Functional Chemometrics:

Automated Spectral Smoothing with Spatially Adaptive Splines

by

Philip M. Fernandes (B.Sc. Eng.)

A thesis submitted to the Department of Chemical Engineering
In conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
September 2012

Abstract

Functional data analysis (FDA) is a demonstrably effective, practical, and powerful method of data analysis, yet it remains virtually unheard of outside of academic circles and has almost no exposure to industry. FDA adds to the milieu of statistical methods by treating functions of one or more independent variables as data objects, analogous to the way in which discrete points are the data objects we are familiar with in conventional statistics.

The first step in functional analysis is to "functionalize" the data, or convert discrete points into a system represented most times by continuous functions. Choosing the type of functions to use is data-dependent and often straightforward – for example, Fourier series lend themselves well to periodic systems, while splines offer great flexibility in approximating more irregular trends, such as chemical spectra.

This work explores the question of how B-splines can be rapidly and reliably used to denoised infrared chemical spectra, a difficult problem not only because of the many parameters involved in generating a spline fit, but also due to the disparate nature of spectra in terms of shape and noise intensity. Automated selection of spline parameters is required to support high-throughput analysis, and the heteroscedastic nature of such spectra presents challenges for existing techniques.

The heuristic knot placement algorithm of Li et al. (2005) for 1D object contours is extended to spectral fitting by optimizing the denoising step for a range of spectral types and signal/noise ratios, using the following criteria: robustness to types of spectra and noise conditions, parsimony of knots, low computational demand, and ease of implementation in high-throughput settings. Pareto-optimal filter configurations are determined using simulated data from factorial experimental designs. The improved heuristic algorithm uses wavelet transforms and provides improved performance in robustness, parsimony of knots and the quality of functional regression models used to correlate real spectral data with chemical composition. In practical applications, functional principal component regression models yielded similar or significantly improved results when compared with their discrete partial least squares counterparts.

Acknowledgements

I am indebted to my supervisor, Dr. Jim McLellan, for taking me on as a Master's student in March of 2011, and cannot find the words to express my gratitude for his guidance, mentorship, and friendship. Thank you.

To Dan Krasznai for recommending Dr. Paul Nomikos's multivariate statistics course that got me on this road and for his work in PCA and PLS that was invaluable to my research. To my colleagues at Queen's and Novelis for making this a great experience, and most of all to my brother and parents for their endless support and motivation. I am very thankful to Queen's University, NSERC, OGSP, and MITACS for financial support.

Philip M. Fernandes

September 2012

"Belief? The whole idea of good science is that you don't believe in anything, you just go with the best you've got so far and keep questioning and improving."

- Unknown

Table of Contents

List of Figures

Chapter 2

Chapter 4

Chapter 5

Chapter 6

Appendix B

List of Tables

Chapter 3

Chapter 6

Appendix A2

List of Abbreviations

AD	Anderson-Darling
AIC	Akaike Information Criterion
AICc	Corrected Akaike Information Criterion
Coif	Coiflet wavelet
CV	Cross-validation
Db	Daubechie wavelet
DWT	Discrete wavelet transform
FDA	Functional data analysis
FLR	Functional linear regression
FPCA	Functional principal components analysis
FPCR	Functional principal components regression
FTIR	Fourier-Transform infrared spectroscopy
FWHM	Full width at half maximum
GCV	Generalized cross-validation
LOESS	Locally-weighted scatterplot smoothing
MCMC	Markov chain Monte Carlo
MIR	Mid-infrared spectroscopy
NIPALS	Nonlinear iterative partial least squares
NIR	Near-infrared spectroscopy
PC	Principal component
PCA	Principal components analysis
PLS	Partial least squares
RMSE	Root mean square error

RMSEC	Root mean square error of calibration
RMSECV	Root mean square error of cross-validation
RMSEP	Root mean square error of prediction
SNR	Signal to noise ratio
SVD	Singular value decomposition
SWT	Stationary wavelet transform
Sym	Symlet wavelet
WP	Wavelet packets

List of Symbols and Nomenclature

Lowercase characters in boldface are vectors, and bold, upper-case letters are matrices.

β	Scalar linear regression coefficient
$\beta(t)$	Regression coefficient function
ε	Residual error – the difference between actual and predicted values
λ	Magnitude of linear differential operator penalty
$v(t)$	Eigenfunction (i.e. functional principal component)
ρ	Eigenvalue
σ	Standard deviation
ϕ, ψ	Basis functions
Φ	*n* x *K* matrix of basis functions
$\boldsymbol{\zeta}$	Vector of scalar regression coefficients
A	Number of principal components
b, **c**	*K*-vectors of basis function coefficients
C	*N* x *K* matrix of basis function coefficients
E	Residuals matrix
df	Degrees of freedom
H	"hat" matrix defining the mapping from $\mathbf{x}$ to $\hat{\mathbf{x}}$
J	Number of variables
K	Number of basis functions
L	Linear differential operator
$\mathcal{L}(\hat{\theta})$	Likelihood function

m	Order of derivative
N	Number of independent observations (may be functional or scalar)
n	Number of discretely observed points
$\mathbf{P}$	P-loadings matrix
$\mathbf{p}$	P-loading
p	Polynomial order or number of model parameters
$\mathbf{R}$	Roughness penalty matrix
$\mathbf{S}$	Sample covariance matrix
S	Sample covariance function
$\mathbf{T}$	Principal component scores matrix
t	t-score
t	Functional argument value
$\mathrm{tr}(\cdot)$	Trace of a matrix
τ	Number of internal knots
u	Knot location
$\mathbf{X}$	N x K or L matrix of observations
$\mathbf{x}$	n-vector containing discretized values of an observation as a function of t over the interval $[t_1,t_\mathrm{n}]$
x_i	Individual discrete or functional observation
$\mathbf{y}$	Vector of discrete observations
$\mathbf{v}$	Eigenvector
$\langle\cdot,\cdot\rangle$	Inner product
$\|\cdot\|$	Magnitude (2-norm)

Chapter 1

Introduction

Functional data analysis (FDA) presents an alternate and potentially more informative approach to traditional statistics in the analysis of data which can be viewed as (usually-continuous) functions, where high correlation exists between neighboring points. As opposed to discrete data, the data objects in FDA are functions – by their nature infinitely-dimensional – defined over a closed interval. Electromagnetic chemical spectra [1-5]; growth charts [6]; polymer molecular weight distributions; topography and thickness of sheet metal on production lines; and periodic phenomena such as unemployment cycles or environmental emissions [7] over the course of a day all lend themselves to this type of analysis.

1.1 Background & Motivation

A vast, well-established suite of discrete data processing and multivariate regression techniques exist for analyzing chemical spectra, and it may be argued that discrete statistics has reached a certain level of maturity in this context. Comparatively little has been done in the functional space, however, and there is much to be explored in a field that only in recent years is moving to mainstream academia and is all but unknown in industrial applications.

The primary objective of this work was to evaluate and improve upon functional statistical techniques in chemometrics, specifically in two related aspects: infrared spectral denoising to obtain observations nearer to the "true" functions, and correlation of such spectra with chemical and physical characteristics. Producing reliable spectral reproductions and correlation models is of particular interest because of the rapid, non-destructive, and inexpensive way these spectra can be used to automatically classify compounds and determine a variety of sample qualities. Since infrared spectra are functions of chemical structure and composition, they can be used to reliably predict not only these characteristics but also many physical properties deriving therefrom [8].

1.2 Scope and Organization of Thesis

Development was geared towards implementing a robust, computationally efficient solution capable of being deployed in industrial settings. Minimizing or eliminating ambiguities in model building and analysis by automating the techniques were therefore the guiding factors in this work. To this end, the first step involved gaining an understanding of FDA, chiefly in how to appropriately functionalize data (i.e., representing a series of discrete data points in functional form) to prepare it for analysis. Chapter 2 discusses this fundamental aspect of FDA and presents a literature review of how various authors have approached this problem with splines, or piece-wise polynomials.

Chapter 3 connects traditional statistical regression techniques utilized in chemometrics (e.g., linear regression, Principal Components Analysis, etc.) with their functional counterparts. This is followed by a discussion in Chapter 4 on the nature of vibrational chemical spectra, an overview of some relevant conventional spectral preprocessing techniques, and finally a review of the current literature in functional chemometrics.

Chapter 5 discusses the performance of selected spline fitting algorithms presented in the second chapter and details the adaptation and optimization of a suitable heuristic algorithm to the problem of spectral smoothing. In Chapter 6, this optimized algorithm is applied to two real datasets to compare traditional spectral correlation techniques with functional regression models.

Further comparisons are made between the functional models based on the heuristic algorithm and those prepared using more traditional functional methodologies.

Finally, Chapter 7 presents a summary of the Thesis, conclusions and contributions made from this work, and recommendations for further investigation on the matters discussed herein.

Chapter 2

Functional Data Analysis – Statistics in the Functional Space

2.1 Constructing the Basis

Since data captured digitally is always in discrete form, the requisite step in functional data analysis is, as the name suggests, fitting a function to the data. Of the many different possible function types, including polynomial, Fourier series, various types of splines, wavelets, among others, the choice of which to use is effectively determined by the nature of the data and, to some extent, what information one wishes to glean from it (for example in the analysis of derivatives or frequency). The functional theory presented in this chapter is referenced primarily from [6], [9].

Fitting a set of basis functions to fit a data set is fundamentally a form of linear regression based on the error model

$$x_i = f(t_i) + \varepsilon_i \tag{1}$$

on the assumptions that the errors ε_i exist only in the response, are i.i.d., and $N(0, \sigma_\varepsilon^2)$, and where $f(\cdot)$ is the unknown (true) smoothing function evaluated at sampling points t_i ($i = 1,\ldots, n$). The error assumptions would be used if statistical inferences (e.g., confidence intervals) are being made, but are not necessary for model building. The fit is defined by the expansion of K basis functions

$$f(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \boldsymbol{c}'\boldsymbol{\phi}(t) = \boldsymbol{\phi}(t)'\mathbf{c} = \begin{bmatrix} c_1 & \cdots & c_K \end{bmatrix} \begin{bmatrix} \phi_1(t) \\ \vdots \\ \phi_K(t) \end{bmatrix} \tag{2}$$

where c_k are the coefficients and $\phi_k(t)$ represent individual basis functions. Substituting Equations (2) into (1) and evaluating the latter at each of the *n* observed sampling points in the curve results in

$$\mathbf{x}_{(n \text{ x } 1)} = \Phi_{(n \text{ x } K)}\mathbf{c}_{(K \text{ x } 1)} + \boldsymbol{\varepsilon}_{(n \text{ x } 1)}. \tag{3}$$

The coefficients can be estimated by minimizing the residual or error sum of squares:

$$\text{SSE} = \sum_{i}^{n}[x_i - f(t_i)]^2 = (\mathbf{x} - \Phi\mathbf{c})'(\mathbf{x} - \Phi\mathbf{c}) \tag{4}$$

to give the ordinary least squares solution

$$\hat{\mathbf{c}} = (\Phi'\Phi)^{-1}\Phi'\mathbf{x}\,. \tag{5}$$

The regression becomes interpolating as *K* approaches *n*.

Of course, the fit will be suboptimal if the ordinary least squares assumptions do not hold. For example, it may be prudent to weight the residuals if the errors are heteroscedastic or autocorrelated, such that Equations (4) and (5) respectively become

$$\text{SSE} = (\mathbf{x} - \Phi\mathbf{c})'\mathbf{W}(\mathbf{x} - \Phi\mathbf{c}) \tag{6}$$

$$\hat{\mathbf{c}} = (\Phi'\mathbf{W}\Phi)^{-1}\Phi'\mathbf{W}\mathbf{x} \tag{7}$$

where **W** is the symmetric, positive definite weighting matrix.

2.2 *B*-Splines

Splines are often the basis of choice for many functional analyses, given their ease of computation and ability to approximate virtually any kind of function. They are defined as a series of connected piecewise polynomials over a specified range of argument values. The range is divided into subintervals, each containing a unique polynomial delimited by break points where one polynomial meets the next. How these polynomials are connected is defined by the knots $U = \{u_0, u_1, u_2,\ldots,u_{\tau+p}\}$, a set of non-decreasing points on the argument interval $[a,b]$, which lie at the break points and enforce a certain degree of smoothness between each segment. The number of interior knots is given by τ. Smoothness, or order of continuity G^m, is the number of derivatives m shared by the spline basis functions at a break point. For a single knot at a breakpoint, G^m is related to the spline order p by

$$m = p - 2 \tag{8}$$

For example adjacent segments of a fourth order or cubic spline have the same first and second derivatives when a single knot is placed at a common breakpoint. In most applications it suffices to have each break point defined by a single knot; however, it is possible to enforce discontinuities in the derivatives using multiple coincident knots – each additional knot at a break point decreases G^m by one.

