Bézier Geometry

Chad B. Hovey[1]

Sandia Injury Biomechanics Laboratory
Sandia National Laboratories[2]

2020-12-29

[1]Author e-mail: chovey@sandia.gov

[2]The Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Contents

Chapter 1

Introduction

We briefly discuss polynomials for three main reasons: (1) To be explicit regarding the terminology of **power** versus **order** and be clear on the conventions and notations here, (2) to lay the framework for more sophisticated interpolations that will rely on polynomials in some form, and (3) to the note shortcomings of regular polynomials as a basis and thus motivate alternative approaches.

First, with regard to *power* versus *order*, [Cottrell et al., 2009] noted the following:

> "There is a terminology conflict between the geometry and analysis communities. Geometers will say a cubic polynomial has degree 3 and order 4. In geometry, order equals degree plus one. Analysts will say a cubic polynomial is order three, and use the term order and degree synonymously. This is the

convention we [Cottrell, Hughes, Bazilevs] adhere to." Page 18, Note 3.

Unlike [Cottrell et al., 2009], we adopt the geometry community convention in this document, thus we distinguish between *order* and *degree*. We have elected this approach primarily for consistency of this document with historical writings of the geometry community.

Next, there is some basic notation used with polynomials that we need to make clear and concrete. Let $f^p(x)$ be the function of degree p (equivalently, order $p+1$) that is a sum of variable x raised to some increasing non-negative integer power, starting from zero, and multiplied by a constant coefficient a, such that

$$f^p(x) \stackrel{\Delta}{=} a_0 x^0 + a_1 x^1 + a_2 x^2 + a_3 x^3 + \cdots + a_p x^p = \sum_{i=0}^{p} a_i x^i. \tag{1.1}$$

The function $f^p(x)$ is a **polynomial** of **degree** p, the highest non-zero **power**[1] used in the summation. Table 1.1 lists the first five polynomials by order, degree, name, and function. Hereinafter, p will be considered the index of the power of a polynomial. The p-index, a subset of non-negative integers, will start from zero and consecutively increase by one.

One note regarding conventions of indices: We use the first item of a series `s` to be `s[0]`, the second item to be `s[1]`, and so forth. We prefer the zero-based indices because it is consistent not only with our work in Python, which is zero-based, but also with concepts

[1] We suggest the mnemonic of "p" as the index used for the "power" in a "polynomial."

Table 1.1: First five orders of a polynomial function.

order	**degree**	**name**	**function**
$\mathcal{O}(1)$	$p = 0$	constant	$f^0(x) = a_0$
$\mathcal{O}(2)$	$p = 1$	linear	$f^1(x) = a_0 + a_1 x$
$\mathcal{O}(3)$	$p = 2$	quadratic	$f^2(x) = a_0 + a_1 x + a_2 x^2$
$\mathcal{O}(4)$	$p = 3$	cubic	$f^3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$
$\mathcal{O}(5)$	$p = 4$	quartic	$f^4(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$

from initial boundary value problems (IBVPs), which denote initial conditions, the first values, as occurring at t_0.

Finally, we state as *prima facie* a well-known shortcoming of polynomials is their propensity to overfit the data. This characteristic arises because a polynomial of degree p has $p-1$ changes of direction, from $-f^p(x)$ to $f^p(x)$, or vice versa. No further discussion on overfitting is given here, though explications of this topic are widely available.

Chapter 2

Bézier Curves

Several of the ideas in this section are due to the excellent references of [Bartels et al., 1995], [Piegl and Tiller, 1997], [Rogers, 2000], [Shiach, 2015b], and [Shiach, 2015a]. A Bézier curve is a parametric curve defined by control points. A control point $\boldsymbol{P}_i$ will have two coordinates (x_i, y_i) in 2D and three coordinates (x_i, y_i, z_i) in 3D. The parameter is typically denoted t. The bounds for t are $0 \leq t \leq 1$ unless otherwise indicated.

Here, t does not denote time. Rather, t can be thought of as a "pseudo-time," wherein the parameterization flows from beginning to end of the t bounds. Also, the bounds of t will later be shown to be arbitrary. For now, however, it is much more convenient to state the bounds as between zero and one.

A Bézier curve of degree p requires $p + 1$ control points. For example, a Bézier curve that is a line (degree $p = 1$) requires two control points. Affine transformations may be used to modify the control points, *e.g.*, to scale, reflect, rotate, or translate (offset) the control points.

2.1 Bézier Line

Let $\mathcal{P}$ be the set of two points $\boldsymbol{P}_0 = (x_0, y_0, z_0) \in \mathbb{R}^3$ and $\boldsymbol{P}_1 = (x_1, y_1, z_1) \in \mathbb{R}^3$. Let the parameter t be a member of $\mathbb{T} = [0, 1] \subset \mathbb{R}$. Then, let $\mathbb{C}(t; \boldsymbol{P}_0, \boldsymbol{P}_1) : \mathcal{P} \times \mathbb{T} \mapsto \mathbb{R}^3$ be **parametric** equation for a line between two points $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$. This parameterization constructs the Bézier line, which is the parameterized linear interpolation $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$,

$$\mathbb{C}(t; \boldsymbol{P}_0, \boldsymbol{P}_1) \triangleq (1 - t)\, \boldsymbol{P}_0 + t\, \boldsymbol{P}_1, \tag{2.1}$$

or in explicit coordinate form,

$$\begin{Bmatrix} x(t) \\ y(t) \\ z(t) \end{Bmatrix} = (1 - t) \begin{Bmatrix} x_0 \\ y_0 \\ z_0 \end{Bmatrix} + t \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}. \tag{2.2}$$

2.2 Bézier Quadratic

Let $\boldsymbol{Q}(t; \boldsymbol{Q}_0, \boldsymbol{Q}_1)$ be the parametric equation for a quadratic between two points $\boldsymbol{Q}_0$ and $\boldsymbol{Q}_1$. Let the position of these two points themselves be parameterized by three points $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, and $\boldsymbol{P}_2$, such that

$$\boldsymbol{Q}_0(t; \boldsymbol{P}_0, \boldsymbol{P}_1) = (1-t)\ \boldsymbol{P}_0 + t\ \boldsymbol{P}_1, \tag{2.3}$$

$$\boldsymbol{Q}_1(t; \boldsymbol{P}_1, \boldsymbol{P}_2) = (1-t)\ \boldsymbol{P}_1 + t\ \boldsymbol{P}_2. \tag{2.4}$$

Thus, the position of $\boldsymbol{Q}_0(t)$ is on the line between points $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$ and parameterized by t. The position of $\boldsymbol{Q}_1(t)$ is on the line between points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ and likewise parameterized by t. At $t = 0$, $\boldsymbol{Q}_0(t)$ resides at $\boldsymbol{P}_0$, $\boldsymbol{Q}_1(t)$ resides at $\boldsymbol{P}_1$. At $t = 1$, $\boldsymbol{Q}_0(t)$ resides at $\boldsymbol{P}_1$, $\boldsymbol{Q}_1(t)$ resides at $\boldsymbol{P}_2$. Finally, let any position along the quadratic Bézier curve $\boldsymbol{Q}(t; \boldsymbol{Q}_0, \boldsymbol{Q}_1)$ be defined as

$$\boldsymbol{Q}(t; \boldsymbol{Q}_0(t), \boldsymbol{Q}_1(t)) \triangleq (1-t)\ \boldsymbol{Q}_0(t) + t\ \boldsymbol{Q}_1(t). \tag{2.5}$$

This curve can be recast in terms of the three points $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, and $\boldsymbol{P}_2$ by substituting (2.3) and (2.4),

$$\boldsymbol{Q}(t; \boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2) = (1-t)^2\ \boldsymbol{P}_0 + 2t(1-t)\ \boldsymbol{P}_1 + t^2\ \boldsymbol{P}_2. \tag{2.6}$$

Figure 2.1(a) illustrates an example quadratic Bézier curve, with the three control points $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, and $\boldsymbol{P}_2$ indicated.[1,2] We will designate the number of control points to be $(n+1)$. Thus, the number of control points in the current example (Figure 2.1) is $(n+1) = 3 \implies n = 2$. Each control point can be identified in sequence as $\boldsymbol{P}_i$, with $i = 0, 1, \ldots n$. The maximum degree Bézier that can be constructed is two (degree p requires $p+1$ control points). Thus we see:

> Given a series of $(n+1)$ control points,
> we can construct a Bézier curve of degree $p = n$.

[1]We do not call $\boldsymbol{Q}_0$ and $\boldsymbol{Q}_1$ control points, since they are dependent on $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, and $\boldsymbol{P}_1$ through parameter t in (2.3) and (2.4).

[2]Figure 2.1(b) shows the same curve as in Figure 2.1(a), just with a recursive notation, discussed in Section 2.3.

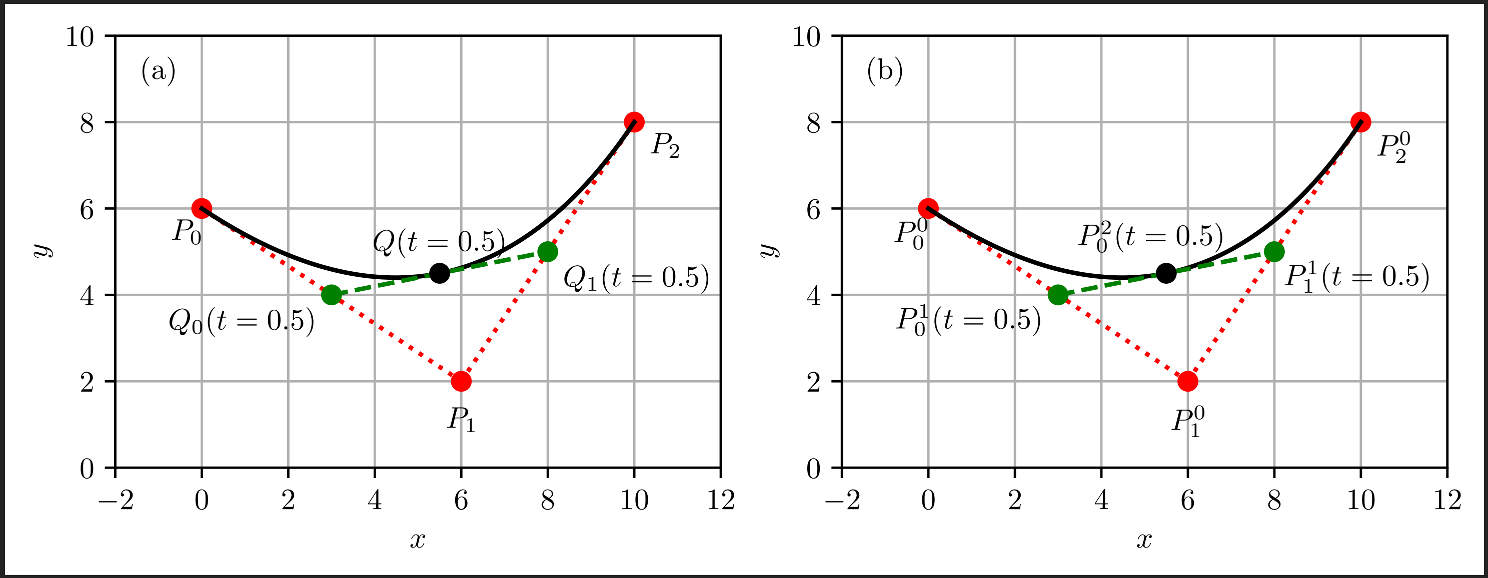

Figure 2.1: A Bézier quadratic curve illustrated at $t = 0.5$ with (a) original notation and (b) the de Casteljau's algorithm notation. Reference: de_casteljau.py.

Example 1.
Figure 2.2 illustrates the same quadratic Bézier curve shown in Figure 2.1, generated at six discrete points in the interval $t \in [0, 1]$. $\square$

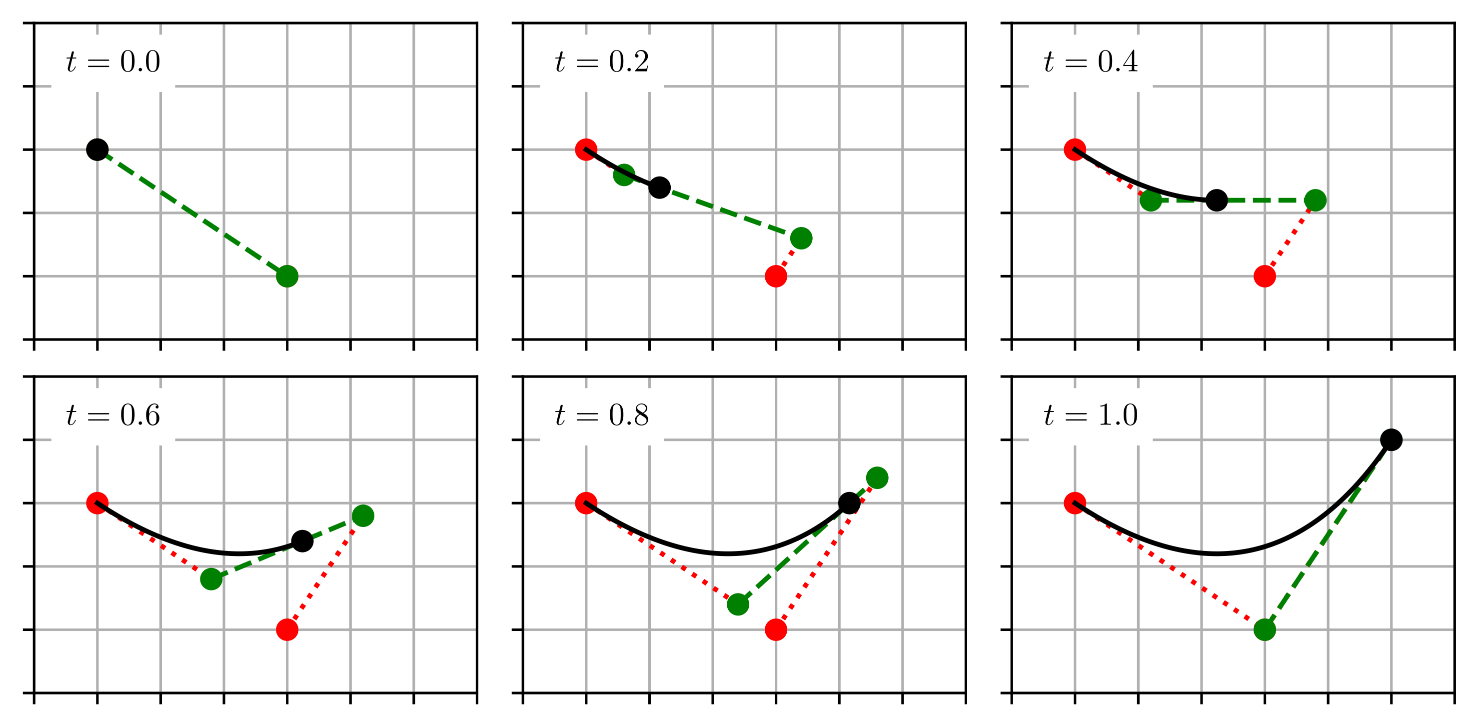

Figure 2.2: The Bézier quadratic curve discussed in Figure 2.1, illustrated in sequence with t starting at 0 and ending at 1. Reference: de_casteljau.py.

With these Figures 2.1–2.2 in mind, a few additional observations[3] can be made:

- The control points sequence $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ is a coarse and discrete *approximation* the continuous quadratic Bézier curve $\boldsymbol{Q}(t)$.
- The end points, $\boldsymbol{P}_0$ and $\boldsymbol{P}_2$, are interpolated, but the interior point $\boldsymbol{P}_1$ is not interpolated.[4]
- The tangent of the curve $\boldsymbol{Q}(t)$ at $t = 0$ is parallel to the line $\boldsymbol{P}_1 - \boldsymbol{P}_0$. The tangent of the curve $\boldsymbol{Q}(t)$ at $t = 1$ is parallel to the line $\boldsymbol{P}_2 - \boldsymbol{P}_0$.
- The entire curve $\boldsymbol{Q}(t)$ is contained in the triangle formed with vertices $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$.

[3] These items are stated as true but not proven here. Consult the references in the Bibliography section for proofs.

[4] This will generalize to all interior points for higher-degree Bézier curves.

2.3 de Casteljau's Algorithm

The foregoing development can be generalized, and is known as **de Casteljau's algorithm**. Let $\boldsymbol{P}_i^d$ be control points where

- i denotes the **control point index**, $i = 0, 1, \ldots n$, (total of $n+1$ control points), and
- d denote the **degree** of the curve, $d = 0, 1, \ldots p$.

Thus $\boldsymbol{P}_i^0$ denote the base level control points. In the preceding section,

$$\begin{aligned}
\boldsymbol{P}_0 &\mapsto \boldsymbol{P}_0^0, \quad d = 0, \text{ thus } \boldsymbol{P}_i^0 \text{ is a point, \textit{importantly} \textbf{not} parameterized by } t && (2.7)\\
\boldsymbol{P}_1 &\mapsto \boldsymbol{P}_1^0, && (2.8)\\
\boldsymbol{P}_2 &\mapsto \boldsymbol{P}_2^0, && (2.9)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{Q}_0(t) &\mapsto \boldsymbol{P}_0^1(t), \quad d = 1, \text{ thus } \boldsymbol{P}_i^1(t) \text{ is a line parameterized by } t, && (2.10)\\
\boldsymbol{Q}_1(t) &\mapsto \boldsymbol{P}_1^1(t), \quad \text{and} && (2.11)
\end{aligned}$$

$$\boldsymbol{Q}(t) \mapsto \boldsymbol{P}_0^2(t), \quad d = 2, \text{ thus } \boldsymbol{P}_i^2(t) \text{ is a quadratic parameterized by } t. \tag{2.12}$$

Then, **de Casteljau's algorithm** states

$$\boldsymbol{P}_i^d(t) = (1-t)\,\boldsymbol{P}_i^{d-1}(t) + t\,\boldsymbol{P}_{i+1}^{d-1}(t). \tag{2.13}$$

The Bézier quadratic written in (2.6) would then be rewritten as

$$\boldsymbol{P}_0^2(t) = (1-t)\ \boldsymbol{P}_0^1(t) + t\ \boldsymbol{P}_1^1(t), \tag{2.14}$$

$$= (1-t)\left[(1-t)\ \boldsymbol{P}_0^0 + t\ \boldsymbol{P}_1^0\right] + t\left[(1-t)\ \boldsymbol{P}_1^0 + t\ \boldsymbol{P}_2^0\right], \tag{2.15}$$

$$= (1-t)^2\ \boldsymbol{P}_0^0 + 2t(1-t)\ \boldsymbol{P}_1^0 + t^2\ \boldsymbol{P}_2^0. \tag{2.16}$$

Figure 2.1 illustrates for the Bézier quadratic development, (a) the original notation and (b) the de Casteljau's algorithm notation.

