Divided Differences, Falling Factorials, and Discrete Splines

Another Look at Trend Filtering and Related Problems

Other titles in Foundations and Trends® in Machine Learning

Risk-Sensitive Reinforcement Learning via Policy Gradient Search
Prashanth L. A. and Michael C. Fu
ISBN: 978-1-63828-026-2

A Unifying Tutorial on Approximate Message Passing
Oliver Y. Feng, Ramji Venkataramanan, Cynthia Rush and Richard J. Samworth
ISBN: 978-1-63828-004-0

Learning in Repeated Auctions
Thomas Nedelec, Clément Calauzènes, Noureddine El Karoui and Vianney Perchet
ISBN: 978-1-68083-938-8

Dynamical Variational Autoencoders: A Comprehensive Review
Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber and Xavier Alameda-Pineda
ISBN: 978-1-68083-912-8

Machine Learning for Automated Theorem Proving: Learning to Solve SAT and QSATe
Sean B. Holden
ISBN: 978-1-68083-898-5

Spectral Methods for Data Science: A Statistical Perspective
Yuxin Chen, Yuejie Chi, Jianqing Fan and Cong Ma
ISBN: 978-1-68083-896-1

Divided Differences, Falling Factorials, and Discrete Splines

Another Look at Trend Filtering and Related Problems

Ryan J. Tibshirani
Carnegie Mellon University
ryantibs@cmu.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

R. J. Tibshirani. *Divided Differences, Falling Factorials, and Discrete Splines.* Foundations and Trends® in Machine Learning, vol. 15, no. 6, pp. 694–846, 2022.

ISBN: 978-1-63828-037-8

Foundations and Trends® in Machine Learning

Volume 15, Issue 6, 2022

Editorial Scope

Topics

Foundations and Trends® in Machine Learning publishes survey and tutorial articles in the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians

Foundations and Trends® in Machine Learning, 2022, Volume 15, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Contents

1 Introduction 2
1.1 Motivation 5
1.2 Summary 10
1.3 Notation 13

2 Background 14
2.1 Divided Differences 14
2.2 Splines 17
2.3 Discrete Splines 18
2.4 Smoothing Splines 20
2.5 Locally Adaptive Splines 23
2.6 Trend Filtering 28
2.7 Bohlmann-Whittaker Filtering 31

3 Falling Factorials 34
3.1 Discrete Differentiation 34
3.2 Discrete Integration 37
3.3 Constructing the Basis 40

4 Smoothness Properties 44
4.1 Discrete Splines 45
4.2 Matching Derivatives 47

5 Dual Basis **52**
5.1 Discrete Differentiation of One "Extra" Order 52
5.2 Constructing the Dual Basis 53
5.3 Falling Factorial Interpolation 54
5.4 Implicit Form Interpolation 55

6 Matrix Computations **59**
6.1 Discrete Differentiation 59
6.2 Extended Discrete Differentiation 61
6.3 Falling Factorial Basis 62
6.4 Fast Matrix Multiplication 62

7 Discrete B-Splines **64**
7.1 Construction at the Design Points 65
7.2 Interpolation to $[a, b]$ 67
7.3 Discrete Natural Splines 71

8 Sparse Knot Sets **74**
8.1 Matrix Computations 75
8.2 Discrete B-Splines 77
8.3 Evaluation at the Design Points 79
8.4 Least Squares Problems 80

9 Representation **83**
9.1 Total Variation Functionals 83
9.2 L_2-Sobolev Functionals 85

10 Approximation **90**
10.1 Proximity of Truncated Power and Falling Factorial Bases . 91
10.2 Approximation of Bounded Variation Functions 92

11 Trend Filtering **95**
11.1 Computational Improvements 98
11.2 Natural Trend Filtering 101

12 BW Filtering **103**
12.1 Weighted BW Filtering 103
12.2 Bounds on the Distance Between Solutions 105
12.3 Connections to Discrete Splines 109

13 Discussion **112**

Acknowledgements **113**

Appendices **114**

A Notation Table **115**

B Proofs **119**
B.1 Proof of Theorem 2.1 119
B.2 Proof of the Linear Combination Formulation (3.10) 120
B.3 Proof of Lemma 3.1 121
B.4 Proof of Lemma 3.3 122
B.5 Lemma B.2 (Helper Result for Corollary 5.6) 124
B.6 Proof of Theorem 9.2 125
B.7 Proof of Lemma 9.4 132
B.8 Proof of Lemma 10.2 134
B.9 Proof of Lemma 12.1 135
B.10 Proof of Theorem 12.2 136
B.11 Proof of Lemma 12.4 137

C B-Splines and Discrete B-Splines **139**
C.1 B-Splines 139
C.2 Discrete B-splines 143

D Fast Matrix Multiplication **148**

References **150**

Divided Differences, Falling Factorials, and Discrete Splines

Ryan J. Tibshirani

Carnegie Mellon University, USA; ryantibs@cmu.edu

ABSTRACT

This monograph reviews a class of univariate piecewise polynomial functions known as *discrete splines*, which share properties analogous to the better-known class of spline functions, but where continuity in derivatives is replaced by (a suitable notion of) continuity in *divided differences*. As it happens, discrete splines bear connections to a wide array of developments in applied mathematics and statistics, from divided differences and Newton interpolation (dating back to over 300 years ago) to trend filtering (from the last 15 years). We survey these connections, and contribute some new perspectives and new results along the way.

Ryan J. Tibshirani (2022), "Divided Differences, Falling Factorials, and Discrete Splines: Another Look at Trend Filtering and Related Problems", Foundations and Trends® in Machine Learning: Vol. 15, No. 6, pp 694–846. DOI: 10.1561/2200000099.

1

Introduction

Nonparametric regression is a fundamental problem in statistics, in which we seek to flexibly estimate a smooth trend from data without relying on specific assumptions about its form or shape. The standard setup is to assume that data comes from a model (often called the "signal-plus-noise" model):

$$y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \dots, n.$$

Here, $f_0 : \mathcal{X} \to \mathbb{R}$ is an unknown function to be estimated, referred to as the *regression function*; $x_i \in \mathcal{X}$, $i = 1, \dots, n$ are *design points*, often (though not always) treated as nonrandom; $\epsilon_i \in \mathbb{R}$, $i = 1, \dots, n$ are random errors, usually assumed to be i.i.d. (independent and identically distributed) with zero mean; and $y_i \in \mathbb{R}$, $i = 1, \dots, n$ are referred to as *response points*. Unlike in a *parametric* problem, where we would assume f_0 takes a particular form (for example, a polynomial function) that would confine it to some finite-dimensional function space, in a *nonparametric* problem we make no such restriction, and instead assume f_0 satisfies some broader smoothness properties (for example, it has two bounded derivatives) that give rise to an infinite-dimensional function space.

The modern nonparametric toolkit contains an impressive collection of diverse methods, based on ideas like kernels, splines, and wavelets, to name just a few. Many estimators of interest in nonparametric regression can be formulated as the solutions to optimization problems based on the observed data. At a high level, such optimization-based methods can be divided into two camps. The first can be called the *continuous-time approach*, where we optimize over a function $f : \mathcal{X} \to \mathbb{R}$ that balances some notion of goodness-of-fit (to the data) with another notion of smoothness. The second can be called the *discrete-time approach*, where we optimize over function evaluations $f(x_1), \dots, f(x_n)$ at the design points, again to balance goodness-of-fit with smoothness.[1]

The main difference between these approaches lies in the optimization variable: in the first it is a function f, and in the second it is a vector $\theta = (f(x_1), \dots, f(x_n)) \in \mathbb{R}^n$. Each perspective comes with its advantages. The discrete-time approach is often much simpler, conceptually speaking, as it often requires only a fairly basic level of mathematics in order to explain and understand the formulation at hand. Consider, for example, a setting with $\mathcal{X} = [a, b]$ (the case of univariate design points), where we assume without a loss of generality that $x_1 < x_2 < \cdots < x_n$, and we define an estimator by the solution of the optimization problem:

$$\underset{\theta}{\text{minimize}} \;\; \frac{1}{2} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}|. \tag{1.1}$$

In the above criterion, each θ_i plays the role of a function evaluation $f(x_i)$; the first term measures the goodness-of-fit (via squared error loss) of the evaluations to the responses; the second term measures the jumpiness of the evaluations across neighboring design points, $\theta_i = f(x_i)$ and $\theta_{i+1} = f(x_{i+1})$; and $\lambda \geq 0$ is a tuning parameter determining the relative importance of the two terms for the overall minimization, with a larger λ translating into a higher importance on encouraging smoothness (mitigating jumpiness).

[1] The use of the word "time" here is completely informal. In some applications, the input $x \in \mathcal{X}$ might actually index time, and thus the names "continuous-time" and "discrete-time" would take on a direct meaning; but in general, they are only to be understood loosely, in reference to the distinction between modeling an entire function, and modeling function evaluations, as in (1.2) and (1.1), respectively.

Reasoning about the discrete-time problem (1.1) can be done without appealing to sophisticated mathematics, both conceptually and formally. Arguably, this could be appropriate for an introductory course on nonparametric statistical estimation. On the other hand, consider the estimator defined by the solution of the optimization problem:[2]

$$\underset{f}{\text{minimize}} \; \frac{1}{2} \sum_{i=1}^{n} \big(y_i - f(x_i)\big)^2 + \lambda \, \mathrm{TV}(f). \tag{1.2}$$

The minimization is taken over functions (for which the criterion is well-defined and finite); the first term measures the goodness-of-fit of the evaluations to the response points, as before; the second term measures the jumpiness of f, now using the total variation operator $\mathrm{TV}(\cdot)$ acting on univariate functions; and $\lambda \geq 0$ is again a tuning parameter. Relative to (1.1), the continuous-time problem (1.2) requires an appreciably higher level of mathematical sophistication, in order to develop any conceptual or formal understanding. However, problem (1.2) does have the distinct advantage of delivering a *function* as its solution, call it $\hat{f}$: this allows us to predict the value of the response at any point $x \in [a, b]$, via $\hat{f}(x)$.

From the solution in (1.1), call it $\hat{\theta}$, it is not immediately clear how to predict the response value at an arbitrary point $x \in [a, b]$. This is about choosing the "right" method for interpolating (or extrapolating, on $[a, x_1) \cup (x_n, b]$) a set of n function evaluations. To be fair, in the particular case of problem (1.1), its solution is generically piecewise-constant over its components $\hat{\theta}_i$, $i = 1, \ldots, n$, which suggests a natural interpolant. In general, however, the task of interpolating the estimated function evaluations from a discrete-time optimization problem into an entire estimated function is far from clear-cut. Likely for this reason, the statistics literature—which places a strong emphasis, both applied and theoretical, on prediction at a new points $x \in [a, b]$—has focused primarily on the continuous-time approach to optimization-based nonparametric regression. While the discrete-time approach is popular in

[2] Here and throughout, we say "the solution" only for simplicity. Problem (1.2), and more generally problem (1.7), need not admit unique solutions. The discrete-time problems (1.1) and (1.3) do, however, always admit unique solutions, because their criteria are strictly convex.

signal processing and econometrics, the lines of work on discrete- and continuous-time smoothing seem to have evolved mostly in parallel, with limited interplay.

The optimization problems in (1.1), (1.2) are not arbitrary examples of the discrete- and continuous-time perspectives, respectively; they are in fact deeply related to the main points of study in this monograph. Interestingly, problems (1.1), (1.2) are equivalent in the sense that their solutions, denoted $\hat{\theta}, \hat{f}$ respectively, satisfy $\hat{\theta}_i = \hat{f}(x_i)$, $i = 1, \dots, n$. In other words, the solution in (1.1) reproduces the evaluations of the solution in (1.2) at the design points. The common estimator here is well-known, called *total variation denoising* (Rudin *et al.*, 1992) in some parts of applied mathematics, and the *fused lasso* (Tibshirani *et al.*, 2005) in statistics.

The equivalence between (1.1), (1.2) is a special case of a more general equivalence between classes of discrete- and continuous-time optimization problems, in which the differences $\theta_i - \theta_{i+1}$ in (1.1) are replaced by higher-order discrete derivatives (based on divided differences), and $\mathrm{TV}(f)$ in (1.2) is replaced by the total variation of a suitable derivative of f. The key mathematical object powering this connection is a linear space of univariate piecewise polynomials called *discrete splines*, which is the central focus of this monograph. We dive into the details, and explain the importance of such equivalences, in the next subsection.

1.1 Motivation

The jumping-off point for the developments that follow is a generalization of the discrete-time total variation denoising problem (1.1), proposed independently by Steidl *et al.* (2006) and Kim *et al.* (2009) (though similar ideas were around earlier, see Section 2.6), defined for an integer $k \geq 0$ by:

$$\underset{\theta}{\text{minimize}} \ \frac{1}{2}\|y - \theta\|_2^2 + \lambda \|\mathbb{C}_n^{k+1} \theta\|_1. \tag{1.3}$$

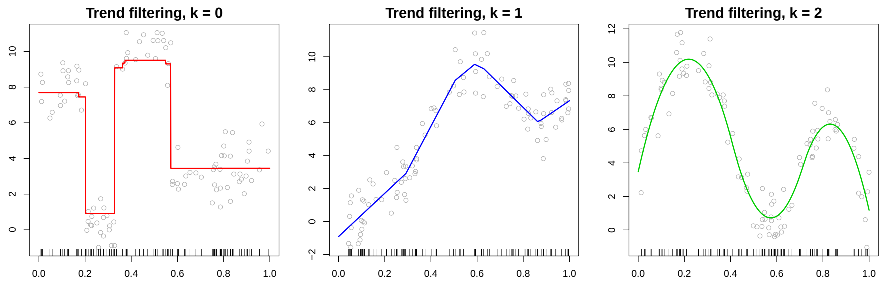
Here, $\lambda \geq 0$ is a tuning parameter, $y = (y_1, \dots, y_n) \in \mathbb{R}^n$ is the vector of response points, $\mathbb{C}_n^{k+1} \in \mathbb{R}^{(n-k-1)\times n}$ is an explicit banded matrix that corresponds to a weighted $(k+1)$st order discrete derivative operator (this can be defined in terms of the $(k+1)$st order divided difference coefficients across the design points; see the construction in (6.1)–(6.5)), and $\|\cdot\|_2$ and $\|\cdot\|_1$ are the standard ℓ_2 and ℓ_1 norms acting on vectors.

The estimator defined by solving problem (1.3) is known as *kth order trend filtering.* A important aspect to highlight right away is computational: since $\mathbb{C}_n^{k+1}$ is a banded matrix (with bandwidth $k+2$), the trend filtering problem (1.3) can be solved efficiently using various convex optimization techniques that take advantage of this structure (see, for example, Kim *et al.* (2009), Arnold and Tibshirani (2016), and Ramdas and Tibshirani (2016)). The original papers on trend filtering Steidl *et al.* (2006) and Kim *et al.* (2009) considered the special case of evenly-spaced design points, $x_{i+1} - x_i = v > 0$, $i = 1, \dots, n-1$, where the penalty term in (1.3) takes a perhaps more familiar form:

$$\|\mathbb{C}_n^{k+1}\theta\|_1 = \begin{cases} \dfrac{1}{v}\displaystyle\sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}| & \text{if } k=0 \\ \dfrac{1}{v^2}\displaystyle\sum_{i=1}^{n-2} |\theta_i - 2\theta_{i+1} + \theta_{i+2}| & \text{if } k=1 \\ \dfrac{1}{v^3}\displaystyle\sum_{i=1}^{n-3} |\theta_i - 3\theta_{i+1} + 3\theta_{i+2} - \theta_{i+3}| & \text{if } k=2, \end{cases} \tag{1.4}$$

and so forth, where for a general $k \geq 0$, the penalty is a $1/v^{k+1}$ times a sum of absolute $(k+1)$st forward differences. (The factor of $1/v^{k+1}$ can always be absorbed into the tuning parameter λ; and so we can see that (1.3) reduces to (1.1) for $k=0$, modulo a rescaling of λ). The extension of trend filtering to arbitrary (unevenly-spaced) design points is due to Tibshirani (2014). The continuous-time (functional) perspective on trend filtering is also due to Tibshirani (2014), which we describe next.

