UNIVERSITY OF CALIFORNIA

Los Angeles

A Fresh Look at Spline Approximation Theory and Its Applications

A dissertation submitted in partial satisfaction of the

requirements for the degree of Doctor of Philosophy

in Mathematics

by

Jean-Michel Maldague

2021

ABSTRACT OF THE DISSERTATION

A Fresh Look at Spline Approximation Theory and Its Applications

by

Jean-Michel Maldague

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2021

Professor Christopher Anderson, Chair

This dissertation gives explicit algorithms for constructing multiple types of high dimensional 1D splines (standard and exponential A-splines, standard and exponential C-splines), and generating L2-orthogonal bases for various families of splines (via the standard and exponential A-spline procedures). These orthogonal bases of spline functions are used in L2 approximation of functions by way of orthogonal projection, and relevant error bounds for these approximations are given in L^2 and L^∞. The 1D spline approximation procedures developed here are used in construction of tensor product approximations of multivariate functions. Computational examples are provided.

The dissertation of Jean-Michel Maldague is approved.

University of California, Los Angeles

2021

Contents

List of Figures

Notation

In the remaining sections we use the following notation.

- Let $[a, b]$ be the interval of interest
- Let $f\colon [a, b] \to \mathbb{R}$ be a function we are trying to approximate
- Let $M \in \mathbb{N}$ be the panel count, i.e., the finite number of subdomains for the piecewise polynomial
- Let $t_0 < t_1 < t_2 < \cdots < t_{M-1} < t_M$ be the *knots* that define the subdomains of the spline (n.b. these can correspond to the $x_1, \ldots, x_n$ where the image of f is known, but in general this need not be the case; also, we often have $t_0 = a$ and $t_M = b$, but this need not be the case).
- Let $h_i := t_i - t_{i-1}$ be the panel widths
- Let $h = \Delta t = \max_i(h_i)$ be the max panel width
- Let $\hat{\mathcal{S}}_k$ be the space of splines of degree k (each polynomial piece of the spline is of degree $\leq k$) with the above knots.
- We will focus on the splines $\mathcal{S}_k$ with $(k-1)$ continuous derivatives. In other words, $\mathcal{S}_k = \hat{\mathcal{S}}_k \cap C^{k-1}[a, b]$.
- $D \in \mathbb{N}$ is the degree of the spline that will be produced; i.e., we will find a spline in S_D to approximate a given function f.

Acknowledgements

I would like to thank my advisor, Chris Anderson, for his immense support through the years. I would also like to thank my friends Yacoub Kureh, Will Oakley, Bonsoon Lin, Xenia Deviatkina-Loh, Peter Cheng, and the many other wonderful grad students in the UCLA math department for providing a cooperative and fruitful environment in which to learn. I would like to thank my professors for the wisdom they bestowed upon me, especially Chris Anderson, Luminita Vese, Joey Teran, Marcus Roper, Mario Bonk, Inwon Kim, Wilfrid Gangbo, Terry Tao, Fraydoun Rezakhanlou, and Michael Klass. Finally, I would like to thank my parents, Laurie and Pierre Maldague, for always supporting my endeavors and providing me with the confidence needed to tackle great challenges.

VITA

2009-2013	B.A. Mathematics, University of California – Berkeley
2014-2018	M.A. Mathematics, University of California – Los Angeles

1 Introduction

Splines have been a subject of interest since the late 1960s, when they were popularized by Schoenberg and de Boor. De Boor's seminal work on the subject, A Practical Guide to Splines ([7]), in which de Boor developed the general theory of B-splines and laid the groundwork for further study of these piecewise-polynomial functions, was an important development in the field. It was around this time that computers were becoming capable of performing simple, computationally inexpensive B-spline routines, many of which de Boor introduced as well ([6]). Shortly after their development, Schoenberg, de Boor, and others set out to establish error bounds for certain low-degree interpolatory (and other) splines in the L^2, L^1, and L^∞ norms ([21] [8] [16] [4] [3] [13]).

For several decades afterwards, the focus was not on L2 projection of functions onto spline spaces, but rather interpolation. There was some work in L2 projection onto specific spline spaces (e.g. least squares with noisy data, [22]), but the bulk of it was in interpolation. Recently, interest was generated in solving PDEs by finite element method incorporating spline approximations as part of isogeometric analysis, and this has led to key theoretical developments in the area of L2 and Ritz projection of functions onto a wide variety of spline spaces (see for example [20], [19], [23]).

Much of the early work on splines was in constructing splines using computationally efficient algorithms for early hardware, and developing error bounds for splines of low degree. Much of the recent work on splines is in theoretical error estimates (most of which not in the L2 setting), and neglects practical considerations for spline construction ([20], [19], [23]). The goal of this dissertation is to fill in the gap left by these two groups. The primary objective is to give explicit algorithms for constructing multiple types of high degree 1D splines (standard and exponential A-splines, standard and exponential C-splines), and generating orthogonal bases for various families of splines (via the standard and exponential A-spline procedures). With the availability of these orthogonal bases, we facilitate L2 approximation of functions by way of projection, and we can give a fairly general class of relevant error bounds for this type of approximation. The 1D spline approximation procedures developed facilitate construction of multidimensional approximants to multivariate functions in the context of tensor product approximations.

In section (2), some background is given to motivate why we look at splines. Specifically, we discuss why polynomials are a sufficiently large class of functions to use as agents of approximation, and how amongst functions which interpolate a given function, splines have a tendency to minimize certain L2 estimates of derivatives. This warrants a general study of splines, and in particular warrants the usage of splines for approximation routines in which we want to reduce unnecessary oscillation, such as for the purpose of obtaining tensor product approximations to multivariate functions. Notation that will be used throughout the dissertation is also given here.

In section (3), an account of B-splines is given. Certain deficiencies are given to motivate the development of A-splines. The B-spline discussion also serves as background for some of the theoretical results in this thesis– in particular, our proof of an L2 error bound.

In section (4), we introduce the procedure for approximating functions f by standard and exponential A-splines. We give reasons for looking at exponential A-splines in addition to standard A-splines; namely, to control the degree of the spline in regions where the function to be approximated experiences great irregularity/lack of smoothness.

In section (5), we motivate another type of spline, C-splines, as tools to help us achieve error bounds for exponential A-splines. These classes of splines are an original contribution to the body of knowledge on splines. Although the C-splines were originally developed as constructs to help prove error bounds, it turns out they exhibit many desirable characteristics as splines in their own right. Some of the immediately recognizable properties of C-splines are given here in order to motivate their study beyond them merely serving as a tool for the error bounds of exponential A-splines.

Section (6) concerns the analysis of L2 error incurred by the approximation procedures described in the previous sections. We begin by proving properties of B-splines that are required for proof of our L2 error estimates. We then move on to proving L2 error estimates for standard A-splines. Next in this section are L^∞ bounds for the C-splines that are necessary for the derivation of error bounds for exponential A-splines. These bounds are new. Following the bounds for C-splines are bounds for exponential A-splines, and a discussion of how to choose tension parameters heuristically. Intuition is given for how to view the additional two exponential terms associated with exponential A-splines compared to standard A-splines.

In section (7), we apply the 1D spline approximation techniques to the problem of tensor product approximation of multivariate functions. This application demonstrates the utility of creating 1D spline approximation using L2 projection. In particular, the use of 1D L2 projection facilitates the construction of multi-dimensional tensor product approximations that have few terms, thus providing a way of generating multi-dimensional approximations that are both accurate and computationally efficient to evaluate.

And finally in section (8), we look at what future work can be undertaken in the direction this dissertation heads, giving some potential additional practical applications for 1D spline approximations as well as ideas for improving the selection of parameters associated with the splines discussed.

2 Background

A common problem that arises in numerical analysis is how to approximate a function of complicated form by a function with simpler form. A more precise mathematical framework defining this problem can be expressed as follows.

Let $C[a,b]$ denote the vector space of real-valued continuous functions on the interval $[a,b]$ and let $||\cdot||$ be some norm on $C[a,b]$. Let $X \subset C[a,b]$ be a finite-dimensional subspace. Given some $f \in C[a,b]$, one may try to find a best approximation $g \in X$ such that

$$||g - f|| = \min_{\hat{g}\in X} ||\hat{g} - f||. \tag{1}$$

If one only knows the image of f on some finite subset

$$\{x_0, x_1, \ldots, x_n\} \subset [a,b]; \text{ where } x_i < x_{i+1}\,\forall i.$$

and the norm is an integral norm, one can consider minimizing the corresponding numerical quadrature to the norms. There are also other norms one can consider; e.g., one can use the squared 2-norm

$$\min_{\hat{g}\in X} \sum_{i=0}^{n} (\hat{g}(x_i) - f(x_i))^2. \tag{2}$$

If the functions in X are twice differentiable, one may incorporate a smoothness penalty:

$$\min_{\hat{g}\in X} \left[\sum_{i=0}^{n} (\hat{g}(x_i) - f(x_i))^2 + \lambda \int_a^b (\hat{g}''(x))^2 dx \right] \tag{3}$$

where $\lambda \geq 0$ is the *smoothing parameter.* This parameter controls the trade-off between fitting the data and smoothness as measured by the second derivative squared. If $\lambda = 0$, then the problem is one of strict interpolation and if $\lambda = \infty$, then the problem is one of linear least squares. Let us now turn our attention to defining the function space X.

2.1 Why Splines?

The Weierstrass approximation theorem guarantees every function in $C[a, b]$ can be uniformly approximated as closely as desired by a polynomial function, with respect to the uniform norm. Bernstein polynomials provide a constructive way to find such approximating polynomials ([11]), but the sequence of polynomial approximations can be slow to converge. Polynomial interpolation provides a fast and simple way to approximate functions, but high-order polynomial interpolation often runs into the problem of Runge's phenomenon (especially when the interpolation nodes are equispaced), causing wild oscillations of the approximating polynomial near the endpoints ([9]). The efficacy of polynomial interpolation is very sensitive to the number and placement of the data points $(x_0, f(x_0)),\ \ldots, (x_n, f(x_n))$; too few, and the polynomial interpolant will not be very accurate; too many, and the polynomial interpolant suffers from Runge's phenomenon near the endpoints, or worse if the underlying function is not sufficiently differentiable.

For interpolation (and smoothing) problems, *splines* (piecewise polynomials), are usually preferred to polynomial interpolation because splines do not suffer from Runge's oscillation phenomenon in the way the high-degree polynomials do. We can increase the number of data points and distribute them arbitrarily within the interval $[a, b]$, and splines can still do a great job approximating a function over that interval. Also, we can tune parameters of our spline to get far more accurate approximants than one can obtain using Bernstein polynomials for a similar amount of work.

Another reason to consider splines is that they (specifically natural cubic splines) minimize the smoothing problem in (3) over the class of twice differentiable functions. To see this, consider two functions– an arbitrary twice continuously differentiable function g, and a twice differentiable piecewise cubic polynomial $\tilde{g}$ that coincides with g on the set $\{x_i\}_{i=0}^n$ with vanishing second derivative at $a = x_0$ and $b = x_n$. Because they coincide on this set, the first term in the minimization is equal for both g and $\tilde{g}$. We now show that

$$\int_a^b (g''(x))^2 dx \geq \int_a^b (\tilde{g}''(x))^2 dx \tag{4}$$

where equality holds iff $g = \tilde{g}$. Let $g(x) = \tilde{g}(x) + \phi(x)$ and consider

$$\int_a^b (g''(x))^2 dx = \int_a^b (\tilde{g}''(x) + \phi''(x))^2 dx = \int_a^b (\tilde{g}''(x))^2 dx + 2\int_a^b \tilde{g}''(x)\phi''(x) dx + \int_a^b (\phi''(x))^2 dx.$$

Using integration by parts and the fact that $\phi(x) = 0$ on $\{x_i\}_{i=0}^n$, we get

$$\begin{aligned}\int_a^b \tilde{g}''(x)\phi''(x)dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \tilde{g}''(x)\phi''(x)dx = \sum_{i=0}^{n-1} \left(\tilde{g}''(x)\phi'(x)|_{x_i}^{x_{i+1}} - \int_{x_i}^{x_{i+1}} \tilde{g}'''(x)\phi'(x)dx \right) \\ &= 0 - \sum_{i=0}^{n-1} \tilde{g}'''\left(\frac{x_i + x_{i+1}}{2}\right) \int_{x_i}^{x_{i+1}} \phi'(x)dx = -\sum_{i=0}^{n-1} \tilde{g}'''\left(\frac{x_i + x_{i+1}}{2}\right) (\phi(x_{i+1}) - \phi(x_i)) \\ &= 0.\end{aligned}$$

Thus

$$\int_a^b (g''(x))^2 dx = \int_a^b (\tilde{g}''(x))^2 dx + 0 + \int_a^b (\phi''(x))^2 dx \geq \int_a^b (\tilde{g}''(x))^2 dx, \tag{5}$$

where equality holds iff $\phi''(x) \equiv 0$, which (since ϕ vanishes on $\{x_i\}_{i=0}^n$) is equivalent to $g = \tilde{g}$. both accurate and computationally efficient to evaluate.

This alludes to a recurring feature of many types of splines, that is noticeably absent from such routines as polynomial interpolation: when tuned and chosen correctly, splines can reduce unwanted oscillation in the approximant. This will be a recurring theme throughout our development and discussion of exponential splines in particular.

3 B-splines

Basis splines, also known as B-splines, are commonly used representation of a finite dimensional vector space of splines using minimally supported functions as the basis. By having basis functions with support concentrated on fewer panels, B-spline interpolation approximations can be evaluated by a stable and computationally efficient procedure.

One downside to using B-splines is it is often difficult to obtain the coefficients of the polynomial pieces making up the approximation. One can often get around this issue by using recursive formulas for evaluation B-splines such as the Cox-deBoor algorithm, but this can be unwieldy in some cases, such as when you need to do orthogonal projection. The reason orthogonal projection doesn't work so cleanly with B-splines is, in order to do the integration, we need to sample the B-spline at various points to make the integration formula accurate. We could employ a quadrature scheme with aptly-chosen knots to perfectly integrate polynomials multiplying B-splines, such as those that are needed for the orthogonal projection, but it may be preferable to have the explicit polynomial representation of the polynomial pieces, because this allows analytic integration techniques to be exploited. If we are integrating more complicated functions multiplying B-splines, such as calculations arising from orthogonally projecting with exponential splines, this problem becomes even more acute, and the desire to have polynomial representations of pieces of splines becomes even greater.

Another problem with the lack of polynomial representation of B-splines is in applications where the application of differential or integral operators is required. For example, the application of a differential operator would in practice require differentiating the equation defining the recursive relationship between B-splines, but it is not always convenient or preferable to have to derive new recursive formulas every time one wants to perform some procedure with B-splines.

These reasons are in part why we turn to A-splines. B-splines, for all their faults, are relatively cheap to compute, and certainly had great importance in the past when computational power was more limited.

4 A-Splines

4.1 Standard A-splines

Motivation

A-splines are an alternative way of generating a basis of the spline family S_D to the well-known B-spline construction ([1]). One major advantage to using A-splines is the ability to easily obtain polynomial representations of splines, as will be apparent through the method described below. The A-spline basis is orthonormal with respect to the standard L2 inner product on $C([a,b];\mathbb{R})$ and this facilitates L2 approximations of f.

Representation

One must choose the local polynomial representation of the spline that is conducive to accurate and stable computational procedures for determining the spline coefficients. Consider the form

$$y = a^0 + a^1 t + \cdots + a^D t^D$$

over a spline panel $[t_{j-1}, t_j]$, where $a^0, \cdots, a^D \in \mathbb{R}$ are the coefficients. Suppose $h = \max(t_j - t_{j-1})$ is small. If a^0 through a^D are stored imperfectly as $\tilde{a}^0$ through $\tilde{a}^D$, respectively, and $|a^i - \tilde{a}^i| \approx \epsilon$ for each i, then the error induced by the term $\tilde{a}^i t^i$ over this panel is roughly $\epsilon |t_j|^i$, which is bad because this means the accuracy of our approximation scheme will be dependent on the distance of the panel from 0. Ideally we would like the accuracy of the approximation to be translationally invariant.

One might attempt to amend this issue using the representation

$$y = a^0 + a^1(t - t_{j-1}) + a^2(t - t_{j-1})^2 + \cdots + a^D(t - t_{j-1})^D$$

but now, assuming the same error in numerical representation of the coefficients, the error contributed by the term $\tilde{a}^i(t - t_{j-1})^i$ is roughly ϵh^i, which is bad because over any given panel, the terms with higher powers of t will be more accurate than terms with lower powers of t. This representation also leads to large coefficients of higher powers of t, and indirectly inhibits the stable construction of spline coefficients.

If one considers the form

$$y = a^0 + a^1\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right) + a^2\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right)^2 + \cdots + a^D\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right)^D$$

then performing the same heuristic error analysis for this representation as in the previous two cases, we find that the error from faulty representation of the coefficient emanating from the term $a_i\left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right)^i$ is roughly ϵ, independent of i, j, and even h. This is clearly the best representation, and that used for A-splines.

Procedure for generating A-splines and approximating f

We are trying to construct a spline approximation to f using the A-spline construction. This entails finding a basis for $\mathcal{S}_D$ and then finding the L2-projection of f onto $\mathcal{S}_D$. The general procedures are described in ([1]), but for completeness we outline them here. The following steps are used to create an A-spline basis:

(1) Associate polynomial coefficients to splines in $\mathcal{S}_D$ over each panel $[t_{j-1}, t_j]$, using notation $h_j = t_j - t_{j-1}$, with

$$S_j(t) = a_j^0 + a_j^1\left(\frac{t - t_{j-1}}{h_j}\right) + \cdots + a_j^D\left(\frac{t - t_{j-1}}{h_j}\right)^D \tag{6}$$

and $S(t) = S_j(t)$ for $t \in [t_{j-1}, t_j]$, $j = 1, \ldots, M$, where $S \in \mathcal{S}_D$. Note: S is well-defined at the knots t_j due to continuity constraints.

(2) The required continuity & smoothness properties of the splines form an underdetermined homogeneous linear system for the coefficients, $Pa = 0$, where $P \in \mathbb{R}^{(M-1)D \times M(D+1)}$. These constraints are

$$S_j^{(l)}(t_j) = S_{j+1}^{(l)}(t_j) \tag{7}$$

for $l = 0, 1, \ldots, D-1$, which translates to equations for the spline coefficients a_j^k

$$\frac{l!}{0!}\frac{a_j^l}{h_j^l} + \frac{(l+1)!}{1!}\frac{a_j^{l+1}}{h_j^l} + \cdots + \frac{D!}{(D-l)!}\frac{a_j^D}{h_j^l} = \frac{l! a_{j+1}^l}{h_{j+1}^l} \tag{8}$$

for $j = 1, 2, \ldots, M-1$ and $l = 0, 1, \ldots, D-1$. We can move all terms to the LHS of this last equation, and then write it in the form $Pa = 0$, where $a \in \mathbb{R}^{M(D+1)}$ are the spline basis coefficients. Multiplying the rows of P by $\min(h_j^l, h_{j+1}^l)$ will in general result in P being better conditioned.

(3) The problem of finding a basis in $C([a,b];\mathbb{R})$ of splines in the subspace $\mathcal{S}_D$ becomes a problem of finding a basis in $\mathbb{R}^{M(D+1)}$ of spline coefficient vectors a solving the homogeneous system above. Note that $Pa = 0 \iff a \in im(P^t)^\perp$, so we may instead find a basis for the orthogonal complement of the image of P^t (with respect to the standard l2 vector inner product).

(4) Orthonormalize the columns of P^t and then form a basis for the complement by creating a collection of $M + D$ vectors with random coefficients and orthonormalizing them with respect to the orthogonalized columns of P^t.

(5) Generate a new basis of coefficient vectors corresponding to a basis of splines in $\mathcal{S}_D$ that is orthonormal with respect to the standard L2 inner product on $C([a,b];\mathbb{R})$. If splines $S, T \in \mathcal{S}_D$ have respective spline coefficients a_j^k, b_j^k, then an elementary integral computation

yields

$$\int_a^b S(x)T(x)dx = \sum_{j=1}^{M} \int_{t_{j-1}}^{t_j} S_j(x)T_j(x)dx = \sum_{j=1}^{M} \sum_{k=0}^{2D} \sum_{p+q=k} \frac{h_j a_j^p b_j^q}{k+1} \tag{9}$$

Note that if we use the (Hilbert matrix) notation

$$J' = \begin{pmatrix} 1 & 1/2 & 1/3 & \cdots & 1/(D+1) \\ 1/2 & 1/3 & & & \vdots \\ 1/3 & & \ddots & & \\ \vdots & & & & \\ 1/(D+1) & & \cdots & & 1/(2D+1) \end{pmatrix}$$

$$J = \begin{pmatrix} h_1 J' & & & \\ & h_2 J' & & \\ & & \ddots & \\ & & & h_M J' \end{pmatrix}, \quad a = \begin{pmatrix} a_1^0 \\ \vdots \\ a_1^D \\ \vdots \\ a_M^0 \\ \vdots \\ a_M^D \end{pmatrix}, \quad b = \begin{pmatrix} b_1^0 \\ \vdots \\ b_1^D \\ \vdots \\ b_M^0 \\ \vdots \\ b_M^D \end{pmatrix}$$

then the L2 inner product of the splines S, T associated with coefficient vectors $a, b \in \mathbb{R}^{M(D+1)}$ is

$$\langle S, T\rangle_{L^2} \;=\; \int_a^b S(x)T(x)dx \;=\; a^t J b \;=:\; \langle a, b\rangle_J \tag{10}$$

Now we orthonormalize our basis vectors found in the previous step w.r.t. this new inner product on $\mathbb{R}^{M(D+1)}$, which is equivalent to performing Gram-Schmidt on the associated splines and then converting back to coefficients. This generates coefficient vectors $\{a_i \in \mathbb{R}^{M(D+1)} : i = 1, \ldots, M+D\}$ corresponding to an L2-orthonormal basis $\{s_i\}$ of $\mathcal{S}_D$. We call this basis $\{s_i\}$ of $\mathcal{S}_D$ the A-spline basis, and the s_i's A-splines.

(6) With an orthonormal basis of $\mathcal{S}_D$, one can create an approximation to a given function f, using projection. Specifically, the L2 projection of f onto $\mathcal{S}_D$ is obtained using

$$S(f) := f^{||} = \sum_{i=1}^{M+D} \langle f, \tilde{s}_i \rangle s_i \tag{11}$$

where $S(\cdot)$ is the spline operator that takes functions to their L2 spline approximants in $\mathcal{S}_D$, the inner product $\langle f, \tilde{s}_i \rangle = \int_{t_0}^{T} f(t) s_i(t) dt$ is computed approximately using a quadrature rule like Trapezoidal rule, and $f = f^{||} + f^{\perp}$, where $f^{\perp}$ would be orthogonal to the space $\mathcal{S}_D$ if our quadrature rule were exact. To combat numerical imprecision brought upon by imperfectly orthogonalized basis splines, we can alternatively solve a least squares problem for the coefficients of each s_i on the RHS of the above equation. See ([1]) for details.

4.2 Exponential A-Splines

Motivation

Exponential A-splines are very similar in construction to A-splines, but they allow for additional parameters to give greater control over monotonicity/smoothing properties of the spline approximations ([15]). Many physical phenomena, e.g. those that can be modeled by a linear differential equation system like

$$\frac{d\vec{x}}{dt} = A\vec{x} \tag{12}$$

exhibit trajectories which can suddenly grow or decay, and it is often the case that polynomials are ill-suited for approximating these trajectories. It is towards this end that we modify the structure of the components of splines in order to accommodate such trajectories. Here, this is accomplished by appending two exponential terms to each piecewise polynomial part.

Two different forms

It should be noted that there exist different generalizations from splines to exponential splines; another common form is the general solution over each panel to the differential equation

$$(\mathrm{D} - \alpha_1)^{k_1}(\mathrm{D} - \alpha_2)^{k_2} \cdots (\mathrm{D} - \alpha_l)^{k_l} y = \sum_{m=1}^{M-1} b_m \delta(t - t_m) \tag{13}$$

for some $\alpha_1, \ldots, \alpha_l \in \mathbb{R}$ distinct, $b_1, \ldots, b_{M-1} \in \mathbb{R}$, and $k_1, \ldots, k_l \in \mathbb{N}$, where D is the standard differential operator, i.e.

$$\mathrm{D}y = y' = \frac{dy}{dt} \tag{14}$$

and $\delta(t - t_m)$ is the Dirac delta, defined distributionally so that

$$\int_a^b f(t)\delta(t - t_m)dt = f(t_m) \tag{15}$$

for continuous f, where $a = t_0 \leq t_1 \leq t_2 \leq \cdots t_{M-1} \leq t_M = b$ are the knots.

This alternative description of exponential splines yields the form

$$y = (a_1^0 + a_1^1 t + \cdots + a_1^{k_1-1} t^{k_1-1}) e^{\alpha_1 t} + \cdots + (a_l^0 + a_l^1 t + \cdots + a_l^{k_l-1} t^{k_l-1}) e^{\alpha_l t} \tag{16}$$

over each panel, with possibly discontinuous Dth derivatives at the knots, where

$$D = -1 + \sum_{m=1}^{M} k_m. \tag{17}$$

Specifically,

$$y^{(D)}(t_m^+) - y^{(D)}(t_m^-) = \lim_{t \to t_m^+} y^{(D)}(t) - \lim_{t \to t_m^-} y^{(D)}(t) = b_m \tag{18}$$

This is the version of exponential splines explored by e.g. Christensen & Massopust ([14]).

The version of exponential splines explored here will instead take on the form

$$y = a^0 + a^1 t + \cdots + a^D t^D + a^{D+1} e^{-\alpha t} + a^{D+2} e^{\alpha t} \tag{19}$$

over each spline panel, i.e. a polynomial part of degree D and two additional exponential terms. In fact, we will do some normalization of these terms, as will be evident in the next subsection, in order to improve computational accuracy. Note that subscripts on the coefficients will generally denote which panel we are on, and the superscript will generally denote the power of t associated with that coefficient. Unlike the other version, we can have a different exponential parameter α over each spline panel, instead of the same fixed $\alpha_1, \ldots, \alpha_l$ over each panel.

The choice of this form for the exponential splines matches that of McCartin ([15]). The reason for this choice is it facilitates the construction of an orthonormal basis of exponential splines, and produces splines which are not much more computationally expensive than standard splines, while still capturing the dominant behavior of the more generalized exponential splines. It is observed heuristically that the two additional exponential terms are sufficient for correctly mimicking the growth/decay, as well as monotonicity, of the function being approximated.

Representation

Just as in the case of standard A-splines, we will employ the following normalization for exponential A-splines:

$$\begin{aligned} y = a^0 + a^1 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}} \right) + a^2 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}} \right)^2 + \cdots + a^D \left(\frac{t - t_{j-1}}{t_j - t_{j-1}} \right)^D \\ + a^{D+1} \exp\left(\rho_j \frac{t_{j-1} - t}{t_j - t_{j-1}} \right) + a^{D+2} \exp\left(\rho_j \frac{t - t_j}{t_j - t_{j-1}} \right) \end{aligned} \tag{20}$$

over the interval $[t_{j-1}, t_j]$.

Generating exponential A-splines and approximating f

The procedure here will resemble that for standard A-splines, but with the two additional exponential terms. The ρ_j parameters associated with each panel $[t_j, t_{j+1}]$ together form the *tension* of the exponential spline. Just as in the case for standard A-splines, we will try to construct a spline approximation to f using the exponential A-spline construction. This entails finding a basis for $\mathcal{E}_D$, the space of exponential splines with polynomial part D, namely the solutions to the differential equation

$$\mathrm{D}^{D+1} \left(\mathrm{D}^2 - \sum_{j=1}^{M} \alpha_j^2 \mathbb{1}_{[t_{j-1}, t_j)}(t) \right) y = \sum_{m=1}^{M-1} b_j \delta(t - t_j) \tag{21}$$

where D is the standard differential operator, $\alpha_1, \ldots, \alpha_M \in \mathbb{R}$ are the tensions over each panel, $\mathbb{1}_A(t) = \begin{cases} 1 & t \in A \\ 0 & t \notin A \end{cases}$ is the standard indicator function over a set, $b_1, \ldots b_{M-1} \in \mathbb{R}$ are the discontinuous jumps in the $(D+2)$th derivative across the knots

$$a = t_0 \leq t_1 \leq \cdots \leq t_{M-1} \leq t_M = b, \tag{22}$$

and $\delta(t - t_j)$ is the Dirac delta defined above. The tensions will be normalized to match the form given above:

$$\rho_j = \alpha_j(t_j - t_{j-1}), \qquad j = 1, 2, \ldots, M \tag{23}$$

We will then find the L2-projection of f onto $\mathcal{E}_D$, by performing the following steps:

(1) Associate coefficients to splines in $\mathcal{E}_D$ over each panel $[t_{j-1}, t_j]$, with

$$\begin{aligned} S_j(t) = a_j^0 + a_j^1 \left(\frac{t - t_{j-1}}{h_j}\right) + a_j^2 \left(\frac{t - t_{j-1}}{h_j}\right) + \cdots + a_j^D \left(\frac{t - t_{j-1}}{h_j}\right)^D \\ + a_j^{D+1} \exp\left(\rho_j \frac{t_{j-1} - t}{h_j}\right) + a_j^{D+2} \exp\left(\rho_j \frac{t - t_j}{h_j}\right) \end{aligned} \tag{24}$$

and $S(t) = S_j(t)$ for $t \in [t_{j-1}, t_j]$, $j = 1, \ldots, M$, where $S \in \mathcal{E}_D$. The tensions $\rho_j \in (0, \infty)$ are prescribed in advance. $h_j = t_j - t_{j-1}$ are the panel widths. Note: S is well-defined at the knots t_j due to continuity constraints.

(2) The required continuity & smoothness properties of the splines as an underdetermined homogeneous linear system for the coefficients, $Pa = 0$, where $P \in \mathbb{R}^{(M-1)(D+2) \times M(D+3)}$. These constraints are

$$S_j^{(l)}(t_j) = S_{j+1}^{(l)}(t_j) \tag{25}$$

for $l = 0, 1, \ldots, D-1$, which translates to equations for the spline coefficients a_j^k

$$\frac{l!}{0!}\frac{a_j^l}{h_j^l} + \frac{(l+1)!}{1!}\frac{a_j^{l+1}}{h_j^l} + \cdots + \frac{D!}{(D-l)!}\frac{a_j^D}{h_j^l} + a_j^{D+1}\left(\frac{-\rho_j}{h_j}\right)^l \exp(-\rho_j) + a_j^{D+2}\left(\rho_j h_j\right)^l = \frac{l! a_{j+1}^l}{h_{j+1}^l} + a_{j+1}^{D+1}\left(\frac{-\rho_{j+1}}{h_{j+1}}\right)^l + a_{j+1}^{D+2}\left(\frac{\rho_{j+1}}{h_{j+1}}\right)^l \exp(-\rho_{j+1}) \tag{26}$$

for $j = 1, 2, \ldots, M-1$ and $l = 0, 1, \ldots, D+1$. We have $(D+2)$ many smoothness constraints on the spline for each interior knot, of which there are $(M-1)$. We can move all terms to the LHS of this last equation, and then write it in the form $Pa = 0$, where $a \in \mathbb{R}^{M(D+3)}$ houses the spline coefficients (we need $(D+3)$ many coefficients per panel, and there are M many panels). Multiplying the rows of P by $\min(h_j^l, h_{j+1}^l)$ will in general result in P being better conditioned, so this is usually a good idea.

(3) The problem of finding a basis in $C([a,b];\mathbb{R})$ of exponential splines in the subspace $\mathcal{E}_D$ becomes a problem of finding a basis in $\mathbb{R}^{M(D+1)}$ of spline coefficient vectors a solving the homogeneous system above. Note that $Pa = 0 \iff a \in im(P^t)^\perp$, so we may instead find a basis for the orthogonal complement of the image of P^t (with respect to the standard l2 vector inner product).

(4) Orthonormalize the columns of P^t and then form a basis for the complement by creating a collection of $M + D + 2$ vectors with random coefficients and orthonormalizing them with respect to the orthogonalized columns of P^t.

(5) Generate a new basis of coefficient vectors corresponding to a basis of splines in $\mathcal{E}_D$ that is orthonormal with respect to the standard L2 inner product on $C([a,b];\mathbb{R})$. If splines $S, T \in E_D$ have respective spline coefficients a_j^k, b_j^k for $k = 0, 1, \ldots, D{+}2$ and $j = 1, 2, \ldots, M$,

then an elementary but tedious integral computation yields

$$
\begin{aligned}
\int_a^b S(x)T(x)dx &= \sum_{j=1}^{M} \int_{t_{j-1}}^{t_j} S_j(x)T_j(x)dx && (27)\\
&= \sum_{j=1}^{M} \sum_{k=0}^{2D} \sum_{p+q=k} \frac{h_j a_j^p b_j^q}{k+1} && (28)\\
&+ \sum_{j=1}^{M} \sum_{k=0}^{D} h_j (a_j^k b_j^{D+1} + a_j^{D+1} b_j^k) \left(\frac{k!(1-e^{-\rho_j})}{\rho_j^{k+1}} - \sum_{l=0}^{k-1} \frac{k! e^{-\rho_j}}{(k-l)! \rho_j^{l+1}} \right)\\
&+ \sum_{j=1}^{M} \sum_{k=0}^{D} h_j e^{-\rho_j} (a_j^k b_j^{D+1} + a_j^{D+1} b_j^k) \left(\frac{(-1)^k k!(e^{\rho_j}-1)}{\rho_j^{k+1}} - \sum_{l=0}^{k-1} \frac{(-1)^l k! e^{\rho_j}}{(k-l)! \rho_j^{l+1}} \right)\\
&+ \sum_{j=1}^{M} \frac{h_j a_j^{D+1} b_j^{D+1}}{2\rho_j} \left(1 - e^{-2\rho_j}\right)\\
&+ \sum_{j=1}^{M} h_j (a_j^{D+1} b_j^{D+2} + a_j^{D+2} b_j^{D+1}) e^{-\rho_j}\\
&+ \sum_{j=1}^{M} \frac{h_j a_j^{D+2} b_j^{D+2}}{2\rho_j} \left(e^{\rho_j} - e^{-\rho_j}\right)
\end{aligned}
$$

Note that if we use the notation

$$
J_j' = \begin{pmatrix}
1 & 1/2 & 1/3 & \cdots & 1/(D+1) & \alpha_j^0 & \beta_j^0\\
1/2 & 1/3 & & & \vdots & \alpha_j^1 & \beta_j^1\\
1/3 & & \ddots & & & \alpha_j^2 & \beta_j^2\\
\vdots & & & & & \vdots & \\
1/(D+1) & & \cdots & & 1/(2D+1) & \alpha_j^D & \beta_j^D\\
\alpha_j^0 & \alpha_j^1 & \alpha_j^2 & \cdots & \alpha_j^D & \gamma_j^1 & \gamma_j^2\\
\beta_j^0 & \beta_j^1 & \beta_j^2 & \cdots & \beta_j^D & \gamma_j^3 & \gamma_j^4
\end{pmatrix}
$$

$$J = \begin{pmatrix} h_1 J' & & & \\ & h_2 J' & & \\ & & \ddots & \\ & & & h_M J' \end{pmatrix}, \quad a = \begin{pmatrix} a_1^0 \\ \vdots \\ a_1^{D+2} \\ \vdots \\ a_M^0 \\ \vdots \\ a_M^{D+2} \end{pmatrix}, \quad b = \begin{pmatrix} b_1^0 \\ \vdots \\ b_1^{D+2} \\ \vdots \\ b_M^0 \\ \vdots \\ b_M^{D+2} \end{pmatrix}$$

$$\alpha_j^k = \frac{k!(1 - e^{-\rho_j})}{\rho_j^{k+1}} - \sum_{l=0}^{k-1} \frac{k! e^{-\rho_j}}{(k-l)! \rho_j^{l+1}}$$

$$\beta_j^k = \frac{(-1)^k k!(e^{\rho_j} - 1)}{\rho_j^{k+1}} - \sum_{l=0}^{k-1} \frac{(-1)^l k! e^{\rho_j}}{(k-l)! \rho_j^{l+1}}$$

$$\gamma_j^1 = \frac{1 - e^{-2\rho_j}}{2\rho_j}$$

$$\gamma_j^2 = e^{-\rho_j}$$

$$\gamma_j^3 = \gamma_j^2$$

$$\gamma_j^4 = \frac{e^{\rho_j} - e^{-\rho_j}}{2\rho_j}$$

then the L2 inner product of the exponential splines $S,\ T$ associated with coefficient vectors $u, v \in \mathbb{R}^{M(D+3)}$ is

$$\langle S, T \rangle_{L^2} \ = \ \int_a^b S(x)T(x)dx \ = \ u^t J v \ =: \ \langle u, v \rangle_J \tag{29}$$

Now we orthonormalize our basis vectors found in the previous step w.r.t. this new inner product on $\mathbb{R}^{M(D+3)}$, which is equivalent to performing Gram-Schmidt on the associated splines and then converting back to coefficients. This generates coefficient vectors $\{a_i \in \mathbb{R}^{M(D+3)} : i = 1, \ldots, M + D + 2\}$ corresponding to an L2-orthonormal basis $\{s_i\}$ of $\mathcal{E}_D$. We call this basis $\{s_i\}$ of $\mathcal{E}_D$ the exponential A-spline basis, and the s_i's exponential

A-splines. Note that this family of exponential A-splines is dependent on a combination of the pre-specified tensions, knots, and polynomial part degree.

(6) With an orthonormal basis of $\mathcal{E}_D$, one can create an approximation to a given function f, using projection. Specifically, the L2 projection of f onto $\mathcal{E}_D$ is obtained using the orthogonal projection formula

$$S(f) := f^{||} = \sum_{i=1}^{M+D+2} \tilde{\langle f, s_i \rangle} s_i \tag{30}$$

where $S(\cdot)$ is the spline operator that takes functions to their L2 spline approximants in $\mathcal{E}_D$, the inner product $\tilde{\langle f, s_i \rangle} = \int_{t_0}^{T} f(t)s_i(t)dt$ is computed approximately using a quadrature rule like Trapezoidal rule, and $f = f^{||} + f^{\perp}$, where $f^{\perp}$ would be orthogonal to the space $\mathcal{E}_D$ if our quadrature rule were exact. To combat numerical imprecision brought upon by imperfectly orthogonalized basis splines, we can alternatively solve a least squares problem for the coefficients of each s_i on the RHS of the above equation.

5 C-splines

5.1 Standard C-splines

Description

In order to prove a certain convergence result for exponential A-splines, we need to introduce another type of spline to help us with that task. These are the C-splines, where the "C" stands for"cycle". The idea behind them is to construct a spline that periodically exactly matches a Taylor polynomial of the desired function to approximate. In the case of a standard spline of degree D, with $(D + 1)$ degrees of freedom per panel, we can accomplish this matching the degree d Taylor polynomial for f exactly once every $(D + 1)$ panels. We assume $d \leq D$ here. The construction yields a Hermite-style spline, where we match not only function value but also derivatives at a point. In contrast to traditional Hermite splines, we do not impose interpolation/osculatory conditions at every knot; rather, we do so only every $(D + 1)$ many knots.

Motivation

The reason this type of spline assists in establishing error bounds for the exponential A-spline procedure, is there is an analogous construction of C-splines that are exponential splines, and we can easily develop error bounds for this type of spline. This will establish an upper bound on the "closeness" of exponential splines to a function being approximated. Since the exponential A-spline procedure finds the closest exponential spline in $\mathcal{E}_D$ to the desired function (in the L2 sense), the error bound for C-splines results in an error bound for exponential splines.

Properties and Discussion of standard C-splines

The defining feature of C-splines is that they match a certain Taylor polynomial of f every $(D+1)$ many panels. This property easily gives us some salient information about these splines.

First, they are semi-local approximations, so that errors in one section of $(D+1)$ panels do not propagate to the next section of $(D+1)$ panels. This will be demonstrated through plots in the analysis section, where we also demonstrate general error bounds for C-splines. Since the Taylor polynomial of degree d is in general accurate to order $\mathcal{O}(h^{d+1})$, assuming sufficient differentiability of the underlying function, we already have a very natural error bound to work with for C-splines. In fact, we can even quantify the closeness of derivatives of C-splines to the corresponding derivatives of the underlying functions they approximate. This will be seen in the analysis section.