Amongst the various types of splines available, *basis* or B-splines lend themselves well to problems in the functional field. B-splines can approximate virtually any smooth function and are compactly supported, meaning that the basis functions are nonzero over a small interval (across no more than $p - 1$ breakpoints) and zero everywhere else. This results in a banded matrix $\Phi'\Phi$, with $p - 1$ diagonals above and below the main diagonal. In addition to improved numerical stability, this confers a significant benefit in that computational effort is proportional to K, compared to K^2 for bases without compact support [9]. Characteristic of B-splines is the fact that the functions are forced to be non-differentiable (i.e., discontinuous) at the boundaries by the placement of an additional $p - 1$ knots at these locations. For example, for a fourth order spline, the end breakpoints have four coincident knots. It is also interesting to note that the basis functions sum to unity at any given point over the interval.

In consideration of the properties outlined above, individual *B*-spline functions $\phi_{k,p}(t)$ can be defined recurrently by the Cox-de Boor relation [10], for the first order case (*p* = 1) as

$$\phi_{k,1}(t) = \begin{cases} 1 \text{ if } u_k \leq t < u_{k+1} \\ 0 \text{ otherwise} \end{cases} \qquad (k = 0, \dots, \tau + p - 1) \tag{9}$$

and for higher orders (*p* > 1) as

$$\phi_{k,p}(t) = \frac{t - u_k}{u_{k+p-1} - u_k} \phi_{k,p-1}(t) + \frac{u_{k+p} - t}{u_{k+p} - u_{k+1}} \phi_{k+1,p-1}(t) \tag{10}$$

$(k = 0, \dots, \tau)$. The convention $\frac{0}{0} = 0$ is applied to Equation (10) if necessary. To aid in visualization, an example fourth-order B-spline basis with 6 equally-spaced interior knots is presented in Figure 2.1*a*, and a basis using 6 arbitrary-placed knots is shown in Figure 2.1*b*.

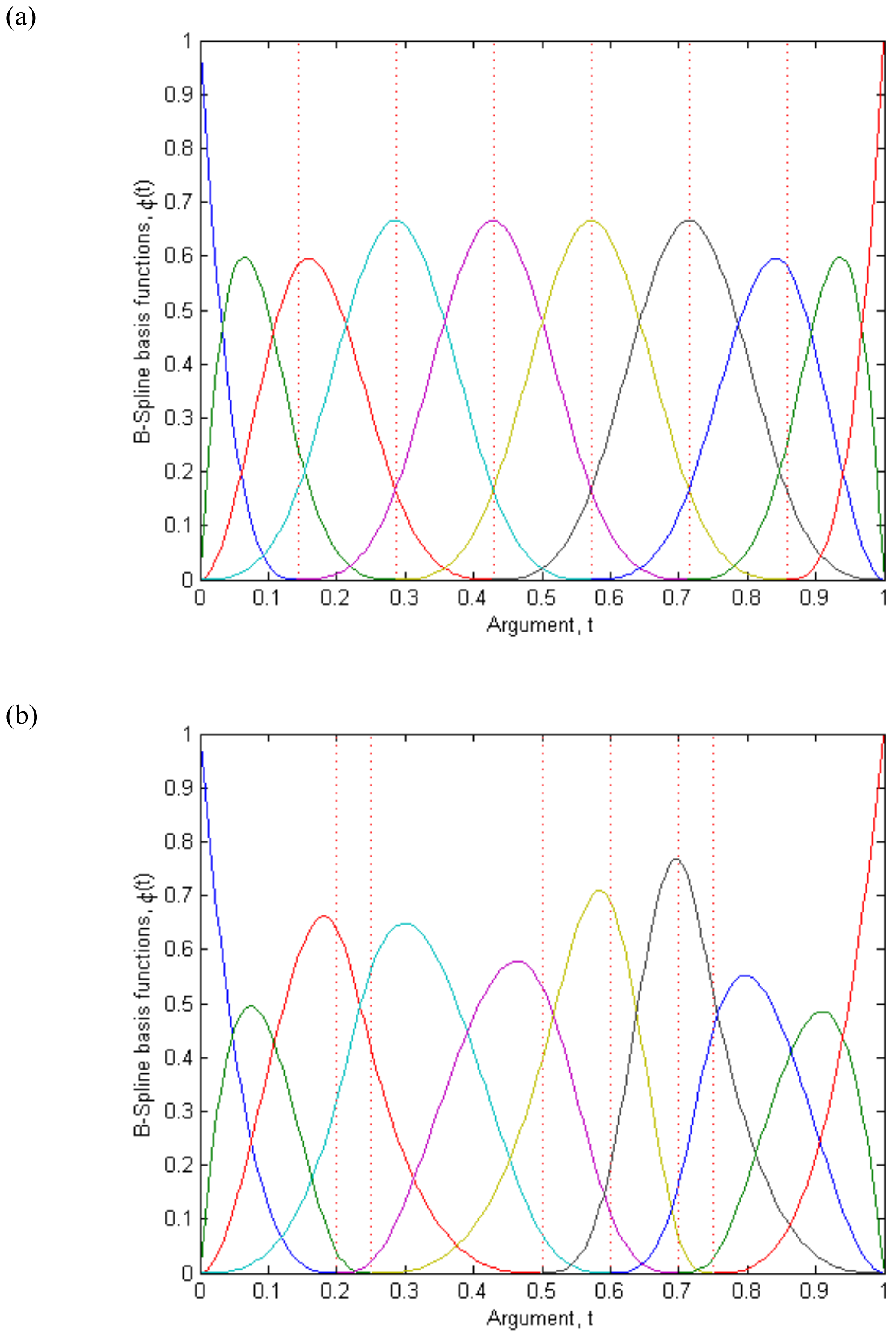


Figure 2.1. Example of ten, fourth-order B-spline basis functions on the interval [0,1] defined by (a) six, uniformly-spaced knots, and (b) six, unequally-spaced knots.

2.2.1 Defining B-Spline Bases

Data-fitting using splines can be accomplished by simply limiting the number of knots (*regression splines*), placing a knot at each data point in conjunction with a roughness penalty (*smoothing splines*), or by applying a penalty to a system with a reduced number of knots (*P-splines*). Knots can either be placed at fixed intervals or data quantiles, or freely according to the features of the data. For penalized systems, an additional term is added to Equation (4) such that the minimization problem becomes

$$\min\left(\sum_{i}^{n}[x_i - f(t_i)]^2 + \lambda\int[Lf(t)]^2 dt\right) \tag{11}$$

where λ is the magnitude of the penalty and L is a linear differential operator (LDO) of order *m*. Adding a penalty reduces the risk of over-fitting [11] and imposes a form of ridge regression, which adds numerical stability to the calculation by emphasizing the diagonal of the regression matrix. This becomes particularly important as $K \rightarrow n$.

The second derivative, D^2, of the fitted curve is often chosen as the penalizing operator, since its square defines the curvature – a common measure of roughness – of the function [9]. Taking $L = D^2$ and substituting the basis expansion into Equation (11) yields

$$\min\left(\sum_{i}^{n}[x_i - \boldsymbol{\phi}'(t_i)\mathbf{c}]^2 + \lambda\mathbf{c}'\left[\int D^2\boldsymbol{\phi}(t)D^2\boldsymbol{\phi}'(t)\,\mathrm{d}t\right]\mathbf{c}\right) \tag{12}$$

such that the coefficients estimates are

$$\hat{\mathbf{c}} = (\Phi'\Phi + \lambda\mathbf{R})^{-1}\Phi'\mathbf{x} \tag{13}$$

where the roughness penalty matrix

$$\mathbf{R} = \int D^2\boldsymbol{\phi}(t)D^2\boldsymbol{\phi}'(t)\,\mathrm{d}t \tag{14}$$

2.2.1.1 A Word on Degrees of Freedom

The degrees of freedom *df* in a typical regression fit is simply the number of parameters p used to define the model. By corollary the degrees of freedom for the residuals or error is $n - p$. For regression splines, $p = K$; however, the use of a roughness penalty in P-splines introduces a new sort of parameter that requires a different interpretation of what p should be.

The "Hat" matrix

$$\mathbf{H} = \Phi(\Phi'\Phi + \lambda\mathbf{R})^{-1}\Phi' \tag{15}$$

defines the mapping $\mathbf{x} \to \hat{\mathbf{x}}$ and has a trace equal to the effective degrees of freedom $df(\lambda, K)$ associated with the spline fit . As $\lambda \to 0$, $df(\lambda, K) \to K$. Conversely, as $\lambda \to \infty$, $df(\lambda, K) \to m$ the least squares estimate becomes the matrix **R** (Equation 14), for which the solution is an $m - 1$ degree polynomial [10].

2.2.2 Parameter Selection for B-splines

Assuming the knots and roughness penalty have been specified, Equation (13) represents the solution to linear problem. However, globally optimizing the number of knots, their locations, and the roughness penalty is a nonlinear issue that rapidly becomes computationally intractable by simple grid search methods. It is no surprise, then, that intense efforts have been dedicated to finding alternative approaches and compromise solutions to the "best" fit (the definition of which is itself somewhat arbitrary).

A plethora of measures exist for assessing goodness of fit, and by corollary, selecting appropriate fitting parameters. Popular objective measures include but are not limited to root mean square errors, various forms of cross-validation (CV) [12-15], maximum likelihood [14], [15], the Anderson-Darling test [16], and penalized information-theoretical functions such as the

Akaike Information Criterion (AIC) [14], [17]. The predictive power of regression models and multivariate techniques such as Principal Components Analysis (PCA), Principal Components Regression (PCR), and Partial Least Squares (PLS) can also be used as fitness measures to determine optimal fitting parameters.

2.2.2.1 Spline Order

The choice of spline order is often straightforward. Fourth order (cubic) splines are preferred because they are continually smooth (defined as D^2 being equal at the knots) and optimal in minimizing the residual sum of squares for a penalized system [18]. Quality of fit is generally not highly sensitive to spline order, however. In fact, using derivatives can sometimes reveal particularly interesting features in the data, in which case higher spline orders are required to ensure that the derivative of interest is continuously smooth [6].

2.2.2.2 Knot Vectors and Roughness Penalties

While selecting spline order is relatively straightforward, the question of how many knots to use, where to place them, and what sort of smoothing penalty to use (if any), is more ambiguous. As noted earlier, the problem here is multimodal and nonlinear; Equation (13) and its related forms tend to have multiple local minima when the penalty and knot vector are considered as free variables. Dozens of approaches have been proposed, ranging from simply minimizing any of various objective functions using equally spaced knots [2], [6], [11], [15], [16], [18] to more advanced techniques involving random search placement employing genetic algorithms [19] or Bayesian Monte-Carlo methods [20], with spatially adaptive penalties [21], [22]. Rapid execution algorithms using empirical rules for knot placement [23], [24], and particle swarm optimization [25] have also met with success. Some of these [25], [26] are able to accurately model discontinuities by automatically placing multiple knots at the same location. Ultimately, the type of data dictates which method(s) may be most appropriate. However, while some techniques are evidently faster and more robust than others, there is no distinctly universal approach. As such, a variety of algorithms was evaluated based on suitability to the problem of interest, availability of code (or where code was unavailable, the ease of implementation), and computation time. A selection of methods investigated in more detail is described below. Suitability assessments are presented in Chapter 3.

Equispaced knot placement is generally best-suited for fitting nonlinear trends that change relatively smoothly, without discontinuities or cusps. Indeed when using P-splines the exact value of the number of knots *K* becomes less important since the problem is partly offloaded onto the selection of a suitable penalty to avoid false oscillations and guard against overfitting [14], [27]. The simplest approach for equally-spaced knots is the rule of thumb observed by [11], [28]:

$$K = \min\left(\frac{n}{4}, 35 \text{ or } 40\right). \tag{16}$$

A spline may also be fit by iteratively searching for a *K* and roughness penalty *λ* that simultaneously minimize some criterion, such as the generalized cross-validation (GCV) measure [11] developed by Craven and Wahba [13] and given by

$$\mathrm{GCV}(\lambda, K) = \left(\frac{n}{n - df(\lambda, K)}\right)\left(\frac{SSE}{n - df(\lambda, K)}\right); \tag{17}$$

or root-mean square error of calibration (RMSEC); cross-validation (RMSECV); or prediction (RMSEP). These three terms are often ambiguously defined in the literature, yet it is important to make clear distinctions. RMSEC is simply the standard deviation of residuals from a model fit:

$$\mathrm{RMSEC} = \sqrt{\frac{SSE}{n - df}}, \tag{18}$$

where *df* is the degrees of freedom of the hat matrix, and can be made arbitrarily small as the number of regression parameters (i.e., knots) approaches the number of points *n* used in the regression (again, which could result in overfitting if a roughness penalty is not used). RMSEP is reserved for predictive models and is defined as

$$\mathrm{RMSEP} = \sqrt{\frac{SSE}{N}} \tag{19}$$

where *N* is the size of the external validation set (not used in model building), and SSE is the error sum of squares between the predicted and actual values.