Connections to continuous-time. To motivate the continuous-time view, consider $\mathbb{C}_n^{k+1}\theta$, the vector of (weighted) $(k+1)$st discrete derivatives of θ across the design points: since discrete differentiation is based on iterated differencing, we can equivalently interpret $\mathbb{C}_n^{k+1}\theta$ as a vector of *differences* of kth discrete derivatives of θ at adjacent design points. By the sparsity-inducing property of the ℓ_1 norm, the penalty in problem


Trend filtering, k = 0

Trend filtering, k = 1

Trend filtering, k = 2

Figure 1.1: (Adapted from Tibshirani (2014).) Example trend filtering estimates for $k = 0$, $k = 1$, and $k = 2$, exhibiting piecewise constant, piecewise linear, and piecewise quadratic behavior, respectively. In each panel, the $n = 100$ design points are marked by ticks on the horizontal axis (note that they are not evenly-spaced).

(1.3) thus drives the kth discrete derivatives of θ to be equal at adjacent design points, and the trend filtering solution $\hat{\theta}$ generically takes on the structure of a kth degree piecewise polynomial (as its kth discrete derivative will be piecewise constant), with adaptively-chosen knots (points at which the kth discrete derivative changes). This intuition is readily confirmed by empirical examples; see Figure 1.1.

These ideas were formalized in Tibshirani (2014), and then developed further in Wang *et al.* (2014). These papers introduced what were called *kth degree falling factorial basis*, a set of functions defined as

$$\begin{aligned} h_j^k(x) &= \frac{1}{(j-1)!} \prod_{\ell=1}^{j-1} (x - x_\ell), \quad j = 1, \dots, k+1, \\ h_j^k(x) &= \frac{1}{k!} \prod_{\ell=j-k}^{j-1} (x - x_\ell) \cdot 1\{x > x_{j-1}\}, \quad j = k+2, \dots, n. \end{aligned} \tag{1.5}$$

(Note that this basis depends on the design points $x_1, \dots, x_n$, though this is notationally suppressed.) The functions in (1.5) are kth degree piecewise polynomials, with knots at $x_{k+1}, \dots, x_{n-1}$. Here and throughout, we interpret the empty product to be equal to 1, for convenience (that is, $\prod_{i=1}^0 a_i = 1$). Note the similarity of the above basis and the standard truncated power basis for splines, with knots at $x_{k+1}, \dots, x_{n-1}$ (see (2.5)); in fact, when $k = 0$ or $k = 1$, the two bases are equal, and the above falling factorial functions are exactly splines; but when

$k \geq 2$, this is no longer true—the above falling factorial functions are piecewise polynomials with *discontinuities* in their derivatives of orders $1, \dots, k-1$ (see (4.1), (4.2)), and thus span a different space than that of kth degree splines.

The key result connecting (1.5) and (1.3) was given in Lemma 5 of Tibshirani (2014) (see also Lemma 2 of Wang *et al.* (2014)), and can be explained as follows. For each $\theta \in \mathbb{R}^n$, there is a function in the span of the falling factorial basis, $f \in \text{span}\{h_1^k, \dots, h_n^k\}$, with two properties: first, f interpolates each θ_i at x_i, which we write as $\theta = f(x_{1:n})$, where $f(x_{1:n}) = (f(x_1), \dots, f(x_n)) \in \mathbb{R}^n$ denotes the vector of evaluations of f at the design points; and second

$$\text{TV}(D^k f) = \big\| \mathbb{C}_n^{k+1} f(x_{1:n}) \big\|_1. \tag{1.6}$$

On the right-hand side is the trend filtering penalty, which, recall, we can interpret as a sum of absolute differences of kth discrete derivatives of f over the design points, and therefore as a type of total variation penalty on the kth discrete derivative. On the left-hand side above, we denote by $D^k f$ the kth derivative of f (which we take to mean the kth left derivative when this does not exist), and by $\text{TV}(\cdot)$ the usual total variation operator on functions. Hence, taking total variation of the kth derivative as our smoothness measure, the property in (1.6) says that the interpolant f of θ is *exactly as smooth* in continuous-time as θ is in discrete-time.

Reflecting on this result, the first property—that f interpolates θ_i at x_i, for $i = 1, \dots, n$—is of course not special in it of itself. Any rich enough function class, of dimension at least n, will admit such a function. However, paired with the second property (1.6), the result becomes interesting, and even somewhat surprising. Said differently, any function f lying in the span of the kth degree falling factorial basis has the property that its discretization to the design points is *lossless* with respect to the total variation smoothness functional $\text{TV}(D^k f)$: this information is exactly preserved by $\theta = f(x_{1:n})$. Denoting by $\mathcal{H}_n^k = \text{span}\{h_1^k, \dots, h_n^k\}$ the span of falling factorial functions, we thus see that the trend filtering problem (1.3) is equivalent to the variational problem:

$$\underset{f \in \mathcal{H}_n^k}{\text{minimize}} \; \frac{1}{2} \sum_{i=1}^n \big(y_i - f(x_i) \big)^2 + \lambda \, \text{TV}(D^k f), \tag{1.7}$$

in the sense that at the solutions $\hat{\theta}, \hat{f}$ in problems (1.3), (1.7), respectively, we have $\hat{\theta} = \hat{f}(x_{1:n})$. Moreover, it turns out that forming $\hat{f}$ from $\hat{\theta}$ is straightforward: starting with the falling factorial basis expansion $\hat{f} = \sum_{j=1}^n \hat{\alpha}_j h_j^k$, and then writing the coefficient vector in block form $\hat{\alpha} = (\hat{a}, \hat{b}) \in \mathbb{R}^{k+1} \times \mathbb{R}^{n-k-1}$, the piecewise polynomial basis coefficients are given by $\hat{b} = \mathbb{C}_n^{k+1}\hat{\theta}$, and the polynomial basis coefficients $\hat{a}$ can also be expressed simply in terms of lower-order discrete derivatives. This shows that $\hat{f}$ is a kth degree piecewise polynomial, with knots occurring at the nonzeros of $\mathbb{C}_n^{k+1}\hat{\theta}$, that is, at changes in the kth discrete derivative of $\hat{\theta}$, formally justifying the intuition about the structure of $\hat{\theta}$ given above.

Reflections on the equivalence. One might say that the developments outlined above bring trend filtering closer to the "statistical mainstream": we move from being able to estimate the values of the regression function f_0 at the design points $x_1, \dots, x_n$ to being able to estimate f_0 itself. This has several uses: practical—we can use the interpolant $\hat{f}$ to estimate $f_0(x)$ at unseen values of x; conceptual—we can better understand what kinds of "shapes" trend filtering is inclined to produce, via the representation in terms of falling factorial functions; and theoretical—we can tie (1.7) to an unconstrained variational problem, where we minimize the same criterion over *all* functions f (for which the criterion is well-defined and finite):

$$\underset{f}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^n \big(y_i - f(x_i)\big)^2 + \lambda \, \mathrm{TV}(D^k f). \tag{1.8}$$

This minimization is in general computationally difficult, but its solution, called the *locally adaptive regression spline* estimator (Mammen and Geer, 1997) has favorable theoretical properties, in terms of its rate of estimation of f_0 (see Section 2.5 for a review). By showing that the falling factorial functions are "close" to certain splines, Tibshirani (2014) and Wang *et al.* (2014) showed that the solution in (1.7) is "close" to that in (1.8), and thus trend filtering inherits the favorable estimation guarantees of the locally adaptive regression spline (which is important because trend filtering is computationally easier; for more, see Sections 2.5 and 2.6).

The critical device in all of this were the falling factorial basis functions (1.5), which provide the bridge between the discrete and continuous worlds. This now brings us to the motivation for the current monograph. One has to wonder: did we somehow get "lucky" with trend filtering and this basis? Do the falling factorial functions have other properties aside from (1.6), that is, aside from equating (1.3) and (1.7)? At the time of writing Tibshirani (2014) and Wang *et al.* (2014) (and even in subsequent work on trend filtering), we were not fully aware of the relationship of the falling factorial functions and what appears to be fairly classical work in numerical analysis. First and foremost:

> The span $\mathcal{H}_n^k = \mathrm{span}\{h_1^k, \dots, h_n^k\}$ of the kth degree falling factorial basis functions is a special space of piecewise polynomials known as *kth degree discrete splines.*

Discrete splines have been studied since the early 1970s by applied mathematicians, beginning with Mangasarian and Schumaker (1971) and Mangasarian and Schumaker (1973). The current monograph recasts some of our previous work on trend filtering to better connect it to the discrete spline literature, reviews some relevant existing results on discrete splines and discusses the implications for trend filtering and related problems, and lastly, contributes some new results and perspectives on discrete splines.

1.2 Summary

An outline and summary of this monograph is as follows.

- In Section 2, we provide relevant background and historical remarks.

- In Section 3, we give a new perspective on how to construct the falling factorial basis "from scratch". We start by defining a natural discrete derivative operator and its inverse, a discrete integrator. We then show that the falling factorial basis functions are given by kth order discrete integration of appropriate step functions (Theorem 3.2).

- In Section 4, we verify that the span of the falling factorial basis is indeed a space of discrete splines (Lemma 4.1), and establish that functions in this span satisfy a key matching derivatives property: their kth discrete derivative matches their kth derivative everywhere, and moreover, they are the *only* kth degree piecewise polynomials with this property (Corollary 4.2).

- In Section 5, we give a dual basis to the falling factorial basis, based on evaluations of discrete derivatives. As a primary use case, we show how to use such a dual basis to perform efficient interpolation in the falling factorial basis, which generalizes Newton's divided difference interpolation formula (Theorem 5.4). We also show that this interpolation formula can be recast in an implicit manner, which reveals that interpolation using discrete splines can be done in *constant-time* (Corollary 5.5), and further, discrete splines are uniquely determined by this implicit result: they are the *only* functions that satisfy such an implicit interpolation formula (Corollary 5.6).

- In Section 6, we present a matrix-centric view of the results given in previous sections, drawing connections to the way some related results have been presented in past papers. We review specialized methods for fast matrix operations with discrete splines from Wang *et al.* (2014).

- In Section 7, we present a new discrete B-spline basis for discrete splines (it is new for arbitrary designs, and our construction here is a departure from the standard one): we first define these basis functions as discrete objects, by fixing their values at the design points, and we then define them as continuum functions, by interpolating these values within the space of discrete splines, using the implicit interpolation view (Lemma 7.2). We show how this discrete B-spline basis can be easily modified to provide a basis for discrete natural splines (Lemma 7.3).

- In Section 8, we demonstrate how the previous results and developments can be ported over to the case where the knot set that defines the space of discrete splines is an arbitrary (potentially sparse) subset of the design points. An important find here is that the discrete B-spline basis provides a much more stable (better-conditioned) basis for solving least squares problems involving discrete splines.

- In Section 9, we present two representation results for discrete splines. First, we review a result from Tibshirani (2014) and Wang *et al.* (2014) on representing the total variation functional $\mathrm{TV}(D^k f)$ for a kth degree discrete spline f in terms of a sum of absolute differences of its kth discrete derivatives (Theorem 9.1). (Recall that we translated this in (1.6).) Second, we establish a new result on representing the L_2-Sobolev functional $\int_a^b (D^m f)(x)^2 \, dx$ for a $(2m-1)$st degree discrete spline f in terms of a quadratic form of its mth discrete derivatives (Theorem 9.2).

- In Section 10, we derive simple (crude) approximation bounds for discrete splines, over bounded variation spaces.

- In Section 11, we revisit trend filtering. We discuss some potential computational improvements, stemming from the development of discrete B-splines and their stability properties. We also show that the optimization domain in trend filtering can be further restricted to the space of discrete natural splines by adding simple linear constraints to the original problem, and that this modification can lead to better boundary behavior.

- In Section 12, we revisit Bohlmann-Whittaker (BW) filtering. In the case of arbitrary design points, we propose a simple modification of the BW filter using a weighted penalty, which for $m = 1$ reduces to the linear smoothing spline. For $m = 2$, we derive a deterministic bound on the ℓ_2 distance between the weighted cubic BW filter and the cubic smoothing spline (Theorem 12.2). We use this, in combination with classical nonparametric regression theory for smoothing splines, to prove that the weighted BW filter attains minimax optimal estimation rates over the appropriate L_2-Sobolev classes (Corollary 12.3).

Most proofs are deferred to Appendix B. Other relevant technical details (background and otherwise) are deferred to Appendices C and D.

1.3 Notation

Here is an overview of some general notation used in this monograph. For integers $a \le b$, we use $z_{a:b} = \{z_a, z_{a+1}, \dots, z_b\}$. For a set C, we use 1_C for the indicator function of C, that is, $1_C(x) = 1\{x \in C\}$. We write $f|_C$ for the restriction of a function f to C. We use D for the differentiation operator, and I for the integration operator: acting on functions f on $[a, b]$, we take If to itself be a function on $[a, b]$, defined by

$$(If)(x) = \int_a^x f(t)\,dt.$$

For a nonnegative integer k, we use D^k and I^k to denote k repeated applications (that is, k times composition) of the differentiation and integration operators, respectively. In general, when the derivative of a function f does not exist, we interpret Df to mean the *left* derivative, assuming the latter exists, and the same with $D^k f$.

An important note: we refer to a kth degree piecewise polynomial that has $k-1$ continuous derivatives as a spline of *degree* k, whereas much of the classical literature refers to this as a spline of *order* $k+1$; we specifically avoid the use of the word “order” when it comes to such functions or functions spaces, to avoid confusion.

Finally, throughout, we use “blackboard” fonts for matrices (such as $\mathbb{F}, \mathbb{G}$, etc.), in order to easily distinguish them from operators that act on functions (for which we use F, G, etc.). The only exceptions are that we reserve $\mathbb{R}$ to denote the set of real numbers and $\mathbb{E}$ to denote the expectation opterator.

For a more detailed summary of notation, and discrete-continuum analogies or equivalences, see Appendix A.

Acknowledgements

We are grateful to Yu-Xiang Wang for his many, many insights and inspiring conversations over the years. His lasting enthusiasm helped fuel our own interest in “getting to the bottom” of the falling factorial basis, and putting this monograph together. We also thank our other collaborators on trend filtering papers: Aaditya Ramdas, Veeranjaneyulu Sadhanala, James Sharpnack, and Alex Smola. Finally, we are grateful to Addison Hu, Alden Green, Pratik Patil, Veeranjaneyulu Sadhanala, and Yu-Xiang Wang for their helpful comments and feedback on this monograph as a whole; to Yining Wang for his help with Section 12.2; and to Pratik Patil for his help with Section 12.3. This monograph is based upon work supported by the National Science Foundation under Grant No. DMS-1554123.