A C-spline approximation can be efficiently computed, just requiring work that is linear in the number of panels M and quartic in the degree D, i.e. the computational complexity of producing C-splines is $\mathcal{O}(MD^4)$.

C-splines are flexible in that they can be "grafted" onto existing splines, extending them as needed. If we wish to extend a spline without redoing it entirely, we can accomplish this by appending C-splines to it. This is a crucial characteristic that is a result of C-splines being semi-local. This is also a unique characteristic of C-splines that is not enjoyed by most other types of splines; notably B-splines. The A-spline procedure can be manufactured to enforce certain osculatory behavior of the spline at the leftmost endpoint, so it is possible to "extend" splines by using A-splines, though doing this would also decrease the number of A-splines in the A-spline basis used for L2 projection.

By selecting the d parameter appropriately, we can limit the degree of the Taylor polynomial to match according to however many derivatives f has. If f has unbounded behavior in one of its later derivatives, we can shield the spline from emulating this blow-up by designing the spline to match a Taylor polynomial of lesser degree where the relevant derivative of f misbehaves. Since the blocks in between the panels where the Taylor polynomial is matched are completely independent of each other, C-splines can be computed in parallel. This is especially advantageous in the modern era, in which processor speeds are stagnating but processor core counts are growing, making parallelization a desirable trait among algorithms.

Overall, C-splines are very robust, cheap, and flexible in their placement. Three downsides are: (1) in between the panels where the Taylor polynomial is matched exactly, we lose a power of h in the accuracy (details in the analysis section); (2) C-splines are relatively inflexible in the role that they play approximating f (i.e., we might not necessarily want to periodically match Taylor polynomials of f, we might want a different idea for our approximation); and (3) in order to construct C-splines, we need to know derivative information about f once every $(D+1)$ panels (though, there are ways around this, e.g. by approximating these derivatives with finite differences). However, independently from their practical use as a family of splines for approximating functions, they are particularly useful for the purpose of establishing error bounds for exponential splines.

Finally, as the algorithm described next will demonstrate, a key idea within C-splines is treating the $(D+1)$th coefficients of each spline panel as controls for whatever purpose we devise. The previous coefficients are locked into place according to the continuity constraints imposed on splines in $\mathcal{S}_D$, but we have full control over the very last coefficient. Parametrizing these "controls" can enable us to satisfy a number of desirable conditions, not just periodically matching Taylor polynomials.

Representation

We will represent standard C-splines in exactly the same way we represent standard A-splines, namely,

$$y_j = a_j^0 + a_j^1 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right) + a_j^2 \left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right)^2 + \cdots + a_j^D \left(\frac{t - t_{j-1}}{t_j - t_{j-1}}\right)^D \tag{31}$$

over the jth panel, $[t_{j-1}, t_j]$. We will also use the panel width notation $h_j = t_j - t_{j-1}$.

Generating standard C-splines and approximating f

Let f be the function we wish to approximate. Let the knots

$$a = t_0 < t_1 < t_2 < \cdots < t_{M-1} < t_M = b$$

be decided upon in advance. Let D be the degree of the spline we wish to produce, existing within $\mathcal{S}_D$ (i.e., the spline will be $(D-1)$ times continuously differentiable). Let $d \in \mathbb{N} \cup \{0\}$ be the degree of the Taylor polynomial of f we wish to match every $(D+1)$ many panels.

First we try to quantify how the choice of a_j^D affects the coefficients a_{j+1}^i for $i = 0, 1, \ldots, D-1$. The mechanism by which the last coefficient will impact the coefficients of the next panel, is of course the continuity constraints on the spline. The derivative continuity condition across node t_j is

$$S_j^{(i)}(t_j) = \frac{i! a_j^i}{h_j^i} + \frac{(i+1)!}{1!} \cdot \frac{a_j^{i+1}}{h_j^i} + \frac{(i+2)!}{2!} \cdot \frac{a_j^{i+2}}{h_j^i} + \cdots + \frac{D!}{(D-i)!} \cdot \frac{a_j^D}{h_j^i} = \frac{i! a_{j+1}^i}{h_j^i} = S_{j+1}^{(i)}(t_j) \tag{32}$$

for $i = 0, 1, \ldots, D-1$. Isolating the coefficient a_{j+1}^i yields

$$a_{j+1}^i = \left(\frac{h_{j+1}}{h_j}\right)^i \left[\binom{i}{i} a_j^i + \binom{i+1}{i} a_j^{i+1} + \cdots + \binom{D}{i} a_j^D \right], \qquad i = 0, 1, \ldots, D-1 \tag{33}$$

We can represent these equations in a linear system, as follows:

$$\begin{bmatrix} a_{j+1}^0 \\ a_{j+1}^1 \\ a_{j+1}^2 \\ \vdots \\ a_{j+1}^{D-1} \end{bmatrix} = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{0}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{1}{0} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-2}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-1}{0} \\ 0 & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{1}{1} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-2}{1} & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-1}{1} \\ 0 & 0 & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-2}{2} & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-1}{2} \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \binom{D-1}{D-1} \end{bmatrix} \begin{bmatrix} a_j^0 \\ a_j^1 \\ a_j^2 \\ \vdots \\ a_j^{D-1} \end{bmatrix}$$

$$+a_j^D \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D}{0} \\ \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D}{1} \\ \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D}{2} \\ \vdots \\ \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \binom{D}{D-1} \end{bmatrix} \tag{34}$$

Using the notation

$$a_j = \begin{bmatrix} a_j^0 \\ \\ a_j^1 \\ \\ a_j^2 \\ \\ \vdots \\ \\ a_j^{D-1} \end{bmatrix}$$

$$A_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{0}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{1}{0} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-2}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-1}{0} \\ 0 & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{1}{1} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-2}{1} & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-1}{1} \\ 0 & 0 & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-2}{2} & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-1}{2} \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \binom{D-1}{D-1} \end{bmatrix}$$

$$b_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \begin{pmatrix} D \\ 0 \end{pmatrix} \\ \left(\frac{h_{j+1}}{h_j}\right)^1 \begin{pmatrix} D \\ 1 \end{pmatrix} \\ \left(\frac{h_{j+1}}{h_j}\right)^2 \begin{pmatrix} D \\ 2 \end{pmatrix} \\ \vdots \\ \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \begin{pmatrix} D \\ D-1 \end{pmatrix} \end{bmatrix}$$

we can condense this "update" equation to

$$a_{j+1} = A_j a_j + a_j^D b_j \tag{35}$$

Now, fixing j, we can begin the cycle of preparing the next matched Taylor polynomial. Specifically, given any coefficient vector a_j, we can use this update equation to express the coefficient vector $(D + 1)$ many panels to the right, namely, $a_{j+(D+1)}$, in terms of a_j and the undetermined coefficients $a_{j+1}^D, \dots, a_{j+D}^D$. Since we have the power to choose these latter coefficients (the coefficients corresponding to the largest power of t in the spline representation over each panel), this gives us D degrees of freedom. Forcing the D-dimensional vector a_{j+D+1} to have coefficients prescribed according to the Taylor polynomial of f of degree d over panel $[t_{j+D}, t_{j+D+1}]$ imposes D constraints. So, we end up with a D by D linear system,

$$a_{j+1} = A_j a_j + a_j^D b_j \tag{36}$$

$$a_{j+2} = A_{j+1} a_{j+1} + a_{j+1}^D b_{j+1} \tag{37}$$

$$= A_{j+1} A_j a_j + a_j^D A_{j+1} b_j + a_{j+1}^D b_{j+1} \tag{38}$$

$$a_{j+3} = A_{j+2} a_{j+2} + a_{j+2}^D b_{j+2} \tag{39}$$

$$= A_{j+2}A_{j+1}A_j a_j + a_j^D A_{j+2}A_{j+1}b_j + a_{j+1}^D A_{j+2}b_{j+1} + a_{j+2}^D b_{j+2} \tag{40}$$

$$\vdots \tag{41}$$

$$a_{j+(D+1)} = A_{j+D}A_{j+D-1}\cdots A_j a_j + \sum_{k=0}^{D} a_{j+k}^D \left[\prod_{l=1}^{D-k} A_{j+D+1-l}\right] b_{j+k} \tag{42}$$

$$= A_{j+D}A_{j+D-1}\cdots A_j a_j + a_j^D A_{j+D}A_{j+D-1}\cdots A_{j+1}b_j + \sum_{k=1}^{D} a_{j+k}^D \left[\prod_{l=1}^{D-k} A_{j+D+1-l}\right] b_{j+k} \tag{43}$$

$$= A_{j+D}A_{j+D-1}\cdots A_{j+1}(A_j a_j + a_j^D b_j) + \sum_{k=1}^{D} a_{j+k}^D \left[\prod_{l=1}^{D-k} A_{j+D+1-l}\right] b_{j+k} \tag{44}$$

Now we can isolate the variables $a_{j+1}^D, \ldots, a_{j+D}^D$ in terms of all the other (known) quantities:

$$C_j \begin{bmatrix} a_{j+1}^D \\ \\ a_{j+2}^D \\ \\ \vdots \\ \\ a_{j+D}^D \end{bmatrix} = a_{j+(D+1)} - A_{j+D}A_{j+D-1}\cdots A_{j+1}(A_j a_j + a_j^D b_j) \tag{45}$$

$$\implies \begin{bmatrix} a_{j+1}^D \\ \\ a_{j+2}^D \\ \\ \vdots \\ \\ a_{j+D}^D \end{bmatrix} = C_j^{-1}\left(a_{j+(D+1)} - A_{j+D}A_{j+D-1}\cdots A_{j+1}(A_j a_j + a_j^D b_j)\right) \tag{46}$$

where

$$C_j = \left[\left[\prod_{l=1}^{D-1} A_{j+D+1-l} \right] b_{j+1} \quad \left[\prod_{l=1}^{D-2} A_{j+D+1-l} \right] b_{j+2} \quad \cdots \quad \left[\prod_{l=1}^{1} A_{j+D+1-l} \right] b_{j+D-1} \quad b_{j+D} \right] \in \mathbb{R}^{D \times D}. \tag{47}$$

It will be demonstrated in the analysis section that this matrix C_j is invertible.

We can explicitly solve for $a_{j+(D+1)}$ by matching the degree d Taylor polynomial at node t_{j+D}, by heeding

$$\frac{k! a^k_{j+(D+1)}}{h^k_{j+(D+1)}} = S^{(k)}_{j+(D+1)}(t_{j+D}) = f^{(k)}(t_{j+D}) \tag{48}$$

for $k = 0, 1, \ldots, d$ and setting $a^k_{j+(D+1)} = 0$ for $k = d+1, \ldots, D$. That is,

$$a_{j+(D+1)} = \begin{bmatrix} a^0_{j+(D+1)} \\ a^1_{j+(D+1)} \\ \vdots \\ a^d_{j+(D+1)} \\ a^{d+1}_{j+(D+1)} \\ \vdots \\ a^D_{j+(D+1)} \end{bmatrix} = \begin{bmatrix} \dfrac{(h_{j+(D+1)})^0 f(t_{j+D})}{0!} \\ \dfrac{(h_{j+(D+1)})^1 f'(t_{j+D})}{1!} \\ \vdots \\ \dfrac{(h_{j+(D+1)})^d f^{(d)}(t_{j+D})}{d!} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{49}$$

The overall procedure is:

(1) Start with any set of coefficients $a_1 \in \mathbb{R}^{D+1}$ on the first panel $[t_0, t_1]$.

(2) Use equation (49) to solve for the spline coefficients needed $(D + 1)$ panels to the right

(3) Use equation (46) to solve for the coefficients of the highest power terms for each intermediate interval

(4) Use the update equation (35) to update all the remaining coefficients over each intermediate panel

(5) Repeat the process over again, cycling through steps (1) through (4) until there are fewer than $(D + 1)$ panels to the right

(6) Once we can no longer continue the process due to not having enough panels to the right, we simply choose the coefficients of the highest powers of t over the remaining panels, namely, $a_{j+1}^D, a_{j+2}^D, \ldots, a_M^D$, so that the Dth derivatives of the spline and the function match up at the nodes $t_j, t_{j+1}, \ldots, t_{M-1}$. Note: if $d < D$, then we may instead simply set these coefficients to 0.

This gives us the standard C-spline approximation to f.

In figures (1) through (10), we give some salient examples of C-splines to demonstrate the properties discussed above. The first figure shows how a quadratic C-spline can periodically match the 0th degree Taylor polynomial of a sine function. We see this manifest in regions where the spline is flat. This procedure is replicated with cubic splines, which demonstrates the same phenomenon. As we increase the degree of the Taylor polynomial matched, the spline becomes more accurate, as evidenced by the shrinking error. Once we reach the maximum degree Taylor polynomial that a spline of a certain degree can replicate (namely, the degree of the spline itself), we may further shrink the error by refining the panel width, as is done in the last few figures. It is important to note two observations with these graphs: (1) due to the semi-local nature of these splines, and the periodicity of the underlying sine function they are approximating, we expect the error plots to look periodic; and (2) due to the crude approximation scheme used when there are not enough panels to the right to match the next Taylor polynomial, we expect the errors to magnify towards the right endpoint. Both of these phenomena are demonstrated in these examples.

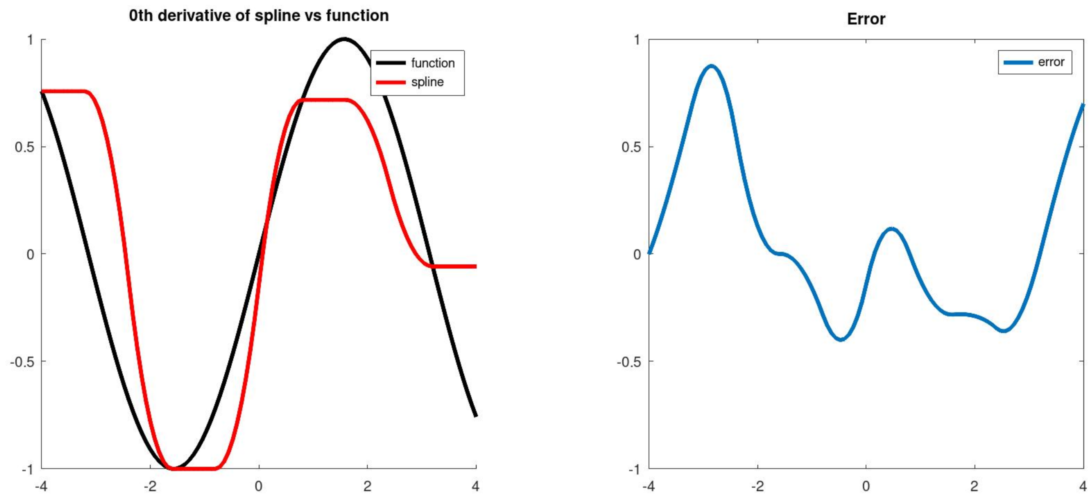


Figure 1: Quadratic C-spline that matches the degree 0 Taylor polynomial of a sine function once every 3 panels, 10 panels in total

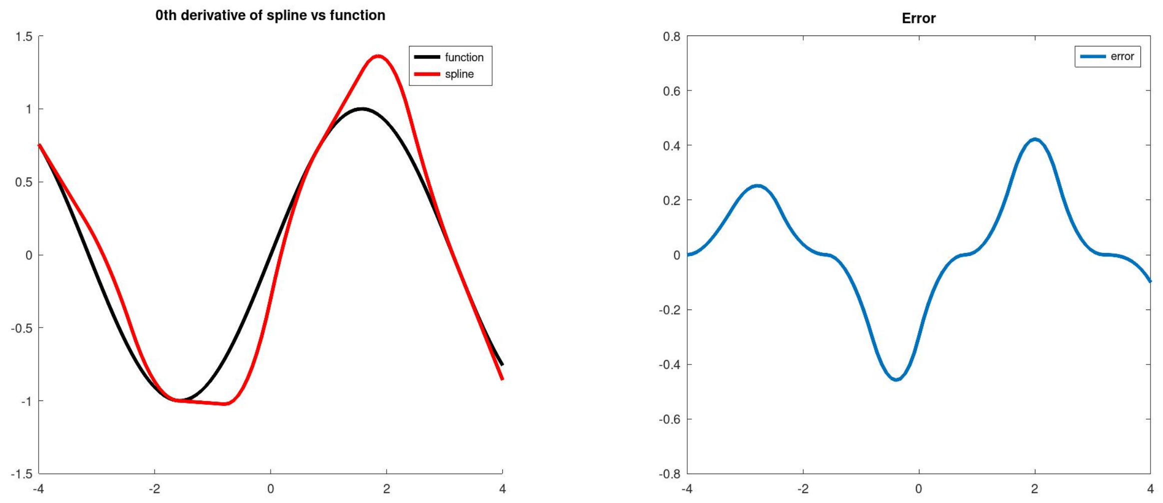


Figure 2: Quadratic C-spline that matches the degree 1 Taylor polynomial of a sine function once every 3 panels, 10 panels in total

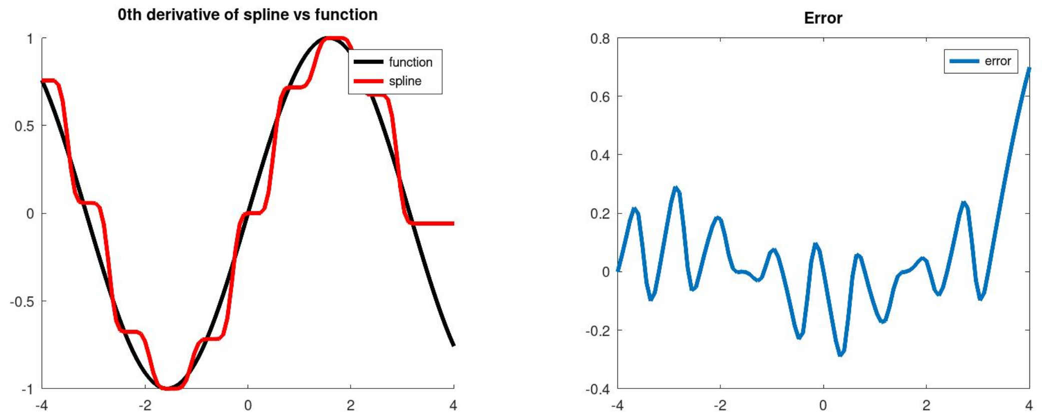


Figure 3: Cubic C-spline that matches the degree 0 Taylor polynomial of a sine function once every 4 panels, 40 panels in total

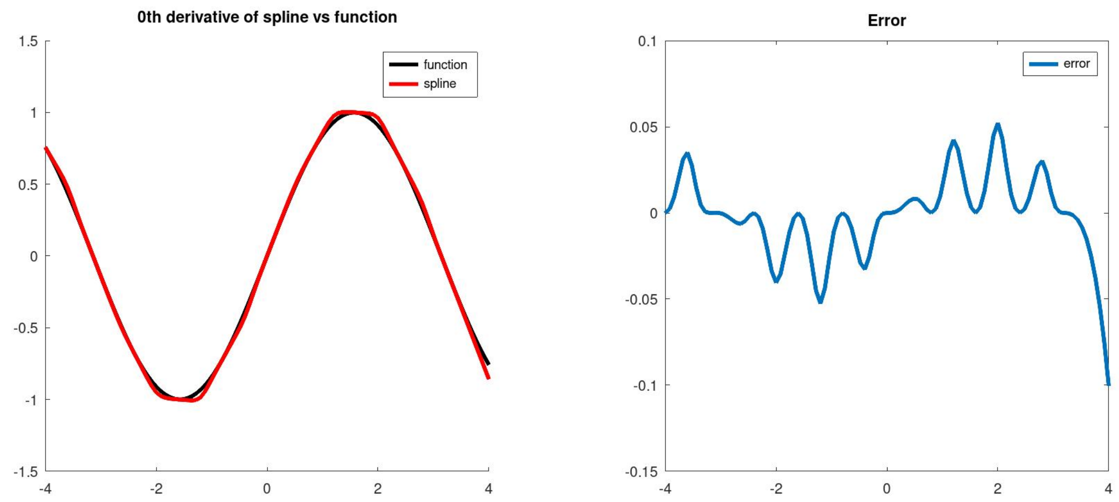


Figure 4: Cubic C-spline that matches the degree 1 Taylor polynomial of a sine function once every 4 panels, 40 panels in total

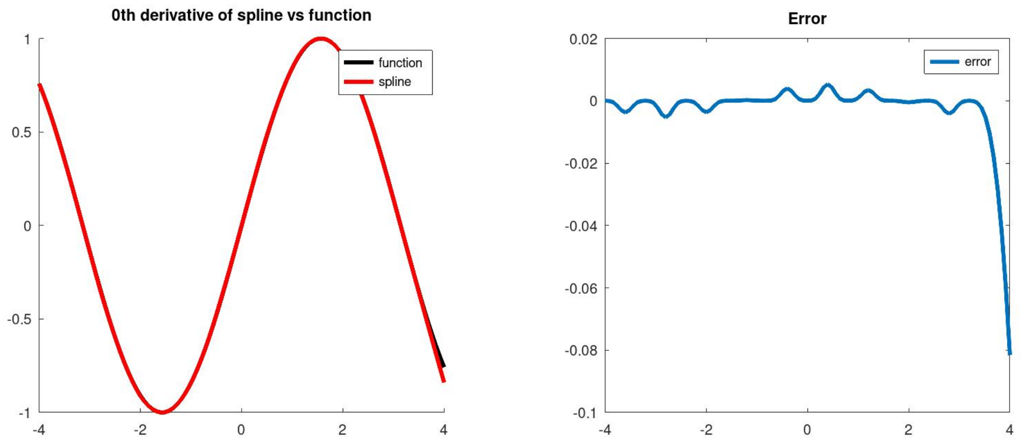


Figure 5: Cubic C-spline that matches the degree 2 Taylor polynomial of a sine function once every 4 panels, 40 panels in total

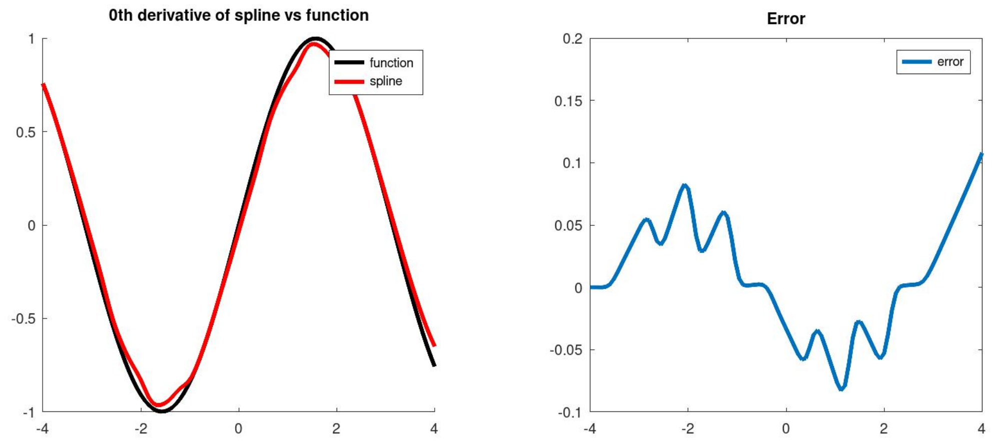


Figure 6: Cubic C-spline that matches the degree 3 Taylor polynomial of a sine function once every 4 panels, 40 panels in total

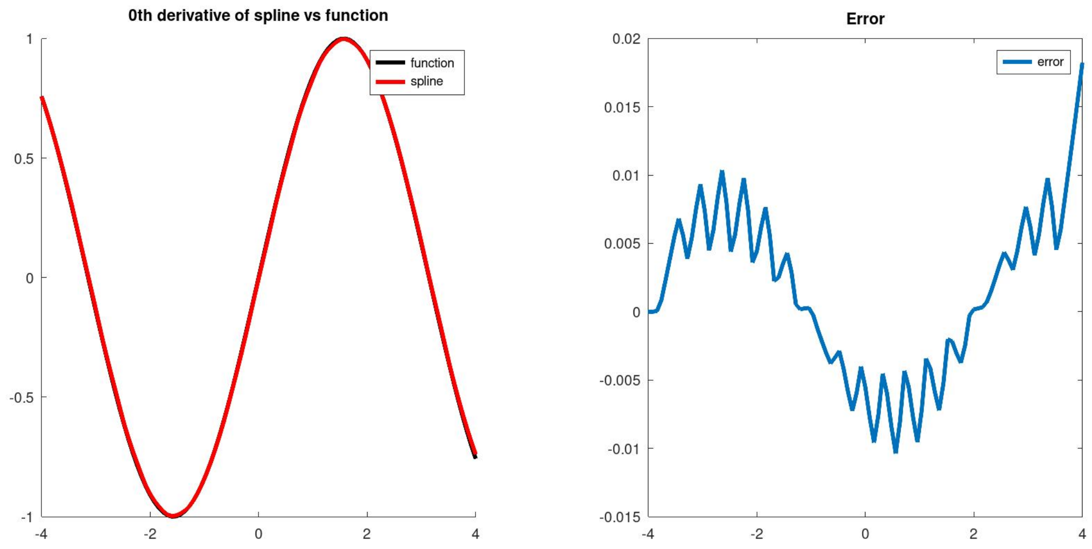


Figure 7: Cubic C-spline that matches the degree 3 Taylor polynomial of a sine function once every 4 panels, 80 panels in total

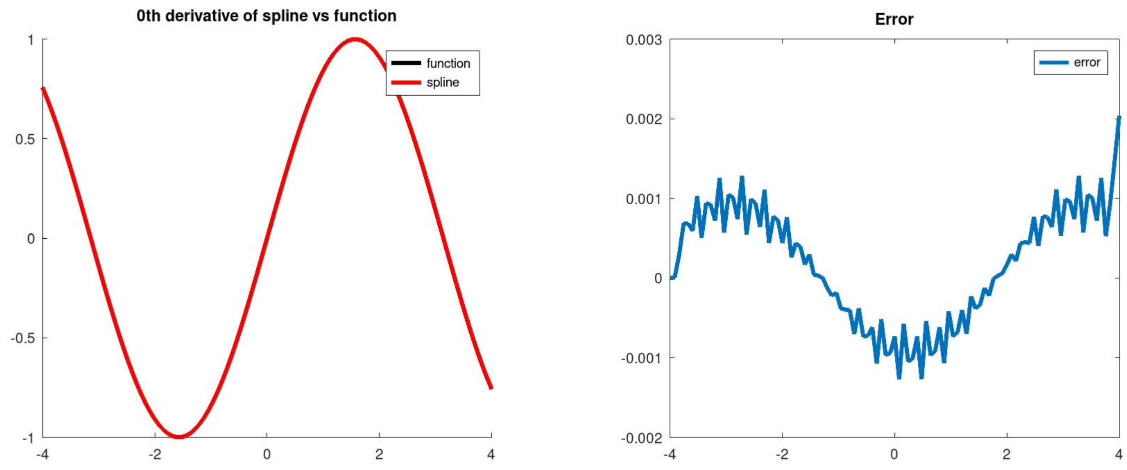


Figure 8: Cubic C-spline that matches the degree 3 Taylor polynomial of a sine function once every 4 panels, 160 panels in total

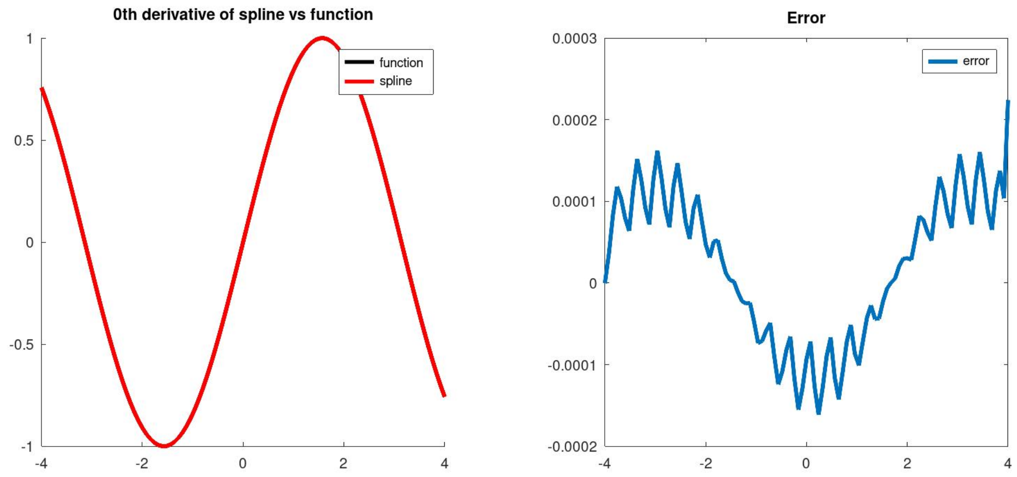


Figure 9: Cubic C-spline that matches the degree 3 Taylor polynomial of a sine function once every 4 panels, 320 panels in total

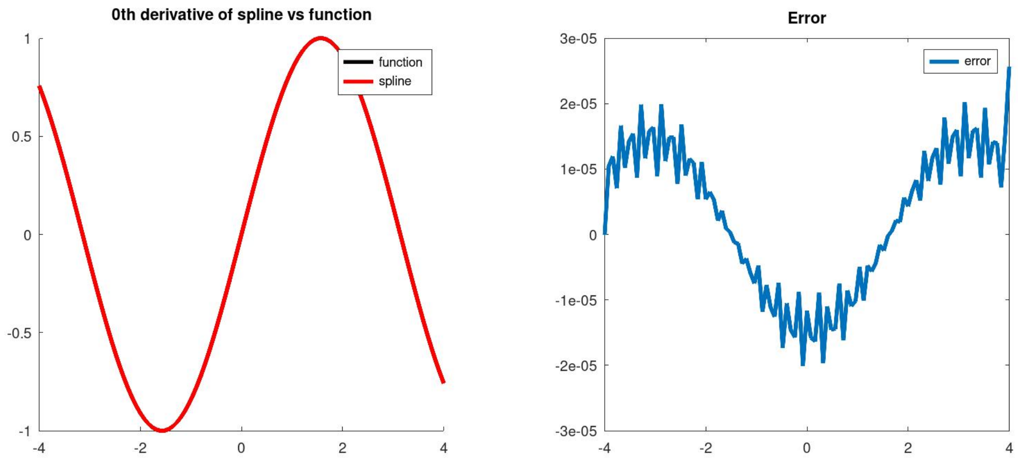


Figure 10: Cubic C-spline that matches the degree 3 Taylor polynomial of a sine function once every 4 panels, 640 panels in total

5.2 Exponential C-splines

Description

In order to demonstrate how close exponential splines can get to functions they approximate (in the L2 sense), we will find at least one candidate exponential spline that is sufficiently close. This will be accomplished with exponential C-splines, which are the exponential variant of the standard C-splines just described.

Exponential C-splines are exponential splines (in the sense defined above) with polynomial part of degree D, that match the 0th through $(d+2)$th derivative of f periodically every $(D+3)$ many panels, where $d \leq D$.

Equivalently, if we expand out the two exponential terms appearing in each spline panel as

$$\exp\left(\rho_j \frac{t_{j-1}-t}{t_j - t_{j-1}}\right) = 1 + \rho_j \left(\frac{t_{j-1}-t}{t_j - t_{j-1}}\right) + \frac{\rho_j^2}{2!}\left(\frac{t_{j-1}-t}{t_j - t_{j-1}}\right)^2 + \cdots \tag{50}$$

and

$$\exp\left(\rho_j \frac{t-t_j}{t_j-t_{j-1}}\right) = 1 + \rho_j \left(\frac{t-t_j}{t_j-t_{j-1}}\right) + \frac{\rho_j^2}{2!}\left(\frac{t-t_j}{t_j-t_{j-1}}\right)^2 + \cdots \tag{51}$$

then we are just ensuring that every $(D+3)$ many panels, our exponential spline approximation to f correctly matches the terms up to order $(D+2)$ of the degree $(d+2)$ Taylor polynomial to f.

Properties of exponential C-splines

The properties of these splines are essentially identical to those of standard C-splines, with the additional benefit that we can select tension parameters $(\rho_j)_{j=1}^M$ to be high on panels where f induces oscillatory behavior in the approximation and low on panels where f doesn't induce oscillatory behavior. The representation being used is

$$\begin{aligned} S_j(t) = a_j^0 + a_j^1\left(\frac{t-t_{j-1}}{h_j}\right) + a_j^2\left(\frac{t-t_{j-1}}{h_j}\right) + \cdots + a_j^D\left(\frac{t-t_{j-1}}{h_j}\right)^D \\ + a_j^{D+1}\exp\left(\rho_j\frac{t_{j-1}-t}{h_j}\right) + a_j^{D+2}\exp\left(\rho_j\frac{t-t_j}{h_j}\right) \end{aligned} \tag{52}$$

Generating exponential C-splines and approximating f

The ith derivative of this approximation is given by

$$\begin{aligned} S_j^{(i)}(t) = a_j^i\frac{i!}{(h_j)^i} + a_j^{i+1}\frac{(i+1)!}{1!(h_j)^i}\left(\frac{t-t_{j-1}}{h_j}\right) + a_j^{i+2}\frac{(i+2)!}{2!(h_j)^i}\left(\frac{t-t_{j-1}}{h_j}\right)^2 + \cdots \\ + a_j^D\frac{(D)!}{(D-i)!(h_j)^i}\left(\frac{t-t_{j-1}}{h_j}\right)^{D-i} + a_j^{D+1}\left(\frac{-\rho_j}{h_j}\right)^i\exp\left(\rho_j\frac{t_{j-1}-t}{h_j}\right) \\ + a_j^{D+2}\left(\frac{\rho_j}{h_j}\right)^i\exp\left(\rho_j\frac{t-t_j}{h_j}\right) \end{aligned} \tag{53}$$

which implies, at the panel endpoints,

$$S_j^{(i)}(t_{j-1}) = a_j^i\frac{i!}{(h_j)^i} + a_j^{D+1}\left(\frac{-\rho_j}{h_j}\right)^i + a_j^{D+2}\left(\frac{\rho_j}{h_j}\right)^i\exp(-\rho_j) \tag{54}$$

and

$$\begin{aligned} S_j^{(i)}(t_j) = {} & a_j^i \frac{i!}{(h_j)^i} + a_j^{i+1} \frac{(i+1)!}{1!(h_j)^i} + a_j^{i+2} \frac{(i+2)!}{2!(h_j)^i} + \cdots + a_j^D \frac{(D)!}{(D-i)!(h_j)^i} + \\ & a_j^{D+1} \left(\frac{-\rho_j}{h_j} \right)^i \exp(-\rho_j) + a_j^{D+2} \left(\frac{\rho_j}{h_j} \right)^i \end{aligned} \tag{55}$$

This makes the derivative continuity condition across node t_j

$$\begin{aligned} & a_j^i \frac{i!}{(h_j)^i} + a_j^{i+1} \frac{(i+1)!}{1!(h_j)^i} + a_j^{i+2} \frac{(i+2)!}{2!(h_j)^i} + \cdots + a_j^D \frac{(D)!}{(D-i)!(h_j)^i} + a_j^{D+1} \left(\frac{-\rho_j}{h_j} \right)^i \exp(-\rho_j) + a_j^{D+2} \left(\frac{\rho_j}{h_j} \right)^i \\ & = S_j^{(i)}(t_j) \\ & = S_{j+1}^{(i)}(t_j) \\ & = a_{j+1}^i \frac{i!}{(h_{j+1})^i} + a_{j+1}^{D+1} \left(\frac{-\rho_{j+1}}{h_{j+1}} \right)^i + a_{j+1}^{D+2} \left(\frac{\rho_{j+1}}{h_{j+1}} \right)^i \exp(-\rho_{j+1}) \end{aligned} \tag{56}$$

for $i = 0, 1, \ldots, D+1$. Also, once every $(D+3)$ many panels, we will want to enforce the matching of the Taylor polynomial, leading to the constraints

$$\begin{aligned} & a_j^i \frac{i!}{(h_j)^i} + a_j^{D+1} \left(\frac{-\rho_j}{h_j} \right)^i + a_j^{D+2} \left(\frac{\rho_j}{h_j} \right)^i \exp(-\rho_j) \\ & = S_j^{(i)}(t_{j-1}) \\ & = f^{(i)}(t_{j-1}) \end{aligned} \tag{57}$$

for $i = 0, 1, \ldots, d+2$, and

$$\begin{aligned} & a_j^i \frac{i!}{(h_j)^i} + a_j^{D+1} \left(\frac{-\rho_j}{h_j} \right)^i + a_j^{D+2} \left(\frac{\rho_j}{h_j} \right)^i \exp(-\rho_j) \\ & = S_j^{(i)}(t_{j-1}) \\ & = 0 \end{aligned} \tag{58}$$

for $i = d+3, d+4, \ldots, D+2$. This will ensure that S_j is the degree $(d+2)$ Taylor polynomial of f centered at t_{j-1}, correct up to order $(D+2)$.

At this point, we could proceed in one of two ways. We could parametrize the last coefficient on each intermediate spline panel, set up a linear system to solve for them, and then use an update equation (determined by the derivative continuity constraints) to update the remaining coefficients over each intermediate spline panel. This is an awkward construction, since the last coefficient of each spline panel doesn't have the same simple interpretation as that of standard C-splines. Instead, we construct a linear system and solve for all coefficients of the exponential spline in a batch of $(D+3)$ panels. This results in a banded system whose solution has similar computational complexity to that of the standard C-spline procedure, provided an optimal band solver is employed to solve the banded linear system.

The smoothness constraints on the exponential spline, across node t_j, result in the linear system

$$L_{j+1}a_{j+1} = R_i a_j \tag{59}$$

where $a_j \in \mathbb{R}^{D+3}$ houses the $(D+3)$ spline coefficients over the jth panel, i.e.

$$a_j = \begin{bmatrix} a_j^0 \\ a_j^1 \\ \vdots \\ a_j^D \\ a_j^{D+1} \\ a_j^{D+2} \end{bmatrix}$$

and L_{j+1} and R_j are representations of the smoothness constraints above, so

$$L_{j+1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & e^{-\rho_{j+1}} \\ 0 & 1 & 0 & \cdots & 0 & \frac{-\rho_{j+1}}{1!} & \frac{\rho_{j+1}}{1!} e^{-\rho_{j+1}} \\ 0 & 0 & 1 & \cdots & 0 & \frac{(-\rho_{j+1})^2}{2!} & \frac{(\rho_{j+1})^2}{2!} e^{-\rho_{j+1}} \\ & & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{(-\rho_{j+1})^D}{D!} & \frac{(\rho_{j+1})^D}{D!} e^{-\rho_{j+1}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+1})^{D+1}}{(D+1)!} & \frac{(\rho_{j+1})^{D+1}}{(D+1)!} e^{-\rho_{j+1}} \end{bmatrix}$$

and

$$R_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{0}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{1}{0} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \frac{(-\rho_j)^0}{0!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^0 \frac{(\rho_j)^0}{0!} \\ & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{1}{1} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D}{1} & \left(\frac{h_{j+1}}{h_j}\right)^1 \frac{(-\rho_j)^1}{1!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^1 \frac{(\rho_j)^1}{1!} \\ & & \ddots & & \vdots & \\ & & & \left(\frac{h_{j+1}}{h_j}\right)^D \binom{D}{D} & \left(\frac{h_{j+1}}{h_j}\right)^D \frac{(-\rho_j)^D}{D!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^D \frac{(\rho_j)^D}{D!} \\ & & & & \left(\frac{h_{j+1}}{h_j}\right)^{D+1} \frac{(-\rho_j)^{D+1}}{(D+1)!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^{D+1} \frac{(\rho_j)^{D+1}}{(D+1)!} \end{bmatrix}$$

It is important to note that both L_{j+1} and R_j are in $\mathbb{R}^{(D+2)\times(D+3)}$.

For exponential C-splines, we will compute the coefficient vectors a_j in blocks of $(D+3)$ at a time. So, given a_j, we will set up a large linear system for $a_{j+1}, \ldots, a_{j+D+3}$ so that a_{j+D+3} causes the exponential spline to match the degree $(d+2)$ Taylor polynomial (to order $(D+2)$) on the $(J+D+3)$th panel.