RMSECV is similar to RMSEP, except that it is a measure of the errors obtained from iteratively parsing the calibration data into multiple, non-identical training and validation sets. This is a widely-used technique that evaluates the predictive power of a statistical model that is particularly useful when the calibration dataset is small. In *K-fold* cross-validation (KCV), for example, the data is randomly divided into *K* subsamples, one of which is used for validating a model derived from the remaining *K* – 1 partitions. Each subsample sequentially takes on the role of the validation set, over *K* iterations. The RMSE cross-validation measure is thus given by

$$\mathrm{RMSECV} = \sqrt{\sum_{k=1}^{K} \frac{SSE_k}{N_k}}. \tag{20}$$

KCV is not usually used in fitting basis functions because it can be computationally taxing to calculate when the number of data points and/or partition size is very large (as with spectral datasets) – GCV is the more expedient option in these situations.

Some care must also be taken when using cross-validation and RMSEs in model fitting. RMSECV and RMSEP may oscillate considerably with changing parameters, such as the number of basis functions, compared to the more monotonic nature of RMSEC [2]. It is therefore best to consider these three metrics in tandem, balancing model parsimony (reflected in RMSEC) against predictive power (RMSECV and RMSEP), or use them as part of a different metric, such as the Akaike Information Criterion [17], discussed next. Furthermore, choosing the number of partitions in KCV is itself somewhat arbitrary. Leave-one-out cross-validation ($K = N$) is not recommended, particularly for larger datasets, since this can result in overfitting and overly-optimistic prediction errors [29]. The opposite case, when K is small, can be very slow to compute or simply unreasonable if the number of samples in the training set becomes too small to build useful models, particularly in dimension-reducing techniques such as PCA and PLS.

The Akaike Information Criterion (AIC) [17] is another popular fitness function often used for model discrimination by incorporating both a goodness of fit measure and a penalty against the number of parameters. In other words, model parsimony is balanced against how well the data is fit. With respect to the number of knots K in smoothing spline system (i.e., no roughness penalty) it may be written as

$$AIC(K) = -2\ln\left(\mathcal{L}(\hat{\theta})\right) + 2K \tag{21}$$

where $\mathcal{L}(\hat{\theta})$ is the likelihood function (RMSE, for example). A corrected version of the criterion (AICc) is recommended in practice, however, to adjust for small sample sizes [30]:

$$AICc(K) = AIC(K) + \frac{2K(K+1)}{N-K-1} \tag{22}$$

The AICc criterion has also been shown to work well in choosing an appropriate λ in penalized systems [31], in which case it is written as

$$AICc(\lambda) = \log\frac{\|(\mathbf{H}-\mathbf{I})\mathbf{y}\|^2}{N} + 1 + \frac{2[\mathrm{tr}(\mathbf{H})+1]}{N-\mathrm{tr}(\mathbf{H})-2} \tag{23}$$

Model distinction is enhanced by representing the individual criterion values to a delta from the minimum [32]:

$$\Delta_i = AICc_i - AICc_{min} \tag{24}$$

such that $\Delta = 0$ for the best (i.e. most plausible) model. As a guideline, $\Delta_i \leq 2$ indicates strong support for a model, $4 \leq \Delta_i \leq 7$, much less so, and there is virtually no support when $\Delta_i > 10$.

More recently, Kauermann and Opsomer [15] proposed an alternative objective function where K is selected by maximizing the log likelihood function of the spline model:

$$l_k(\hat{\mathbf{c}}, \hat{\sigma}_\varepsilon^2, \hat{\alpha}) = -\frac{N}{2}\ \log(\hat{\sigma}_\varepsilon^2) - \frac{1}{2}\ \log\left|V_{K,\hat{\alpha}}\right| \tag{25}$$

where $V_{K,\hat{\alpha}} = I_n + \frac{1}{\hat{\alpha}}\Phi\Phi'$ and $\hat{\alpha} = \frac{\hat{\sigma}_\varepsilon^2}{\hat{\sigma}_{\mathbf{c}}^2}$, with the "hat" indicating that these variables are maximum likelihood estimators of the variance of the error $\hat{\sigma}_\varepsilon^2$ and spline coefficients $\hat{\sigma}_{\mathbf{c}}^2$. While the study was small in scope and limited to a few simulated datasets, the authors found their approach to behave similarly to parameter selection by minimizing GCV.

Rowlands and Elliot [16] introduced a novel method of choosing K for automatically denoising chemical spectra that uses the Anderson-Darling (AD) test statistic as the objective function. The algorithm operates under the assumption that the residuals from an optimal spline fit follow the noise distribution characteristic of the given spectrum. While the authors used exclusively Raman spectra, whose noise can be assumed essentially normal, the algorithm is applicable to any situation for which the noise distribution is known and the AD test is available. The AD test is a robust tool for determining whether or not a data sample comes from a specific distribution [33]. The test requires the data, in this case the residuals ε, to first be mean-centered and scaled to unit variance. The data is then sorted into ascending order, such that $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_n$. The test statistic is defined as

$$A^2 = -2 - S \tag{26}$$

where

$$S = \sum_{i=1}^{n} \frac{2i-1}{n}\left(\ln\left(F(\varepsilon_i)\right) + \ln[1 - F(\varepsilon_{n+1-i})]\right), \tag{27}$$

F(x) being the cumulative distribution function assumed for the data, which for a normal distribution is

$$F(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \tag{28}$$

and ε is assumed $N(0,1)$. While it is possible to calculate the probability associated with the test statistic, the algorithm only calls for a minimization of A, which is achieved using a simplex algorithm to search for the number of knots that minimizing the test statistic.

A number of spatially adaptive roughness penalty algorithms have also been developed to produce better spline fits to functions displaying strong heterogeneity; that is, data with both relatively smooth regions and areas of sharp oscillations. More recent examples include Bayesian approaches for heterogeneous data with homoscedastic [21] and heteroscedastic [22] noise. These employ Markov-Chain Monte Carlo (MCMC) sampling for estimating the posterior density for spline coefficients of a secondary spline that defines a varying roughness penalty. Both approaches require the number of knots for each spline to be user-specified. However, the number of knots did not play a large role in goodness of fit for the artificial data on which the algorithm was tested – little change was seen as K for each spline increased beyond a roughly data-dependent minimum. Krivibokova et al. [34] improved upon the method by Crainiceanu et al. [22] by employing a Laplace approximation to circumvent MCMC, resulting in much faster calculations. Results were effectively on par with the MCMC method.

Using a data-driven approach, Yao and Lee [24] proposed enhancing a *P*-spline fit by placing additional knots at local minima and maxima. A spline is initially fit using equally spaced knots, the number of which follows Ruppert's heuristic rule, and a roughness penalty obtained by minimizing the GCV. New knots are then placed at peaks and valleys, identified by a sign change in the first derivative. The procedure can be repeated to identify more local extrema, however the authors concluded from simulations that further iterations did not, in general, significantly improve quality of fit.

An algorithm originally designed to fit B-splines to one-dimensional profiles of automotive parts [23] presents yet another unique and computationally rapid approach to adaptive knot placement. The method first calculates a discrete curvature, defined as the reciprocal of the radius of a circle bounded by three adjacent data points. The curvature is denoised using a lowpass filter and used to place knots in a spatially adaptive manner such that the final fit satisfies as closely as possible a heuristic rule stating that the angle between two vectors constructed from three consecutive points is less than π/6. Although it is perhaps more sensitive to noise on account of the discrete curvature calculations, this method compares favorably with a number of other approaches in speed of execution and ability to accurately model a variety of complex functions [25].

Chapter 3

Functional Regression & Multivariate FDA

As one might expect, regression techniques and analysis of variance in the functional space are grounded on methods originally developed for discrete data. FDA introduces a broader gamut of possible conditions for analysis. In regression, for example, both the regressor(s) and response(s) can be functional. Perhaps more commonly though, and as is the case in the present work, only the regressor is functional. A brief discussion on some pertinent traditional methods is in order before introducing their respective functional descendants. Unless otherwise indicated, the functional methodologies described in this chapter are referenced from [6] and [9].

3.1 Moving from Discrete to Functional Spaces

The construction of functional methods for data analysis requires that the operators used in discrete statistics have corresponding analogues available in the functional space. As noted in the exposition Chapter 1, data objects in FDA are functions specified over a fixed interval. Only the univariate case is considered herein, in which curves are functions of a single variable, but the methods described are readily extensible to multivariate functions.

The addition operator is perhaps trivial, as is multiplication by a constant; however, the concepts of magnitude and inner product bear some exposition. In a finite-dimensional Euclidian vector space $\mathbb{R}^n$, the magnitude of a vector **u** (also known as the 2-norm) is the square root of the sum of squares of its components:

$$\|\mathbf{u}\| = \sqrt{u_1^2 + u_1^2 + \cdots + u_n^2} = \sqrt{\mathbf{u}'\mathbf{u}}\,. \tag{29}$$

The relationship between magnitude and the inner product $\langle \cdot, \ \cdot \rangle$ can be readily appreciated considering the definition of the latter:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}'\mathbf{y} = \sum_{i=1}^{n} x_i y_i \, . \tag{30}$$

In the limit, where the vectors are infinite-dimensional (i.e., are functions) the summation becomes the integral of the product of two functions bounded by a closed interval [*a*,*b*]:

$$\langle f(t), g(t) \rangle = \int_a^b f(t) g(t) dt \tag{31}$$

This definition permits many equations and methods in discrete statistics to be readily converted to their functional equivalents; those applied in this work are described in the following sections.

3.2 Functional Linear Regression

The ubiquitous multiple linear model with *J* regressors plus constant intercept

$$y_i = \beta_0 + \sum_{j=1}^{J} x_{ij}\beta_j + \varepsilon_i = \beta_0 + \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \varepsilon_i, \qquad i = 1, \dots, N \tag{32}$$

where $\mathbf{x}_i$ is a vector containing all *i*th observations for *J* variables and $\boldsymbol{\beta}$ contains the *J* regression coefficients, can be reformulated for scalar responses and functional predictors as

$$y_i = \beta_0 + \int_a^b x_i(t)\beta(t) dt + \varepsilon_i. \tag{33}$$

This is an underdetermined problem (more unknowns than equations) because of the infinite dimension of the regression coefficient function and a finite number of observations *N*. In other words, there are *N* equations and infinite unknowns, so an infinite number of regression coefficients exist that all predict **y** equally well. As such, the problem requires a reduction of dimensionality, which can be accomplished by representing $\beta(t)$ as a basis function expansion in the same way that the regressors are. A simple functional linear regression with constant intercept in basis expansion form is then given by

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \int_a^b \mathbf{C}_{(N\times K_z)}\boldsymbol{\phi}(t)_{(K_z\times 1)}\boldsymbol{\psi}(t)'_{(1\times K_\beta)}\hat{\mathbf{b}}_{(K_\beta\times 1)}dt \tag{34}$$

where $\mathbf{b}$ and $\mathbf{C}$ are the coefficient vector and matrix for the basis function vectors $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$ of the regressor and regression coefficient function, respectively. K_z is the number of basis functions representing the function observations, while $K_\beta$is the number of basis functions used to construct the regression coefficient function. Letting $\mathbf{J} = \int_a^b \boldsymbol{\phi}(t)\boldsymbol{\psi}(t)'dt$ and pulling out the constant terms from the integral, Equation (34) can be rewritten as

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \mathbf{CJ}\hat{\mathbf{b}} \tag{35}$$

Further simplification is possible by combining all the regression coefficients into single vector $\hat{\boldsymbol{\zeta}} = \left[\hat{\beta}_0\ \hat{b}_1\ \hat{b}_2\ \cdots\ \hat{b}_{K_\beta}\right]'$ and the remaining terms into $\mathbf{Z} = [1\ \ \mathbf{CJ}]$ to give

$$\hat{\mathbf{y}} = \mathbf{Z}\hat{\boldsymbol{\zeta}} \tag{36}$$

Under the constraint that $K_\beta < K_z$, the coefficients estimates $\hat{\boldsymbol{\zeta}}$ can be obtained by the least squares solution.

