



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «»

---

КАФЕДРА «»

---

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

### *НА ТЕМУ:*

*«Моделирование построения поверхностных и  
объемных геометрий с помощью операции движения»*

*2024 г.*

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Кривые и поверхности</b>	<b>4</b>
1.1 Способы описания кривых и поверхностей . . . . .	4
1.1.1 Явный вид . . . . .	4
1.1.2 Неявный вид . . . . .	4
1.1.3 Параметрический вид . . . . .	4
1.2 Кривые и поверхности Безье . . . . .	5
1.2.1 Кривые Безье . . . . .	5
1.2.2 Поверхности Безье . . . . .	6
1.3 Рациональные кривые и поверхности Безье . . . . .	8
1.3.1 Рациональные кривые Безье . . . . .	8
1.3.2 Рациональные поверхности Безье . . . . .	10
1.4 В-сплайны . . . . .	11
1.4.1 В-сплайн кривая . . . . .	11
1.4.2 В-сплайн поверхности . . . . .	15

# ВВЕДЕНИЕ

Для моделирование построения поверхностных и объемных геометрий необходимо использовать методы геометрического моделирования.

Поэтому, прежде, чем мы приступим к описанию построения геометрий с помощью операции движения, рассмотрим основные методы построения поверхностей и кривых, а именно кривые и поверхности Безье, рациональные кривые и поверхности Безье, B-Spline и NURBS.

# 1 Кривые и поверхности

## 1.1 Способы описания кривых и поверхностей

Существует три основных подхода к описанию кривых и поверхностей.

### 1.1.1 Явный вид

Для кривой:

$$y = f(x), z = g(x)$$

Для поверхности:

$$z = f(x, y)$$

Этот метод имеет несколько недостатков:

- Нельзя однозначно описать замкнутые кривые, например, окружности.
- Полученное описание не обладает инвариантностью относительно поворотов.
- При попытке задать кривые с очень большими углами наклона возникают большие вычислительные сложности.

### 1.1.2 Неявный вид

$$f(x, y, z) = 0$$

Недостатки:

- Кривая в трёхмерном пространстве задаётся как пересечение двух поверхностей, т.е. требуется решать систему алгебраических уравнений.
- Сложности в процессе объединения неявно заданных фрагментов кривых

### 1.1.3 Параметрический вид

Параметрическое задание кривой и поверхности преодолевает недостатки явного и неявного способов описания. С его помощью можно задавать многозначные кривые, т.е. такие зависимости, которые могут принимать несколько значений при одном значении аргумента.

Для кривой:

$$\begin{cases} x = x(u) \\ y = y(u) \\ a \leq u \leq b \end{cases} \quad (1.1)$$

и также будем пользоваться обозначением

$$\mathbf{C}(u) = (x(u), y(u)), \quad a \leq u \leq b \quad (1.2)$$

Для поверхности:

$$\begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \\ a \leq u \leq b \\ c \leq v \leq d \end{cases} \quad (1.3)$$

и также будем пользоваться обозначением

$$\mathbf{S}(u, v) = (x(u, v), y(u, v), z(u, v)), \quad a \leq u \leq b, \quad c \leq v \leq d \quad (1.4)$$

## 1.2 Кривые и поверхности Безье

### 1.2.1 Кривые Безье

Пусть заданы  $n + 1$  точек  $\mathbf{P}_i = (x_i, y_i, z_i)$ , называемых контрольными точками. Они определяют форму и пространственное положение кривой.

Тогда кривую Безье  $n$ -ой степени можно задать с помощью уравнения:

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i, \quad 0 \leq u \leq 1 \quad (1.5)$$

где  $B_{i,n}$  - полиномы Бернштейна.

$$B_{i,n}(u) = C_n^i u^i (1 - u)^{n-i} = \frac{n!}{i!(n-i)!} u^i (1 - u)^{n-i} \quad (1.6)$$

Для вычисления точек кривой Безье удобно использовать алгоритм де Кастельжо:

## Листинг 1.1 – Псевдокод алгоритма де Кастельжо

```

1 deCasteljau(P, n, u, C)
2 {
3     /*Вычисление точки на кривой Безье*/
4     /*[in]: P, n, u*/
5     /*[out]: C (точка)*/
6     for(i=0; i<=n; i++) /* Используем локальный массив, */
7         Q[i] = P[i]      /* чтобы не изменить исходный массив
8                             контрольных точек */
9     for(k=1; k<=n; k++)
10         for(i=0; i<=n-k; i++)
11             Q[i] = (1.0-u)*Q[i] + u*Q[i+1]
12     C = Q[0]
13 }
```

Например, на рисунке 1.1 показана кривая Безье для контрольных точек  $P_1 = (0, 0)$ ,  $P_2 = (0, 1)$ ,  $P_3 = (1, 2)$ ,  $P_4 = (3, 0)$ .