Although de Casteljau's algorithm is well-suited for computer implementation, it is instructive to write the quadratic Bézier $\boldsymbol{P}_0^2(t)$ in matrix form for **any sequence** of **three control points**[5] $\boldsymbol{P}_i$, $\boldsymbol{P}_{i+1}$, and $\boldsymbol{P}_{i+2}$. With

$$\boldsymbol{P}_0^0 \mapsto \boldsymbol{P}_i, \tag{2.17}$$

$$\boldsymbol{P}_1^0 \mapsto \boldsymbol{P}_{i+1}, \tag{2.18}$$

$$\boldsymbol{P}_2^0 \mapsto \boldsymbol{P}_{i+2}, \text{ and} \tag{2.19}$$

$$\boldsymbol{P}_0^2(t) \mapsto \mathbb{C}^2(t), \text{ to denote a curve } \mathbb{C}^p \text{ of degree } p = 2, \tag{2.20}$$

[5]This is somewhat of a return map, undoing the notation adopted to explain de Casteljau's algorithm with $i = 0$.

the quadratic Bézier $\mathbb{C}^2(t)$ is written in matrix form as

$$\mathbb{C}^2(t) = \begin{bmatrix} \boldsymbol{P}_i & \boldsymbol{P}_{i+1} & \boldsymbol{P}_{i+2} \end{bmatrix} \cdot \boldsymbol{f}(t^2, t, 1), \tag{2.21}$$

$$\begin{Bmatrix} x(t) \\ y(t) \\ z(t) \end{Bmatrix} = \underbrace{\begin{bmatrix} x_i & x_{i+1} & x_{i+2} \\ y_i & y_{i+1} & y_{i+2} \\ z_i & z_{i+1} & z_{i+2} \end{bmatrix}}_{\substack{\text{control points} \\ \text{curve dependent}}} \underbrace{\begin{bmatrix} 1 & \text{-2} & 1 \\ & 2 & 0 \\ \text{sym.} & & 0 \end{bmatrix} \begin{Bmatrix} t^2 \\ t \\ 1 \end{Bmatrix}}_{\text{curve independent}}. \tag{2.22}$$

2.4 Bézier Cubic

The cubic Bézier curve $\boldsymbol{P}_0^3(t)$, given four control points $\boldsymbol{P}_0^0$, $\boldsymbol{P}_1^0$, $\boldsymbol{P}_2^0$, and $\boldsymbol{P}_3^0$, is given by

$$\boldsymbol{P}_0^3(t) = (1-t)^3\ \boldsymbol{P}_0^0 + 3t(1-t)^2\ \boldsymbol{P}_1^0 + 3t^2(1-t)\ \boldsymbol{P}_2^0 + t^3\ \boldsymbol{P}_3^0. \tag{2.23}$$

Like the quadratic, the cubic Bézier $\boldsymbol{P}_0^3(t)$ may be writtin in matrix form for **any sequence** of **four control points** $\boldsymbol{P}_i$, $\boldsymbol{P}_{i+1}$, $\boldsymbol{P}_{i+2}$ and $\boldsymbol{P}_{i+3}$. With

$$\boldsymbol{P}_0^0 \mapsto \boldsymbol{P}_i, \tag{2.24}$$

$$\boldsymbol{P}_1^0 \mapsto \boldsymbol{P}_{i+1}, \tag{2.25}$$

$$\boldsymbol{P}_2^0 \mapsto \boldsymbol{P}_{i+2}, \tag{2.26}$$

$$\boldsymbol{P}_3^0 \mapsto \boldsymbol{P}_{i+3}, \text{ and} \tag{2.27}$$

$$\boldsymbol{P}_0^3(t) \mapsto \mathbb{C}^3(t), \text{ to denote a curve } \mathbb{C}^p \text{ of degree } p = 3, \tag{2.28}$$

the cubic Bézier $\mathbb{C}^3(t)$ is written in matrix form as

$$\mathbb{C}^3(t) = \begin{bmatrix} \boldsymbol{P}_i & \boldsymbol{P}_{i+1} & \boldsymbol{P}_{i+2} & \boldsymbol{P}_{i+3} \end{bmatrix} \cdot \boldsymbol{f}(t^2, t, 1), \tag{2.29}$$

$$\begin{Bmatrix} x(t) \\ y(t) \\ z(t) \end{Bmatrix} = \underbrace{\begin{bmatrix} x_i & x_{i+1} & x_{i+2} & x_{i+3} \\ y_i & y_{i+1} & y_{i+2} & y_{i+3} \\ z_i & z_{i+1} & z_{i+2} & z_{i+3} \end{bmatrix}}_{\substack{\text{control points} \\ \text{curve dependent}}} \underbrace{\begin{bmatrix} -1 & 3 & -3 & 1 \\ & -6 & 3 & 0 \\ & & 0 & 0 \\ \text{sym.} & & & 0 \end{bmatrix} \begin{Bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{Bmatrix}}_{\text{curve independent}}. \tag{2.30}$$

2.5 Bernstein Polynomials

The general form of a degree p Bézier curve $\mathbb{C}^p(t)$ defined by $p+1$ control points $\boldsymbol{P}_i$, $i = 0, 1, \ldots, p$, is given by

$$\mathbb{C}^p(t) \triangleq \sum_{i=0}^{p} B_i^p(t)\boldsymbol{P}_i, \tag{2.31}$$

where $B_i^p(t)$ is a **Bernstein polynomial**, defined as

$$B_i^p(t) = \begin{pmatrix} p \\ i \end{pmatrix} t^i(1-t)^{p-i}, \quad \text{and where} \quad \begin{pmatrix} p \\ i \end{pmatrix} = \frac{p!}{i!\,(p-i)!} \tag{2.32}$$

is the **binomial coefficient**. The binomial coefficients are readily attained from **Pascal's triangle**, written up to $p = 4$ in Table 2.1.

Using (2.31), the linear ($p = 1$), quadratic ($p = 2$), cubic ($p = 3$), and quartic ($p = 4$) Bézier curves can be written, respectively, in explicit form as

$$\mathbb{C}^1(t) = B_0^1(t)\boldsymbol{P}_0 + B_1^1(t)\boldsymbol{P}_1, \tag{2.33}$$

$$\mathbb{C}^2(t) = B_0^2(t)\boldsymbol{P}_0 + B_1^2(t)\boldsymbol{P}_1 + B_2^2(t)\boldsymbol{P}_2, \tag{2.34}$$

$$\mathbb{C}^3(t) = B_0^3(t)\boldsymbol{P}_0 + B_1^3(t)\boldsymbol{P}_1 + B_2^3(t)\boldsymbol{P}_2 + B_3^3(t)\boldsymbol{P}_3, \tag{2.35}$$

$$\mathbb{C}^4(t) = B_0^4(t)\boldsymbol{P}_0 + B_1^4(t)\boldsymbol{P}_1 + B_2^4(t)\boldsymbol{P}_2 + B_3^4(t)\boldsymbol{P}_3 + B_4^4(t)\boldsymbol{P}_4. \tag{2.36}$$

Table 2.1: First five degrees of binomial coefficients from Pascal's triangle.

degree										**binomial coefficient**
$p = 0$					1					
$p = 1$				1		1				
$p = 2$			1		2		1			
$p = 3$		1		3		3		1		
$p = 4$	1		4		6		4		1	

Observations[6] include (see Figure 2.3 for a visual illustration of these properties):

- The Bernstein polynomials always sum to one,

$$\sum_{i=0}^{p} B_i^p(t) = 1.0. \tag{2.37}$$

  This concept is called **partition of unity**.

- The first basis is unity $B_0^p(0) = 1$ at the start of the interval $t = 0$, when all other bases are zero. Similarly, the last basis is unity $B_p^p(1) = 1$ and the end of the interval $t = 1$, when all other bases go to zero.

- The polynomials are **non-negative**,

$$B_i^p(t) \geq 0 \text{ for all } {}_i^p \text{ with } t \in [0, 1]. \tag{2.38}$$

- Each polynomial $B_i^p(t)$ has a single maximum in the parameter space $t \in [0, 1]$ at $t = i/p$.

- All polynomials are symmetric in t about $t = 1/2$.

[6] See Bibliography for references containing proofs.

Example 2.

The Bernstein polynomials for a linear $(p = 1)$ Bézier curve are

$$B_0^1(t) = \binom{1}{0} t^0 (1-t)^{1-0} = (1-t), \tag{2.39}$$

$$B_1^1(t) = \binom{1}{1} t^1 (1-t)^{1-1} = t, \tag{2.40}$$

which match the coefficients in (2.1). These polynomials are shown in Figure 2.3(a). □

Example 3.
The Bernstein polynomials for a quadratic $(p = 2)$ Bézier curve are

$$B_0^2(t) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} t^0(1-t)^{2-0} = (1-t)^2, \tag{2.41}$$

$$B_1^2(t) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} t^1(1-t)^{2-1} = 2t(1-t), \tag{2.42}$$

$$B_2^2(t) = \begin{pmatrix} 2 \\ 2 \end{pmatrix} t^2(1-t)^{2-2} = t^2, \tag{2.43}$$

which match the coefficients in (2.16). These polynomials are shown in Figure 2.3(b). □

Example 4.
The Bernstein polynomials for a cubic ($p = 3$) Bézier curve are

$$B_0^3(t) = \binom{3}{0} t^0 (1-t)^{3-0} = (1-t)^3, \tag{2.44}$$

$$B_1^3(t) = \binom{3}{1} t^1 (1-t)^{3-1} = 3t(1-t)^2, \tag{2.45}$$

$$B_2^3(t) = \binom{3}{2} t^2 (1-t)^{3-2} = 3t^2(1-t), \tag{2.46}$$

$$B_3^3(t) = \binom{3}{3} t^3 (1-t)^{3-3} = t^3, \tag{2.47}$$

which match the coefficients in (2.23). These polynomials are shown in Figure 2.3(c). □

Example 5.
The Bernstein polynomials for a quartic ($p = 4$) Bézier curve are

$$B_0^4(t) = \binom{4}{0} t^0(1-t)^{4-0} = (1-t)^4, \tag{2.48}$$

$$B_1^4(t) = \binom{4}{1} t^1(1-t)^{4-1} = 4t(1-t)^3, \tag{2.49}$$

$$B_2^4(t) = \binom{4}{2} t^2(1-t)^{4-2} = 6t^2(1-t)^2, \tag{2.50}$$

$$B_3^4(t) = \binom{4}{3} t^3(1-t)^{4-3} = 4t^3(1-t), \tag{2.51}$$

$$B_4^4(t) = \binom{4}{4} t^4(1-t)^{4-4} = t^4. \tag{2.52}$$

These polynomials are shown in Figure 2.3(d). Figure 2.4 shows the Bernstein polynomials for $p = 5, 6, 7, 8$. $\square$

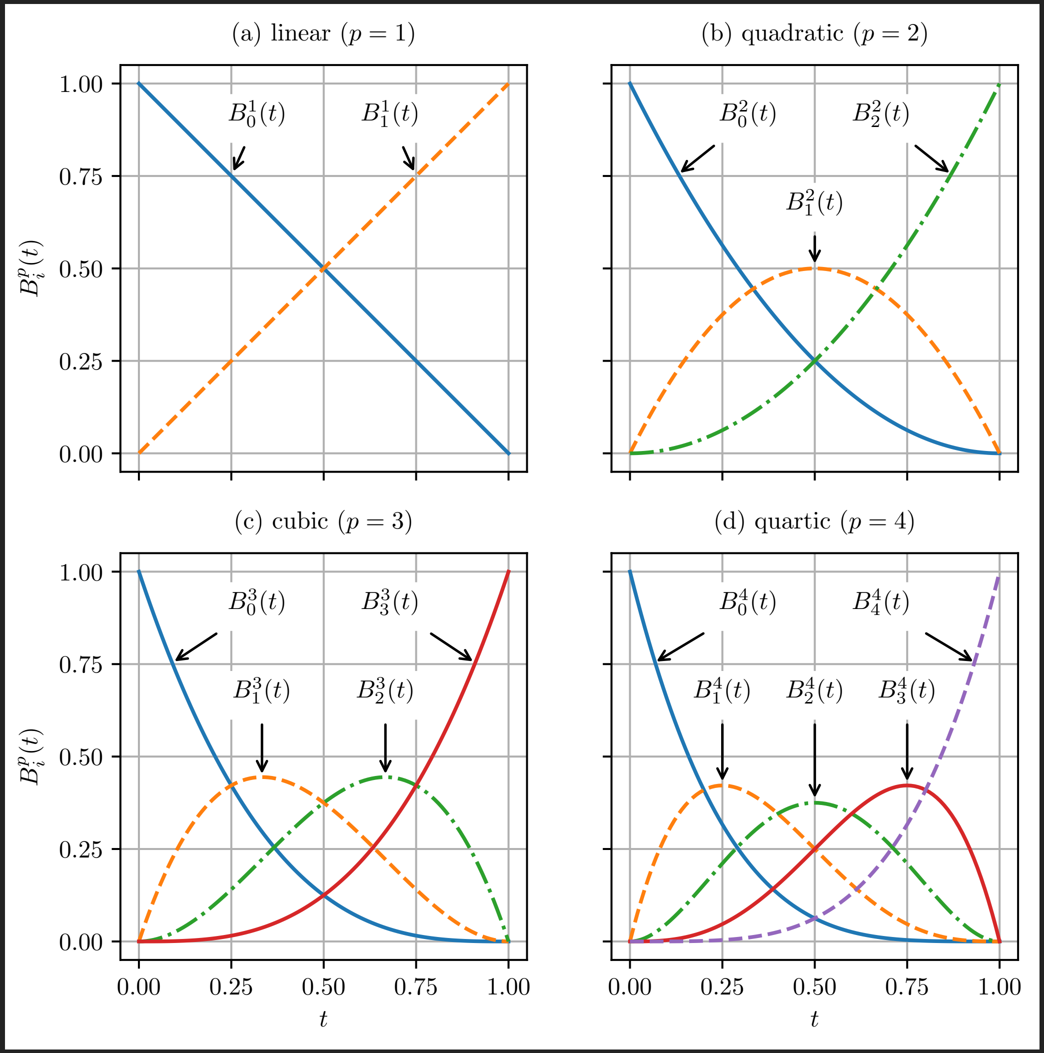

Figure 2.3: Bernstein polynomials for Bézier (a) linear, (b) quadratic, (c) cubic, and (d) quartic curves. Reference: bernstein.py, bernstein_polynomial.py.

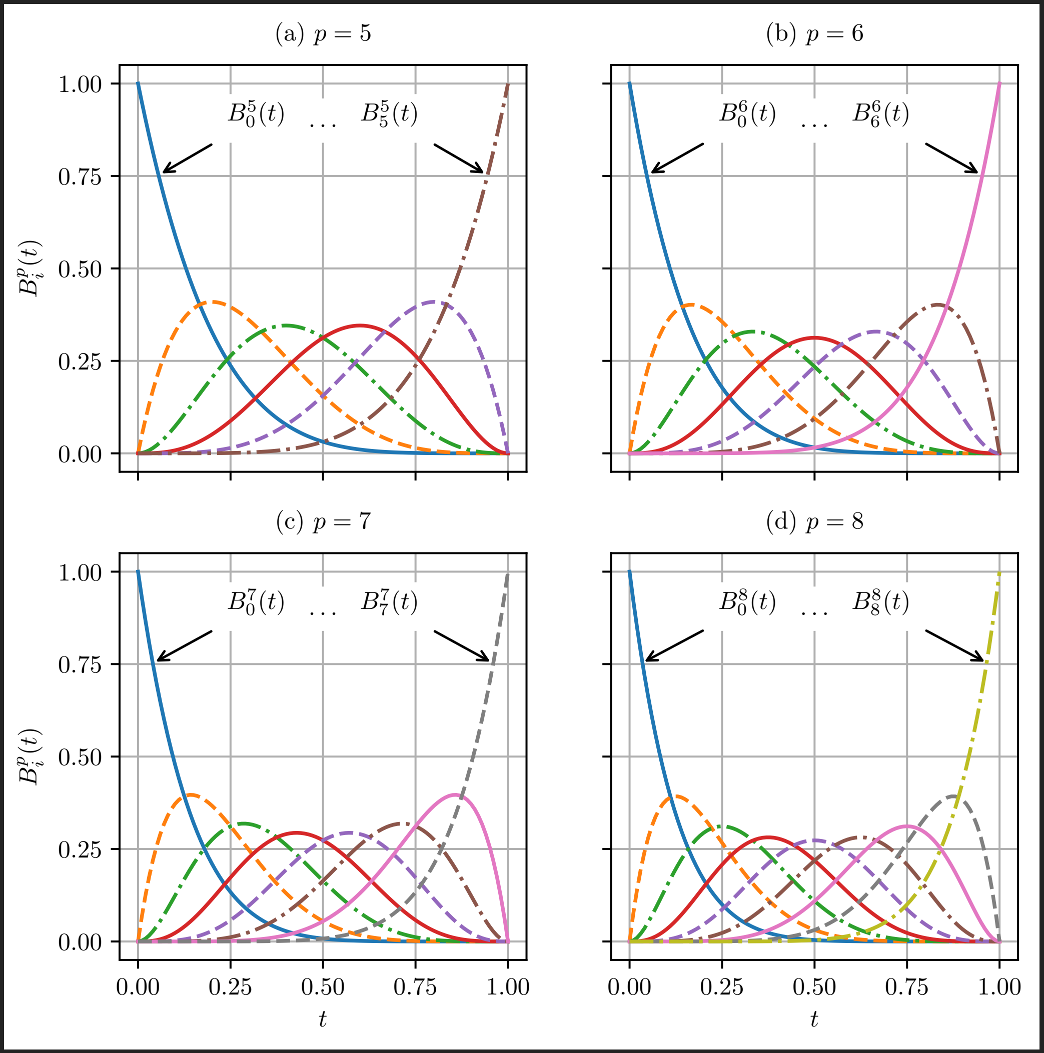

Figure 2.4: Bernstein polynomials for Bézier (a) $p = 5$, (b) $p = 6$, (c) $p = 7$, and (d) $p = 8$ curves. Reference: bernstein_extended.py.

Example 6.

Using the Bernstein polynomials for cubic ($p = 3$) Bézier curve, illustrate the curves generated by the following control points, labeled $0, 1, 2, 3$, as shown in Figure 2.5. $\square$

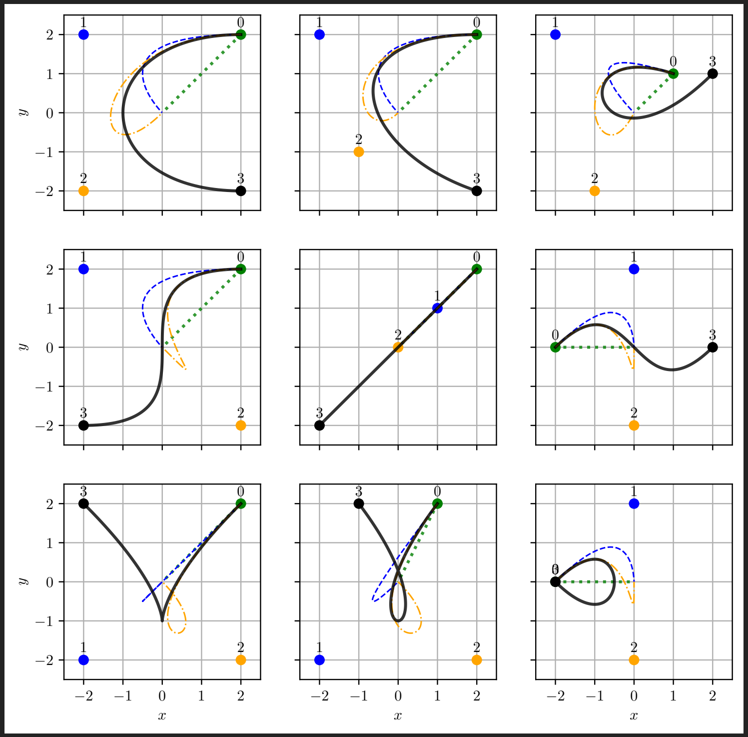

Figure 2.5: Cubic ($p = 3$) Bézier curves constructed from Bernstein polynomials and control points labeled $0, 1, 2, 3$. Dashed, dotted, and dashed-dotted lines show the incremental construction of the curve as each control point is added. Reference: bernstein_sum.py.