Appendices

A

Notation Table

Table A.1: Main notation, and discrete-continuum analogies/equivalences in this monograph. We omit notational dependence on the domain $[a, b]$ for simplicity.

Discrete object	Reference	Continuum obj.	Reference	Notes
Operators				
$\Delta_n^k = \Delta^k(\cdot; x_{1:n})$, kth order discrete differentiation with respect to design point $x_{1:n}$	(3.1)	D^k, kth order differentiation	–	$(\Delta_n^k f)(x) = (D^k f)(x)$, for $f \in \mathcal{H}_n^k$ and $x > x_k$ (Corollary 4.2)
$S_n^k = S^k(\cdot; x_{1:n})$, kth order discrete integration	(3.8), (3.9)	I^k, kth order integration	–	$S_n^k = (D_n^k)^{-1}$ (Lemma 3.1)

Table A.1: (Continued)

Discrete object	Reference	Continuum obj.	Reference	Notes
Spaces				
$\mathcal{DS}_n^k(t_{1:r})$, kth degree discrete splines with knots $t_{1:r}$ and design points $x_{1:n}$	Definition 4.1	$\mathcal{S}^k(t_{1:r})$, kth degree splines with knots $t_{1:r}$	Definition 2.1	These spaces coincide for $k=0$ and $k=1$
$\mathcal{H}_n^k = \mathcal{DS}_n^k$ $(x_{(k+1):(n-1)})$	–	$\mathcal{G}_n^k =$ $\mathcal{S}^k(x_{(k+1):(n-1)})$	–	Abbreviations for the "canonical" spaces, with knots $x_{(k+1):(n-1)}$
Bases				
h_j^k, $j=1,\dots,n$, kth degree falling factorial basis for $\mathcal{H}_n^k$	(1.5)	g_j^k, $j=1,\dots,n$, kth degree truncated power basis for $\mathcal{G}_n^k$	(2.5)	Falling factorials can be seen as truncated Newton polynomials, and have dual relationship to discrete differentiation (Lemma 5.3)
Q_j^k and N_j^k, $j=1,\dots,n$, unnormalized and normalized kth degree DB-spline basis for $\mathcal{H}_n^k$	(7.2), (7.3), (7.5)	P_j^k and M_j^k, $j=1,\dots,n$, unnormalized and normalized kth degree B-spline basis for $\mathcal{G}_n^k$	(C.3), (C.4), (C.6)	The basis in (C.6) is actually defined for an arbitrary knot set $t_{1:r}$; for arbitrary knots in the DB-spline setting, see (8.5), (8.6)

Table A.1: (Continued)

Discrete object	Reference	Continuum obj.	Reference	Notes
Matrices				
$\mathbb{D}_n^k$, kth order discrete derivative matrix with respect to design points $x_{1:n}$	(6.1), (6.2), (6.3)	–	–	Multiplying by a vector of evaluations gives discrete derivatives at design points $x_{(k+1):n}$, as in (6.4)
$\mathbb{B}_n^k$, kth order extended discrete derivative matrix with respect to design points $x_{1:n}$	(6.6), (6.7), (6.8)	–	–	Multiplying by a vector of evaluations gives discrete derivatives at all design points $x_{1:n}$, as in (6.9)
$\mathbb{H}_n^k$, kth degree falling factorial basis matrix with respect to design points $x_{1:n}$	Basis in kth degree trend filter (2.20)	$\mathbb{G}_n^k$, kth degree truncated power basis matrix with respect to design points $x_{1:n}$	Basis in kth degree restricted locally adaptive regression spline (2.19)	$\mathbb{H}_n^k = (\mathbb{Z}_n^{k+1} \mathbb{B}_n^{k+1})^{-1}$, see (6.12); results in fast algorithms for matrix computations in $\mathbb{H}_n^k$, see Appendix D

Table A.1: (Continued)

Discrete object	Reference	Continuum obj.	Reference	Notes
Smoothness functionals				
$\|\mathbb{W}_n^{k+1}\mathbb{D}_n^{k+1}\theta\|_1 = \sum_{i=1}^{n-k-1} \|(\mathbb{D}_n^k\theta)_{i+1} - (\mathbb{D}_n^k\theta)_i\|$, kth order discrete total variation of vector θ	Penalty in kth degree trend filter (2.21)	$\mathrm{TV}(D^k f)$, kth order total variation of function f	Penalty in kth degree locally adaptive regression spline (1.8)	Equal for $\theta = f(x_{1:n})$ and $f \in \mathcal{H}_n^k$ (Theorem 9.1)
$\|(\mathbb{W}_n^m)^{\frac{1}{2}}\mathbb{D}_n^m\theta\|_2^2 = \sum_{i=1}^{n-m}(\mathbb{D}_n^m\theta)_i^2 (x_{i+m} - x_i)/m$, mth order discrete Sobolev seminorm of vector θ	Penalty in kth degree BW filter (12.1), for $k = 2m-1$	$\int_a^b (D^m f)(x)^2 \, dx$, mth order Sobolev seminorm of function f	Penalty in kth degree smoothing spline (2.7), for $k = 2m-1$	Equal for $\theta = f(x_{1:n})$ and $m = 1$ (Lemma 12.4), but not in general; see also Theorem 9.2

B

Proofs

B.1 Proof of Theorem 2.1

Since f is a natural spline of degree $2m-1$ with knots in $x_{1:n}$, we know that $D^m f$ is a spline of degree $m-1$ with knots in $x_{1:n}$, and moreover, it is supported on $[x_1, x_n]$. Thus we can expand $D^m f = \sum_{i=1}^{n-m} \alpha_i P_i^{m-1}$ for coefficients α_i, $i = 1, \dots, n-m$, and

$$\int_a^b (D^m f)(x)^2 \, dx = \alpha^\mathsf{T} \mathbb{Q} \alpha, \tag{B.1}$$

where $\mathbb{Q} \in \mathbb{R}^{(n-m)\times(n-m)}$ has entries $\mathbb{Q}_{ij} = \int_a^b P_i^{m-1}(x) P_j^{m-1}(x) \, dx$. But we can also write

$$\begin{aligned} \int_a^b (D^m f)(x)^2 \, dx &= \int_a^b (D^m f)(x) \sum_{i=1}^{n-m} \alpha_i P_i^{m-1}(x) \, dx \\ &= \sum_{i=1}^{n-m} \alpha_i \int_a^b (D^m f)(x) P_i^{m-1}(x) \, dx \\ &= \frac{1}{m} \sum_{i=1}^{n-m} \alpha_i \big(\mathbb{D}_n^m f(x_{1:n})\big)_i \\ &= \frac{1}{m} \alpha^\mathsf{T} \mathbb{D}_n^m f(x_{1:n}), \end{aligned} \tag{B.2}$$

where in the third line, we used the Peano representation for B-splines, as described in (C.1) in Appendix C.1, which implies that for $i = 1, \dots, n-m$,

$$(m-1)! \cdot f[x_i, \dots, x_{i+m}] = \int_a^b (D^m f)(x) P_i^{m-1}(x)\, dx.$$

Comparing (B.1) and (B.2), we learn that $\mathbb{Q}\alpha = \mathbb{D}_n^m f(x_{1:n})/m$, that is, $\alpha = \mathbb{Q}^{-1}\mathbb{D}_n^m f(x_{1:n})/m$, and therefore

$$\begin{aligned}\int_a^b (D^m f)(x)^2\, dx &= \frac{1}{m^2}\big(\mathbb{Q}^{-1}\mathbb{D}_n^m f(x_{1:n})\big)^\mathsf{T} \mathbb{Q}\, \mathbb{Q}^{-1}\mathbb{D}_n^m f(x_{1:n}) \\ &= \frac{1}{m^2}\big(\mathbb{D}_n^m f(x_{1:n})\big)^\mathsf{T} \mathbb{Q}^{-1}\mathbb{D}_n^m f(x_{1:n}),\end{aligned}$$

which proves (2.8), (2.9) with $\mathbb{K}_n^m = (1/m^2)\mathbb{Q}^{-1}$, that is, $(\mathbb{K}_n^m)^{-1} = m^2\mathbb{Q}$.

When $m = 1$, for each $i = 1, \dots, n-1$, we have the simple form for the constant B-spline:

$$P_i^0(x) = \begin{cases} \dfrac{1}{x_{i+1} - x_i} & \text{if } x \in I_i \\ 0 & \text{otherwise.} \end{cases}$$

where $I_1 = [x_1, x_2]$, and $I_i = (x_i, x_{i+1}]$ for $i = 2, \dots, n-1$. The result (2.10) comes from straightforward calculation of $\int_a^b P_i^0(x)^2\, dx$. Lastly, when $m = 2$, for each $i = 1, \dots, n-2$, we have the linear B-spline:

$$P_i^1(x) = \begin{cases} \dfrac{x - x_i}{(x_{i+2} - x_i)(x_{i+1} - x_i)} & \text{if } x \in I_i^- \\ \dfrac{x_{i+2} - x}{(x_{i+2} - x_i)(x_{i+2} - x_{i+1})} & \text{if } x \in I_i^+ \\ 0 & \text{otherwise,} \end{cases}$$

where $I_1^- = [x_1, x_2]$, $I_i^- = (x_i, x_{i+1}]$, $i = 2, \dots, n-2$, $I_i^+ = (x_{i+1}, x_{i+2}]$, $i = 1, \dots, n-2$. The two cases in (2.11) again come from straightforward calculation of the integrals $\int_a^b P_i^1(x)^2\, dx$ and $\int_a^b P_i^1(x) P_{i-1}^1(x)\, dx$, completing the proof.

B.2 Proof of the Linear Combination Formulation (3.10)

Denote by $g(x)$ the right-hand side of (3.10). We will show that $\Delta_n^k g = f$. Note by Lemma 3.1, this would imply that $g = S_n^k f$, proving (3.10). An inductive argument similar to that in the proof of Lemma 3.3 shows that, for $x \in (x_i, x_{i+1}]$ and $i \geq k$,

$$(\Delta_n^k g)(x) = \sum_{j=1}^{k} (\Delta_n^k h_j^{k-1})(x) \cdot f(x_j) + \sum_{j=k+1}^{i} (\Delta_n^k h_j^{k-1})(x) \cdot \frac{x_j - x_{j-k}}{k} \cdot f(x_j) + (\Delta_n^k h_{i+1}^{k-1})(x) \cdot \frac{x - x_{i-k+1}}{k} \cdot f(x).$$

By Lemmas 5.1 and 5.2, all discrete derivatives here are zero except the last, which is $(\Delta_n^k h_{i+1}^{k-1})(x)(x - x_{i-k+1})/k = 1$. Thus we have shown $(\Delta_n^k g)(x) = f(x)$. Similarly, for $x \in (x_i, x_{i+1}]$ and $i < k$,

$$(\Delta_n^k g)(x) = \sum_{j=1}^{i} (\Delta_n^k h_j^{k-1})(x) \cdot f(x_j) \; + \; (\Delta_n^k h_{i+1}^{k-1})(x) \cdot f(x),$$

and by Lemma (5.2), all discrete derivatives here are zero except the last, which is $(\Delta_n^k h_{i+1}^{k-1})(x) = 1$. For $x \leq x_1$, we have $g(x) = f(x)$ by definition. This establishes the desired claim and completes the proof.

B.3 Proof of Lemma 3.1

We use induction, beginning with $k = 1$. Using (3.8), (3.6), we can express the first order discrete integral operator S_n more explicitly as

$$(S_n f)(x) = \begin{cases} f(x_1) + \sum_{j=2}^{i} f(x_j)(x_j - x_{j-1}) + f(x)(x - x_i) & \text{if } x \in (x_i, x_{i+1}] \\ f(x) & \text{if } x \leq x_1. \end{cases} \tag{B.3}$$

Compare (3.3) and (B.3). For $x \leq x_1$, clearly $(\Delta_n S_n f)(x) = f(x)$ and $(S_n \Delta_n f)(x) = f(x)$, and for $x \in (x_i, x_{i+1}]$,

$$\begin{aligned}(\Delta_n S_n f)(x) &= \frac{(S_n f)(x) - (S_n f)(x_i)}{x - x_i} \\ &= \frac{f(x_1) + \sum_{j=2}^{i} f(x_j)(x_j - x_{j-1}) + f(x)(x - x_i)}{x - x_i} -\end{aligned}$$

$$\frac{\big(f(x_1) + \sum_{j=2}^{i} f(x_j)(x_j - x_{j-1})\big)}{x - x_i}$$
$$= f(x),$$

and also

$$\begin{aligned}(S_n \Delta_n f)(x) &= f(x_1) + \sum_{j=2}^{i} (\Delta_n f)(x_j) \cdot (x_j - x_{j-1}) + (\Delta_n f)(x) \cdot (x - x_i) \\ &= f(x_1) + \sum_{j=2}^{i} \big(f(x_j) - f(x_{j-1})\big) + f(x) - f(x_i) \\ &= f(x).\end{aligned}$$

Now assume the result is true for the order $k-1$ operators. Then, we have from (3.7), (3.9),

$$\Delta_n^k \circ S_n^k = (W_n^k)^{-1} \circ \overline{\Delta}_{n-k+1} \circ \Delta_n^{k-1} \circ S_n^{k-1} \circ \overline{S}_{n-k+1} \circ W_n^k = \mathrm{Id},$$

and also

$$S_n^k \circ \Delta_n^k = S_n^{k-1} \circ \overline{S}_{n-k+1} \circ W_n^k \circ (W_n^k)^{-1} \circ \overline{\Delta}_{n-k+1} \circ \Delta_n^{k-1} = \mathrm{Id},$$

where Id denotes the identity operator. This completes the proof.

B.4 Proof of Lemma 3.3

The case $d = 0$. Beginning with the case $d = 0$, the desired result in (3.13) reads

$$\frac{1}{k!} \prod_{m=j-k}^{j-1} (x - x_m) = \sum_{\ell=j}^{i} \frac{1}{(k-1)!} \prod_{m=\ell-k+1}^{\ell-1} (x - x_m) \frac{x_\ell - x_{\ell-k}}{k} + \\ \frac{1}{(k-1)!} \prod_{m=i-k+2}^{i} (x - x_m) \frac{x - x_{i-k+1}}{k},$$

or more succintly,

$$\eta(x; x_{(j-k):(j-1)}) = \sum_{\ell=j}^{i} \eta(x; x_{(\ell-k+1):(\ell-1)})(x_\ell - x_{\ell-k}) + \eta(x; x_{(i-k+2):i}),$$

The above display is a consequence of an elementary result (B.4) on Newton polynomials. We state and prove this result next, which we note completes the proof for the case $d = 0$.