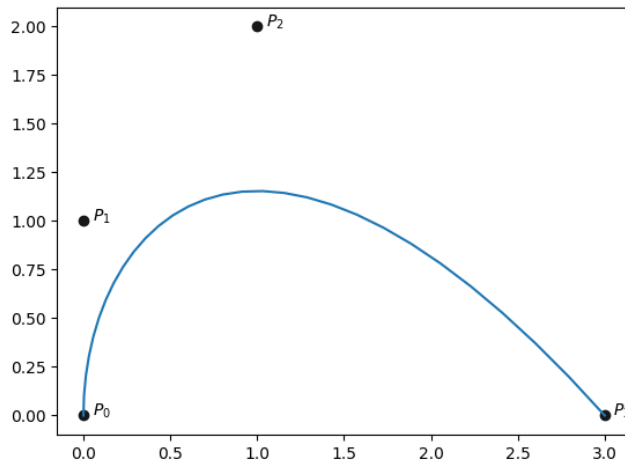


Рисунок 1.1 – Пример кривой Безье

### 1.2.2 Поверхности Безье

Пусть заданы контрольные точки  $P_{i,j}$ , где  $0 \leq i \leq n$  и  $0 \leq j \leq m$ . Тогда поверхность Безье можно задать с помощью следующего уравнения:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) P_{i,j}, \quad 0 \leq u, v \leq 1 \quad (1.7)$$

Аналогично кривым Безье, точки поверхности Безье можно находить с помощью алгоритма де Кастельжо из листинга 1.1.

Листинг 1.2 – Псевдокод алгоритма де Кастельжо для поверхности

```
1 deCasteljauForSurface(P, n, m, u0, v0, S)
2 {
3     /*Вычисление точки на поверхности Безье*/
4     /*[in]: P, n, m, u0, v0*/
5     /*[out]: S (точка)*/
6     if (n <= m)
7     {
8         for(j=0; j<=m; j++) /* P[j][] - j-ая строка */
9             deCasteljau(P[j][], n, u0, Q[j]);
10        deCasteljau(Q, m, v0, S);
11    }
12    else
13    {
14        for(i=0; i<=n; i++)
15            deCasteljau(P[][i], m, v0, Q[i]);
16        deCasteljau(Q, n, u0, S);
17    }
18 }
```

На рисунке 1.2 показан пример поверхности Безье для 15 контрольных точек.

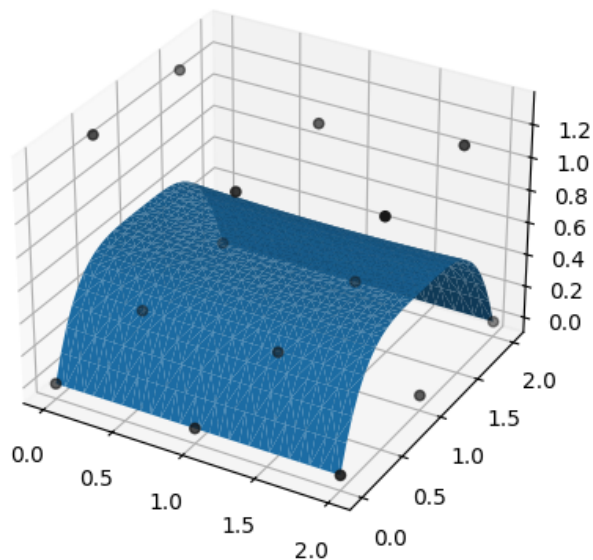


Рисунок 1.2 – Пример поверхности Безье

## 1.3 Рациональные кривые и поверхности Безье

### 1.3.1 Рациональные кривые Безье

Так как кривые Безье - полиномиальные кривые, они имеют существенный недостаток, а именно с их помощью невозможно задать некоторые виды кривых, такие как окружности, эллипсы, гиперболы и прочие. Данные виды кривых можно задать с помощью рациональных функций, то есть как частное двух полиномов.

$$x(u) = \frac{X(u)}{W(u)} \quad y(u) = \frac{Y(u)}{W(u)}, \quad (1.8)$$

где  $X(u)$ ,  $Y(u)$  и  $W(u)$  - полиномы.