The regression may of course also incorporate a roughness penalty of its own. This is advantageous in improving numerical conditioning when K_β is large, as may be required to capture the relevant features in particularly heterogeneous functional observations (certain chemical spectra, for example). Analogously to the penalized spline fit described in Equation (11) in Section 2.2.1, the penalized regression problem is solved by minimizing

$$\min\left(\sum_i^n \left[y_i - \beta_0 - \int_a^b x_i(t)\beta(t)dt\right]^2 + \lambda_\beta \int [L\beta(t)]^2 dt\right). \tag{37}$$

The coefficients are thus

$$\hat{\boldsymbol{\zeta}} = (\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{R_0})^{-1}\mathbf{Z}'\mathbf{y} \tag{38}$$

where the penalty matrix $\mathbf{R_0}$ is the same as in Eq. (14) but augmented with a leading row and column of zeros to accommodate the constant intercept term (and of course with $\psi(t)$ in place of $\phi(t)$). As before, the roughness penalty λ_β is typically obtained by minimizing GCV(λ_β), but a number of other objective functions or semi-subjective criteria may also be used.

3.3 Principal Components Analysis (PCA)

Principal Components Analysis, which goes by various names depending on the field of application (including Karhunen-Loève transform, factor analysis, and proper orthogonal decomposition), is a method for analysis of variance in multivariate systems. PCA and its functional counterpart allow one to determine which variables introduce the greatest variance in a system, how they correlate with one another, and to explain how and why individual observations or samples differ from one another.

3.3.1 Traditional PCA

PCA operates by mapping a multivariate dataset onto a different basis to represent the data with what amounts to a new set of variables – the principal components (PC). The PCs form an orthogonal basis that identifies the sources of greatest variation within the dataset, with the first PC explaining the maximum possible variance and each subsequent component accounting for less and less. The unit-variance and column-wise mean-centered data matrix **X** of *N* samples and *J* variables is represented under this new basis with *A* ($A \leq J$) principal components as

$$\mathbf{X}_{(\boldsymbol{N}\times\boldsymbol{J})} = \mathbf{T}_{(\boldsymbol{N}\times\boldsymbol{A})}\mathbf{P}'_{(\boldsymbol{A}\times\boldsymbol{J})} + \mathbf{E}_{(\boldsymbol{N}\times\boldsymbol{J})} \tag{39}$$

where **E** contains the residuals and **T** the t-scores – the new "observations" in the space spanned by the orthogonal p-loadings **P**. The elements of **X** are given by

$$x_{ij} = \sum_{a=1}^{A} t_{ia}p_{ja} + \varepsilon_{ij} = \langle \mathbf{t}_i, \mathbf{p}_j \rangle + \varepsilon_{ij} \tag{40}$$

$$(i = 1, \dots, N) \text{ and } (j = 1, \dots, J).$$

where $\mathbf{t}_i$ and $\mathbf{p}_j$ are, respectively, the i^{th} and j^{th} rows of the matrices **T** and **P**.

A PCA decomposition may contain up to *J* principal components; however, the value in PCA lies behind simplification of the data (i.e., dimensionality reduction), and typically only the first few PCs contain useful information, with the remainder explaining mostly noise. Choosing the number of PCs to retain can often be somewhat ambiguous and subject to the analyst's discretion, but a number of guidelines exist [35] that make this process easier, some of which are outlined in Table 3.1.

Table 3.1. Summary of some popular stopping rules for selecting the number of principal components in principal components analysis.

PCA Component Selection Method	Stopping Rule
Cross-validation (for calculating fraction of explained variance R^2 and predictive power Q^2 of each PC)	When adding more PCs does not appreciably increase R^2 and/or Q^2 begins to decrease
Broken-stick rule	Retain the PCs whose eigenvalues are larger than $b_k = \frac{1}{J}\sum_{i=k}^{J}\frac{1}{i}$
Scree plot (eigenvalues plotted against their indices)	Retain the first consecutive PCs whose eigenvalues lie in the nonlinear region of the scree plot
Test of sphericity	When the PCs of the residuals **X** matrix all have similar eigenvalues and the remaining variance can be represented as a multidimensional sphere.

Many algorithms exist for the computation of principal components, two of the most popular being Nonlinear Iterative Partial Least Squares (NIPALS) and the use of singular value decomposition (SVD). The procedure via SVD perhaps provides a deeper understanding of what occurs in PCA, and is summarized as follows: the decomposition of **X** is given as the product of three matrices:

$$\mathbf{TP}' = \mathbf{ULV}' \tag{41}$$

where **T** = **U**, **V** contains the eigenvectors **v**, and **L** is a diagonal matrix with the square root of the (orthogonal) eigenvalues ρ of the J x J covariance matrix $\mathbf{\Sigma} = \mathbf{X}'\mathbf{X}$. In practice, the sample covariance matrix $\mathbf{S} = \frac{1}{N-1}\mathbf{X}'\mathbf{X}$ is used. From here, it follows that the first principal component can also be obtained by finding the solution to the eigenequation

$$\mathbf{Sv} = \rho\mathbf{v} \tag{42}$$

with the largest eigenvalue, the second PC from the solution with the second-largest eigenvalue, and so on.

3.3.2 Functional PCA (FPCA)

Data amenable to functionalization often suffers from the so-called "curse of dimensionality" [36], meaning that there are far more variables than observations, which can result in poor models. This is readily appreciated in spectral analysis, where an individual observation (a spectrum) in discrete form comprises many hundreds or thousands of variables (the sampling wavelengths). In addition to overcoming the problem of too many variables relative to the number of observations, the fact that each observation is considered a function in FPCA affords a richer analysis in which relationships within observations are preserved in addition to those between them. While the fundamental idea is the same as in traditional PCA, a variety of "flavors" have been developed incorporating different forms of smoothing and computational approaches. The theory for the approaches used herein is described below, and the reader is referred to Section 4.3 and [37] for a review of other recent developments in this area.

In contrast to PCA, the maximum number of principal component functions or eigenfunctions that can be obtained in FPCA, assuming linearly independent observations, is equal to $N - 1$, and not the number of variables (which in the functional case are infinite). Mean-centered observations in FPCA are approximated using an orthogonal basis of A ($A \leq N - 1$) eigenfunctions:

$$\hat{x}_i(t) = \sum_{a=1}^{A} t_{ia} p_a(t) \qquad (i = 1, \dots, N) \tag{43}$$

The procedure for obtaining the functional principal components and scores is somewhat more involved, but can perhaps be best appreciated via the eigenequation approach. Instead of a discrete matrix, the covariance is given by the function

$$\begin{aligned} S(q,t) &= \frac{1}{N-1}\sum_{i=1}^{N}[x_i(q)-\bar{x}(q)]\,[x_i(t)-\bar{x}(t)] \\ &= \frac{1}{N-1}\langle \mathbf{x}(q), \mathbf{x}(t)\rangle \end{aligned} \tag{44}$$

where $\mathbf{x}(\cdot)$ is an *N*-vector of mean-centered functions of the observations (and not discrete values of $x(t)$ as in Section 2.1 on basis expansions). The corresponding eigenequation is then

$$\int_a^b S(q,t)v(t)dt = \rho v(q), \tag{45}$$

which can be solved in a few different ways, summarized in Table 3.2.

Table 3.2. Computational approaches for functional principal components analysis

Method	Description
Discretization	Functional observations are discretized over a fine, equally-spaced mesh to which traditional PCA is applied. Eigenfunctions are reconstructed by normalization of their discrete approximations, followed by interpolation using the appropriate functional basis.
Basis function expansion	Each observation and the eigenfunctions are represented in basis expansion form, such that the eigenequation, Equation (45), can be solved in terms of basis function coefficients.
Numerical integration	Similar to discretization, except that the integral in Equation (45) is approximated via numerical quadrature and regular PCA is applied to a weighted matrix of the covariance function evaluated at quadrature points. Eigenfunctions can be reconstructed by interpolation.

3.3.2.1 Solving FPCA Using Basis Function Expansions

The basis function expansion method was applied in the present work because of its favorable computational qualities and seamless integration with the rest of the FDA package for Matlab, available from Professor James Ramsay's website at (). At the expense of requiring the same basis for each observation, FPCA done in this manner is computationally efficient due to the comparatively small covariance matrix of basis function coefficients. The covariance function can be rewritten as

$$S(q,t) = \frac{1}{N-1}\boldsymbol{\phi}(t)'_{(1\times K)}\mathbf{C}'_{(K\text{x}N)}\mathbf{C}_{(N\text{x}K)}\boldsymbol{\phi}(t)_{(K\times 1)} \tag{46}$$

and the eigenfunctions as

$$v(t) = \boldsymbol{\phi}(t)'_{(1\times K)}\mathbf{b}_{(K\times 1)} \tag{47}$$

so that the eigenequation becomes

$$\frac{1}{N-1}\int_a^b \boldsymbol{\phi}(q)'\mathbf{C}'\mathbf{C}\boldsymbol{\phi}(t)\boldsymbol{\phi}(t)'\mathbf{b}\,dt = \rho\boldsymbol{\phi}(q)'\mathbf{b}. \tag{48}$$

Given that the equation must hold for all q and by defining the symmetric K x K matrix $\mathbf{W} = \int_a^b \boldsymbol{\phi}(t)\boldsymbol{\phi}(t)'\,dt$, Equation (48) can be further simplified to

$$\frac{1}{N-1}\mathbf{C}'\mathbf{C}\mathbf{W}\mathbf{b} = \rho\mathbf{b}. \tag{49}$$

In order to reframe the problem for ease of computation by making the matrix on the left side symmetric, let $\mathbf{u} = \mathbf{W^{1/2}b}$, to give

$$\frac{1}{N-1}\mathbf{W}^{1/2}\mathbf{C}'\mathbf{C}\mathbf{W}^{1/2} = \rho\mathbf{u}. \tag{50}$$

The vectors $\mathbf{u}$ can be solved for using standard SVD algorithms as in regular PCA, and the eigenfunction coefficients are then simply $\mathbf{b} = \mathbf{W}^{-1/2}\mathbf{u}$. Finally, the individual component scores (t-scores) are given by

$$t_{ia} = \int_a^b v_a(t)[x_i(t) - \bar{x}(t)]\, dt. \tag{51}$$

FPCA with Roughness Penalties

Roughness penalties may also be applied to eigenfunctions to complement or substitute smoothing in the observations. Penalizing the roughness of the eigenfunctions changes the form of Equation (49) to

$$\boldsymbol{\phi}(q)' \frac{1}{N-1} \mathbf{C}'\mathbf{C}\mathbf{W}\,\mathbf{b} = \rho(\mathbf{J} + \lambda\mathbf{K})\mathbf{b}, \tag{52}$$

where $\mathbf{J} = \int_a^b \boldsymbol{\phi}(q)\boldsymbol{\phi}(q)'\, dq$ and $\mathbf{K} = \int D^2\boldsymbol{\phi}(q)D^2\boldsymbol{\phi}'(q)\ \mathrm{d}t$. With the appropriate factorization and rearrangement, the eigenfunction coefficients can then be obtained via standard PCA similarly to the non-penalized case. The reader is referred to [6] for finer details of the mathematical treatment and practical algorithms for computation.

3.3.2.2 Interpretation of the FPCA

Unrotated eigenfunctions output from a FPCA are rather predictable in terms of the type of variation they describe about the mean. That is, the PCs tend to represent a series of orthogonal polynomials, the first characterizing more or less a constant shift from the mean, the second a linear variation, the third a quadratic, and so on. Rotating the principal components can aid in interpretation. Using the VARIMAX algorithm, implemented in the FDA package, large sources of variance are amplified and small ones are attenuated, so while the amount of variance explained by each eigenfunction changes, the total sum of explained variability remains the same.

Furthermore, while one can simply observe the eigenfunctions directly, it can be more revealing to plot the mean observation function along with the addition and subtraction of a small multiple of the eigenfunction. The effect of each eigenfunction can therefore be observed as changes to the mean across the argument range.

3.4 Functional Principal Component Regression (FPCR)

As in regular PCR, FPCR for scalar responses and functional regressors operates via straightforward multiple linear regression of the response vector on the principal component t-scores, following the familiar error model

$$y_i(t) = \beta_0 + \sum_{a=1}^{A} t_{ia}\beta_a + \varepsilon_j \tag{53}$$

where β_a is the scalar regression coefficient for the a^{th} principal component function.