Lemma B.1. For any $k \geq 1$, and points $t_{1:r}$ with $r \geq k$, the Newton polynomials defined in (2.2) satisfy, at any x,

$$\eta(x;t_{1:k}) - \eta(x;t_{(r-k+1):r}) = \sum_{\ell=k+1}^{r} \eta(x;t_{(\ell-k+1):(\ell-1)})(t_\ell - t_{\ell-k}). \tag{B.4}$$

Proof. Observe that

$$\begin{aligned} \eta(x;t_{1:k}) - \eta(x;t_{2:(k+1)}) &= \eta(x;t_{2:k})\big((x-t_1) - (x-t_{k+1})\big) \\ &= \eta(x;t_{2:k})(t_{k+1} - t_1). \end{aligned} \tag{B.5}$$

Therefore

$$\begin{aligned} \eta(x;t_{1:k}) - \eta(x;t_{(r-k+1):r}) = {} & \underbrace{\eta(x;t_{1:k}) - \eta(x;t_{2:(k+1)})}_{a_1} + \\ & \underbrace{\eta(x;t_{2:(k+1)}) - \eta(x;t_{3:(k+2)})}_{a_2} + \cdots + \\ & \underbrace{\eta(x;t_{(r-k):r-1}) - \eta(x;t_{(r-k+1):r})}_{a_{r-k}}. \end{aligned}$$

In a similar manner to (B.5), for each $i = 1, \ldots, k$, we have $a_i = \eta(x;t_{(i+1):(i+k-1)})(t_{i+k} - t_i)$, and the result follows, after making the substitution $\ell = i + k$. □

The case $d \geq 1$. We now prove the result (3.13) for $d \geq 1$ by induction. The base case was shown above, for $d = 0$. Assume the result holds for discrete derivatives of order $d-1$. If $x \leq x_d$ (or $d > n$), then $(\Delta_n^d f)(x) = (\Delta_n^{d-1} f)(x)$ for all functions f and thus the desired result holds trivially. Hence assume $x > x_d$ (which implies that $i \geq d$). By the inductive hypothesis,

$$\begin{aligned} &(\Delta_n^{d-1} h_j^k)(x) - (\Delta_n^{d-1} h_j^k)(x_i) \\ &= \sum_{\ell=j}^{i} (\Delta_n^{d-1} h_\ell^{k-1})(x) \cdot \frac{x_\ell - x_{\ell-k}}{k} + (\Delta_n^{d-1} h_{i+1}^{k-1})(x) \cdot \frac{x - x_{i-k+1}}{k} - \\ &\quad \sum_{\ell=j}^{i} (\Delta_n^{d-1} h_\ell^{k-1})(x_i) \cdot \frac{x_\ell - x_{\ell-k}}{k} \end{aligned}$$

$$= \sum_{\ell=j}^{i} \big((\Delta_n^{d-1} h_\ell^{k-1})(x) - (\Delta_n^{d-1} h_\ell^{k-1})(x_i)\big) \cdot \frac{x_\ell - x_{\ell-k}}{k} + \\ \big((\Delta_n^{d-1} h_{i+1}^{k-1})(x) - (\Delta_n^{d-1} h_{i+1}^{k-1})(x_i)\big) \cdot \frac{x - x_{i-k+1}}{k},$$

where in the last line we used the fact that $h_{i+1}^{k-1} = 0$ on $[a, x_i]$, and thus $(\Delta_n^{d-1} h_{i+1}^{k-1})(x_i) = 0$. This means, using (3.4),

$$\begin{aligned}(\Delta_n^d h_j^k)(x) &= \frac{(\Delta_n^{d-1} h_j^k)(x) - (\Delta_n^{d-1} h_j^k)(x_i)}{(x - x_{i-d+1})/d} \\ &= \sum_{\ell=j}^{i} \frac{(\Delta_n^{d-1} h_\ell^{k-1})(x) - (\Delta_n^{d-1} h_\ell^{k-1})(x_i)}{(x - x_{i-d+1})/d} \cdot \frac{x_\ell - x_{\ell-k}}{k} + \\ &\quad \frac{(\Delta_n^{d-1} h_{i+1}^{k-1})(x) - (\Delta_n^{d-1} h_{i+1}^{k-1})(x_i)}{(x - x_{i-d+1})/d} \cdot \frac{x - x_{i-k+1}}{k} \\ &= \sum_{\ell=j}^{i} (\Delta_n^d h_\ell^{k-1})(x) \cdot \frac{x_\ell - x_{\ell-k}}{k} + (\Delta_n^d h_{i+1}^{k-1})(x) \cdot \frac{x - x_{i-k+1}}{k},\end{aligned}$$

as desired. This completes the proof.

B.5 Lemma B.2 (Helper Result for Corollary 5.6)

Lemma B.2. Given distinct points $t_i \in [a, b]$, $i = 1, \dots, r$ and evaluations $f(t_i)$, $i = 1, \dots, r$, if f satisfies

$$f[t_1, \dots, t_r, x] = 0, \quad \text{for } x \in [a, b],$$

then f is a polynomial of degree r.

Proof. We will actually prove a more general result, namely, that if f satisfies

$$f[t_1, \dots, t_r, x] = p_\ell(x), \quad \text{for } x \in [a, b], \tag{B.6}$$

where p_ℓ is a polynomial of degree ℓ, then f is a polynomial of degree $r + \ell$. We use induction on r. For $r = 0$, the statement (B.6) clearly holds for all ℓ, because $f[x] = f(x)$ (a zeroth order divided difference is simply evaluation). Now assume (B.6) holds for any $r - 1$ centers and all degrees ℓ. Then

$$p_\ell(x) = f[t_1, \dots, t_r, x] = \frac{f[t_2, \dots, t_r, x] - f[t_1, \dots, t_r]}{x - t_1},$$

which means $f[t_2, \dots, t_r, x] = (x - t_1)p_\ell(x) + f[t_1, \dots, t_r]$. As the right-hand side is a polynomial of degree $\ell + 1$, the inductive hypothesis implies that f is a polynomial of degree $r - 1 + \ell + 1 = r + \ell$, completing the proof. □

B.6 Proof of Theorem 9.2

Let h_j^k, $j = 1, \dots, n$ denote the falling factorial basis, as in (1.5). Consider expanding f in this basis, $f = \sum_{j=1}^n \alpha_j h_j^k$. Define $\mathbb{Q} \in \mathbb{R}^{n \times n}$ to have entries

$$\mathbb{Q}_{ij} = \int_a^b (D^m h_i^k)(x)(D^m h_j^k)(x)\,dx. \tag{B.7}$$

Observe

$$\begin{aligned} \int_a^b (D^m f)(x)^2\,dx &= \int_a^b \sum_{i,j=1}^n \alpha_i \alpha_j (D^m h_i^k)(x)(D^m h_j^k)(x)\,dx \\ &= \alpha^\mathsf{T} \mathbb{Q} \alpha \\ &= f(x_{1:n})^\mathsf{T} (\mathbb{H}_n^k)^{-\mathsf{T}} \mathbb{Q} (\mathbb{H}_n^k)^{-1} f(x_{1:n}) \\ &= f(x_{1:n})^\mathsf{T} (\mathbb{B}_n^{k+1})^\mathsf{T} \mathbb{Z}_n^{k+1}\, \mathbb{Q}\, \mathbb{Z}_n^{k+1}\, \mathbb{B}_n^{k+1} f(x_{1:n}). \end{aligned} \tag{B.8}$$

In the third line above we used the expansion $f(x_{1:n}) = \mathbb{H}_n^k \alpha$, where $\mathbb{H}_n^k$ is the kth degree falling factorial basis with entries $(\mathbb{H}_n^k)_{ij} = h_j^k(x_i)$, and in the fourth line we applied the inverse relationship in (6.12), where $\mathbb{B}_n^{k+1}$ is the $(k+1)$st order extended discrete derivative matrix in (6.8) and $\mathbb{Z}_n^{k+1}$ is the extended weight matrix in (6.7). Now note that we can unravel the recursion in (6.8) to yield

$$\mathbb{B}_n^{k+1} = (\mathbb{Z}_n^{k+1})^{-1} \underbrace{\overline{\mathbb{B}}_{n,k+1} (\mathbb{Z}_n^k)^{-1} \overline{\mathbb{B}}_{n,k} \cdots (\mathbb{Z}_n^{m+1})^{-1} \overline{\mathbb{B}}_{n,m+1}}_{\mathbb{F}}\, \mathbb{B}_n^m, \tag{B.9}$$

and returning to (B.8), we get

$$\int_a^b (D^m f)(x)^2\,dx = f(x_{1:n})^\mathsf{T} (\mathbb{B}_n^m)^\mathsf{T} \mathbb{F}^\mathsf{T} \mathbb{Q}\, \mathbb{F}\, \mathbb{B}_n^m f(x_{1:n}). \tag{B.10}$$

We break up the remainder of the proof up into parts for readability.

Reducing (B.10) to involve only discrete derivatives. First we show that the right-hand side in (B.10) really depends on the discrete derivatives $\mathbb{D}_n^m f(x_{1:n})$ only (as opposed to extended discrete derivatives $\mathbb{B}_n^m f(x_{1:n})$). As the first m basis functions $h_1^k, \dots, h_m^k$ are polynomials of degree at most $m-1$, note that their mth derivatives are zero, and hence we can write

$$\mathbb{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{M} \end{bmatrix},$$

where $\mathbb{M} \in \mathbb{R}^{(n-m)\times(n-m)}$ has entries as in (9.6). Furthermore, note that $\mathbb{F}$ as defined in (B.9) can be written as

$$\mathbb{F} = \begin{bmatrix} \mathbb{I}_m & 0 \\ 0 & \mathbb{G} \end{bmatrix},$$

for a matrix $\mathbb{G} \in \mathbb{R}^{(n-m)\times(n-m)}$. Therefore

$$\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{G}^\mathsf{T} \mathbb{M} \mathbb{G} \end{bmatrix}, \tag{B.11}$$

and hence (B.10) reduces to

$$\int_a^b (D^m f)(x)^2 \, dx = f(x_{1:n})^\mathsf{T} (\mathbb{D}_n^m)^\mathsf{T} \underbrace{\mathbb{G}^\mathsf{T} \mathbb{M} \mathbb{G}}_{\mathbb{V}_n^m} \mathbb{D}_n^m f(x_{1:n}), \tag{B.12}$$

recalling that $\mathbb{D}_n^m$ is exactly given by the last $n-m$ rows of $\mathbb{B}_n^m$.

Casting $\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F}$ in terms of scaled differences. Next we prove that $\mathbb{V}_n^m = \mathbb{G}^\mathsf{T} \mathbb{M} \mathbb{G}$, as defined in (B.12), is a banded matrix. To prevent unnecessary indexing difficulties, we will actually just work directly with $\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F}$, and then in the end, due to (B.11), we will be able to read off the desired result according to the lower-right submatrix of $\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F}$, of dimension $(n-m) \times (n-m)$. Observe that

$$\begin{aligned} \mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F} = {} & (\bar{\mathbb{B}}_{n,m+1})^\mathsf{T} (\mathbb{Z}_n^{m+1})^{-1} \cdots (\bar{\mathbb{B}}_{n,k})^\mathsf{T} (\mathbb{Z}_n^k)^{-1} \\ & (\bar{\mathbb{B}}_{n,k+1})^\mathsf{T} \mathbb{Q} \bar{\mathbb{B}}_{n,k+1} (\mathbb{Z}_n^k)^{-1} \bar{\mathbb{B}}_{n,k} \cdots (\mathbb{Z}_n^{m+1})^{-1} \bar{\mathbb{B}}_{n,m+1}. \end{aligned} \tag{B.13}$$

To study this, it helps to recall the notation introduced in Lemma 9.3: for a matrix $\mathbb{A}$ and positive integers i, j, let

$$\mathbb{A}(i,j) = \begin{cases} \mathbb{A}_{ij} & \text{if } \mathbb{A} \text{ has at least } i \text{ rows and } j \text{ columns} \\ 0 & \text{otherwise,} \end{cases}$$

as well as

$$
\begin{aligned}
\delta^r_{ij}(\mathbb{A}) &= \mathbb{A}(i,j) - \mathbb{A}(i+1,j),\\
\delta^c_{ij}(\mathbb{A}) &= \mathbb{A}(i,j) - \mathbb{A}(i,j+1).
\end{aligned}
$$

Now to compute (B.13), we first compute the product

$$
\mathbb{F}^{\mathsf{T}}\mathbb{Q} = (\overline{\mathbb{B}}_{n,m+1})^{\mathsf{T}}(\mathbb{Z}_n^{m+1})^{-1}\cdots(\overline{\mathbb{B}}_{n,k})^{\mathsf{T}}(\mathbb{Z}_n^{k})^{-1}(\overline{\mathbb{B}}_{n,k+1})^{\mathsf{T}}\mathbb{Q}.
$$

We will work "from right to left". From (6.6), we have

$$
(\overline{\mathbb{B}}_{n,k+1})^{\mathsf{T}} = \left[\begin{array}{ccccccccc}
1 & 0 & \dots & 0 & & & & & \\
0 & 1 & \dots & 0 & & & 0 & & \\
\vdots & & & & & & & & \\
0 & 0 & \dots & 1 & & & & & \\
 & & & 1 & -1 & 0 & \dots & 0 & 0\\
 & & & 0 & 1 & -1 & \dots & 0 & 0\\
 & 0 & & \vdots & & & & & \\
 & & & 0 & 0 & 0 & \dots & 1 & -1\\
 & & & 0 & 0 & 0 & \dots & 0 & 1
\end{array}\right]
\begin{array}{l}
\left.\vphantom{\begin{array}{c}1\\0\\\vdots\\0\end{array}}\right\} k \text{ rows}\\
\left.\vphantom{\begin{array}{c}1\\0\\\vdots\\0\\0\end{array}}\right\} n-k \text{ rows}
\end{array}
$$

This shows left multiplication by $(\overline{\mathbb{B}}_{n,k+1})^{\mathsf{T}}$ gives row-wise differences, $((\overline{\mathbb{B}}_{n,k+1})^{\mathsf{T}}\mathbb{A})_{ij} = \delta^r_{ij}(\mathbb{A})$, for $i > k$. Further, from (6.7), we can see that left multiplication by $(\mathbb{Z}_n^k)^{-1}$ applies a row-wise scaling, $(\mathbb{Z}_n^k)^{-1}\mathbb{A} = \mathbb{A}_{ij}\cdot k/(x_i - x_{i-k})$, for $i > k$. Thus letting $\mathbb{U}^{1,0} = (\mathbb{Z}_n^k)^{-1}(\overline{\mathbb{B}}_{n,k+1})^{\mathsf{T}}\mathbb{Q}$, its entries are:

$$
\mathbb{U}^{1,0}_{ij} = \begin{cases}
\mathbb{Q}_{ij} & \text{if } i \le k\\
\delta^r_{ij}(\mathbb{Q}) \cdot \dfrac{k}{x_i - x_{i-k}} & \text{if } i > k.
\end{cases}
$$

The next two products to consider are left multiplication by $(\overline{\mathbb{B}}_{n,k})^{\mathsf{T}}$ and by $(\mathbb{Z}_n^{k-1})^{-1}$, which act similarly (they again produce row-wise differencing and scaling, respectively). Continuing on in this same manner, we get that $\mathbb{F}^{\mathsf{T}}\mathbb{Q} = \mathbb{U}^{m,0}$, where $\mathbb{U}^{\ell,0}$, $\ell = 1,\dots,m-1$ satisfy the recursion relation (setting $\mathbb{U}^{0,0} = \mathbb{Q}$ for convenience):

$$
\mathbb{U}^{\ell,0}_{ij} = \begin{cases}
\mathbb{U}^{\ell-1,0}_{ij} & \text{if } i \le k+1-\ell\\
\delta^r_{ij}(\mathbb{U}^{\ell-1,0}) \cdot \dfrac{k+1-\ell}{x_i - x_{i-(k+1-\ell)}} & \text{if } i > k+1-\ell,
\end{cases} \tag{B.14}
$$

and where (using $k+1-m=m$):

$$\mathbb{U}_{ij}^{m,0} = \begin{cases} \mathbb{U}_{ij}^{m-1,0} & \text{if } i \le m \\ \delta_{ij}^{r}(\mathbb{U}^{m-1,0}) & \text{if } i > m. \end{cases} \tag{B.15}$$

The expressions (B.14), (B.15) are equivalent to (9.7), (9.8), the row-wise recursion in Lemma 9.3 (the main difference is that Lemma 9.3 is concerned with the lower-right $(n-m)\times(n-m)$ submatrices of these matrices, and so these recursive expressions are written with i, j replaced by $i+m, j+m$, respectively).

