2024

Penalized Interpolating B-splines and Their Applications

Kylee L. Hartman-Caballero
Virginia Commonwealth University

PENALIZED INTERPOLATING B-SPLINES AND THEIR APPLICATIONS

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

KYLEE L. HARTMAN-CABALLERO

Bachelor of Science, William & Mary, 2021

Director: Dr. Laura Ellwein Fix,

Associate Professor, Department of Mathematics

Virginia Commonwewalth University

Richmond, Virginia

April, 2024

Acknowledgements

To God the Father, Jesus Christ His Son, and the Holy Spirit.

To Dr. Laura Ellwein Fix, who shows me everyday that there is nothing that women cannot do.

To my committee, Dr. Fix, Dr. Cheng Ly, and Dr. Oyita Udiani, without whose sacrifice of time and effort, there would be no thesis defense to have in the first place.

To Dr. Marco Aldi, for allowing me to come to VCU when it felt like I had no place to go. To Dr. Rebecca Segal, Dr. Rani Satyam, Dr. Reed Ogrosky, and Dr. Craig Larson for being my cheerleaders behind the scenes.

To my fellow graduate students, my greatest of friends: Katherine Javonillo, Richard Foster, Roberto Estrada, Maya Tennant, Paul Fay, Robert Hicks, Samuel Bevins, Christopher Flippen, John Carney, and many others. It was the best of times, it was the worst of times.

And finally to my family: my mom, my aunt, my uncle. But most of all, to Nana and Papa. I think about you every day.

And as always... Go Pack Go.

Abstract

One of the most studied data analysis techniques in Numerical Analysis is interpolation. Interpolation is used in a variety of fields, namely computer graphic design and biomedical research. Among interpolation techniques, cubic splines have been viewed as the standard since at least the 1960s, due to their ease of computation, numerical stability, and the relative smoothness of the interpolating curve. However, cubic splines have notable drawbacks, such as their lack of local control and necessary knowledge of boundary conditions. Arguably a more versatile interpolation technique is the use of B-splines. B-splines, a relative of Bézier curves, allow local control through knot insertion, do not require knowledge or assumption of boundary conditions for computation, and have continuous curvature. Another way to exert control on the B-spline curve is to minimize its roughness. Penalized B-splines, also called *P-splines*, are an emerging method of approximation and interpolation formulated in the mid-1990s by Eilers and Marx. Through definition, example, and application to cerebrovascular resistance data, we will explore the utility and benefits of P-splines.

VITA

Kylee Lynn Hartman-Caballero was born on November 27, 1998, in Newport News, Virginia, and is an American citizen. She graduated from Woodside High School, Newport News, Virginia in 2017. She received her Bachelor of Science in Mathematics from William & Mary, Williamsburg, Virginia in 2021 where she was inducted into the Pi Mu Epsilon mathematics honor society and was active in the Latin American Student Union and campus ministry organizations. She interned at NASA Langley in Hampton, Virginia in 2020 and continues to volunteer in her community.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Just after the end of World War II, in 1946, Romanian-American mathematician I.J. Schoenberg published "Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions" in *Quarterly of Applied Mathematics* [1]. It is in this publication that the word *spline* is first used to refer to a smooth, piecewise polynomial function that approximates data. It wasn't until the late 1950s, however, that further research into the application of Schoenberg's splines occurred. This was largely due to the Computer Revolution, which saw a new interest in computational mathematics and the problems that could be solved beyond the limits of by-hand computation. The foremost names during this period were Paul de Casteljau, Pierre Bézier, and Carl de Boor.

In 1972, de Boor published "On Calculating with B-Splines" in *Journal of Approximation Theory* [2] where he revisited the work by Schoenberg from 1946. This eventually led to the publication of his book *A Practical Guide to Splines* in 1978 (of which the 2001 revised edition will be further referenced in this thesis), where de Boor defined a recursive formula for the calculation of B-spline basis functions, called the Cox-de Boor recursion formula. Research in splines died down until the mid-1990s when Eilers and Marx published "Flexible Smoothing with B-splines and Penalties," [3][4]. This publication gave way to a new topic in spline research: the penalized B-spline, or *P-spline*.

The motivation for assessing a penalty on the curve is to balance the spline's roughness and its approximation of the given data. Eilers and Marx generalized work that had been published in 1986 by Finbarr O'Sullivan [4]. O'Sullivan recognized that the integral of the squared second derivative of the B-spline curve can be expressed as a quadratic function of the spline's control points [4], [5]. Eilers and Marx eliminated the derivative completely and expressed a measure of roughness as the sum of differences between control points.

This thesis will first define important terms related to B-splines, including the definition of a spline itself. Splines primarily have two utilities: approximation and interpolation. We will explore penalization on interpolating B-splines, which will require defining an interpolation spline. Next, we will build the foundation of interpolating B-splines by defining the B-spline basis functions and exploring their properties. Faced with the task of interpolating given data, we will explore two routes. The de Boor algorithm calculates the value of a B-spline at a value of interest in one dimension. Lim's *Universal Method* simplifies the process of parameterization and knot vector generation to then easily interpolate in any number of dimensions.

We will then discuss penalization as it relates to reducing curvature and roughness. Because we are interpolating data and not approximating it, closeness of fit will not be a factor we will consider. In our discussion of penalization, we will consider briefly the O'Sullivan penalty and extensively the Eilers and Marx difference penalty. Chapter 5 will demonstrate penalized B-splines on air flow data and briefly discuss limitations to the concepts presented in this thesis, and conclude with possibilities for future work.

CHAPTER 2

TERMINOLOGY

2.1 Defining a spline

Before being able to conceptualize mathematically what a *penalized spline* is, we must be able to understand to what it is we are referring. Admittedly, it seems that *penalized spline* does not have a uniform definition across disciplines, and furthermore the interdisciplinary use of the term *spline* itself means that we require careful definition here to avoid further ambiguity. The reason that it is not simple to find a concise definition of a *spline* in relevant literature is that its definition tends to vary by its purpose. For the purpose of data interpolation, Späth presents the following definition of a *spline interpolant*, also referred to as an *interpolating spline* [6]

Definition 2.1.1 (Interpolating spline)**.** For a non-decreasing sequence $u := (u_i)_0^m$, an *interpolating spline* is a set of m functions s_k defined on $[u_k, u_{k+1}]$, respectively, with $0 \leq k \leq m$, that are stitched together so as to be continuously differentiable at every u_i and which satisfy the condition

$$s_k(u_k) = y_k \quad s_k(u_{k+1}) = y_{k+1}$$

for some values y_k and y_{k+1}

We call the values u_i the *knots*. An essential aspect of the definition of a *spline*, no matter its utility, is its fundamental structure as a piecewise function. Interpolating splines need not be comprised of entirely polynomials. However, polynomials lend themselves to being predictable in their continuity and differentiability, which makes manipulation between knots and consideration of boundary conditions easier. As such, we next define a *piecewise polynomial function* [7].

Definition 2.1.2 (Piecewise Polynomial Function)**.** Let $\zeta := (\zeta_i)_1^{l+1}$ be a strictly increasing sequence of points, and let p be a positive integer. If $f_1, \ldots, f_l$ is any sequence of l polynomials, each of order p (that is, degree $< p$), then we define the corresponding polynomial f of order p by

$$f(x) := f_i(x) \quad \text{if} \quad \zeta_i < x < \zeta_{i+1}, \quad 1 \leq i \leq l.$$

A notable discrepancy between Definitions 2.1.1 and 2.1.2 is the defining of u as a non-decreasing sequence versus the defining of ζ as strictly increasing. This will not prove to be an issue since our method of B-spline construction (see section 3.4) is built on the allowance of repeated knots, and so our *knot spans* (the interval between a knot and the one after it in the knot sequence) will be half-open intervals, where we have that $[u_i, u_{i+1})$ and $[u_{i+1}, u_{i+2})$ are successive knot spans. In fact, this difference is mostly notational. Definition 2.1.2 does not assume that $f_i(\zeta_{i+1}) = f_{i+1}(\zeta_{i+1})$. If we were to make this assumption in that definition, then we would have a piecewise polynomial function that interpolates its knots. *Cubic splines* are just this, with the requirement that $p = 4$.

A foundational step of using polynomial interpolating splines is to identify what degree polynomial is most effective in fitting given data. It is common to choose cubic polynomials because they offer derivative continuity and have fewer concavity changes than higher degree polynomials. Interpolation with cubic polynomials is implemented with cubic splines, which possess the following properties:

1. $s_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \quad \forall\, [x_j, x_{j+1}], \quad j = 0, \ldots, n-1$ where n is the number of knots, $a_j, b_j, c_j, d_j \in \mathbb{R}$, and $[x_j, x_{j+1}]$ is a given knot span.

2. $s_i(x_i) = y_i$ and $s_{i+1}(x_{i+1}) = y_{i+1}$ for each data point (x_i, y_i).

3. $s_j(x_{j+1}) = s_{j+1}(x_{j+1})$

4. $s_j^{'}(x_{j+1}) = s_{j+1}^{'}(x_{j+1})$

5. $s_j^{''}(x_{j+1}) = s_{j+1}^{''}(x_{j+1})$

A cubic spline's control points are determined by the value of its derivatives at its endpoints and the function value at the endpoints. We use these to determine the coefficients on the polynomials that comprise the piecewise function we call the cubic spline. What is rather limiting about this is that we need the values of all the control points to determine all the polynomials. One may also hope that if we are interpolating a data point and we shift said point within an arbitrary neighborhood, that the rest of the interpolation would not be vastly different, but shifting a point in a cubic spline has the potential to change the control points, which in turn could affect the rest of the interpolation.

2.1.1 Bézier or Basis?

In this thesis, we will use the abbreviation *B-spline* to refer to a *basis spline curve* as defined in section 3.1. The reader should be cautioned that the literature across disciplines may use this abbreviation to refer to *Bézier basis curves*. Both of these curve types interpolate points using a basis construction. However *Bézier basis curves* use *Bernstein polynomials* as the chosen basis and interpolate with the *de Casteljau Algorithm* [8], [9]. Comparatively, *basis spline curves* use the *Cox-de-Boor recursion formula* to construct the basis and interpolate using *de Boor's Algorithm*, which will be discussed in detail later in this thesis. The construction of Bézier basis curves is beyond the scope of this thesis.

CHAPTER 3

MATHEMATICAL CONSTRUCTION OF B-SPLINES

3.1 Background

There are several ways to interpolate data with B-splines. However, all B-spline interpolations require data points, knots, control points, and parameters. We define the *B-spline basis functions* as follows [7], [8], [10]:

Definition 3.1.1 (B-spline basis function)**.** Given a knot vector $U = [u_0, u_1, \ldots, u_m]$, the associated *i-th B-spline basis function* $N_{i,p}(u)$ of degree p is defined by the function

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

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

for $p > 0$ and $0 \leq i \leq m$.

From this we define the *B-spline curve interpolation* problem [7], [11].

Definition 3.1.2 (B-spline curve interpolation problem)**.** Given a set of data points $D_i \in \mathbb{R}^k, 0 \leq i \leq n$, the *B-spline curve interpolation problem* is to find: (1) the knot vector $U = [u_0, \ldots, u_m]$, (2) the parameter value t_i for each $D_i, 0 \leq i \leq n$, and (3) the control points P_i (also called the *de Boor points*) such that the resulting B-spline curve

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) \cdot P_i$$

has the property $C(t_i) = D_i$ for $0 \leq i \leq n$.