Example 7.
The letters in "cubic" can be created from cubic ($p = 3$) Bernstein polynomials. The "u" shows the canonical form of the cubic Bézier, with the first and last points as the anchors, and the second and penultimate points as the tangents from their respective anchors. See Figure 2.6.
□

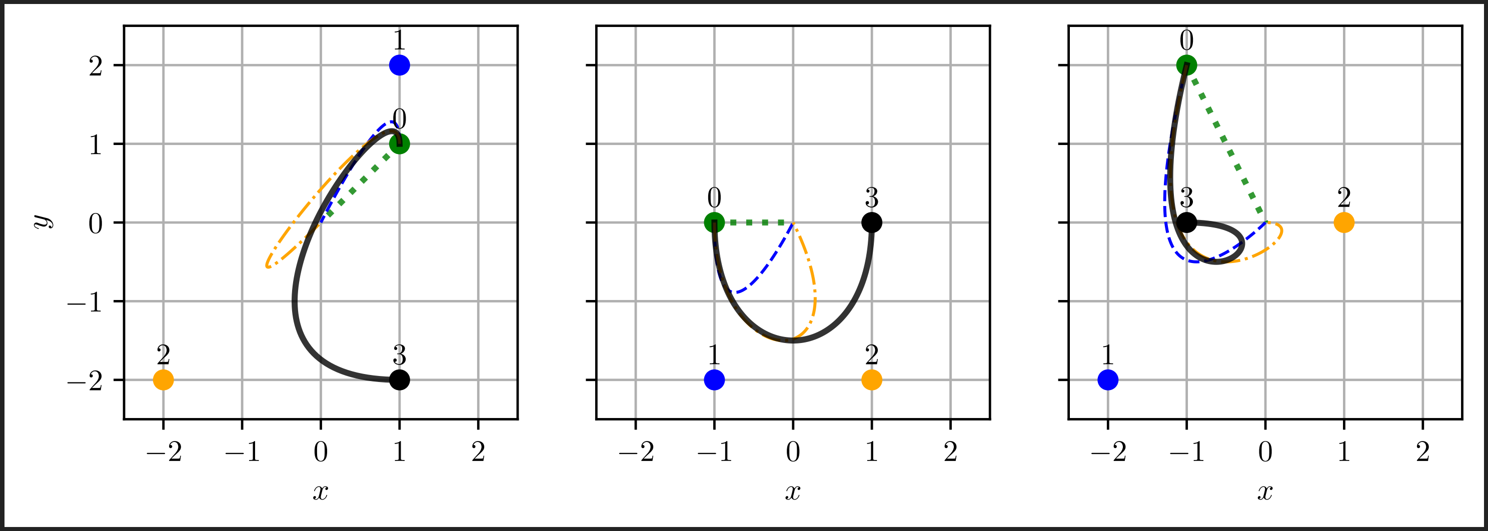

Figure 2.6: Spelling of the first three letters in the word "cubic" as a mnemonic for the shapes created with cubic Bézier curves. The "u" is the canonical shape. Capital "I" (created with co-linear control points) and lowercase "c" are created as shown in Figure 2.5. Reference: bernstein_sum_ext.py.

Example 8.
Modern vectorized fonts are created from Bézier curves. Figure 2.7 shows the letter "e", Georgia font family, made from eleven (11) Bézier cubic ($p = 3$) curves and twenty-seven (27) control points. □

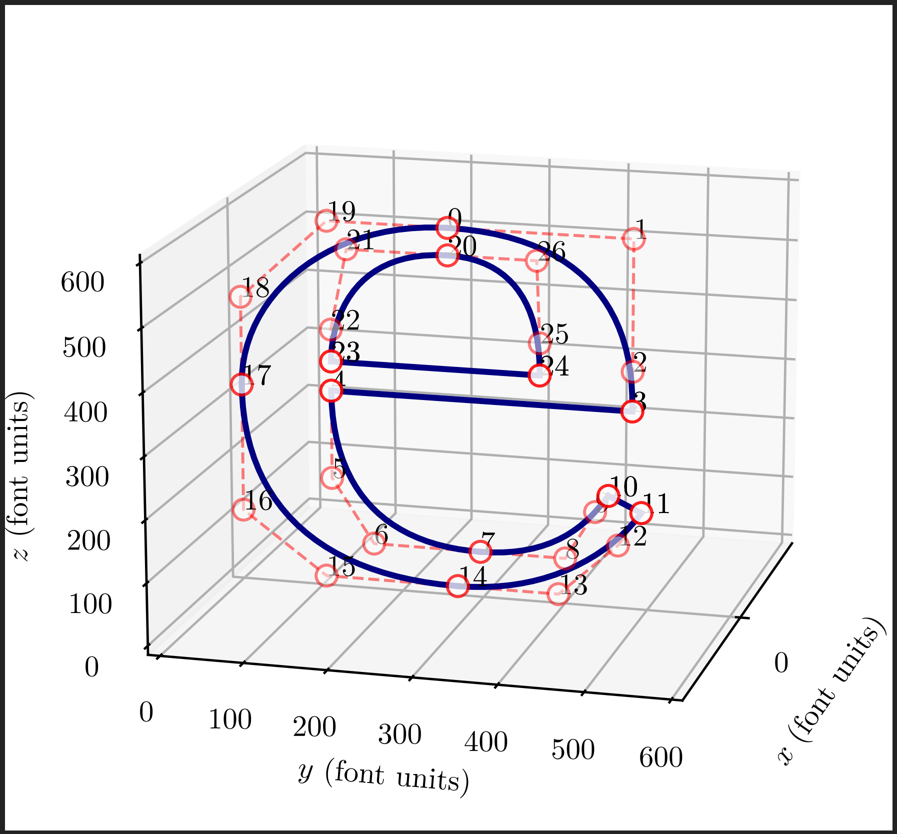

Figure 2.7: Letter "e" composed of Bézier curves. See `view_bezier.py` and `Georgia-e-config.json` on the GitHub SIBL repository.

Chapter 3

Bézier Surfaces

The Bézier curve, $\mathbb{C}$, was parameterized by t. For the Bézier surface, $\mathbb{S}$, we have two parameters, t and u,[1] with $t \in [0,1]$ and $u \in [0,1]$. The Bézier surface is an extension of the Bézier curve, defined in (2.31).

Let $\mathcal{P}$ be the set of **control points**, $\boldsymbol{P}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \in \mathbb{R}^3 \;\; \forall \;\; i \in 0 \ldots p, \; j \in 0 \ldots q$, be arranged in a non-decreasing sequence in two dimensions, referred to as the **control grid**,[2] $\mathcal{N}$. The control grid $\mathcal{N}$ is the *arrangement* of control points by control point index into a non-decreasing grid in (t,u) space:

[1] Bézier volumes, $\mathbb{V}$, will then have three parameters t, u, and v. See Chapter 4 for details.

[2] The term **control net** is sometimes used interchangeably with the term control grid.

	$u=0$	$0<u<1$		$u=1$
$t=0$	$\boldsymbol{P}_{0,0}$	$\boldsymbol{P}_{0,1}$	$\cdots$	$\boldsymbol{P}_{0,q}$
$0<t<1$	$\boldsymbol{P}_{1,0}$ $\vdots$	$\boldsymbol{P}_{1,1}$ $\vdots$	$\cdots$ $\ddots$	$\boldsymbol{P}_{1,q}$ $\vdots$
$t=1$	$\boldsymbol{P}_{p,0}$	$\boldsymbol{P}_{p,1}$	$\cdots$	$\boldsymbol{P}_{p,q}$

The general form of a Bézier surface $\mathbb{S}^{p,q}(t,u)$ of degree p and $p+1$ control points for the t parameter and of degree q and $q+1$ control points for the u parameter is defined as

$$\mathbb{S}^{p,q}(t,u) \triangleq \sum_{i=0}^{p} \sum_{j=0}^{q} B_i^p(t)\ B_j^q(u)\ \boldsymbol{P}_{i,j}. \tag{3.1}$$

The Bézier basis functions are defined as the outer product of two Bernstein polynomials,

$$B_{i,j}^{p,q}(t,u) \triangleq B_i^p(t) \otimes B_j^q(u). \tag{3.2}$$

While not necessary, it is often the case in practice that the number of control points for the t and u parameters are taken to be the same, *i.e.*, $(p+1)=(q+1)$. In this case, the foregoing definition reduces to

$$B_{i,j}^{p}(t,u) \triangleq B_i^p(t) \otimes B_j^p(u). \tag{3.3}$$

Three examples of basis functions are presented:
$B_{i,j}^1(t,u)$ bi-linear in Figure 3.1,
$B_{i,j}^2(t,u)$ bi-quadratic in Figure 3.5, and
$B_{i,j}^3(t,u)$ bi-cubic in Figure 3.14.

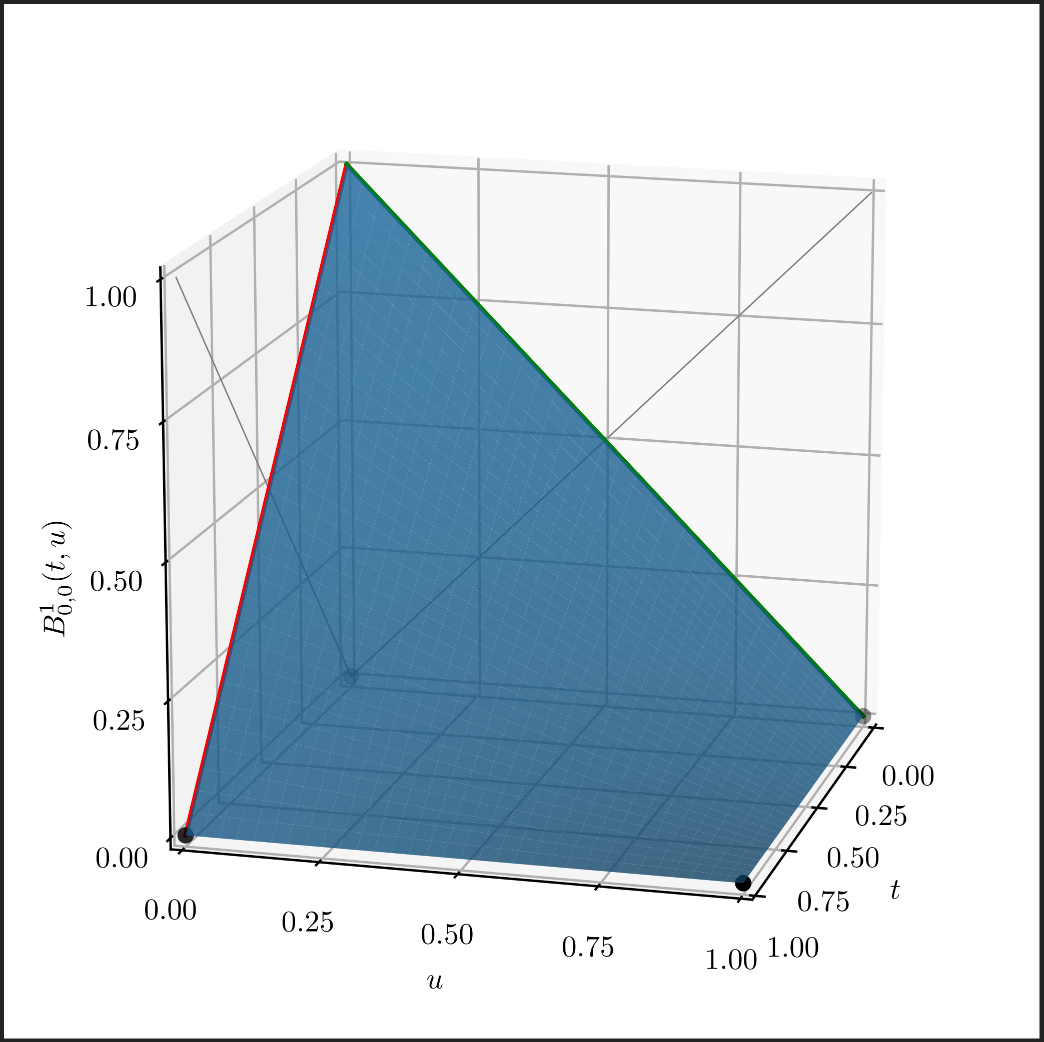

Figure 3.1: Bézier bi-linear basis functions. See `view_bernstein_surface.py` on the GitHub SIBL repository.

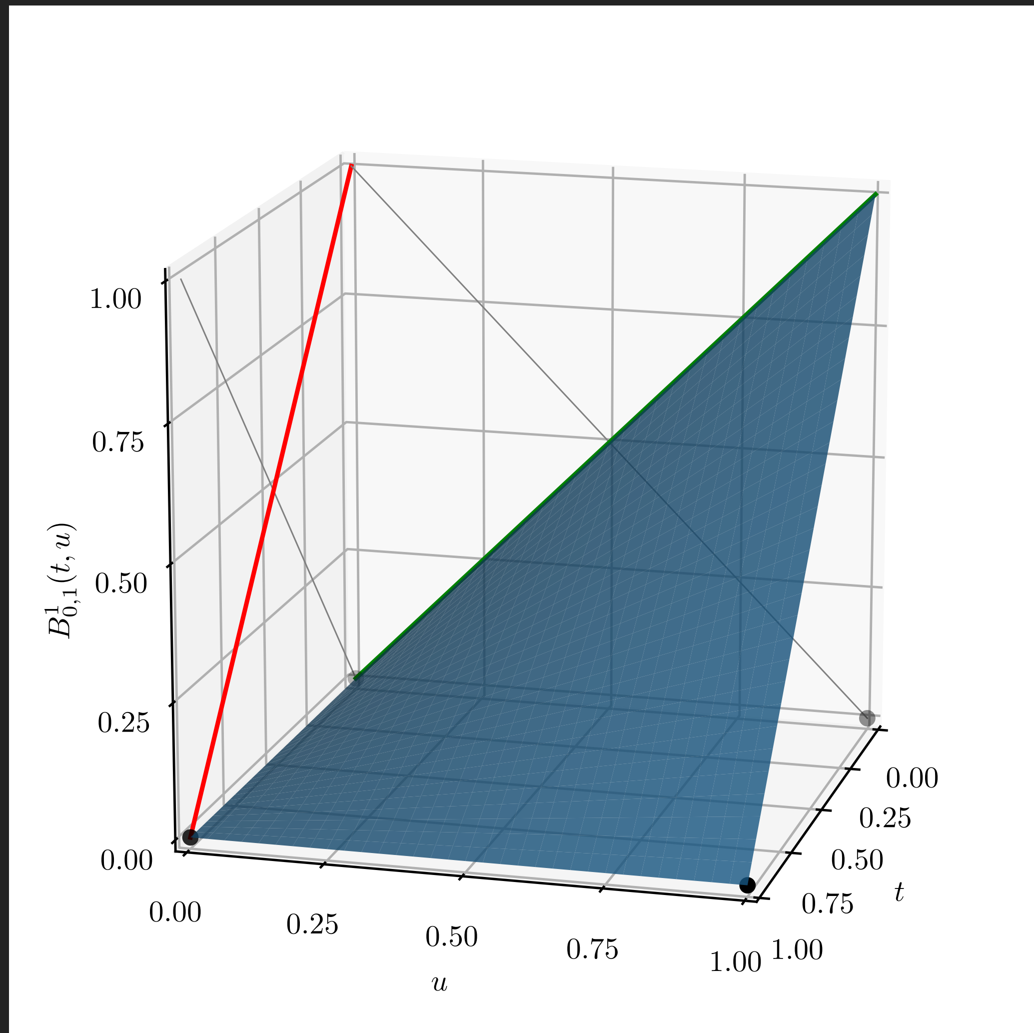

Figure 3.2: Continued from previous figure.

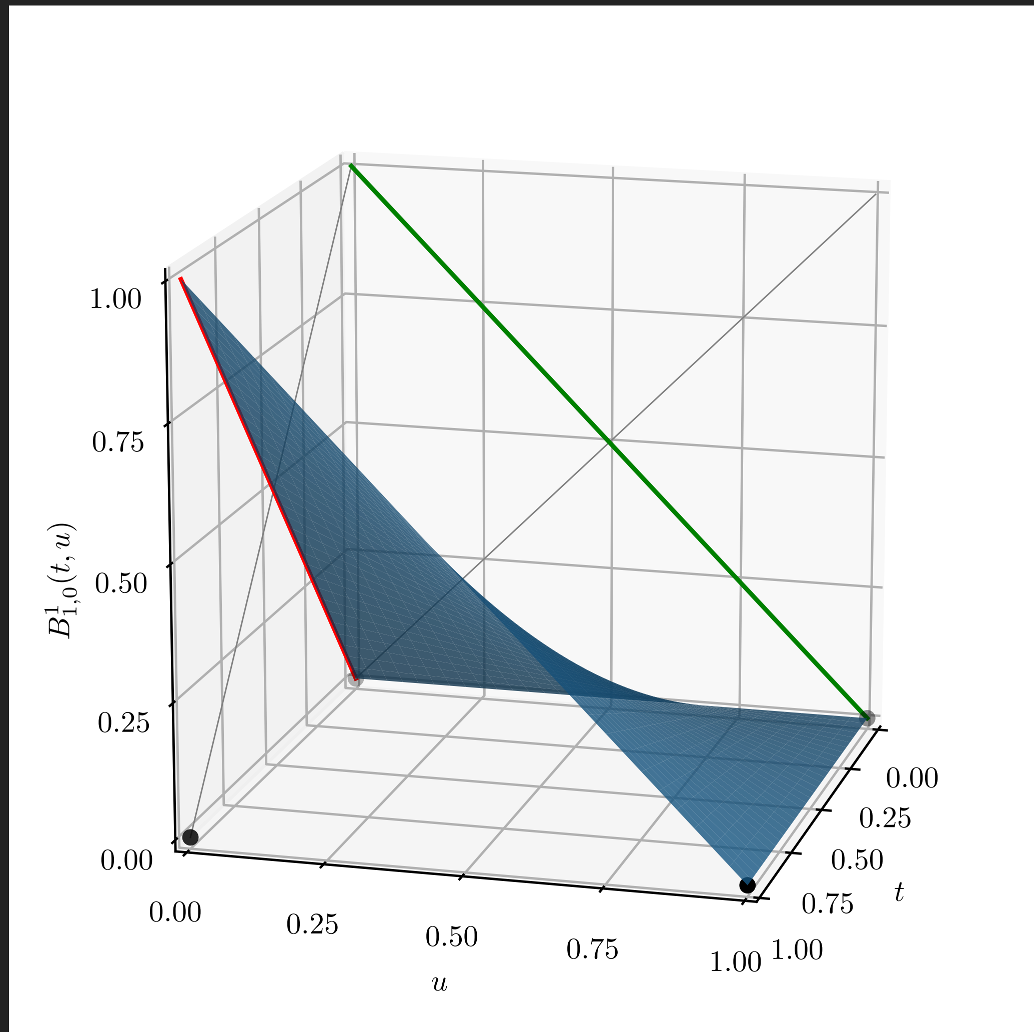

Figure 3.3: Continued from previous figure.