The large linear system to solve for the block of coefficient vectors $a_{j+1}, \dots, a_{j+D+3}$ is:

$$\underbrace{\begin{bmatrix} \mathtt{I}_{\mathtt{D+3}} & & & & & & \\ -R_j & L_{j+1} & & & & & \\ & -R_{j+1} & L_{j+2} & & & & \\ & & -R_{j+2} & L_{j+3} & & & \\ & & & \ddots & & & \\ & & & & -R_{j+D+2} & L_{j+D+3} & \\ & & & & & \tilde{L}_{j+D+3} & \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} a_j^0 \\ \vdots \\ a_j^{D+2} \\ \\ \vdots \\ \\ a_{j+D+3}^0 \\ \vdots \\ a_{j+D+3}^{D+2} \end{bmatrix}}_{\vec{a}} = \underbrace{\begin{bmatrix} a_j^0 \\ \vdots \\ a_j^{D+2} \\ 0 \\ \vdots \\ 0 \\ (h_{j+D+3})^0 f(t_{j+D+2})/0! \\ \vdots \\ (h_{j+D+3})^{D+2} f^{(D+2)}(t_{j+D+2})/(D+2)! \end{bmatrix}}_{\vec{b}} \tag{60}$$

where

$$\tilde{L}_{j+D+3} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & e^{-\rho_{j+D+3}} \\ 0 & 1 & 0 & \cdots & 0 & \frac{-\rho_{j+D+3}}{1!} & \frac{\rho_{j+D+3}}{1!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 1 & \cdots & 0 & \frac{(-\rho_{j+D+3})^2}{2!} & \frac{(\rho_{j+D+3})^2}{2!} e^{-\rho_{j+D+3}} \\ & & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{(-\rho_{j+D+3})^D}{D!} & \frac{(\rho_{j+D+3})^D}{D!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+D+3})^{D+1}}{(D+1)!} & \frac{(\rho_{j+D+3})^{D+1}}{(D+1)!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+D+3})^{D+2}}{(D+2)!} & \frac{(\rho_{j+D+3})^{D+2}}{(D+2)!} e^{-\rho_{j+D+3}} \end{bmatrix} \in \mathbb{R}^{(D+3)\times(D+3)}$$

and where for simplicity's sake, we assumed $d = D$. If $d < D$, then we just replace $f^{(k)}(t_{j+D+2})$ with 0 for $k = d+3, \dots, D+2$ in the vector on the right hand side.

In the equation directly above, we already know $a_j^0, \dots, a_j^{D+2}$. The large linear system is square and of size $\mathbb{R}^{(D+4)(D+3)\times(D+4)(D+3)}$. It is an elementary procedure to perform Gaussian elimination on this system to advance it to lower triangular form with nonzero diagonal entries, at which point we may conclude it is invertible. The calculation is omitted.

The overall procedure for approximating f via exponential C-splines is:

(1) Start with any set of coefficients $a_1 \in \mathbb{R}^{D+1}$ on the first panel $[t_0, t_1]$.

(2) Use the equation $E\vec{a} = \vec{b}$ to solve for the block of spline coefficients $a_{j+1}, \ldots, a_{j+D+3}$

(3) Repeat the process over again, cycling through steps (1) through (2) until there are fewer than $(D+3)$ panels to the right

(4) Once we can no longer continue the process due to not having enough panels to the right, we choose the coefficients of the remaining panels so that the $(D+2)$th derivative of the spline and the function match up at the nodes $t_j, t_{j+1}, \ldots, t_{M-1}$. Note: if $d < D$, then we may instead treat these derivatives of f as 0.

This gives us the exponential C-spline approximation to f.

In the next section, we discuss the crucial convergence results pertaining to these splines.

In figures (11) through (18), we give some salient examples of exponential C-splines to demonstrate the properties discussed above. The first figure shows how an exponential C-spline with linear polynomial part can periodically match the 0th degree Taylor polynomial of an exponential function. We see this manifest in periodic regions where the spline is flat. As we increase the degree of the Taylor polynomial matched, the spline becomes more accurate, as evidenced by the shrinking error. Once we reach the maximum degree Taylor polynomial that our exponential spline (namely, the degree of the spline's polynomial part, plus two), we may further shrink the error by refining the panel width, as is done in the last few figures. It is important to note two observations with these graphs: (1) due to the semi-local nature of these splines, and the exponential growth of the underlying exponential function they are approximating, we expect the error curves to look periodic but increasing exponentially in scale; and (2) due to the crude approximation scheme used when there are not enough panels to the right to match the next Taylor polynomial, we expect the errors to magnify towards the right endpoint. Both of these phenomena are demonstrated in these examples. It should also be noted that even though we are using exponential terms in our spline approximation, this does not mean the approximation will be exact, since the coefficients of the exponents are different.

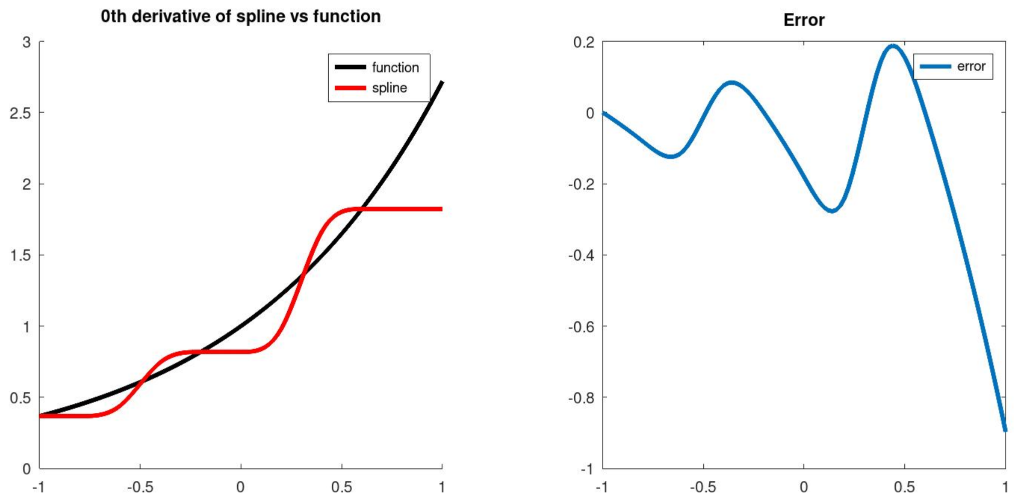


Figure 11: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 0 Taylor polynomial of an exponential function once every 4 panels, 10 panels in total. Uniform panel width, uniform tensions equal to 1.

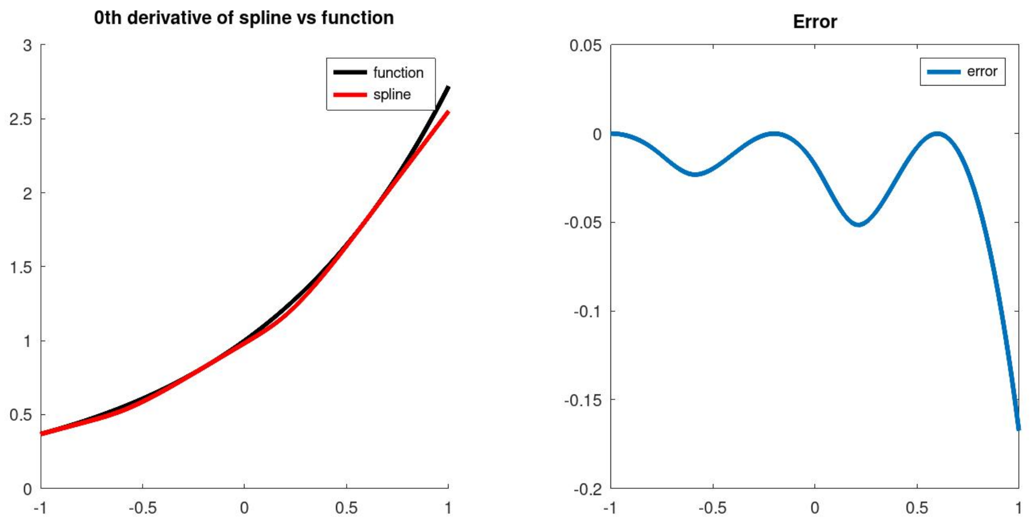


Figure 12: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 1 Taylor polynomial of an exponential function once every 4 panels, 10 panels in total. Uniform panel width, uniform tensions equal to 1.

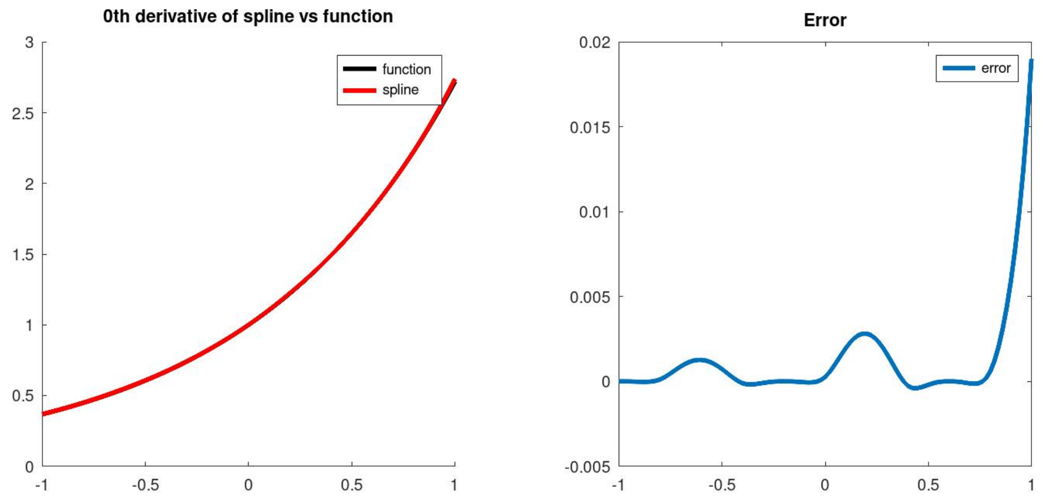


Figure 13: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 2 Taylor polynomial of an exponential function once every 4 panels, 10 panels in total. Uniform panel width, uniform tensions equal to 1.

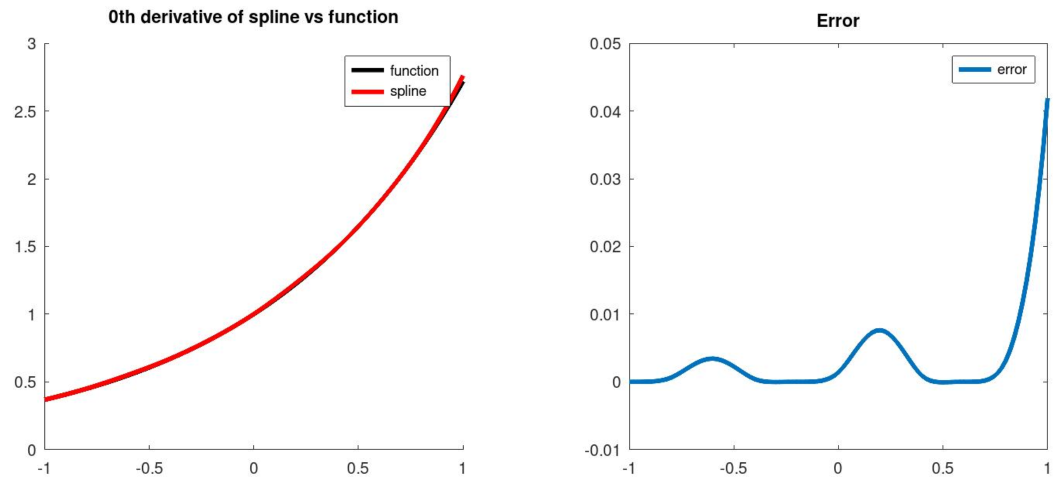


Figure 14: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 3 Taylor polynomial of an exponential function once every 4 panels, 10 panels in total. Uniform panel width, uniform tensions equal to 1.

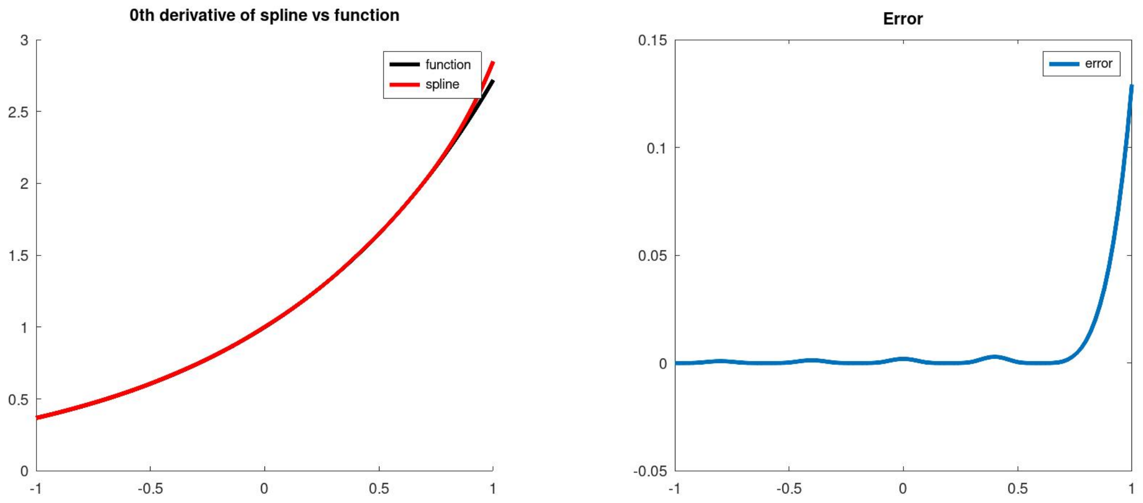


Figure 15: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 3 Taylor polynomial of an exponential function once every 4 panels, 20 panels in total. Uniform panel width, uniform tensions equal to 1.

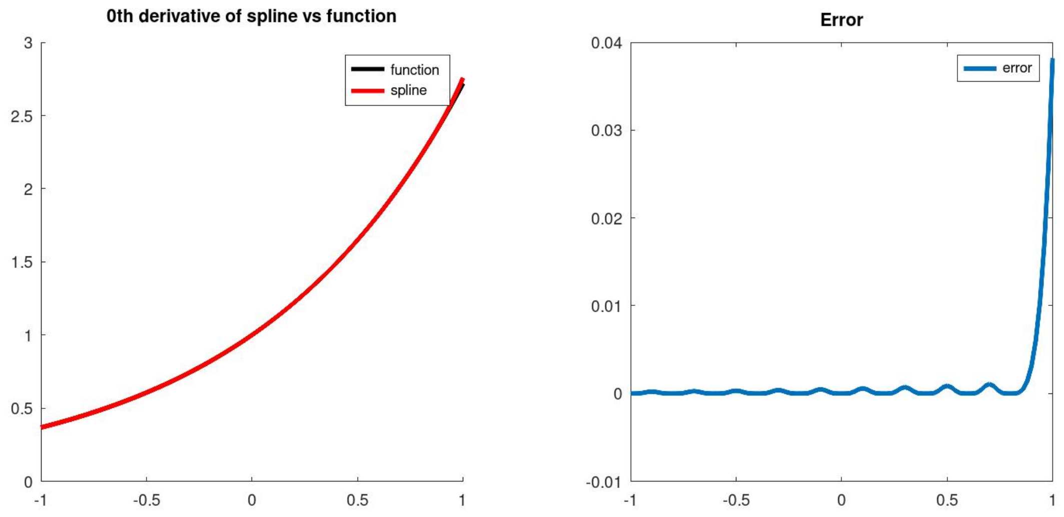


Figure 16: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 3 Taylor polynomial of an exponential function once every 4 panels, 40 panels in total. Uniform panel width, uniform tensions equal to 1.

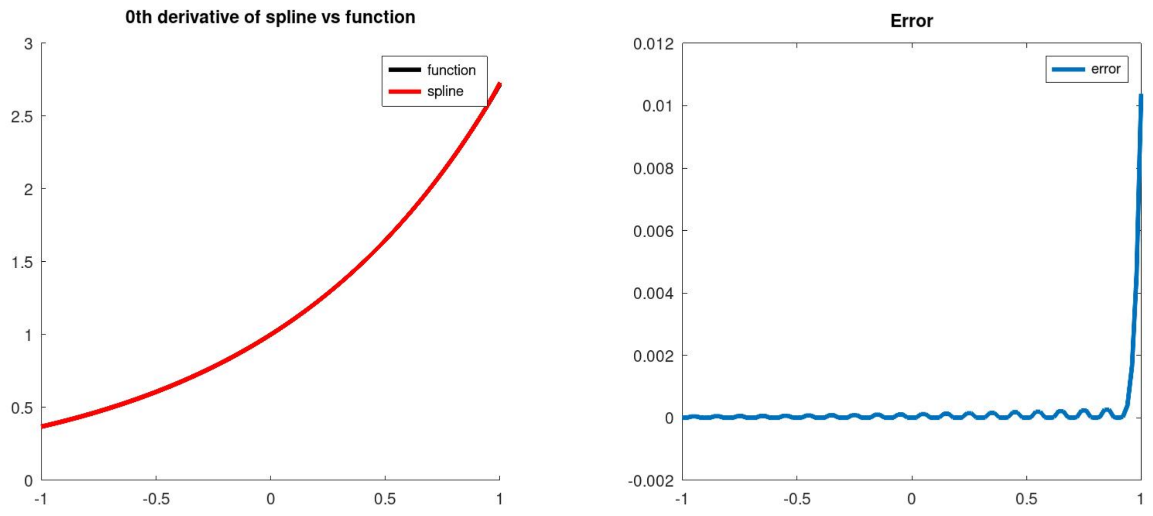


Figure 17: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 3 Taylor polynomial of an exponential function once every 4 panels, 80 panels in total. Uniform panel width, uniform tensions equal to 1.

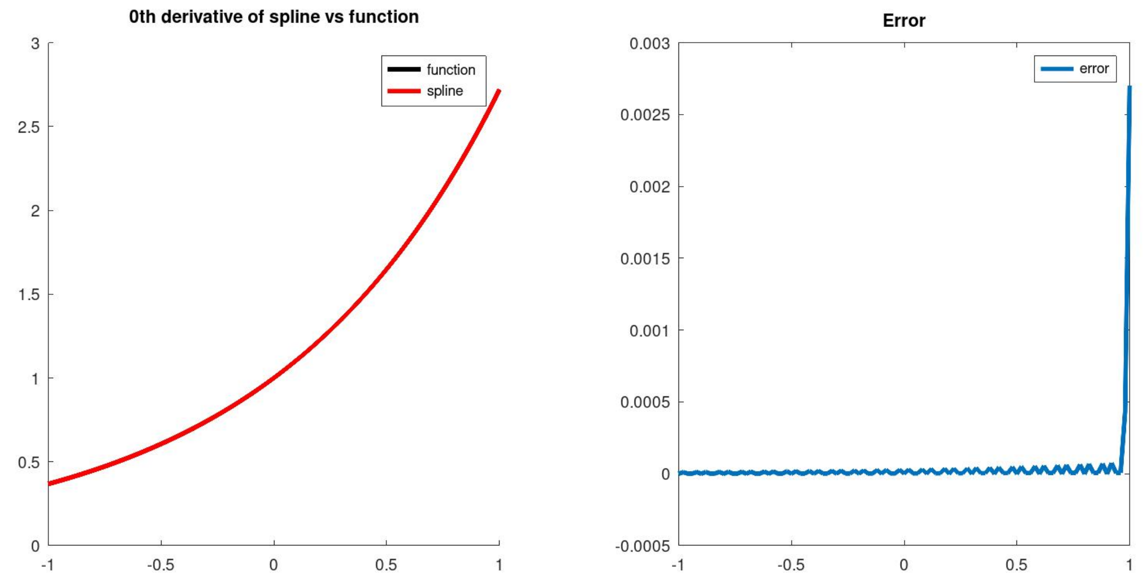


Figure 18: Cubic exponential C-spline (i.e., with polynomial part of degree 1) that matches the degree 3 Taylor polynomial of an exponential function once every 4 panels, 160 panels in total. Uniform panel width, uniform tensions equal to 1.

6 Convergence results

6.1 Main convergence result for A-splines

In this section we develop error bounds for the determination of an approximation to a function f obtained by projecting f onto an orthonormal basis of the spline space $\mathcal{S}_D$. We assume the function to be approximated, f, is orthogonally projected onto the relevant spline space, $\mathcal{S}_D$, without error.

We derive an error bound which shows that for any continuously differentiable function f, the spline operator produces approximations of f that are more and more accurate as $h \searrow 0$ (in the absence of roundoff error), and one obtains higher order convergence with h the more continuous derivatives f has.

The L2 error bound derived is:

$$||f - S(f)||_2 \leq \frac{(D+1)!}{2^j \cdot (D-j)!} \cdot (\Delta t)^{j+1} \cdot \sqrt{t_M - t_0} \cdot ||f^{(j+1)}||_\infty, \qquad j = 0, 1, \ldots, D \tag{61}$$

with the slight improvement of an additional factor of 2 in the denominator when $j = D$.

Here, S is the spline operator that orthogonally projects functions onto the spline space $\mathcal{S}_D$, in the L2 sense (i.e., $\forall s \in \mathcal{S}_D$ we have $S(s) = s$, and for any function ϕ we have $S(\phi) \perp (\phi - S(\phi))$ in the L2 sense). The proof of the bound is based on several theorems from Carl deBoor's celebrated book on splines (see the Thm. 12.1 Jackson type, from chapter 12), as well as elementary computations involving L2 projections. Since the proof relies on some delicate facts regarding B-splines, we first give a lemma containing all the needed facts about B-splines before proving the main convergence result.

Given the knot sequence $t_0 < t_1 < t_2 < \cdots < t_M$, we can define the family of B-splines of degree D with respect to these knots, via a simple recursive formula. In the recursive formula, B-splines of degree k will be linear combinations of two B-splines of degree $k-1$, one associated with the same panel, and one associated with the panel one place to the right. In order to make the recursive definition well-defined, we need to append D additional degenerate panels at the right endpoint, so we set $t_{M+1} = \cdots = t_{M+D} = t_M$. In order to have our family of B-splines span all of $\mathcal{S}_D$, we will also need to append D additional degenerate panels at the left endpoint, so we set $t_{-D} = \cdots = t_{-1} = t_0$. With these additional knots specified, we are ready to give the recursive definition of the B-spline family of degree D. In the process of obtaining this family, we will also generate B-spline families of all lower degree with respect to the same knot sequence.

The following recursive definition of B-splines is known as the Cox-de Boor formula. Although it is not the definition of B-splines that de Boor gave in his seminal work in the field (he used divided differences for the definition), it is quicker to demonstrate the desired properties using this definition, even if it partly hides some intuition behind the construction. Define:

$$B_j^0(x) = \begin{cases} 1 & t_j \le x < t_{j+1} \\ 0 & \text{else} \end{cases} \quad , \quad j = -D, \ldots, M+D-1 \tag{62}$$

$$B_j^k(x) = \frac{x - t_j}{t_{j+k} - t_j} B_j^{k-1}(x) + \frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} B_{j+1}^{k-1}(x), \quad k = 1, \ldots, D, \quad j = -D, \ldots, M+D-k-1 \tag{63}$$

Then for each $k = 0, 1, \ldots D$, we say $\{B_j^k\}_{j=-k}^{M-1}$ is the B-spline family of degree k, with respect to the knot sequence $t_0 < t_1 < t_2 < \cdots < t_M$.

The following is a lemma establishing the properties of B-splines we'll use in the error bound for A-splines.

Lemma 6.1 (B-spline properties)**.** *Let the B-spline construction be as above. Then the following properties hold for all $k = 0, 1, \ldots, D$ and all $j = -k, \ldots, M-1$:*

(1) each B_j^k is nonnegative

(2) each B_j^k has support $[t_j, t_{j+k+1}]$

(3) each B_j^k is positive on the interior of its support

(4) for all $x \in [t_0, t_M]$, we have $\sum_j B_j^k(x) = 1$ (that is, $\{B_j^k\}_j$ is a partition of unity)

(5) each B_j^k is a member of S_k (that is, B_j^k is a C^{k-1}-smooth degree k piecewise polynomial)

Proof. By the construction, each spline B_j^k gains one more panel of support on the right side, due to the presence of the B_{j+1}^{k-1} term in the recursive definition. Since each B_j^0 has support over the single panel $[t_j, t_{j+1}]$, we get that each B_j^k has support contained in $[t_j, t_{j+k+1}]$.

From the recursive definition of the B-splines, we can use an induction argument to see that each B-spline is positive on the interior of its support. This is clearly true of B_j^0 for all j. Suppose it is true for B-splines of degree $k-1$. Then since B_j^{k-1} has support on $[t_j, t_{j+k}]$, the term $\frac{x - t_j}{t_{j+k} - t_j} B_j^{k-1}(x)$ is positive for $x \in (t_j, t_{j+k})$ and 0 everywhere else. Similarly, $\frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} B_{j+1}^{k-1}(x)$ is positive for $x \in (t_{j+1}, t_{j+k+1})$ and 0 everywhere else. Adding these terms together, we get that B_j^k is positive on (t_j, t_{j+k+1}) and 0 everywhere else, completing the inductive step. This establishes properties (1), (2), and (3).

Also by this construction, for all $x \in [a, b]$ we have

$$\sum_j B_j^0(x) = 1 \tag{64}$$

To show that the B-splines B_j^k of each degree k sum to 1 for all $x \in [a, b]$, we appeal to the recursive definition, as follows:

$$\sum_j B_j^k(x) = \sum_j \left[\frac{x-t_j}{t_{j+k}-t_j} B_j^{k-1}(x) + \frac{t_{j+k+1}-x}{t_{j+k+1}-t_{j+1}} B_{j+1}^{k-1}(x) \right] \tag{65}$$

$$= \sum_j \frac{x-t_j}{t_{j+k}-t_j} B_j^{k-1}(x) + \sum_j \frac{t_{j+k+1}-x}{t_{j+k+1}-t_{j+1}} B_{j+1}^{k-1}(x) \tag{66}$$

$$= \sum_j \frac{x-t_j}{t_{j+k}-t_j} B_j^{k-1}(x) + \sum_j \frac{t_{j+k}-x}{t_{j+k}-t_j} B_j^{k-1}(x) \qquad \text{(index shift)} \tag{67}$$

$$= \sum_j \frac{t_{j+k}-t_j}{t_{j+k}-t_j} B_j^{k-1}(x) \tag{68}$$

$$= \sum_j B_j^{k-1}(x) \tag{69}$$

at which point we may argue by induction, having already verified the base case. It is important to note here that the index shift is valid, since the extra terms gained and lost at a and b respectively do not contribute to the sum, as those B-splines have a degenerate interval as their support. Thus,

$$\forall k = 0, 1, \dots, D, \quad \forall x \in [a, b], \quad \sum_j B_j^k(x) = 1 \tag{70}$$

and property (4) is established.

Since each B_j^0 is a 0th degree polynomial, and the factors multiplying B_j^{k-1} and B_{j+1}^{k-1} in the recursive definition for B_j^k are linear, we can see by an induction argument that the degree of each polynomial part of each B_j^k is $\leq k$. In fact, we will now show by induction that each B-spline of degree k has the following explicit form:

$$B_j^k(x) = (t_{j+k+1} - t_j) \sum_{i=j}^{j+k+1} \frac{(x-t_i)_+^k}{\prod_{l=j,\ l\neq i}^{j+k+1} (t_l - t_i)} \tag{71}$$

where

$$(x - t_i)_+^k = \begin{cases} (x - t_i)^k & x \geq t_i \\ 0 & x < t_i \end{cases} \tag{72}$$

For reference, this form was found in the notes by Michael S. Floater with the University of Oslo, on pg. 53 of chapter 2 of his notes for Math 5340. Another way to arrive at this form is to use the divided difference definition of B-splines introduced by de Boor, and then convert the divided difference to an explicit form.

For $k = 0$, the explicit form is

$$(t_{j+1} - t_j) \sum_{i=j}^{j+1} \frac{(x - t_i)_+^0}{\prod_{l=j,\ l\neq i}^{j+1} (t_l - t_i)} = (t_{j+1} - t_j) \left(\frac{(x - t_j)_+^0}{t_{j+1} - t_j} + \frac{(x - t_{j+1})_+^0}{t_j - t_{j+1}} \right) \tag{73}$$

$$= \begin{cases} 1 & t_j \leq x < t_{j+1} \\ 0 & \text{else} \end{cases} \tag{74}$$

$$= B_j^0(x) \tag{75}$$

which verifies the base case. For the inductive step, assume the explicit form holds for all B-splines of degree $k - 1$. Then by the recursive definition for B-splines, we have

$$B_j^k(x) = \frac{x - t_j}{t_{j+k} - t_j} B_j^{k-1}(x) + \frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} B_{j+1}^{k-1}(x) \tag{76}$$

$$= \frac{x - t_j}{t_{j+k} - t_j} (t_{j+k} - t_j) \sum_{i=j}^{j+k} \frac{(x - t_i)_+^{k-1}}{\prod_{l=j,\ l\neq i}^{j+k} (t_l - t_i)} \tag{77}$$

$$+ \frac{t_{j+k+1} - x}{t_{j+k+1} - t_{j+1}} (t_{j+k+1} - t_{j+1}) \sum_{i=j+1}^{j+k+1} \frac{(x - t_i)_+^{k-1}}{\prod_{l=j+1,\ l\neq i}^{j+k+1} (t_l - t_i)}$$

$$= (x-t_j)\sum_{i=j}^{j+k}\frac{(x-t_i)_+^{k-1}}{\prod\limits_{l=j,\ l\neq i}^{j+k}(t_l-t_i)} + (t_{j+k+1}-x)\sum_{i=j+1}^{j+k+1}\frac{(x-t_i)_+^{k-1}}{\prod\limits_{l=j+1,\ l\neq i}^{j+k+1}(t_l-t_i)} \tag{78}$$

$$= \frac{(x-t_j)(x-t_j)_+^{k-1}}{\prod\limits_{l=j+1}^{j+k}(t_l-t_j)} + (x-t_j)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_j-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)} \tag{79}$$

$$- \frac{(x-t_{j+k+1})(x-t_{j+k+1})_+^{k-1}}{\prod\limits_{l=j+1}^{j+k}(t_l-t_{j+k+1})} + (t_{j+k+1}-x)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_{j+k+1}-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)}$$

$$= \frac{(x-t_j)_+^{k}}{\prod\limits_{l=j+1}^{j+k}(t_l-t_j)} + (x-t_j)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_j-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)} \tag{80}$$

$$- \frac{(x-t_{j+k+1})_+^{k}}{\prod\limits_{l=j+1}^{j+k}(t_l-t_{j+k+1})} + (t_{j+k+1}-x)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_{j+k+1}-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)}$$

$$= (t_{j+k+1}-t_j)\frac{(x-t_j)_+^{k}}{\prod\limits_{l=j+1}^{j+k+1}(t_l-t_j)} + (x-t_j)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_j-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)} \tag{81}$$

$$+ (t_{j+k+1}-t_j)\frac{(x-t_{j+k+1})_+^{k}}{\prod\limits_{l=j}^{j+k}(t_l-t_{j+k+1})} + (t_{j+k+1}-x)\sum_{i=j+1}^{j+k}\frac{(x-t_i)_+^{k-1}}{(t_{j+k+1}-t_i)\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)}$$

$$= (t_{j+k+1}-t_j)\frac{(x-t_j)_+^{k}}{\prod\limits_{l=j+1}^{j+k+1}(t_l-t_j)} + (t_{j+k+1}-t_j)\frac{(x-t_{j+k+1})_+^{k}}{\prod\limits_{l=j}^{j+k}(t_l-t_{j+k+1})} \tag{82}$$

$$+ \sum_{i=j+1}^{j+k}\frac{\left(\dfrac{x-t_j}{t_j-t_i}+\dfrac{t_{j+k+1}-x}{t_{j+k+1}-t_i}\right)(x-t_i)_+^{k-1}}{\prod\limits_{l=j+1,\ l\neq i}^{j+k}(t_l-t_i)}$$

$$= (t_{j+k+1}-t_j)\frac{(x-t_j)_+^{k}}{\prod\limits_{l=j+1}^{j+k+1}(t_l-t_j)} + (t_{j+k+1}-t_j)\frac{(x-t_{j+k+1})_+^{k}}{\prod\limits_{l=j}^{j+k}(t_l-t_{j+k+1})} \tag{83}$$

$$+ \sum_{i=j+1}^{j+k}\frac{\left((x-t_j)(t_{j+k+1}-t_i)+(t_{j+k+1}-x)(t_j-t_i)\right)(x-t_i)_+^{k-1}}{\prod\limits_{l=j,\ l\neq i}^{j+k+1}(t_l-t_i)}$$

$$= (t_{j+k+1} - t_j)\frac{(x-t_j)_+^k}{\prod_{l=j+1}^{j+k+1}(t_l - t_j)} + (t_{j+k+1} - t_j)\frac{(x-t_{j+k+1})_+^k}{\prod_{l=j}^{j+k}(t_l - t_{j+k+1})} \tag{84}$$

$$+ \sum_{i=j+1}^{j+k} \frac{(xt_{j+k+1} - t_j t_{j+k+1} - xt_i + t_j t_i + t_{j+k+1}t_j - xt_j - t_{j+k+1}t_i + xt_i)\,(x-t_i)_+^{k-1}}{\prod_{l=j,\ l\neq i}^{j+k+1}(t_l - t_i)}$$

$$= (t_{j+k+1} - t_j)\frac{(x-t_j)_+^k}{\prod_{l=j+1}^{j+k+1}(t_l - t_j)} + (t_{j+k+1} - t_j)\frac{(x-t_{j+k+1})_+^k}{\prod_{l=j}^{j+k}(t_l - t_{j+k+1})} \tag{85}$$

$$+ \sum_{i=j+1}^{j+k} \frac{(xt_{j+k+1} + t_j t_i - xt_j - t_{j+k+1}t_i)\,(x-t_i)_+^{k-1}}{\prod_{l=j,\ l\neq i}^{j+k+1}(t_l - t_i)}$$

$$= (t_{j+k+1} - t_j)\frac{(x-t_j)_+^k}{\prod_{l=j+1}^{j+k+1}(t_l - t_j)} + (t_{j+k+1} - t_j)\frac{(x-t_{j+k+1})_+^k}{\prod_{l=j}^{j+k}(t_l - t_{j+k+1})} \tag{86}$$

$$+ \sum_{i=j+1}^{j+k} \frac{(t_{j+k+1} - t_j)(x-t_i)(x-t_i)_+^{k-1}}{\prod_{l=j,\ l\neq i}^{j+k+1}(t_l - t_i)}$$

$$= (t_{j+k+1} - t_j)\frac{(x-t_j)_+^k}{\prod_{l=j+1}^{j+k+1}(t_l - t_j)} + (t_{j+k+1} - t_j)\frac{(x-t_{j+k+1})_+^k}{\prod_{l=j}^{j+k}(t_l - t_{j+k+1})} \tag{87}$$

$$+ (t_{j+k+1} - t_j)\sum_{i=j+1}^{j+k} \frac{(x-t_i)_+^k}{\prod_{l=j,\ l\neq i}^{j+k+1}(t_l - t_i)}$$

$$= (t_{j+k+1} - t_j)\sum_{i=j}^{j+k+1} \frac{(x-t_i)_+^k}{\prod_{l=j,\ l\neq i}^{j+k+1}(t_l - t_i)} \tag{88}$$

finally reaching the required form, and concluding the inductive step.

Clearly, the functions $x \mapsto (x-t_i)_+^k$ are $(k-1)$-times differentiable (with derivatives up to order $k-1$ at $x = t_i$ equal to 0). Then since we've expressed each B_j^k as a linear combination of C^{k-1}-smooth functions, it follows that B_j^k is C^{k-1}-smooth. This establishes property (5), and concludes the proof of the lemma. $\square$

In the figures that follow, we plot B-splines of increasing degree for a fixed sequence of knots. The properties discussed in the preceding theorem are immediately evident in these examples, for instance the properties that at any given input the family of B-splines sums to 1 (partition of unity property), and with each increasing degree of the B-spline one more panel of support is appended to the right. The B-splines plotted below in figures (19) through (22) were evaluated using the Cox-deBoor algorithm.

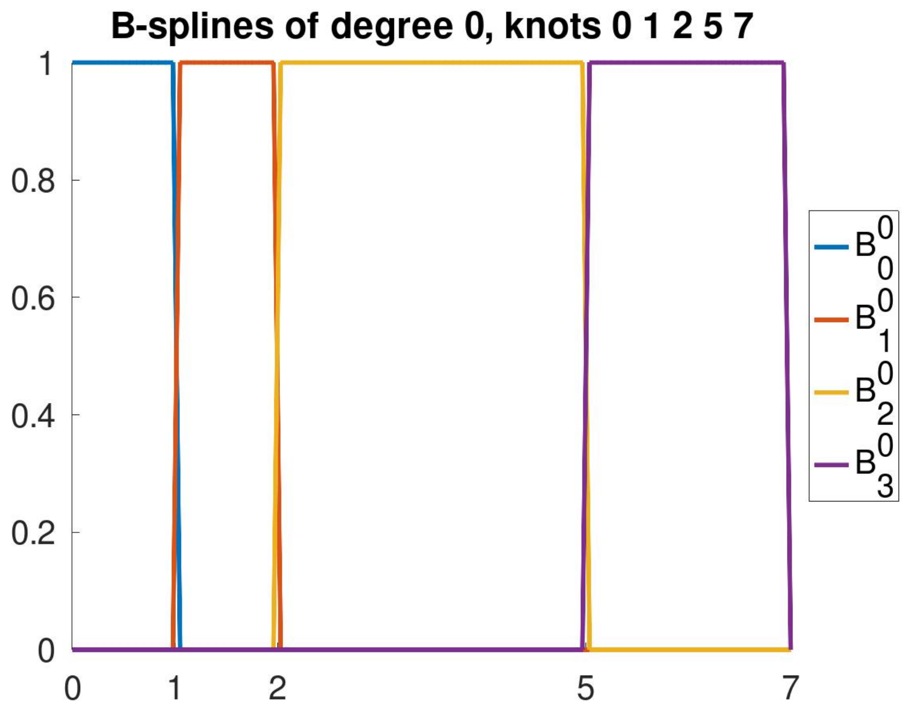


Figure 19: As we can see in this collection of degree 0 B-splines, they are just step functions that are 1 over a single panel and 0 otherwise.

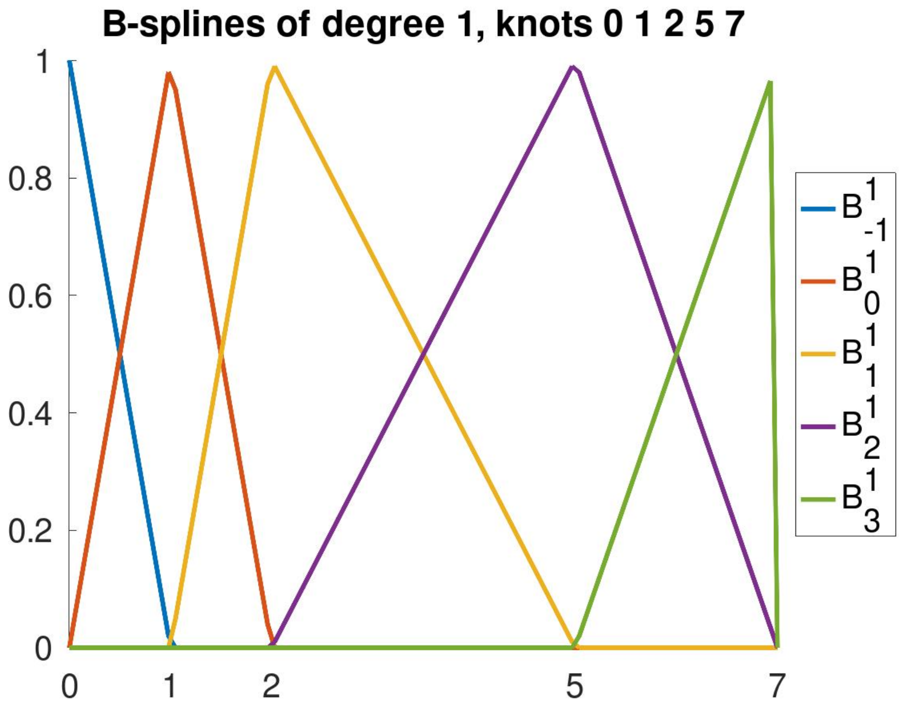


Figure 20: Increasing the degree of the B-splines by 1 causes the support of each spline to increase by one panel on the right, compared to the B-splines of degree 0. We can also see that there is one more B-spline added to the family over the lesser degree family.