In more common methods of interpolation, such as cubic splines, data points and knots are often used as terms interchangeably. In B-spline interpolation, the x-values of our data points will inform our choice of knots and we will not consider the y-values until solving a corresponding linear system for the unknown control points. In the intermediary, we will have to determine a single parameter that corresponds to each data point such that the interpolated curve evaluated at that parameter is equal to the data point.

3.2 Basis function and curve properties

The B-spline basis functions $N_{i,p}(u)$ defined above have the following properties [7], [8].

1. $N_{i,p}(u) > 0$ for $u_i < u < u_{i+p+1}$
2. $N_{i,p}(u) = 0$ for $u_0 \le u \le u_i$ and $u_{i+p+1} \le u \le u_m$
3. $\sum_{i=0}^{n} N_{i,p}(u) = 1$ for $u_p \le u \le u_{n+1}$
4. $N_{i,p}(u)$ has C^{p-1} continuity at each knot u_i
5. $N_{i,p}(u)$ has support on the interval $[u_i, u_{i+p+1}]$

de Boor enumerates twelve properties of the B-spline curve function $C(u)$ defined in Definition 3.1.2. Some of these properties pertain only to the qualities of the basis functions, which Hoschek and Lasser [8] summarize, while a majority of them are properties of the spline curve itself. The most discussed property of the spline curve is its *convex hull property* [7]. Following from Properties 1, 2, and 3 above, we are able to say that the spline curve is a convex combination of its control points and the basis functions. That is,

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) \cdot P_i, \quad \text{with } N_{i,p}(u) \ge 0 \text{ and } \sum_{i=0}^{n} N_{i,p}(u) = 1 \quad \text{for } u_p \le u \le u_{n+1}.$$

If the control points P_i are connected with linear functions, these segments create a *control polygon* [1]. The convex hull property implies that the B-spline curve lies below the convex portions of the control polygon and above the concave portions. Another implication of this property, then, is that the B-spline curve is bounded. In two dimensions, where control points are ordered pairs (x_i, y_i), the B-spline curve is bounded above by

$$\max_{0 \leq i \leq n} y_i$$

and below by

$$\min_{0 \leq i \leq n} y_i.$$

Figure 1 [9] depicts a B-spline curve with its associated control polygon. The first four control points form a convex curve underneath which the spline curve sits. The last four control points form a concave curve above which the spline curve sits.

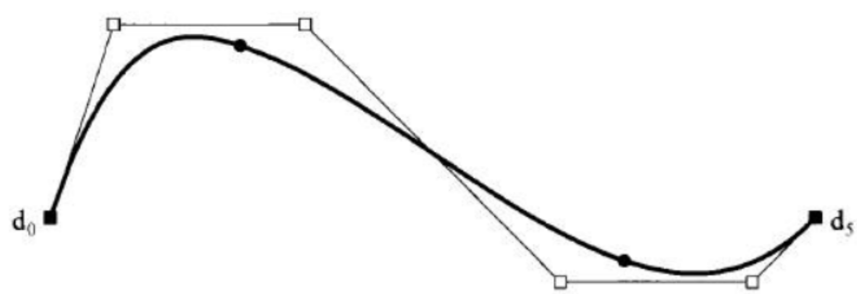

Fig. 1.: A B-spline curve (bold) with its control polygon (non-bold). Knots are circles and control points are squares.

Because of the convex hull property of the B-spline curve, we are able to estimate the visual shape of the B-spline curve given the control polygon.

[1]The term *polygon* is misleading in the context where we wish to interpolate points that do not create a closed curve, but it is the standard terminology in spline curve literature, so we will continue to use it here.

3.3 de Boor's Algorithm

de Boor's Algorithm [7] provides a method for evaluating the B-spline curve $C(u)$ at any given point within the bounds of the knot sequence. Considering that $C(u)$ is constructed to interpolate its knots, the use of de Boor's algorithm for interpolation purposes is equivalent to inserting a knot within the knot vector. The algorithm is as follows:

Given control points P_i with $0 \leq i \leq n$, knots u_j for $0 \leq j \leq m$, degree p, and value of interest u:

1. Define k by $u \in [u_k, u_{k+1})$
2. If $u \neq u_k$, let $h = p$ and $s = 0$. If $u = u_k$, let s be the multiplicity of u_k in the knot vector and let $h = p - s$.
3. Set $\widehat{P}_{i,0} = P_i$ for $k - p \leq i \leq k - s$
4. $C(u) = \widehat{P}_{k-s,h}$, where we define, for $r = 1, \ldots, h$ and $k - p + r \leq i \leq k - s$,

$$\widehat{P}_{i,r} = (1 - a_{i,r})\widehat{P}_{i-1,r-1} + a_{i,r}\widehat{P}_{i,r-1} \tag{3.1}$$

$$a_{i,r} = \frac{u - u_i}{u_{i+p-r+1} - u_i} \tag{3.2}$$

3.3.1 A Brief Example

Let $U = [0, 0, 3/5, 6/5, 9/5, 12/5, 3, 3]$ be the knot vector, $p = 3$, and our control points be $P_0 = 0, P_1 = 1/2, P_2 = 5/2, P_3 = 3$. We want to find $C(7/5)$. The steps are as follows, with the numbering corresponding to the algorithm outlined above.

1. $7/5 \in [6/5, 9/5) = [u_3, u_4)$, therefore $k = 3$.
2. $7/5 \neq 6/5$, so we let $h = 3$ and $s = 0$.
3. Let $\widehat{P}_{3,0}, \widehat{P}_{2,0}, \widehat{P}_{1,0}, \widehat{P}_{0,0}$ be P_3, P_2, P_1, P_0, respectively.

The coefficients $a_{j,r}$ and equations $\widehat{P}_{j,r}$ are given in the following tables

$a_{j,r}$	Equation	Value
$a_{1,1}$	$\frac{u-u_1}{u_4-u_1}$	7/9
$a_{2,1}$	$\frac{u-u_2}{u_5-u_2}$	4/9
$a_{3,1}$	$\frac{u-u_3}{u_6-u_3}$	1/9
$a_{2,2}$	$\frac{u-u_2}{u_4-u_2}$	2/3
$a_{3,2}$	$\frac{u-u_3}{u_5-u_3}$	2/3
$a_{3,3}$	$\frac{u-u_3}{u_4-u_3}$	1/3

$\widehat{P}_{j,r}$	Equation	Value
$\widehat{P}_{0,0}$	P_0	0
$\widehat{P}_{1,0}$	P_1	1/2
$\widehat{P}_{2,0}$	P_2	5/2
$\widehat{P}_{3,0}$	P_3	3
$\widehat{P}_{1,1}$	$(1-a_{1,1})\widehat{P}_{0,0} + a_{1,1}\widehat{P}_{1,0}$	7/18
$\widehat{P}_{2,1}$	$(1-a_{2,1})\widehat{P}_{1,0} + a_{2,1}\widehat{P}_{2,0}$	25/18
$\widehat{P}_{3,1}$	$(1-a_{3,1})\widehat{P}_{2,0} + a_{3,1}\widehat{P}_{3,0}$	23/9
$\widehat{P}_{2,2}$	$(1-a_{2,2})\widehat{P}_{1,1} + a_{2,2}\widehat{P}_{2,1}$	57/54
$\widehat{P}_{3,2}$	$(1-a_{3,2})\widehat{P}_{2,1} + a_{3,2}\widehat{P}_{3,1}$	13/6
$\widehat{P}_{3,3}$	$(1-a_{3,3})\widehat{P}_{2,2} + a_{3,3}\widehat{P}_{3,2}$	231/162

Table 1.: The coefficients and $\widehat{P}$ calculations for the de Boor Algorithm example.

Therefore, $C(7/5) = 231/162$.

3.4 The Universal Method

While the de Boor Algorithm is able to interpolate without needing to know the basis functions explicitly, it does require knowledge of control points. The choice of control points is not always intuitive, especially if the shape of the data is not already known. For this reason, in 1999, South Korean mathematician Choong-Gyoo Lim proposed the *Universal Method* for determining the interpolating B-spline curve given a set of data points, and it is this method [11] that we will make use of in the rest of this paper.

The B-spline curve $C(u)$ is a parametric curve. In order to interpolate a set of data points, we must have a means to associate each data point with a parameter passed into the spline curve. The process of making this association is called *parameterization*, and there are countless methods to achieve this. Typically, one would first parameterize and then generate the knot vector. Lim proposes, however, that rather than letting the choice of data point parameters determine the knot vector, we should let equally spaced knots determine the parameters. The set up of Lim's Universal Method for parameter and control point selection, and the solving for the corresponding B-spline curve, is described below.

Given $n+1$ data points $D_0 = (x_0, y_0), D_1 = (x_1, y_1), \ldots, D_n = (x_n, y_n)$ (ordered such that $x_i < x_{i+1}$), and desiring a B-spline interpolating curve of degree p, we require knots $U = [u_0, u_1, \ldots, u_m]$. From Definition 3.1.1, the recursion formula which defines the B-spline basis functions requires that

$$N_{0,p}(u) = \frac{u - u_0}{u_p - u_0} N_{0,p-1}(u) + \frac{u_{p+1} - u}{u_{p+1} - u_1} N_{1,p-1}(u)$$
$$N_{n,p}(u) = \frac{u - u_n}{u_{n+p} - u_n} N_{n,p-1}(u) + \frac{u_{n+p+1} - u}{u_{n+p+1} - u_{n+1}} N_{n+1,p-1}(u).$$

Since we will assign one data point parameter per data point D_i, and these parameters will be defined by the maximum of each of the p-th degree basis functions respectively, we must have $n+1$ p-th degree basis functions. Therefore $N_{n,p}(u)$ is the last iteration of the p-th degree basis functions; wo we only require knots u_0 to u_{n+p+1}. We will define $m = n + p + 1$.

We will set the first $p-1$ knots to be equal to x_0 and the last $p-1$ knots to be equal to x_n. The remaining $p+1$ knots will divide the range of x_0 to x_n evenly. The first knot span will be $[u_0, u_1)$ and the last will be $[u_{m-1}, u_m)$. For example, if we have $x_0 = 1$, $x_n = x_6 = 10$, and $p = 3$, then we will require $m = 3 + 6 + 1 = 10$ — eleven knots. We will set the first and last $p-1$ knots as $u_0 = u_1 = 1$ and $u_{10} = u_9 = 10$, with the first and last knot spans as $[1, 1)$ and $[10, 10)$, respectively. Since we still need to allot u_2 to u_8, we must subdivide the domain evenly. Figure 2 shows the knots

u_i along a line, where knots u_0, u_1, u_9, and u_{10} have multiplicity 2. Between the end knots, there are eight knot spans, and so we divide the range of the knots by eight. Therefore, each knot is 9/8 apart.

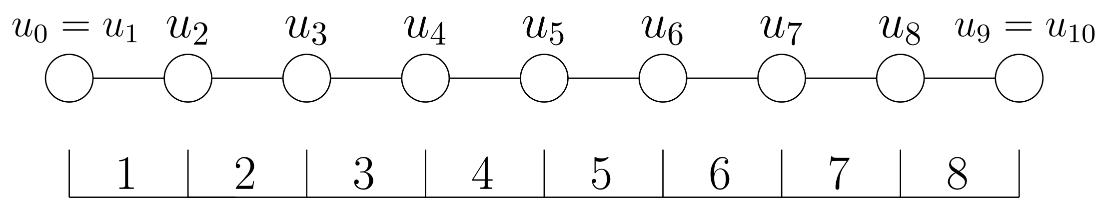

Fig. 2.: Knots along a line, with the inner knot spans numbered.