Remembering that the t-scores are given by Equation (51), the regression coefficient function for the FPCR model is a summation of the scalar regression coefficients multiplied by the eigenfunctions

$$\beta(t) = \sum_{a=1}^{A} \beta_a v_a(t) \tag{54}$$

with the predictive model itself given by

$$\hat{y}_i = \beta_0 + \int_a^b \beta(t)\,[x_i(t) - \bar{x}(t)]dt. \tag{55}$$

3.4.1 A Word on Partial Least Squares (PLS)

Though PLS was not directly applied in this work, it deserves mention since comparisons are made between its performance and that of the aforementioned functional regression techniques. Unlike PCR, PLS regression takes into account the relationship between both the regressors and response(s) and operates by iteratively maximizing the linear correlation between both **X** and a usually multivariate **Y**. Both these matrices are decomposed according to

$$\mathbf{X}_{(\boldsymbol{N}\times\boldsymbol{J})} = \mathbf{T}_{(\boldsymbol{N}\times\boldsymbol{A})}\mathbf{P}'_{(\boldsymbol{A}\times\boldsymbol{J})} + \mathbf{E}_{(\boldsymbol{N}\times\boldsymbol{J})} \tag{56}$$

and

$$\mathbf{Y}_{(\boldsymbol{N}\times\boldsymbol{L})} = \mathbf{U}_{(\boldsymbol{N}\times\boldsymbol{A})}\mathbf{Q}'_{(\boldsymbol{A}\times\boldsymbol{L})} + \mathbf{F}_{(\boldsymbol{N}\times\boldsymbol{L})}, \tag{57}$$

where **U**, **Q**, and **F** are the corresponding score, loadings, and error matrices for **Y**, and L is the number of response variables. The linear regression coefficients for the model ($\mathbf{Y} = \mathbf{XB}$) are obtained iteratively using an algorithm such as NIPALS, in which successive loadings are constructed, subject to the constraint of orthogonality, using the correlation between the residual **X** and residual **Y**, a process that continues up to A ($A \leq L$) PLS components.

Chapter 4

Vibrational Chemical Spectra and Chemometrics

A predominant aspect of chemometrics is the correlation of vibrational chemical spectra with specific characteristics such as chemical composition. This is of considerable practical value since it obviates the need for continually running costly, time-consuming, and potentially destructive lab tests. However, spectra must first be preprocessed to produce reliable correlations. Prior to discussing these techniques, however, it helps to understand the fundamental nature of these types of spectra.

4.1 Physical Origin of Vibrational Spectra and Their Applications

Vibrational spectra consist of measurements of reflected or attenuated light across a range of wavelengths, and are obtained by irradiating samples with low-frequency light in the near to mid-infrared range. The recorded signal is heavily dependent on chemical structure, the frequency of light that is used, and to a lesser extent, the physical state of the sample. Of course, noise from background radiation, equipment imperfections, and other confounding factors also plays a small role.

4.1.1 Raman Spectroscopy

Raman spectroscopy operates on the principle of inelastic light scattering by molecules excited with a monochromatic visible or near-IR laser [38]. It is typically used to determine chemical composition of mixtures or pure compounds in solid, liquid, or gaseous form. Physical properties such as crystallographic orientation and temperature may also be derived.

Inelastic scattering occurs when molecules in the ground state (*n* = 0) excited by the laser to a virtual state return to the first vibrational state (*n* = 1) or when molecules starting at *n* = 1 return to *n* = 0. The vibrational state can, of course, only take on discrete values, or quanta (i.e. *n* = 0, 1, 2, 3,…), and as per the Boltzmann distribution, it is predominantly equal to zero at room temperature. In Raman spectroscopy, the radiation emitted by a drop in vibration energy has a wavenumber in the MIR region in the range of 4,000 to 50 cm^{-1} [38].

4.1.2 Mid-Infrared (MIR) Spectroscopy

Mid-infrared spectra also report molecular vibrational frequencies but work on the principle of absorption rather than scattering, and from wave numbers between 4,000 to 200 cm^{-1} [38]. They are complementary to Raman spectra – if the sample is measured in the same physical state, the MIR and Raman frequencies for a given molecular vibrational transition are identical [39]. However, because samples are excited primarily from *n* = 0 to *n* = 1 by polychromatic light, MIR spectral band intensities may be considerably different from those of a Raman spectrum. Strong intensities in one are usually weak or completely absent in the other. For example, signals corresponding to homonuclear functionalities (e.g. C=C, C–C, S–S, etc.) are more apparent in Raman spectra, whereas polar groups (e.g. C–F, Si–O, C=0, etc.) dominate MIR [38]. In addition to these fundamental vibrations, transitions exist in MIR in which Δ*n* = ±2, ±3, …. These are termed "overtones" and may also be accompanied by less frequent combination transitions as well as resonance effects due to molecular symmetry.

4.1.3 Near-Infrared (NIR) Spectroscopy

The third type of vibrational spectroscopy is the near-infrared (12,500 to 4,000 cm^{-1}), which has become an excellent tool for process monitoring and quality control given its robustness and flexibility [38]. The physical principles behind NIR spectra are the same as in MIR; however, NIR is dominated by overtones and combination transitions, reflected by a decrease in absorption band intensity as the incident light frequency increases. Signal intensities for the fundamental vibrations predominant in MIR and Raman spectra, meanwhile, vary irregularly over the frequency range.

4.1.4 The Functional Nature of Vibrational Spectra

Infrared spectral forms can be represented by individual, summed, and/or convolved combinations of Gaussian and Lorentzian peaks, termed Voigt profiles. Liquids produce spectra containing peaks shaped as Voigt profiles [40], which can be well-approximated by a combination (i.e., summed) Gaussian-Lorentzian (G-L) in which each type contributes a fraction to the overall shape. The G-L is favorable in peak analysis since Voigt profiles are not available analytically. These three types of profiles are compared in Figure 4.1.

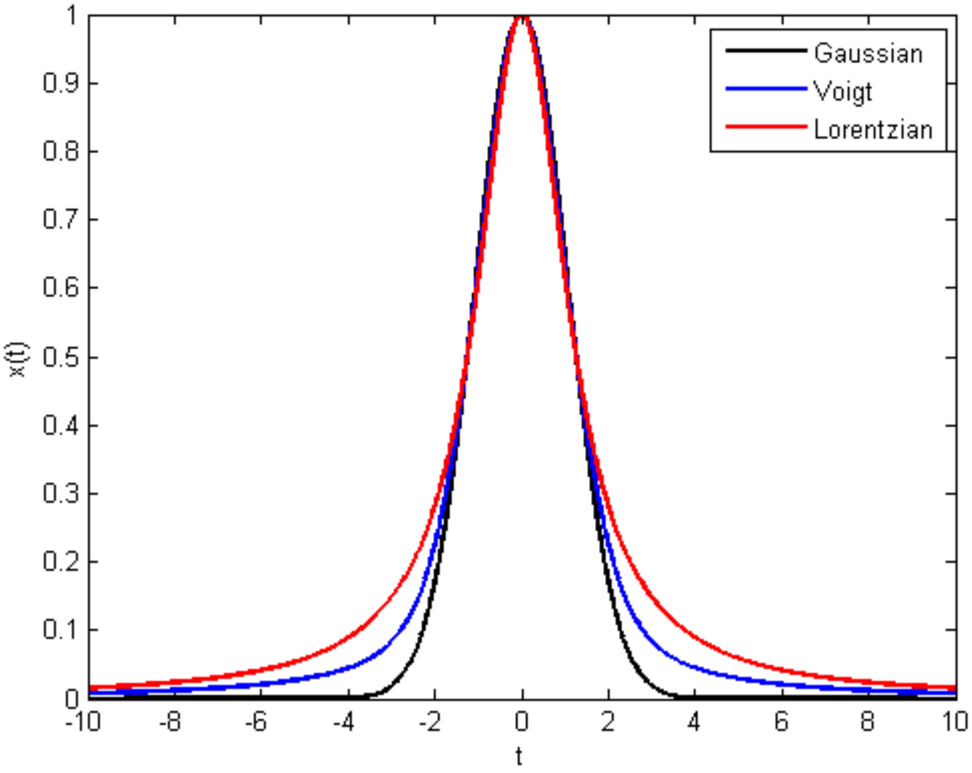


Figure 4.1. Representative Gaussian, Lorentzian, and Voigt profiles with the same full-width at half maximum. Combinations of these make up IR spectra.

While the fingerprint regions and characteristic peaks present in both Raman and IR spectroscopy can elucidate much about the structure and composition of a given material, in practice the spectra alone are usually insufficient to identify the exact substance [39]. In these cases, comparison against spectral databases will often find a match. The content of mixtures, on the other hand, may be impossible to identify unambiguously without recourse to sample history and other analytical techniques.

As with all real signals, chemical spectra are contaminated by noise and other unwanted background distortions. Both of these depend on a number of factors, including the type of spectrum being acquired (i.e., MIR, NIR, etc.); quality of the spectrometer; ambient conditions;

physical nature of the sample (particularly if it is non-liquid and/or heterogeneous); and more specifically for noise, the number of scans, or co-adds.

When sampled at high enough resolution and intensity, the predominant noise component in infrared spectra can be often reliably treated as Gaussian, even though its actual nature is fundamentally somewhat different [16]. In Raman spectra, for example, the heteroscedastic shot noise is due to fluctuations in the number of photons detected from the irradiated sample, and is particularly evident in spectrometers with less sensitive photoreceptors and weaker lasers designed for lower excitation wavelengths. Though the variance of shot noise scales with signal intensity, is often overshadowed by homoscedastic noise generated by higher-power spectrometers (i.e., from the detector, laser, etc.) when dealing with longer wavelengths. The noise in absorption spectra (MIR, NIR) meanwhile, is Poisson distributed [41], but this can be approximated as Gaussian since the signal is usually quite strong [16]. Heteroscedastic noise in IR spectra can also be generated when converting from transmittance (*T*) to absorbance (*A*) spectra, as per the Beer transform [42]:

$$A = -\log T \tag{58}$$

Noise intensity is commonly measured in decibels, and defined in terms of the signal to noise ratio (SNR). For a discretely sampled signal *s* with noise *ε*, it is defined as

$$\mathrm{SNR(dB)} = 10\log\frac{\frac{1}{n}\sum_{i=1}^{n}[s_i]^2}{\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i}. \tag{59}$$

4.2 An Overview of Preprocessing Techniques

The objective of preprocessing is to remove uninformative variance, and may be considered by two aspects often applied in tandem: first, smoothing to remove variance due to noise, which permits faster throughput by reducing the number of scans or co-adds required to achieve a useful SNR; and second, baseline correction to reduce variability caused by background radiation or inconsistencies in sample preparation. A brief overview of variance correction techniques relevant to the present work is discussed next. Denoising methods are presented in Chapter 5, and the reader is referred to various textbooks [8], [38], [43] and the work by Krasznai [44] for more comprehensive discussions on the matter.

4.2.1.1 Mean-centering and Unit Variance Scaling

The mean-centering and scaling to unit variance, jointly referred to as autoscaling and alluded to in the discussion on PCA in the previous chapter is one of the most common preprocessing techniques. Mean centering simply involves subtracting the sample mean from each spectrum, which simplifies regression models by obviating the need for an intercept term (though zero-intercept models may not necessarily be the most precise) [38]. Unit variance scaling, meanwhile, operates by dividing each data point by the standard deviation of the sample at that wavenumber. This eliminates scaling effects that can cause large variables to disproportionately influence models by overshadowing smaller ones, which can sometimes contain important information.

4.2.1.2 First or Second Derivative

Taking the first or second derivative can correct for baseline distortion; however, this can come at the expense of amplifying any existing noise and it is prudent to combine this approach with a denoising technique to ensure relatively smooth derivatives. Taking the first derivative will also cause peak shifting, which can complicate spectral interpretation; the second derivate does not suffer from this problem.

4.2.1.3 Principal Components Analysis (PCA)

Although PCA is in and of itself a useful tool for analysis of variance and sample classification, it can also be used as a precursor to building regression models by helping identify important variance and reducing the dimensionality of the data. This makes it possible to construct linear regression models (i.e., principal components regression) that are well conditioned and provide accurate results in cases where multiple linear regression fails due to the problem of underdetermination.

4.3 Functional Chemometrics

While traditional spectral preprocessing and correlation is well established, with a long history, comparatively little attention has been given to treating spectra as continuous functions. Functional regression models in fact generally possess similar predictive capability as their discrete counterparts [1-3], [45], however the former possess important advantages that cannot be discounted. For one, the correlation structure between observations is preserved in functional regression models, whereas in PCR or PLS the observation vectors can be permuted arbitrarily and the predictions will remain the same. Secondly, representing data as basis function expansions achieves significant dimensionality reduction, improving conditioning of the regression problem from the start. This can in turn reduce the number principal components necessary to produce accurate models [3], [45]. Third, as a consequence of the first two points, functional methods will generally yield smoother, more meaningful, and more easily-interpretable regression coefficients [2], [45]. Most functional chemometric studies thus far have focused on NIR spectra and employed B-spline-functionalization using equally spaced or semi-subjective user-defined knot placement. To the best of the author's knowledge, comparisons between equispaced and adaptively-placed knots on regression model quality have not yet been explored. A summary of some recent developments in functional chemometrics follows below.