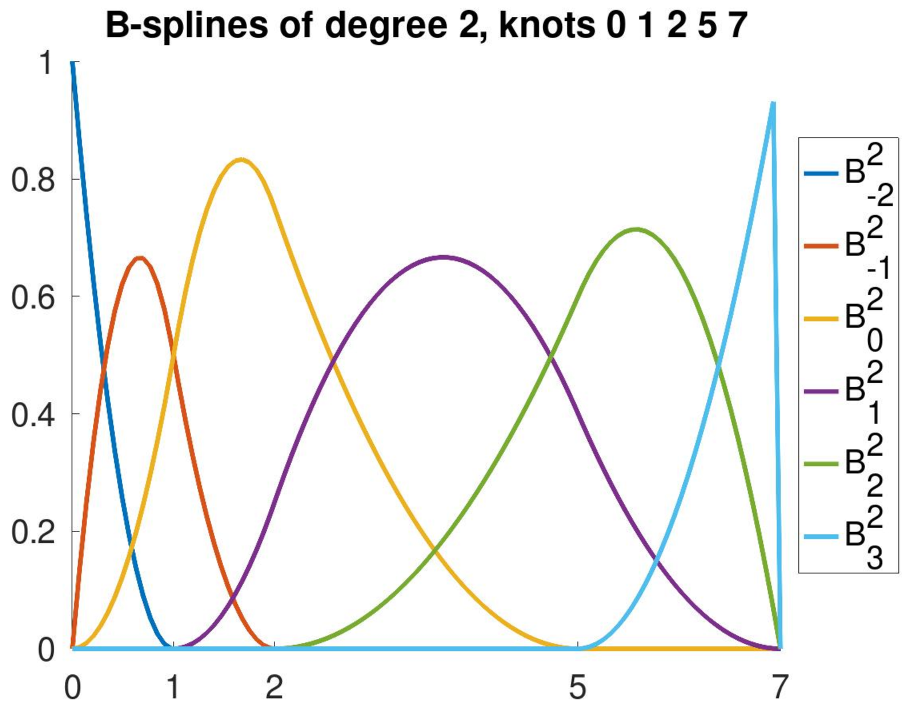


Figure 21: Further increasing the degree of B-splines by 1 causes the support of each spline to extend by one panel to the right over the linear splines of the last generation. Since these are quadratic B-splines, they have 1+2=3 panels of support each (including degenerate panels at the left- and right-most endpoints).

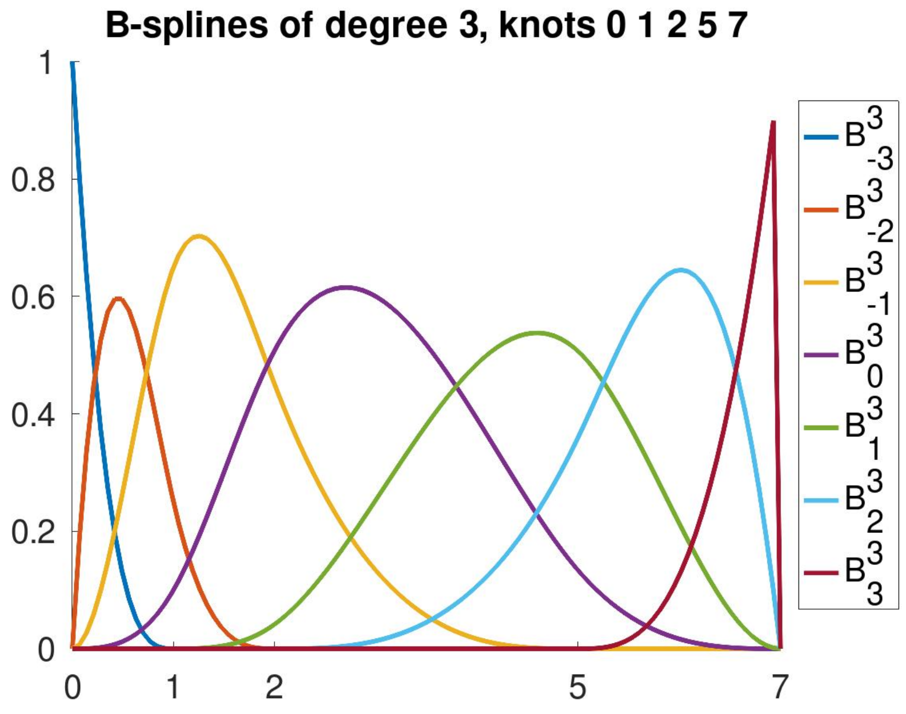


Figure 22: Cubic B-splines, with 1+3=4 panels of support.

With these B-spline properties, we are now able to prove the main convergence result for A-splines. This next theorem will provide a reasonably tight bound on $\mathrm{dist}(f, \mathcal{S}_D)$ in the L2 sense. For reference, similar bounds (albeit with different coefficients) already exist (e.g. [20], which uses Fourier transform techniques), and de Boor gives a skeleton argument in ([7]). The theorem given here, while yielding the same order of convergence as previous work, has the benefit of utilizing an elementary approach.

Theorem 6.2 (A-spline Convergence Theorem). *Let $t_0 < t_1 < \cdots < t_M$ be a knot sequence. Let $f\colon [t_0, t_M] \to \mathbb{R}$ be some function. Let $S\colon Map([t_0, t_M], \mathbb{R}) \to \mathcal{S}_D$ be the projection mapping that orthogonally projects functions onto the spline space $\mathcal{S}_D$ of degree D splines with the knots above, w.r.t. the continuous L2 inner product. Then the L2 error bound is:*

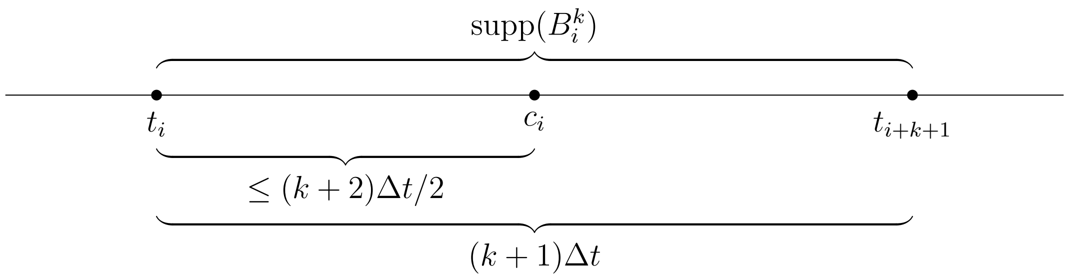
$$||f - S(f)||_2 \leq \frac{(D+1)!}{2^j \cdot (D-j)!} \cdot (\Delta t)^{j+1} \cdot \sqrt{t_M - t_0} \cdot ||f^{(j+1)}||_\infty, \qquad j = 0, 1, \ldots, D \tag{89}$$

with the slight improvement of an additional factor of 2 in the denominator when $j = D$.

Proof. Let $\{B_j^k\}_{j=-k}^{M-1}$ be the B-spline family of degree $k \in \{0, 1, \ldots, D\}$ with the knots above. Define the following smoothing spline operator, applied to some function $g\colon [t_0, t_M] \to \mathbb{R}$:

$$A_k g := \sum_{i=-k}^{M-1} g(c_i) B_i^k, \qquad c_i = \begin{cases} (t_{i+k/2} + t_{i+k/2+1})/2 & k \text{ even} \\ t_{i+(k+1)/2} & k \text{ odd} \end{cases} \tag{90}$$

Note that $\mathrm{supp}(B_i^k) = [t_i, t_{i+k+1}]$ for each $i \in \{-k, \ldots, M-1\}$. So the only B-splines which are nonzero on $[t_j, t_{j+1}]$ are $B_{j-k}^k, \ldots, B_j^k$, and $\forall x \in [t_j, t_{j+1}]$, $1 = \sum\limits_{i=-k}^{M-1} B_i^k = \sum\limits_{i=j-k}^{j} B_i^k$. So, the c_i points are roughly the midpoints of the support of B_i^k.



Then for all $x \in [t_j, t_{j+1})$ we have

$$\begin{aligned}
|A_k g(x) - g(x)| &= |\sum_{i=-k}^{M-1} g(c_i)B_i^k(x) - \sum_{i=-k}^{M-1} g(x)B_i^k(x)| && (91)\\
&= |\sum_{i=j-k}^{j} g(c_i)B_i^k(x) - \sum_{i=j-k}^{j} g(x)B_i^k(x)| && (92)\\
&= |\sum_{i=j-k}^{j} (g(c_i) - g(x))B_i^k(x)| && (93)\\
&\leq \sum_{i=j-k}^{j} |g(c_i) - g(x)|B_i^k(x) && (94)\\
&\leq \left(\sup_{i=j-k,\dots,j} |g(c_i) - g(x)|\right) \sum_{i=j-k}^{j} B_i^k(x) && (95)\\
&= \sup_{i=j-k,\dots,j} |g(c_i) - g(x)| && (96)\\
&\leq \omega\left(g; \left(\frac{k+1}{2}\right)\Delta t\right) && (97)\\
&\leq \mathrm{ceil}((k+1)/2)\omega(g;\Delta t) \qquad \text{(by triangle inequality)} && (98)\\
&\leq \frac{(k+2)}{2}\omega(g;\Delta t) && (99)
\end{aligned}$$

where

$$\omega(g;\Delta t) := \sup_{|x-y|\leq \Delta t} |g(x) - g(y)| \qquad (100)$$

is the modulus of continuity.

Since $A_k g \in \mathcal{S}_k$, we can now use inequality (97) to say that

$$\operatorname{dist}(g, \mathcal{S}_k) \leq ||A_k g - g||_\infty \leq \omega\left(g; \left(\frac{k+1}{2}\right)\Delta t\right) \leq \left(\frac{k+2}{2}\right)\omega(g; \Delta t) \tag{101}$$

where the distance function here is with respect to the ∞-norm. This bound (101) holds for any nonnegative integer k. Also, this bound holds for arbitrary functions g, even discontinuous ones. But now let us assume g is sufficiently differentiable for the following calculations, and attempt to express $\operatorname{dist}(g, \mathcal{S}_D)$ in terms of higher powers of Δt and higher derivatives of g.

The following calculation uses the fact that $\mathcal{S}_j' = \mathcal{S}_{j-1}$, which can be demonstrated by simply differentiating/integrating polynomial pieces and keeping track of continuity conditions at the knots.

For any $s \in \mathcal{S}_D$ we have

$$\begin{aligned}
\operatorname{dist}(g, \mathcal{S}_D) &= \operatorname{dist}(g - s, \mathcal{S}_D) && \text{(since } \mathcal{S}_D \text{ is a vector space)} && (102)\\
&\leq \omega\left(g - s; \left(\frac{D+1}{2}\right)\Delta t\right) && \text{(by (101))} && (103)\\
&\leq \left(\frac{D+1}{2}\right)\Delta t ||g' - s'||_\infty && \text{(by mean value theorem)} && (104)
\end{aligned}$$

Continuing this line, for any $\epsilon > 0$ we can find a spline $s^0 \in \mathcal{S}_{D-1}$ such that

$$||g' - s^0||_\infty < \operatorname{dist}(g', \mathcal{S}_{D-1}) + \epsilon \tag{105}$$

Now letting s be any antiderivative of s^0, it must be that $s \in \mathcal{S}_D$, and so by the above calculation,

$$\operatorname{dist}(g, \mathcal{S}_D) \leq \left(\frac{D+1}{2}\right)\Delta t ||g' - s'||_\infty \tag{106}$$

$$= \left(\frac{D+1}{2}\right) \Delta t ||g' - s^0||_\infty \tag{107}$$

$$\leq \left(\frac{D+1}{2}\right) \Delta t \cdot (\text{dist}(g', \mathcal{S}_{D-1}) + \epsilon) \tag{108}$$

Since this holds for all $\epsilon > 0$, it follows that

$$\text{dist}(g, \mathcal{S}_D) \leq \left(\frac{D+1}{2}\right) \Delta t \cdot \text{dist}(g', \mathcal{S}_{D-1}) \tag{109}$$

Now by repeating this argument j times, we obtain

$$\text{dist}(g, \mathcal{S}_D) \leq \left(\frac{D+1}{2}\right) \left(\frac{D}{2}\right) \cdots \left(\frac{D+2-j}{2}\right) (\Delta t)^j \cdot \text{dist}(g^{(j)}, \mathcal{S}_{D-j}), \qquad j = 0, 1, \ldots, D \tag{110}$$

At this point we may reuse inequality (101) (with $g^{(j)}$ and $\mathcal{S}_{D-j}$ playing the roles of g and $\mathcal{S}_k$, respectively) to assert

$$\text{dist}(g^{(j)}, \mathcal{S}_{D-j}) \leq \omega(g^{(j)}; \left(\frac{D-j+1}{2}\right) \Delta t) \tag{111}$$

and now combining this and the mean value theorem with the result immediately above, we see

$$\text{dist}(g, \mathcal{S}_D) \leq \frac{(D+1)!}{2^j \cdot (D+1-j)!} (\Delta t)^j \cdot \text{dist}(g^{(j)}, \mathcal{S}_{D-j}) \tag{112}$$

$$\leq \frac{(D+1)!}{2^j \cdot (D+1-j)!} (\Delta t)^j \cdot \omega(g^{(j)}; \left(\frac{D-j+1}{2}\right) \Delta t) \tag{113}$$

$$\leq \frac{(D+1)!}{2^{j+1} \cdot (D-j)!} (\Delta t)^{j+1} \cdot ||g^{(j+1)}||_\infty, \qquad j = 0, 1, \ldots, D \tag{114}$$

In fact, when $j = D$, we may do slightly better (by a factor of 2), by using the top inequality for $j = D - 1$ and using a broken line (in $\mathcal{S}_1$) approximation of $g^{(D-1)}$, which has uniform error bound $(\Delta t)^2||g^{(D+1)}||_\infty/4$:

$$\begin{aligned} \operatorname{dist}(g,\mathcal{S}_D) &\leq \frac{(D+1)!}{2^{D-1}\cdot(2)!}(\Delta t)^{(D-1)}\cdot \operatorname{dist}(g^{(D-1)},\mathcal{S}_1) && (115)\\ &\leq \frac{(D+1)!}{2^D}(\Delta t)^{(D-1)}\cdot\frac{(\Delta t)^2||g^{(D+1)}||_\infty}{4} && (116)\\ &= \frac{(D+1)!}{2^{D+2}}(\Delta t)^{D+1}\cdot||g^{(D+1)}||_\infty && (117)\end{aligned}$$

Then using the fact that the (linear) spline operator S is an orthogonal projection (w.r.t. the L^2 inner product) of an input function onto the vector space $\mathcal{S}_D$, the Pythagorean Theorem asserts that for any input function ϕ, we have

$$||\phi||_2^2 = ||S(\phi)||_2^2 + ||\phi - S(\phi)||_2^2 \tag{118}$$

which implies

$$||S(\phi)||_2 \leq ||\phi||_2 \tag{119}$$

and so we may bound the operator norm of the spline operator S by

$$||S||_{2,\text{op}} =: ||S||_2 \leq 1 \tag{120}$$

Now if we let $\epsilon > 0$ and select $s \in \mathcal{S}_D$ such that $||f-s||_\infty < \operatorname{dist}(f,\mathcal{S}_D)+\epsilon$, then $S(s) = s$ (as S reproduces splines of degree D and with the knots declared above) and we can use inequality (114)

$$\operatorname{dist}(f,\mathcal{S}_D) \leq \frac{(D+1)!}{2^{j+1}\cdot(D-j)!}(\Delta t)^{j+1}\cdot||f^{(j+1)}||_\infty, \qquad j = 0,1,\ldots,D \tag{121}$$

to say that

$$||f-S(f)||_2 \leq ||f-s||_2 + ||s-S(f)||_2 \tag{122}$$

$$= ||f - s||_2 + ||S(s) - S(f)||_2 \tag{123}$$

$$= ||f - s||_2 + ||S(s - f)||_2 \tag{124}$$

$$\leq ||f - s||_2 + ||S||_2 ||s - f||_2 \tag{125}$$

$$= (1 + ||S||_2) \cdot ||f - s||_2 \tag{126}$$

$$\leq 2||f - s||_2 \tag{127}$$

$$= 2\sqrt{\int_{t_0}^{t_M} (f(x) - s(x))^2 dx} \tag{128}$$

$$\leq 2\sqrt{\int_{t_0}^{t_M} ||f - s||_\infty^2 dx} \tag{129}$$

$$= 2||f - s||_\infty \sqrt{\int_{t_0}^{t_M} dx} \tag{130}$$

$$= 2\sqrt{t_M - t_0}||f - s||_\infty \tag{131}$$

$$\leq 2\sqrt{t_M - t_0}(\text{dist}(f, \mathcal{S}_D) + \epsilon) \tag{132}$$

Since this holds for all $\epsilon > 0$, it follows that

$$||f - S(f)||_2 \leq 2\sqrt{t_M - t_0}\text{dist}(f, \mathcal{S}_D) \leq \frac{(D+1)!}{2^j \cdot (D-j)!} \cdot (\Delta t)^{j+1} \cdot \sqrt{t_M - t_0} \cdot ||f^{(j+1)}||_\infty, \quad j = 0, 1, \ldots, D \tag{133}$$

and again in the special case $j = D$, we get the slightly improved bound

$$||f - S(f)||_2 \leq \frac{(D+1)!}{2^{D+1}} \cdot (\Delta t)^{D+1} \cdot \sqrt{t_M - t_0} \cdot ||f^{(D+1)}||_\infty \tag{134}$$

□

In figures (23) through (30), we demonstrate the efficacy of approximating a function with a sharp peak by projection onto an orthogonal basis of cubic splines with equispaced knots. In each subsequent figure, the panel count is doubled. In theory, the error should diminish by a factor of $2^4 = 16$ with each subsequent approximation, but several factors impede us from observing this order of convergence: (1) the underlying function has a relatively fast-growing derivatives, so the 4th order convergence is only expected to be realized for small h values, and (2) the quadrature rule induces an error which is accurate of a lesser order than the error associated with the projection. These two factors detract from the theoretical order of convergence for large and small h values, respectively. Nevertheless, we observe an improvement in the approximation with each refinement of the spline grid, both in the error metrics and by observation. The closeness of the first derivative of the function to that of the approximation is also recorded.

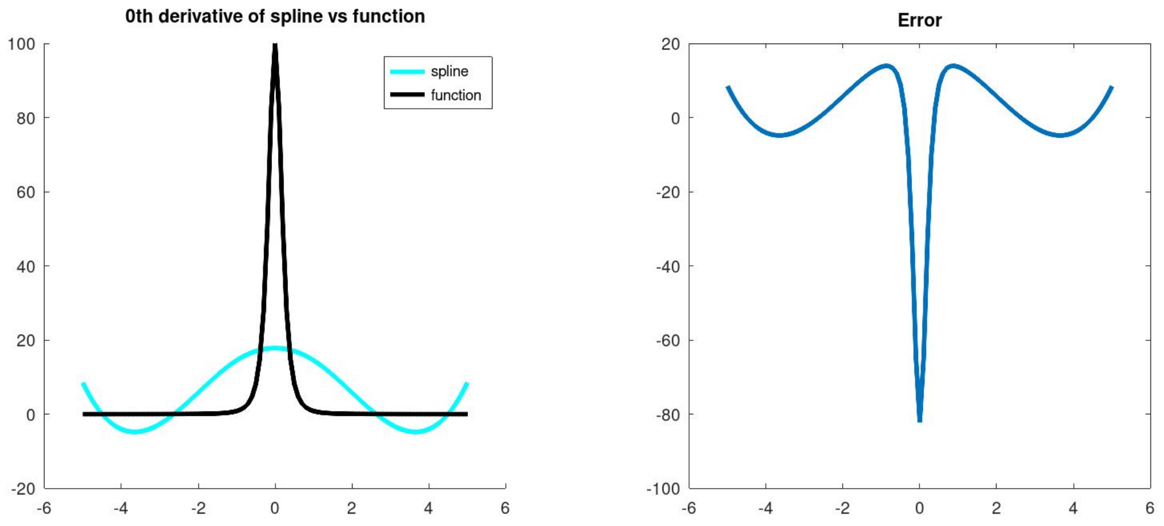


Figure 23: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 0th derivative of spline vs function, 5 panels. Average error (computed as L2 error divided by (b-a)) = 15; max error = 82; relative error = .82

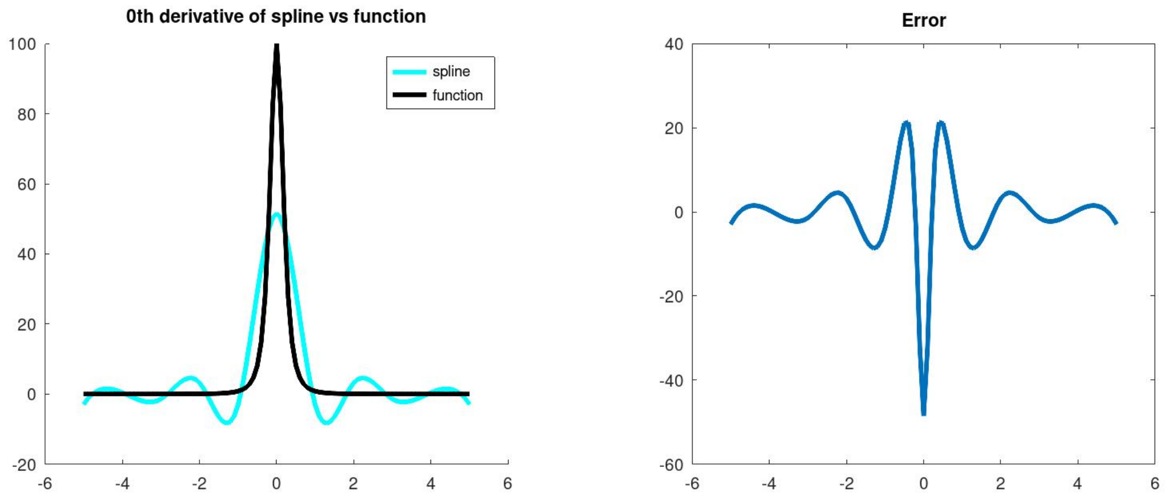


Figure 24: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 0th derivative of spline vs function, 10 panels. Average error (computed as L2 error divided by (b-a)) = 9.3; max error = 48; relative error = .48

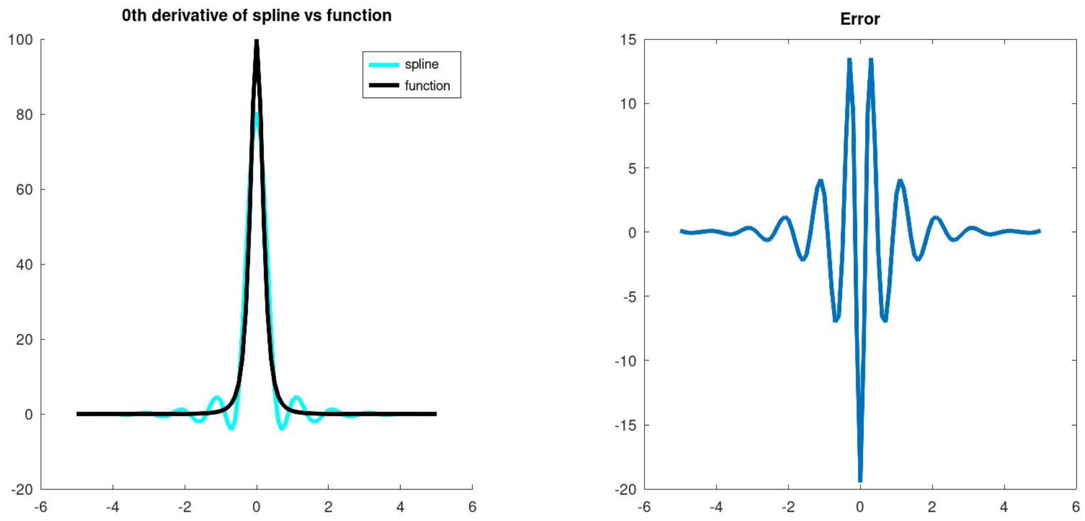


Figure 25: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 0th derivative of spline vs function, 20 panels. Average error (computed as L2 error divided by (b-a)) = 3.8; max error = 19; relative error = .19

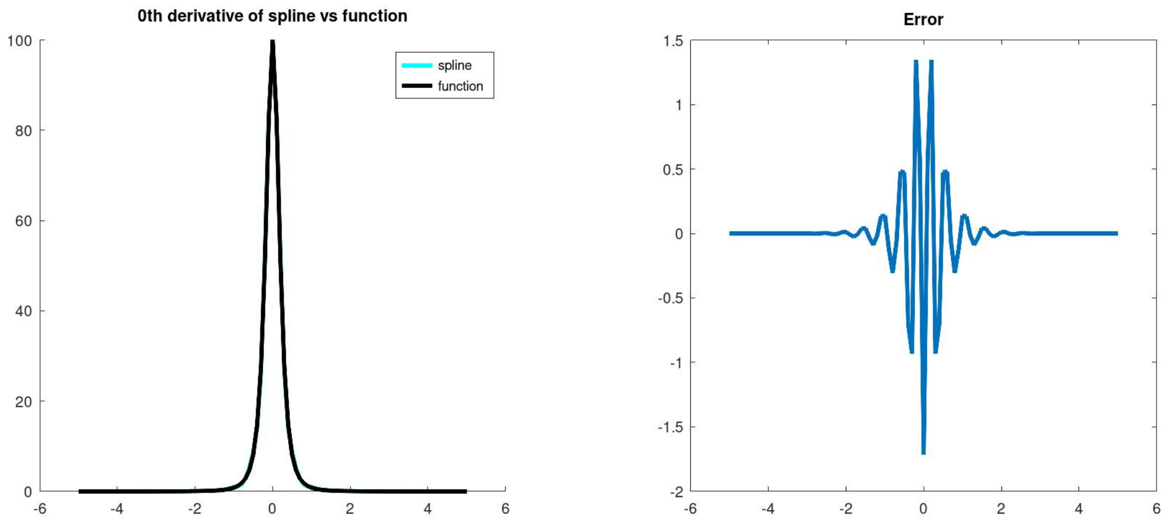


Figure 26: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 0th derivative of spline vs function, 40 panels. Average error (computed as L2 error divided by (b-a)) = .33; max error = 1.7; relative error = .017

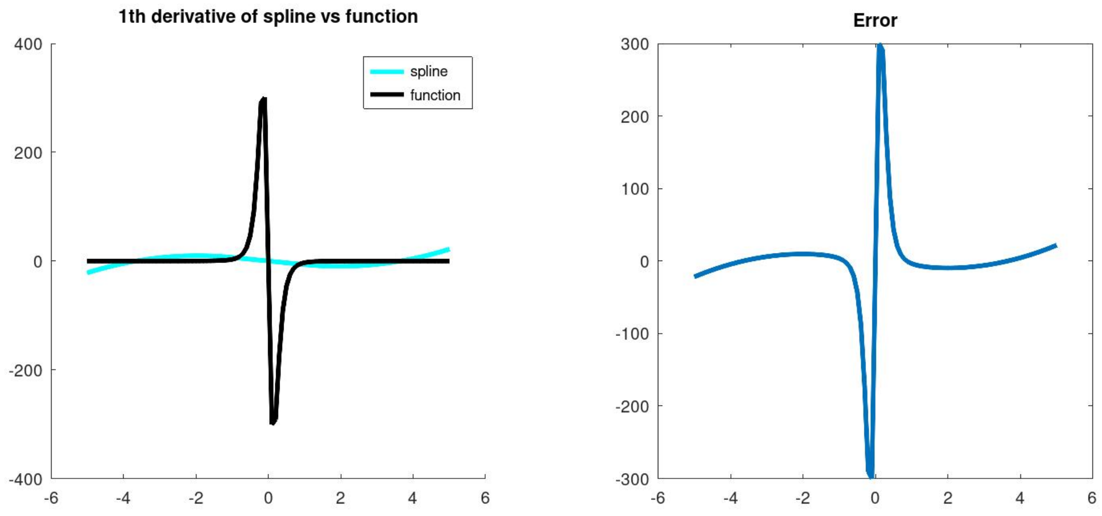


Figure 27: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 1st derivative of spline vs function, 5 panels. Average error (computed as L2 error divided by (b-a)) = 66; max error = 300; relative error = 1.0

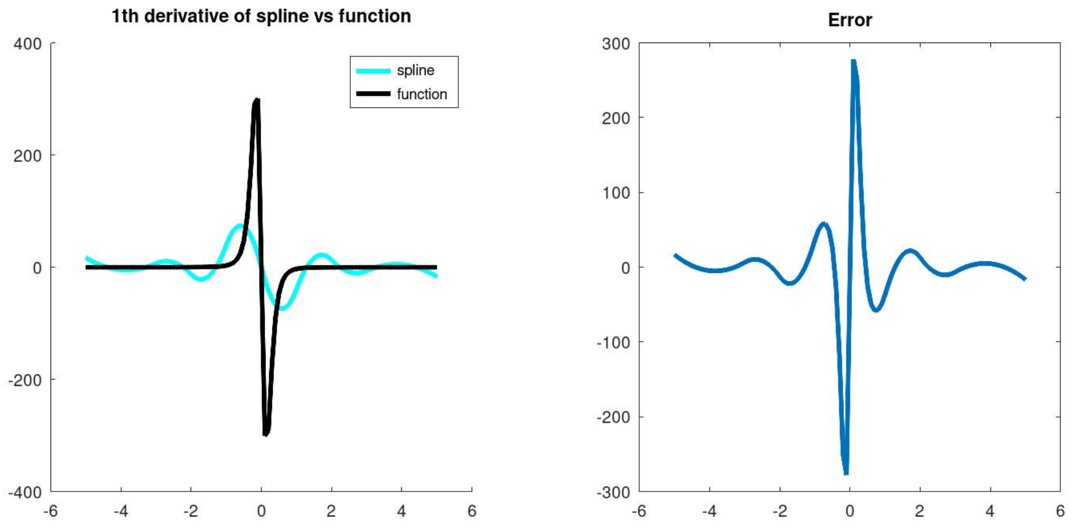


Figure 28: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 1st derivative of spline vs function, 10 panels. Average error (computed as L2 error divided by (b-a)) = 59; max error = 280; relative error = .92

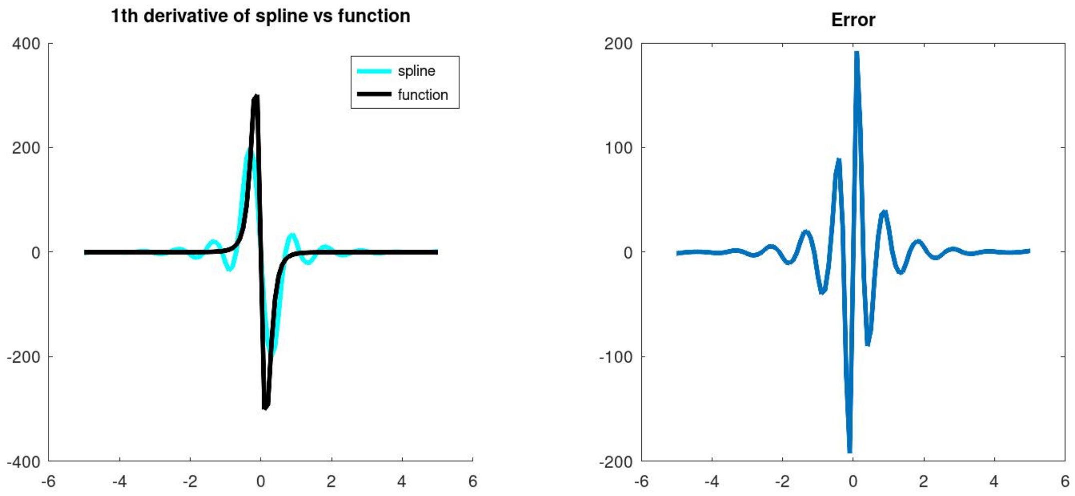


Figure 29: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 1st derivative of spline vs function, 20 panels. Average error (computed as L2 error divided by (b-a)) = 37; max error = 190; relative error = .64

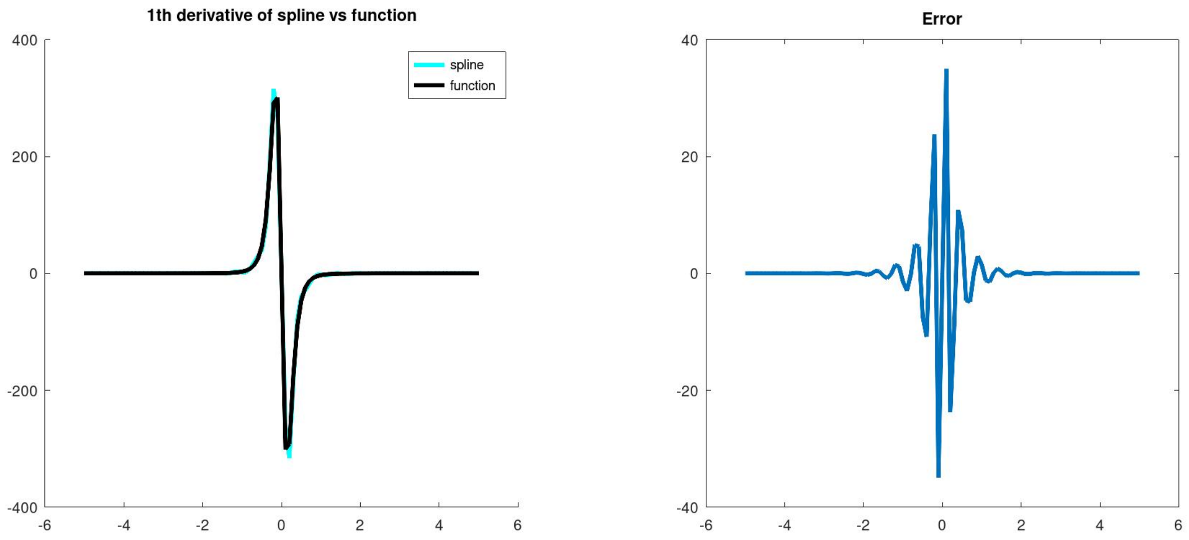


Figure 30: Cubic A-spline approximation of $f(t) = \dfrac{1}{(t^2+.1)^2}$, 1st derivative of spline vs function, 40 panels. Average error (computed as L2 error divided by (b-a)) = 6.5; max error = 35; relative error = .12

In the next series of plots, figures (31) through (40), we demonstrate the efficacy of approximating a sine function by projection onto an orthogonal basis of cubic splines with equispaced knots. In each subsequent figure, the panel count is doubled. In theory, the error should diminish by a factor of $2^4 = 16$ with each subsequent approximation. Though the derivatives of the underlying function do not grow as they did in the previous example, the theoretical order of convergence is still impeded by imprecision brought on by quadrature error. Despite this, we observe a decay of error that is within appreciable tolerance of the theoretical error decay. The closeness of the first derivative of the function to that of the approximation is also recorded. We expect the derivative to converge of one order less than that of the underlying function; this is observed for the first few grid refinements, before the error from quadrature outweighs the error from projection.