Заметим также, что каждая координатная функция имеет одинаковый знаменатель  $W(u)$ .

Рациональные кривые с координатными функциями в виде (1.8) имеют элегантную геометрическую интерпретацию, которая дает эффективные методы построения этих кривых и небольшие требования к памяти компьютера.

Оказывается, что можно использовать однородные координаты, чтобы задать рациональные кривые в  $n$ -мерном пространстве с помощью полиномиальной кривой в  $(n + 1)$ -мерном пространстве.

Рассмотрим точку в евклидовом пространстве  $\mathbf{P} = (x, y, z)$ . Затем запишем точку  $\mathbf{P}$  как  $\mathbf{P}^\omega = (\omega x, \omega y, \omega z, \omega) = (X, Y, Z, W)$  в четырех-мерном пространстве, причем  $\omega \neq 0$ . Тогда  $\mathbf{P}$  можно получить из  $\mathbf{P}^\omega$  делением всех координат на четвертую координату  $W$ , то есть с помощью отображения  $P^\omega$  на гиперплоскость  $W = 1$

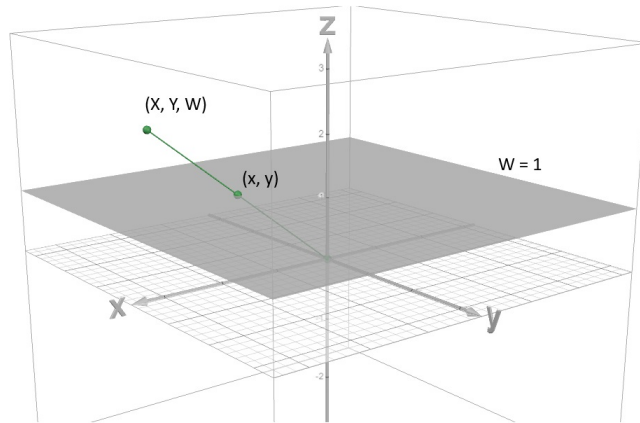


Рисунок 1.3 – Представление точки евклидова пространства в однородной форме для двумерного случая



Данное отображение  $H$  является перспективной проекцией с центром в начале координат:

$$\mathbf{P} = H\{\mathbf{P}^\omega\} = H\{(X, Y, Z, W)\} = \left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}\right) \quad (1.9)$$

Тогда для множества контрольных точек  $\{\mathbf{P}_i\}$  и множества весов  $\{\omega_i\}$  зададим множество взвешенных контрольных точек  $\mathbf{P}_i^\omega = (\omega_i x_i, \omega_i y_i, \omega_i z_i, \omega_i)$ . Тогда нерациональная (полиномиальная) кривая Безье в 4-х мерном пространстве

$$\mathbf{C}^\omega(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i^\omega \quad (1.10)$$

Уравнение (1.10) в координатном виде:

$$X(u) = \sum_{i=0}^n B_{i,n}(u) \omega_i x_i \quad Y(u) = \sum_{i=0}^n B_{i,n}(u) \omega_i y_i$$

$$Z(u) = \sum_{i=0}^n B_{i,n}(u) \omega_i z_i \quad W(u) = \sum_{i=0}^n B_{i,n}(u) \omega_i$$

Заметим, что  $W \neq 0$  поскольку мы выбираем  $\omega_i > 0$ .

Применяя к (1.10) отображение (1.9), получим искомую рациональную кривую Безье в 3-х мерном пространстве, задающуюся формулами

$$x(u) = \frac{X(u)}{W(u)} = \frac{\sum_{i=0}^n B_{i,n}(u) \omega_i x_i}{\sum_{i=0}^n B_{i,n}(u) \omega_i}$$

$$y(u) = \frac{Y(u)}{W(u)} = \frac{\sum_{i=0}^n B_{i,n}(u) \omega_i y_i}{\sum_{i=0}^n B_{i,n}(u) \omega_i}$$

$$z(u) = \frac{Z(u)}{W(u)} = \frac{\sum_{i=0}^n B_{i,n}(u) \omega_i z_i}{\sum_{i=0}^n B_{i,n}(u) \omega_i}$$

или в векторной записи

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n B_{i,n}(u)\omega_i \mathbf{P}_i}{\sum_{i=0}^n B_{i,n}(u)\omega_i} \quad (1.11)$$

Например, если взять  $\mathbf{P}_0 = (1, 0)$ ,  $\mathbf{P}_1 = (1, 1)$ ,  $\mathbf{P}_2 = (0, 1)$  и  $\omega_i = (1, 1, 2)$ , получим дугу окружности (рисунок 1.4).