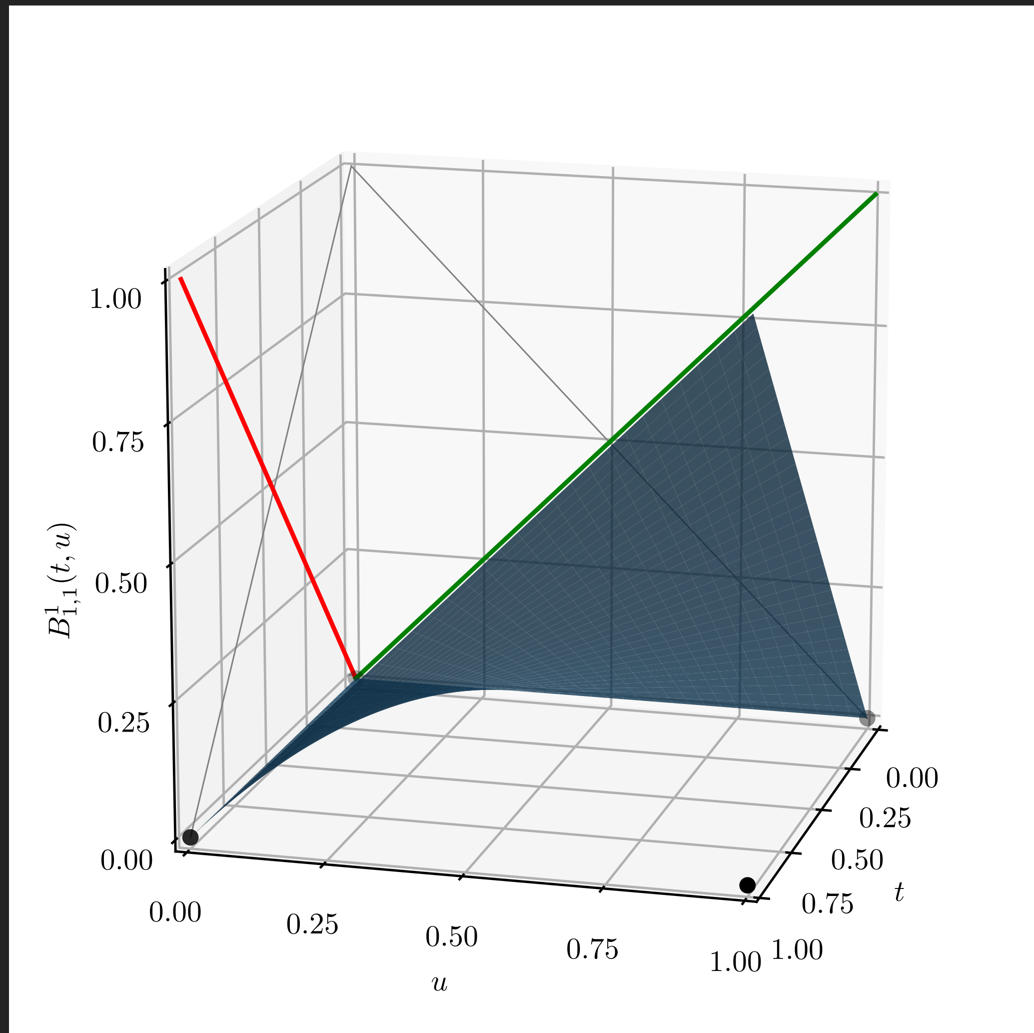

Figure 3.4: Continued from previous figure.

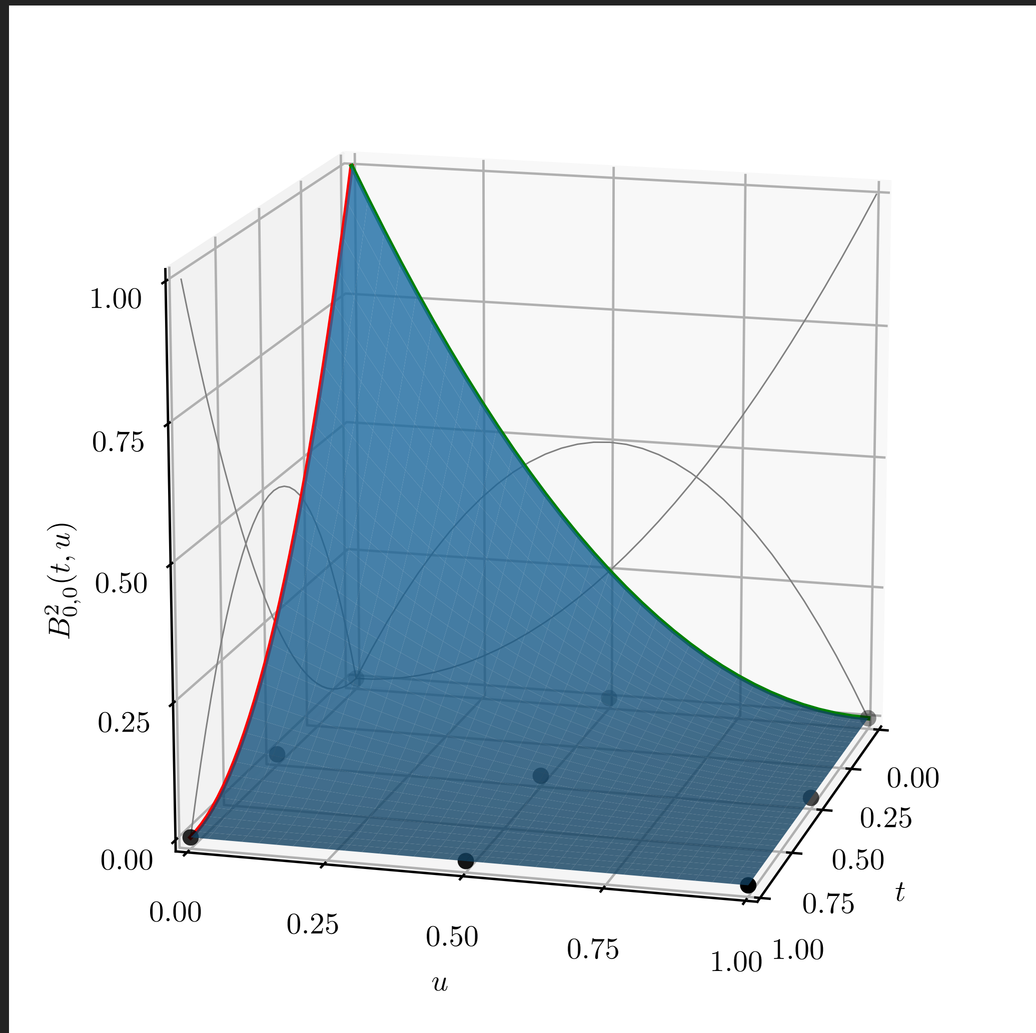

Figure 3.5: Bézier bi-quadratic basis functions.

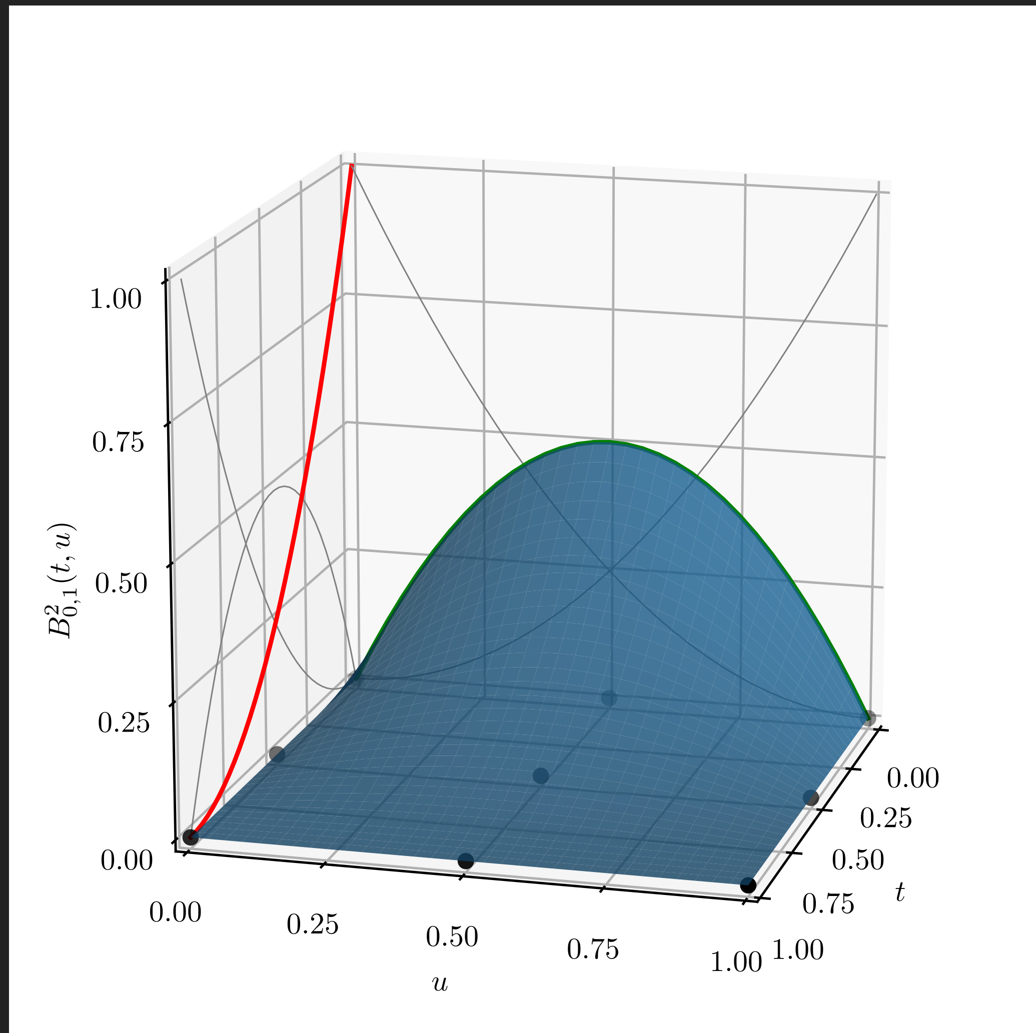

Figure 3.6: Continued from previous figure.

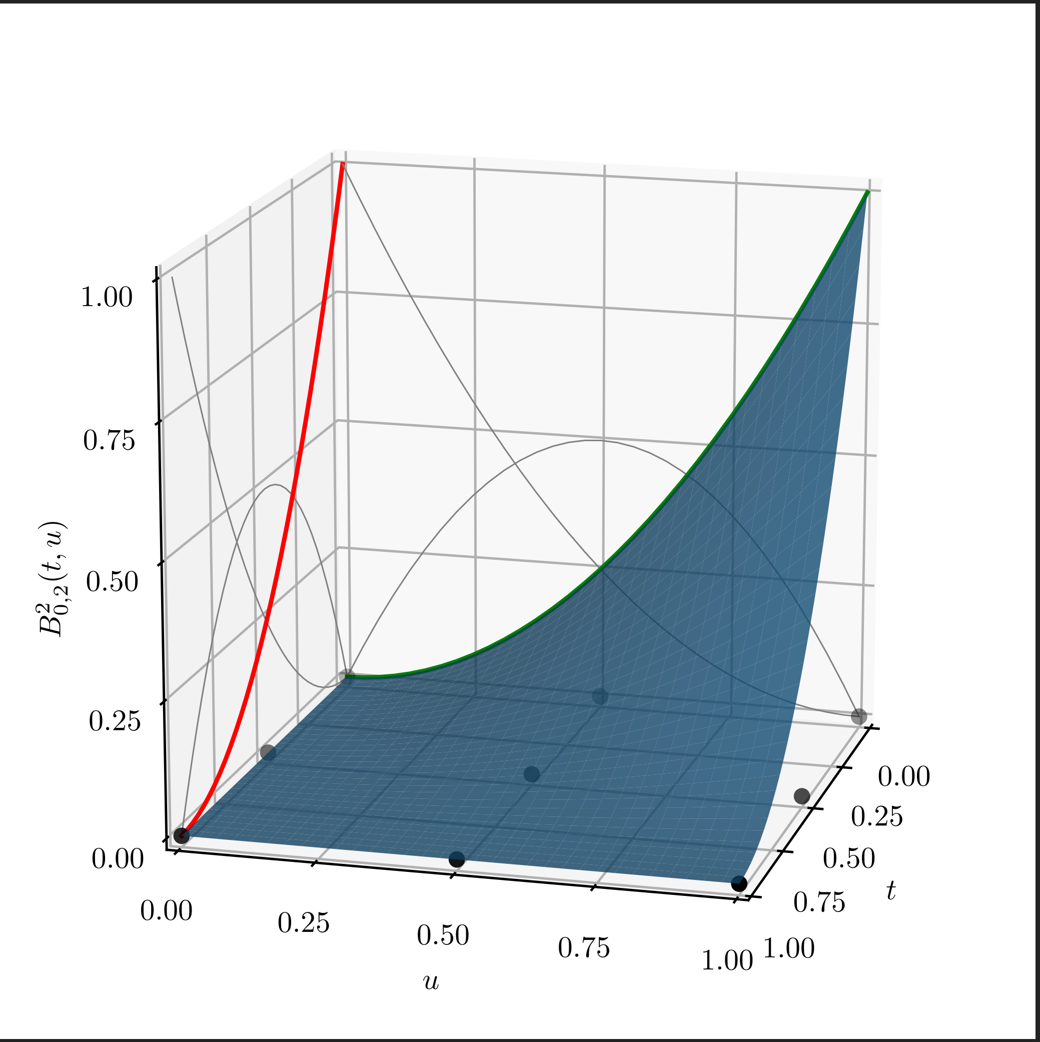

Figure 3.7: Continued from previous figure.

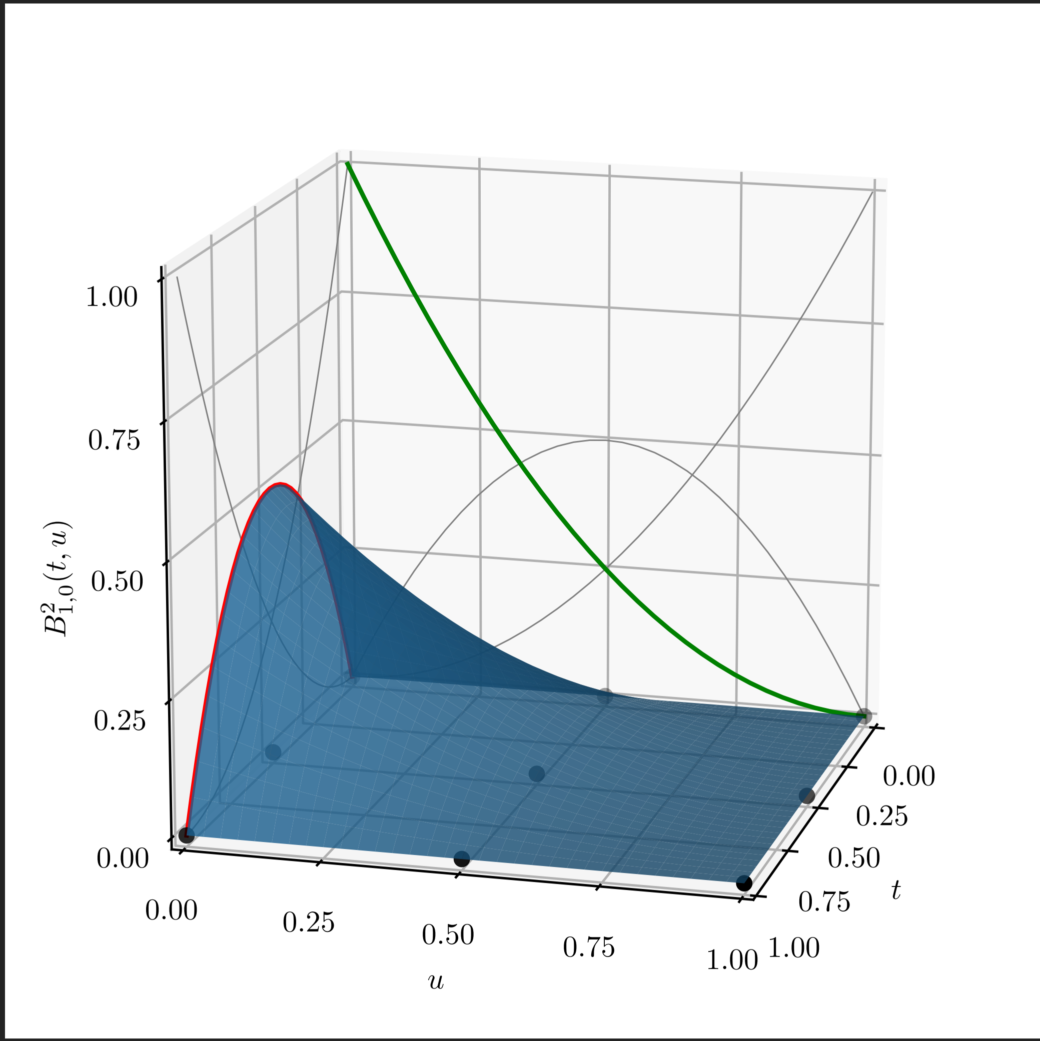

Figure 3.8: Continued from previous figure.

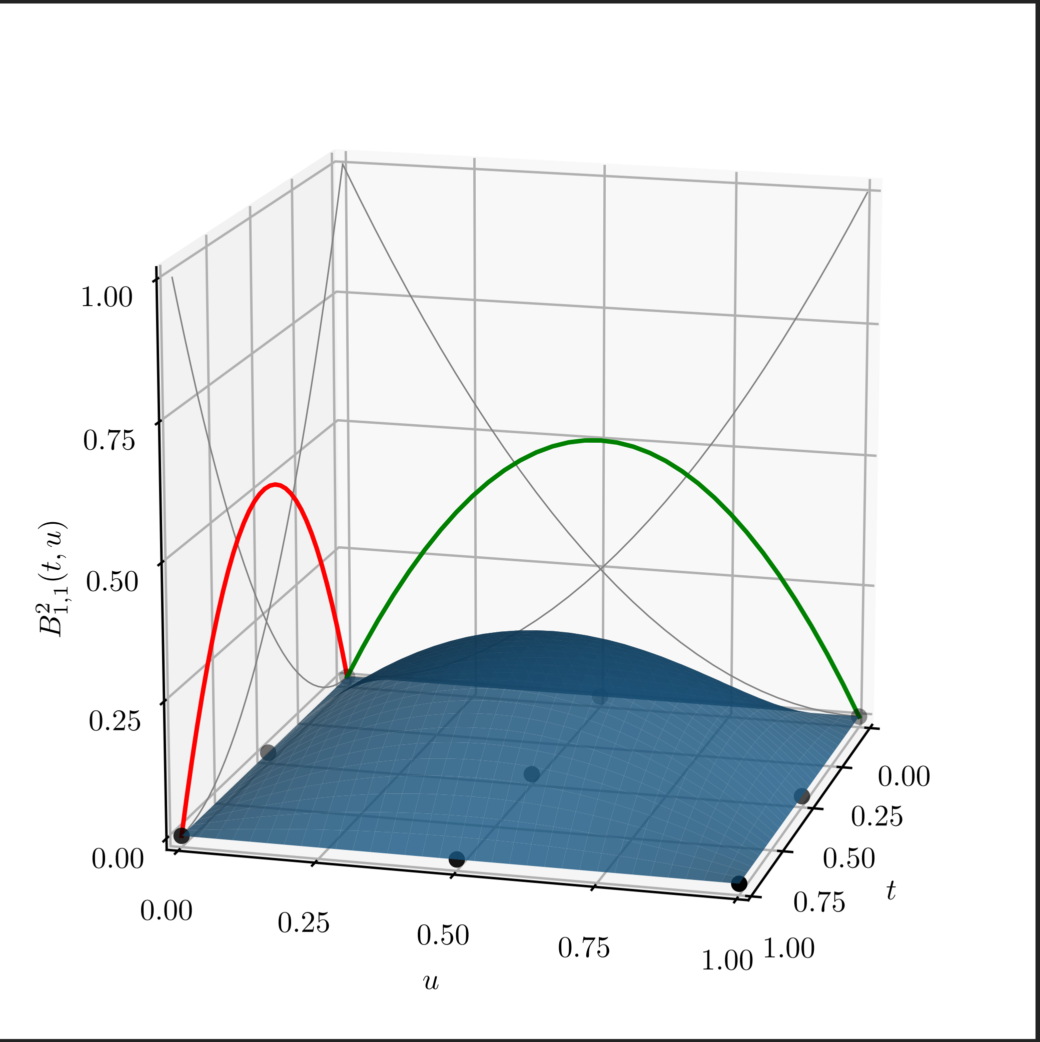

Figure 3.9: Continued from previous figure.