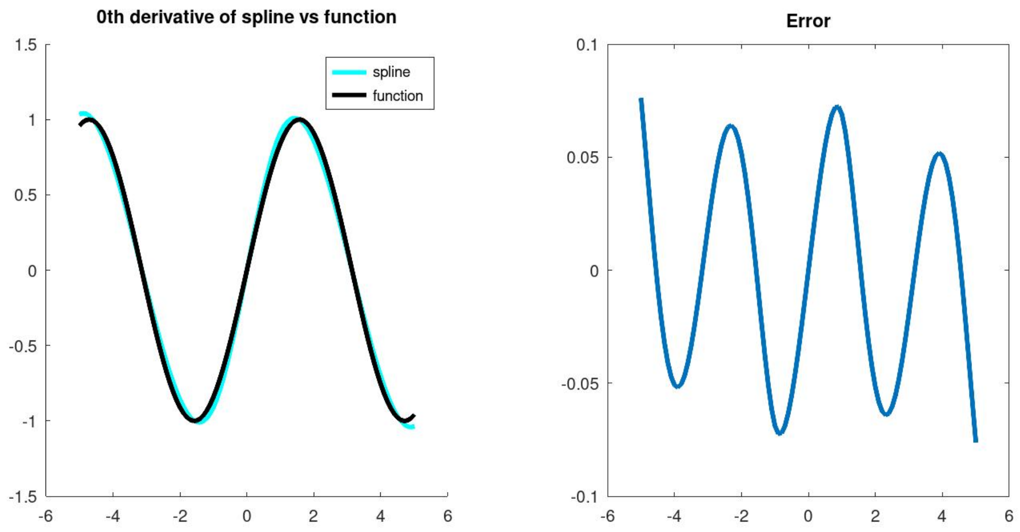


Figure 31: Cubic A-spline approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 5 panels. Average error (computed as L2 error divided by (b-a)) = 4.5e-2; max error = 7.6e-2; relative error = 7.6e-2

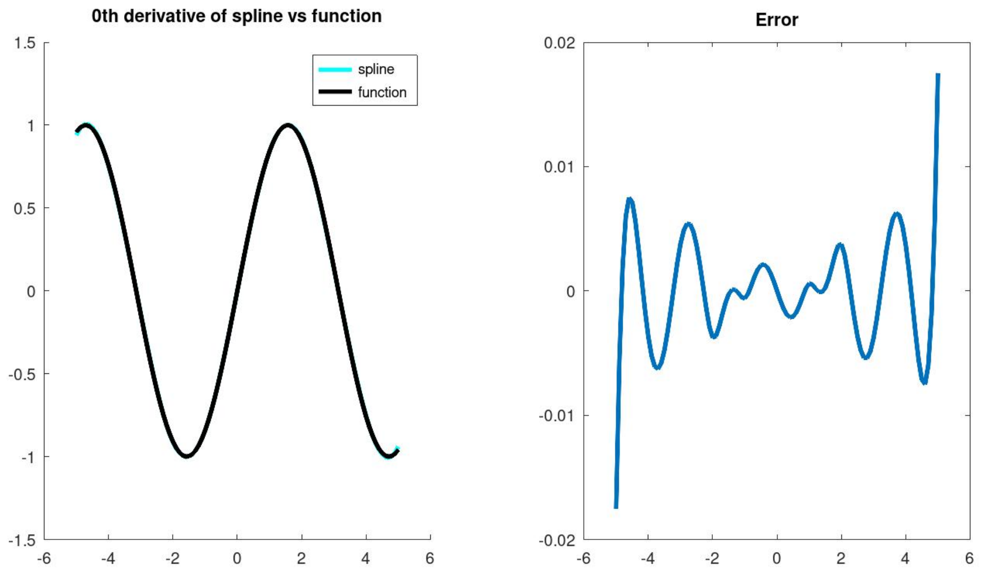


Figure 32: Cubic A-spline approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 10 panels. Average error (computed as L2 error divided by (b-a)) = 3.8e-3; max error = 1.8e-2; relative error = 1.8e-2

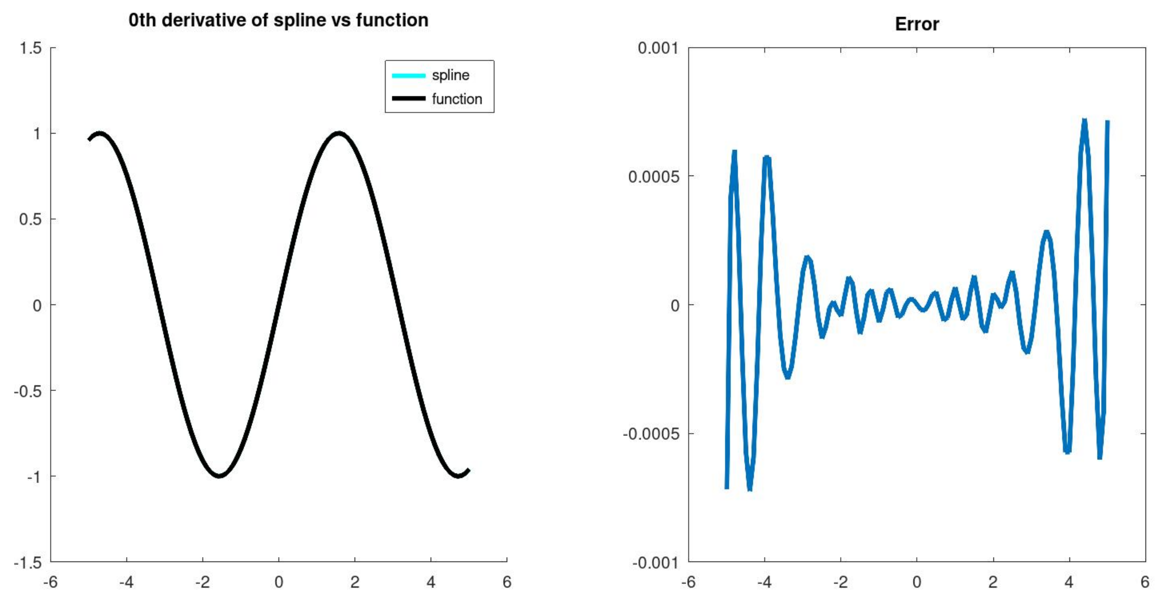


Figure 33: Cubic A-spline approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 20 panels. Average error (computed as L2 error divided by (b-a)) = 2.6e-4; max error = 7.2e-4; relative error = 7.2e-4

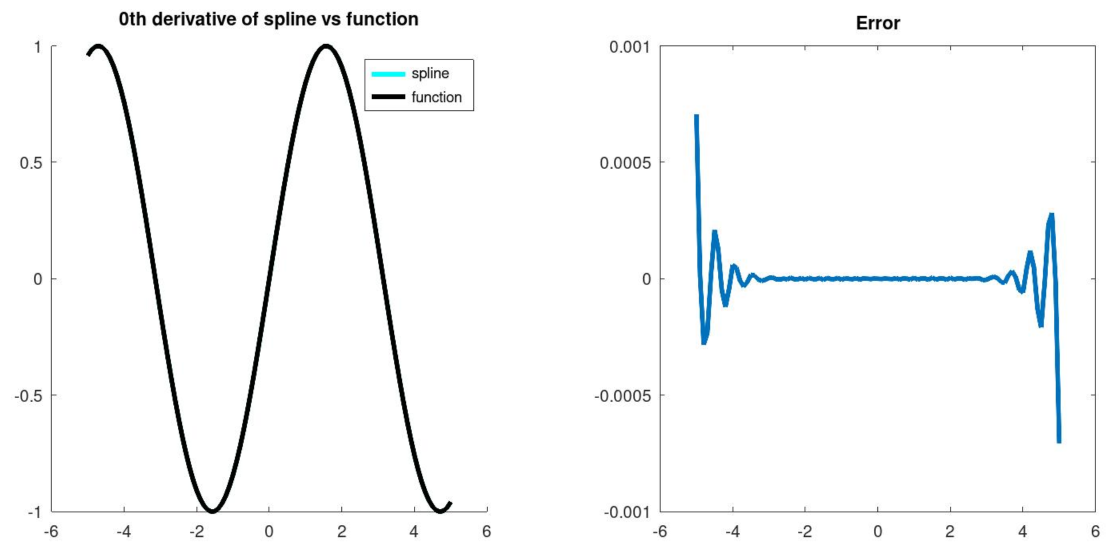


Figure 34: Cubic A-spline approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 40 panels. Average error (computed as L2 error divided by (b-a)) = 8.3e-5; max error = 7.1e-4; relative error = 7.1e-4

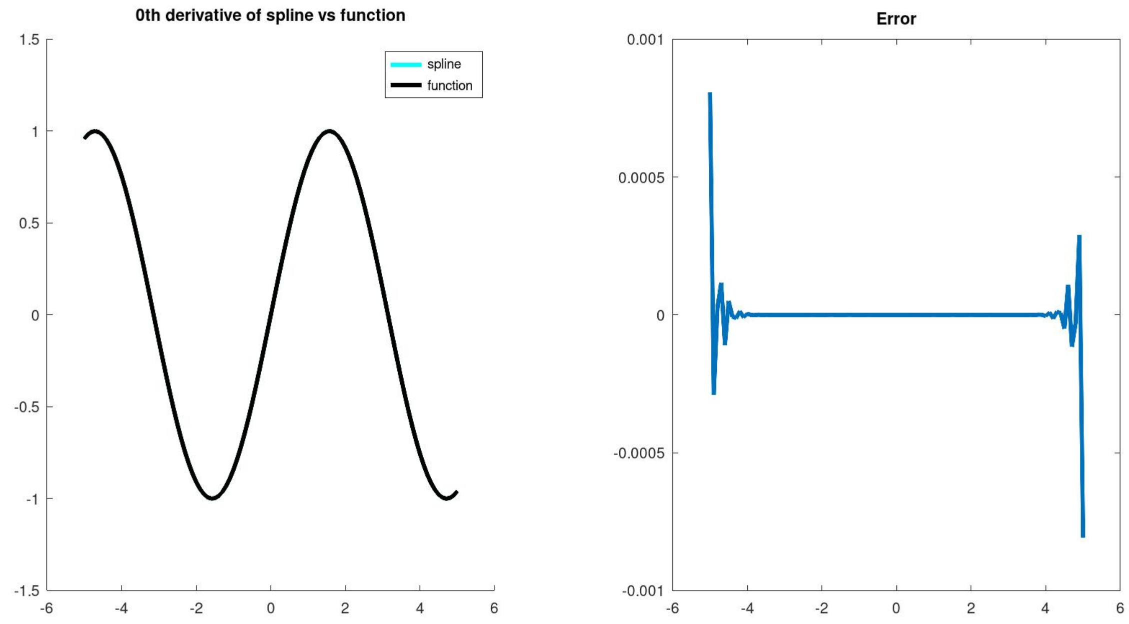


Figure 35: Cubic A-spline approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 80 panels. Average error (computed as L2 error divided by (b-a)) = 7.7e-5; max error = 8.1e-4; relative error = 8.1e-4. The quadrature rule is a bottleneck on the error, since Trapezoidal Rule is used, which is only 2nd order accurate, whereas cubic A-splines are 4th order accurate.

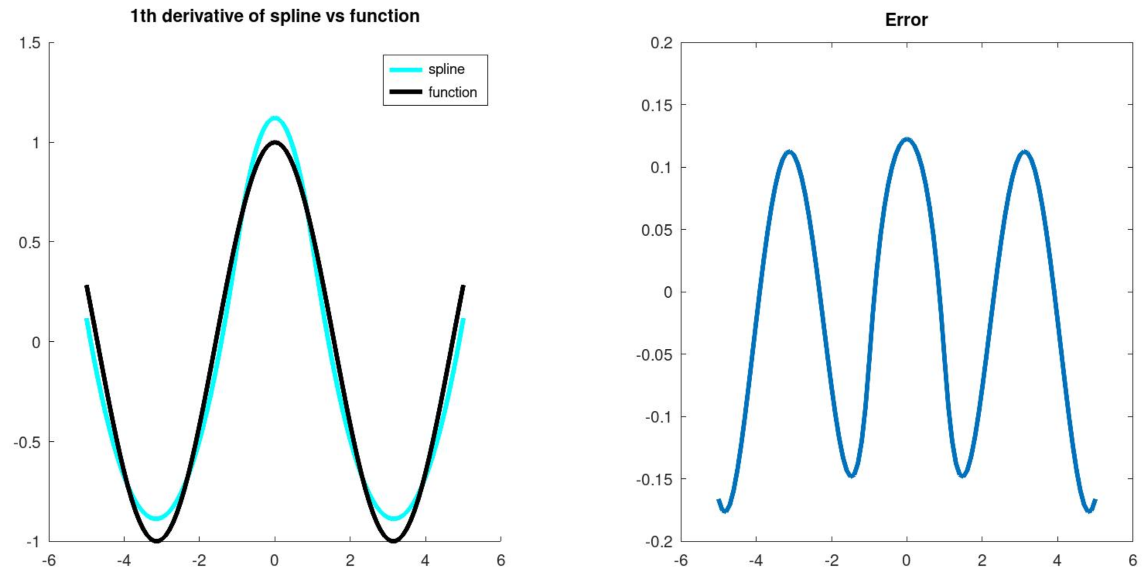


Figure 36: Cubic A-spline approximation of $f(t) = \sin(t)$, 1st derivative of spline vs function, 5 panels. Average error (computed as L2 error divided by (b-a)) = 1.0e-1; max error = 1.8e-1; relative error = 1.8e-1

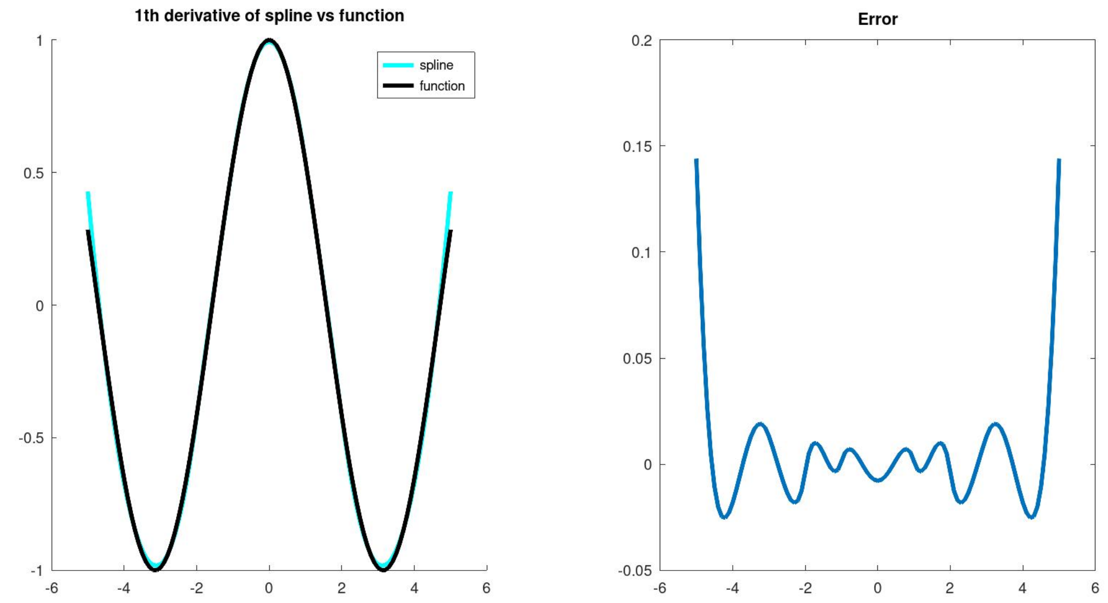


Figure 37: Cubic A-spline approximation of $f(t) = \sin(t)$, 1st derivative of spline vs function, 10 panels. Average error (computed as L2 error divided by (b-a)) = 2.3e-2; max error = 1.4e-1; relative error = 1.4e-1

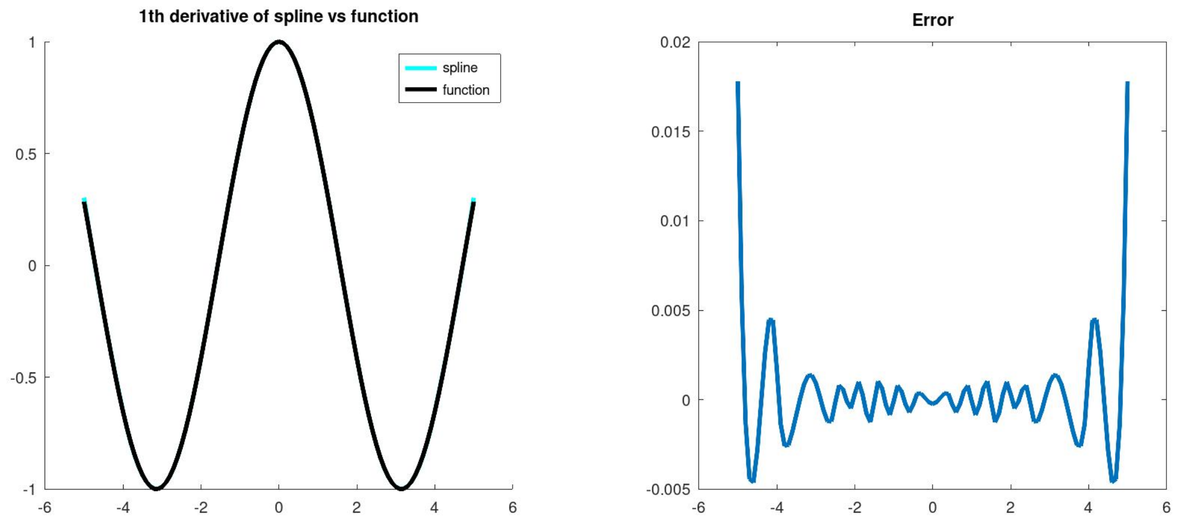


Figure 38: Cubic A-spline approximation of $f(t) = \sin(t)$, 1st derivative of spline vs function, 20 panels. Average error (computed as L2 error divided by (b-a)) = 2.3e-3; max error = 1.8e-2; relative error = 1.8e-2

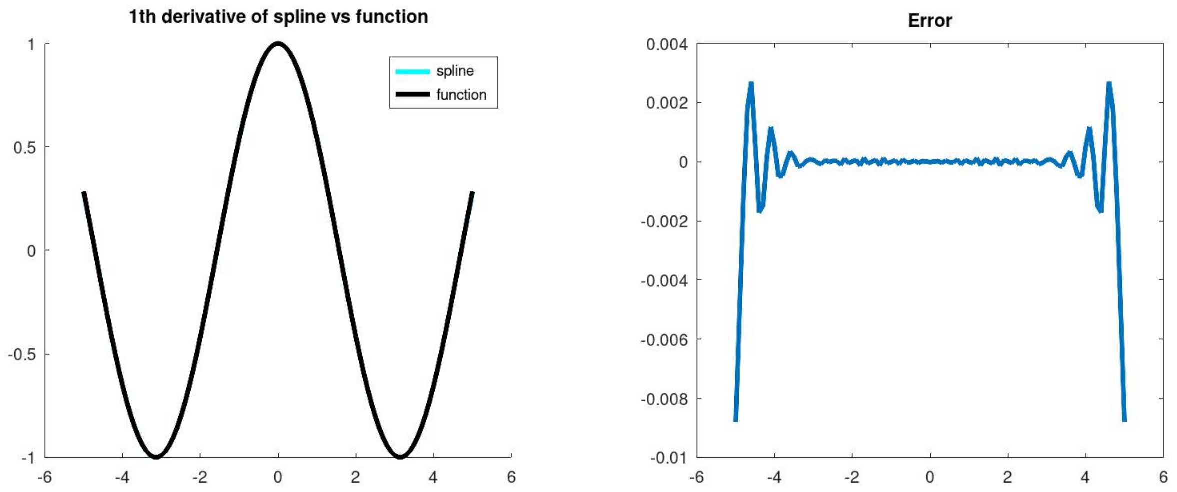


Figure 39: Cubic A-spline approximation of $f(t) = \sin(t)$, 1st derivative of spline vs function, 40 panels. Average error (computed as L2 error divided by (b-a)) = 1.2e-3; max error = 8.8e-3; relative error = 8.8e-3

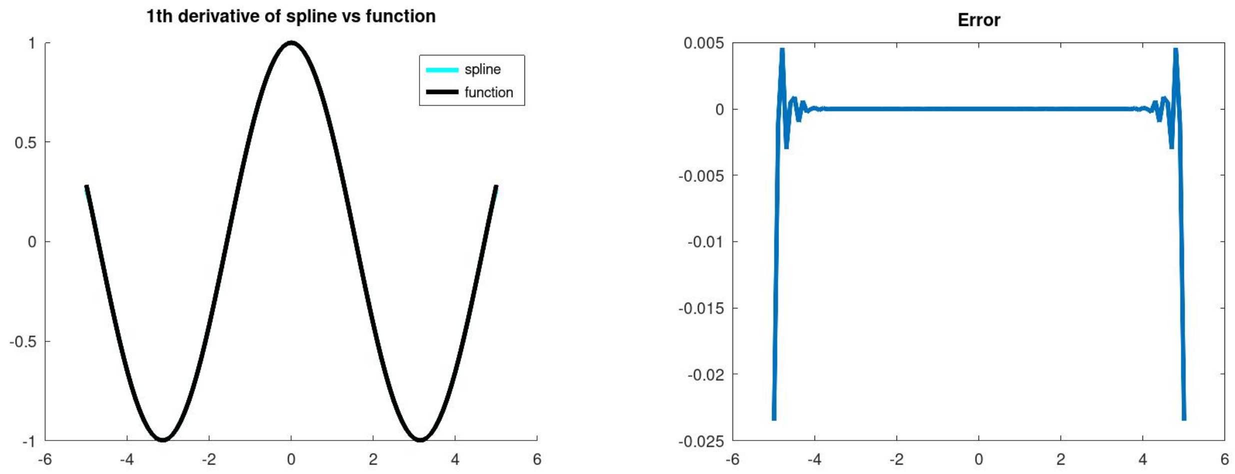


Figure 40: Cubic A-spline approximation of $f(t) = \sin(t)$, 1st derivative of spline vs function, 80 panels. Average error (computed as L2 error divided by (b-a)) = 1.9e-3; max error = 2.3e-2; relative error = 2.3e-2. The quadrature rule is a bottleneck on the error, since Trapezoidal Rule is used, which is only 2nd order accurate, whereas cubic A-splines are 4th order accurate.

Note regarding a limitation of this result: this only applies to the orthogonal spline operator S w.r.t. the continuous L2 inner product, that reproduces $\mathcal{S}_D$. So, It does not directly apply to, say, natural or clamped splines, which have various pre-determined derivatives at the endpoints. It does, however, apply to orthogonal spline operators which produce splines with not-a-knot conditions, because we may simply eliminate knots from the formula for the error bound and treat the spline operator as a projection onto the resulting space of splines. This can be done by increasing Δt in the error bound.

Tightness of the error bound

This bound is likely not optimal in the coefficient. Indeed, there are already-existing tight bounds for cubic splines, and the coefficients in those error formulas are lower than the coefficients here ([12]). But there is good reason to believe this error bound is optimal in the power of Δt. We give several example scenarios here to demonstrate this.

If we try to approximate a function f using piecewise-constant splines, it is clear that over each interval t_i, t_{i+1}, the spline approximant will have L^∞ error over that panel greater than or equal to $(\sup_{t\in[t_i,t_{i+1}]}(f) - \inf_{t\in[t_i,t_{i+1}]}(f))/2$, with the optimal approximant (in the L^∞ sense) being an average of those two values over the panel $[t_j, t_{j+1}]$. Then since

$$\frac{\sup_{t\in[t_i,t_{i+1}]}(f) - \inf_{t\in[t_i,t_{i+1}]}(f)}{2} \leq (t_{i+1} - t_i)||f'||_\infty \leq (\Delta t)||f'||_\infty \tag{135}$$

by the mean value theorem, we can see that the power of Δt is correct in the case $D = 0$. More specifically, we can approximate the function $f(t) = t$ using a piecewise-constant spline to get this.

Another indication of the tightness of this error bound in the derivative of f present in it is discovered when we try to approximate a certain function $f(t)$ which is orthogonal to the basis of A-splines (in the L^2 sense) over $[0,1]$ using a single panel. So the knots are just $t_0 = 0$ and $t_1 = 1$ in this example, and splines of degree D w.r.t. these knots are simply polynomials of degree D. Since orthogonal projection is independent of the particular basis chosen for the procedure, we may express f as an infinite sum of Legendre polynomials $\{L_i\}_{i=0}^{\infty}$ (which are orthonormal w.r.t. the L^2 inner product), and we may also obtain our orthogonal projection onto $\mathcal{S}_D = \mathbb{P}_D$ via

$$S(f) = \langle f, L_0 \rangle L_0 + \cdots + \langle f, L_D \rangle L_D \tag{136}$$

Now if $f(t) = \alpha t^{D+1}$ for some $\alpha > 0$ then f does not lie inside the spline space $\mathcal{S}_D$, and so the above A-spline approximation will necessarily not be exact. By linearity of the inner products above, we see that the error incurred in this approximation is proportional to α, and is due to $f^{(D+1)}$ not being 0. $f^{(D+1)}$ is also proportional to α. This reaffirms the existence of the derivative of f present in the error bound.

We can also compare the bound derived here to that of Lagrange interpolation with nodes $x_0, \ldots, x_D$:

$$f(t) - L_D(t) = (t - x_0) \cdots (t - x_D) \frac{f^{(D+1)}(\xi)}{(D+1)!} \tag{137}$$

which, even if the nodes are chosen judiciously (e.g. Chebyshev points), is still an $\mathcal{O}(h^{D+1})$ approximation.

Taylor polynomial approximation of course has the same order error, over a single panel:

$$f(t) - P_D(t) = (t - x_0)^{D+1} \frac{f^{(D+1)}(\xi)}{(D+1)!} \tag{138}$$

It is a general trend we observe here that polynomials of degree D cannot approximate functions with greater accuracy than $\mathcal{O}(h^{D+1})$.

Closeness of Derivatives

It would be ideal to extend the error bounds already obtained for A-splines, to error bounds which discuss the closeness of derivatives of A-splines to the corresponding derivatives of the functions they approximate. In particular, the desired theorem is:

Conjecture. (Closeness of derivatives for A-splines).
Let S_D be the degree D A-spline approximation to f with knots $t_0 < \cdots < t_M$. Let $h = \max(t_j - t_{j-1})$ be the maximum panel width. Then

$$||f^{(k)} - S_D^{(k)}||_\infty = ||f^{(j)}||_\infty \mathcal{O}(h^{j-k}) \tag{139}$$

for $k = 0, 1, \dots, D$ and $j = k+1, \dots, D+1$.

Such bounds already exist for popular spline routines such as cubic spline interpolation, and there's no reason to suspect they don't apply to A-splines. As was shown above in figures (36) through (40), there is numerical evidence to support the above conjecture.

The proof of this conjecture will require a different set of tools to the ones used for the error bound above, because in our proof, we just find one particular spline in $\mathcal{S}_D$ to satisfy each error bound (depending on how many derivatives f has) in order to declare that the best spline amongst every spline in $\mathcal{S}_D$ must also satisfy the bound. But, this doesn't tell us that the derivatives of the particular best spline will also be close to the corresponding derivatives of f. Even if we find one particular spline in $\mathcal{S}_D$ satisfying this property, it is possible for the optimal spline in the L^2 sense might have some derivatives that are further away from those of f than the non-optimal spline in $\mathcal{S}_D$ with close derivatives.

One idea for such a proof would be the introduction of a new norm on $\mathcal{S}_D$ more akin to a Sobolev norm but with weights on derivatives; each subsequent derivative term getting multiplied by an additional factor of h. Finding the optimal spline in such a space, and showing that this new norm on it is no more than $\mathcal{O}(h^{D+1})$ would at least give this result for $j = D + 1$.

Quadrature error

In the standard (and exponential) A-spline procedures, the projection of f onto $\mathcal{S}_D$ relies on certain L^2 inner products of f with the basis splines being sufficiently accurate. These inner products are computed approximately according to a quadrature routine of our choosing. If we wish to determine an $\mathcal{O}(h^p)$ accurate approximation to a general, smooth f, then the spline degree D must be sufficiently high ($D \geq p - 1$) and the quadrature error should be of order $\mathcal{O}(h^p)$. If we are using something like trapezoidal rule for the quadrature, then in order to achieve the desired accuracy, it may be necessary to refine the grid we use for quadrature. Alternatively, we can select higher-order schemes for quadrature such as Gauss quadrature, and require fewer (but still carefully spaced) nodes for the quadrature step. The latter is used for the plots generated here. Note that the knots of the spline need not coincide with the nodes for the quadrature, but we will need to know the image of f at each of these points for the procedure to complete. For reference, one treatment of the influence of quadrature errors on the projection is given in ([17]). The influence of the quadrature error can also be reduced by orthonormalizing the spline basis functions with respect to the associated discrete inner product. See ([1]) for details.

6.2 Standard C-spline error bounds

As mentioned in the motivation section for C-splines, our main use for them is in establishing error bounds for exponential A-splines. First we establish some lemmas that will help us bound the spline coefficients, which we will then use to prove L^∞ error bounds for standard C-splines. The lemmas will also establish that the C-spline procedure is well-defined, since it will show the matrix C_j appearing in the algorithm is invertible. We begin demonstrating the relevant bounds in the case of uniform panel widths, and them move on to the case of nonuniform panel widths.

Lemma 6.3 (standard C-spline lemma, uniform panel width case)**.** *Let $S \in C^{D-1}$ be the C-spline of degree D to the function f. Assume we are periodically matching the Taylor polynomial of degree d, where $d \le D$. Let $a = t_0 < t_1 < \cdots < t_{M-1} < t_M = b$ be the knot sequence for our spline. Assume $f \in C^d([a,b];\mathbb{R})$. Let $h = t_m - t_{m-1}$ be the uniform panel width. Let A_j, C_j, and b_j be as in the construction of standard C-splines. Then*

$$||A_j||_\infty, \quad ||C_j||_\infty, \quad ||C_j^{-1}||_\infty, \quad \text{and} \quad ||b_j||_\infty$$

are bounded independently of h. In particular, C_j is invertible.

Proof.

$$A_j = \begin{bmatrix} \binom{0}{0} & \binom{1}{0} & \cdots & \binom{D-2}{0} & \binom{D-1}{0} \\ 0 & \binom{1}{1} & \cdots & \binom{D-2}{1} & \binom{D-1}{1} \\ 0 & 0 & \cdots & \binom{D-2}{2} & \binom{D-1}{2} \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \binom{D-1}{D-1} \end{bmatrix}$$

so

$$
\begin{aligned}
||A_j||_\infty &= \max_{0\le l\le D-1} \sum_{i=l}^{D-1} \binom{i}{l} && (140)\\
&\le \max_{0\le l\le D-1} \sum_{i=l}^{D-1} i! && (141)\\
&\le \max_{0\le l\le D-1} \sum_{i=l}^{D-1} (D-1)! && (142)\\
&\le \max_{0\le l\le D-1} (D-1)! \cdot D && (143)\\
&= D! && (144)
\end{aligned}
$$

next,

$$
b_j = \begin{bmatrix} \binom{D}{0} \\ \binom{D}{1} \\ \binom{D}{2} \\ \vdots \\ \binom{D}{D-1} \end{bmatrix}
$$

so

$$
\begin{aligned}
||b_j|| &= \max_{0\le l\le D-1} \binom{D}{l} && (145)\\
&\le D! && (146)
\end{aligned}
$$

next,

$$C_j = \left[\left[\prod_{l=1}^{D-1} A_{j+D+1-l} \right] b_{j+1} \quad \left[\prod_{l=1}^{D-2} A_{j+D+1-l} \right] b_{j+2} \quad \cdots \quad \left[\prod_{l=1}^{1} A_{j+D+1-l} \right] b_{j+D-1} \quad b_{j+D} \right] \tag{147}$$

which means C_j has only positive entries. Thus, the $||\cdot||_\infty$ norm will be realized by right-multiplying C_j by the vector $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$. We get

$$||C_j||_\infty = ||C_j \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} ||_\infty \tag{148}$$

$$= || \left[\prod_{l=1}^{D-1} A_{j+D+1-l} \right] b_{j+1} + \left[\prod_{l=1}^{D-2} A_{j+D+1-l} \right] b_{j+2} + \cdots + \left[\prod_{l=1}^{1} A_{j+D+1-l} \right] b_{j+D-1} + b_{j+D} ||_\infty \tag{149}$$

$$\leq \left[\prod_{l=1}^{D-1} ||A_{j+D+1-l}||_\infty \right] ||b_{j+1}||_\infty + \left[\prod_{l=1}^{D-2} ||A_{j+D+1-l}||_\infty \right] ||b_{j+2}||_\infty + \cdots + \left[\prod_{l=1}^{1} ||A_{j+D+1-l}||_\infty \right] ||b_{j+D-1}||_\infty + ||b_{j+D}||_\infty \tag{150}$$

at which point the claim follows for C_j, since we have already established the claims for A_j and b_j.

To prove C_j is invertible, we will show its columns are linearly independent. Since each

A_j is identical, and same for the b_js, put

$$A = \begin{bmatrix} \binom{0}{0} & \binom{1}{0} & \cdots & \binom{D-2}{0} & \binom{D-1}{0} \\ 0 & \binom{1}{1} & \cdots & \binom{D-2}{1} & \binom{D-1}{1} \\ 0 & 0 & \cdots & \binom{D-2}{2} & \binom{D-1}{2} \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \binom{D-1}{D-1} \end{bmatrix}$$

and

$$b = \begin{bmatrix} \binom{D}{0} \\ \binom{D}{1} \\ \binom{D}{2} \\ \vdots \\ \binom{D}{D-1} \end{bmatrix}$$

so that

$$C := C_j = \begin{bmatrix} A^{D-1}b & A^{D-2}b & \cdots & Ab & b \end{bmatrix} \tag{151}$$

Since $A - I$ is strictly upper triangular (i.e. with diagonal entries of 0), with positive entries above the diagonal, a simple induction argument shows that for any $k \in \{0, 1, \dots, D-1\}$,

$(A-I)^k$ will be upper triangular with a diagonal band of zeros of width k; specifically, $((A-I)^k)_{(i,j)} = 0$ for $j - i < k$, and $((A-I)^k)_{i,j)} > 0$ for $j - i \geq k$. Combining this with the fact that b is a vector with only positive entries, we arrive at

$$b = \begin{bmatrix} * \\ * \\ \vdots \\ * \\ * \\ * \end{bmatrix}, \qquad (A-I)b = \begin{bmatrix} * \\ * \\ \vdots \\ * \\ * \\ 0 \end{bmatrix}, \qquad (A-I)^2 b = \begin{bmatrix} * \\ * \\ \vdots \\ * \\ 0 \\ 0 \end{bmatrix}, \qquad \ldots, \qquad (A-I)^{D-1} b = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{152}$$

where $*$ represents a positive number. Then clearly the set of vectors

$$\{b,\ (A-I)b,\ (A-I)^2 b,\ \ldots,\ (A-I)^{D-1} b\} \tag{153}$$

is linearly independent; for they form the columns of an upper triangular matrix with nonzero diagonal entries.

Now to show the columns of C are linearly independent, we note that

$$\begin{aligned} 0 &= c_0 b + c_1 A b + c_2 A^2 b + \cdots + c_{D-1} A^{D-1} b && (154) \\ &= c_0 b + c_1 (A - I + I) b + c_2 (A - I + I)^2 b + \cdots + c_{D-1} (A - I + I)^{D-1} b && (155) \\ &= c_0 b + c_1 \left((A-I) + I \right) b + \cdots + c_{D-1} \left(\sum_{j=0}^{D-1} \begin{pmatrix} D-1 \\ j \end{pmatrix} (A-I)^j \right) b && (156) \\ &\implies c_{D-1} = 0 && (157) \end{aligned}$$

and we can continue this strategy, showing $c_{D-2} = c_{D-3} = \ldots = c_0 = 0$. Thus the columns of C_j are linearly independent, so C_j is invertible. Since C_j is composed of A and b, neither of which are dependent on h, we see that $||C_j^{-1}||_\infty$ is bounded w.r.t. h. □

We now proceed with the more general case of nonuniform panel widths.

Lemma 6.4 (standard C-spline lemma, nonuniform panel width case)**.** *Let $S \in C^{D-1}$ be the C-spline of degree D to the function f. Assume we are periodically matching the Taylor polynomial of degree d, where $d \leq D$. Let $a = t_0 < t_1 < \cdots < t_{M-1} < t_M = b$ be the knot sequence for our spline. Assume $f \in C^d([a, b]; \mathbb{R})$. Let $h_m = t_m - t_{m-1}$ be the panel widths and $h = \max h_m$ be the maximum panel width. Assume there exist $u, U > 0$ such that $0 < u < \dfrac{h_m}{h_{m-1}} < U$ for all m. Let A_j, C_j, and b_j be as in the construction of standard C-splines. Then*

$$||A_j||_\infty, \quad ||C_j||_\infty, \quad ||C_j^{-1}||_\infty, \quad \textit{and} \quad ||b_j||_\infty$$

are bounded independently of h. In particular, C_j is invertible.

Proof.

$$A_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{0}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{1}{0} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-2}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D-1}{0} \\ 0 & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{1}{1} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-2}{1} & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D-1}{1} \\ 0 & 0 & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-2}{2} & \left(\frac{h_{j+1}}{h_j}\right)^2 \binom{D-1}{2} \\ \vdots & & & & \\ 0 & 0 & \cdots & 0 & \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \binom{D-1}{D-1} \end{bmatrix}$$

so

$$||A_j||_\infty = \max_{0 \leq l \leq D-1} \sum_{i=l}^{D-1} \left(\frac{h_{j+1}}{h_j}\right)^l \binom{i}{l} \tag{158}$$

$$\leq \max_{0\leq l\leq D-1} \sum_{i=l}^{D-1} U^l \begin{pmatrix} i \\ l \end{pmatrix} \tag{159}$$

$$\leq \max_{0\leq l\leq D-1} \sum_{i=l}^{D-1} (\max(U,1))^{D-1} \begin{pmatrix} i \\ l \end{pmatrix} \tag{160}$$

$$\leq (\max(U,1))^{D-1} \max_{0\leq l\leq D-1} \sum_{i=l}^{D-1} i! \tag{161}$$

$$\leq (\max(U,1))^{D-1} \max_{0\leq l\leq D-1} \sum_{i=l}^{D-1} (D-1)! \tag{162}$$

$$\leq (\max(U,1))^{D-1} \max_{0\leq l\leq D-1} (D-1)! \cdot D \tag{163}$$

$$= (\max(U,1))^{D-1} \cdot D! \tag{164}$$

next,

$$b_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \begin{pmatrix} D \\ 0 \end{pmatrix} \\ \left(\frac{h_{j+1}}{h_j}\right)^1 \begin{pmatrix} D \\ 1 \end{pmatrix} \\ \left(\frac{h_{j+1}}{h_j}\right)^2 \begin{pmatrix} D \\ 2 \end{pmatrix} \\ \vdots \\ \left(\frac{h_{j+1}}{h_j}\right)^{D-1} \begin{pmatrix} D \\ D-1 \end{pmatrix} \end{bmatrix}$$

so

$$||b_j|| = \max_{0\leq l\leq D-1} \left(\frac{h_{j+1}}{h_j}\right)^l \begin{pmatrix} D \\ l \end{pmatrix} \tag{165}$$

$$\leq (\max(U,1))^{D-1} \cdot D! \tag{166}$$

next,

$$C_j = \left[\left[\prod_{l=1}^{D-1} A_{j+D+1-l} \right] b_{j+1} \quad \left[\prod_{l=1}^{D-2} A_{j+D+1-l} \right] b_{j+2} \quad \cdots \quad \left[\prod_{l=1}^{1} A_{j+D+1-l} \right] b_{j+D-1} \quad b_{j+D} \right] \tag{167}$$

which means C_j has only positive entries. Thus, the $||\cdot||_\infty$ norm will be realized by right-multiplying C_j by the vector $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$. We get

$$||C_j||_\infty = ||C_j \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} ||_\infty \tag{168}$$

$$= || \left[\prod_{l=1}^{D-1} A_{j+D+1-l} \right] b_{j+1} + \left[\prod_{l=1}^{D-2} A_{j+D+1-l} \right] b_{j+2} + \cdots + \left[\prod_{l=1}^{1} A_{j+D+1-l} \right] b_{j+D-1} + b_{j+D} ||_\infty \tag{169}$$

$$\leq \left[\prod_{l=1}^{D-1} ||A_{j+D+1-l}||_\infty \right] ||b_{j+1}||_\infty + \left[\prod_{l=1}^{D-2} ||A_{j+D+1-l}||_\infty \right] ||b_{j+2}||_\infty + \cdots + \left[\prod_{l=1}^{1} ||A_{j+D+1-l}||_\infty \right] ||b_{j+D-1}||_\infty + ||b_{j+D}||_\infty \tag{170}$$

at which point the claim follows for C_j, since we have already established the claims for A_j and b_j.

To prove C_j is invertible, we will show its columns are linearly independent.

Put P_m equal to the diagonal matrix whose diagonal is equal to that of A_m, i.e.

$$P_m = \begin{bmatrix} \left(\frac{h_{m+1}}{h_m}\right)^0 & & & & \\ & \left(\frac{h_{m+1}}{h_m}\right)^1 & & & \\ & & \ddots & & \\ & & & \left(\frac{h_{m+1}}{h_m}\right)^{D-2} & \\ & & & & \left(\frac{h_{m+1}}{h_m}\right)^{D-1} \end{bmatrix}$$

Observe that for each m and each $k \in \{0, 1, \ldots, D-1\}$, $A_m - P_m$ is strictly upper triangular (i.e. with diagonal entries of 0), with positive entries above the diagonal. A simple induction argument shows that for any $k \in \{0, 1, \ldots, D-1\}$, it will be the case that

$$\prod_{l=1}^{k} (A_{j+D+1-l} - P_{j+D+1-l}) \tag{171}$$

will be upper triangular with a diagonal band of zeros of width k; specifically,

$$\prod_{l=1}^{k} (A_{j+D+1-l} - P_{j+D+1-l})_{(i,j)} \begin{cases} = 0 \text{ for } j - i < k \\ > 0 \text{ for } j - i \geq k \end{cases} \tag{172}$$

Combining this with the fact that each b_m is a vector with only positive entries, we arrive at

$$b_{j+D} = \begin{bmatrix} * \\ * \\ \vdots \\ * \\ * \\ * \end{bmatrix}, \quad \prod_{l=1}^{1} (A_{j+D+1-l} - P_{j+D+1-l}) b_{j+D-1} = \begin{bmatrix} * \\ * \\ \vdots \\ * \\ * \\ 0 \end{bmatrix}, \tag{173}$$

$$\prod_{l=1}^{2}(A_{j+D+1-l}-P_{j+D+1-l})b_{j+D-2}=\begin{bmatrix} * \\ * \\ \vdots \\ * \\ 0 \\ 0 \end{bmatrix}, \quad \ldots, \quad \prod_{l=1}^{D-1}(A_{j+D+1-l}-P_{j+D+1-l})b_{j+1}=\begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{174}$$

where $*$ represents a positive number. Then clearly the set of vectors

$$\{b_{j+D},\ \prod_{l=1}^{1}(A_{j+D+1-l}-P_{j+D+1-l})b_{j+D-1},\ \prod_{l=1}^{2}(A_{j+D+1-l}-P_{j+D+1-l})b_{j+D-2},\ \ldots, \\ \prod_{l=1}^{D-1}(A_{j+D+1-l}-P_{j+D+1-l})b_{j+1}\} \tag{175}$$

is linearly independent; for they form the columns of an upper triangular matrix with nonzero diagonal entries.

Now to show the columns of C_j are linearly independent, we note that

$$0=c_0b_{j+D}+c_1\prod_{l=1}^{1}(A_{j+D+1-l})b_{j+D-1}+c_2\prod_{l=1}^{2}(A_{j+D+1-l})b_{j+D-2}+\cdots+c_{D-1}\prod_{l=1}^{D-1}(A_{j+D+1-l})b_{j+1} \tag{176}$$

$$=c_0b_{j+D}+c_1\prod_{l=1}^{1}(A_{j+D+1-l}-P_{j+D+1-l}+P_{j+D+1-l})b_{j+D-1} \tag{177}$$
$$+c_2\prod_{l=1}^{2}(A_{j+D+1-l}-P_{j+D+1-l}+P_{j+D+1-l})b_{j+D-2}+\cdots$$
$$+c_{D-1}\prod_{l=1}^{D-1}(A_{j+D+1-l}-P_{j+D+1-l}+P_{j+D+1-l})b_{j+1}$$

at which point we may continue in a manner analogous to that of the uniform case. The details are omitted.

Thus the columns of C_j are linearly independent, so C_j is invertible.

Parametrizing

$$\tau_j = \frac{h_{j+1}}{h_j} \tag{178}$$

we can express

$$g(\tau_j, \tau_{j+1}, \ldots, \tau_{j+D}) = ||C_j^{-1}||_\infty \tag{179}$$

g is the multiplicative reciprocal of the smallest singular value of C_j. We are guaranteed this is positive and finite for each set of inputs $(\tau_k)_k$, since C_j is invertible. Since singular values vary continuously with the matrix entries, g is continuous on its compact domain $[u, U]^{D+1}$. g therefore has a maximum on its domain, independent of h. This concludes the lemma. $\square$

We now prove the error bound for a C-spline approximation.