The other half of computing (B.13) is of course to compute the product

$$\mathbb{F}^{\mathsf{T}}\mathbb{Q}\,\mathbb{F} = \mathbb{F}^{\mathsf{T}}\mathbb{Q}\,\overline{\mathbb{B}}_{n,k+1}(\mathbb{Z}_n^k)^{-1}\overline{\mathbb{B}}_{n,k}\cdots(\mathbb{Z}_n^{m+1})^{-1}\overline{\mathbb{B}}_{n,m+1}.$$

Working now "from left to right", this calculation proceeds analogously to the case just covered, but with column-wise instead of row-wise updates, and we get $\mathbb{F}^{\mathsf{T}}\mathbb{Q}\,\mathbb{F} = \mathbb{U}^{m,m}$, where $\mathbb{U}^{m,\ell}$, $\ell = 1,\dots,m-1$ satisfy the recursion:

$$\mathbb{U}_{ij}^{m,\ell} = \begin{cases} \mathbb{U}_{ij}^{m,\ell-1} & \text{if } j \le k+1-\ell \\ \delta_{ij}^{c}(\mathbb{U}^{m,\ell-1}) \cdot \dfrac{k+1-\ell}{x_j - x_{j-(k+1-\ell)}} & \text{if } j > k+1-\ell, \end{cases} \tag{B.16}$$

and where:

$$\mathbb{U}_{ij}^{m,m} = \begin{cases} \mathbb{U}_{ij}^{m,m-1} & \text{if } j \le m \\ \delta_{ij}^{c}(\mathbb{U}^{m,m-1}) & \text{if } j > m. \end{cases} \tag{B.17}$$

Similarly, (B.16), (B.17) are equivalent to (9.9), (9.10), the column-wise recursion in in Lemma 9.3 (again, the difference is that Lemma 9.3 is written in terms of the lower-right $(n-m)\times(n-m)$ submatrices). This establishes the result in Lemma 9.3.

Exchanging the order of scaled differencing with integration and differentiation. Now that we have shown how to explicitly write the entries of $\mathbb{F}^{\mathsf{T}}\mathbb{Q}\,\mathbb{F}$ via recursion, it remains to prove bandedness. To this end, for each $x \in [a,b]$, define $\mathbb{Q}^x \in \mathbb{R}^{n\times n}$ to have entries $\mathbb{Q}_{ij}^x = (D^m h_i^k)(x)(D^m h_j^k)(x)$, and note that by linearity of integration,

$$\mathbb{F}^{\mathsf{T}}\mathbb{Q}\,\mathbb{F} = \int_a^b \mathbb{F}^{\mathsf{T}}\mathbb{Q}^x\,\mathbb{F}\,dx,$$

where the integral on the right-hand side above is meant to be interpreted elementwise. Furthermore, defining $a^x \in \mathbb{R}^n$ to have entries $a^x_i = (D^m h^k_i)(x)$, we have $\mathbb{Q}^x = a^x(a^x)^\mathsf{T}$, and defining $b^x \in \mathbb{R}^n$ to have entries $b^x_i = h^k_i(x)$, note that by linearity of differentiation,

$$\mathbb{F}^\mathsf{T} a^x = D^m \mathbb{F}^\mathsf{T} b^x,$$

where again the derivative on the right-hand side is meant to be interpreted elementwise. This means that

$$\mathbb{F}^\mathsf{T} \mathbb{Q}^x \, \mathbb{F} = (D^m \mathbb{F}^\mathsf{T} b^x)(D^m \mathbb{F}^\mathsf{T} b^x)^\mathsf{T}.$$

By the same logic as that given above (see the development of (B.14), (B.15)), we can view $\mathbb{F}^\mathsf{T} b^x$ as the endpoint of an m-step recursion. First initialize $u^{x,0} = b^x$, and define for $\ell = 1, \dots, m-1$,

$$u^{x,\ell}_i = \begin{cases} u^{x,\ell-1}_i & \text{if } i \le k+1-\ell \\ (u^{x,\ell-1}_i - u^{x,\ell-1}_{i+1}) \cdot \dfrac{k+1-\ell}{x_i - x_{i-(k+1-\ell)}} & \text{if } i > k+1-\ell, \end{cases} \tag{B.18}$$

as well as

$$u^{x,m}_i = \begin{cases} u^{x,m-1}_i & \text{if } i \le m \\ u^{x,m-1}_i - u^{x,m-1}_{i+1} & \text{if } i > m. \end{cases} \tag{B.19}$$

Here, we set $u^{x,\ell}_{n+1} = 0$, $\ell = 1, \dots, m$, for convenience. Then as before, this recursion terminates at $u^{x,m} = \mathbb{F}^\mathsf{T} b^x$.

In what follows, we will show that

$$(D^m u^{x,m}_i)(D^m u^{x,m}_j) = 0, \quad \text{for } x \in [a,b] \text{ and } |i-j| > m. \tag{B.20}$$

Clearly this would imply that $(\mathbb{F}^\mathsf{T} \mathbb{Q}^x \, \mathbb{F})_{ij} = 0$ for $x \in [a,b]$ and $|i-j| > m$, and so $(\mathbb{F}^\mathsf{T} \mathbb{Q} \, \mathbb{F})_{ij} = 0$ for $|i-j| > m$; focusing on the lower-right submatrix of dimension $(n-m) \times (n-m)$, this would mean $(\mathbb{G}^\mathsf{T} \mathbb{M} \, \mathbb{G})_{ij} = (\mathbb{V}^m_n)_{ij} = 0$ for $|i-j| > m$, which is the claimed bandedness property of $\mathbb{V}^m_n$.

Proof of the bandedness property (B.20) for $i > k+1, j > k+1$. Consider $i > k+1$. At the first iteration of the recursion (B.18), (B.19), we get

$$u^{x,1}_i = (h^k_i(x) - h^k_{i+1}(x)) \cdot \frac{k}{x_i - x_{i-k}}, \tag{B.21}$$

where we set $h^k_{n+1} = 0$ for notational convenience. Next we present a helpful lemma, which is an application of the elementary result in Lemma B.1, on differences of Newton polynomials (recall this serves as the main driver behind the proof of Lemma 3.3). Since (B.22) is a direct consequence of (B.4) (more specifically, a direct consequence of the special case highlighted in (B.5)), we state the lemma without proof.

Lemma B.3. For any $k \geq 1$, the piecewise polynomials in the kth degree falling factorial basis, given in the second line of (1.5), satisfy for each $k+2 \leq i \leq n-1$,

$$h^k_i(x) - h^k_{i+1}(x) = h^{k-1}_i(x) \cdot \frac{x_i - x_{i-k}}{k}, \quad \text{for } x \notin (x_{i-1}, x_i]. \tag{B.22}$$

Fix $i \leq n-m$. Applying Lemma B.3 to (B.21), we see that for $x \notin (x_{i-1}, x_i]$, we have simply $u^{x,1}_i = h^{k-1}_i(x)$. By the same argument, for $x \notin (x_{i-1}, x_{i+1}]$,

$$\begin{aligned} u^{x,2}_i &= (u^{x,1}_i - u^{x,1}_{i+1}) \cdot \frac{k-1}{x_i - x_{i-(k-1)}} \\ &= (h^{k-1}_i(x) - h^{k-1}_{i+1}(x)) \cdot \frac{k-1}{x_i - x_{i-(k-1)}} \\ &= h^{k-2}_i(x). \end{aligned}$$

Iterating this over $u^{x,\ell}_i$, $\ell = 3, \ldots, m$, we get that for $x \notin (x_{i-1}, x_{i+m-1}]$,

$$\begin{aligned} u^{x,m}_i &= u^{x,m-1}_i - u^{x,m-1}_{i+1} \\ &= h^m_i(x) - h^m_{i+1}(x) \\ &= h^{m-1}_i(x) \cdot \frac{x_i - x_{i-m}}{m}. \end{aligned}$$

As $h^{m-1}_i = 0$ on $[a, x_{i-1}]$ and it is a polynomial of degree $m-1$ on $(x_{i-1}, b]$, we therefore conclude that $D^m u^{x,m}_i = 0$ for $x \notin (x_{i-1}, x_{i+m-1}]$.

For $i \geq n-m+1$, note that we can still argue $u^{x,m}_i = 0$ for $x \leq x_{i-1}$, as $u^{x,m}_i$ is just a linear combination of the evaluations $h^k_i(x), h^k_{i+1}(x), \ldots, h^k_n(x)$, each of which are zero. Thus, introducing the convenient notation $\bar{x}_i = x_i$ for $i \leq n-1$ and $\bar{x}_i = b$ for $i \geq n$, we can still write $D^m u^{x,m}_i = 0$ for $x \notin (x_{i-1}, \bar{x}_{i+m-1}]$.

Altogether, for $i > k+1, j > k+1$, the product $(D^m u^{x,m}_i)(D^m u^{x,m}_j)$ can only be nonzero if $x \notin (x_{i-1}, \bar{x}_{i+m-1}] \cap (x_{j-1}, \bar{x}_{j+m-1}]$, which can only happen (this intersection is only nonempty) if $|i-j| \le m$. This proves (B.20) for $i > k+1, j > k+1$.

Proof of the bandedness property (B.20) **for $\boldsymbol{i \le k+1, j > k+1}$.** Consider $i = k+1$. At the first iteration of the recursion (B.18), (B.19), we get

$$u^{x,1}_{k+1} = (h^k_{k+1}(x) - h^k_{k+2}(x)) \cdot \frac{k}{x_{k+1} - x_1}. \tag{B.23}$$

We give another helpful lemma, similar to Lemma B.3. As (B.24) is again a direct consequence of (B.4) from Lemma B.1 (indeed a direct consequence of the special case in (B.5)), we state the lemma without proof.

Lemma B.4. For any $k \ge 1$, the last of the pure polynomials and the first of the piecewise polynomials in the kth degree falling factorial basis, given in (1.5), satisfy

$$h^k_{k+1}(x) - h^k_{k+2}(x) = h^{k-1}_{k+1}(x) \cdot \frac{x_{k+1} - x_1}{k}, \quad \text{for } x > x_{k+1}. \tag{B.24}$$

Applying Lemma B.4 to (B.23), we see that for $x > x_{k+1}$, it holds that $u^{x,2}_{k+1} = h^{k-1}_{k+1}(x)$. Combined with our insights from the recursion for the case $i > k+1$ developed previously, at the next iteration we see that for $x > x_{k+2}$,

$$\begin{aligned} u^{x,2}_{k+1} &= (u^{x,1}_{k+1} - u^{x,1}_{k+2}) \cdot \frac{k-1}{x_{k+1} - x_2} \\ &= (h^{k-1}_i(x) - h^{k-1}_{i+1}(x)) \cdot \frac{k-1}{x_{k+1} - x_2} \\ &= h^{k-2}_{k+1}(x). \end{aligned}$$

Iterating this over $u^{x,\ell}_i$, $\ell = 3, \dots, m$, we get that for $x > x_{k+m}$,

$$\begin{aligned} u^{x,m}_{k+1} &= u^{x,m-1}_{k+1} - u^{x,m-1}_{k+2} \\ &= h^m_{k+1}(x) - h^m_{k+2}(x) \\ &= h^{m-1}_{k+1}(x) \cdot \frac{x_{k+1} - x_{k+1-m}}{m}. \end{aligned}$$

and as before, we conclude that $D^m u^{x,m}_{k+1} = 0$ for $x > x_{k+m}$.

For $i < k+1$, the same argument applies, but just lagged by some number of iterations (for $\ell = 1, \dots, k+1-i$, we stay at $u_i^{x,\ell} = h_i^k(x)$, then for $\ell = k+2-i$, we get $u_i^{x,\ell} = (h_i^k(x) - h_{i+1}^k(x)) \cdot (i-1)/(x_i - x_1)$, so Lemma B.4 can be applied, and so forth), which leads us to $D^m u_i^{x,m} = 0$ for $x > x_{i+m-1}$.

Finally, for $i \le k+1$ and $|i-j| > m$, we examine the product $(D^m u_i^{x,m})(D^m u_j^{x,m})$. As $|i-j| > m$, we must have either $j < m$ or $j > k+1$. For $j < m$, we have already shown $(\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F})_{ij} = 0$, and so for our ultimate purpose (of establishing (B.20) to establish bandedness of $\mathbb{F}^\mathsf{T} \mathbb{Q} \mathbb{F}$), we only need to consider the case $j > k+1$. But then (from our analysis in the last part) we know $(D^m u_j^{x,m}) = 0$ for $x \le x_{j-1}$, whereas (from our analysis in the current part) $(D^m u_i^{x,m}) = 0$ for $x > x_{i+m-1}$, and since $x_{j-1} > x_{i+m-1}$, we end up with $(D^m u_i^{x,m})(D^m u_j^{x,m}) = 0$ for all x. This establishes the desired property (B.20) over all i, j, and completes the proof of the theorem.

B.7 Proof of Lemma 9.4

To avoid unnecessary indexing difficulties, we will work directly on the entries of $\mathbb{Q}$, defined in (B.7), and then we will be able to read off the result for the entries of $\mathbb{M}$, defined in (9.6), by inspecting the lower-right submatrix of dimension $(n-m) \times (n-m)$. Fix $i \ge j$, with $i > 2m$. Applying integration by parts on each subinterval of $[a,b]$ in which the product $(D^m h_i^k)(D^m h_j^k)$ is continuous, we get

$$\int_a^b (D^m h_i^k)(x)(D^m h_j^k)(x)\, dx =$$

$$(D^m h_i^k)(x)(D^{m-1} h_j^k)(x)\Big|_{a, x_{j-1}, x_{i-1}}^{x_{j-1}, x_{i-1}, b} - \int_a^b (D^{m+1} h_i^k)(x)(D^{m-1} h_j^k)(x)\, dx,$$

where we use the notation

$$f(x)\Big|_{a_1,\dots,a_r}^{b_1,\dots,b_r} = \sum_{i=1}^r \big(f^-(b_i) - f^+(a_i)\big).$$

as well as $f^-(x) = \lim_{t \to x^-} f(t)$ and $f^+(x) = \lim_{t \to x^+} f(t)$. As h_i^k and h_j^k are supported on $(x_{i-1}, b]$ and $(x_{j-1}, b]$, respectively, so are there derivatives, and as $x_{i-1} \geq x_{j-1}$ (since $i \geq j$) the second to last display reduces to

$$\int_a^b (D^m h_i^k)(x)(D^m h_j^k)(x)\,dx = \\ (D^m h_i^k)(x)(D^{m-1} h_j^k)(x)\Big|_{x_{i-1}}^b - \int_a^b (D^{m+1} h_i^k)(x)(D^{m-1} h_j^k)(x)\,dx,$$

Applying integration by parts $m-2$ more times (and using $k = 2m-1$) yields

$$\begin{aligned} \int_a^b (D^m h_i^k)(x)(D^m h_j^k)(x)\,dx \\ &= \sum_{\ell=1}^{m-1} (-1)^{\ell-1} (D^{m+\ell-1} h_i^k)(x)(D^{m-\ell} h_j^k)(x)\Big|_{x_{i-1}}^b + \\ &\quad (-1)^{m-1} \int_a^b (D^k h_i^k)(x)(D h_j^k)(x)\,dx \\ &= \sum_{\ell=1}^{m-1} (-1)^{\ell-1} (D^{m+\ell-1} h_i^k)(x)(D^{m-\ell} h_j^k)(x)\Big|_{x_{i-1}}^b + \\ &\quad (-1)^{m-1} \big(h_j^k(b) - h_j^k(x_{i-1})\big), \end{aligned} \tag{B.25}$$

where in the second line we used $(D^k h_i^k)(x) = 1\{x > x_{i-1}\}$ and the fundamental theorem of calculus. The result for the case $i \leq 2m$ is similar, the only difference being that we apply integration by parts a total of $i-m-1$ (rather than $m-1$ times), giving

$$\int_a^b (D^m h_i^k)(x)(D^m h_j^k)(x)\,dx = \\ \sum_{\ell=1}^{i-m-1} (-1)^{\ell-1} (D^{m+\ell-1} h_i^k)(x)(D^{m-\ell} h_j^k)(x)\Big|_a^b + \\ (-1)^{i-m-1} \big(h_j^k(b) - h_j^k(a)\big). \tag{B.26}$$

Putting together (B.25), (B.26) establishes the desired result (9.11) (recalling that the latter is cast in terms of the lower-right $(n-m) \times (n-m)$ submatrix of $\mathbb{Q}$, and is hence given by replacing i, j with $i+m, j+m$, respectively).