Reiss and Ogden [1] developed two flavors of FPCR and FPLS using B-splines and different roughness penalty approaches: one in which smoothing is applied to the principal components, and in the other a penalty is incorporated into the regression on the t-scores. The authors constructed scalar response models using real and artificial NIR spectra to predict water and protein content of wheat and octane number of gasoline. Spectra were functionalized using 40 equally-spaced knots and roughness penalties selected by cross-validation or restricted maximum likelihood. FPCR and FPLS with smoothing penalties in the regression performed similarly and typically outperformed other functional models as well as traditional PCR and PLS.

Saeys et al. [2] fit B-splines to NIR spectra of hog manure and diesel using the objective function

$$\arg\min_{\lambda,K} \mathrm{RMSECV}(\lambda, K), \tag{60}$$

to compare FLR with PLS. Interestingly, while this approach often yielded very poor reproductions of the original spectra, predictive power of the functional regressions was on par with that of traditional PLS. The authors note, however, that their chosen method of parameter selection may not be optimal and other approaches could be considered.

Another form of FPLS was developed by Aguilera et al. [3], in which the regression of scalar responses on functional observations is reduced to a PLS regression on a transformed matrix of the B-spline basis coefficients. Knots locations were user-defined according to features in the observations. When using NIR spectra of minced meat, FPLS predicted fat content more accurately than FPCR. By contrast, both functional methods performed similarly in predicting cookie quality from curves of dough resistance as a function of kneading time. The authors also found the predictive power of functional and discrete models to be very similar.

More recently, Zhao et al. [5] compared B-splines with wavelets as the functional bases in FLR models using simulated data and the NIR spectra of wheat and minced meat. Spline knots and the roughness penalty were selected automatically by minimizing GCV, while wavelet bases were obtained via the discrete wavelet transform (DWT) with Daubechies 8 wavelets. In most cases, the wavelet models were more accurate than their B-spline counterparts.

Chapter 5

Algorithm Selection and Optimization

While a variety of well-established discrete methods exist to denoise and otherwise pretreat signals, the issue of optimally functionalizing chemical spectra (with B-splines or otherwise) has received little attention. B-spline fits afford rapid and (assuming a good fit) comparatively smooth analytical integrals and derivatives for potentially richer analysis. Spline smoothing also acts as a direction invariant filter, in contrast to many traditional filtering techniques that treat data as a sequential series and denoise in a directional sense. This chapter explores of the suitability of the various spline knot selection methods outlined in Chapter 2 to the problem of spectral fitting. A suitable algorithm is identified and optimized using synthetic and real data to quantify performance.

5.1 Fitting Splines to Spectra – Preliminary Evaluation of the Algorithms

5.1.1 Methodology

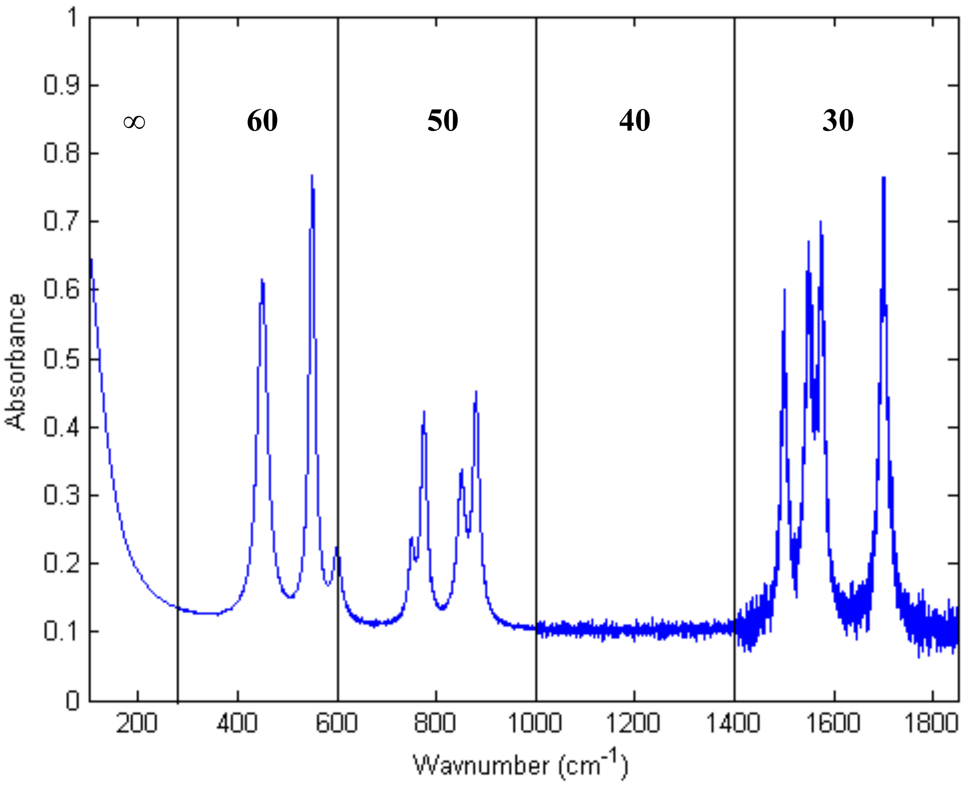
The various knot placement techniques identified in Section 2.2.2.2 were tested for quality of fit and robustness to different types of data in a qualitative screening process that considered how well the spline approximated the given functions, the distribution of residuals, and model parsimony as measured by the number of knots. Test functions consisted of selected MIR spectra generously provided by Daniel Krasznai, Raman spectra from the RRUFF mineralogical database available at http://rruff.info/, six noise-free artificial Raman spectra based on real compounds, and two functions commonly found in the spline literature:

$$f_1(x) = \frac{\sin(2\pi t) - \sin(\pi t)}{\pi t} \tag{61}$$

$$f_2(x) = \sin(2t) + 2\exp(-16t^2) \tag{62}$$

The simulated Raman spectra – three absorbance and their transmittance counterparts – were constructed to be representative of a solid (the mineral Sillimanite), liquid (ethanol), and gas (methylamine). A progression of increasing levels of added noise for the spectrum based on methylamine in presented in Figure 5.1*a*, and the noise-free synthetic spectra are presented in Figure 5.1*b*.

(a)



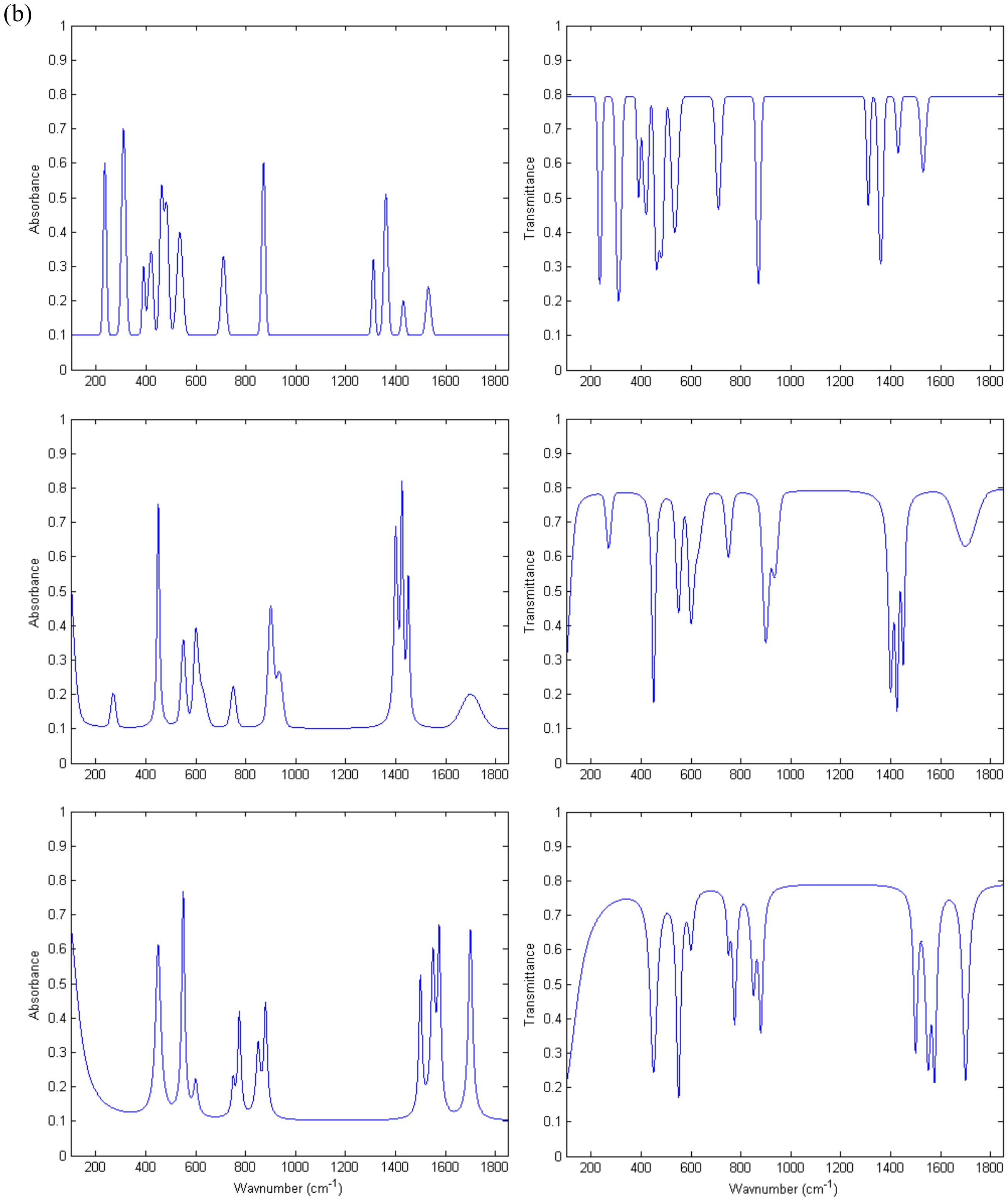


Figure 5.1. Artificial Raman spectra used in evaluation of spline fitting algorithms: (a) varying noise levels for the artificial liquid spectrum (SNR increases from left to right as indicated), and (b) noise-free absorbance (left) and transmittance (right) spectra.

5.1.2 Algorithms for Uniformly-Spaced Knots

Vibrational spectra can display considerable heterogeneity through rapid signal change (peaks and troughs) interspersed with noisy but otherwise slowly varying regions. As one might expect, this presented difficulties for many of the equidistant knot placement algorithms, and a number of these were quickly discounted due to poor performance. Ruppert's simple heuristic rule where $K = \min(n/4, 35 \text{ or } 40)$ is clearly not suitable for fitting spectra and other complex functions because of the heterogeneity and high sampling rate inherent in these datasets – far too few knots are placed to capture all the salient features. Relying on the GCV(K, λ) criterion also yielded poor results. GCV(K) frequently took on the shape of a mildly-oscillating, broad valley or basin, and significant trends in the residuals remained apparent over a considerable range along this valley. Selecting λ in this manner was also problematic, as min GCV(λ) often occurred in a shallow depression near $\lambda = 0$, yet the residual distribution actually improved when $\lambda >> 0$. Indeed, Jarrow et al. [46] showed that choosing λ in this manner can often result in under- or over-smoothing. Kauermann and Opsomer's [15] maximum log likelihood method, Equation (25), often failed because of the matrix $V_{K,\hat{\alpha}}$ becoming singular as K increased beyond a relatively small value prior to l_k reaching a maximum, as can be seen in Figure 5.2, and at a point where the spline fit was clearly still poor.

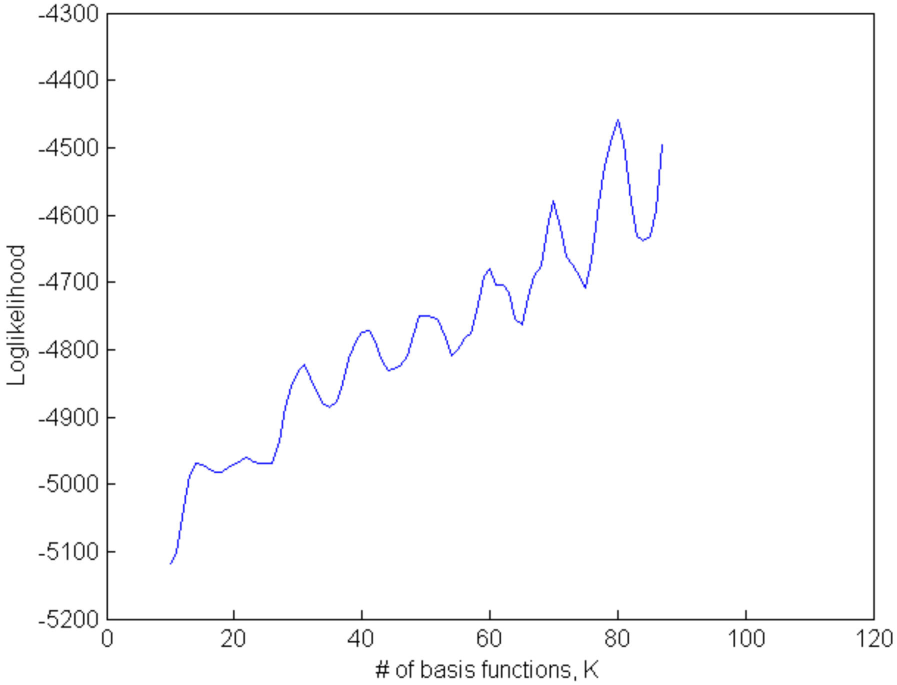


Figure 5.2. Kauermann and Opsomer's [15] maximum log likelihood as a function of the number of basis functions for a spline fit of a Raman spectrum. The log likelihood for this spectrum becomes undefined when $K > 85$.