We now have our knot vector $U = [u_0, u_1, u_2, \ldots, u_m]$ and the associated knot spans $[u_0, u_1)$, $[u_1, u_2)$, et cetera. Our next step is to determine our basis functions. Using Definition 3.1.1, we will find the $n+1$ p-th degree basis functions. Note that for $m+1$ knots, we will have m knot spans and m 0-degree basis functions. Because we have set the first $p-1$ knots to be all equal to the same value, and likewise for the last $p-1$ knots, the first and last $p-1$ knot spans will result in $N_{i,0}(u) = 0$ always, since there is never a u such that u is both greater than or equal to, and also strictly less than, one value. We will find, then, that in computing these basis functions towards creating our interpolating spline curve, we will only need to concern ourselves with those basis functions that do not utilize an $N_{i,p-1}(u)$ and $N_{i+1,p-1}(u)$ that are always zero at the same time.

After we have found all $N_{i,p}(u)$ functions, we will choose our data point parameters t_i to be the x-value at which the maximum of each $N_{i,p}(u)$ occurs. Since we have chosen one parameter to correspond to one data point, in order to interpolate the data points exactly we must have

$$D_k = C(t_k) = \sum_{i=0}^{n} N_{i,p}(t_k) \cdot P_i$$

. After setting these data point parameters, we are left with only one set of unknowns, which are the control points P_i.

We can find these most efficiently by solving a corresponding linear system. We can arrange the basis functions into an $(n+1) \times (n+1)$ matrix N where, for our data point parameters $t_0, \ldots, t_n$,

$$N = \begin{bmatrix} N_{0,p}(t_0) & N_{1,p}(t_0) & \cdots & N_{n,p}(t_0) \\ N_{0,p}(t_1) & N_{1,p}(t_1) & \cdots & N_{n,p}(t_1) \\ \vdots & \vdots & \vdots & \vdots \\ N_{0,p}(t_n) & N_{1,p}(t_n) & \cdots & N_{n,p}(t_n) \end{bmatrix}$$

We also define a matrix D with our data points.

$$D = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

We can now solve for the matrix P, which will have the same dimensions as D. The corresponding equation is

$$D = NP$$

and so

$$P = N^{-1}D$$

Upon solving this system, we have our control points and thus the interpolating B-spline curve.

3.5 Example

Suppose we wish to interpolate the data points $D_0 = (0,1), D_1 = (1.5, 0.5), D_2 = (2,3), D_3 = (3, 0.25)$ with a B-spline curve of degree 3 using Lim's *Universal Method*. We have four data points, so $n = 3$ and our degree $p = 3$. The number of $m+1$ knots is governed by the equation $m = p + n + 1$, so we have $m = 3 + 3 + 1 = 7$, therefore we need eight knots. We set the first $p - 1$ knots, u_0, u_1, to be equal to 0 and the last $p - 1$ knots, u_6, u_7 to be equal to 3. We then choose the remaining $p + 1$ knots to divide the range of x values evenly. Figure 3 shows the inner knot intervals on a number line.

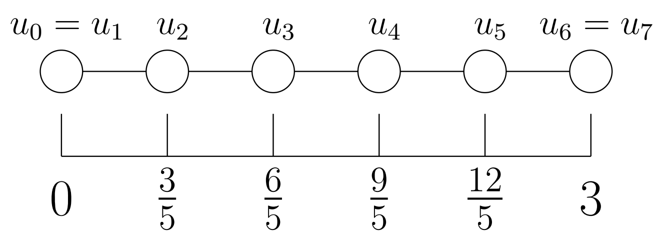

Fig. 3.: Knot intervals on a number line for eight equidistant knots between 0 and 3.

We will write the knot vector as

$$U = \begin{bmatrix} 0 & 0 & \frac{3}{5} & \frac{6}{5} & \frac{9}{5} & \frac{12}{5} & 3 & 3 \end{bmatrix}.$$

Next, we must compute all the basis functions using the recursion formula in Definition 3.1.1. Separating the knot vector into its knot spans gives $[u_0, u_1) = [0, 0), [u_1, u_2) = [0, 3/5), [u_2, u_3) = [3/5, 6/5), [u_3, u_4) = [6/5, 9/5), [u_4, u_5) = [9/5, 12/5), [u_5, u_6) = [12/5, 3)$, and $[u_6, u_7) = [3, 3)$. The basis functions on the relevant intervals are contained in Tables 1-4. For the sake of brevity beyond the 0-th degree, where a basis function's equation is not given for an interval, it is assumed to be 0.

Table 2.: B-spline example Degree 0 basis functions.

Degree 0		
Basis Function	Equation	Interval
$N_{0,0}(u)$	0	everywhere
$N_{1,0}(u)$	1	$[0, 3/5)$
	0	elsewhere
$N_{2,0}(u)$	1	$[3/5, 6/5)$
	0	elsewhere
$N_{3,0}(u)$	1	$[6/5, 9/5)$
	0	elsewhere
$N_{4,0}(u)$	1	$[9/5, 12/5)$
	0	elsewhere
$N_{5,0}(u)$	1	$[12/5, 3)$
	0	elsewhere
$N_{6,0}(u)$	0	everywhere

Table 3.: B-spline example Degree 1 basis functions.

Degree 1		
Basis Function	Equation	Interval
$N_{0,1}(u)$	$1 - \frac{5}{3}u$	$[0, 3/5)$
$N_{1,1}(u)$	$\frac{5}{3}u$	$[0, 3/5)$
	$2 - \frac{5}{3}u$	$[3/5, 6/5)$
$N_{2,1}(u)$	$\frac{5}{3}u - 1$	$[3/5, 6/5)$
	$3 - \frac{5}{3}u$	$[6/5, 9/5)$
$N_{3,1}(u)$	$\frac{5}{3}u - 2$	$[6/5, 9/5)$
	$4 - \frac{5}{3}u$	$[9/5, 12/5)$
$N_{4,1}(u)$	$\frac{5}{3}u - 3$	$[9/5, 12/5)$
	$5 - \frac{5}{3}u$	$[12/5, 3)$
$N_{5,1}(u)$	$\frac{5}{3}u - 4$	$[12/5, 3)$

Table 4.: B-spline Degree 2 basis functions.

Degree 2		
Basis Function	Equation	Interval
$N_{0,2}(u)$	$-\frac{25}{9}u^2 + \frac{35}{18}u$	$[0, 3/5)$
	$\frac{25}{18}u^2 - \frac{10}{3}u + 2$	$[3/5, 6/5)$
$N_{1,2}(u)$	$\frac{25}{18}u^2$	$[0, 3/5)$
	$-\frac{25}{9}u^2 + 5u - \frac{3}{2}$	$[3/5, 6/5)$
	$\frac{25}{18}u^2 - 5u + \frac{9}{2}$	$[6/5, 9/5)$
$N_{2,2}(u)$	$\frac{25}{18}u^2 - \frac{5}{3}u + \frac{1}{2}$	$[3/5, 6/5)$
	$-\frac{25}{9}u^2 + \frac{25}{3}u - \frac{11}{2}$	$[6/5, 9/5)$
	$\frac{25}{18}u^2 - \frac{20}{3}u + 8$	$[9/5, 12/5)$
$N_{3,2}(u)$	$\frac{25}{18}u^2 - \frac{10}{3}u + 2$	$[6/5, 9/5)$
	$-\frac{25}{9}u^2 + \frac{35}{3}u - \frac{23}{2}$	$[9/5, 12/5)$
	$\frac{25}{18}u^2 - \frac{25}{3}u + \frac{25}{2}$	$[12/5, 3)$
$N_{4,2}(u)$	$\frac{25}{18}u^2 - 5u + \frac{9}{2}$	$[9/5, 12/5)$
	$-\frac{25}{6}u^2 + \frac{65}{3}u - \frac{55}{2}$	$[12/5, 3)$

Table 5.: B-spline example Degree 3 basis functions.

Degree 3		
Basis Function	Equation	Interval
$N_{0,3}(u)$	$\frac{5}{6}u\left(\frac{10}{3}u - \frac{75}{18}u\right) + \left(1 - \frac{5}{9}u\right)\frac{25}{18}u^2$	$[0, 3/5)$
	$\frac{5}{6}u\left(\frac{25}{18}u^2 - \frac{10}{3}u + 2\right) + \left(1 - \frac{5}{9}u\right)\left(-\frac{25}{9}u^2 - 5u - \frac{3}{2}\right)$	$[3/5, 6/5)$
	$\left(1 - \frac{5}{9}u\right)\left(\frac{25}{18}u^2 - 5u + \frac{9}{2}\right)$	$[6/5, 9/5)$
$N_{1,3}(u)$	$\frac{5}{9}u\left(\frac{25}{18}u^2\right)$	$[0, 3/5)$
	$\frac{5}{9}u\left(-\frac{25}{9}u^2 + 5u - \frac{3}{2}\right) + \left(\frac{4}{3} - \frac{5}{9}u\right)\left(\frac{25}{18}u^2 - \frac{5}{3}u + \frac{1}{2}\right)$	$[3/5, 6/5)$
	$\frac{5}{9}u\left(\frac{25}{18}u^2 - 5u + \frac{9}{2}\right) + \left(\frac{4}{3} - \frac{5}{9}u\right)\left(-\frac{25}{9}u^2 + \frac{25}{3} - \frac{11}{2}\right)$	$[6/5, 9/5)$
	$\left(\frac{4}{3} - \frac{5}{9}u\right)\left(\frac{25}{18}u^2 - \frac{20}{3}u + 8\right)$	$[9/5, 12/5)$
$N_{2,3}(u)$	$\left(\frac{5}{9}u - \frac{1}{3}\right)\left(\frac{25}{18}u^2 - \frac{5}{3}u + \frac{1}{2}\right)$	$[3/5, 6/5)$
	$\left(\frac{5}{9}u - \frac{1}{3}\right)\left(-\frac{25}{9}u^2 + \frac{25}{3}u - \frac{11}{2}\right) + \left(\frac{5}{3} - \frac{5}{9}u\right)\left(\frac{25}{18}u^2 - \frac{10}{3}u + 2\right)$	$[6/5, 9/5)$
	$\left(\frac{5}{9}u - \frac{1}{3}\right)\left(\frac{25}{18}u^2 - \frac{20}{3}u + 8\right) + \left(\frac{5}{3} - \frac{5}{9}u\right)\left(-\frac{25}{9}u^2 + \frac{35}{3} - \frac{23}{2}\right)$	$[9/5, 12/5)$
	$\left(\frac{5}{3} - \frac{5}{9}u\right)\left(\frac{25}{18}u^2 - \frac{25}{3}u + \frac{25}{2}\right)$	$[12/5, 3)$
$N_{3,3}(u)$	$\left(\frac{5}{9}u - \frac{2}{3}\right)\left(\frac{25}{18}u^2 - \frac{10}{3}u + 2\right)$	$[6/5, 9/5)$
	$\left(\frac{5}{9}u - \frac{2}{3}\right)\left(-\frac{25}{9}u^2 + \frac{35}{3}u - \frac{23}{2}\right) + \left(\frac{5}{2} - \frac{5}{6}u\right)\left(\frac{25}{18}u^2 - 5u + \frac{9}{2}\right)$	$[9/5, 12/5)$
	$\left(\frac{5}{9}u - \frac{2}{3}\right)\left(\frac{25}{18}u^2 - \frac{25}{3}u + \frac{25}{2}\right) + \left(\frac{5}{2} - \frac{5}{6}u\right)\left(-\frac{25}{6}u^2 + \frac{65}{3} - \frac{5}{2}\right)$	$[12/5, 3)$

Though we must calculate the degree 0, 1, and 2 basis functions in order to obtain the degree 3 functions, we are only concerned with the degree 3 functions in the calculation of the data point parameters. We choose these parameters to be the u-value of the maximum of each degree 3 basis function $N_{i,3}(u)$. Below is a graph of each basis function with the coordinates of the maximum value labeled.