B.8 Proof of Lemma 10.2

For $k = 0$ or $k = 1$, we can use elementary piecewise constant or continous piecewise linear interpolation. For $k = 0$, we set g to be the piecewise constant function that has knots in $x_{1:(n-1)}$, and $g(x_i) = f(x_i)$, $i = 1\ldots, n$; note clearly, $\mathrm{TV}(g) \le \mathrm{TV}(f)$. For $k = 1$, we again set g to be the continous piecewise linear function with knots in $x_{2:(n-1)}$, and $g(x_i) = f(x_i)$, $i = 1\ldots, n$; still clearly, $\mathrm{TV}(Dg) \le \mathrm{TV}(Df)$. This proves (10.4).

For $k \ge 2$, we can appeal to well-known approximation results for kth degree splines, for example, Theorem 6.20 of Schumaker (2007). First we construct a quasi-uniform partition from $x_{(k+1):(n-1)}$, call it $x^*_{1:r} \subseteq x_{(k+1):(n-1)}$, such that $\delta_n/2 \le \max_{i=1,\ldots,r-1} (y_{i+1} - y_i) \le 3\delta_n/2$, and an extended partition $y_{1:(r+2k+2)}$,

$$\begin{aligned} y_1 = \cdots = y_{k+1} = a, \\ y_{k+2} = x^*_1 < \cdots < y_{r+k+1} = x^*_r, \\ y_{r+k+2} = \cdots = y_{r+2k+2} = b. \end{aligned}$$

Now for each $\ell = k+1, \ldots, r+k+1$, define $I_\ell = [y_\ell, y_{\ell+1}]$ and $\bar{I}_\ell = [y_{\ell-k}, y_{\ell+k+1}]$. Then there exists a kth degree spline g with knots in $x^*_{1:r}$, such that, for any $d = 0, \ldots, k$, and a constant $b_k > 0$ that depends only on k,

$$\|D^d(f-g)\|_{L_\infty(\bar{I}_\ell)} \le b_k \delta_n^{k-d} \omega(D^k f; \delta_n)_{L_\infty(\bar{I}_\ell)}, \tag{B.27}$$

Here $\|h\|_{L_\infty(I)} = \sup_{x\in I} |f(x)|$ denotes the L_∞ norm of a function h an interval I, and

$$\omega(h; v)_{L_\infty(I)} = \sup_{x,y\in I,\, |x-y|\le v} |h(x) - h(y)|$$

denotes the modulus of continuity of h on I. Note that

$$\omega(D^k f; \delta_n)_{L_\infty(\bar{I}_\ell)} \le \mathrm{TV}(D^k f).$$

Thus setting $d = 0$ in (B.27), and taking a maximum over $\ell = k+1, \ldots, r+k+1$, we get $\|f - g\|_{L_\infty} \le b_k \delta_n^k \cdot \mathrm{TV}(D^k f)$. Further, the importance of the result in (B.27) is that it is *local* and hence allows us to make statements about total variation as well. Observe

$$\begin{aligned}
\mathrm{TV}(D^k g) &= \sum_{i=k+2}^{r+k+2} |D^k g(y_i) - D^k g(y_{i-1})| \\
&\leq \sum_{i=k+2}^{r+k+2} \Big(|D^k f(y_i) - D^k g(y_i)| + \\
&\qquad |D^k f(y_{i-1}) - D^k g(y_{i-1})| + |D^k f(y_i) - D^k f(y_{i-1})| \Big) \\
&\leq \underbrace{(2(k+2)b_k + 1)}_{a_k} \cdot \mathrm{TV}(D^k f),
\end{aligned}$$

In the last step above, we applied (B.27) with $d = k$, and the fact that each interval $\bar{I}_\ell$ can contain at most $k+2$ of the points y_i, $i = k+1, \ldots, r+k+2$. This proves (10.5).

B.9 Proof of Lemma 12.1

Observe that, by adding and subtracting y and expanding,

$$\|\hat{\theta}_a - \hat{\theta}_b\|_2^2 = (y - \hat{\theta}_a)^\mathsf{T}(\hat{\theta}_b - \hat{\theta}_a) + (y - \hat{\theta}_b)^\mathsf{T}(\hat{\theta}_a - \hat{\theta}_b). \tag{B.28}$$

By the stationarity condition for problem (12.5), we have $y - \hat{\theta}_a = \lambda_a \mathbb{A}\hat{\theta}_a$, so that

$$\begin{aligned}
(y - \hat{\theta}_a)^\mathsf{T}(\hat{\theta}_b - \hat{\theta}_a) &\leq \lambda_a \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_b - \lambda_a \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_a \\
&\leq \frac{1}{2}\lambda_a \hat{\theta}_b^\mathsf{T} \mathbb{A} \hat{\theta}_b - \frac{1}{2}\lambda_a \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_a,
\end{aligned}$$

where in the second line we used the inequality $u^\mathsf{T}\mathbb{A}v \leq u^\mathsf{T}\mathbb{A}u/2 + v^\mathsf{T}\mathbb{A}v/2$. By the same logic,

$$(y - \hat{\theta}_b)^\mathsf{T}(\hat{\theta}_a - \hat{\theta}_b) \leq \frac{1}{2}\lambda_b \hat{\theta}_a^\mathsf{T} \mathbb{B} \hat{\theta}_a - \frac{1}{2}\lambda_b \hat{\theta}_b^\mathsf{T} \mathbb{B} \hat{\theta}_b.$$

Applying the conclusion in the last two displays to (B.28),

$$\begin{aligned}
\|\hat{\theta}_a - \hat{\theta}_b\|_2^2 &\leq \frac{1}{2}\lambda_a \hat{\theta}_b^\mathsf{T} \mathbb{A} \hat{\theta}_b - \frac{1}{2}\lambda_a \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_a + \frac{1}{2}\lambda_b \hat{\theta}_a^\mathsf{T} \mathbb{B} \hat{\theta}_a - \frac{1}{2}\lambda_b \hat{\theta}_b^\mathsf{T} \mathbb{B} \hat{\theta}_b \\
&\leq \frac{1}{2}\sigma\lambda_a \hat{\theta}_b^\mathsf{T} \mathbb{B} \hat{\theta}_b - \frac{1}{2}\lambda_a \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_a + \frac{1}{2}(\lambda_b/\tau) \hat{\theta}_a^\mathsf{T} \mathbb{A} \hat{\theta}_a - \frac{1}{2}\lambda_b \hat{\theta}_b^\mathsf{T} \mathbb{B} \hat{\theta}_b,
\end{aligned}$$

where in the second line we twice used the spectral similarity property (12.4). The desired result follows by grouping terms.

B.10 Proof of Theorem 12.2

Note that

$$\begin{aligned} &\mathbb{K}_n^2, \mathbb{W}_n^2 \text{ are } (\sigma,\tau)\text{-spectrally-similar} \\ &\iff (\mathbb{K}_n^2)^{-1}, (\mathbb{W}_n^2)^{-1} \text{ are } (1/\sigma, 1/\tau)\text{-spectrally-similar} \\ &\iff \mathbb{W}_n^2(\mathbb{K}_n^2)^{-1}\mathbb{W}_n^2, \mathbb{W}_n^2 \text{ are } (1/\sigma, 1/\tau)\text{-spectrally-similar.} \end{aligned}$$

Set $\mathbb{A} = \mathbb{W}_n^2(\mathbb{K}_n^2)^{-1}\mathbb{W}_n^2$. From (2.11), we can see that

$$\mathbb{A}_{ij} = \begin{cases} \dfrac{x_{i+2} - x_i}{3} & \text{if } i = j \\ \dfrac{x_{i+1} - x_i}{6} & \text{if } i = j + 1. \end{cases}$$

Now define $a_i = (x_{i+2} - x_i)/3$ and $b_i = (x_{i+2} - x_{i+1})/6$, for $i = 1, \ldots, n-2$. Also denote $q_i = (x_{i+2} - x_i)/2$, for $i = 1, \ldots, n-2$. Fix $u \in \mathbb{R}^n$. For notational convenience, set $b_0 = u_0 = 0$ and $u_{n-1} = 0$. Then

$$\begin{aligned} u^{\mathsf{T}}\mathbb{A}u &= \sum_{i=1}^{n-2}\Big(a_i u_i^2 + b_{i-1}u_{i-1}u_i + b_i u_i u_{i+1}\Big) \\ &\le \sum_{i=1}^{n-2}\bigg(a_i u_i^2 + \frac{b_{i-1}}{2}(u_{i-1}^2 + u_i^2) + \frac{b_i}{2}(u_i^2 + u_{i+1}^2)\bigg) \\ &= \sum_{i=1}^{n-2}(a_i + b_{i-1} + b_i)u_i^2 \\ &= \sum_{i=1}^{n-2} q_i u_i^2 - \frac{x_2 - x_1}{6}u_1^2 - \frac{x_{n-1} - x_{n-2}}{6}u_{n-2}^2 \\ &\le \sum_{i=1}^{n-2} q_i u_i^2. \end{aligned}$$

In the second line above, we used $2st \le s^2 + t^2$, and in the fourth we used $a_i + b_{i-1} + b_i = q_i$, for $i = 1, \ldots, n-2$. This shows that we can take $1/\tau = 1$, that is, $\tau = 1$.

As for the other direction, using $2st \ge -s^2 - t^2$, we have

$$u^{\mathsf{T}}Wu \ge \sum_{i=1}^{n}\bigg(a_i u_i^2 - \frac{b_{i-1}}{2}(u_{i-1}^2 + u_i^2) - \frac{b_i}{2}(u_i^2 + u_{i+1}^2)\bigg)$$

$$
\begin{aligned}
&= \sum_{i=1}^{n-2} (a_i - b_{i-1} - b_i) u_i^2 \\
&= \frac{1}{2} \sum_{i=1}^{n-2} q_i u_i^2 + \frac{x_2 - x_1}{6} u_1^2 + \frac{x_{n-1} - x_{n-2}}{6} u_{n-2}^2 \\
&\geq \frac{1}{3} \sum_{i=1}^{n-2} q_i u_i^2,
\end{aligned}
$$

where in the third line we used the fact that $a_i - b_{i-1} - b_i = q_i/3$, for $i = 1, \ldots, n-2$. This shows that we can take $1/\sigma = 1/3$, that is, $\sigma = 3$, which completes the proof.

B.11 Proof of Lemma 12.4

To keep indexing simple in the current case of $m = 1$, we will compute the entries of the matrix $\mathbb{Q}$ in (B.7), then carry out the recursion (B.14)–(B.17), and the desired matrix $\mathbb{V}_n$ will be given be reading off the lower-right $(n-1) \times (n-1)$ submatrix of the result. Consider $i \geq j$. For $i \geq 3$, observe that

$$
\begin{aligned}
\mathbb{Q}_{ij} &= \int_a^b (Dh_i^1)(x)(Dh_j^1)(x)\, dx \\
&= \int_a^b 1\{x > x_{i-1}\}\, dx \\
&= b - x_{i-1}.
\end{aligned}
$$

Meanwhile, for $i = 2$, by a similar calculation, $\mathbb{Q}_{ij} = b - a$. Therefore, introducing the convenient notation $\bar{x}_i = x_i$ for $i \geq 3$ and $\bar{x}_i = a$ for $i = 2$, we get

$$
\mathbb{Q}_{ij} = b - \bar{x}_{i-1},
$$

for all $i \geq 2$. We know that the result of the recursion in (B.14)–(B.17) will be diagonal. As $m = 1$, this recursion reduces to simply (B.15), (B.17), which together give

$$
\begin{aligned}
\mathbb{U}_{ii}^{1,1} &= (\mathbb{Q}_{ii} - \mathbb{Q}_{i+1,i}) - (\mathbb{Q}_{i,i+1} - \mathbb{Q}_{i+1,i+1}) \\
&= ((b - \bar{x}_{i-1}) - (b - \bar{x}_i)) - ((b - \bar{x}_i) - (b - \bar{x}_i)) \\
&= \bar{x}_i - \bar{x}_{i-1}.
\end{aligned}
$$

This proves (12.15) (recalling that this is written in terms of $\mathbb{V}_n = \mathbb{V}^{1,1}$, the lower-right $(n-1) \times (n-1)$ submatrix of $\mathbb{U}^{1,1}$, and so for (12.15) we simply replace i with $i+1$).

C

B-Splines and Discrete B-Splines

C.1 B-Splines

Though the truncated power basis (2.5) is the simplest basis for splines, the *B-spline basis* is just as fundamental, as it was "there at the very beginning", appearing in Schoenberg's original paper on splines (Schoenberg, 1946a). Here we are quoting Boor (1976), who gives a masterful survey of the history and properties of B-splines (and points out that the name "B-spline" is derived from Schoenberg's use of the term "basic spline", to further advocate for the idea that B-splines can be seen as *the* basis for splines). A key feature of B-splines is that they have local support, and are thus extremely useful for computational purposes.