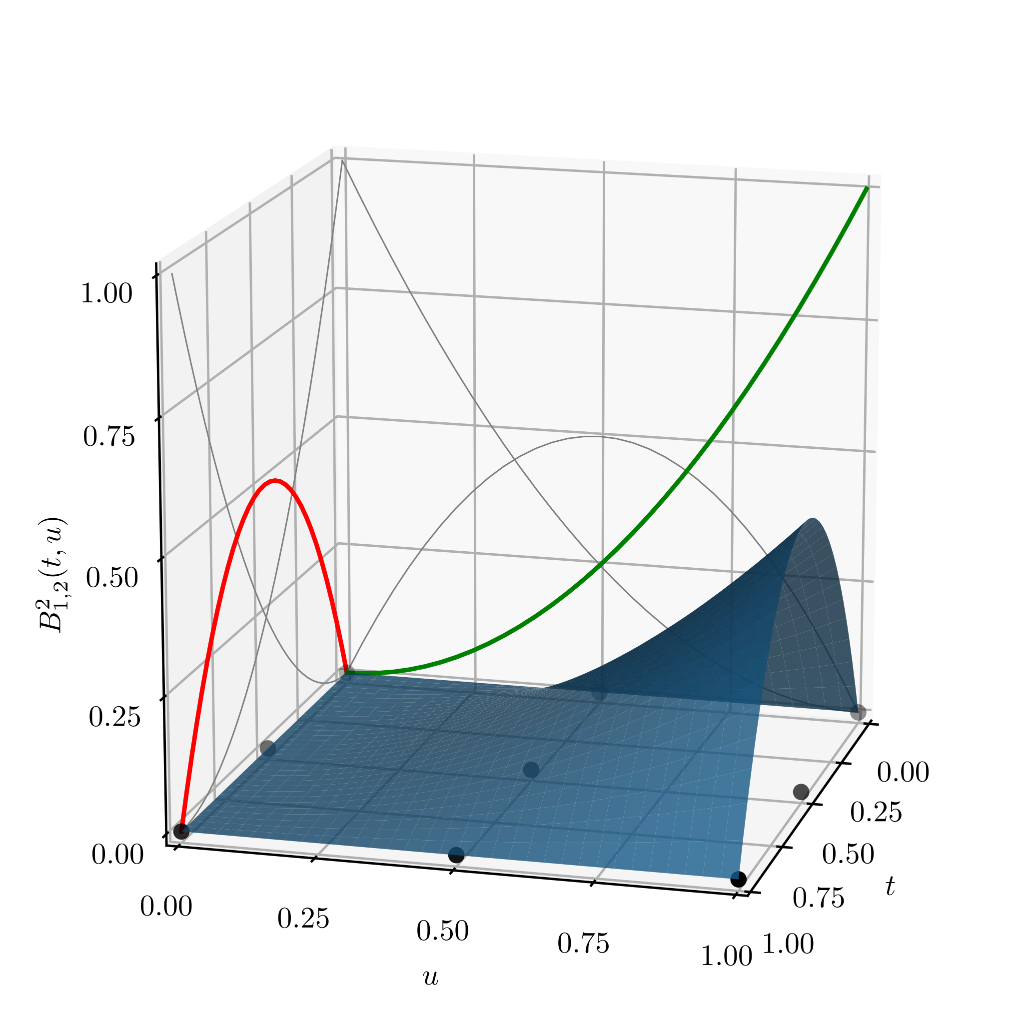

Figure 3.10: Continued from previous figure.

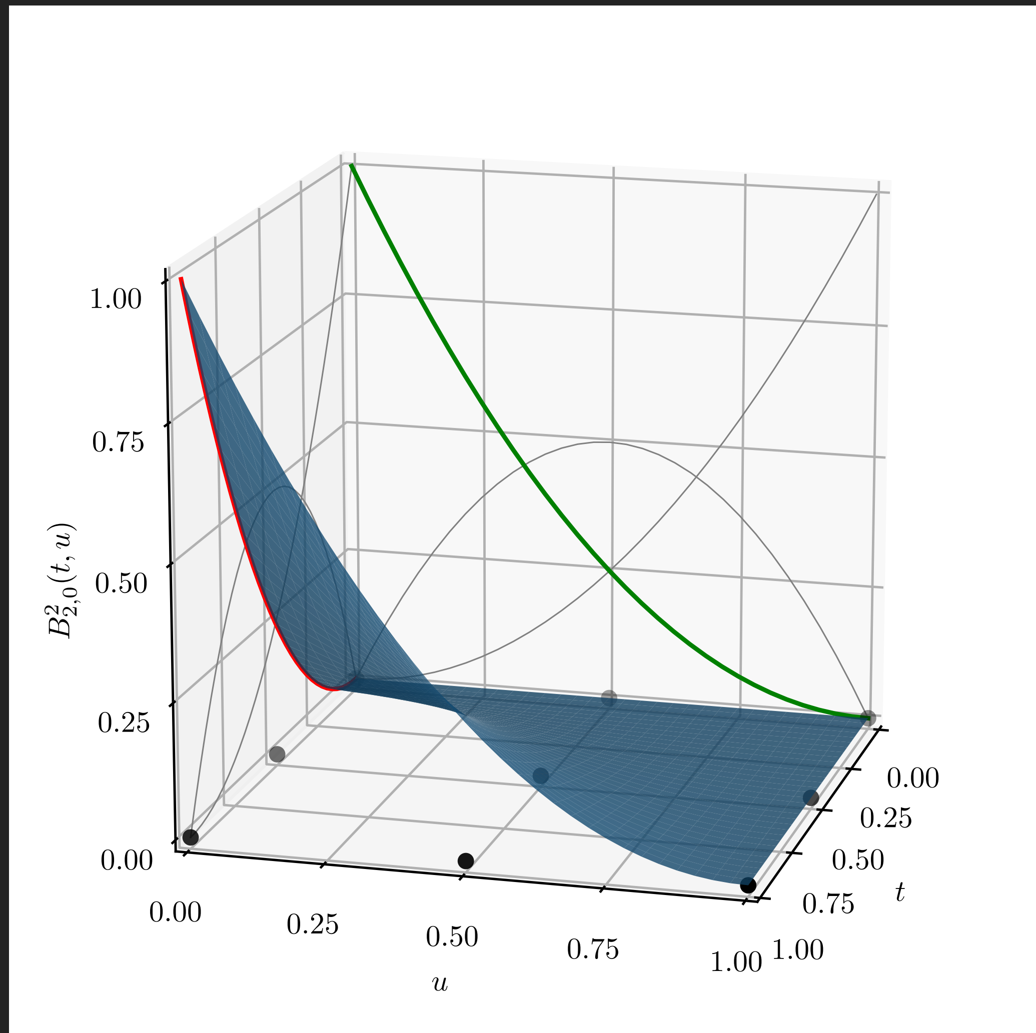

Figure 3.11: Continued from previous figure.

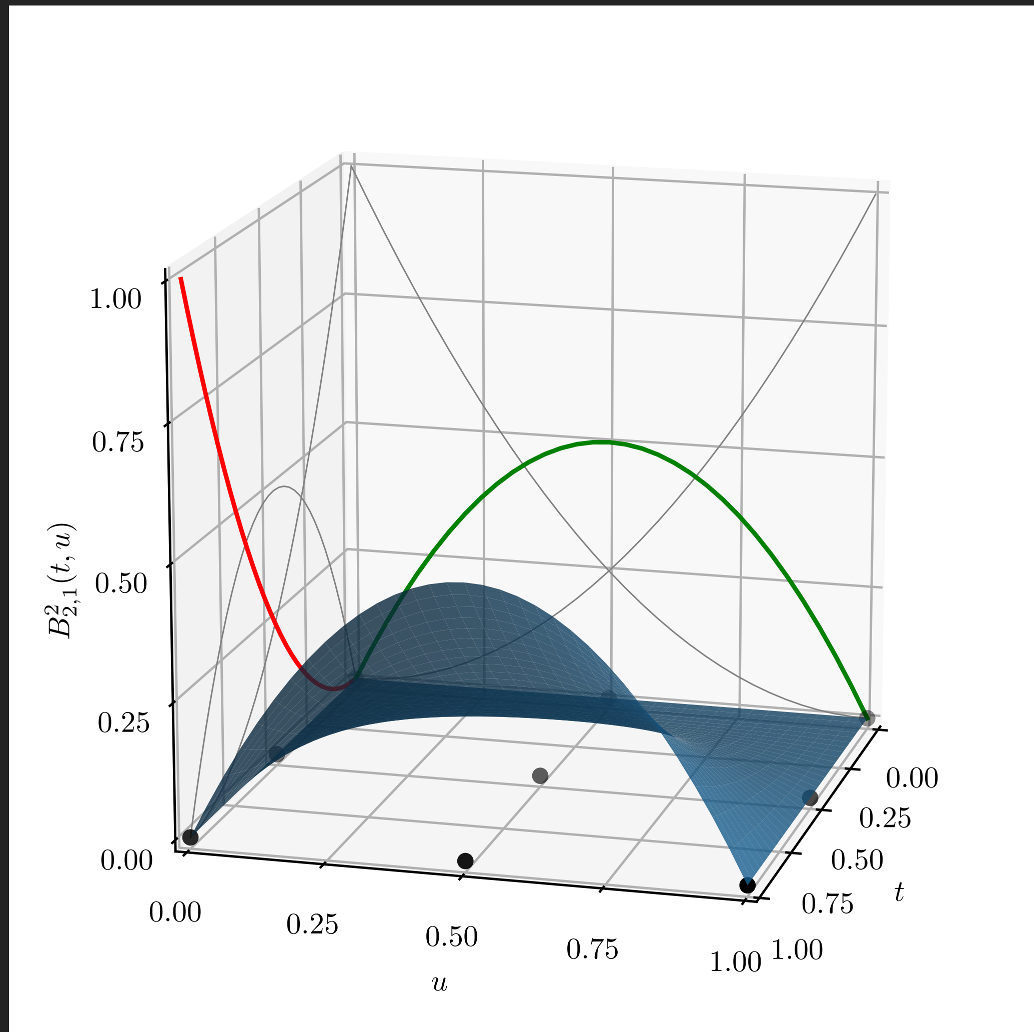

Figure 3.12: Continued from previous figure.

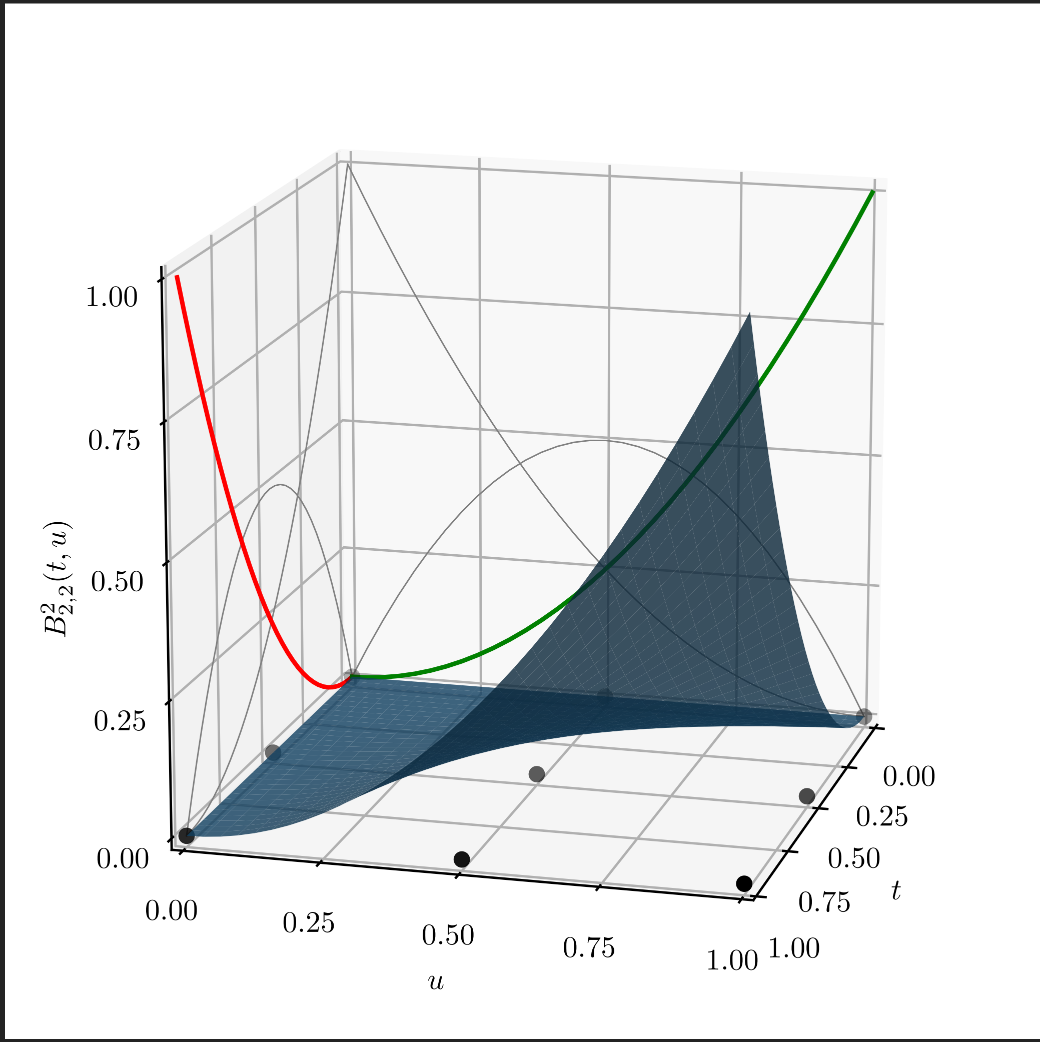

Figure 3.13: Continued from previous figure.

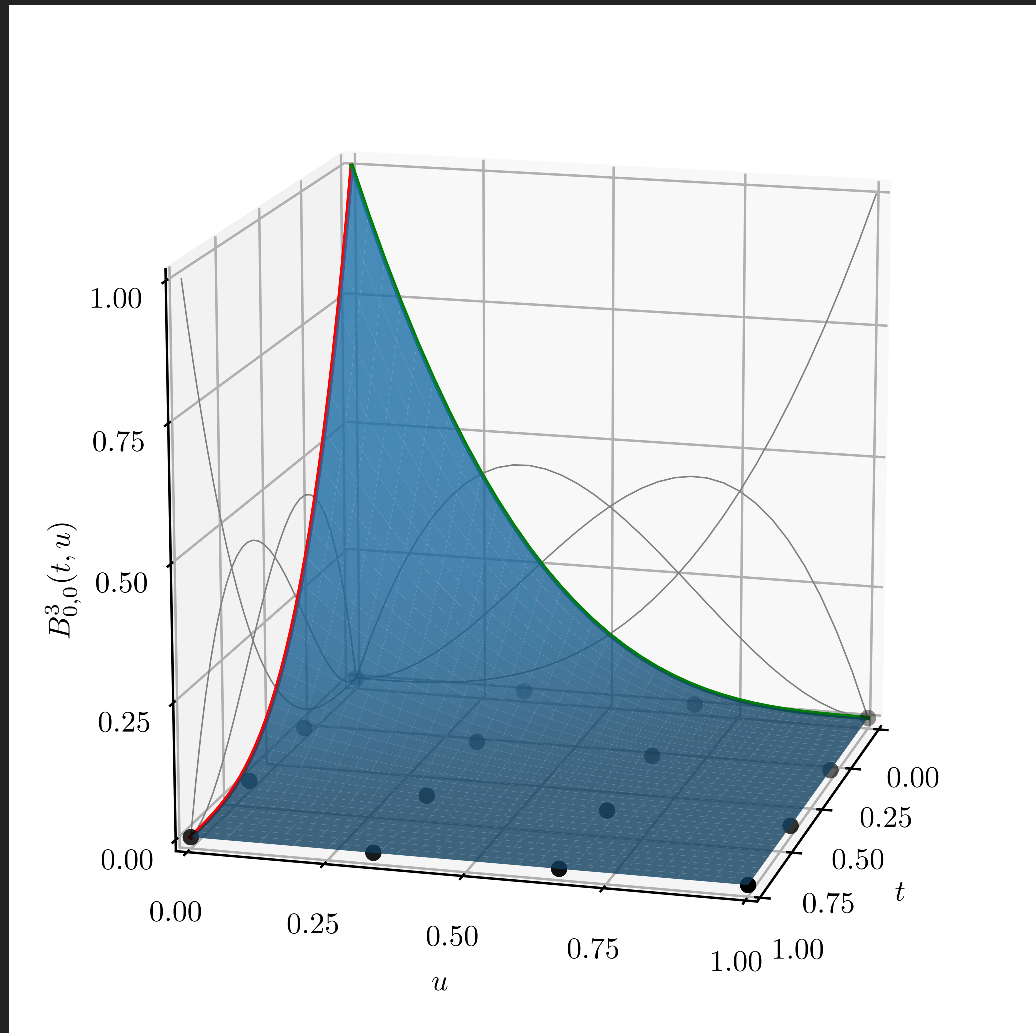

Figure 3.14: Bézier bi-cubic basis functions.

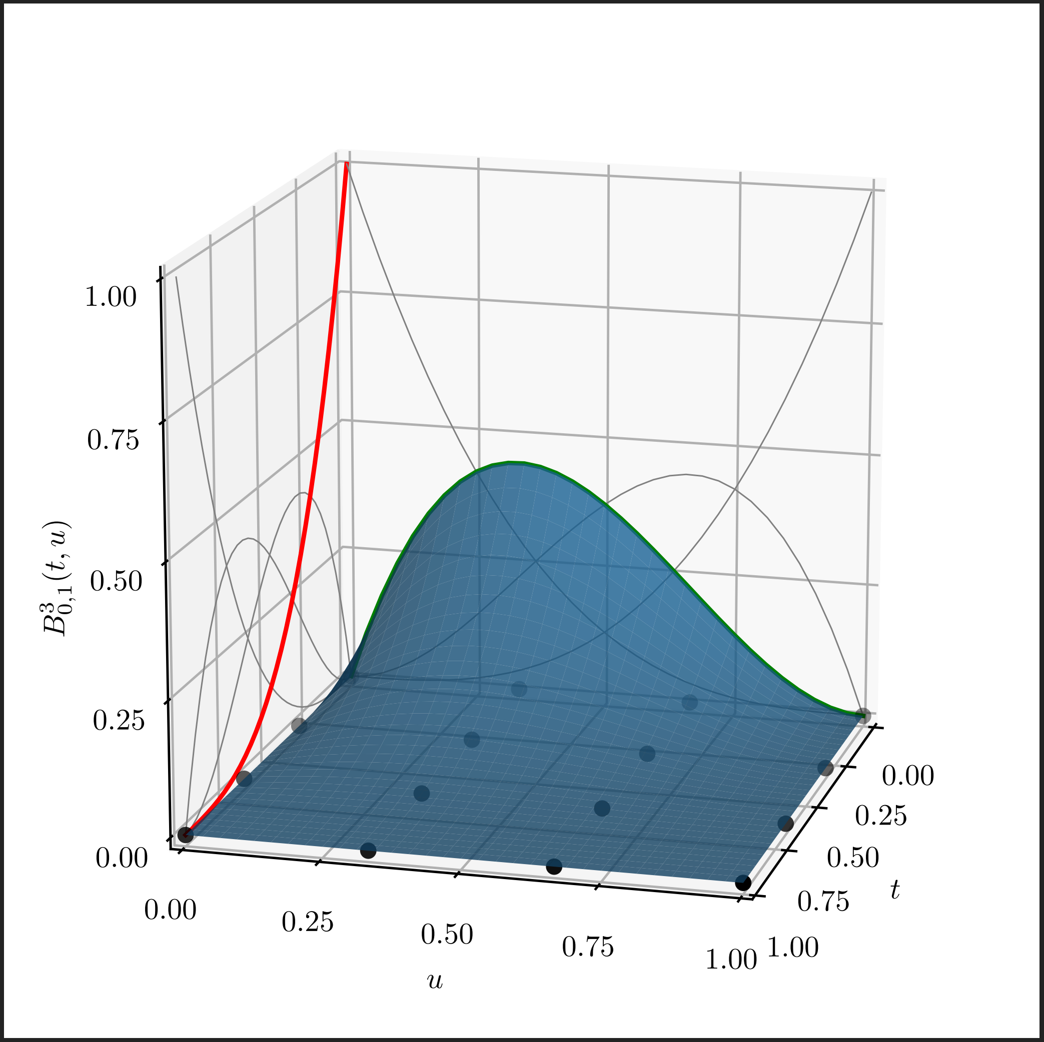

Figure 3.15: Continued from previous figure.

