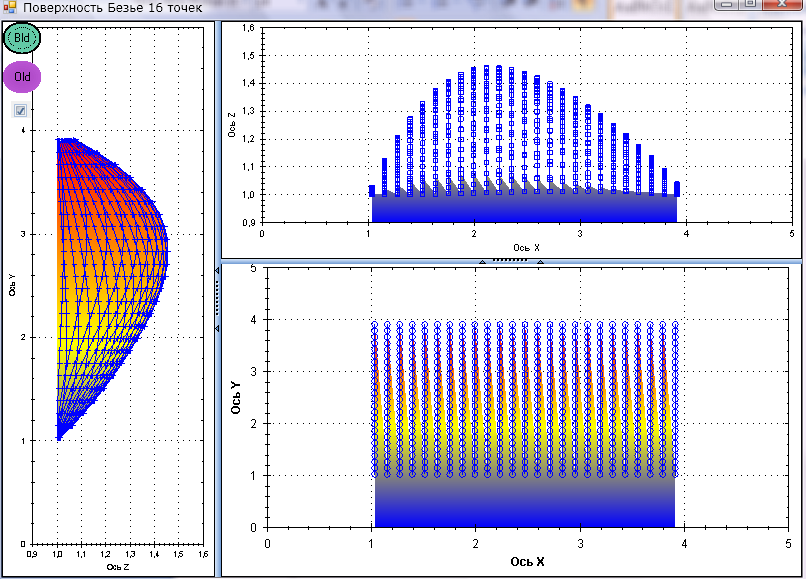
Пример работы программы

Рисунок 3 Построенная поверхность

Рисунок 2 Каркас

Вывод

Поверхность Безье берет своей простотой формулы, хоть это и работа с матрицами. Если учесть тот факт, что 16 точек это не предел алгоритма, то можно попытаться обработать какие-нибудь сложные поверхности. В плане оптимизации можно попытаться избавиться от оператора умножения.