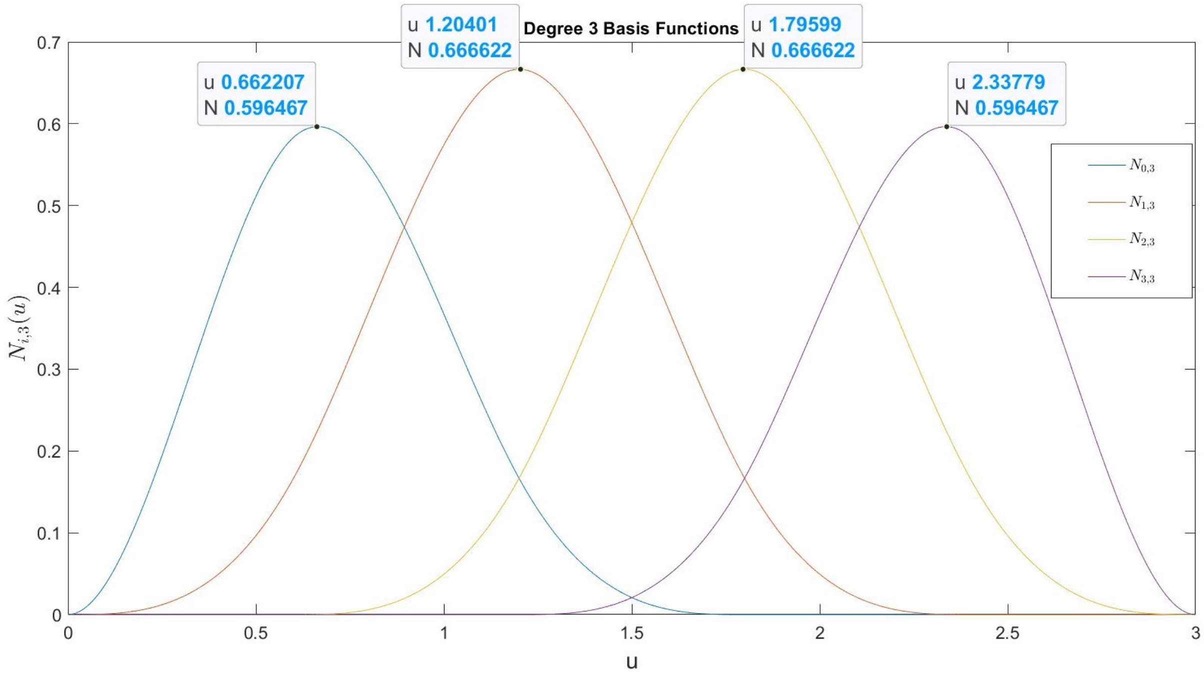

Fig. 4.: Each basis function with their respective maximum values.

The data point parameters will be $t_0 = 0.66207$, $t_1 = 1.20401$, $t_2 = 1.79599$, and $t_3 = 2.33779$, corresponding to the data points D_0, D_1, D_2, and D_3, respectively. The matrix N, which is the matrix of each degree 3 basis function evaluated at each parameter t_i is

$$N = \begin{bmatrix} 0.59647 & 0.22332 & 1.85746 \times 10^{-4} & 0 \\ 0.16334 & 0.66662 & 0.17003 & 4.98799 \times 10^{-8} \\ 4.98799 \times 10^{-8} & 0.17003 & 0.66662 & 0.16334 \\ 0 & 1.85746 \times 10^{-4} & 0.22332 & 0.59647 \end{bmatrix}.$$

The symmetry of this matrix is unique only to this particular problem and is not a property of all matrices N of this form. The system we are then presented with to solve for the control points is

$$P = \begin{bmatrix} 0.59647 & 0.22332 & 1.85746 \times 10^{-4} & 0 \\ 0.16334 & 0.66662 & 0.17003 & 4.98799 \times 10^{-8} \\ 4.98799 \times 10^{-8} & 0.17003 & 0.66662 & 0.16334 \\ 0 & 1.85746 \times 10^{-4} & 0.22332 & 0.59647 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 \\ 1.5 & 0.5 \\ 2 & 3 \\ 3 & 0.25 \end{bmatrix}$$

Determination of N^{-1} and the matrix multiplication problem $N^{-1}D$ are subject to the expected constraints of computational complexity and efficiency. There is the possibility that N may be singular, which leads to an inability to find the inverse. If the inverse of N cannot be found, then $N^{-1}D$ cannot be computed and thus we cannot find the control points. As well, the size of N presents restraints to computational time both in finding the inverse, if there is one, and in computing the product $N^{-1}D$. The larger N is, the more computationally taxing it is to find the inverse and multiply on either side of N. For the purposes of this exposition, we will exploit built-in solvers in MATLAB [12] As such, P is calculated to be

$$P = \begin{bmatrix} -0.78527 & 2.07562 \\ 2.09623 & -1.07016 \\ 1.35783 & 5.14225 \\ 4.52058 & -1.50584 \end{bmatrix}.$$

The i-th row of P corresponds to the i-th control point P_i. Since our data points D_i are ordered pairs, the control points are also. Thus $P_0 = (-0.78527, 2.07562), P_1 = (2.09623, -1.07016)$, et cetera. As mentioned in section 3.4, $C(u)$ is a parameterized function, and so we express any given coordinate $(x(u), y(u))$ on the B-spline curve by

$$x(u) = -0.78527 \cdot N_{0,3}(u) + 2.09623 \cdot N_{1,3}(u) + 1.35783 \cdot N_{2,3}(u) + 4.52058 \cdot N_{3,3}(u)$$

$$y(u) = 2.07562 \cdot N_{0,1}(u) - 1.07016 \cdot N_{1,3}(u) + 5.14225 \cdot N_{2,3}(u) - 1.50584 \cdot N_{3,3}(u)$$

for some $0 \leq u \leq 3$. The graph of the B-spline curve and the associated control polygon is depicted in Figure 5.

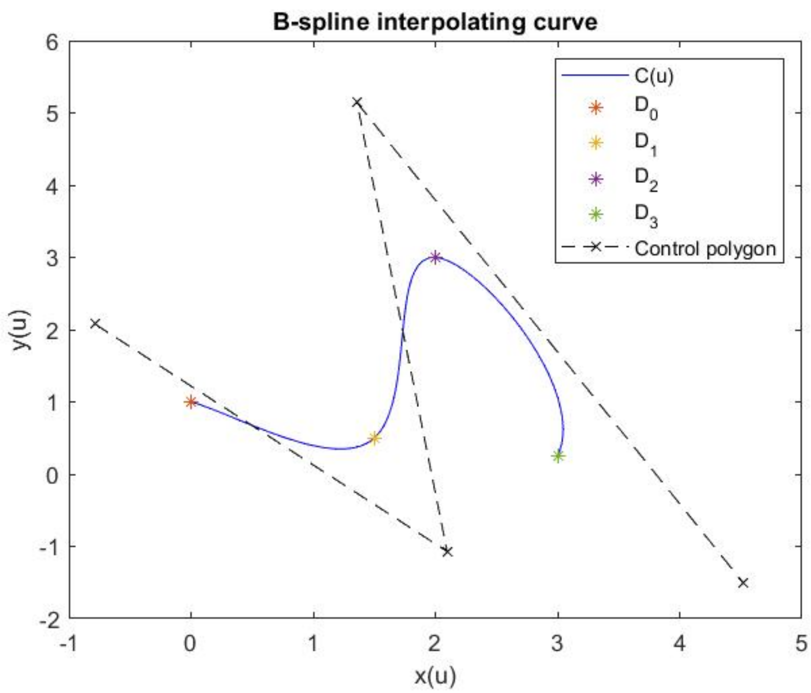

Fig. 5.: The curve $C(u)$ in blue, with the interpolated points as asterisks. The dashed line is the control polygon with the control points marked as an X.

CHAPTER 4

PENALIZATION

4.1 Motivation and Formulation

In Figure 5, we observe between D_2 and D_3 a bulging of the interpolating B-spline curve $C(u)$. The curvature for a parametric curve is defined as

$$\kappa = \frac{|x'y'' - y'x''|}{((x')^2 + (y')^2)^{3/2}}.$$

The derivatives of $x(u)$ and $y(u)$ are calculated with respect to the parameter u. Calculating curvature aims to quantify how much a curve deviates from a straight line. As such, the curvature of a straight line is 0. For the spline curve depicted in Figure 5, the maximum curvature κ between D_2 and D_3 is calculated to be approximately 5.73 at $u \approx 2.216$. Without knowledge of data between D_2 and D_3, we have no reason to assume that there is an increased curvature between those data points. We may then wish to reduce this curvature to facilitate a smoother interpolation. Assessing a penalty on the control points is one way to achieve a smoother curve.

Much of the literature that discusses penalization surveyed for this thesis is written in the context of using splines to *approximate* a function given data, rather than interpolating it. As such, the formulations of the assessed penalty in the literature include a term aiming to account for closeness to a known solution by minimization of a least squares objective. This term is not relevant to the discussion of this thesis as we are considering spline curves that interpolate the data exactly, regardless of whether there is an exact function the data is supposed to fit. We will not neglect this approximation term altogether; it will simply evaluate to 0 and is a moot point in the ensuing discussion. Subsequently, we will discuss penalization as it relates to reducing curvature, not as it relates to closeness of fit.

Nonetheless we aim to minimize

$$||D - NP||^2 + \lambda \cdot \mathrm{PEN}(P) \tag{4.1}$$

where D, N, and P are as defined in Chapter 3 [13]. PEN(P) is a placeholder for the chosen quantification of roughness to be discussed in subsequent sections. $\lambda \in \mathbb{R}$ is a parameter allowing for fine-tuning of the roughness quantity, and the norm $||\cdot||$ is the Euclidean norm.

It should be noted that since we have constructed N and P such that $D = NP$, the first term will be equal to zero for an interpolation problem. Therefore, we are truly only seeking to minimize $\lambda \cdot \mathrm{PEN}(P)$.

4.2 The O'Sullivan Penalty

One way we may try to reduce this curvature is by minimizing the second derivative of the B-spline curve. We would need to minimize the coefficients on the terms containing $u^{p-2}, u^{p-3}, \ldots, u$. Unfortunately, since the basis functions of the B-spline curve are constructed with a knot vector that facilitates interpolation of the data points, we cannot alter these coefficients, or we will sacrifice the interpolation. Another option is to minimize the integral of the second derivative. The higher the curvature of the B-spline curve, the higher the area underneath it.