Comparing the aforementioned techniques with that by Rowlands and Elliot [16], it seems clear that the appropriateness of a given objective function is highly data-dependent when using uniform knots. As the authors demonstrated, using the Anderson-Darling test as the objective function for choosing the number of equally-spaced knots is an effective means of automatically denoising spectra provided the noise distribution is known. However, simply using a large enough bases to capture the important features may be suboptimal in terms of goodness of fit and parsimony since it does not consider the shape of the spectrum. Indeed, tests on the MIR dataset showed evidence of under-fitting in some cases – see Figure 5.3*a* – that may cause issues with spectral interpretation. Furthermore, peaks show some evidence of shifting and attenuation and are also not explicitly identified. That is, because of the constant knot density, points in important signal regions are given the same weight as purely noisy regions.

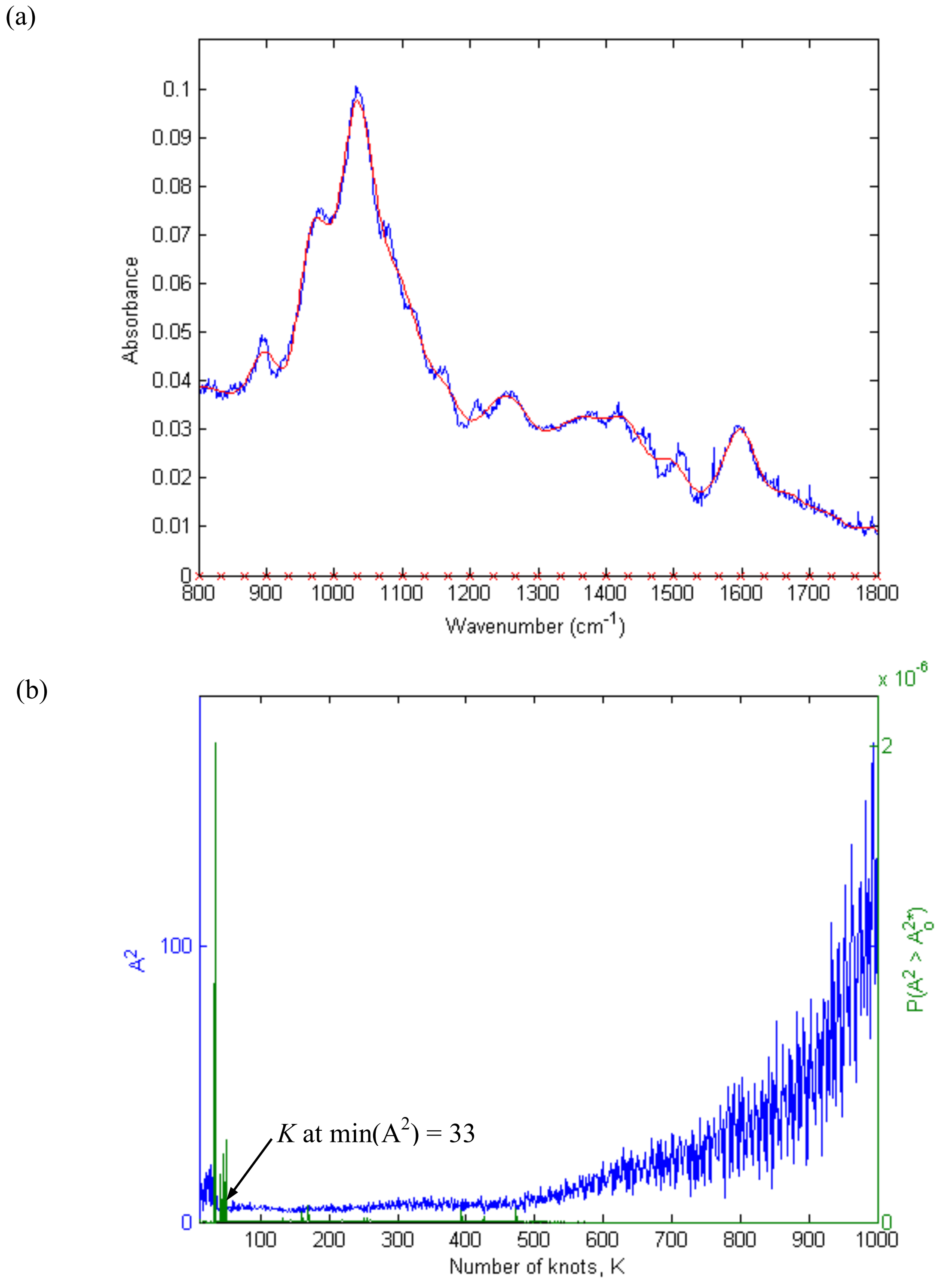


Figure 5.3. B-spline fit (red, smooth curve) to a biomass MIR spectrum (blue) with equally-spaced knots, location shown along the abscissa (a), the number of which was chosen by minimizing the Anderson-Darling test statistic A^2 (b).

5.1.3 Algorithms for Adaptive Knot Placement

The Bayesian MCMC approaches for spatially adaptive knot placement [20] and those employing adaptive smoothing penalties [22], [47] were found to be very slow and unreliable for the problems at hand, taking on the order of tens of minutes to resolve a knot basis, if solved at all. The R code for the approach by Krivobokova et al. [34] also failed to execute correctly for many of the spectra which were tested. Finally, given the unsuitability of Ruppert's heuristic rule in the present context, it was unsurprising that the method by Yao and Lee [24] also yielded poor results, even after multiple iterations, due to significant peak shifting and attenuation.

Tests using the algorithm by Li et al. [23] demonstrated a good balance between quality of fit and parsimony of knots for the three noise-free test functions and four of the six noise-free test spectra. Curiously, the algorithm produced poor fits in regions of minimally changing curvature, such as that between 1000 and 1400 cm^{-1} in Figure 5.1. Reevaluating the spline fit with roughness penalties on the second derivative did not improve the fit. Additional knots were thus added using Ruppert's heuristic rule (i.e., min(n/4, 35)) to the sparse regions containing knots (those originally placed by the Li algorithm) that were greater than 40 data points apart. This virtually eliminated the undulations without greatly increasing the total number of knots. Finally, spectra were also normalized along both dimensions prior to calculating curvature, since Li's method was designed for data having the same units (i.e., distance) in both axes. Because of its relative simplicity and rapid execution, the heuristic algorithm was selected for further refinement and testing for fitting splines to infrared spectra.

5.2 The Proposed Algorithm

The proposed algorithm – hereafter referred to as the “heuristic algorithm” – is based on the technique by Li et al. [23] combined with Ruppert’s heuristic rule and an optimized filtering step that has been tailored to the problem of spectral smoothing. The heuristic algorithm requires certain options to be set a priori, namely the type of denoising applied to the discrete curvature and any parameters associated with the denoising filter. The robustness of the method by Li et al. and flexibility of the implemented filtering step allow the heuristic algorithm to operate under a range of filter-specific parameters without a detrimental impact on performance. Once knots have been placed, the magnitude of the roughness penalty is obtained by minimizing GCV. However, tests on real data indicate that a roughness penalty is often not required, especially at high SNRs.

The following section details the optimization of the proposed heuristic algorithm with respect to the denoising step. As previously noted, the objective was to obtain an automated technique that is widely applicable to range of spectra while avoiding the need for end-user input. This goal was incorporated into the test data, which represents a range of spectral conditions, both real and synthetic. The steps to the heuristic algorithm itself are detailed in Appendix B.

5.3 Algorithm Optimization

5.3.1 Methodology

Factorial designs were used to optimize the denoising step in the heuristic algorithm by evaluating a variety of filtering techniques, described in the next section, on a test bed of the six artificial Raman spectra introduced in Section 5.1. Varying levels of Gaussian white noise were added to the transmittance spectra to produce samples with SNRs ranging from 30 to 100 in intervals of 10. Heteroscedastic noise conditions (i.e., the absorbance spectra) were generated via the Beer Transform. Robustness of the filters to data sampling rate was tested using four different spectral resolutions, ranging from 0.5/cm^{-1} to 1.25/cm^{-1} in increments of 0.25/cm^{-1}, such that each spectral sample was made up of 1400 to 3500 data points.

No explanation was provided by Li et al. [23] as to why their algorithm was not tested by first denoising the shape profile prior to calculating its discrete curvature; presumably, part of the reason may have been since some profiles of interest could not be represented as explicit functions, which cannot be processed using one-dimensional filters. Infrared spectra are continuous along one dimension, however, so optimization of the algorithm also involved testing the sequence in which denoising was done. In other words, the question was, "is it better to denoise the spectrum first, and then use an unfiltered discrete curvature, or to only denoise the curvature?" The test sequence is presented schematically in Figure 5.4.

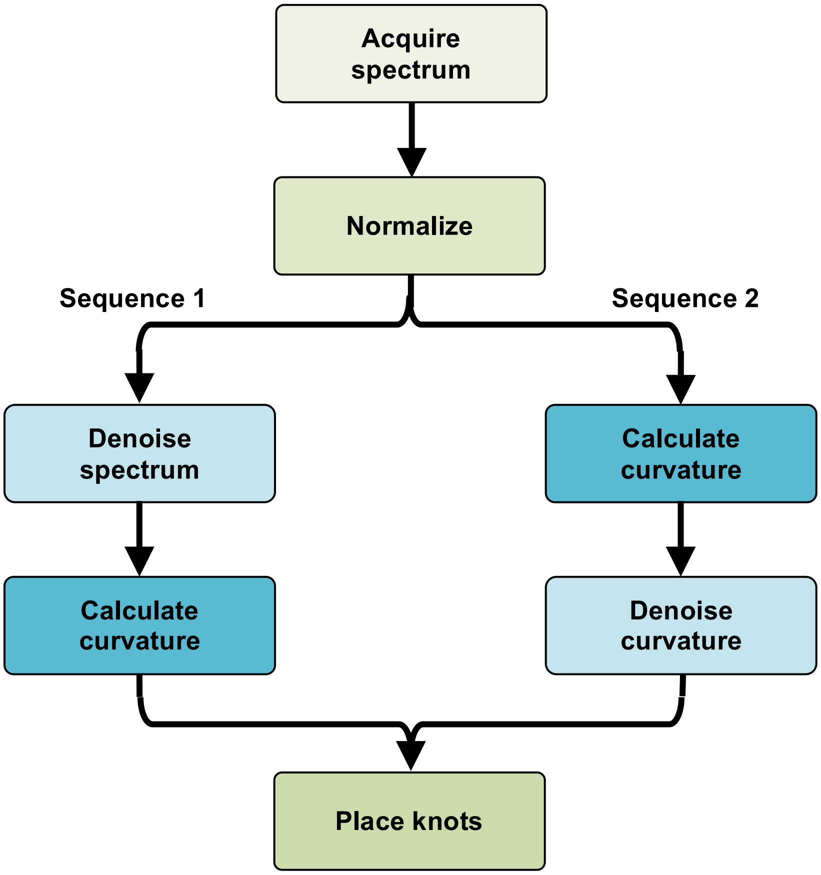


Figure 5.4. Process for testing effect of denoising sequence on knot placement in the heuristic algorithm by Li et al. [23].

Goodness of fit was quantified by the corrected Akaike Information Criterion (AICc) for four fitness functions: the RMSEs of residuals, peak shifts, differences in peak integral area, and differences in full width at half maximum (FWHM), each holding the role of the likelihood function penalized against the number of knots. Thus, each run of the factorial design ultimately had six replicates with four responses. To reduce the possibility of distorted results under very noisy conditions, peak integral area and FWHW criteria were only considered for isolated peaks; that is, those for which the FWHM was clearly defined. Integral area was calculated for the argument ranges delimiting the FWHW.

Given the multiobjective nature of the problem, the "best" parameters for each denoising technique were selected by identifying Pareto-optimal points (i.e., Pareto-optimal over the factorial design conditions) and examining Y by X group plots (i.e., the responses versus explanatory variables). Preference was given to minimizing RMSEs of residual and peak shift over FWHM and peak area. Individually optimized denoising methods were compared against each other to arrive at an overall best approach.