Theorem 6.5 (Standard C-spline Error Bounds). *Let $S \in C^{D-1}$ be the C-spline of degree D to the function f. Assume we are periodically matching the Taylor polynomial of degree d, where $d \leq D$. Let $a = t_0 < t_1 < \cdots < t_{M-1} < t_M = b$ be the knot sequence for our spline. Assume $f \in C^d([a,b];\mathbb{R})$. Let $h_m = t_m - t_{m-1}$ be the panels widths and $h = \max h_m$ be the maximum panel width. Assume there exist $u, U > 0$ such that $0 < u < \dfrac{h_m}{h_{m-1}} < U$ for all m. Then for each $k \in \{0, 1, \ldots, d-1\}$, we have*

$$||S^{(k)} - f^{(k)}||_\infty = \mathcal{O}(h^{d-k}) \tag{180}$$

Proof. Let $t \in [a,b]$ be arbitrary. Find $j \in \{0, 1, \ldots, M\}$ so that $t_{j-1} \leq t \leq t_{j+D+1}$ and the C-spline matches the degree d Taylor polynomial over the panels $[t_{j-1}, t_j]$ and $[t_{j+D}, t_{j+D+1}]$. Suppose t is in the jth panel; that is, $t \in [t_{j^*-1}, t_{j^*}]$. Then, making heavy use of the

Fundamental Theorem of Calculus, we get

$$S^{(k)}(t) - f^{(k)}(t) = [S^{(k)}(t) - f^{(k)}(t)] - \underbrace{[S^{(k)}(t_{j-1}) - f^{(k)}(t_{j-1})]}_{0} \tag{181}$$

$$= \int_{t_{j-1}}^{t} [S^{(k+1)}(t^{(k+1)}) - f^{(k+1)}(t^{(k+1)})]dt^{(k+1)} \tag{182}$$

$$= \int_{t_{j-1}}^{t} [S^{(k+1)}(t^{(k+1)}) - f^{(k+1)}(t^{(k+1)})] - \underbrace{[S^{(k+1)}(t_{j-1}) - f^{(k+1)}(t_{j-1})]}_{0} dt^{(k+1)} \tag{183}$$

$$= \int_{t_{j-1}}^{t} \int_{t_{j-1}}^{t^{(k+1)}} [S^{(k+2)}(t^{(k+2)}) - f^{(k+2)}(t^{(k+2)})]dt^{(k+2)}dt^{(k+1)} \tag{184}$$

$$= \cdots \tag{185}$$

$$= \int_{t_{j-1}}^{t} \int_{t_{j-1}}^{t^{(k+1)}} \int_{t_{j-1}}^{t^{(k+2)}} \cdots \int_{t_{j-1}}^{t^{(d-2)}} [S^{(d-1)}(t^{(d-1)}) - f^{(d-1)}(t^{(d-1)})]dt^{(d-1)} \cdots dt^{(k+2)}dt^{(k+2)}dt^{(k+1)} \tag{186}$$

$$= \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d-1)}} [S^{(d)}(t^{(d)}) - f^{(d)}(t^{(d)})]dt^{(d)} \cdots dt^{(k+1)} \tag{187}$$

The last step in the above chain of inequalities works despite the final integrand being (potentially) merely piecewise-continuous, e.g. if the function $g\colon [a,b] \to \mathbb{R}$ has a derivative which exists and is continuous over each panel $[t_{l-1}, t_l]$, and $t \in [a,b]$ is arbitrary, say $t_{l-1} \leq t \leq t_l$, then we have

$$g(t) - g(t_{j-1}) = [g(t) - g(t_{l-1})] + [g(t_{l-1}) - g(t_{l-2})] + [g(t_{l-2}) - g(t_{l-3})] + \cdots + [g(t_j) - g(t_{j-1})] \tag{188}$$

$$= \int_{t_{l-1}}^{t} g'(s)ds + \int_{t_{l-2}}^{t_{l-1}} g'(s)ds + \int_{t_{l-3}}^{t_{l-2}} g'(s)ds + \cdots + \int_{t_{j-1}}^{t_j} g'(s)ds \tag{189}$$

$$= \int_{t_{j-1}}^{t} g'(s)ds \tag{190}$$

which demonstrates that FTC holds regardless, as expected.

Now bounding the above quantity, we have by heavy use of the triangle inequality,

$$
\begin{aligned}
|S^{(k)}(t) - f^{(k)}(t)| &= \left| \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d-1)}} [S^{(d)}(t^{(d)}) - f^{(d)}(t^{(d)})] dt^{(d)} \cdots dt^{(k+1)} \right| && (191)\\
&\leq \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d-1)}} \left| S^{(d)}(t^{(d)}) - f^{(d)}(t^{(d)}) \right| dt^{(d)} \cdots dt^{(k+1)} && (192)\\
&\leq \left(||S^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} \right) \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d-1)}} 1 dt^{(d)} \cdots dt^{(k+1)} && (193)\\
&\leq \left(||S^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d)}||_{\infty} \right) \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d-1)}} 1 dt^{(d)} \cdots dt^{(k+1)} && (194)\\
&= \left(||S^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d)}||_{\infty} \right) \frac{(t-t_{j-1})^{(d-k)}}{(d-k)!} && (195)\\
&\leq \left(||S^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d)}||_{\infty} \right) \frac{((D+2)h)^{(d-k)}}{(d-k)!} && (196)\\
&= \left(||S^{(d)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d)}||_{\infty} \right) \cdot \frac{(D+2)^{d-k}}{(d-k)!} \cdot h^{d-k} && (197)
\end{aligned}
$$

From the standard spline form, over the mth panel $[t_{m-1}, t_m]$, we have

$$
\begin{aligned}
S_m^{(i)}(t) = a_m^i \frac{i!}{(h_m)^i} + a_m^{i+1} \frac{(i+1)!}{1!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right) + a_m^{i+2} \frac{(i+2)!}{2!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right)^2 + \cdots \quad (198)\\
+ a_m^D \frac{(D)!}{(D-i)!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right)^{D-i}
\end{aligned}
$$

for each $i \in \{0, 1, \ldots, D\}$.

Setting $i = d$ and taking the $|| \cdot ||_\infty$ norm over the mth panel $[t_{m-1}, t_m]$, we get

$$
||S_m^{(d)}||_\infty \leq |a_m^d| \frac{d!}{(h_m)^d} + |a_m^{d+1}| \frac{(d+1)!}{1!(h_m)^d} + |a_m^{d+2}| \frac{(d+2)!}{2!(h_m)^d} + \cdots + |a_m^D| \frac{(D)!}{(D-d)!(h_m)^d} \quad (199)
$$

The proof will be complete once we show that the ratio of the spline coefficient to the panel width to the dth power, $\frac{a_m^i}{(h_m)^d}$, is bounded w.r.t. h for each $i = d, d+1, \ldots, D$.

Put $\hat{h} = \max_{l=j,\ldots,j+D+1} h_l$. So $\hat{h} \leq h$.

It is enough to show this for $\hat{h}$ in place of h_m, since

$$\left|\frac{a_m^i}{(h_m)^d}\right| = \left|\frac{a_m^i}{\hat{h}^d}\right| \cdot \frac{\hat{h}^d}{(h_m)^d} \leq \left|\frac{a_m^i}{\hat{h}^d}\right| \cdot \left(\max_{j+1 \leq l \leq j+D+1} \left(\frac{h_l}{h_{l-1}}\right)\right)^{Dd} \leq \left|\frac{a_m^i}{\hat{h}^d}\right| \cdot U^{Dd} \tag{200}$$

Note that at the leftmost endpoint of the block of panels under consideration, due to matching of the degree d Taylor polynomial over this panel, we have

$$a_j^d = \frac{(h_j)^d f^{(d)}(t_{j-1})}{d!}, \quad a_j^{d+1} = 0, \quad \ldots, \quad a_j^D = 0 \tag{201}$$

The update equation for $a_m^d, \ldots, a_m^D$ is given according to

$$a_m = A_{m-1} a_{m-1} + a_{m-1}^D b_{m-1} \tag{202}$$

and recursing down,

$$\begin{aligned}
a_m &= A_{m-1} a_{m-1} + a_{m-1}^D b_{m-1} && (203)\\
&= A_{m-1}(A_{m-2} a_{m-2} + a_{m-2}^D b_{m-2}) + a_{m-1}^D b_{m-1} && (204)\\
&= A_{m-1} A_{m-2} a_{m-2} + a_{m-2}^D A_{m-1} b_{m-2} + a_{m-1}^D b_{m-1} && (205)\\
&= A_{m-1} A_{m-2}(A_{m-3} a_{m-3} + a_{m-3}^D b_{m-3}) + a_{m-2}^D A_{m-1} b_{m-2} + a_{m-1}^D b_{m-1} && (206)\\
&= A_{m-1} A_{m-2} A_{m-3} a_{m-3} + a_{m-3}^D A_{m-1} A_{m-2} b_{m-3} + a_{m-2}^D A_{m-1} b_{m-2} + a_{m-1}^D b_{m-1} && (207)\\
&= \cdots && (208)\\
&= \left(\prod_{l=1}^{m-j} A_{m-l}\right) a_j + \sum_{i=1}^{m-j} a_{m-i}^D \left(\prod_{l=1}^{i-1} A_{m-l}\right) b_{m-l} && (209)
\end{aligned}$$

We note that since $\left(\prod_{l=1}^{m-j} A_{m-l}\right)$ is upper triangular, with $||\cdot||_\infty$ norm bounded independently of h (according to the previous lemma), looking at the $(d+1)$th row of this vector equation reveals the first term contributes a magnitude of no more than $\mathcal{O}(\hat{h}^d)$, as

$$a_j^d = \frac{(h_j)^d f^{(d)}(t_{j-1})}{d!} = \mathcal{O}(\hat{h}^d), \quad a_j^{d+1} = 0, \quad \ldots, \quad a_j^D = 0 \tag{210}$$

All that remains to show is that the second term in the sum above is also of magnitude $\mathcal{O}(\hat{h}^d)$. Again by the previous lemma, the boundedness of $||A_{m-l}||_\infty$ and $||b_{m-l}||_\infty$ w.r.t. h means it is enough to show that $a_l^D = \mathcal{O}(\hat{h}^d)$ for $l = j, j+1, \ldots, m-1$. Due to the matching of the degree d Taylor polynomial on panel $[t_{j-1}, t_j]$, we already know $a_j^D = \mathcal{O}(\hat{h}^d)$.

From the procedure for standard C-splines, we have

$$\begin{bmatrix} a_{j+1}^D \\ \\ a_{j+2}^D \\ \\ \vdots \\ \\ a_{j+D}^D \end{bmatrix} = \overbrace{C_j^{-1}}^{\text{bounded}} \left(\underbrace{a_{j+(D+1)} - \overbrace{A_{j+D}A_{j+D-1}\cdots A_{j+1}}^{\text{bounded}}(A_j a_j}_{(*)} + \overbrace{a_j^D b_j}^{\text{small}}) \right) \tag{211}$$

but the portion $(*)$ is also small (in particular, $\mathcal{O}(\hat{h}^d)$), because it represents the difference in weighted coefficients over panel $[t_{j+D}, t_{j+D+1}]$ between the degree $(d-1)$ Taylor polynomial of f centered at node t_{j+D}, and the degree $(d-1)$ Taylor polynomial of f centered at node t_{j-1} but recentered at node t_{j+D}. According to the Taylor error formula, we do indeed get that this difference in coefficients is $\mathcal{O}((D\hat{h})^d) = \mathcal{O}(\hat{h}^d)$, which is all that is needed. This concludes the proof. $\square$

In figures (41) through (46), we approximate a sine function by a 5th degree C-spline that periodically matches the 3rd degree Taylor polynomial of f. With each successive approximation, we refine the grid by a factor of 2. Thus, according to the error bound just derived, we expect the error to roughly diminish by a factor of $2^3 = 8$ with every successive refinement. We then compare the first derivatives of the function and this C-spline. The theoretical error bounds just derived suggest the error in the first derivative should shrink by a factor of $2^2 = 4$ with each successive grid refinement. This is observed in the error estimates given. The robustness of the error bounds is tested by using nonuniform panel widths.

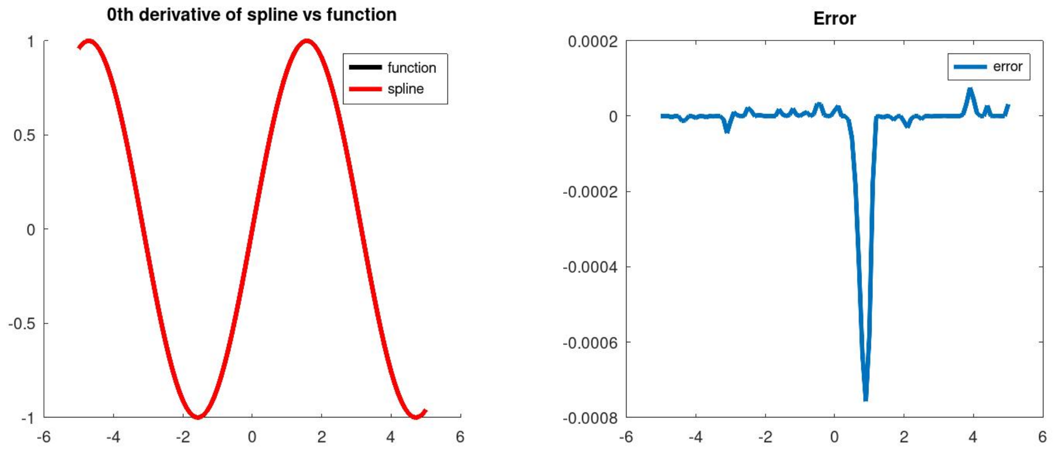


Figure 41: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 0th derivative of spline vs function, 160 panels of random width. max error = 7.6e-4

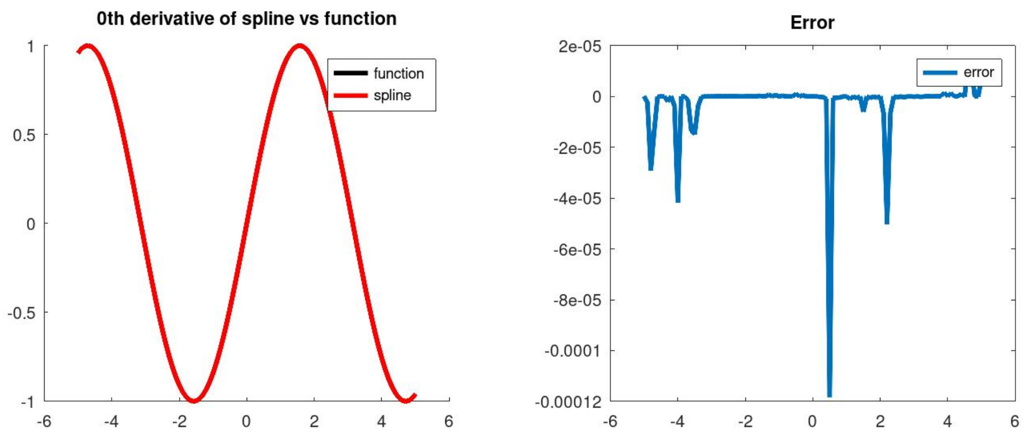


Figure 42: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 0th derivative of spline vs function, 320 panels of random width. max error = 1.2e-4

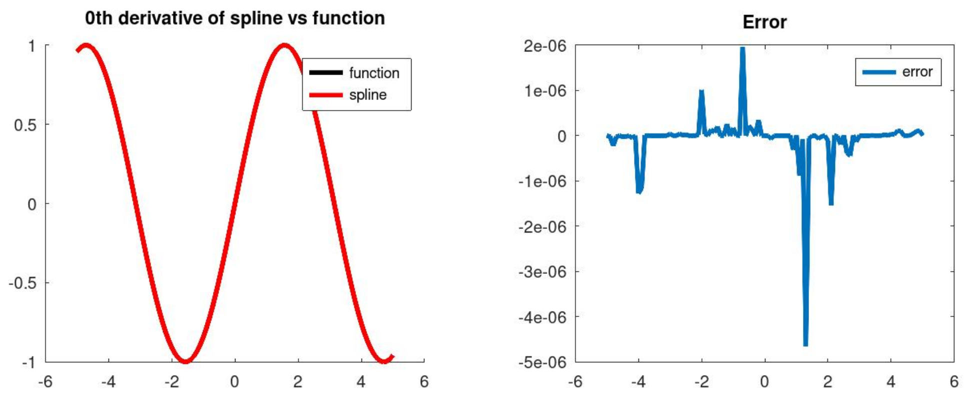


Figure 43: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 0th derivative of spline vs function, 640 panels of random width. max error = 4.6e-6

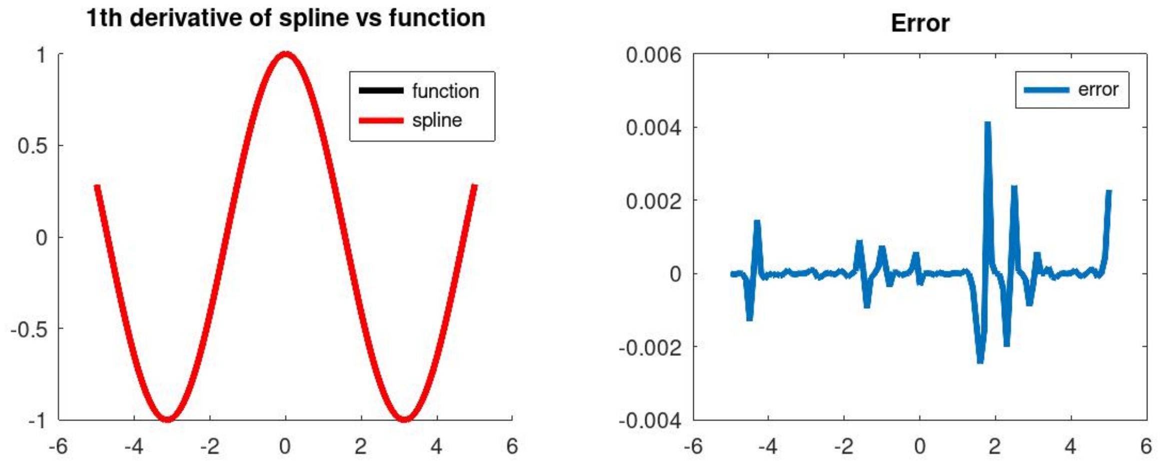


Figure 44: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 1st derivative of spline vs function, 160 panels of random width. max error = 4.1e-3

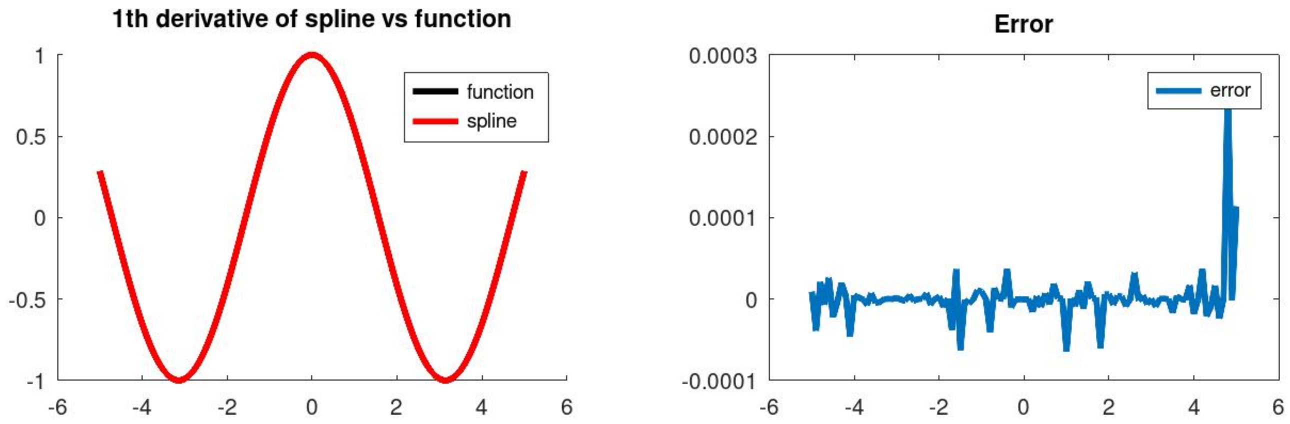


Figure 45: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 1st derivative of spline vs function, 320 panels of random width. max error = 2.7e-4

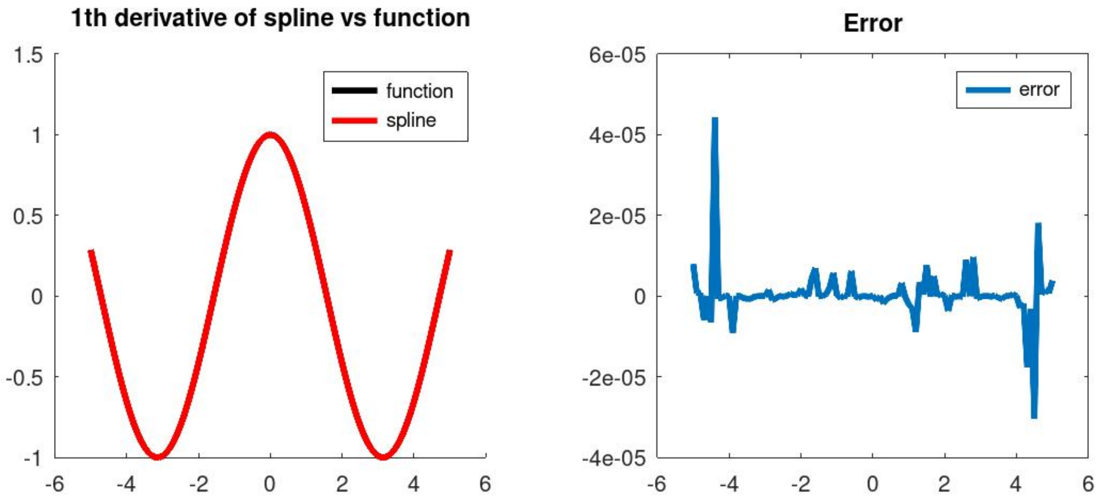


Figure 46: Quintic C-spline approximation of $f(t) = \sin(t)$ matching 3rd degree Taylor polynomial every 6 panels, 1st derivative of spline vs function, 160 panels of random width. max error = 4.4e-5

In the next two figures, figures (47) and (48), we demonstrate that the C-spline approximation is impervious to faulty initial data. This is due to its nature as a semi-local approximation.

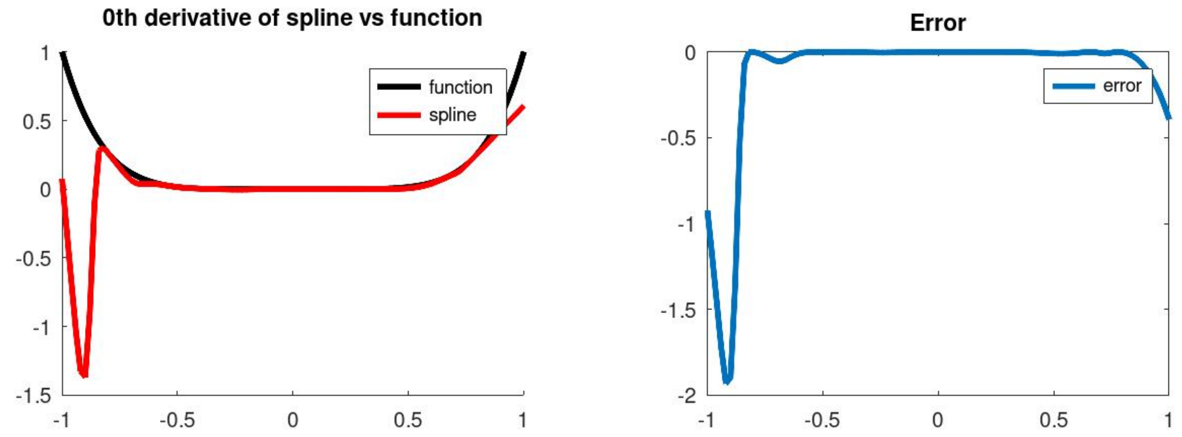


Figure 47: Quartic C-spline approximation of $f(t) = t^6$ matching 1st degree Taylor polynomial every 5 panels, 0th derivative of spline vs function, 40 panels of random width. Faulty initial data.

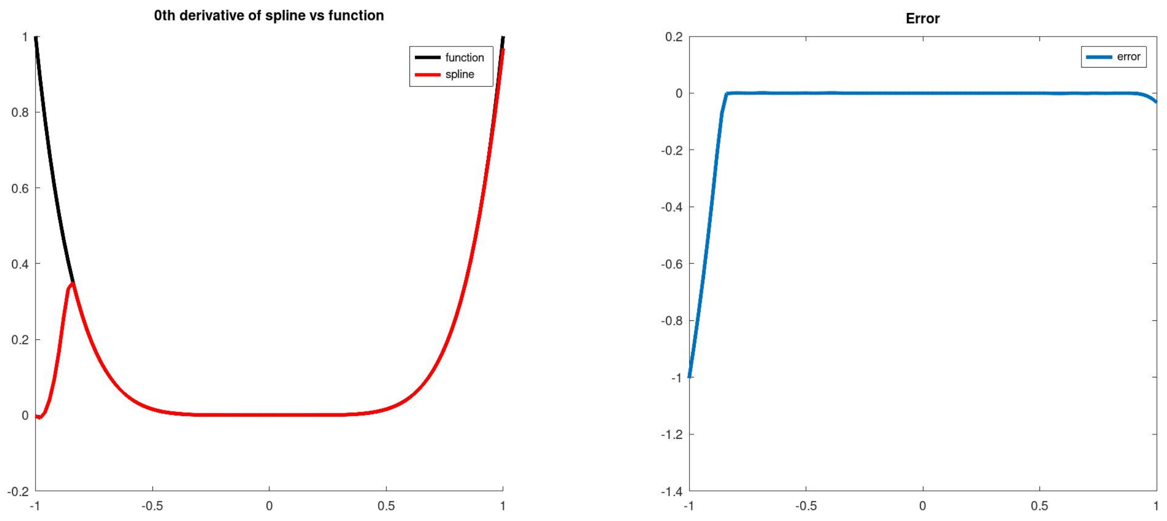


Figure 48: Cubic C-spline approximation of $f(t) = t^6$ matching 2nd degree Taylor polynomial every 4 panels, 0th derivative of spline vs function, 80 panels of random width. Faulty initial data.

6.3 Exponential C-spline error bounds

The error bound for exponential C-splines is the main tool for establishing error bounds for exponential A-splines.

In the following we give the general error bound for exponential C-splines.

Theorem 6.6 (Exponential C-spline error bounds)**.** *Let $S \in C^{D+1}$ be the exponential C-spline with polynomial part of degree D to the function f. Assume we are periodically matching the Taylor polynomial of degree $(d+2)$, where $d \leq D$. Let $a = t_0 < t_1 < \cdots < t_{M-1} < t_M = b$ be the knot sequence for our spline. Assume $f \in C^{d+2}([a,b];\mathbb{R})$. Let $\rho_l > 0$, $l = 1, 2, \dots, M$ be the spline tensions over each panel. Let $h_m = t_m - t_{m-1}$ be the panel widths and $h = \max h_m$ be the maximum panel width. Assume there exist $u, U > 0$ such that $0 < u < \dfrac{h_m}{h_{m-1}} < U$ for all m. Then for each $k \in \{0, 1, \dots, d+1\}$, we have*

$$||S^{(k)} - f^{(k)}||_\infty = \mathcal{O}(h^{d+2-k}) \tag{212}$$

Proof. Let $t \in [a,b]$ be arbitrary. Find $j \in \{0,1,\dots,M\}$ so that $t_{j-1} \leq t \leq t_{j+D+1}$ and the exponential C-spline matches the degree $(d+2)$ Taylor polynomial over the panels $[t_{j-1}, t_j]$ and $[t_{j+D}, t_{j+D+1}]$. Suppose t is in the jth panel; that is, $t \in [t_{j^*-1}, t_{j^*}]$. Then, making heavy use of the Fundamental Theorem of Calculus, we get

$$S^{(k)}(t) - f^{(k)}(t) = [S^{(k)}(t) - f^{(k)}(t)] - \underbrace{[S^{(k)}(t_{j-1}) - f^{(k)}(t_{j-1})]}_{0} \tag{213}$$

$$= \int_{t_{j-1}}^{t} [S^{(k+1)}(t^{(k+1)}) - f^{(k+1)}(t^{(k+1)})]dt^{(k+1)} \tag{214}$$

$$= \int_{t_{j-1}}^{t} [S^{(k+1)}(t^{(k+1)}) - f^{(k+1)}(t^{(k+1)})] - \underbrace{[S^{(k+1)}(t_{j-1}) - f^{(k+1)}(t_{j-1})]}_{0} dt^{(k+1)} \tag{215}$$

$$= \int_{t_{j-1}}^{t} \int_{t_{j-1}}^{t^{(k+1)}} [S^{(k+2)}(t^{(k+2)}) - f^{(k+2)}(t^{(k+2)})]dt^{(k+2)}dt^{(k+1)} \tag{216}$$

$$= \cdots \tag{217}$$

$$= \int_{t_{j-1}}^{t} \int_{t_{j-1}}^{t^{(k+1)}} \int_{t_{j-1}}^{t^{(k+2)}} \cdots \int_{t_{j-1}}^{t^{(d)}} [S^{(d+1)}(t^{(d+1)}) - f^{(d+1)}(t^{(d+1)})]dt^{(d+1)} \cdots dt^{(k+2)}dt^{(k+2)}dt^{(k+1)} \tag{218}$$

$$= \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d+1)}} [S^{(d+2)}(t^{(d+2)}) - f^{(d+2)}(t^{(d+2)})]dt^{(d+2)} \cdots dt^{(k+1)} \tag{219}$$

The last step in the above chain of inequalities works despite the final integrand being (potentially) merely piecewise-continuous, e.g. if the function $g\colon [a,b] \to \mathbb{R}$ has a derivative which exists and is continuous over each panel $[t_{l-1}, t_l]$, and $t \in [a,b]$ is arbitrary, say $t_{l-1} \leq t \leq t_l$, then we have

$$g(t) - g(t_{j-1}) = [g(t) - g(t_{l-1})] + [g(t_{l-1}) - g(t_{l-2})] + [g(t_{l-2}) - g(t_{l-3})] + \cdots + [g(t_j) - g(t_{j-1})] \tag{220}$$

$$= \int_{t_{l-1}}^{t} g'(s)ds + \int_{t_{l-2}}^{t_{l-1}} g'(s)ds + \int_{t_{l-3}}^{t_{l-2}} g'(s)ds + \cdots + \int_{t_{j-1}}^{t_j} g'(s)ds \tag{221}$$

$$= \int_{t_{j-1}}^{t} g'(s)ds \tag{222}$$

which demonstrates that FTC holds regardless, as expected.

Now bounding the above quantity, we have by heavy use of the triangle inequality,

$$
\begin{aligned}
|S^{(k)}(t) - f^{(k)}(t)| &= \left| \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d+1)}} [S^{(d+2)}(t^{(d+2)}) - f^{(d+2)}(t^{(d+2)})] dt^{(d+2)} \cdots dt^{(k+1)} \right| && (223)\\
&\leq \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d+1)}} \left| S^{(d+2)}(t^{(d+2)}) - f^{(d+2)}(t^{(d+2)}) \right| dt^{(d+2)} \cdots dt^{(k+1)} && (224)\\
&\leq \left(||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} \right) \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d+1)}} 1 dt^{(d+2)} \cdots dt^{(k+1)} && (225)\\
&\leq \left(||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d+2)}||_{\infty} \right) \int_{t_{j-1}}^{t} \cdots \int_{t_{j-1}}^{t^{(d+1)}} 1 dt^{(d+2)} \cdots dt^{(k+1)} && (226)\\
&= \left(||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d+2)}||_{\infty} \right) \frac{(t - t_{j-1})^{(d+2-k)}}{(d+2-k)!} && (227)\\
&\leq \left(||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d+2)}||_{\infty} \right) \frac{((D+4)h)^{(d+2-k)}}{(d+2-k)!} && (228)\\
&= \left(||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]} + ||f^{(d+2)}||_{\infty} \right) \cdot \frac{(D+4)^{d+2-k}}{(d+2-k)!} \cdot h^{d+2-k} && (229)
\end{aligned}
$$

The proof will be complete once we show that $||S^{(d+2)}||_{\infty,[t_{j-1},t_{j+D+1}]}$ is bounded w.r.t. h.

From the exponential spline form, we have

$$
\begin{aligned}
S_m^{(i)}(t) = a_m^i \frac{i!}{(h_m)^i} + a_m^{i+1} \frac{(i+1)!}{1!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right) + a_m^{i+2} \frac{(i+2)!}{2!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right)^2 + \cdots \quad (230)\\
+ a_m^D \frac{(D)!}{(D-i)!(h_m)^i} \left(\frac{t - t_{m-1}}{h_m} \right)^{D-i} + a_m^{D+1} \left(\frac{-\rho_m}{h_m} \right)^i \exp\left(\rho_m \frac{t_{m-1} - t}{h_m} \right)\\
+ a_m^{D+2} \left(\frac{\rho_m}{h_m} \right)^i \exp\left(\rho_m \frac{t - t_m}{h_m} \right)
\end{aligned}
$$

for each $i \in \{0, 1, \ldots, D+2\}$.

Setting $i = d+2$, $\hat{h} = \max_{l=j,\ldots,j+D+3} h_l$, and taking the $|| \cdot ||_\infty$ norm over the mth panel

$[t_{m-1}, t_m]$, we get

$$||S_m^{(d+2)}||_\infty \leq |a_m^{d+2}|\frac{(d+2)!}{(h_m)^{d+2}} + |a_m^{d+3}|\frac{(d+3)!}{1!(h_m)^{d+2}} + |a_m^{d+4}|\frac{(d+4)!}{2!(h_m)^{d+2}} + \cdots + |a_m^D|\frac{(D)!}{(D-(d+2))!(h_m)^{d+2}} + ||g_m||_\infty \tag{231}$$

$$\leq U^{(D+2)(d+2)}(|a_m^{d+2}|\frac{(d+2)!}{(\hat{h})^{d+2}} + |a_m^{d+3}|\frac{(d+3)!}{1!(\hat{h})^{d+2}} + |a_m^{d+4}|\frac{(d+4)!}{2!(\hat{h})^{d+2}} + \cdots + |a_m^D|\frac{(D)!}{(D-(d+2))!(\hat{h})^{d+2}}) + ||g_m||_\infty \tag{232}$$

where the last inequality is due to

$$\frac{1}{(h_m)^{d+2}} = \frac{1}{\hat{h}^{d+2}} \cdot \frac{\hat{h}^{d+2}}{(h_m)^{d+2}} \leq \frac{1}{\hat{h}^{d+2}} \cdot \left(\max_{j+1 \leq l \leq j+D+3} \left(\frac{h_l}{h_{l-1}} \right) \right)^{(D+2)(d+2)} \leq \frac{1}{\hat{h}^{d+2}} \cdot U^{(D+2)(d+2)} \tag{233}$$

and where

$$g_m(t) = a_m^{D+1} \left(\frac{-\rho_m}{h_m} \right)^{d+2} \exp\left(\rho_m \frac{t_{m-1} - t}{h_m} \right) + a_m^{D+2} \left(\frac{\rho_m}{h_m} \right)^{d+2} \exp\left(\rho_m \frac{t - t_m}{h_m} \right) \tag{234}$$

Thus it will be sufficient to show that

$$|a_m^l| = \mathcal{O}(\hat{h}^{d+2}) \tag{235}$$

for $l = d+2, d+3, \ldots, D$ and

$$||g_m||_\infty = \mathcal{O}(1) \tag{236}$$

i.e. g_m is bounded w.r.t. h.

We will do this by rephrasing the linear system for the block of exponential spline coefficients according to the coefficient substitutions

$$c_m^i = \frac{a_m^i}{(h_m)^i} \qquad i = 0, 1, \ldots, D \tag{237}$$

$$c_m^{D+1} = \frac{a_m^{D+1}}{(h_m)^{D+1}} \tag{238}$$

$$c_m^{D+2} = \frac{a_m^{D+2}}{(h_m)^{D+1}} \tag{239}$$

for $m = 1, 2, \dots, M$.

Equivalently,

$$\underbrace{\begin{bmatrix} c_m^0 \\ c_m^1 \\ c_m^2 \\ \vdots \\ c_m^D \\ c_m^{D+1} \\ c_m^{D+2} \end{bmatrix}}_{=:c_m} = \underbrace{\begin{bmatrix} (h_m)^0 & & & & & & \\ & (h_m)^{-1} & & & & & \\ & & (h_m)^{-2} & & & & \\ & & & \ddots & & & \\ & & & & (h_m)^{-D} & & \\ & & & & & (h_m)^{-(D+1)} & \\ & & & & & & (h_m)^{-(D+1)} \end{bmatrix}}_{=:H_m^{-1}} \underbrace{\begin{bmatrix} a_m^0 \\ a_m^1 \\ a_m^2 \\ \vdots \\ a_m^D \\ a_m^{D+1} \\ a_m^{D+2} \end{bmatrix}}_{a_m} \tag{240}$$

or in compact notation,

$$c_m = H_m^{-1} a_m \tag{241}$$

It is noted that H_m is

$$H_m = \begin{bmatrix} (h_m)^0 & & & & & & \\ & (h_m)^1 & & & & & \\ & & (h_m)^2 & & & & \\ & & & \ddots & & & \\ & & & & (h_m)^D & & \\ & & & & & (h_m)^{(D+1)} & \\ & & & & & & (h_m)^{(D+1)} \end{bmatrix}$$

We will be done when we show $|c_m^i| = \mathcal{O}(1)$ for $i = j, \ldots, j + D + 3$ and $||g_m||_\infty = \mathcal{O}(1)$. In fact, when bounding the latter, we may ignore second-order terms and just show that $|g_m(t_{m-1})| = \mathcal{O}(1)$ for $m = j, \ldots, j + D + 3$.

Now, to rephrase the linear system for the block of spline coefficients

$$\underbrace{\begin{bmatrix} \mathtt{I_{D+3}} & & & & & \\ -R_j & L_{j+1} & & & & \\ & -R_{j+1} & L_{j+2} & & & \\ & & -R_{j+2} & L_{j+3} & & \\ & & & \ddots & & \\ & & & & -R_{j+D+2} & L_{j+D+3} \\ & & & & & \tilde{L}_{j+D+3} \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} a_j^0 \\ \vdots \\ a_j^{D+2} \\ \\ \vdots \\ \\ a_{j+D+3}^0 \\ \vdots \\ a_{j+D+3}^{D+2} \end{bmatrix}}_{\vec{a}} = \underbrace{\begin{bmatrix} a_j^0 \\ \vdots \\ a_j^{D+2} \\ 0 \\ \vdots \\ 0 \\ (h_{j+D+3})^0 f(t_{j+D+2})/0! \\ \vdots \\ (h_{j+D+3})^{D+2} f^{(D+2)}(t_{j+D+2})/(D+2)! \end{bmatrix}}_{\vec{b}} \tag{242}$$

where

$$\tilde{L}_{j+D+3} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & e^{-\rho_{j+D+3}} \\ 0 & 1 & 0 & \cdots & 0 & \frac{-\rho_{j+D+3}}{1!} & \frac{\rho_{j+D+3}}{1!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 1 & \cdots & 0 & \frac{(-\rho_{j+D+3})^2}{2!} & \frac{(\rho_{j+D+3})^2}{2!} e^{-\rho_{j+D+3}} \\ & & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{(-\rho_{j+D+3})^D}{D!} & \frac{(\rho_{j+D+3})^D}{D!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+D+3})^{D+1}}{(D+1)!} & \frac{(\rho_{j+D+3})^{D+1}}{(D+1)!} e^{-\rho_{j+D+3}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+D+3})^{D+2}}{(D+2)!} & \frac{(\rho_{j+D+3})^{D+2}}{(D+2)!} e^{-\rho_{j+D+3}} \end{bmatrix} \in \mathbb{R}^{(D+3)\times(D+3)}$$

$$L_{j+1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & e^{-\rho_{j+1}} \\ 0 & 1 & 0 & \cdots & 0 & \frac{-\rho_{j+1}}{1!} & \frac{\rho_{j+1}}{1!} e^{-\rho_{j+1}} \\ 0 & 0 & 1 & \cdots & 0 & \frac{(-\rho_{j+1})^2}{2!} & \frac{(\rho_{j+1})^2}{2!} e^{-\rho_{j+1}} \\ & & & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \frac{(-\rho_{j+1})^D}{D!} & \frac{(\rho_{j+1})^D}{D!} e^{-\rho_{j+1}} \\ 0 & 0 & 0 & \cdots & 0 & \frac{(-\rho_{j+1})^{D+1}}{(D+1)!} & \frac{(\rho_{j+1})^{D+1}}{(D+1)!} e^{-\rho_{j+1}} \end{bmatrix}$$

and

$$R_j = \begin{bmatrix} \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{0}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{1}{0} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^0 \binom{D}{0} & \left(\frac{h_{j+1}}{h_j}\right)^0 \frac{(-\rho_j)^0}{0!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^0 \frac{(\rho_j)^0}{0!} \\ & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{1}{1} & \cdots & \left(\frac{h_{j+1}}{h_j}\right)^1 \binom{D}{1} & \left(\frac{h_{j+1}}{h_j}\right)^1 \frac{(-\rho_j)^1}{1!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^1 \frac{(\rho_j)^1}{1!} \\ & & \ddots & & \vdots & \\ & & & \left(\frac{h_{j+1}}{h_j}\right)^D \binom{D}{D} & \left(\frac{h_{j+1}}{h_j}\right)^D \frac{(-\rho_j)^D}{D!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^D \frac{(\rho_j)^D}{D!} \\ & & & & \left(\frac{h_{j+1}}{h_j}\right)^{D+1} \frac{(-\rho_j)^{D+1}}{(D+1)!} e^{-\rho_j} & \left(\frac{h_{j+1}}{h_j}\right)^{D+1} \frac{(\rho_j)^{D+1}}{(D+1)!} \end{bmatrix}$$

we get

$$EH\vec{c} = \vec{b} \tag{243}$$

where $\vec{c} = \begin{bmatrix} c_j \\ \vdots \\ c_{j+D+3} \end{bmatrix}$ and $H = \begin{bmatrix} H_j & & & \\ & H_{j+1} & & \\ & & \ddots & \\ & & & H_{j+D+3} \end{bmatrix}$.