It is important to remember that the penalty is a function of the control points. As such, we minimize the penalty by altering our choice in control points. The first term of (4.1) above is a least-squares problem. We can also apply a least-squares approach to the second derivative. Using the square of the second derivative as a roughness quantity is introduced first by Reinsch [14], but its application to B-splines is attributed to O'Sullivan in 1986 [3], [5]. The derivative penalty applied to a B-spline curve, sometimes called an *O-spline*[13], is

$$\lambda \int_{u_0}^{u_n} \{C''(u)\}^2 \, du = \lambda \int_{u_0}^{u_n} \left\{ \sum_{j=0}^{n} P_j \cdot N''_{j,p}(u) \right\}^2 \, du \tag{4.2}$$

4.3 The Standard P-Spline

Another approach to a penalty on a B-spline is to minimize the difference between control points. Recall that we have defined our knot vector to specifically be uniform, meaning the knots of multiplicity 1 are evenly spaced. Using a different method for choosing the knots will inherently change how we choose the parameters that correspond to our data points. Other methods of knot vector generation (and by extension parameterization) include the *chordal*, *centripetal*, and *Foley* methods. Lim [11] argues that there is no overall advantage in these other knot vector generation choices over the uniform choice in interpolation problems. Part of the justification for this is that B-spline curves are *transformation invariant* for the same knot vector and parameterization. Lim defines this in a lemma, given here with adapted notation and terminology.

Lemma 1. Suppose that P_i and P_i^*, $0 \le i \le n$ are two lists of control points such that $P_i M = P_i^*$, where M is a transformation matrix. Define the B-spline curves

$$C(u) = \sum_{i=0}^{n} P_i N_{i,p}(u) \quad \text{and} \quad C^*(u) = \sum_{i=0}^{n} P_i^* N_{i,p}^*(u)$$

for degree p. If the same knot vector $U = (u_0, u_1, \ldots, u_m)$ is used both for $N_{i,p}(u)$ and for $N_{i,p}^*(u)$, then the B-spline curve is *transformation invariant*, i.e., $C(u)M = C^*(u)$ for all $u \in [u_0, u_n]$.

When Li and Cao [13] discuss penalties on B-splines, they make a necessary delineation for a penalty on those with uniform knots and with non-uniform knots. This is because in an approximation problem, what we will define as the *standard* *difference penalty* fails to measure roughness in the valleys of the approximating spline. Li and Cao subsequently call a uniform B-spline with a difference penalty a *standard P-spline* and a non-uniform B-spline with a difference penalty a *generalized P-spline*. Owing to our knot vector formulation in Chapter 3, we will consider the standard P-spline (SPS).

We first define the *order-1 difference* ΔP_i, sometimes called the *first ordered difference*,

$$\Delta P_i = P_i - P_{i-1}$$

for our control points P_i. The *order-2 difference* is defined by applying the operator Δ to the order-1 difference. Thus

$$\begin{aligned}\Delta^2 P_i &= \Delta\Delta P_i \\ &= \Delta(P_i - P_{i-1}) \\ &= (P_i - P_{i-1}) - (P_{i-1} - P_{i-2}) \\ &= P_i - 2P_{i-1} + P_{i-2}\end{aligned}$$

Similarly, the *order-3 difference* $\Delta^3 P_i$ is $P_i - 3P_{i-1} + 3P_{i-2} + P_{i-3}$. We can define the *order-k difference* $\Delta^k P_i$ as applying Δ a k number of times [15] It is important to note that since the k-th ordered difference requires control points $P_i, P_{i-1}, \ldots, P_{i-k}$ and we must have that $i - k \geq 0$ so that there are no negative indices, then $i \geq k$. We can now define the *standard difference penalty* as

$$\lambda \sum_{i=k}^{p} (\Delta^k P_i)^2$$

Since Δ is a linear map, Δ can be expressed in a matrix formulation. Let $\mathcal{D}$ be Δ

expressed a matrix. The order-1 difference matrix $\mathcal{D}$ has entries d_{ij} such that

$$d_{ij} = -1, \qquad \text{when } i = j$$

$$d_{ij} = 1, \qquad \text{when } i + 1 = j$$

$$d_{ij} = 0, \qquad \text{otherwise.}$$

The order-2 difference matrix $\mathcal{D}_2$ with entries d_{ij} is

$$d_{ij} = 1, \qquad \text{when } i = j \text{ and } i + 2 = j$$

$$d_{ij} = -2, \qquad \text{when } i + 1 = j$$

$$d_{ij} = 0, \qquad \text{otherwise}$$

and the order-3 difference matrix $\mathcal{D}_3$ with entries d_{ij}

$$d_{ij} = 1, \qquad \text{when } i = j \text{ and } i + 3 = j$$

$$d_{ij} = -3, \qquad \text{when } i + 1 = j$$

$$d_{ij} = 3, \qquad \text{when } i + 2 = j$$

$$d_{ij} = 0, \qquad \text{otherwise.}$$

The size of $\mathcal{D}_k$ is $(p - k + 1) \times (n + 1)$. The number of columns comes from having $n + 1$ control points since we have $n + 1$ data points. The number of rows comes from the fact that in the summation expression of the standard difference penalty, we sum from k to p, meaning there are $p - k + 1$ terms of the summation. To then square each term of the sum is, in matrix formulation, equivalent to calculating the square of the Euclidean norm of $\mathcal{D}_k$ applied to the matrix of control points. Therefore, we can rewrite the standard difference penalty as

$$\lambda ||\mathcal{D}_k P||^2.$$

Note that the number of columns of P does not matter. In section 3.5, we calculated P to be a 4×2 matrix because we were desiring a curve in the x, y-plane. With control points as ordered pairs, the product $\mathcal{D}_k P$ will be a matrix of size $p - k + 1 \times 2$.

4.4 Brief comments on choosing λ

There are many proposed methods for choosing λ optimally. The two most common of which are by minimizing generalized cross-validation (GCV) score or maximizing restricted likelihood (REML) [3], [16]–[18]. While these methods are useful for many large scale approximation problems, especially in approximation problems where large intervals are missing data, it is not necessary to find a precise λ, especially because there is not a significant change in the B-spline curve when λ is changed by 10% or less [15]. In fact, Li and Cao [16] aim only to find an optimal interval for λ rather than a specific value. Eilers and Marx discuss choosing λ by minimizing cross-validation score (CV). They define cross-validation score as

$$CV = \sqrt{\sum_{i=0}^{n} \frac{(D_i - \hat{\mu}_i)^2}{n(1 - H_{i+1,i+1})^2}}$$

where D_i, $0 \leq i \leq n$ are the $n+1$ data points being interpolated. We define $\hat{\mu}_i = HD_i$, where $H = N(N^T N + \mathcal{D}_k^T \mathcal{D}_k)^{-1} N^T$ is an $(n+1) \times (n+1)$ matrix called the *hat matrix*. The derivation and importance of this matrix is discussed in [3], [15], [16], [18], and [19].

Upon determining the optimal λ, we find the ideal control points $\widetilde{P}$ by the equation $\widetilde{P} = (N^T N + \lambda \mathcal{D}_k^T \mathcal{D}_k)^{-1} N^T D$ [15].

4.5 Application

With the publication of their book in 2021, Eilers and Marx published an *R* package, JOPS, in the Comprehensive R Archive Network (CRAN) [20], [21]. This package contains the method `psNormal` which calculates the control points and function values of a P-spline optimally *approximating* the data passed in as a required argument. Eilers and Marx [15] assert that any approximation with the `psNormal` function can become an interpolation by increasing the number of evenly spaced segments between the maximum and minimum x-values and plotting on a finer x grid. We have applied the `psNormal` function to the data defined in section 3.5. We chose 1000 evenly spaced segments, B-spline degree equal to 3, an order-2 difference penalty, $\lambda = 1$, and an x-grid of 100 values. Figure 6 is the resulting plot of the P-spline.

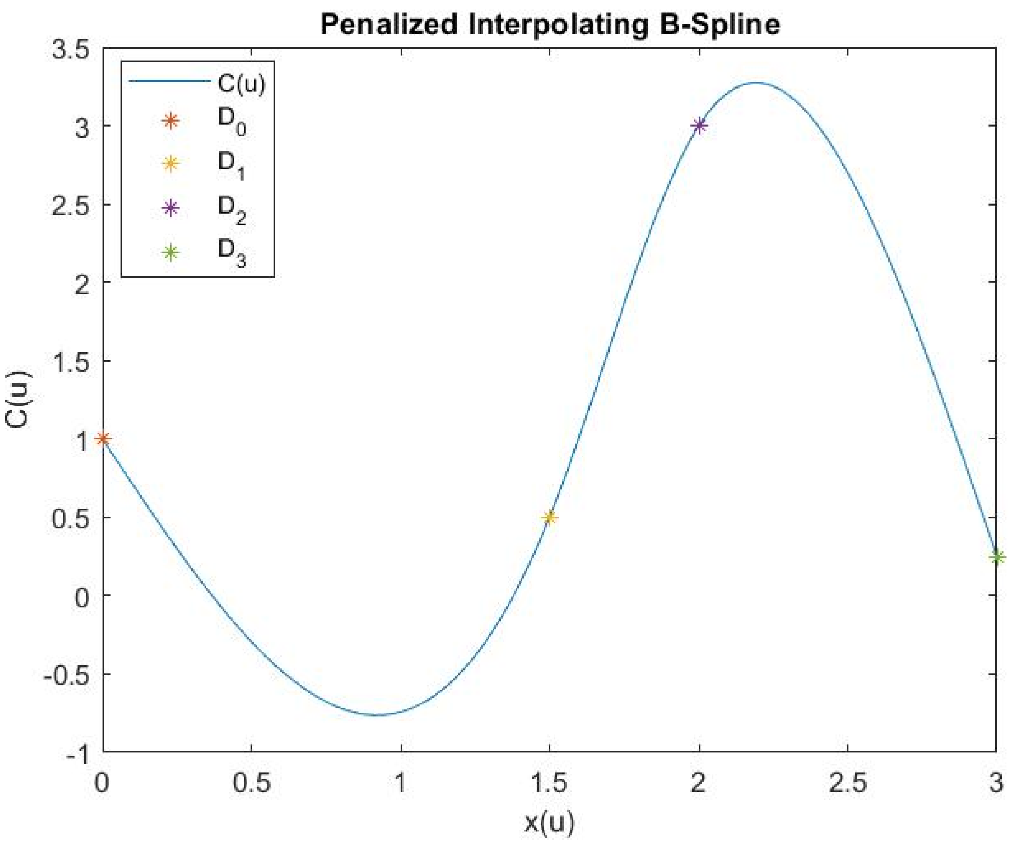

Fig. 6.: P-spline of section 3.5 data.

The `psNormal` function considers the roughness across the entire domain and so there is a trade-off: in order to reduce the overall roughness in intervals like that between D_2 and D_3, some of the roughness is "redistributed" to other portions of the curve, such as between D_0 and D_1. In this example, the "redistribution" results in a curve closely approximating a cubic function.

CHAPTER 5

APPLICATION AND CONCLUSION

In this final chapter, we will apply our discussions to cerebrovascular resistance data. First, we will apply a B-spline generated using Lim's *Universal Method*. Next, we will apply a P-spline with the aid of the JOPS R package. We will conclude with identifying the limitations to the study of the concepts in this thesis and posit ideas for future work.

5.1 Application to Cerebrovascular Resistance Data

We now apply our discussion of B-splines and penalization to a set of data points that was a piecewise linear optimized output from Ellwein et al [22]. This data, depicted in Figure 7, appear to resemble but not exactly replicate a decreasing sigmoid. We hypothesize that the oscillations are artifacts rather than exact representations of the underlying physiology. We are interested if the splines discussed here can give a continuous functional form that could resemble the dynamics and be used to predict resistance at different time points.