Peano representation. There are different ways to construct B-splines; here we cover a construction based on what is called the *Peano representation* for B-splines (see, for example, Theorem 4.23 in Schumaker (2007)). If f is a $k+1$ times differentiable function f on an interval $[a, b]$ (and its $(k+1)$st derivative is integrable), then by Taylor expansion

$$f(z) = \sum_{i=0}^{k} \frac{1}{i!}(D^i f)(a)(z-a)^i + \int_a^z \frac{1}{k!}(D^{k+1} f)(x)(z-x)^k \, dx.$$

Note that we can rewrite this as

$$f(z) = \sum_{i=0}^{k} \frac{1}{i!}(D^i f)(a)(z-a)^i + \int_a^b \frac{1}{k!}(D^{k+1} f)(x)(z-x)_+^k \, dx.$$

Next we take a divided difference with respect to arbitrary centers $z_1, \ldots, z_{k+2} \in [a,b]$, where we assume without a loss of generality that $z_1 < \cdots < z_{k+2}$. Then by linearity we can exchange divided differentiation with integration, yielding

$$k! \cdot f[z_1, \ldots, z_{k+2}] = \int_a^b (D^{k+1} f)(x) \underbrace{(\cdot - x)_+^k [z_1, \ldots, z_{k+2}]}_{P^k(x; z_{1:(k+2)})} \, dx, \quad \text{(C.1)}$$

where we have also used the fact that a $(k+1)$st order divided difference (with respect to any $k+2$ centers) of a kth degree polynomial is zero (for example, see (4.6)), and lastly, we multiplied both sides by $k!$. To be clear, the notation $(\cdot - x)_+^k [z_1, \ldots, z_{k+2}]$ means that we are taking the divided difference of the function $z \mapsto (z-x)_+^k$ with respect to centers $z_1, \ldots, z_{k+2}$.

B-spline definition. The result in (C.1) shows that the $(k+1)$st divided difference of any (smooth enough) function f can be written as a weighted average of its $(k+1)$st derivative, in a local neighborhood around the corresponding centers, where the weighting is given by a universal kernel $P^k(\cdot; z_{1:(k+2)})$ (that does not depend on f), which is called the *Peano kernel* formulation for the B-spline; to be explicit, this is

$$P^k(x; z_{1:(k+2)}) = (\cdot - x)_+^k [z_1, \ldots, z_{k+2}]. \quad \text{(C.2)}$$

Since

$$(z-x)_+^k - (-1)^{k+1}(x-z)_+^k = (z-x)^k,$$

and any $(k+1)$st order divided difference of the kth degree polynomial $z \mapsto (z-x)^k$ is zero, we can rewrite the above (C.2) as:

$$P^k(x; z_{1:(k+2)}) = (-1)^{k+1}(x - \cdot)_+^k [z_1, \ldots, z_{k+2}]. \quad \text{(C.3)}$$

The function $P^k(\cdot; z_{1:(k+2)})$ is called a kth degree *B-spline* with knots $z_{1:(k+2)}$. It is a linear combination of kth degree truncated power functions and is hence indeed a kth degree spline.

It is often more convenient to deal with the *normalized B-spline*:

$$M^k(x; z_{1:(k+2)}) = (-1)^{k+1}(z_{k+2} - z_1)(x - \cdot)_+^k[z_1, \dots, z_{k+2}]. \tag{C.4}$$

It is easy to show that

$$M^k(\cdot; z_{1:(k+2)}) \text{ is supported on } [z_1, z_{k+2}], \text{ and} \\ M^k(x; z_{1:(k+2)}) > 0 \text{ for } x \in (z_1, z_{k+2}). \tag{C.5}$$

To see the support result, note that for $x > z_{k+2}$, we are taking a divided difference of all zeros, which of course zero, and for $x < z_1$, we are taking a $(k+1)$st order divided difference of a polynomial of degree k, which is again zero. To see the positivity result, we can, for example, appeal to induction on k and the recursion to come later in (C.8).

B-spline basis. To build a local basis for $\mathcal{S}^k(t_{1:r}, [a, b])$, the space of kth degree splines with knots $t_{1:r}$, where we assume $a < t_1 < \cdots < t_r < b$, we first define boundary knots

$$t_{-k} < \cdots < t_{-1} < t_0 = a, \quad \text{and} \quad b = t_{r+1} < t_{r+2} < \cdots < t_{r+k+1}.$$

(Any such values for $t_{-k}, \dots, t_0$ and $t_{r+1}, \dots, t_{r+k+1}$ will suffice to produce a basis; in fact, setting $t_{-k} = \cdots = t_0$ and $t_{r+1} = \cdots = t_{r+k+1}$ would suffice, though this would require us to understand how to properly interpret divided differences with repeated centers; as in Definition 2.49 of Schumaker (2007).) We then define the normalized B-spline basis M_j^k, $j = 1, \dots, r+k+1$ for $\mathcal{S}^k(t_{1:r}, [a, b])$ by

$$M_j^k = M^k(\cdot; t_{(j-k-1):j})\Big|_{[a,b]}, \quad j = 1, \dots, r+k+1. \tag{C.6}$$

It is clear that each M_j^k, $j = 1, \dots, r+k+1$ is a kth degree spline with knots in $t_{1:r}$; hence to verify that they are a basis for $\mathcal{S}^k(t_{1:r}, [a, b])$, we only need to show their linear independence, which is straightforward using the structure of their supports (for example, see Theorem 4.18 of Schumaker (2007)).

For concreteness, we note that the 0th degree normalized B-splines basis for $\mathcal{S}^0(t_{1:r}, [a, b])$ is simply

$$M_j^0 = 1_{I_j}, \quad j = 1, \dots, r+1. \tag{C.7}$$

Here $I_0 = [t_0, t_1]$ and $I_i = (t_i, t_{i+1}]$, $i = 1, \dots, r$, and we use $t_{r+1} = b$ for notational convenience. We note that this particular choice for the half-open intervals (left- versus right-side open) is arbitrary, but consistent with our definition of the truncated power basis (2.5) when $k = 0$. Figure C.1 shows example normalized B-splines of degrees 0 through 3.

Recursive formulation. B-splines satisfy a recursion relation that can be seen directly from the recursive nature of divided differences: for any $k \geq 1$ and centers $z_1 < \cdots < z_{k+2}$,

$$\begin{aligned}
&(x - \cdot)_+^k[z_1, \dots, z_{k+2}] \\
&= \frac{(x - \cdot)_+^k[z_2, \dots, z_{k+2}] - (x - \cdot)_+^k[z_1, \dots, z_{k+1}]}{z_{k+2} - z_1} \\
&= \frac{(x - z_{k+2})(x - \cdot)_+^{k-1}[z_2, \dots, z_{k+2}] - (x - z_1)(x - \cdot)_+^{k-1}[z_1, \dots, z_{k+1}]}{z_{k+2} - z_1},
\end{aligned}$$

where in the second line we applied the Leibniz rule for divided differences (for example, Theorem 2.52 of Schumaker (2007)), $fg[z_1, \dots, z_{k+1}] = \sum_{i=1}^{k+1} f[z_1, \dots, z_i] g[z_i, \dots, z_{k+1}]$, to conclude that

$$\begin{aligned}
(x - \cdot)_+^k[z_1, \dots, z_{k+1}] &= (x - z_1) \cdot (x - \cdot)_+^{k-1}[z_1, \dots, z_{k+1}] \\
(x - \cdot)_+^k[z_2, \dots, z_{k+2}] &= (x - \cdot)_+^{k-1}[z_2, \dots, z_{k+2}] \cdot (x - z_{k+2}).
\end{aligned}$$

Translating the above recursion over to normalized B-splines, we get

$$\begin{aligned}
M^k(x; z_{1:(k+2)}) = \frac{x - z_1}{z_{k+1} - z_1} \cdot M^{k-1}(x; z_{1:(k+1)}) &+ \\
\frac{z_{k+2} - x}{z_{k+2} - z_2} \cdot M^{k-1}(x; z_{2:(k+2)}),& \qquad \text{(C.8)}
\end{aligned}$$

which means that for the normalized basis,

$$\begin{aligned}
M_j^k(x) = \frac{x - t_{j-k-1}}{t_{j-1} - t_{j-k-1}} \cdot M_{j-1}^{k-1}(x) &+ \\
\frac{t_j - x}{t_j - t_{j-k}} \cdot M_j^{k-1}(x), \quad j = 1, \dots, r + k + 1.& \qquad \text{(C.9)}
\end{aligned}$$

Above, we naturally interpret $M_0^{k-1} = M^{k-1}(\cdot; t_{-k:0})|_{[a,b]}$ and $M_{r+k+1}^{k-1} = M^{k-1}(\cdot; t_{(r+1):(r+k+1)})|_{[a,b]}$.

The above recursions are very important, both for verifying numerous properties of B-splines and for computational purposes. In fact, many authors prefer to use recursion to define a B-spline basis in the first place: they start with (C.7) for $k = 0$, and then invoke (C.9) for all $k \geq 1$.

C.2 Discrete B-splines

Here we will assume the design points are evenly-spaced, taking the form $[a, b]_v = \{a, a + v, \dots, b\}$ for $v > 0$ and $b = a + Nv$. As covered in Chapter 8.5 of Schumaker (2007), in this evenly-spaced case, discrete B-splines can be developed in a similar fashion to B-splines. Below we will jump directly into defining the discrete B-spline, which is at face value just a small variation on the definition of the usual B-spline given above. Chapter 8.5 of Schumaker (2007) develops several properties for discrete B-splines (for evenly-spaced design points)—such as a Peano kernel result for the discrete B-spline, with respect to a discrete integral—that we do not cover here, for simplicity.

Discrete B-spline definition. Let $z_{1:(k+2)} \subseteq [a, b]_v$. Assume without a loss of generality that $z_1 < \cdots < z_{k+2}$, and also $z_{k+2} \leq b - kv$. We define the kth degree *discrete B-spline* or DB-spline with knots $z_1, \dots, z_{k+2}$ by

$$U^k(x; z_{1:(k+2)}) = \Big((\cdot - x)^{k,v} \cdot 1\{\cdot > x\} \Big)[z_1, \dots, z_{k+2}], \tag{C.10}$$

where now we denote by $(z)^{k,v} = z(z + v) \cdots (z + (k - 1)v)$ the rising factorial polynomial of degree k with gap v, which we take to be equal to 1 when $k = 0$. To be clear, the notation $((\cdot - x)^{k,v} \cdot 1\{\cdot > x\})[z_1, \dots, z_{k+2}]$ means that we are taking the divided difference of the function $z \mapsto (z - x)^{k,v} \cdot 1\{z > x\}$ with respect to the centers $z_1, \dots, z_{k+2}$. Since

$$(z - x)^{k,v} \cdot 1\{z > x\} - (-1)^{k+1}(x - z)_{k,v} \cdot 1\{x > z\} = (z - x)^{k,v},$$

and any $(k + 1)$st order divided difference of the kth degree polynomial $z \mapsto (z - x)^{k,v}$ is zero, we can equivalently rewrite (C.10) as:

$$U^k(x; z_{1:(k+2)}) = (-1)^{k+1} \Big((x - \cdot)_{k,v} \cdot 1\{x > \cdot\} \Big)[z_1, \dots, z_{k+2}]. \tag{C.11}$$

We see (C.11) is just as in the usual B-spline definition (C.3), but with a truncated falling factorial polynomial instead of a truncated power function. Also, note $U^k(\cdot; z_{1:(k+2)})$ is a linear combination of kth degree truncated falling factorial polynomials and is hence a kth degree discrete spline.

As before, it is convenient to define the *normalized discrete B-spline* or normalized DB-spline:

$$V^k(x; z_{1:(k+2)}) = (-1)^{k+1}(z_{k+2} - z_1)\Big((x - \cdot)_{k,v} \cdot 1\{x > \cdot\}\Big)[z_1, \dots, z_{k+2}]. \tag{C.12}$$

We must emphasize that

$$V^k(x; z_{1:(k+2)}) = M^k(x; z_{1:(k+2)}) \quad \text{for } k = 0 \text{ or } k = 1$$

(and the same for the unnormalized versions). This should not be a surprise, as discrete splines are themselves exactly splines for degrees $k = 0$ and $k = 1$. Back to a general degree $k \geq 0$, it is easy to show that

$$V^k(\cdot; z_{1:(k+2)}) \text{ is supported on } [z_1, z_{k+2}]. \tag{C.13}$$

Curiously, $V^k(\cdot; z_{1:(k+2)})$ is no longer positive on the whole interval (z_1, z_{k+2}): for $k \geq 2$, it has a negative "ripple" close to the leftmost knot z_1. This is more pronounced when the knots are closer together (separated by fewer design points), see Figure C.1.

Discrete B-spline basis. To develop a local basis for $\mathcal{DS}^k_v(t_{1:r}, [a,b]_v)$, the space of kth degree discrete splines with knots in $t_{1:r}$, where $a < t_1 < \cdots < t_r < b$, and also $t_{1:r} \subseteq [a,b]_v$ and $t_r \leq b - kv$, we first define boundary knots

$$t_{-k} < \cdots < t_{-1} < t_0 = a, \quad \text{and} \quad b = t_{r+1} < t_{r+2} < \cdots < t_{r+k+1},$$

as before. We then define the normalized discrete B-spline basis V^k_j, $j = 1, \dots, r+k+1$ for $\mathcal{DS}^k_v(t_{1:r}, [a,b]_v)$ by

$$V^k_j = V^k(\cdot; t_{(j-k-1):j})\Big|_{[a,b]}, \quad j = 1, \dots, r+k+1. \tag{C.14}$$

It is clear that each V^k_j, $j = 1, \dots, r+k+1$ is a kth degree discrete spline with knots in $t_{1:r}$; hence to verify that they form a basis for

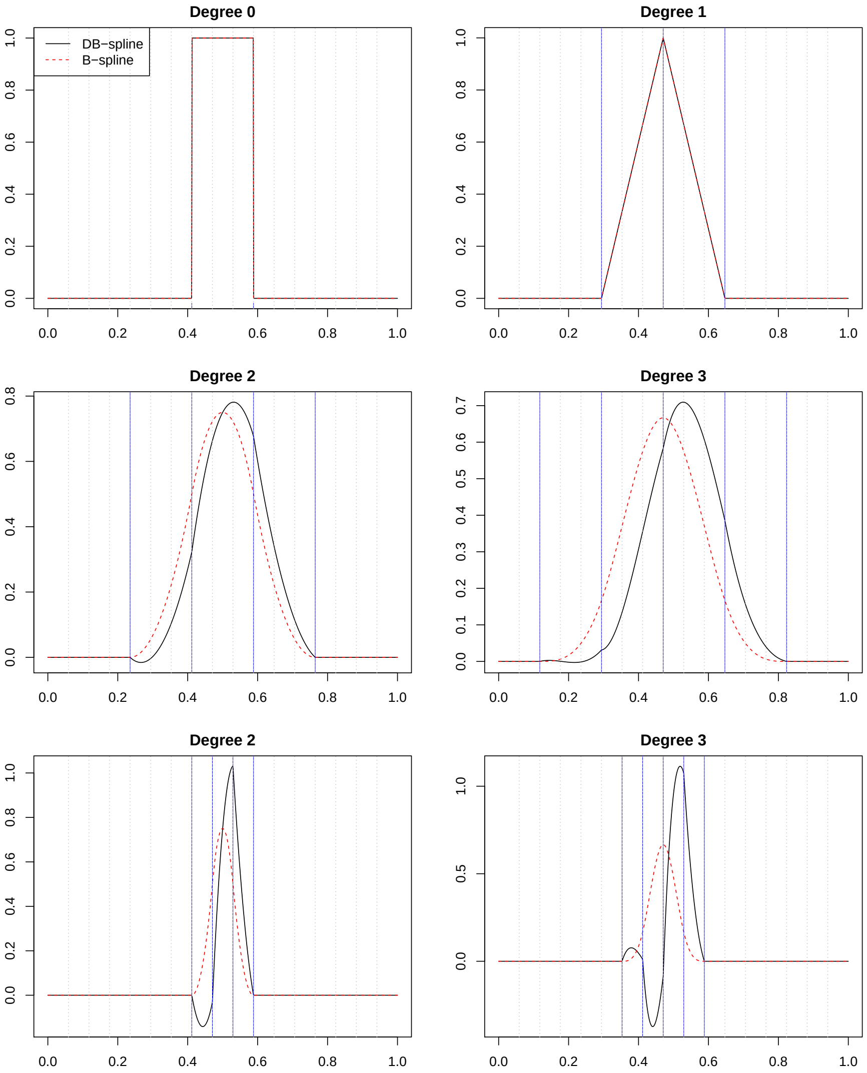


Figure C.1: Normalized DB-splines in black, and normalized B-splines in dashed red, of degrees 0 through 3. In each example, the $n = 16$ design points are evenly-spaced between 0 and 1, and marked by dotted vertical lines. The knot points are marked by blue vertical lines (except for $k = 0$, as here these would obscure the B-splines, so in this case we use small blue ticks on the horizontal axis). In the bottom row, the knots are closer together; we can see that the DB-splines of degrees 2 and 3 have negative "ripples" near their leftmost knots, which is much more noticeable when the knot points are closer together.