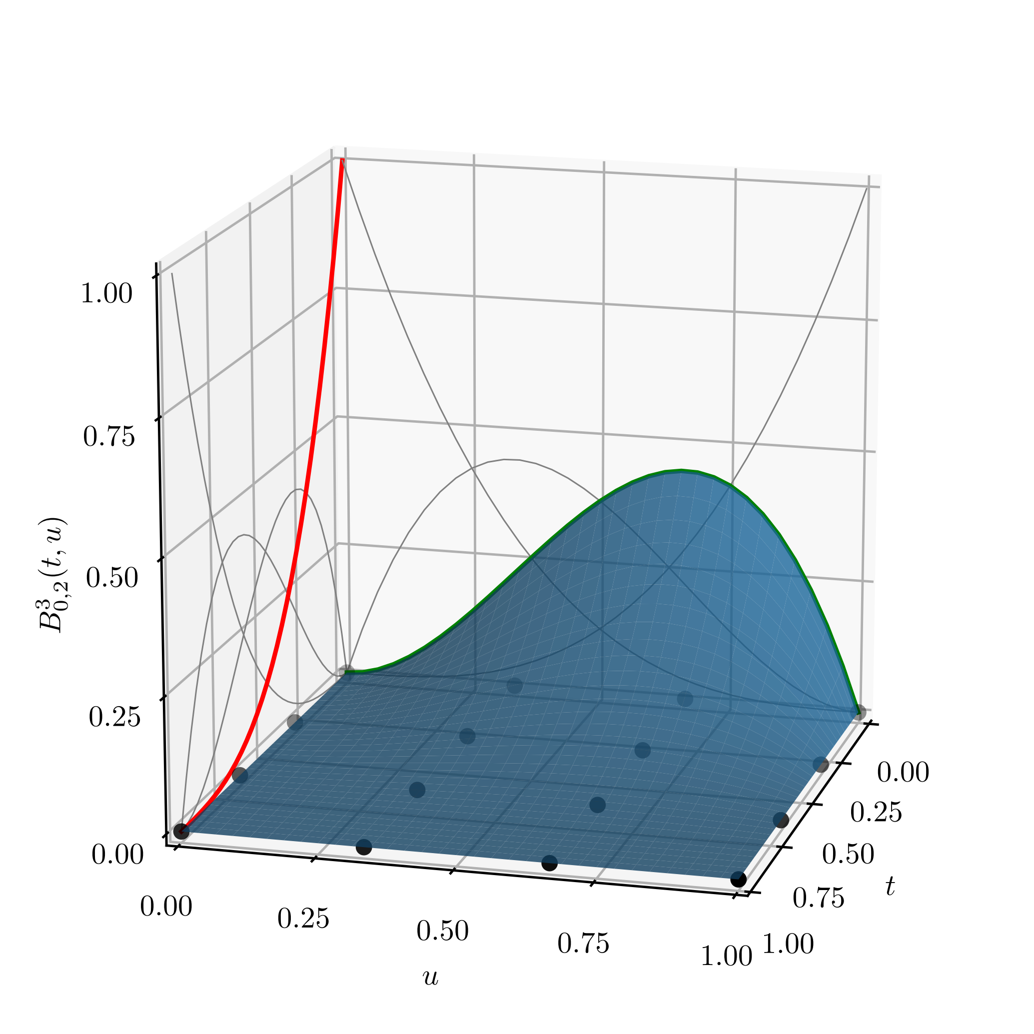

Figure 3.16: Continued from previous figure.

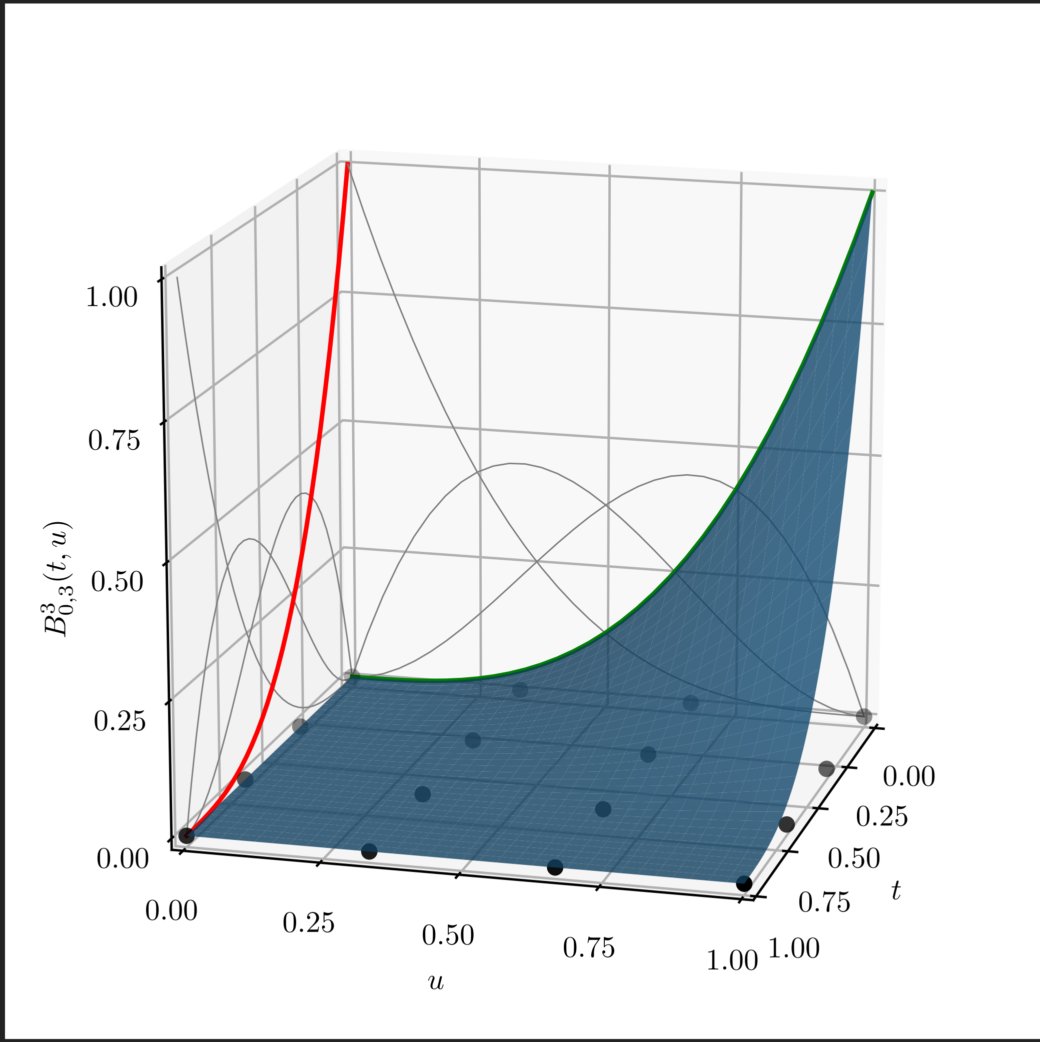

Figure 3.17: Continued from previous figure.

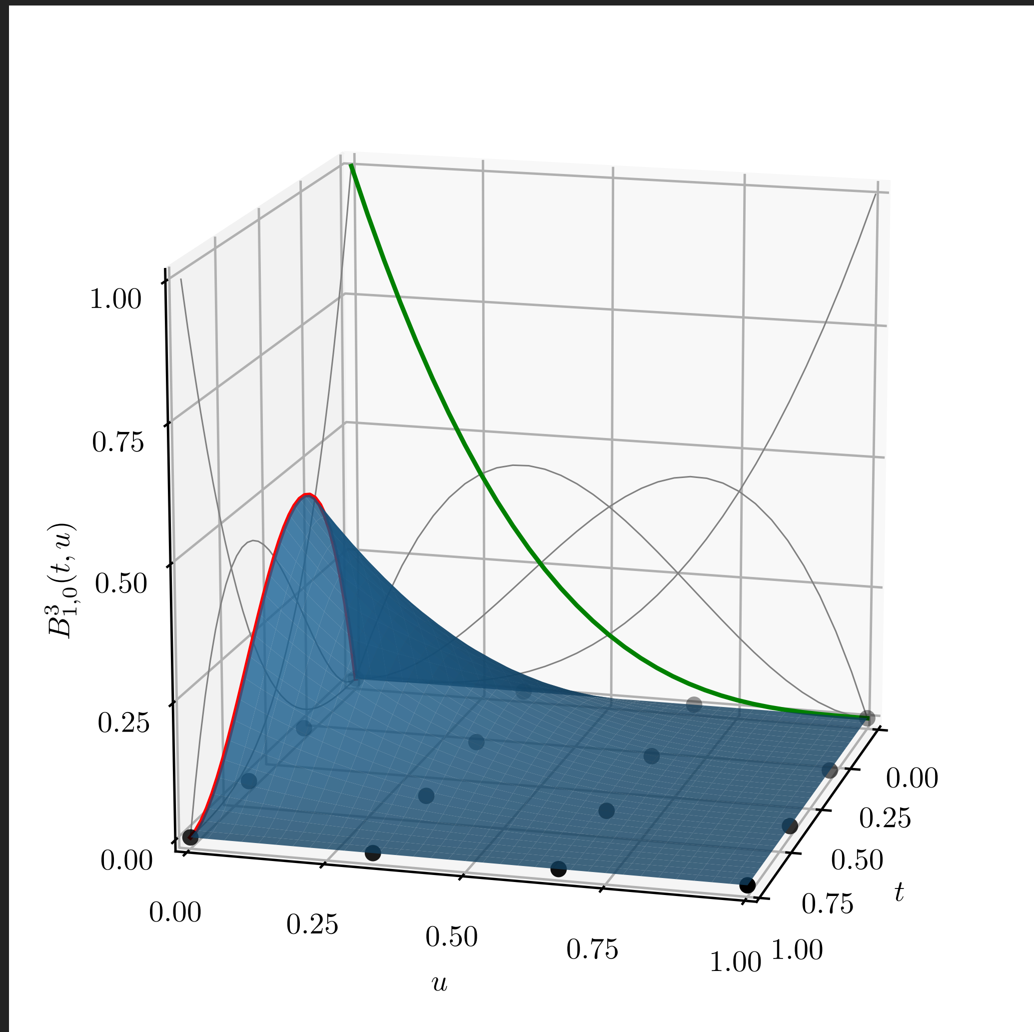

Figure 3.18: Continued from previous figure.

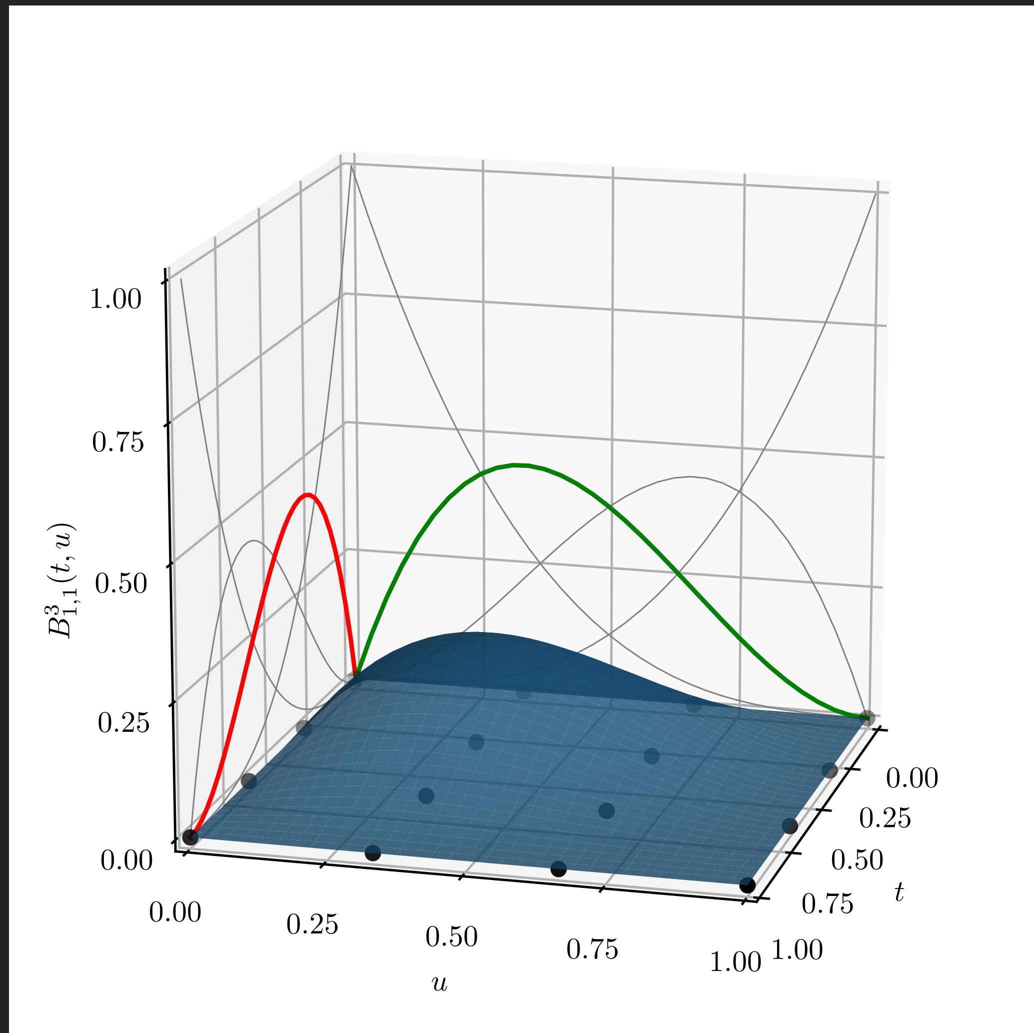

Figure 3.19: Continued from previous figure.

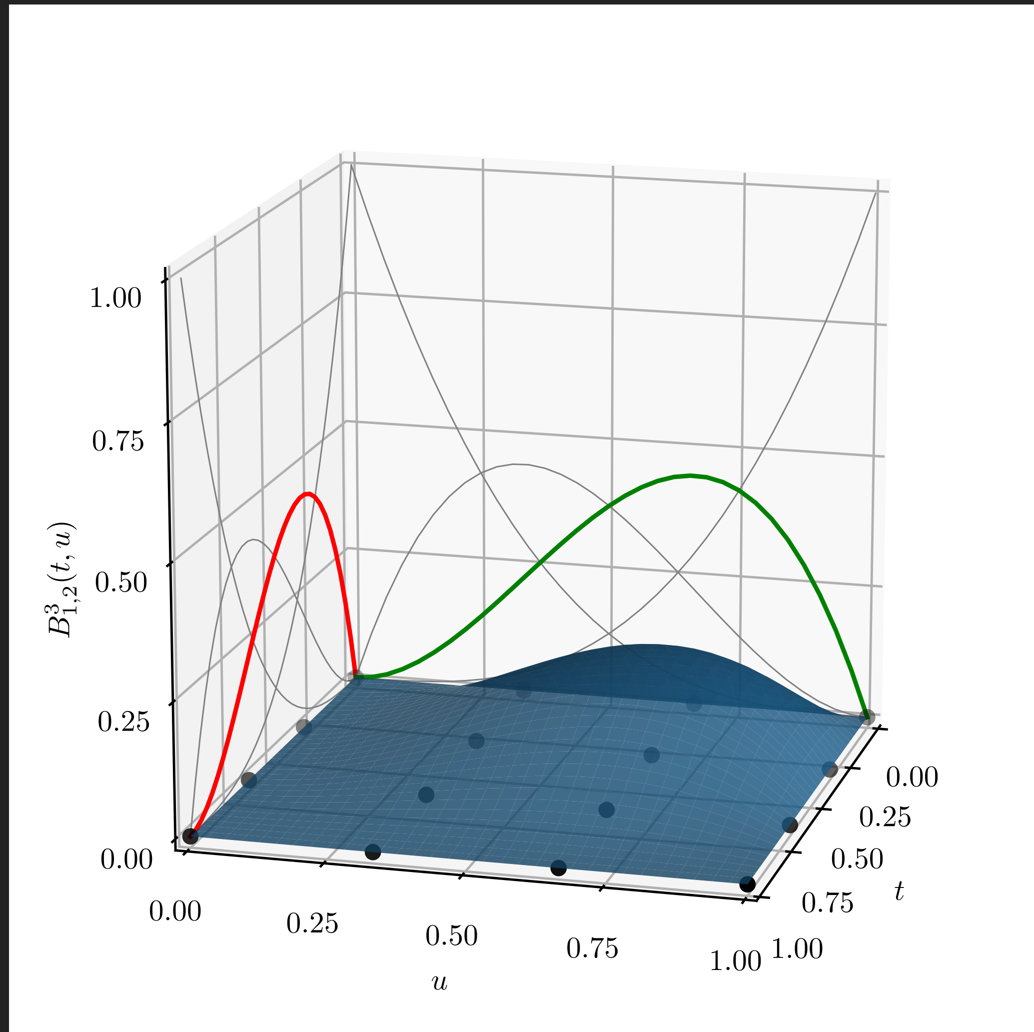

Figure 3.20: Continued from previous figure.

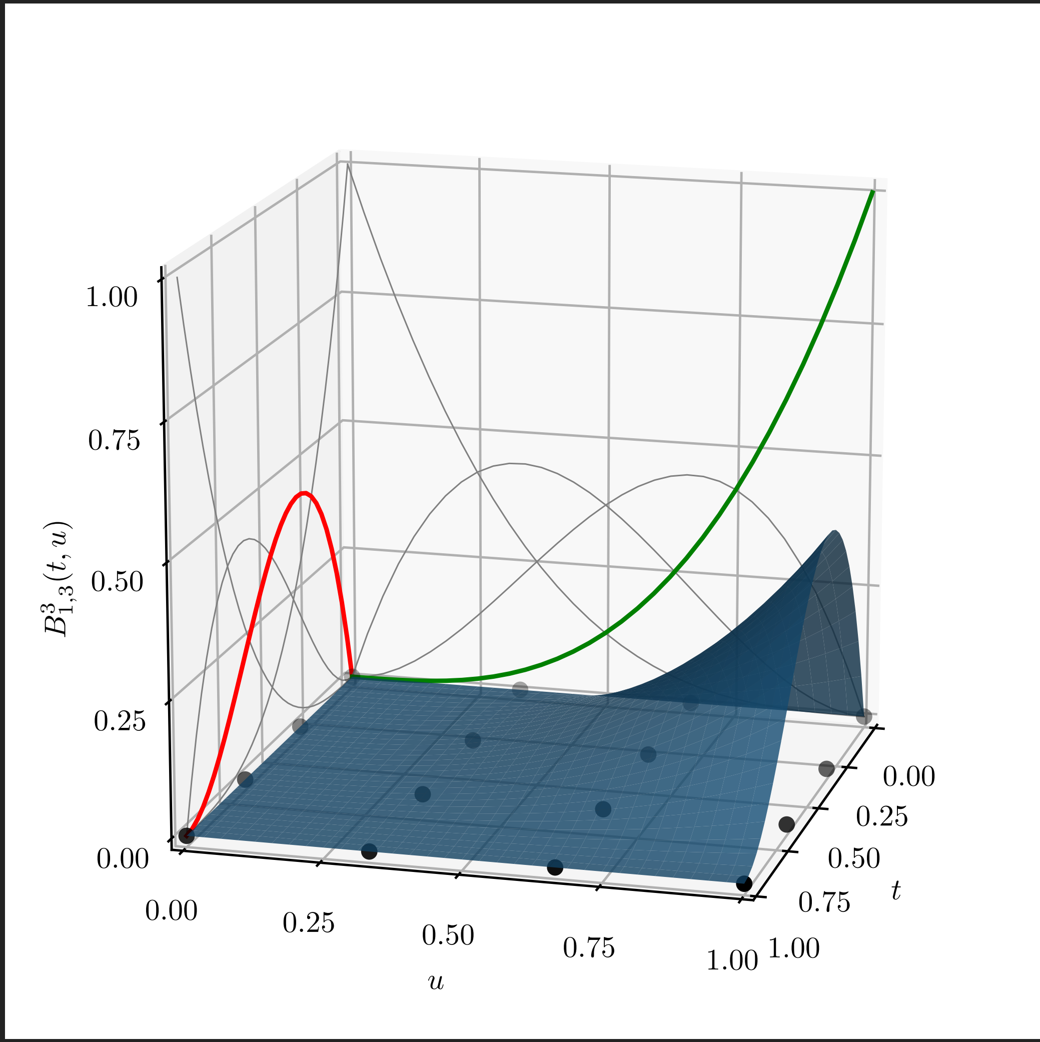

Figure 3.21: Continued from previous figure.