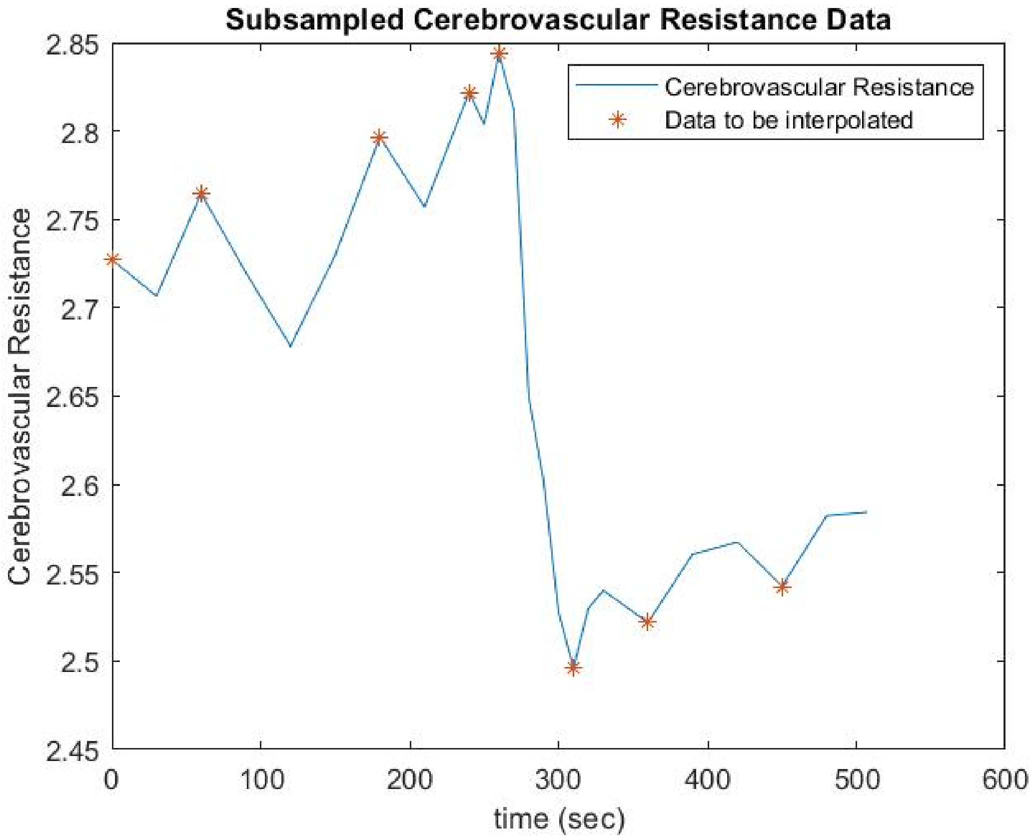

Fig. 7.: Cerebrovascular resistance data. Red asterisks are points to be interpolated with a B-spline and P-spline.

We first fit an interpolating B-spline curve to the data points marked by a red asterisk in Figure 7, which were chosen to delineate a smooth curve for this data.

That is, our data matrix D is defined as

$$D = \begin{bmatrix} 0 & 2.727 \\ 60 & 2.765 \\ 180 & 2.797 \\ 240 & 2.822 \\ 260 & 2.844 \\ 310 & 2.496 \\ 360 & 2.522 \\ 450 & 2.542 \end{bmatrix}$$

5.1.1 B-spline construction

Let us use $p = 3$. We choose this p to avoid the roughness of higher degree basis functions and to avoid the limitations of only using parabolas to try and fit the data. With $n = 7$, we must have $m = 11$, and therefore 12 knots. The first and last $p - 1$ knots will be set at 0 and 450, respectively. Thus we have $u_0 = u_1 = 0$ and $u_{10} = u_{11} = 450$. We allot the middle nine knots to evenly divide 450. Each knot is then $450/9 = 50$ units apart. In Appendix A, Table 5 exhibits the knot values, and the subsequent knot spans upon which the basis functions will be constructed.

With the knot spans, we can construct eight degree 3 basis functions, the computational details of which are also included in Appendix A. Eight basis functions yields eight maximum values to be assigned as parameters corresponding to our eight data points. Our matrix N is then of size 8×8. We solve the system $D = NP$ for the matrix P of control points, as is described in Section 3.4. We are then able to evaluate $C(u)$ at each value in a vector of equally spaced u values between 0 and 450 to get coordinates along the interpolating B-spline curve. Figure 8 shows the resulting plot of $C(u)$ with the interpolated values marked by red asterisks.

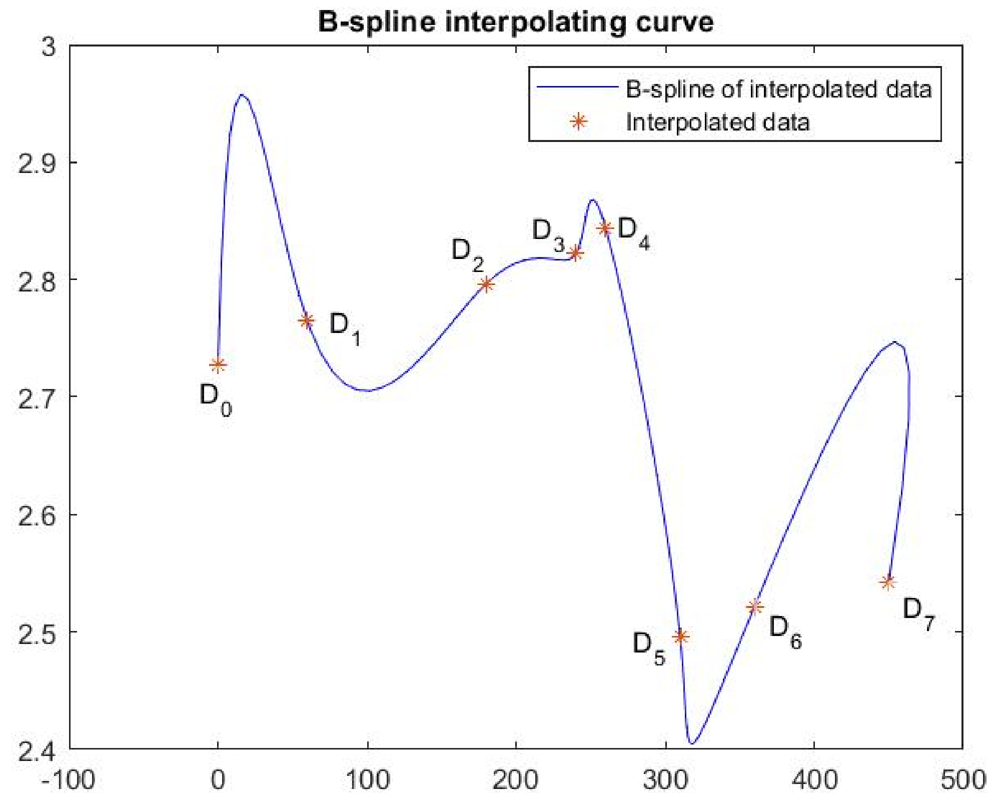

Fig. 8.: B-spline interpolating curve for cerebrovascular resistance data

Visually, the B-spline curve does well to interpolate D_2 to D_5 without drastic changes in curvature. The ends of the curve, however, do not fare as well in their smoothness, reaching sharp peaks between knots that seem an unjustified assumption given the general trend of the data. This is an example of when using a P-spline would be beneficial.

5.1.2 P-spline interpolation

Using the JOPS R package introduced in Chapter 4, we apply a 500 knot-span P-spline with an order-2 difference penalty on 100 evenly spaced points between 0 and 450. λ is set to 1, thereby eliminating any fine-tuning of the roughness of the curve and leaving the minimization of roughness entirely to the choice of control points. The graphical results are depicted in Figure 9.

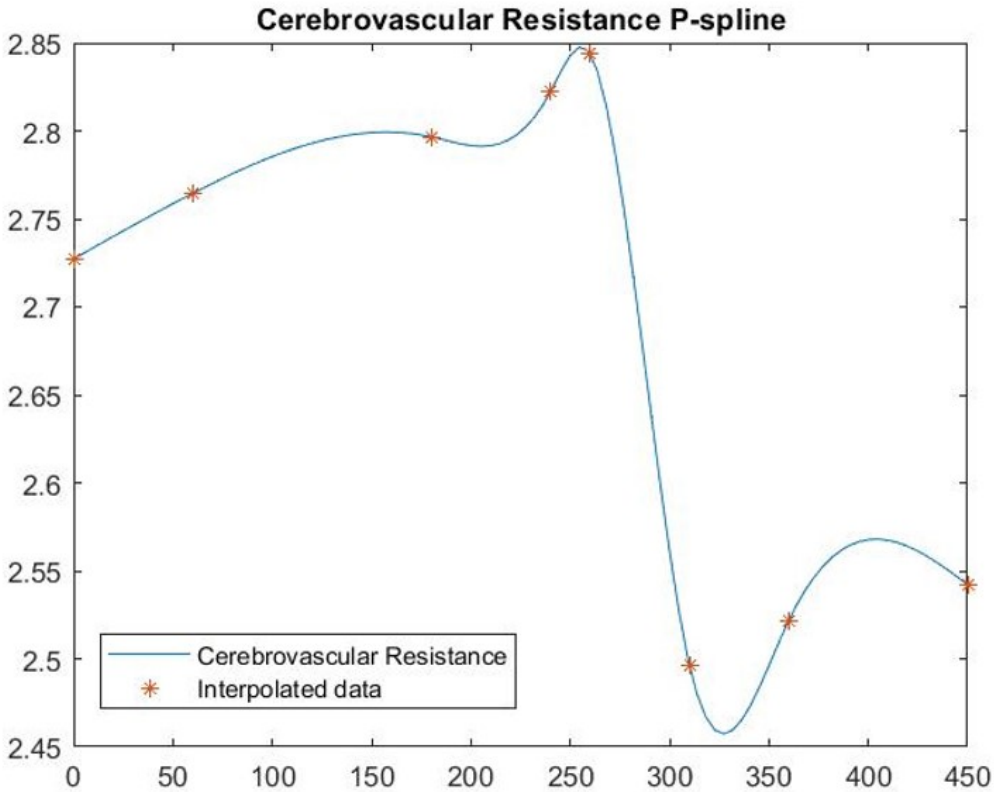

Fig. 9.: P-spline interpolation for cerebrovascular resistance data

The P-spline has kept much of the B-spline smoothness between D_2 and D_5 with only subtle adjustments to concavity between D_2 and D_4. The majority of the work of the P-spline was in reducing the drastic peaks between the first two and last two data points. The P-spline is certainly not perfect, as even though the curve is substantially smoother between D_5 and D_7 than in the B-spline without penalty, we may desire to reduce the roughness even further. This would be best achieved with knowledge of data between these two points. Interestingly, the data set from which we extracted our interpolating data in Figure 7 *does* curve between D_6 and D_7 similarly to what was predicted by the P-spline. However, this phenomenon seems to be coincidental, as the P-spline was calculated without providing any knowledge of this portion of the data and we would anticipate that the P-spline would be calculated only based on the data provided regardless of any other data existing.