Now define

$$\hat{H}_m = \begin{bmatrix} (h_m)^0 & & & & & \\ & (h_m)^1 & & & & \\ & & (h_m)^2 & & & \\ & & & \ddots & & \\ & & & & (h_m)^D & \\ & & & & & (h_m)^{D+1} \end{bmatrix}$$

and

$$\hat{H} = \begin{bmatrix} H_j & & & & \\ & \hat{H}_{j+1} & & & \\ & & \ddots & & \\ & & & \hat{H}_{j+D+3} & \\ & & & & H_{j+D+3} \end{bmatrix}$$

Left-multiplying both sides of (243) by $\hat{H}^{-1}$ yields

$$\hat{H}^{-1}EH\vec{c} = \hat{H}\vec{b} = \begin{bmatrix} c_j^0 \\ \vdots \\ c_j^{D+2} \\ 0 \\ \vdots \\ 0 \\ f(t_{j+D+2})/0! \\ \vdots \\ f^{(D)}(t_{j+D+2})/D! \\ f^{(D+1)}(t_{j+D+2})/(D+1)! \\ h_{j+D+3}f^{(D+2)}(t_{j+D+2})/(D+2)! \end{bmatrix} \tag{244}$$

A careful treatment of matrix multiplication yields

$$\hat{H}^{-1}EH = \begin{bmatrix} H_j^{-1}\mathtt{I}_{D+3}H_j & & & & \\ -\hat{H}_{j+1}^{-1}R_jH_j & \hat{H}_{j+1}^{-1}L_{j+1}H_{j+1} & & & \\ & -\hat{H}_{j+2}^{-1}R_{j+1}H_{j+1} & \hat{H}_{j+2}^{-1}L_{j+2}H_{j+2} & & \\ & & \ddots & & \\ & & -\hat{H}_{j+D+3}^{-1}R_{j+D+2}H_{j+D+2} & \hat{H}_{j+D+3}^{-1}L_{j+D+3}H_{j+D+3} \\ & & & H_{j+D+3}^{-1}\tilde{L}_{j+D+3}H_{j+D+3} \end{bmatrix} \tag{245}$$

The effect on the Rs and L is the diagonal entries are unaffected, but every subsequent diagonal above the main diagonal gets multiplied by a factor of h_m for an appropriate m, except that the $(D+3)$rd column has one less factor of h_m and the $(D+3)$rd row has one more one factor of h_m. In the limit as $h \to 0$, we can see from this that

$$\hat{H}_{m+1}^{-1}R_mH_m \to \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & \frac{(-\rho_m)^{D+1}}{(D+1)!}e^{-\rho_m} & \frac{(\rho_m)^{D+1}}{(D+1)!} \end{bmatrix} \tag{246}$$

$$\hat{H}_{m}^{-1}L_mH_m \to \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & \frac{(-\rho_m)^{D+1}}{(D+1)!} & \frac{(\rho_m)^{D+1}}{(D+1)!}e^{-\rho_m} \end{bmatrix} \tag{247}$$

and

$$\hat{H}_m^{-1}\tilde{L}_m H_m \to \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & \frac{(-\rho_m)^{D+1}}{(D+1)!} & \frac{(\rho_m)^{D+1}}{(D+1)!}e^{-\rho_m} \\ & & & & \frac{(-\rho_m)^{D+2}}{(D+2)!} & \frac{(\rho_m)^{D+2}}{(D+2)!}e^{-\rho_m} \end{bmatrix} \tag{248}$$

From the resulting form of the matrix $\hat{H}^{-1}EH$ in the limit as $h \to 0$, we can see that the dominant behavior of matrix equation (244) is for information to be passed between corresponding coefficients c_m^i for fixed i and different m, for $i = 0, 1, \dots, D$, and also for information to be passed between the exponential coefficients c_m^{D+1}, c_m^{D+2} for varying m. There are other interactions, but the information passed in other ways is all second-order in comparison ($\mathcal{O}(h)$).

By forward substitution and the fact that

$$\hat{H}\vec{b}$$

is $\mathcal{O}(1)$, we deduce that
$c_j^0, \dots, c_j^{D+2}, \dots, c_{j+D+3}^0, \dots, c_{j+D+3}^{D+2}$ are all $\mathcal{O}(1)$ w.r.t. h.

All that remains is to show that $|g_m(t_{m-1})| = \mathcal{O}(1)$.

$$g_m(t_{m-1}) = a_m^{D+1}\left(\frac{-\rho_m}{h_m}\right)^{d+2} + a_m^{D+2}\left(\frac{\rho_m}{h_m}\right)^{d+2} e^{-\rho_m} \tag{249}$$

$$= [c_m^{D+1}(-\rho_m)^{d+2} + c_m^{D+2}(\rho_m)^{d+2}e^{-\rho_m}]/h_m \tag{250}$$

We may compute the last two coefficients over the rightmost panel in the block directly, since they are solely determined by the constraint

$$\tilde{L}_{j+D+3}H_{j+D+3}c_{j+D+3} = \begin{bmatrix} (h_{j+D+3})^0 f(t_{j+D+2})/0! \\ \vdots \\ (h_{j+D+3})^{D+2} f^{(D+2)}(t_{j+D+2})/(D+2)! \end{bmatrix} \tag{251}$$

i.e.

$$\begin{cases} \dfrac{(-\rho_{j+D+3})^{D+1}}{(D+1)!}h_{j+D+3}^{D+1}c_{j+D+3}^{D+1} + \dfrac{(\rho_{j+D+3})^{D+1}}{(D+1)!}e^{-\rho_{j+D+3}}h_{j+D+3}^{D+1}c_{j+D+3}^{D+2} \\ \quad = (h_{j+D+3})^{D+1}f^{(D+1)}(t_{j+D+2})/(D+1)! \\ \dfrac{(-\rho_{j+D+3})^{D+2}}{(D+2)!}h_{j+D+3}^{D+1}c_{j+D+3}^{D+1} + \dfrac{(\rho_{j+D+3})^{D+2}}{(D+2)!}e^{-\rho_{j+D+3}}h_{j+D+3}^{D+1}c_{j+D+3}^{D+2} \\ \quad = (h_{j+D+3})^{D+2}f^{(D+2)}(t_{j+D+2})/(D+2)! \end{cases} \tag{252}$$

and dividing by h_{j+D+3}^{D+1}, this becomes

$$\begin{cases} \dfrac{(-\rho_{j+D+3})^{D+1}}{(D+1)!}c_{j+D+3}^{D+1} + \dfrac{(\rho_{j+D+3})^{D+1}}{(D+1)!}e^{-\rho_{j+D+3}}c_{j+D+3}^{D+2} = f^{(D+1)}(t_{j+D+2})/(D+1)! \\ \dfrac{(-\rho_{j+D+3})^{D+2}}{(D+2)!}c_{j+D+3}^{D+1} + \dfrac{(\rho_{j+D+3})^{D+2}}{(D+2)!}e^{-\rho_{j+D+3}}c_{j+D+3}^{D+2} = h_{j+D+3}f^{(D+2)}(t_{j+D+2})/(D+2)! \end{cases} \tag{253}$$

The LHS of the last equation above is an $\mathcal{O}(h_{j+D+3})$ multiple of $g_{j+D+3}(t_{j+D+2})$, so in the case $d = D$, this tells us that $g_{j+D+3}(t_{j+D+2}) = \mathcal{O}(1)$, since the RHS of this equation is also $\mathcal{O}(1)$. If $d < D$, then $g_{j+D+3}(t_{j+D+2})$ will be an $\mathcal{O}(1)$ multiple of an earlier line of the equation $H_{j+D+3}^{-1}\tilde{L}_{j+D+3}H_{j+D+3}c_{j+D+3} = \mathcal{O}(1)$ after moving an $\mathcal{O}(1)$ term over to the RHS, since the derivatives of f past the $(d+2)$nd will be set to zero. In either case, this tells us $g_{j+D+3}(t_{j+D+2}) = \mathcal{O}(1)$. Similarly, since the degree $(d+2)$ Taylor polynomial of f is also matched on panel $[t_{j-1}, t_j]$, we get $g_j(t_{j-1}) = \mathcal{O}(1)$, by the same computation. The intermediate values, $g_m(t_{m-1})$ for $m = j+1, j+2, \ldots, j+D+2$, are then deduced to be $\mathcal{O}(1)$ according to the manner in which information is passed in matrix equation (244) in the limit $h \to 0$ as discussed above. This completes the proof. □

In figures (49) and (50), we approximate a 6th degree polynomial by a quintic exponential spline (i.e. with cubic polynomial part), periodically matching the 4th degree Taylor polynomial. The panel widths are random, but the error still diminishes by a factor near the expected factor $2^4 = 16$. With continued refinement of the grid, the theoretical factor by which the error shrinks is matched more closely.

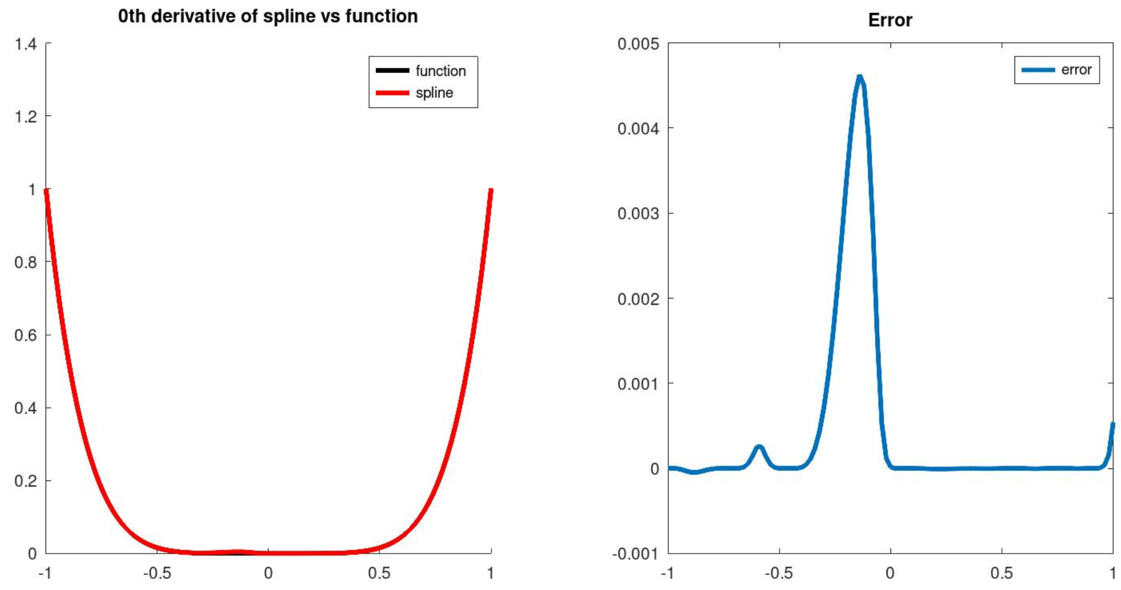


Figure 49: Quintic exponential C-spline (i.e. with polynomial part of degree 3) approximation of $f(t) = t^6$ matching 4th degree Taylor polynomial every 6 panels, 0th derivative of spline vs function, 40 panels of random width. max error = 4.6e-3

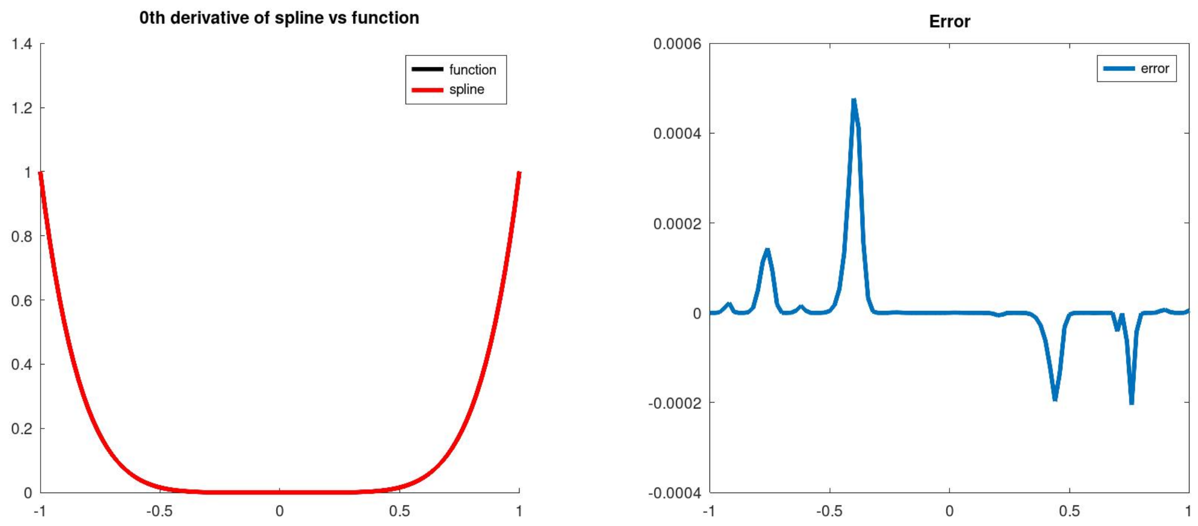


Figure 50: Quintic exponential C-spline (i.e. with polynomial part of degree 3) approximation of $f(t) = t^6$ matching 4th degree Taylor polynomial every 6 panels, 0th derivative of spline vs function, 80 panels of random width. max error = 4.8e-4

6.4 Exponential A-spline error bound

With an exponential C-spline error bound, we can then deduce an error bound for exponential A-splines. The reason for this, is the exponential A-spline procedure finds the closest spline in $\mathcal{E}_D$, the exponential spline space with polynomial part degree D and the pre-specified knots $t_0 < \cdots < t_M$ and tensions $\rho_1, \ldots, \rho_M > 0$; if we already know that some exponential spline satisfies some closeness condition to f (in the L^2 sense), then we may immediately conclude the best spline in $\mathcal{E}_D$ also satisfies this condition.

Theorem 6.7 (Exponential A-spline error bound)**.** *Let $E\colon C^{D+2}([a,b];\mathbb{R}) \to \mathcal{E}_D$ be the exponential spline operator defined by the exponential A-spline procedure, with knots $a = t_0 < t_1 < \cdots < t_{M-1} < t_M = b$ and tensions $\rho_1, \ldots, \rho_M > 0$. Then*

$$||f - E(f)||_2 = \mathcal{O}(h^{D+2}) \tag{254}$$

Proof. The exponential C-spline from the preceding section is an element of $\mathcal{E}_D$ that satisfies the bound, and by the exponential A-spline procedure, $E(f)$ is even closer to f than the exponential C-spline, and so the bound follows trivially. $\square$

Note that the exponential C-splines satisfy an L^∞ error bound which implies they also satisfy the same error bound (multiplied by $\sqrt{b-a}$) in the L^2 sense. This is the norm we need for exponential A-splines, because afterall, the exponential A-spline procedure finds optimal splines with respect to the L^2 error rather than the L^∞ error.

It is hypothesized that the actual error is an additional factor of h better than the one just presented, but in order to prove this a different technique would be needed, since exponential C-splines are only accurate to order $\mathcal{O}(h^{D+2})$. Though, they are accurate to order $\mathcal{O}(h^{D+3})$ once every $(D+3)$ many panels, by the Taylor remainder formula. This "periodic" bound doesn't trickle down to exponential A-splines however, due to the shift from L^∞ norm to L^2 norm. We give this error bound as a conjecture.

Conjecture. (Optimal bounds for exponential A-splines).
Let S_D be the exponential A-spline approximation to f with polynomial part degree D, with knots $t_0 < \cdots < t_M$ and tensions $\rho_1, \ldots, \rho_M > 0$. Let $h = \max(t_j - t_{j-1})$ be the maximum panel width. Then

$$||f - S_D||_\infty = ||f^{(j)}||_\infty \mathcal{O}(h^j) \tag{255}$$

for $j = 1, \ldots, D+3$.

We give numerical evidence of the veracity of the above claim in the following.

One could of course attempt to prove the tighter error bound with the additional factor of h by using a technique analogous to that used in finding standard A-spline error bounds, but the difficulty there lies in finding the exponential analogue to B-splines from which to form a smoothing spline as we did for standard A-splines. Such analogues do exist (see e.g.

[5], [24]), but they are for the alternate description of exponential splines described when we introduced exponential splines here, so that will have limited usefulness here.

Closeness of derivatives

Just like for standard A-splines, we suspect there is a bound on the closeness of derivatives of exponential A-splines to the corresponding derivatives of the functions they approximate. We summarize this in the following conjecture.

Conjecture. (Closeness of derivatives for exponential A-splines).
Let S_D be the exponential A-spline approximation to f with polynomial part degree D, with knots $t_0 < \cdots < t_M$ and tensions $\rho_1, \ldots, \rho_M > 0$. Let $h = \max(t_j - t_{j-1})$ be the maximum panel width. Then

$$||f^{(k)} - S_D^{(k)}||_\infty = ||f^{(j)}||_\infty \mathcal{O}(h^{j-k}) \tag{256}$$

for $k = 0, 1, \ldots, D+2$ and $j = k+1, \ldots, D+3$.

Many of the comments surrounding closeness of derivatives for standard A-splines hold in the exponential A-spline setting as well. We suspect that looking at a modified Sobolev-style norm, where subsequent derivatives are multiplied by additional factors of h, would be prudent in demonstrating these error bounds on derivatives.

Tension control

Perhaps the most useful feature of exponential splines over standard splines is the ability to tune the tension parameters ρ_j to our needs. In general, setting a high tension parameter over some panel will result in the spline essentially producing a degree D polynomial over said panel (i.e., essentially ignoring the two exponential terms in its form); setting a low tension parameter will cause the spline to more closely resemble a standard spline of degree

$(D+2)$; i.e., the two exponential terms are treated as if they collectively contribute two additional monomial terms of subsequent degree added to the already-existing polynomial part. A heuristic analysis of this phenomenon is hinted at by the following:

$$\begin{aligned}\exp\left(\rho_j \frac{t_{j-1}-t}{h_j}\right) &= 1 + \rho_j\left(\frac{t_{j-1}-t}{h_j}\right) + \frac{(\rho_j)^2}{2!}\left(\frac{t_{j-1}-t}{h_j}\right)^2 + \cdots + \frac{(\rho_j)^D}{D!}\left(\frac{t_{j-1}-t}{h_j}\right)^D \\ &\quad + \frac{(\rho_j)^{D+1}}{(D+1)!}\left(\frac{t_{j-1}-t}{h_j}\right)^{D+1} + \frac{(\rho_j)^{D+2}}{(D+2)!}\left(\frac{t_{j-1}-t}{h_j}\right)^{D+2} + \frac{(\rho_j)^{D+3}}{(D+3)!}\left(\frac{t_{j-1}-t}{h_j}\right)^{D+3} \\ &\quad + \cdots \end{aligned} \tag{257}$$

$$\begin{aligned}\exp\left(\rho_j \frac{t-t_j}{h_j}\right) &= 1 + \rho_j\left(\frac{t-t_j}{h_j}\right) + \frac{(\rho_j)^2}{2!}\left(\frac{t-t_j}{h_j}\right)^2 + \cdots + \frac{(\rho_j)^D}{D!}\left(\frac{t-t_j}{h_j}\right)^D \\ &\quad + \frac{(\rho_j)^{D+1}}{(D+1)!}\left(\frac{t-t_j}{h_j}\right)^{D+1} + \frac{(\rho_j)^{D+2}}{(D+2)!}\left(\frac{t-t_j}{h_j}\right)^{D+2} + \frac{(\rho_j)^{D+3}}{(D+3)!}\left(\frac{t-t_j}{h_j}\right)^{D+3} \\ &\quad + \cdots \end{aligned} \tag{258}$$

where $t_{j-1} \leq t \leq t_j$.

We first observe that $\left(\dfrac{t-t_j}{h_j}\right)$ and $\left(\dfrac{t_{j-1}-t}{h_j}\right)$ are both order $\mathcal{O}(1)$ terms, so the relative size of each of the terms in the power series expansions above will be determined by the factors $\dfrac{(\rho_j)^i}{i!}$ in each term.

When i is large, the terms vanish, since factorial growth beats geometric growth, as is known from the exponential power series converging on all of $\mathbb{R}$. We now specialize to looking at the first $(D+3)$ terms in the above expansions, since these are the terms that will have the largest impact on exponential splines.

When $\rho_j > 0$ is small, the contribution of the $\mathcal{O}((\rho_j)^{D+3})$ term is vanishingly small relative to that of the previous $\mathcal{O}((\rho_j)^{D+2})$ term. As such, in the exponential A-spline and C-spline procedures outlined above, the two exponential terms will essentially function jointly as the $\mathcal{O}(h^{D+1})$ and $\mathcal{O}(h^{D+2})$ terms of a standard spline of degree $(D+2)$. Specifically, in the C-spline procedure, the coefficients of the exponential terms are found so as to match the $(D+1)$st and $(D+2)$nd derivatives of f over the appropriate panels, and the polynomial part handles the lesser derivatives. In the A-spline procedure, the basis of exponential A-splines, the span of which is what f is projected onto, also resembles polynomials of degree $(D+2)$. So when ρ_j is small, the exponential spline with polynomial part degree D essentially becomes a standard spline with degree $(D+2)$.

When ρ_j is large, the exact opposite effect is observed: the contribution of the $\mathcal{O}((\rho_j)^{D+3})$ term is large relative to that of the previous $\mathcal{O}((\rho_j)^{D+2})$ term. In the exponential C-spline procedure, the correct Taylor polynomials are still matched every $(D+3)$ many panels, but in the panels between them, it is observed that the coefficients of the two exponential terms are small, leading to exponential splines that resemble their polynomial parts. For exponential A-splines, the large ρ_j values manifest in very small weighting of the two exponential terms, since the $\mathcal{O}((\rho_j)^{D+3})$ terms in the expansion of the exponentials are too large to have any decent approximation abilities to the function f; increasing the magnitude of the coefficients of the two exponential terms would result in non-optimal L^2 error arising from the large $\mathcal{O}((\rho_j)^{D+3})$ terms. So when ρ_j is large, the exponential spline with polynomial part D essentially becomes a standard spline with degree D.

Another intuitive way to see this behavior with respect to the tension parameter is from looking at the differential equation which is satisfied by the spline over any given panel:

$$\mathrm{D}^{D+1}\left(\mathrm{D}^2-\rho_j^2\right)y=0 \tag{259}$$

When ρ_j is large, y essentially needs to lie in the kernel of the D^{D+1} operator for the differential equation to be satisfied; when ρ_j is small, the differential equation looks more like $\mathrm{D}^{D+3}y = 0$. The solutions to these differential equations are polynomials of degree D and $D + 2$ respectively.

This means we can code an adaptive routine that assigns panel tensions according to how many derivatives f has over that panel. Too few, and we might need to "drop down" the degree of our spline by setting high tensions. Whereas if f has sufficiently many derivatives, then we might want to select low tension parameters in that region, so as to take advantage of superior approximating abilities of splines with higher degree polynomial parts.

Another reason we might want to reduce spline tension in a certain area, is to decrease the oscillations of our spline there. High-degree polynomials often exhibit this oscillatory behavior, and so by dropping two degrees, we limit the spline from oscillating too much there. This phenomenon is studied in e.g. ([15]), but only with respect to cubic exponential splines. There, the author uses exponential splines with linear polynomial part, and he finds that increasing the tensions cause the spline to look more and more like a linear function over the affected panels, and this manifests in a certain monotonicity theorem saying that increasing tensions causes the spline to exhibit monotonicity past a certain threshold tension. We don't quite get this in the setting of general degree polynomial parts, but the behavior that increasing tensions decreases oscillation is noted nonetheless.

Figures (51) through (59) demonstrate the effect the tension parameter has on a cubic exponential A-spline approximation of the sine function. What we can observe from these plots is when the tension is low, the error resembles that of linear A-splines (which converge of order 2), and when the tension is high, the error resembles that of cubic A-splines (which converge of order 4). This is as expected, because the cubic exponential spline has a linear polynomial part.

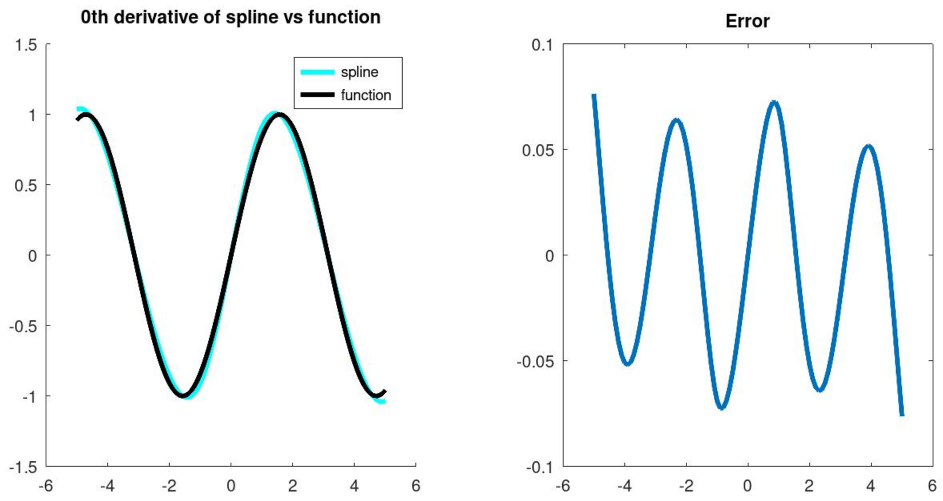


Figure 51: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 5 panels of uniform width. Average error = 4.5e-2, max error = 7.6e-2. Low tension parameter of $\rho = .1$

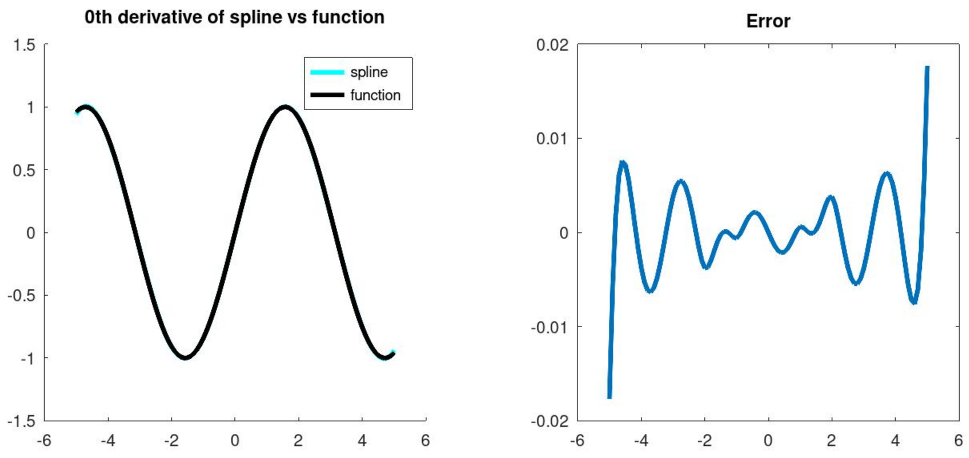


Figure 52: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 10 panels of uniform width. Average error = 3.8e-3, max error = 1.8e-2. Low tension parameter of $\rho = .1$

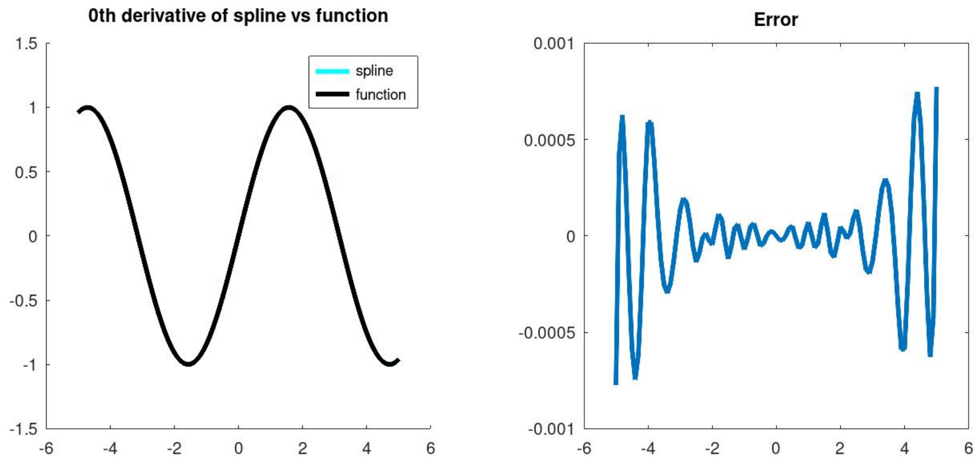


Figure 53: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 20 panels of uniform width. Average error = 2.7e-4, max error = 7.7e-4. Low tension parameter of $\rho = .1$

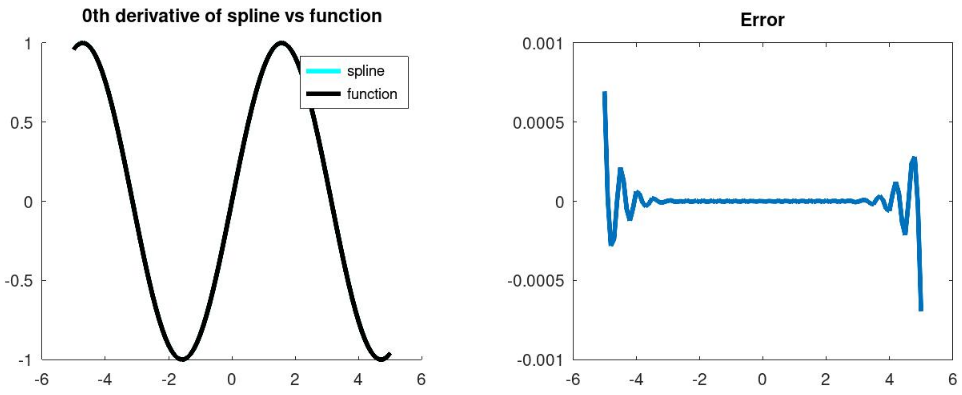


Figure 54: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$, 0th derivative of spline vs function, 40 panels of uniform width. Average error = 8.3e-5, max error = 6.9e-4. The quadrature error arising from comparatively low-order Trapezoidal method is causing the error to plateau. Low tension parameter of $\rho = .1$

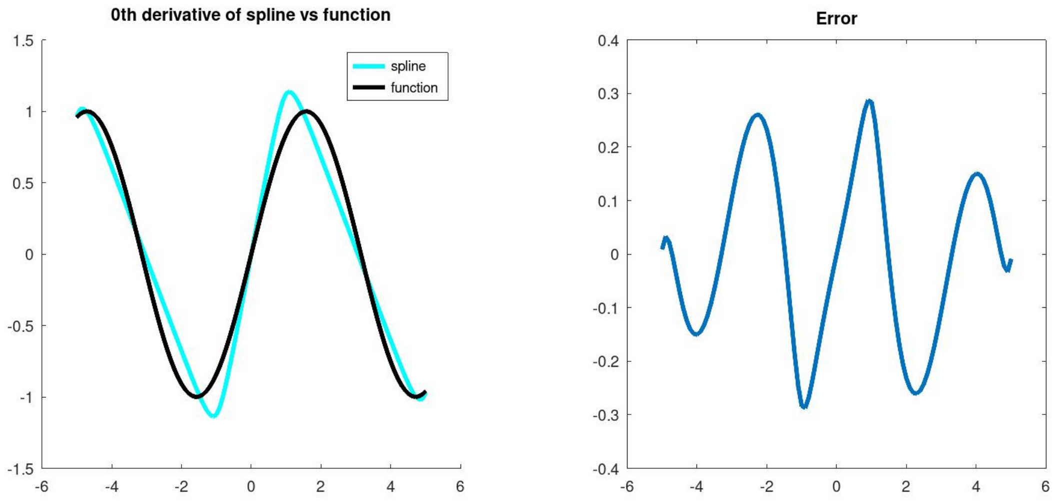


Figure 55: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$s, 0th derivative of spline vs function, 5 panels of uniform width. Average error = 1.6e-1, max error = 2.9e-1. High tension parameter of $\rho = 10$

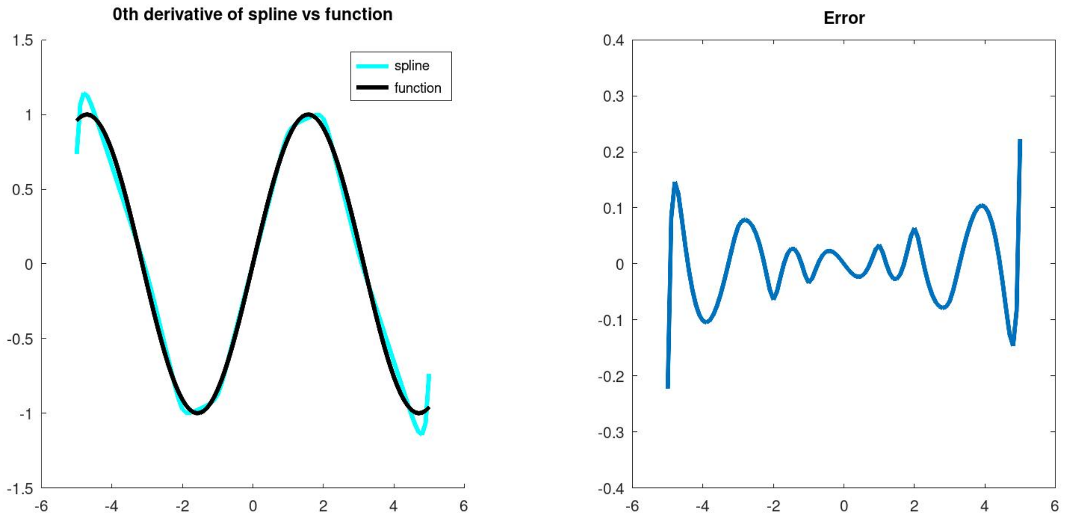


Figure 56: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$s, 0th derivative of spline vs function, 10 panels of uniform width. Average error = 6.0e-2, max error = 2.2e-1. High tension parameter of $\rho = 10$

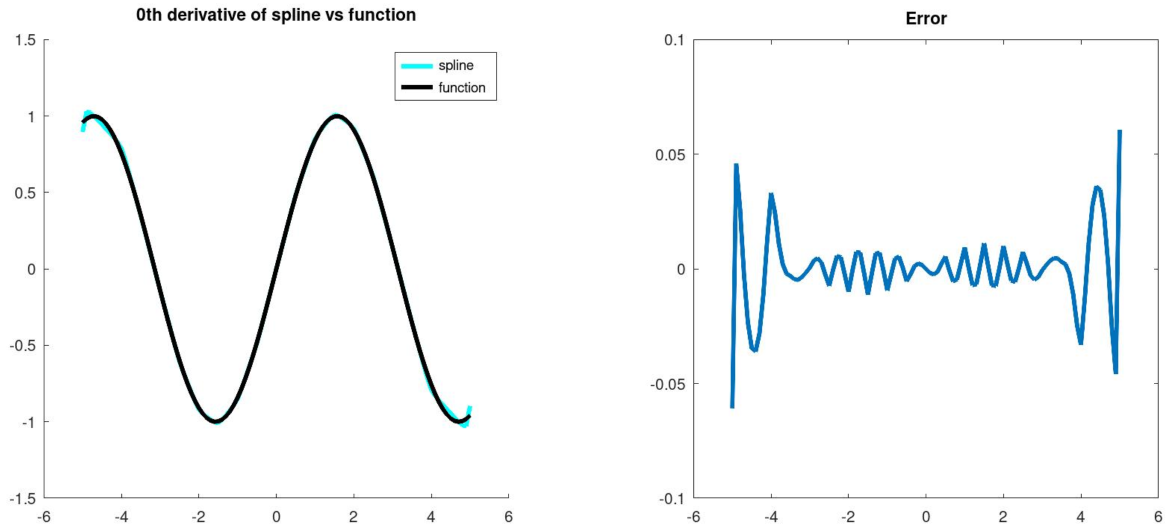


Figure 57: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$s, 0th derivative of spline vs function, 20 panels of uniform width. Average error = 1.5e-2, max error = 6.1e-2. High tension parameter of $\rho = 10$

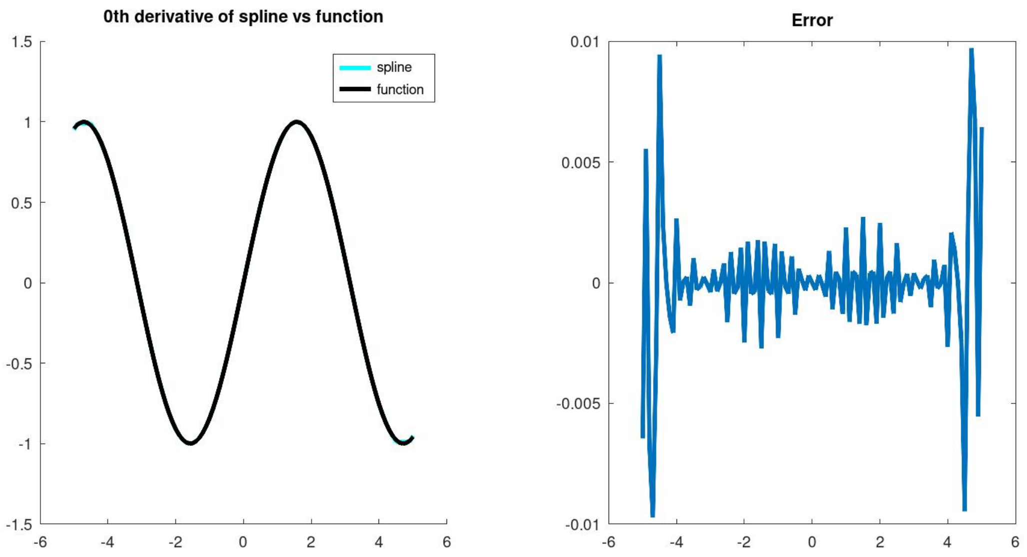


Figure 58: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$s, 0th derivative of spline vs function, 40 panels of uniform width. Average error = 2.7e-3, max error = 9.7e-3. High tension parameter of $\rho = 10$

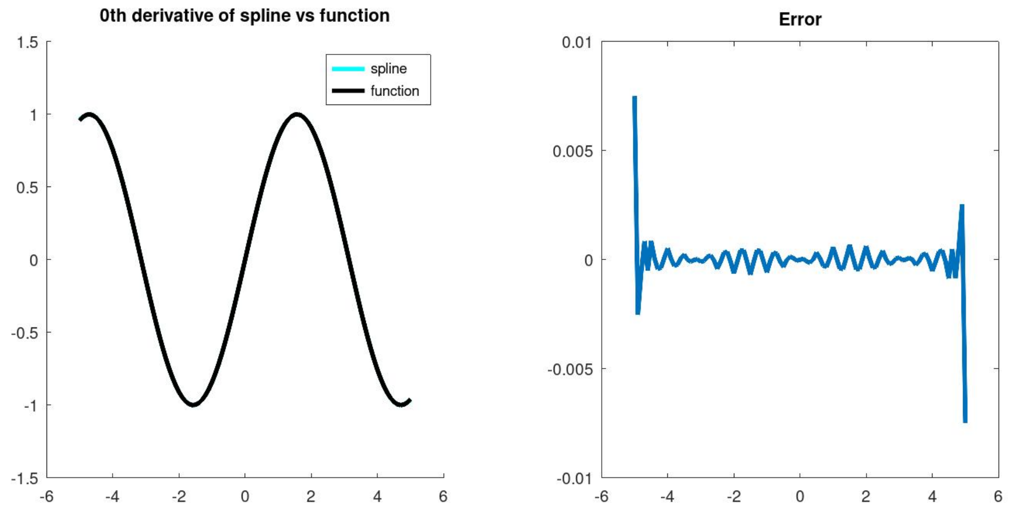


Figure 59: Cubic exponential A-spline (i.e. with polynomial part of degree 1) approximation of $f(t) = \sin(t)$s, 0th derivative of spline vs function, 80 panels of uniform width. Average error = 7.4e-4, max error = 7.5e-3. The quadrature error arising from comparatively low-order Trapezoidal method is beginning to cause the error to plateau. High tension parameter of $\rho = 10$

7 Tensor Product Approximations with A-splines

Suppose $f\colon \mathcal{D} \to \mathbb{R}$ is some multivariate function defined on a subset $\mathcal{D} \subset \mathbb{R}^2$ of the plane. If we suppose f is analytic, and we fix a point in $\mathcal{D}$, say, $(x_0, y_0) \in \mathcal{D}$, then there exists a disk of positive radius centered about (x_0, y_0) on which the following multivariate Taylor series representation of f is valid:

$$f(x,y) = \sum_{k=0}^{\infty} \sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \frac{(x-x_0)^j (y-y_0)^{k-j}}{k!} \left(\left. \frac{\partial^k f}{\partial^j x \partial^{k-j} y} \right|_{(x_0,y_0)} \right) \tag{260}$$

If our disk $\mathcal{D}$ has finite positive radius, then for any $\epsilon > 0$ there exists an $N \in \mathbb{N}$ such that for all $n > N$ we have

$$\left| f(x,y) - \sum_{k=0}^{n} \sum_{j=0}^{k} \begin{pmatrix} k \\ j \end{pmatrix} \frac{(x-x_0)^j (y-y_0)^{k-j}}{k!} \left(\left. \frac{\partial^k f}{\partial^j x \partial^{k-j} y} \right|_{(x_0,y_0)} \right) \right| < \epsilon \tag{261}$$

This last fact is a consequence of the uniform convergence of the partial sums of a power series to the power series itself on a compact set contained in its disk of convergence.