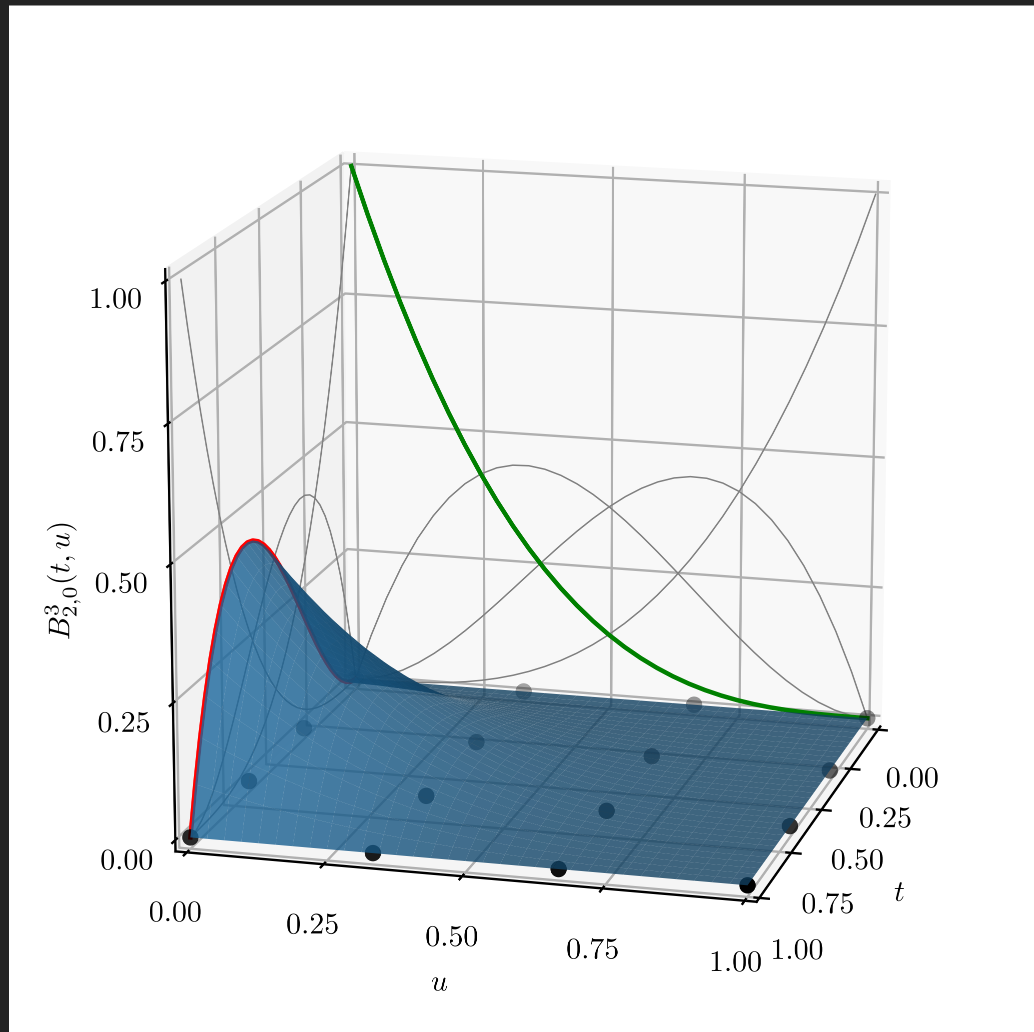

Figure 3.22: Continued from previous figure.

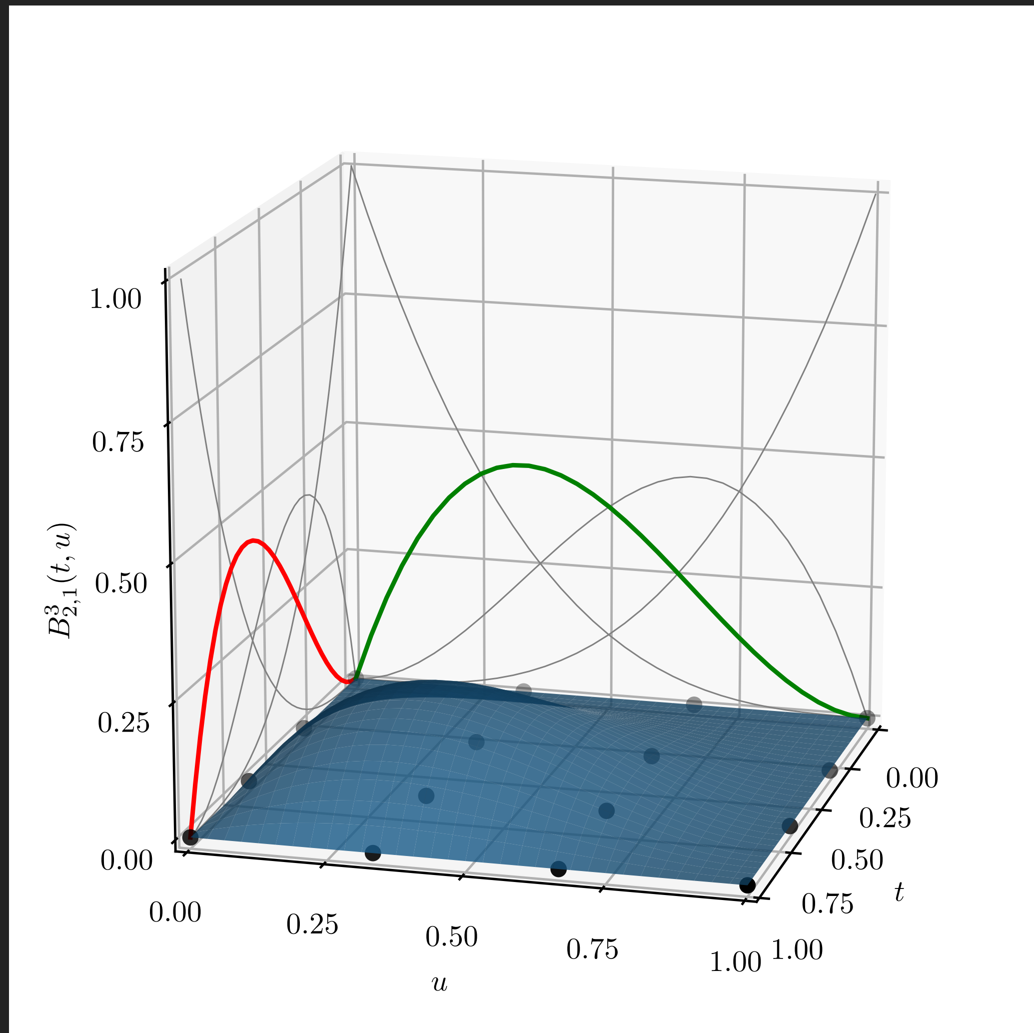

Figure 3.23: Continued from previous figure.

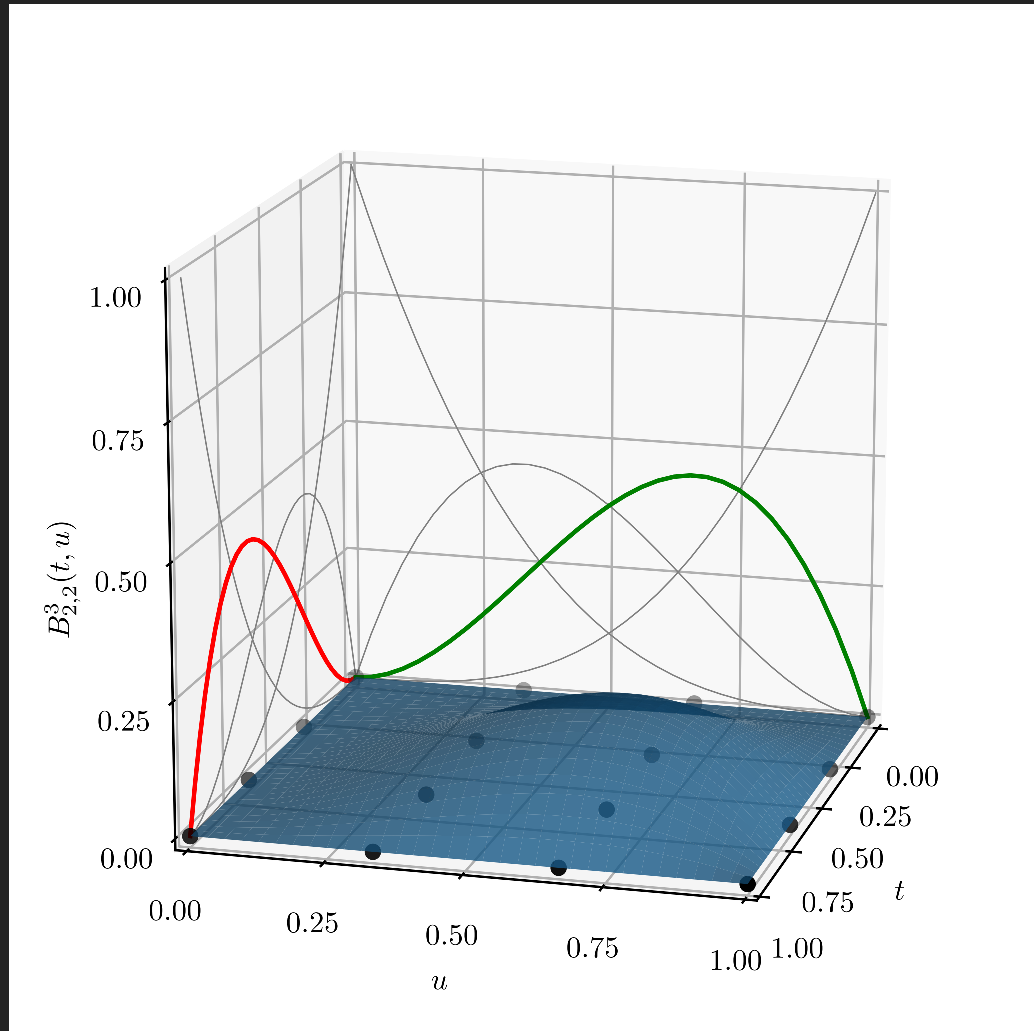

Figure 3.24: Continued from previous figure.

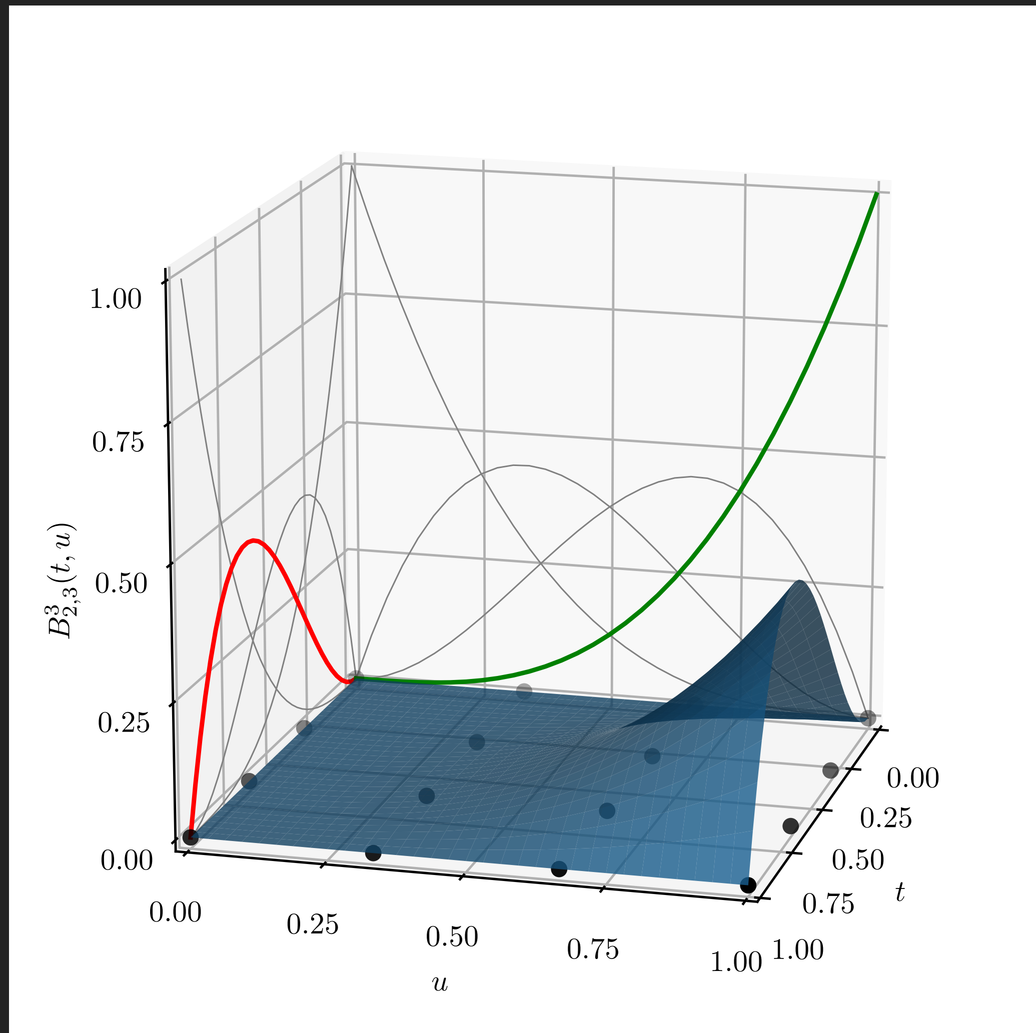

Figure 3.25: Continued from previous figure.

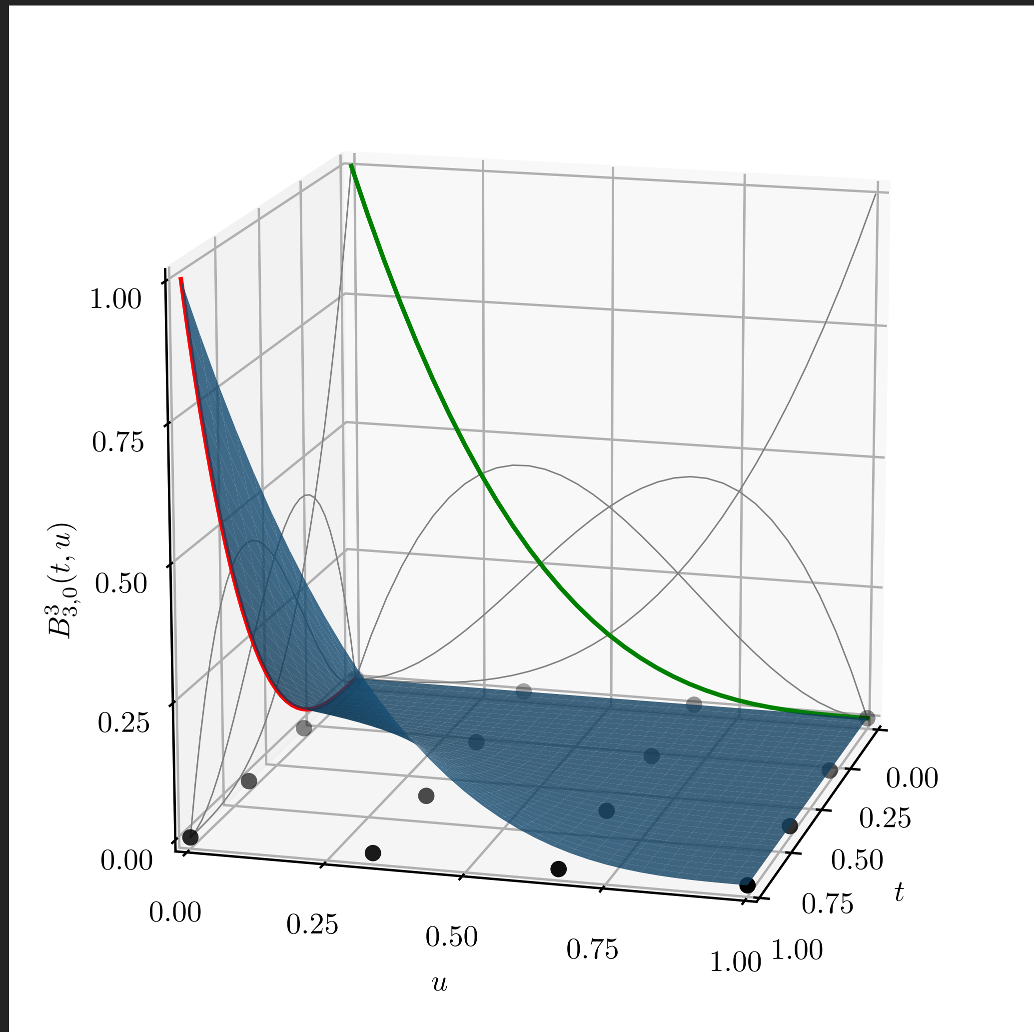

Figure 3.26: Continued from previous figure.