$\mathcal{DS}^k_n(t_{1:r}, [a,b])$, we only need to show their linear independence, which follows from similar arguments to the result for the usual B-splines (see also Theorem 8.55 of Schumaker (2007)).

Recursive formulation. To derive a recursion for discrete B-splines, we proceed as in the usual B-spline case, using the recursion that underlies divided differences: for any $k \geq 1$ and centers $z_1 < \cdots < z_{k+2}$ (such that $z_{1:(k+2)} \subseteq [a,b]_v$ and $z_{k+2} \leq b - kv$),

$$\begin{aligned}
&\Big((x-\cdot)_{k,v} \cdot 1\{x > \cdot\} \Big)[z_1, \ldots, z_{k+2}] \\
&= \bigg(((x-\cdot)_{k,v} \cdot 1\{x > \cdot\})[z_2, \ldots, z_{k+2}] - \\
&\qquad ((x-\cdot)_{k,v} \cdot 1\{x > \cdot\})[z_1, \ldots, z_{k+1}] \bigg) / (z_{k+2} - z_1) \\
&= \bigg((x - z_{k+2} - (k-1)v) \cdot ((x-\cdot)_{k-1,v} \cdot 1\{x > \cdot\})[z_2, \ldots, z_{k+2}] - \\
&\qquad (x - z_1 - (k-1)v) \cdot ((x-\cdot)_{k-1,v} \cdot 1\{x > \cdot\})[z_1, \ldots, z_{k+1}] \bigg) / \\
&\qquad (z_{k+2} - z_1),
\end{aligned}$$

where as before, in the second line, we applied the Leibniz rule for divided differences to conclude

$$\begin{aligned}
&\Big((x-\cdot)_{k,v} \cdot 1\{x > \cdot\} \Big)[z_1, \ldots, z_{k+1}] = \\
&\qquad\qquad (x - z_1 - (k-1)v) \cdot \Big((x-\cdot)_{k-1,v} \cdot 1\{x > \cdot\} \Big)[z_1, \ldots, z_{k+1}],
\end{aligned}$$

and

$$\begin{aligned}
&\Big((x-\cdot)_{k,v} \cdot 1\{x > \cdot\} \Big)[z_2, \ldots, z_{k+2}] = \\
&\qquad\qquad \Big((x-\cdot)_{k-1,v} \cdot 1\{x > \cdot\} \Big)[z_1, \ldots, z_{k+1}] \cdot (x - z_{k+2} - (k-1)v).
\end{aligned}$$

Translating the above recursion over normalized DB-splines, we get

$$\begin{aligned}
V^k(x; z_{1:(k+2)}) = \frac{x - z_1 - (k-1)v}{z_{k+1} - z_1} &\cdot V^{k-1}(x; z_{1:(k+1)}) + \\
\frac{z_{k+2} + (k-1)v - x}{z_{k+2} - z_2} &\cdot V^{k-1}(x; z_{2:(k+2)}), \quad \text{(C.15)}
\end{aligned}$$

which means that for the normalized basis,

$$\begin{aligned} V_j^k(x) = \frac{x - t_{j-k-1} - (k-1)v}{t_{j-1} - t_{j-k-1}} \cdot V_{j-1}^{k-1}(x) + \\ \frac{t_j + (k-1)v - x}{t_j - t_{j-k}} \cdot V_j^{k-1}(x), \quad j = 1, \dots, r+k+1. \end{aligned} \tag{C.16}$$

Above, we naturally interpret $V_0^{k-1} = V^{k-1}(\cdot; t_{-k:0})|_{[a,b]}$ and $V_{r+k+1}^{k-1} = V^{k-1}(\cdot; t_{(r+1):(r+k+1)})|_{[a,b]}$.

D

Fast Matrix Multiplication

We recall the details of the algorithms from Wang *et al.* (2014) for fast multiplication by $\mathbb{H}_n^k$, $(\mathbb{H}_n^k)^{-1}$, $(\mathbb{H}_n^k)^{\mathsf{T}}$, $(\mathbb{H}_n^k)^{-\mathsf{T}}$, in Algorithms 1–4. In each case, multiplication takes $O(nk)$ operations (at most $4nk$ operations), and is done in-place (no new memory required). We use cumsum to denote the cumulative sum operator, $\mathrm{cumsum}(v) = (v_1, v_1 + v_2, \ldots, v_1 + \cdots + v_n)$, for $v \in \mathbb{R}^n$, and diff for the pairwise difference operator, $\mathrm{diff}(v) = (v_2 - v_1, v_3 - v_2, \ldots, v_n - v_{n-1})$. We also use rev for the reverse operator, $\mathrm{rev}(v) = (v_n, \ldots, v_1)$, and $\odot$ for elementwise multiplication between vectors.

Algorithm 1 Multiplication by $\mathbb{H}_n^k$

```
Input: Integer degree $k \geq 0$, design points $x_{1:n}$ (assumed in sorted order), vector to be multiplied $v \in \mathbb{R}^n$.
Output: $v$ is overwritten by $\mathbb{H}_n^k v$.
for $i = k$ to $0$ do
    $v_{(i+1):n} = \mathrm{cumsum}(v_{(i+1):n})$
    if $i \neq 0$ then
        $v_{(i+1):n} = v_{(i+1):n} \odot \frac{x_{(i+1):n} - x_{1:(n-i)}}{i}$
    end if
end for
Return $v$.
```

Algorithm 2 Multiplication by $(\mathbb{H}_n^k)^{-1}$

```
Input: Integer degree $k \geq 0$, design points $x_{1:n}$ (assumed in sorted order), vector to be multiplied $v \in \mathbb{R}^n$.
Output: $v$ is overwritten by $(\mathbb{H}_n^k)^{-1}v$.
for $i = 0$ to $k$ do
  if $i \neq 0$ then
    $v_{(i+1):n} = v_{(i+1):n} \odot \frac{i}{x_{(i+1):n} - x_{1:(n-i)}}$
  end if
  $v_{(i+2):n} = \mathrm{diff}(v_{(i+1):n})$
end for
Return $v$.
```

Algorithm 3 Multiplication by $(\mathbb{H}_n^k)^{\mathsf{T}}$

```
Input: Integer degree $k \geq 0$, design points $x_{1:n}$ (assumed in sorted order), vector to be multiplied $v \in \mathbb{R}^n$.
Output: $v$ is overwritten by $(\mathbb{H}_n^k)^{-1}v$.
for $i = 0$ to $k$ do
  if $i \neq 0$ then
    $v_{(i+1):n} = v_{(i+1):n} \odot \frac{x_{(i+1):n} - x_{1:(n-i)}}{i}$
  end if
  $v_{(i+1):n} = \mathrm{rev}(\mathrm{cumsum}(\mathrm{rev}(v_{(i+1):n})))$
end for
Return $v$.
```

Algorithm 4 Multiplication by $(\mathbb{H}_n^k)^{-\mathsf{T}}$

```
Input: Integer degree $k \geq 0$, design points $x_{1:n}$ (assumed in sorted order), vector to be multiplied $v \in \mathbb{R}^n$.
Output: $v$ is overwritten by $(\mathbb{H}_n^k)^{-\mathsf{T}}v$.
for $i = k$ to $0$ do
  $v_{(i+1):n-1} = \mathrm{rev}(\mathrm{diff}(\mathrm{rev}(v_{(i+1):n})))$
  if $i \neq 0$ then
    $v_{(i+1):n} \odot \frac{i}{x_{(i+1):n} - x_{1:(n-i)}}$
  end if
end for
Return $v$.
```

References

Arnold, T. and R. J. Tibshirani. (2016). "Efficient implementations of the generalized lasso dual path algorithm". *Journal of Computational and Graphical Statistics*. 25(1): 1–27.

Batson, J., D. Spielman, N. Srivastava, and S.-H. Teng. (2013). "Spectral sparsification of graphs: theory and algorithms". *Communications of the ACM*. 56(8): 87–94.

Bohlmann, G. (1899). "Ein Ausgleichungsproblem". *Nachrichten von der Gesellschaft der Wissenschaften zu Gottingen, Mathematisch-Physikalische Klasse*: 260–271.

Boor, C. de. (1976). "Splines as linear combinations of B-splines". In: *Approximation Theory II*. Ed. by G. G. Lorentz, C. K. Chui, and L. L. Schumaker. Academic Press. 1–47.

Boor, C. de. (1978). *A Practical Guide to Splines*. Springer.

Boor, C. de. (2005). "Divided differences".

Chen, S., D. L. Donoho, and M. Saunders. (1998). "Atomic decomposition for basis pursuit". *SIAM Journal on Scientific Computing*. 20(1): 33–61.

Craven, P. and G. Wahba. (1978). "Smoothing noisy data with spline functions". *Numerische Mathematik*. 31(4): 377–403.

DeVore, R. and G. Lorentz. (1993). *Constructive Approximation*. Springer.

Donoho, D. L. and I. M. Johnstone. (1998). "Minimax estimation via wavelet shrinkage". *Annals of Statistics*. 26(8): 879–921.

Friedman, J., T. Hastie, H. Hoefling, and R. Tibshirani. (2007). "Pathwise coordinate optimization". *Annals of Applied Statistics*. 1(2): 302–332.

Geer, S. van de. (2000). *Empirical Processes in M-Estimation*. Cambdrige University Press.

Green, P. J. and B. W. Silverman. (1993). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall/CRC Press.

Greville, T. N. E. (1944). "The general theory of osculatory interpolation". *Transactions of the Acturial Society of America*. 45(112): 202–265.

Henderson, R. (1924). "A new method of graduation". *Transactions of the Actuarial Society of America*. 25: 29–53.

Hodrick, R. J. and E. C. Prescott. (1981). "Postwar U.S. Business Cycles: An Empirical Investigation".

Hodrick, R. J. and E. C. Prescott. (1997). "Postwar U.S. Business Cycles: An Empirical Investigation". *Journal of Money, Credit, and Banking*. 29(1): 1–16.

Johnson, N. (2013). "A dynamic programming algorithm for the fused lasso and L_0-segmentation". *Journal of Computational and Graphical Statistics*. 22(2): 246–260.

Kim, S.-J., K. Koh, S. Boyd, and D. Gorinevsky. (2009). "ℓ_1 trend filtering". *SIAM Review*. 51(2): 339–360.

Koenker, R., P. Ng, and S. Portnoy. (1994). "Quantile smoothing splines". *Biometrika*. 81(4): 673–680.

Lyche, T. (1975). "Discrete polynomial spline approximation methods". In: *Spline Functions*. Ed. by K. Bohmer, G. Meinardus, and W. Schempp. Springer. 144–176.

Mammen, E. and S. van de Geer. (1997). "Locally apadtive regression splines". *Annals of Statistics*. 25(1): 387–413.

Mangasarian, O. L. and L. L. Schumaker. (1971). "Discrete splines via mathematical programming". *SIAM Journal on Control*. 9(2): 174–183.

Mangasarian, O. L. and L. L. Schumaker. (1973). "Best summation formulae and discrete splines". *SIAM Journal on Numerical Analysis*. 10(3): 448–459.

Newton, I. (1687). *Philosophiae Naturalis Principia Mathematica.*

Newton, I. (1711). *Methodus Differentialis.*

Ramdas, A. and R. J. Tibshirani. (2016). "Fast and Flexible ADMM Algorithms for Trend Filtering". *Journal of Computational and Graphical Statistics*. 25(3): 839–858.

Reinsch, C. H. (1967). "Smoothing by spline functions". *Numerische Mathematik*. 10(3): 177–183.

Rudin, L. I., S. Osher, and E. Faterni. (1992). "Nonlinear total variation based noise removal algorithms". *Physica D: Nonlinear Phenomena*. 60(1): 259–268.

Sadhanala, V. and R. J. Tibshirani. (2019). "Additive models via trend filtering". *Annals of Statistics*. 47(6): 3032–3068.

Sadhanala, V., Y.-X. Wang, and R. J. Tibshirani. (2016). "Graph Sparsification Approaches for Laplacian Smoothing". *International Conference on Artificial Intelligence and Statistics*. 19.

Schoenberg, I. J. (1946a). "Contributions to the problem of approximation of equidistant data by analytic functions, Part A: on the problem of smoothing of graduation, a first class of analytic approximation formulae". *Quarterly of Applied Mathematics*. 4(1): 45–99.

Schoenberg, I. J. (1946b). "Contributions to the problem of approximation of equidistant data by analytic functions, Part B: on the problem of smoothing of graduation, a second class of analytic approximation formulae". *Quarterly of Applied Mathematics*. 4(2): 112–141.

Schoenberg, I. J. (1964). "Spline functions and the problem of graduation". *Proceeding of the National Academy of Sciences*. 52(4): 947–950.

Schuette, D. R. (1978). "A linear programming approach to graduation". *Transactions of Society of Actuaries*. 30.

Schumaker, L. L. (1973). "Constructive aspects of discrete polynomial spline functions". In: *Approximation Theory*. Ed. by G. G. Lorentz. Academic Press. 469–476.

Schumaker, L. L. (2007). *Spline Functions: Basic Theory*. Cambridge University Press.

Steidl, G., S. Didas, and J. Neumann. (2006). "Splines in higher order TV regularization". *International Journal of Computer Vision*. 70(3): 214–255.

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". *Journal of the Royal Statistical Society: Series B*. 58(1): 267–288.

Tibshirani, R., M. Saunders, S. Rosset, J. Zhu, and K. Knight. (2005). "Sparsity and smoothness via the fused lasso". *Journal of the Royal Statistical Society: Series B*. 67(1): 91–108.

Tibshirani, R. J. (2014). "Adaptive piecewise polynomial estimation via trend filtering". *Annals of Statistics*. 42(1): 285–323.

Tibshirani, R. J. and J. Taylor. (2011). "The Solution Path of the Generalized Lasso". *Annals of Statistics*. 39(3): 1335–1371.

Tibshirani, R. J. and J. Taylor. (2012). "Degrees of Freedom in Lasso Problems". *Annals of Statistics*. 40(2): 1198–1232.

Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer.

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM.

Wang, Y.-X., A. Smola, and R. J. Tibshirani. (2014). "The falling factorial basis and its statistical applications". *International Conference on Machine Learning*. 31.

Whittaker, E. T. (1923). "On a new method of graduation". *Proceedings of the Edinburgh Mathematical Society*. 41: 63–73.

Whittaker, E. T. and G. Robinson. (1924). *The Calculus of Observations: A Treatise on Numerical Mathematics*. Blackie and Son, Ltd.