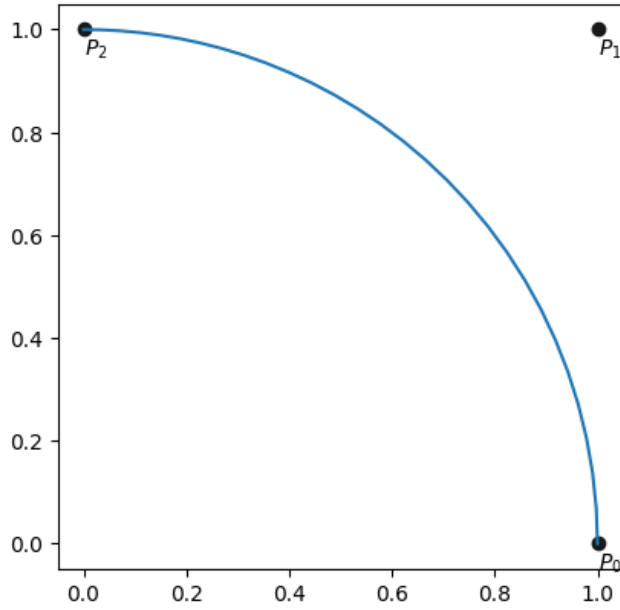


Рисунок 1.4 – Дуга окружности, построенная с помощью рациональной кривой Безье

Если веса всех вершин равны, то получим обычную кривую Безье, поскольку в таком случае знаменатель в уравнение (1.11) – это просто сумма полиномов Бернштейна, которая равна 1. Таким образом, рациональные кривые Безье являются обобщением полиномиальных кривых Безье.

### 1.3.2 Рациональные поверхности Безье

Аналогично рациональным кривым Безье, рациональные поверхности Безье можно представить как перспективную проекцию 4-х мерной полиномиальной поверхности Безье

$$\mathbf{S}^\omega(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \mathbf{P}_{i,j}^\omega$$

$$\mathbf{S}(u, v) = H\{\mathbf{S}^\omega(u, v)\} = \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \omega_{i,j}} = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) \mathbf{P}_{i,j}, \quad (1.12)$$

где

$$R_{i,j}(u, v) = \frac{B_{i,n}(u) B_{j,m}(v)}{\sum_{r=0}^n \sum_{s=0}^m B_{r,n}(u) B_{s,m}(v) \omega_{r,s}} \quad (1.13)$$

На рисунке 1.5 изображена цилиндрическая поверхность, построенная с помощью рациональной поверхности Безье. Она представляет собой поверхность, полученную движением дуги окружности из рисунка 1.4.

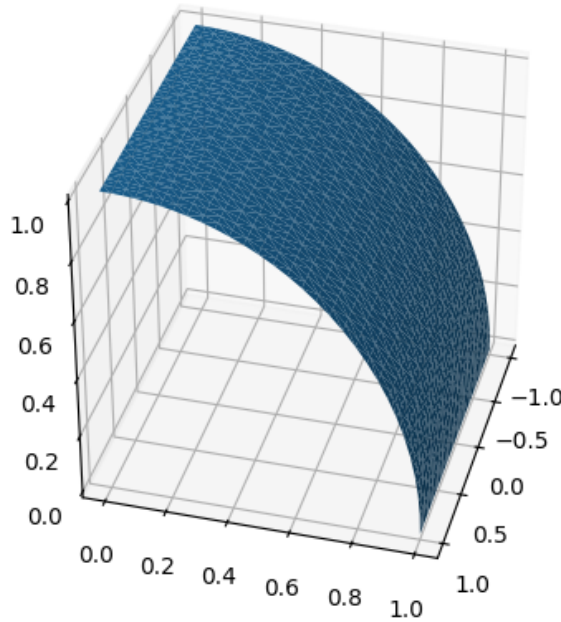


Рисунок 1.5 – Цилиндрическая поверхность, построенная с помощью рациональной поверхности Безье

## 1.4 В-сплайны

### 1.4.1 В-сплайн кривая

У кривых заданных полиномами или рациональными функциями есть несколько минусов.