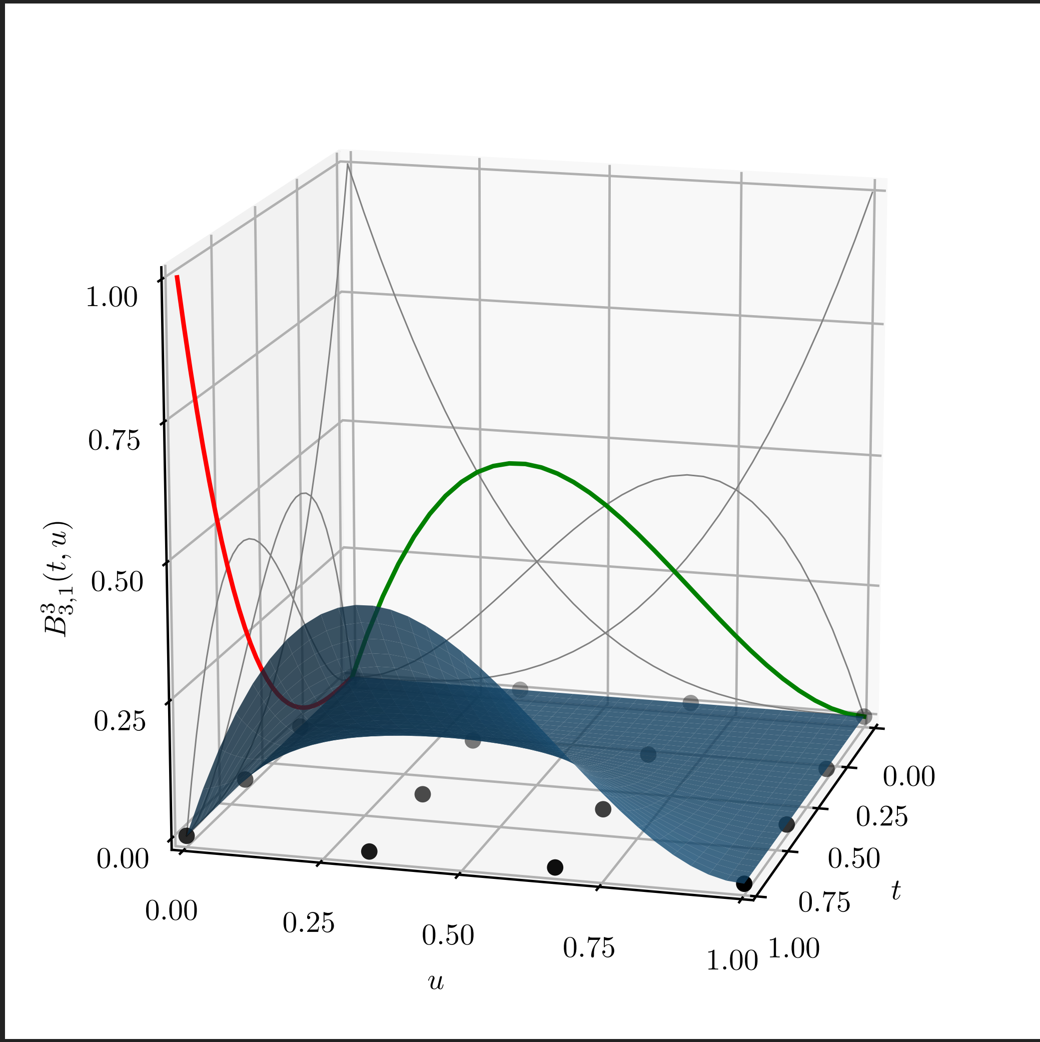

Figure 3.27: Continued from previous figure.

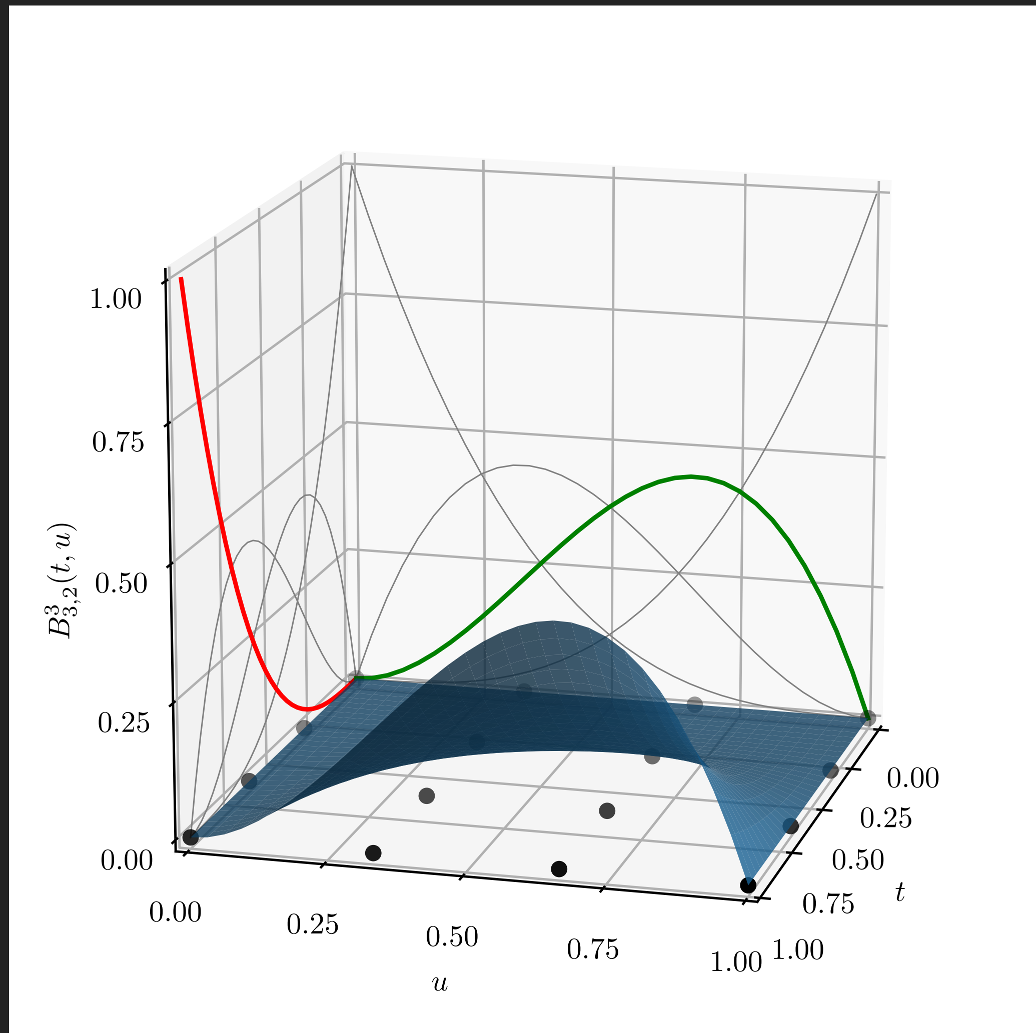

Figure 3.28: Continued from previous figure.

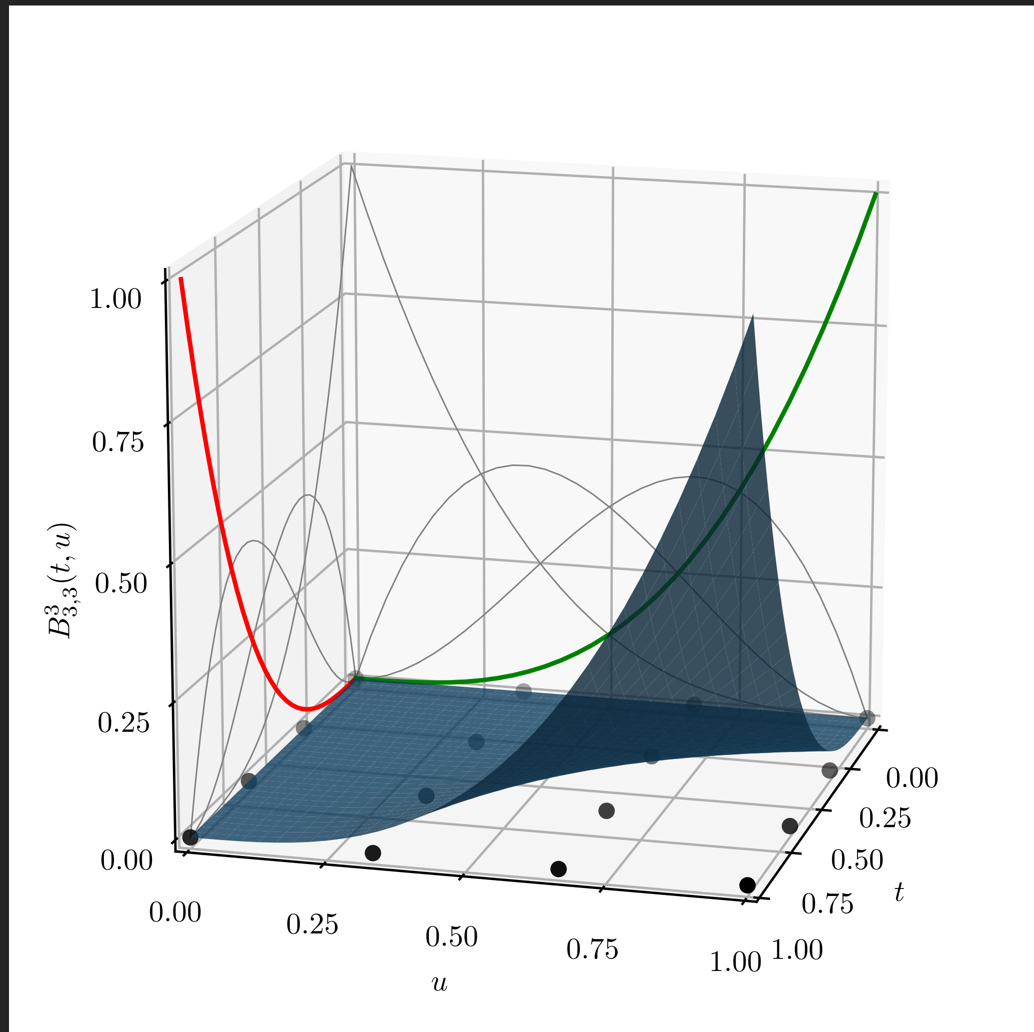

Figure 3.29: Continued from previous figure.

Example 9.
The Utah teapot is a cononical example of shape composition from Bézier surfaces. In Figure 3.30, we show the quarter-model (and half-symmetry) version of the Utah teapot created from ten (10) Bézier bi-cubic ($p = 3$, $q = 3$) surfaces and one-hundred-twenty-seven (127) control points. $\square$

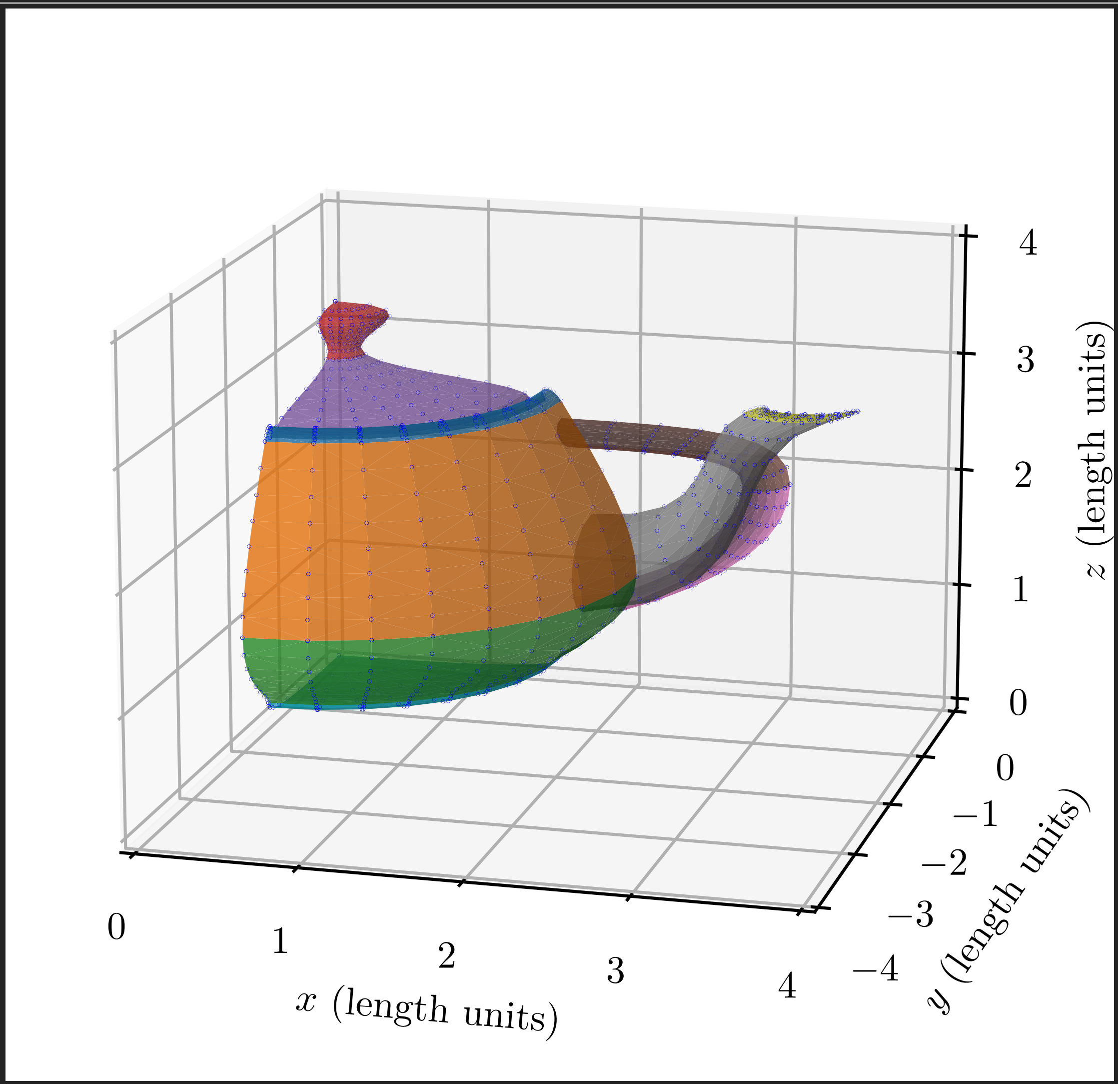

Figure 3.30: The quarter-model (and half-symmetry) version of the Utah teapot composed of Bézier surfaces. See `view_bezier.py` and `utah-teapot-config.json` on the GitHub SIBL repository.

Chapter 4

Bézier Volumes

Bézier volumes derive as a natural dimensional extension of Bézier surfaces. The control point grid used for surfaces becomes a control point lattice for volumes. The general form of a Bézier volume $\mathbb{V}^{p,q,r}(t,u,v)$

> of degree p and $p+1$ control points for the t parameter,
> of degree q and $q+1$ control points for the u parameter, and
> of degree r and $r+1$ control points for the v parameter,

is defined as

$$\mathbb{V}^{p,q,r}(t,u,v) \triangleq \sum_{i=0}^{p} \sum_{j=0}^{q} \sum_{k=0}^{r} B_i^p(t)\; B_j^q(u)\; B_k^r(v)\; \boldsymbol{P}_{i,j,k}. \tag{4.1}$$

The Bézier basis functions are defined as the outer product of three Bernstein polynomials,

$$B_{i,j,k}^{p,q,r}(t,u,v) \triangleq B_i^p(t) \otimes B_j^q(u) \otimes B_k^r(v). \tag{4.2}$$

While not necessary, it is often the case in practice that the number of control points for the t, u, and v parameters are taken to be the same, *i.e.*, $(p+1) = (q+1) = (r+1)$. In this case, the foregoing definition reduces to

$$B_{i,j,k}^{p}(t,u,v) \triangleq B_i^p(t) \otimes B_j^p(u) \otimes B_k^p(u). \tag{4.3}$$

Example 10.
In Figure 4.1, we show a quarter-symmetry thick pipe constructed from one (1) Bézier tri-quadratic ($p = q = r = 2$) volume and twenty-seven (27) control points (three control points for each of the three dimensions).

$\square$

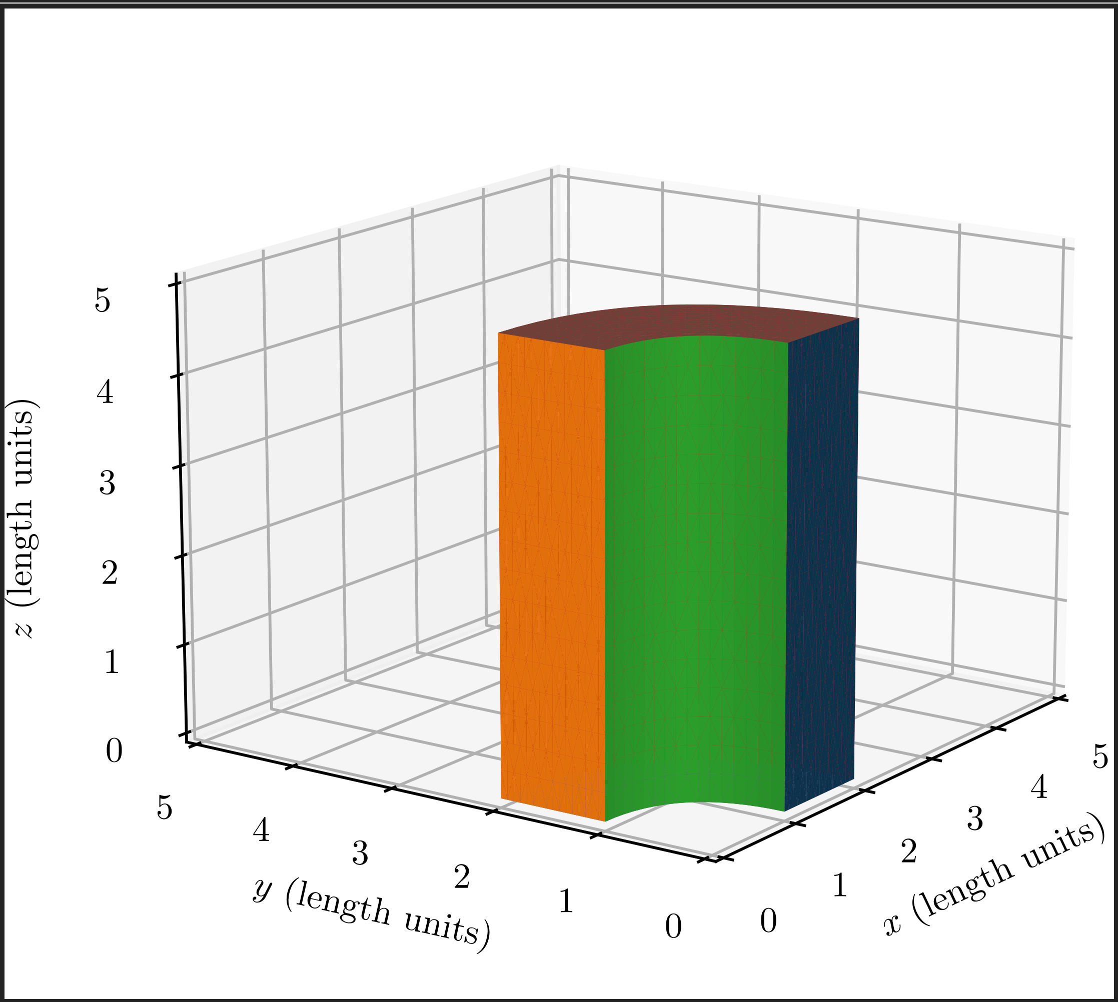

Figure 4.1: The quarter-symmetry thick pipe composed a Bézier volume. See `view_bezier.py` and `triquad-qtr-cyl-config.json` on the GitHub SIBL repository.

Chapter 5

Acknowledgements

We gratefully acknowledge support from the Office of Naval Research (Dr. Timothy Bentley) under Special Studies grant N0001418IP00054.

Bibliography

[Bartels et al., 1995] Bartels, R. H., Beatty, J. C., and Barsky, B. A. (1995). *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann.

[Cottrell et al., 2009] Cottrell, J. A., Hughes, T. J., and Bazilevs, Y. (2009). *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons.

[Piegl and Tiller, 1997] Piegl, L. and Tiller, W. (1997). *The NURBS book*. Springer Science & Business Media.

[Rogers, 2000] Rogers, D. F. (2000). *An introduction to NURBS: with historical perspective*. Elsevier.

[Shiach, 2015a] Shiach, J. (2015a). *B-splines, mathematics of computer graphics and virtual environments*. https://youtu.be/qhQrRCJ-mVg accessed March 10, 2020.

[Shiach, 2015b] Shiach, J. (2015b). *Bézier curves, mathematics of computer graphics and virtual environments*. https://youtu.be/2HvH9cmHbG4 accessed March 9, 2020.