5.2 Concluding Remarks

5.2.1 Limitations

Calculating with B- and P-splines was not without its difficulties. Early on, we discovered how sensitive B-spline construction is to truncation error. In the early stages of understanding how to interpolate with B-splines, we attempted to interpolate the example in Section 3.5 by hand. This included calculating the basis functions and their maxima analytically. As the maxima were not integer values, we wrote them down up to three decimal places and carried this truncation through to the rest of the computations. When it came to evaluating $C(u)$ at various u-values for plotting purposes, this would ultimately place a u-value into the neighboring knot span that it should not have been in, which led to an entirely different evaluation in the basis functions. For example, in Section 3.5 we calculate the maximum value t_2 corresponding to $N_{2,3}(u)$ to be 1.79599, just shy of being included in the knot span $[9/5, 12/5)$. We recorded the value of t_2 on paper as 1.80, merely a four-thousandths difference. Yet when we evaluated $C(u)$ at $u = t_2$ in order to plot the B-spline curve, t_2 was evaluated within $[9/5, 12/5)$, when it should have been evaluated in $[6/5, 9/5)$. This difficulty was easily overcome by coding the entire computation process so that calculations were stored in MATLAB's 16-digit precision and carried through computation at a higher precision than is feasible to write down.

Another difficulty encountered was in the computation of optimal control points in the P-spline. Eilers and Marx present a formula for the optimal control points, given a matrix N of the basis functions evaluated at equally spaced values, an order-k difference matrix $\mathcal{D}_k$, and a given λ. In an approximation problem, we can multiply $(N^T N + \lambda \mathcal{D}_k^T \mathcal{D}_k)^{-1}$ on the right by $N^T y$ with data values contained in y to get the optimal control points. However, this merely constructs a P-spline *approximating* the data, not interpolating it. In order to interpolate the data, we need to plot on a fine grid along many knot spans, which requires an increasing number of knots. The number of knot spans required to achieve the interpolation from the approximation is not intuitive and requires much trial and error. This all lends itself to relying on numerical methods to handle the computational legwork. The *Universal Method*, while choosing control points informed by the data we are interpolating, does not take into account any roughness of the curve it computes. Furthermore, the de Boor Algorithm becomes moot because it requires knowledge of the control points to begin with. So we are faced with two choices, neither of which is ideal: (1) easily compute a curve that interpolates the points we desire, but potentially be left with an unnecessarily rough curve, as in Figure 8; or (2) compute control points that minimize the roughness of the B-spline, but be left with the work of forcing the interpolation afterwards.

Finally, much of the literature regarding not just P-splines but B-splines in general is dedicated to approximation problems, not interpolation. Research on P-spline approximation presents another layer to explore due to not only aiming to minimize roughness, but to also maximize closeness of fit to the data. However, we argue that this should not reduce interpolation problems to a sort of "base case" or simplification on which approximation problems are founded. Rather, interpolation problems present their own layer of complexity to the conversation surrounding B- and P-splines, as it seems even more of a restriction to require the data to be an exact fit rather than simply a close fit. Leading into our discussion of future work in this area, a conversation surrounding the value of interpolation versus that of approximation seems long overdue.

5.2.2 Future Work

Throughout chapters 3 and 4, we used only uniformly spaced knot vectors. This was for both ease of computation and because both Lim [11] and Eilers and Marx [15] assert that there is no significant advantage of using a different knot vector generation technique. Nonetheless, it would be interesting see how other techniques affect the resulting B-spline.

We could also choose a different parameterization of our interpolating B-spline. We must determine a parameter value to correspond to our data points, but choosing the maximum of the basis functions is not the only option. The chordal method for parameterization is ideal when the data to be interpolated closely follows the control polygon. In this case, the choice of parameters can be approximated with the parameter corresponding to the nearest control point. The centripetal method extends the chordal method to reduce the angle at which the B-spline curves at the junctures of the control polygon segments. It may be worth discovering if the centripetal method diminishes the utility of P-splines.

There is much instructional text about elevating the degree of a B-spline. We have chosen to use $p = 3$ in this thesis because this causes the basis functions to be cubic, subsequently making every $N_{i,3}(u)$ similar to a cubic spline, the utility of which is not to be overlooked. Future work may involve varying p to understand if there is an optimal choice of degree given the number of data points to be interpolated and the parameterization method chosen.

Appendices

Appendix A

COMPUTATIONS FOR CEREBROVASCULAR RESISTANCE DATA

Knots	Knot spans
0	
	$[0, 0)$
0	
	$[0, 50)$
50	
	$[50, 100)$
100	
	$[100, 150)$
150	
	$[150, 200)$
200	
	$[200, 250)$
250	
	$[250, 300)$
300	
	$[300, 350)$
350	
	$[350, 400)$
400	
	$[400, 450)$
450	
	$[450, 450)$
450	

Table 6.: Knot values and knot spans for the cerebrovascular resistance data B-spline

Table 7.: Degree 0 Basis Functions for Cerebrovascular Resistance Data B-spline

Degree 0		
Basis Function	Equation	Interval
$N_{0,0}(u)$	0	everywhere
$N_{1,0}(u)$	1	$[0, 50)$
	0	elsewhere
$N_{2,0}(u)$	1	$[50, 100)$
	0	elsewhere
$N_{3,0}(u)$	1	$[100, 150)$
	0	elsewhere
$N_{4,0}(u)$	1	$[150, 200)$
	0	elsewhere
$N_{5,0}(u)$	1	$[200, 250)$
	0	elsewhere
$N_{6,0}(u)$	1	$[250, 300)$
	0	elsewhere
$N_{7,0}(u)$	1	$[300, 350)$
	0	elsewhere
$N_{8,0}(u)$	1	$[350, 400)$
	0	elsewhere
$N_{9,0}(u)$	1	$[400, 450)$
	0	elsewhere
$N_{10,0}(u)$	0	everywhere

Table 8.: Degree 1 Basis Functions for Cerebrovascular Resistance Data B-spline

Degree 1		
Basis Function	Equation	Interval
$N_{0,1}(u)$	$1 - \frac{1}{50}u$	$[0, 50)$
$N_{1,1}(u)$	$\frac{1}{50}u$	$[0, 50)$
	$2 - \frac{1}{50}u$	$[50, 100)$
$N_{2,1}(u)$	$\frac{1}{50}u - 1$	$[50, 100)$
	$3 - \frac{1}{50}u$	$[100, 150)$
$N_{3,1}(u)$	$\frac{1}{50}u - 2$	$[100, 150)$
	$4 - \frac{1}{50}u$	$[150, 200)$
$N_{4,1}(u)$	$\frac{1}{50}u - 3$	$[150, 200)$
	$5 - \frac{1}{50}u$	$[200, 250)$
$N_{5,1}(u)$	$\frac{1}{50}u - 4$	$[200, 250)$
	$6 - \frac{1}{50}u$	$[250, 300)$
$N_{6,1}(u)$	$\frac{1}{50}u - 5$	$[250, 300)$
	$7 - \frac{1}{50}u$	$[300, 350)$
$N_{7,1}(u)$	$\frac{1}{50}u - 6$	$[300, 350)$
	$8 - \frac{1}{50}u$	$[350, 400)$
$N_{8,1}(u)$	$\frac{1}{50}u - 7$	$[350, 400)$
	$9 - \frac{1}{50}u$	$[400, 450)$
$N_{9,1}(u)$	$\frac{1}{50}u - 8$	$[400, 450)$

Table 9.: Degree 2 Basis Functions for Cerebrovascular Resistance Data B-spline

Degree 2		
Basis Function	Equation	Interval
$N_{0,2}(u)$	$-\frac{1}{2500}u^2 + \frac{1}{25}u$	$[0, 50)$
	$\frac{1}{5000}u^2 - \frac{1}{25}u + 2$	$[50, 100)$
$N_{1,2}(u)$	$\frac{1}{5000}u^2$	$[0, 50)$
	$-\frac{1}{5000}u^2 - \frac{1}{100}u + \frac{3}{2}$	$[50, 100)$
	$\frac{1}{5000}u^2 - \frac{3}{50}u - \frac{9}{2}$	$[100, 150)$
$N_{2,2}(u)$	$\frac{1}{5000}u^2 - \frac{1}{50}u - \frac{1}{2}$	$[50, 100)$
	$-\frac{1}{2500}u^2 + \frac{1}{25}u - \frac{11}{2}$	$[100, 150)$
	$\frac{1}{5000}u^2 - \frac{1}{25}u + 8$	$[150, 200)$
$N_{3,2}(u)$	$\frac{1}{5000}u^2 - \frac{1}{25}u + 2$	$[100, 150)$
	$-\frac{1}{2500}u^2 + \frac{11}{22}u - \frac{23}{2}$	$[150, 200)$
	$\frac{1}{5000}u^2 - \frac{1}{10}u + \frac{25}{2}$	$[200, 250)$
$N_{4,2}(u)$	$\frac{1}{5000}u^2 - \frac{1}{16}u + \frac{9}{2}$	$[150, 200)$
	$-\frac{3}{5000}u^2 + \frac{3}{25}u - \frac{15}{2}$	$[200, 250)$
	$\frac{1}{5000}u^2 - \frac{49}{400}u + 18$	$[250, 300)$
$N_{5,2}(u)$	$\frac{1}{5000}u^2 - \frac{2}{25}u + 8$	$[200, 250)$
	$-\frac{1}{2500}u^2 + \frac{9}{200}u - \frac{59}{2}$	$[250, 300)$
	$\frac{1}{5000}u^2 - \frac{1}{50}u + \frac{49}{2}$	$[300, 350)$
$N_{6,2}(u)$	$\frac{1}{5000}u^2 - \frac{1}{5}u + \frac{25}{2}$	$[250, 300)$
	$-\frac{1}{2500}u^2 + \frac{57}{400}u - \frac{83}{2}$	$[300, 350)$
	$\frac{1}{5000}u^2 - \frac{4}{25}u + 32$	$[350, 400)$

$N_{7,2}(u)$	$\frac{1}{5000}u^2 - \frac{3}{25}u + 18$	$[300, 350)$
	$-\frac{1}{2500}u^2 + \frac{3}{10}u - \frac{111}{2}$	$[350, 400)$
	$\frac{1}{5000}u^2 - \frac{9}{50}u + \frac{81}{2}$	$[400, 450)$
$N_{8,2}(u)$	$\frac{1}{5000}u^2 - \frac{7}{50}u + \frac{49}{2}$	$[350, 400)$
	$-\frac{3}{5000}u^2 + \frac{1}{2}u - \frac{207}{2}$	$[400, 450)$