- Для большого числа точек требуется полиномы большой степени. Так для того, чтобы построить кривую через  $n$  точек, требуется полином  $n - 1$  степени. Кривые, заданные полиномами с большими степенями, тяжело обрабатывать, а также они численно неустойчивы.
- Для сложных кривых также требуется большая степень полинома.
- Полиномиальные кривые не очень подходят для проектирования кривой. Хотя в кривых Безье и можно менять форму кривой, изменяя контрольные точки и значения весов в них, кривая меняется нелокально, т.е. изменение параметров одной точки меняет всю кривую.

В-сплайны лишены этих недостатков: степень полинома В-сплайна можно задать независимо от числа контрольных точек, а также они В-сплайны допускают локальный контроль над формой кривой.

Поставим задачу следующим образом. Пусть даны контрольные точки  $\mathbf{P}_i$ . Определим кривую по формуле

$$C(u) = \sum_{i=0}^n N_i(u) \mathbf{P}_i, \quad u_{min} \leq u \leq u_{max} \quad (1.14)$$

где  $N_i(u)$  - набор кусочно-полиномиальных функций, таких, что

1.  $N_i(u) = 0$  при  $u \notin [a_i, b_i] \subset [u_{min}, u_{max}]$ ;
2.  $N_i(u)$  линейно независимы и образуют базис;
3.  $\sum_{i=0}^n N_i(u) = 1$  для каждого  $u \in [u_{min}, u_{max}]$ .

Решение поставленной задачи даётся В-сплайнами (сокр. от basis).

Общее выражение для расчёта координат точек В-сплайна:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i \quad (1.15)$$

В 1972 году Кокс и де Бур предложили использовать функции  $N_{i,p}$ , определяемые рекурсивно. Пусть  $U = \{u_0, \dots, u_m\}$  - неубывающая последовательно вещественных чисел, т.е.  $u_i \leq u_{i+1}, i = 0, \dots, m - 1$ .  $u_i$  называют

узлами (knot), а  $U$  - вектором узлов (knot vector). Тогда  $i$ -тая базисная функция В-сплайна  $p$ -ой степени, обозначаемой  $N_{i,p}(u)$ , выражается следующим образом:

$$N_{i,0}(u) = \begin{cases} 1, u \in [u_i, u_{i+1}] \\ 0, u \notin [u_i, u_{i+1}] \end{cases} \quad (1.16)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (1.17)$$

В силу свойств базисных функций В-сплайна в любом заданном промежутке  $[u_i, u_{i+1}]$  могут быть отличны от нуля только  $p + 1$  функций:  $N_{i-p,p}, \dots, N_{i,p}$ . Например, единственные кубические функции, отличные от нуля на  $[u_3, u_4]$  - это функции  $N_0^3, \dots, N_3^3$ .

Поэтому при вычислении базисных функций  $N^k$  в точке  $u$ , важно уметь находить индекс  $i$  в векторе узлов, при котором выполняется соотношение  $u_i \leq u \leq u_{i+1}$ . Для этого например можно использовать алгоритм бинарного поиска:

Листинг 1.3 – Алгоритм бинарного поиска индекса  $i$

```

1  int FindSpan(n, p, u, U)
2  {
3      /* Поиск i-го индекса в векторе узлов*/
4      /*[in]: n, p, u, U*/
5      /*[out]: i*/
6      if(u == U[n+1]) return n; /*Специальный случай*/
7      low = p; high = n+1; /*Бинарный поиск*/
8      mid = (low+high)/2;
9      while(u < U[mid] || u >= U[mid+1])
10     {
11         if(u < U[mid])    high = mid;
12         else              low = mid;
13         mid = (low+high)/2;
14     }
15     return mid;
16 }
```

Для вычисления ненулевых базисных функций  $N_{i-p,p}, \dots, N_{i,p}$  можно использовать следующий алгоритм:

Листинг 1.4 – Алгоритм вычисления базисных функций  $N_{i-p,p}, \dots, N_{i,p}$

```

1  BasicFunc(i, u, p, U, N)
```

```

2 {
3     /*Вычисление ненулевых базисных функций*/
4     /*[in]: i, u, p, U*/
5     /*[out]: N*/
6     N[0] = 1.0;
7     for(j=1; j<=p; j++){
8         left[j] = u-U[i+1-j];
9         right[j] = U[i+j]-u;
10        saved = 0.0;
11        for(r=0; r<j; r++)
12        {
13            temp = N[r]/(right[r+1]+left[j-r]);
14            N[r] = saved+right[r+1]*temp;
15            saved = left[j-r]*temp;
16        }
17        N[j] = saved;
18    }
19 }