The inequality above tells us that f may be approximated to within ϵ by a finite sum of products of single-variable functions of x and y. So,

$$f(x,y) \approx \sum_{l=0}^{m} g_l(x) h_l(y) \tag{262}$$

for aptly chosen m, g_l, and h_l.

It is in this sense that we attempt to find a tensor product approximation to $f(\cdot,\cdot)$ using splines.

In particular, one may consider an interpolatory tensor product of the form

$$f(x,y) \approx \sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) S_i^x(x) S_j^y(y) \tag{263}$$

where (x_i, y_j), $i = 0, 1, \ldots, M_x$, $j = 0, 1, \ldots, M_y$ are grid points on a rectangular grid, $S_i^x(\cdot)$ is an interpolatory spline of degree $D_x \in \mathbb{N}$ such that

$$S_i^x(x_k) = \delta_{i,k} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} \tag{264}$$

and $S_j^y(\cdot)$ is an interpolatory spline of degree $D_y \in \mathbb{N}$ such that

$$S_j^y(x_k) = \delta_{j,k} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases} \tag{265}$$

Given the splines $\{S_i^x\}$ and $\{S_j^y\}$ this approximation is easy to construct, but it is very computationally expensive to evaluate, since at each of the $(M_x + 1)(M_y + 1)$ grid points (x_i, y_j) we must evaluate a product of two polynomials. The amount of work needed to evaluate the approximation at a single point is $\mathcal{O}(M_x M_y (D_x + D_y))$, which can be prohibitively expensive over fine grids.

A fix to the exorbitantly high cost of evaluation is to collect like terms, thereby obtaining coefficients $c_{p,q}^{k,l} \in \mathbb{R}$ such that

$$\sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) S_i^x(x) S_j^y(y) = \sum_{k=0}^{D_x} \sum_{l=0}^{D_y} c_{p,q}^{k,l} \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l \tag{266}$$

valid on the square $[x_{p-1}, x_p] \times [y_{q-1}, y_q]$. This reduces the cost of evaluation to just $\mathcal{O}(D_x D_y)$, and since

$$D_x D_y << M_x M_y (D_x + D_y) \tag{267}$$

this reduces the computational complexity of evaluation.

More specifically, if

$$S_i^x(x) = \sum_{k=0}^{D_x} a_{i,p}^k \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k \tag{268}$$

for $x \in [x_{p-1}, x_p]$, and

$$S_j^y(x) = \sum_{l=0}^{D_y} b_{j,q}^l \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l \tag{269}$$

for $y \in [y_{q-1}, y_q]$, then for $(x, y) \in [x_{p-1}, x_p] \times [y_{q-1}, y_q]$ we have

$$\begin{aligned}
\sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) S_i^x(x) S_j^y(y) &= \sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) \left[\sum_{k=0}^{D_x} a_{i,p}^k \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k \right] \left[\sum_{l=0}^{D_y} b_{j,q}^l \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l \right] && (270) \\
&= \sum_{k=0}^{D_x} \sum_{l=0}^{D_y} \sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) a_{i,p}^k \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k b_{j,q}^l \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l && (271) \\
&= \sum_{k=0}^{D_x} \sum_{l=0}^{D_y} \underbrace{\left(\sum_{i=0}^{M_x} \sum_{j=0}^{M_y} f(x_i, y_i) a_{i,p}^k b_{j,q}^l \right)}_{c_{p,q}^{k,l}} \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l && (272) \\
&= \sum_{k=0}^{D_x} \sum_{l=0}^{D_y} c_{p,q}^{k,l} \left(\frac{x - x_{p-1}}{x_p - x_{p-1}} \right)^k \left(\frac{y - y_{q-1}}{y_q - y_{q-1}} \right)^l && (273)
\end{aligned}$$

as desired.

However, one still needs to store $\mathcal{O}(M_x M_y D_x D_y)$ coefficients $c_{p,q}^{k,l}$, which can be a very large number of coefficients.

Given the explosive growth of coefficients when using the type of approximation (266), this suggests that one is likely using too many coefficients, i.e. for smooth functions, many of the coefficients are very small or vanish. Such an inference suggests another type of tensor product spline approximation where one incrementally adds terms until a desired accuracy is achieved.

In order to implement such a procedure, one must develop a technique for determining the coefficients of a tensor product spline with a set number of terms. One method for doing this is as follows.

If we have an orthonormal (w.r.t. L^2) basis of splines $\{\phi_i(x)\}_{i=1}^{p}$ for $\mathcal{S}_{D_x}$ with respect to the knots $x_0 < \cdots < x_{M_x}$ and an orthonormal basis of splines $\{\gamma_j(y)\}_{j=1}^{q}$ for $\mathcal{S}_{D_y}$ with respect to the knots $y_0 < \cdots < y_{M_x}$ then $\{\phi_i(x)\gamma_j(y)\}_{i,j}$ form an orthonormal set of multivariate functions on $[x_0, x_{M_x}] \times [y_0, y_{M_y}]$, since

$$\int_{x_0}^{x_{M_x}} \int_{y_0}^{y_{M_y}} \phi_{i_1}(x)\gamma_{j_1}(y) \cdot \phi_{i_2}(x)\gamma_{j_2}(y) dy dx = \left(\int_{x_0}^{x_{M_x}} \phi_{i_1}(x)\phi_{i_2}(x) dx \right) \left(\int_{y_0}^{y_{M_y}} \gamma_{j_1}(y)\gamma_{j_2}(y) dy \right) = \delta_{i_1,i_2} \cdot \delta_{j_1,j_2} \tag{274}$$

and we can form the approximation

$$f(x,y) \approx \sum_{i=1}^{p} \sum_{j=1}^{q} \alpha_{i,j} \phi_i(x)\gamma_j(y) \tag{275}$$

where, by taking inner products and using orthonormality of the basis functions, we can arrive at the formula for the coefficients

$$\alpha_{i,j} =< f, \phi_i \gamma_j >= \int_{x_0}^{x_{M_x}} \int_{y_0}^{y_{M_y}} f(x,y)\phi_i(x)\gamma_j(y) dx dy \tag{276}$$

A small issue arises when computing the continuous inner product though, since in general we will be approximating the continuous inner product $< \cdot, \cdot >$ with a discrete one $< \cdot, \cdot >'$: the basis functions $\phi_i(x)\gamma_j(y)$ will no longer necessarily be orthonormal with respect to the discrete inner product. This can be partially remedied in one of two ways: (1) by setting up a linear system for the coefficients $\alpha_{i,j}$ after taking discrete inner products:

$$\begin{bmatrix} <\phi_1\gamma_1,\phi_1\gamma_1>' & \cdots & <\phi_1\gamma_1,\phi_1\gamma_q>' & \cdots & <\phi_1\gamma_1,\phi_p\gamma_1>' & \cdots & <\phi_1\gamma_1,\phi_p\gamma_q>' \\ \vdots & & & & & & \vdots \\ <\phi_1\gamma_q,\phi_1\gamma_1>' & \cdots & <\phi_1\gamma_q,\phi_1\gamma_q>' & \cdots & <\phi_1\gamma_q,\phi_p\gamma_1>' & \cdots & <\phi_1\gamma_q,\phi_p\gamma_q>' \\ \vdots & & & & & & \vdots \\ <\phi_p\gamma_1,\phi_1\gamma_1>' & \cdots & <\phi_p\gamma_1,\phi_1\gamma_q>' & \cdots & <\phi_p\gamma_1,\phi_p\gamma_1>' & \cdots & <\phi_p\gamma_1,\phi_p\gamma_q>' \\ \vdots & & & & & & \vdots \\ <\phi_p\gamma_q,\phi_1\gamma_1>' & \cdots & <\phi_p\gamma_q,\phi_1\gamma_q>' & \cdots & <\phi_p\gamma_q,\phi_p\gamma_1>' & \cdots & <\phi_p\gamma_q,\phi_p\gamma_q>' \end{bmatrix} \begin{bmatrix} \alpha_{1,1} \\ \vdots \\ \alpha_{1,q} \\ \vdots \\ \alpha_{p,1} \\ \vdots \\ \alpha_{p,q} \end{bmatrix} = \begin{bmatrix} <f,\phi_1,\gamma_1>' \\ \vdots \\ <f,\phi_1\gamma_q>' \\ \vdots \\ <f,\phi_p,\gamma_1>' \\ \vdots \\ <f,\phi_p\gamma_q>' \end{bmatrix} \tag{277}$$

or (2) by selecting N data points $(x'_k, y'_k)_{k=1}^N$ from amongst the rectangle $[x_0, x_{M_x}] \times [y_0, y_{M_y}]$ and setting up a least-squares problem

$$\begin{bmatrix} \phi_1(x'_1)\gamma_1(y'_1) & \cdots & \phi_1(x'_1)\gamma_p(y'_1) & \cdots & \phi_q(x'_1)\gamma_1(y'_1) & \cdots & \phi_q(x'_1)\gamma_p(y'_1) \\ \phi_1(x'_2)\gamma_1(y'_2) & \cdots & \phi_1(x'_2)\gamma_p(y'_2) & \cdots & \phi_q(x'_2)\gamma_1(y'_2) & \cdots & \phi_q(x'_2)\gamma_p(y'_2) \\ \vdots & & & & & & \vdots \\ \phi_1(x'_N)\gamma_1(y'_N) & \cdots & \phi_1(x'_N)\gamma_p(y'_N) & \cdots & \phi_q(x'_N)\gamma_1(y'_N) & \cdots & \phi_q(x'_N)\gamma_p(y'_N) \end{bmatrix} \begin{bmatrix} \alpha_{1,1} \\ \vdots \\ \alpha_{1,q} \\ \vdots \\ \alpha_{p,1} \\ \vdots \\ \alpha_{p,q} \end{bmatrix} = \begin{bmatrix} f(x'_1, y'_1) \\ f(x'_2, y'_2) \\ \vdots \\ f(x'_N, y'_N) \end{bmatrix} \tag{278}$$

where $N >> pq$. Using method (1) can be expected to minimize the discrete L^2 norm of the error between f and our approximation, and using method (2) can be expected to minimize the discrete L^∞ norm of the error over the N selected data points.

Similarly to (266), one we may collect like terms to achieve an approximation requiring only $\mathcal{O}(D_x D_y)$ evaluations, though we do still need to store $M_x M_y(D_x+1)(D_y+1)$ many coefficients, even after this simplification.

To avoid excessive growth of coefficients of the approximation, one considers an adaptive construction. The adaptive procedure is based upon an approximation

$$f(x,y) \approx \sum_r u_r(x) v_r(y) \tag{279}$$

where u_r and v_r are splines in x and y respectively, and additional products are added as we go (i.e. the upper bound on the sum increments by 1 with each iteration of the procedure). Here is how we do this:

(1) Start with a spline v_1 in the y variable.

(2) Given $v_1, \ldots, v_r$, take inner products w.r.t. the y variable of both sides of (279) with the v_i's, and set up a linear system for $u = u(x)$ according to

$$\begin{bmatrix} < v_1, v_1 > & \cdots & < v_1, v_r > \\ \vdots & & \vdots \\ < v_r, v_1 > & \cdots & < v_r, v_r > \end{bmatrix} \begin{bmatrix} u_1(x) \\ \vdots \\ u_r(x) \end{bmatrix} = \begin{bmatrix} < v_1, f > (x) \\ \vdots \\ < v_r, f > (x) \end{bmatrix} \tag{280}$$

and fit $u_i(x)$ according to this system, for various x (possibly the knots in the x direction).

(3) With the u_i's just found, we now take inner products w.r.t. the x variable of both

sides of (279) with the u_i's, and set up a linear system for $v = v(y)$ according to

$$\begin{bmatrix} < u_1, u_1 > & \cdots & < u_1, u_r > \\ \vdots & & \vdots \\ < u_r, u_1 > & \cdots & < u_r, u_r > \end{bmatrix} \begin{bmatrix} v_1(y) \\ \vdots \\ v_r(y) \end{bmatrix} = \begin{bmatrix} < u_1, f > (y) \\ \vdots \\ < u_r, f > (y) \end{bmatrix} \tag{281}$$

and fit $v_i(y)$ according to this system, for various y (possibly the knots in the y direction).

(4) cycle through steps (2) and (3) until the difference between successive u_i's and v_i's is small. If the tensor product approximation overall is poor (which would be reflected as a large discrete L^2 norm of the error, or large discrete L^∞ norm of the error) and the systems aren't close to singular, increment r (the number of terms) by 1, generate a spline $v_r(y)$ which is linearly independent to the previous v_i's, and return to step (2).

The nice thing about this procedure is it runs very fast compared to the previous attempts at solving the problem, since we are only using a couple well-picked splines to capture the dominant behavior of f. Another feature of this procedure is it is fully generalizable to high dimensions, and still only requires being able to fit splines in 1D. We can incorporate any of the 1D splines developed earlier into this procedure.

There are sensible alterations we can make to this procedure, such as fixing past products $u_i(x)v_i(y)$ and iterating only on the residual, or imposing orthogonality conditions on the new splines being generated in step (4), but these are variations on a theme. Below, we demonstrate the efficacy of this procedure with numerical examples inspired by the Genz functions in ([2]), using both standard and exponential A-splines as the 1D approximants.

There are error bounds for such types of tensor approximation procedures (e.g. [10]), though this does not yet exist for the procedures outlined above.

Figures (60) and (61) demonstrate interpolatory approximations of Kronecker deltas. The tensor product of splines appears as a sharp bump where the Kronecker delta takes on the value 1. We demonstrate this for an interior point and a corner point. Due to the astronomical computational complexity of these algorithms, the panel count is severely limited.

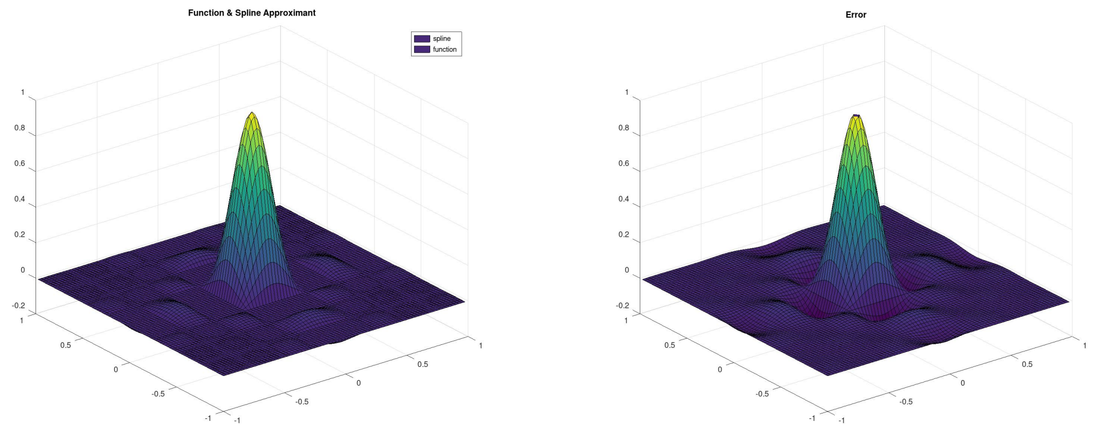


Figure 60: Cubic A-spline tensor product approximation of $f(x, y) = \delta_{(0,0)}(x, y)$, effectively producing the basis spline which acts as $\delta_{(0,0)}$. 8 panels of uniform width along each axis. Interpolatory spline approximation.

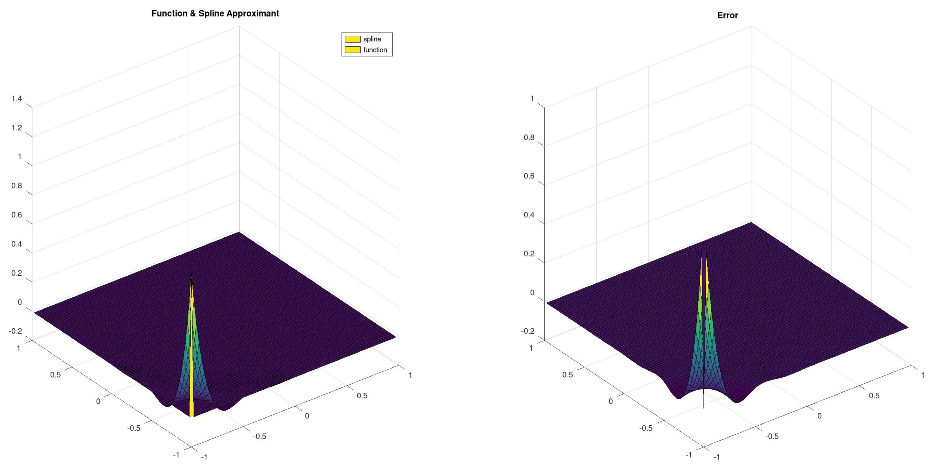


Figure 61: Cubic exponential A-spline tensor product approximation of $f(x, y) = \delta_{(-1,-1)}(x, y)$, effectively producing a corner basis spline that acts as $\delta_{(-1,-1)}$. 8 panels of uniform width along each axis. Interpolatory spline approximation. Uniform tension of $\rho = 1$.

Figures (62) and (63) showcase how well a multivariate function can be approximated using the interpolatory spline approximation. The multivariate function was chosen so that it cannot be expressed as a finite sum of products of single-variable functions in x and y. As in the previous examples, the panel count is severely limited due to high computational complexity of the algorithm.

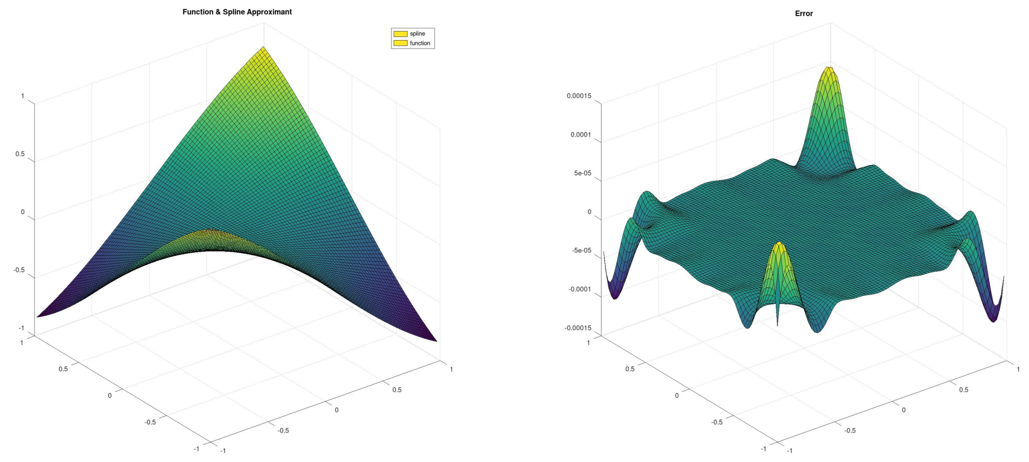


Figure 62: Cubic A-spline tensor product approximation of $f(x, y) = \sin(xy)$, 8 panels of uniform width along each axis. Interpolatory spline approximation. max error = 1.0e-4, relative error = 1.2e-4

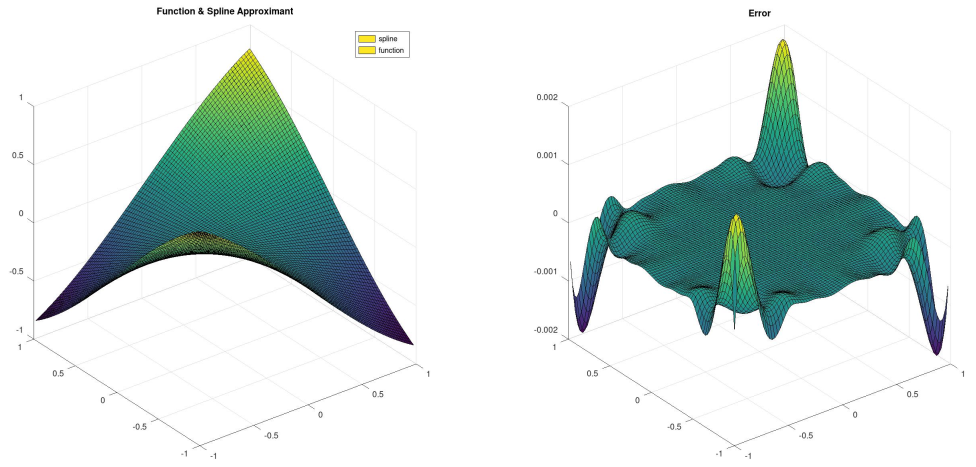


Figure 63: Cubic exponential A-spline tensor product approximation of $f(x, y) = \sin(xy)$, 8 panels of uniform width along each axis. Interpolatory spline approximation. Uniform tension of $\rho = 1$. max error = 1.9e-3, relative error = 2.2e-3

Figures (64) through (67) demonstrate the same multivariate function being approximated using the adaptive tensor spline approximation. Only 3 terms were used before the program terminated due to the residual error being sufficiently small. Because few terms are needed, the algorithm is much more computationally efficient, and larger panel counts can be used. We observe a general trend that high panel counts in the x and y directions correspond to smaller errors for the same number of terms, but the resulting error in the approximation depends mostly on the tolerance we set for the residual error stopping condition.

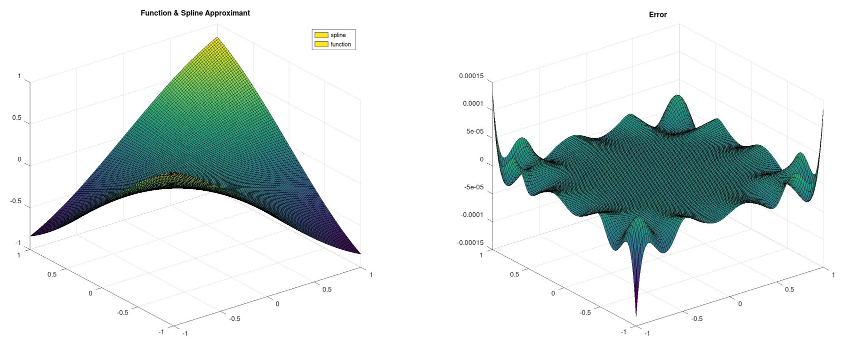


Figure 64: Cubic A-spline tensor product approximation of $f(x, y) = \sin(xy)$, 8 panels of uniform width along each axis. Adaptive tensor spline approximation, 3 terms used. Gaussian quadrature is used for projection formulas. max error = 1.3e-4, relative error = 1.6e-4

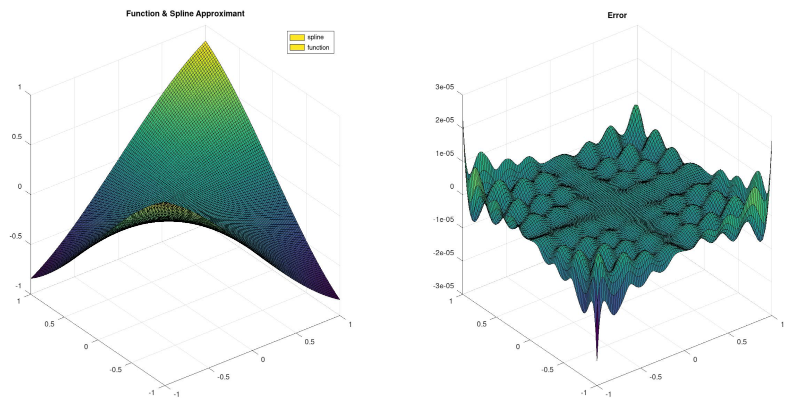


Figure 65: Cubic A-spline tensor product approximation of $f(x, y) = \sin(xy)$, 16 panels of uniform width along each axis. Adaptive tensor spline approximation, 3 terms used. max error = 2.3e-5, relative error = 2.7e-5

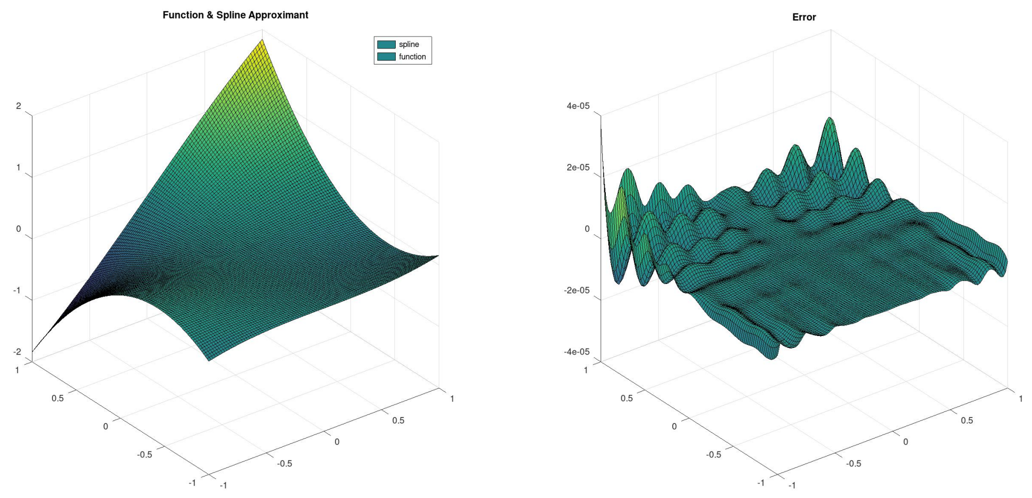


Figure 66: Cubic A-spline tensor product approximation of $f(x, y) = x\sin(y) + yx^2$, 16 panels of uniform width along each axis. Adaptive tensor spline approximation, 3 terms used. max error = 3.7e-5, relative error = 2.0e-5

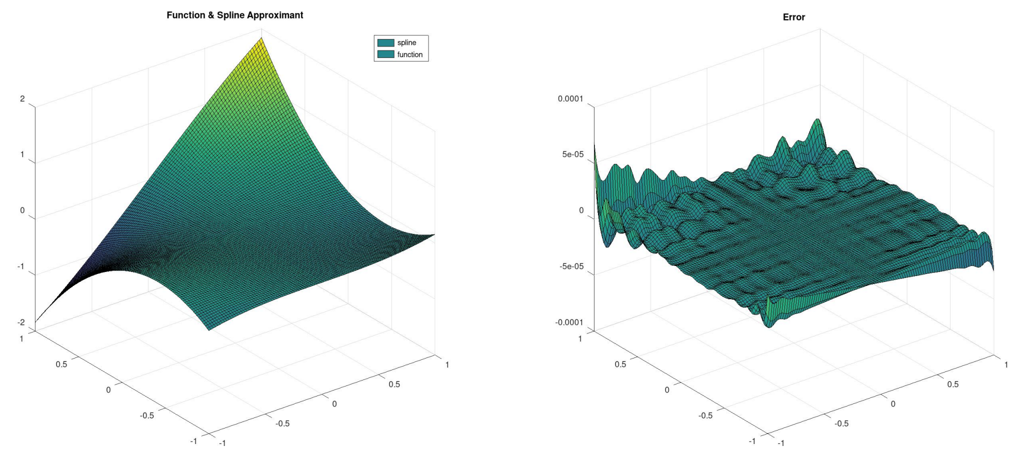


Figure 67: Cubic A-spline tensor product approximation of $f(x, y) = x\sin(y) + yx^2$, 32 panels of uniform width along each axis. Adaptive tensor spline approximation, 3 terms used. max error = 6.9e-5, relative error = 3.7e-5

The last two figures, (68) and (69), show approximations of step functions by 5th and 6th degree tensor spline approximations. The L^∞ error cannot fall below 0.5 due to the discontinuous nature of the step functions. We observe that the higher-order splines lead to more oscillatory approximations of the discontinuous function, as one would expect in the single-variable case as well.

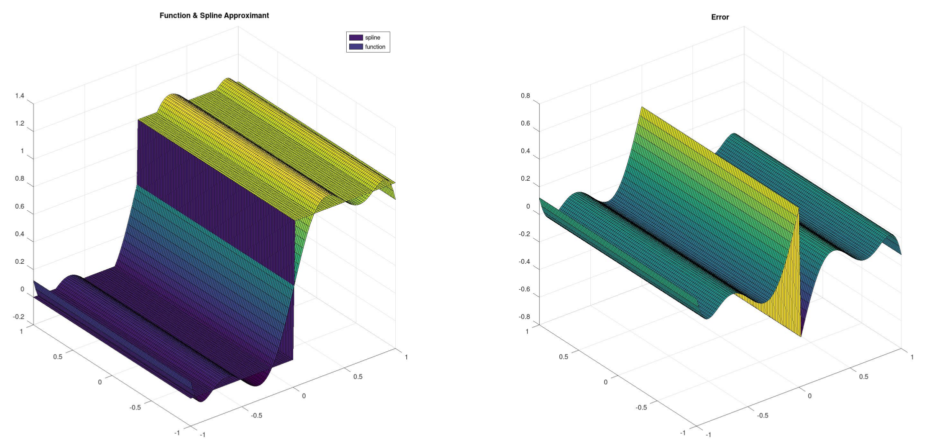


Figure 68: Quintic A-spline tensor product approximation of a step function in y, 8 panels of uniform width along each axis. Adaptive tensor spline approximation, 2 terms used. max error = 5.0e-1, relative error = 5.0e-1

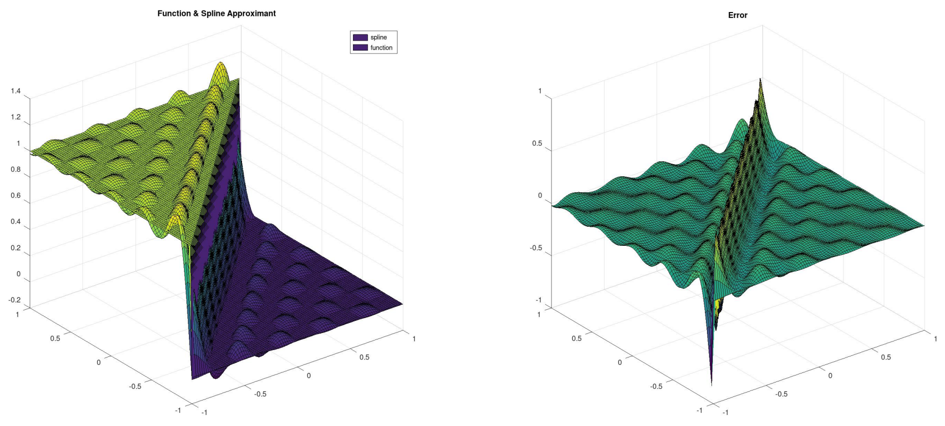


Figure 69: 6th degree A-spline tensor product approximation of a diagonal step function, 16 panels of uniform width along each axis. Adaptive tensor spline approximation, 10 terms used. max error = 8.4e-1, relative error = 8.4e-1

8 Future work

8.1 Closeness of Derivatives

As mentioned in the convergence analysis section above, it would be nice to have theorems pertaining to the closeness of derivatives of the various spline procedures mentioned herein. These were left as conjectures where we discussed them above.

8.2 Rational Splines

Rational splines are an interesting tool to use for situations where we need to approximate functions with either the decay or singularities associated with rational functions. In particular, it would be nice to develop a procedure for using rational splines as approximators, that is amenable to finding nice error bounds for. Rational splines can be put in the A-spline framework.

8.3 Splines on Lines

Linear ODEs are not difficult to factor into the constraint matrix as additional constraints that must be satisfied at each node, or at certain subsets of nodes. Nonlinear ODEs would require more work to solve using the A-spline framework, as we can't as neatly package them as linear constraints; but the basis of A-splines generated by the A-spline procedure can still be used to form a span that the true solution is projected onto. It is then feasible for the method of lines of solving PDEs to be used in conjunction with A-splines, the solution to the PDE along each "line" being fitted to a certain spline. Such a procedure would be an interesting application of using A-splines to solve PDEs.

8.4 Derivative Approximator

Developing a general nth-order accurate, kth-derivative approximating routine given a set of function values and some subset of various of its derivatives on a finite subset of the real line, would be helpful in assessing where to increase tension parameters in the context of modeling with exponential splines. For reference, some work has already tried to answer this question of optimal tension parameters (e.g. [18]), but this doesn't rely on derivative approximators as described here. It would also help in determining adaptive stepsize routines for standard splines in addition. One could use forward/backward difference formulas, and divided differences in general, but this doesn't always lead to optimal usage of the data given (e.g. osculatory information of f at certain nodes), and can introduce complications when the panel size is not uniform (e.g. some divided difference formulas no longer approximate the derivative at one particular knot, but rather at some nearby point). This wouldn't be too difficult, but would require some careful choices in the procedure, and merits a separate paper in the future.

9 Conclusion

In this thesis we provided explicit algorithms for constructing multiple types of high degree 1D splines (standard and exponential A-splines, standard and exponential C-splines) as well as algorithms for generating orthogonal bases for families of splines $\mathcal{S}_D$ and $\mathcal{E}_D$ (via the standard and exponential A-spline procedures). With these orthogonal bases, one is able to perform L2 approximation of functions by way of projection, and we gave relevant error bounds in L^2 and L^∞ for approximations constructed in this way for each class of splines that we've considered. We also considered the use of a tensor product of splines to approximate multi-dimensional functions. This application demonstrates the utility of creating 1D splines approximation using L2 projection. In particular, the use of 1D L2 projection facilitates the construction of multi-dimensional tensor product approximations that have few terms, thus providing a way of generating multi-dimensional approximations that are both accurate and computationally efficient to evaluate.

It is the hope of the author that the framework we established for A-splines and C-splines herein takes afoot in the spline community, to aid in L2 approximation techniques, and also to aid in providing inspiration for related spline families waiting to be discovered.

References

[1] Christopher Anderson, Peter Cheng, and Jean-Michel Maldague. Alternative spline basis construction and application. 2021, in preparation.

[2] Daniele Bigoni, Allan P. Engsig-Karup, and Youssef M. Marzouk. Spectral tensor-train decomposition. *SIAM Journal on Scientific Computing*, 38(4):A2405–A2439, Jan 2016.

[3] Garrett Birkhoff and Carl de Boor. Error bounds for spline interpolation. *Journal of Mathematics and Mechanics*, 13(5):827–835, 1964.

[4] J. H. Bramble and A. H. Schatz. Estimates for spline projections. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 10(R2):5–37, 1976.

[5] Ole Christensen and Peter Massopust. Exponential b-splines and the partition of unity property. *Advances in Computational Mathematics*, 37:301–318, 09 2012.

[6] Carl de Boor. On calculating with b-splines. *Journal of Approximation Theory*, 6(1):50–62, 1972.

[7] Carl de Boor. *A Practical Guide to Splines*, volume 27. Springer-Verlag New York, 1978.

[8] Carl de Boor. Convergence of cubic spline interpolation with the not-a-knot condition. 01 1985.

[9] James F. Epperson. On the runge example. *The American Mathematical Monthly*, 94(4):329–341, 1987.

[10] Michael Griebel and Jens Oettershagen. On tensor product approximation of analytic functions. *Journal of Approximation Theory*, 207:348–379, 2016.

[11] Zhong Guan. Iterated bernstein polynomial approximations, 2009.

[12] Charles A Hall and W.Weston Meyer. Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2):105–122, 1976.

[13] T. Lyche and L. Schumaker. Local spline approximation methods. *Journal of Approximation Theory*, 15:294–325, 1975.

[14] Peter Massopust. Exponential splines of complex order. 11 2013.

[15] Brian J McCartin. Theory of exponential splines. *Journal of Approximation Theory*, 66(1):1–23, 1991.

[16] J Nitsche and A Schatz. On local approximation properties of l2-projection on spline-subspaces†. *Applicable Analysis*, 2(2):161–168, 1972.

[17] Paul D. Patent. The effect of quadrature errors in the computation of l^2 piecewise polynomial approximations. *SIAM Journal on Numerical Analysis*, 13(3):344–361, 1976.

[18] Karl O. Riedel. Locally optimal knots and tension parameters for exponential splines. *Journal of Computational and Applied Mathematics*, 196(1):94–114, 2006.

[19] Espen Sande, Carla Manni, and Hendrik Speleers. Sharp error estimates for spline approximation: Explicit constants, n-widths, and eigenfunction convergence. *Mathematical Models and Methods in Applied Sciences*, 29(06):1175–1205, Jun 2019.

[20] Espen Sande, Carla Manni, and Hendrik Speleers. Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis. *Numerische Mathematik*, 144(4):889–929, Jan 2020.

[21] I. J. Schoenberg. Spline interpolation and best quadrature formulae. *Bulletin of the American Mathematical Society*, 70(1):143 – 148, 1964.

[22] Hubert Schwetlick and Torsten Schütze. Least squares approximation by splines with free knots, 1995.

[23] Stefan Takacs and Thomas Takacs. Approximation error estimates and inverse inequalities for b-splines of maximum smoothness. *Mathematical Models and Methods in Applied Sciences*, 26(07):1411–1445, May 2016.

[24] Michael Unser and Thierry Blu. Cardinal exponential splines: Part i—theory and filtering algorithms. *IEEE Trans. Signal Process*, 53:1425–1438, 2005.