Table 10.: Degree 3 Basis Functions for Cerebrovascular Resistance Data B-spline

Degree 3		
Basis Function	Equation	Interval
$N_{0,3}(u)$	$\frac{1}{100}u\left(-\frac{1}{2500}u^2+\frac{1}{25}u\right)+\left(1-\frac{1}{150}u\right)\left(\frac{1}{500}u^2\right)$	$[0, 50)$
	$\frac{1}{100}u\left(\frac{1}{5000}u^2-\frac{1}{25}u+2\right)+\left(1-\frac{1}{150}u\right)\left(-\frac{1}{5000}u^2-\frac{1}{100}u+\frac{3}{2}\right)$	$[50, 100)$
	$\left(1-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{3}{50}u-\frac{9}{2}\right)$	$[100, 150)$
$N_{1,3}(u)$	$\frac{1}{150}u\left(\frac{1}{500}u^2\right)$	$[0, 50)$
	$\frac{1}{150}u\left(-\frac{1}{5000}u^2-\frac{1}{100}u+\frac{3}{2}\right)+\left(\frac{4}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{50}u-\frac{1}{2}\right)$	$[50, 100)$
	$\frac{1}{150}u\left(\frac{1}{5000}u^2-\frac{3}{50}u-\frac{9}{2}\right)+\left(\frac{4}{3}-\frac{1}{150}u\right)\left(-\frac{1}{2500}u^2+\frac{1}{25}u-\frac{11}{2}\right)$	$[100, 150)$
	$\left(\frac{4}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{25}u+8\right)$	$[150, 200)$
$N_{2,3}(u)$	$\left(\frac{1}{150}u-\frac{1}{3}\right)\left(\frac{1}{500}u^2-\frac{1}{50}u-\frac{1}{2}\right)$	$[50, 100)$
	$\left(\frac{1}{150}u-\frac{1}{3}\right)\left(-\frac{1}{2500}u^2+\frac{1}{25}u-\frac{11}{2}\right)+\left(\frac{5}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{25}u+2\right)$	$[100, 150)$
	$\left(\frac{1}{150}u-\frac{1}{3}\right)\left(\frac{1}{5000}u^2-\frac{1}{25}u+8\right)+\left(\frac{5}{3}-\frac{1}{150}u\right)\left(-\frac{1}{2500}u^2+\frac{11}{200}u-\frac{23}{2}\right)$	$[150, 200)$
	$\left(\frac{5}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{10}u+\frac{25}{2}\right)$	$[200, 250)$
$N_{3,3}(u)$	$\left(\frac{1}{150}u-\frac{2}{3}\right)\left(\frac{1}{500}u^2-\frac{1}{25}u+2\right)$	$[100, 150)$
	$\left(\frac{1}{150}u-\frac{2}{3}\right)\left(-\frac{1}{2500}u^2+\frac{11}{200}u-\frac{23}{2}\right)+\left(2-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{16}u+\frac{9}{2}\right)$	$[150, 200)$
	$\left(\frac{1}{150}u-\frac{2}{3}\right)\left(\frac{1}{5000}u^2-\frac{1}{10}u+\frac{25}{2}\right)+\left(2-\frac{1}{150}u\right)\left(-\frac{3}{5000}u^2+\frac{3}{25}u-\frac{15}{2}\right)$	$[200, 250)$
	$\left(2-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{49}{400}u+18\right)$	$[250, 300)$
$N_{4,3}(u)$	$\left(\frac{1}{150}u-1\right)\left(\frac{1}{500}u^2-\frac{1}{16}u+\frac{9}{2}\right)$	$[150, 200)$
	$\left(\frac{1}{150}u-1\right)\left(-\frac{3}{5000}u^2+\frac{3}{25}u-\frac{15}{2}\right)+\left(\frac{7}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{2}{25}u+8\right)$	$[200, 250)$
	$\left(\frac{1}{150}u-1\right)\left(\frac{1}{5000}u^2-\frac{49}{400}u+18\right)+\left(\frac{7}{3}-\frac{1}{150}u\right)\left(-\frac{1}{2500}u^2+\frac{9}{200}u-\frac{59}{2}\right)$	$[250, 300)$
	$\left(\frac{7}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{50}u+\frac{49}{2}\right)$	$[300, 350)$

$N_{5,3}(u)$	$\left(\frac{1}{150}u-\frac{4}{3}\right)\left(\frac{1}{5000}u^2-\frac{2}{25}u+8\right)$	$[200,250)$
	$\left(\frac{1}{150}u-\frac{4}{3}\right)\left(-\frac{1}{2500}u^2+\frac{9}{200}u-\frac{59}{2}\right)+\left(\frac{8}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{1}{5}u+\frac{25}{2}\right)$	$[250,300)$
	$\left(\frac{1}{150}u-\frac{4}{3}\right)\left(\frac{1}{5000}u^2-\frac{1}{50}u+\frac{49}{2}\right)+\left(\frac{8}{3}-\frac{1}{150}u\right)\left(-\frac{1}{2500}u^2+\frac{57}{400}u-\frac{83}{2}\right)$	$[300,350)$
	$\left(\frac{8}{3}-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{4}{25}u+32\right)$	$[350,400)$
$N_{6,3}(u)$	$\left(\frac{1}{150}u-\frac{5}{3}\right)\left(\frac{1}{5000}u^2-\frac{1}{5}u+\frac{25}{2}\right)$	$[250,300)$
	$\left(\frac{1}{150}u-\frac{5}{3}\right)\left(-\frac{1}{2500}u^2+\frac{57}{400}u-\frac{83}{2}\right)+\left(3-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{3}{25}u+18\right)$	$[300,350)$
	$\left(\frac{1}{150}u-\frac{5}{3}\right)\left(\frac{1}{5000}u^2-\frac{4}{25}u+32\right)+\left(3-\frac{1}{150}u\right)\left(-\frac{1}{2500}u^2+\frac{3}{10}u-\frac{111}{2}\right)$	$[350,400)$
	$\left(3-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{9}{50}u+\frac{81}{2}\right)$	$[400,450)$
$N_{7,3}(u)$	$\left(\frac{1}{150}u-2\right)\left(\frac{1}{5000}u^2-\frac{3}{25}u+18\right)$	$[300,350)$
	$\left(\frac{1}{150}u-2\right)\left(-\frac{1}{2500}u^2+\frac{3}{10}u-\frac{111}{2}\right)+\left(3-\frac{1}{150}u\right)\left(\frac{1}{5000}u^2-\frac{7}{50}u+\frac{99}{2}\right)$	$[350,400)$
	$\left(\frac{1}{150}u-2\right)\left(\frac{1}{5000}u^2-\frac{9}{50}u+\frac{81}{2}\right)+\left(3-\frac{1}{150}u\right)\left(-\frac{3}{5000}u^2+\frac{1}{2}u-\frac{207}{2}\right)$	$[400,450)$

The data point parameters are

$$t_0 = 55.6856$$

$$t_1 = 99.3311$$

$$t_2 = 150.5017$$

$$t_3 = 200.1672$$

$$t_4 = 249.8328$$

$$t_5 = 299.4983$$

$$t_6 = 350.6689$$

$$t_7 = 394.3144$$

The matrix N of basis functions evaluated at the data point parameters is below.

$$\begin{bmatrix} 0.5965 & 0.2293 & 2.4506 \times 10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 0.1734 & 0.6665 & 0.1601 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1617 & 0.6666 & 0.1717 & 1.6834 \times 10^{-9} & 0 & 0 & \\ 0 & 0 & 0.1650 & 0.6667 & 0.1683 & 6.2350 \times 10^{-9} & 0 & 0 \\ 0 & 0 & 6.2350 \times 10^{-9} & 0.1683 & 0.6667 & 0.1650 & 0 & 0 \\ 0 & 0 & 0 & 1.6834 \times 10^{-7} & 0.1717 & 0.6666 & 0.1617 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1601 & 0.6665 & 0.1734 \\ 0 & 0 & 0 & 0 & 0 & 2.4506 \times 10^{-4} & 0.2293 & 0.5965 \end{bmatrix}$$

The control point matrix P

$$P = \begin{bmatrix} -18.6507 & 3.6021 \\ 48.3170 & 2.5214 \\ 193.8694 & 2.8693 \\ 250.1565 & 2.7735 \\ 244.9935 & 2.9676 \\ 330.6742 & 2.4156 \\ 293.8158 & 2.3274 \\ 641.3898 & 3.3666 \end{bmatrix}$$

Bibliography

[1] I. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions: Part a - on the problem of smoothing or graduation. a first class of analytic approximation formulae," *Quarterly of Applied Mathematics*, vol. 4, no. 1, pp. 45–99, 1946. DOI: 10.1090/qam/15914.

[2] C. de Boor, "On calculating with b-splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50–62, 1972. DOI: 10.1016/0021-9045(72)90080-9.

[3] P. Eilers and B. Marx, "Flexible smoothing with b-splines and penalties," *Stastical Science*, vol. 11, no. 2, pp. 89–102, 1996. DOI: 10.1214/ss/1038425655.

[4] P. Eilers, B. Marx, and M. Durbán, "Twenty years of p-splines," *Statistics and Operations Research Transactions*, vol. 39, no. 2, pp. 149–186, 2015.

[5] F. O'Sullivan, "A statistical perspective on ill-posed inverse problems," *Stastical Science*, vol. 1, no. 4, pp. 502–527, 1986. DOI: 10.1214/ss/1177013525.

[6] H. Späth, *One Dimensional Spline Interpolation Algorithms*, 1st ed. Boca Raton, FL: CRC Press, 1995, 404 pp., ISBN: 1-56881-016-4.

[7] C. de Boor, *A Practical Guide to Splines* (Applied Mathematical Sciences), Revised. New York: Springer, 2001, vol. 27, 339 pp., ISBN: 0-387-95366-3.

[8] J. Hoschek and D. Lasser, *Fundamentals of Computer-Aided Geometric Design*, trans. by A. Peters. Wellesley, MA: A.K. Peters, 1993, 727 pp., ISBN: 978-1-56881-007-2.

[9] G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed. San Francisco, CA: Morgan Kaufmann, 2001, 497 pp., ISBN: 978-1-55860-737-8.

[10] D. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, 2nd ed. McGraw-Hill, Inc., 1990, 611 pp., ISBN: 0-07-053529-9.

[11] C.-G. Lim, "A universal parameterization in b-spline curve and surface parameterization," *Computer Aided Geometric Design*, vol. 16, pp. 407–422, 1999.

[12] MathWorks, *Matlab*, version 9.10.0 R2021a, 2021.

[13] Z. Li and J. Cao, *General p-splines for non-uniform b-splines*, 2022.

[14] C. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, pp. 177–183, 1967. DOI: `https://doi.org/10.1007/BF02162161`.

[15] P. Eilers and B. Marx, *Practical Smoothing: The Joys of P-splines*. Cambridge, UK: Cambridge University Press, 2021, 199 pp., ISBN: 978-1-108-61024-7.

[16] Z. Li and J. Cao, "Automatic search intervals for the smoothing parameter in penalized splines," *Statistics and Computing*, vol. 33, no. 1, pp. 1–18, 2023.

[17] M. Lu, Y. Liu, and C.-S. Li, "Efficient estimation of a linear transformation model for current status data via penalized splines," *Statistical Methods in Medical Research*, vol. 29, no. 1, pp. 3–14, 2020.

[18] P. Reiss and R. T. Ogden, "Smoothing parameter selection for a class of semiparametric linear models," *Journal of the Royal Statistical Society*, B, vol. 71, no. 2, pp. 505–523, 2009. DOI: `10.1111/j.1467-9868.2008.00695.x`.

[19] C. Gascoigne and T. Smith, "Penalized smoothing splines resolve the curvature identifiability problem in age-period-cohort models with unequal intervals," *Statistics in Medicine*, pp. 1888–1908, 2023.

[20] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: `https://www.R-project.org/`.

[21] P. Eilers, B. Marx, B. Li, J. Gampe, and M. X. Rodriguez-Alvarez, *Jops: Practical smoothing with p-splines*, R package version 0.1.19, 2023. [Online]. Available: https://CRAN.R-project.org/package=JOPS.

[22] L. Ellwein, S. Pope, A. Xie, J. Batzel, C. Kelley, and M. Olufsen, "Patient-specific modeling of cardiovascular and respiratory dynamics during hypercapnia," *Mathematical Biosciences*, vol. 241, no. 1, pp. 56–74, 2013, ISSN: 0025-5564. DOI: https://doi.org/10.1016/j.mbs.2012.09.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0025556412001824.