```

Так, например, для контрольных точек  $\mathbf{P}_0 = (0, 0)$ ,  $\mathbf{P}_1 = (0, 1)$ ,  $\mathbf{P}_2 = (1, 2)$ ,  $\mathbf{P}_3 = (3, 0)$  и узлового вектора  $U = \{0, 0, 0, \frac{1}{4}, 1, 1, 1\}$  В-сплайн 2-степени будет иметь вид:

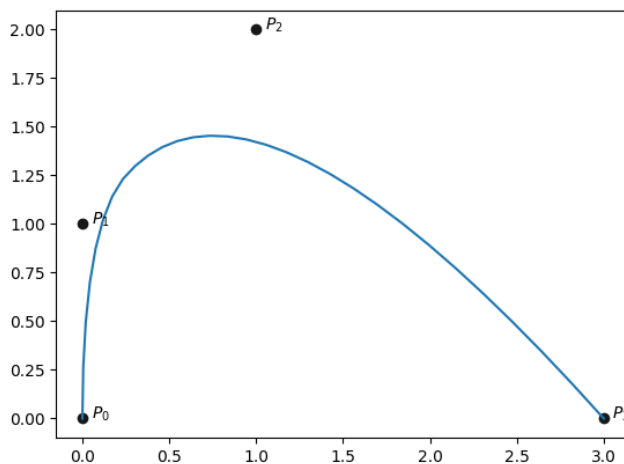


Рисунок 1.6 – Пример В-сплайн кривой

### 1.4.2 В-сплайн поверхности

В-сплайн поверхность задается с помощью контрольных точек и двух векторов узлов. Ее точки можно найти с помощью формулы:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} \quad (1.18)$$

Например, для набора контрольных точек

$$\begin{aligned} \mathbf{P}_{0,0} &= (0, 0, 0), & \mathbf{P}_{0,1} &= (1, 0, 0), & \mathbf{P}_{0,2} &= (2, 0, 0), \\ \mathbf{P}_{1,0} &= (0, 0.5, 1.3), & \mathbf{P}_{1,1} &= (1, 0.5, 1.2), & \mathbf{P}_{1,2} &= (2, 0.5, 1.3), \\ \mathbf{P}_{2,0} &= (0, 1, 0), & \mathbf{P}_{2,1} &= (1, 1, 0), & \mathbf{P}_{2,2} &= (2, 1, 0), \\ \mathbf{P}_{3,0} &= (0, 1.5, 1.3), & \mathbf{P}_{3,1} &= (1, 1.5, 1.2), & \mathbf{P}_{3,2} &= (2, 1.5, 1.3), \\ \mathbf{P}_{4,0} &= (0, 2, 0), & \mathbf{P}_{4,1} &= (1, 2, 0), & \mathbf{P}_{4,2} &= (2, 2, 0), \end{aligned}$$

векторов узлов  $U = \{0, 0, 0, 1/2, 1/2, 1, 1, 1\}$ ,  $V = \{0, 0, 0, 1, 1, 1\}$  и  $p = q = 2$  получим следующую В-сплайн поверхность:

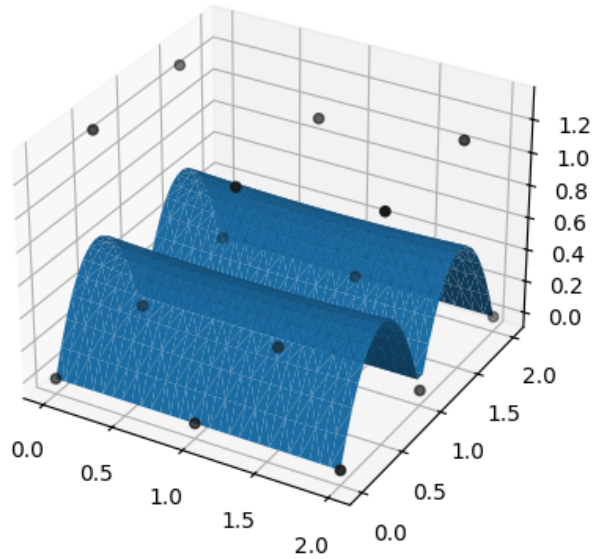


Рисунок 1.7 – Пример В-сплайн поверхности