Finally, performance of the optimized Li algorithm was evaluated against the method of Rowlands and Elliot [16] using the predictive power of FLR and FPCR models, built from real MIR and NIR datasets, as metrics. The best of these models were then compared with their discrete PLS counterparts.

Computational Details

Simulations and spline fitting were done in Matlab v. 7.12.0 (The MathWorks, 2011), using Professor Ramsay's Functional Data Analysis Toolbox, various functions provided by the Matlab user community, and custom-written Matlab code. Statistical analyses were performed in Matlab. Runtime benchmarks were obtained by executing programs with ten replicates apiece in four different operating environments (OEs). Three were in Windows 7 (64-bit), running on

- OE1: a quad-core Intel Core i7 2600 (3.4 GHz) processor;
- OE2: a quad-core Intel Core i7 860 (2.8 GHz);
- OE3: a MacBook Pro with a dual-core Intel Core i5 2435M (2.4 GHz) in Parallels™ virtualization;

and the fourth (OE4) using the same version of Matlab for Mac in OS X 10.7.3 on the same MacBook Pro.

5.3.2 Denoising Techniques

While spline smoothing itself is a demonstrably good tool in many cases, it is clear that many knot placement algorithms would not fare well against the amplified noise and highly oscillatory nature of the discretized curvature of an already noisy signal. It was therefore deemed most appropriate to evaluate a number of filters designed for the removal of high-frequency noise. Three forms of wavelet filtering and four other common denoising techniques – used by Rowlands and Elliot [16] for comparison with their algorithm – were tested. Moving average, locally weighted scatterplot smoothing (LOESS), Savitzky-Golay filtering, Butterworth filtering, discrete wavelet transform, wavelet packets, and stationary wavelet transform comprised the test suite for optimization. As noted in Chapter 4, these methods are also commonly applied to denoising spectra as part of typical chemometric preprocessing routines.

5.3.2.1 Moving Average

As the name implies, moving average smoothing simply replaces each data point with the average of the surrounding points. The number of points (always odd and centered around the point of interest) is defined as the window size. Naturally, more aggressive smoothing is achieved by increasing the window, which was herein varied from 3 to 351 in increments of two.

5.3.2.2 Locally-Weighted Scatterplot Smoothing (LOESS)

Locally weighted scatterplot smoothing (LOESS) involves fitting a low-degree polynomial at each point in the data set, using weighted least squares regression on a small subset of data centered at said point. Decreasing weight is applied to points further from the center. The subset size, or span, is specified as a percentage of the total number of data points. Curvature denoising with LOESS was done using a second-degree polynomial and spans varying from 0.2 to 0.7% in increments of 0.05%.

5.3.2.3 Savitzky-Golay Filtering

Savitzky-Golay smoothing is similar to LOESS except that the least squares regression is unweighted. The polynomial order and span were respectively set at two or four and from 3 to 351 in increments of two. As might be expected, span size is inversely proportional to the amount of smoothing, with small windows producing noisier signals but larger ones resulting in more signal distortion. The opposite is true for polynomial degree.

5.3.2.4 Butterworth Filtering

Lowpass Butterworth filters remove frequencies below a specified cutoff value, possess maximally flat frequency responses in the passband, and are monotonic in the pass- and stop bands. Filter order and normalized cutoff frequency were varied from 3 to 7 in increments of one, and 0.01 to 0.95 in steps of 0.01, respectively. The effects of these parameters on filter frequency response (the gain and phase shift) are demonstrated in Figure 5.5. The signal was processed in both forward and reverse directions in order to preserve phase.

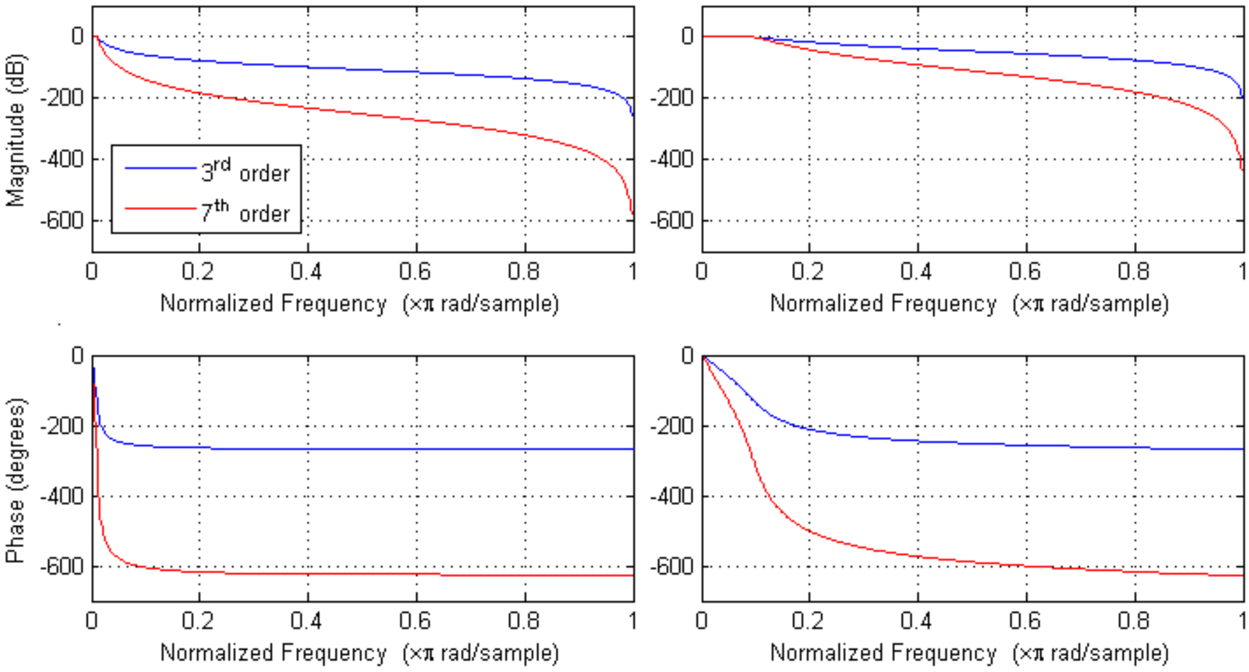


Figure 5.5. Bode plot contrasting 3rd (blue) and 7th (red) order Butterworth filters with normalized cutoff frequencies of 0.05 (left) and 0.10 (right).

5.3.2.5 Wavelet Filtering

Wavelets, essentially short-lasting waveforms, are another basis for representing data in the functional context and, because of their flexible properties and shape, are eminently suited to the problem of spectral denoising[42]. Most wavelets, such as the Daubechies 12 shown in Figure 5.6, have no continuous, closed-form representation, and can only be evaluated discretely.

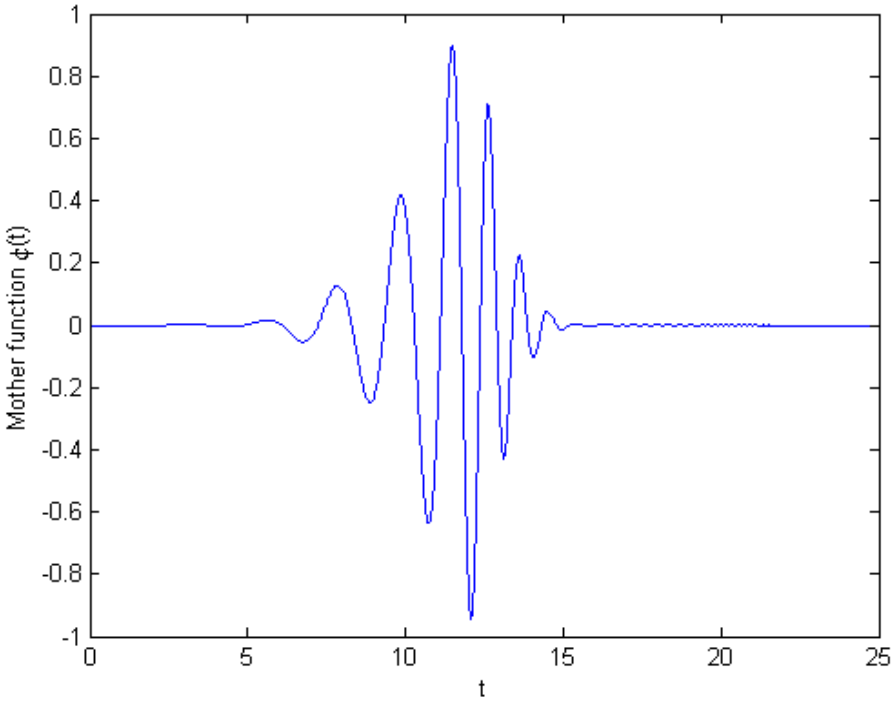


Figure 5.6. Discrete-form representation of a Daubechies 12 wavelet.

As the name implies, wavelet denoising is designed to recover a signal s, from noisy data x, where, similarly to Equation (1), the model is assumed to be

$$x(t) = s(t) + \sigma\varepsilon(t), \tag{63}$$

with equally spaced sampling points t_i (i = 1, 2,…, n). The three tested wavelet methods are outlined in brief below.

Discrete Wavelet Transform (DWT)

The simplest form of wavelet denoising is the discrete wavelet transform (DWT), which involves convolving the signal with the time-reversed high- and low-pass filters associated with the wavelet, followed by dyadic decimation (downsampling), and subsequent reconstruction. First, the signal is orthogonally decomposed by highpass and lowpass decomposition filters into high-frequency wavelet detail D_1 and low-frequency approximation A_1 coefficients. The vector A_1 is then decimated (halved in length) and again convolved with the wavelet filters to produce the next level of detail D_2 and approximation A_2. Lower levels thus contain higher frequency (often mainly noise) components of the signal, whereas higher levels capture broader trends.

Decimation occurs by removing either even- or odd-indexed data points (in Matlab, DWT decimation discards the odd elements). Denoising then proceeds by thresholding the detail coefficients. Small coefficients below the threshold are discarded and, depending on the thresholding rule, larger ones may be attenuated, the notion being that small coefficients are entirely noise [6]. Finally, the denoised signal is reconstructed using the thresholded coefficients by passing the detail and approximation coefficients through reconstruction highpass and lowpass filters, respectively (the inverse discrete wavelet transform).

Wavelet Packets (WP)

An alternative to the DWT is wavelet packet decomposition, or simply wavelet packets (WP). This method extends the DWT by filtering the details in addition to the approximations at each level. The result is a complex tree of many candidate bases, the optimal of which is a denoised signal of minimum entropy relative to the rest. The idea of entropy in a signal, or more broadly in information theory, is based on the original concept in statistical thermodynamics – a measure of system disorder or uncertainty. Signal entropy can thus effectively be thought of as noise, since this is what precludes us from knowing the exact (expected) value of the signal. While information entropy can be defined in various ways, the default used in Matlab for WP denoising is the Steins Unbiased Risk Estimate (SURE) entropy criterion, defined as

$$E(s) = n - \sum_{i}^{n} E_S(s_i) + \sum_{i}^{n} \min(s_i^2, p^2) \tag{64}$$

where $E_S(s_i) = 1$ if $|s_i| \leq p$ and 0 otherwise for a threshold p.

Stationary Wavelet Transform (SWT)

The decimation step in both DWT and WP denoising precludes these methods from being shift- or translation-invariant, meaning that the denoised reconstruction will change if the input signal is translated [49]. The undecimated discrete wavelet transform, also known as the stationary wavelet transform (SWT), overcomes this shortcoming (at the cost of some additional computation), yielding better and more robust denoising performance in applications such as image and spectral processing where phase shifts may have a significant impact on quality and usefulness [49]. Instead of decimating the signal as in DWT and WP, the SWT operates by stretching the wavelet filters by a factor of two at each level. In this way aliasing is avoided and the signal remains intact.

Wavelet Parameters

The choice of parameters for wavelet denoising was made considering previous works on wavelet denoising of spectra [42], [50] and recommendations in the Matlab documentation. Twenty-two candidate wavelets were evaluated from the families available in Matlab that satisfy the properties of compact support and orthogonality: Daubechies (Db), Symlets (Sym), and Coiflets (Coif), which are widely-used in denoising applications. A given wavelet is referenced by its family followed by the order *m*, or the number of initial points used to build the wavelet filter. The more points, the more complex the wavelet. In the present experimental design, wavelets were specified from Sym4 to Sym10 and Coif1 to Coif5 in increments of one, and Db2 to Db20 in increments of two.

The number of decomposition levels J can be user-specified but is ultimately limited by the signal length n, which must be divisible by 2^J, to at most $\log_2 n$ levels. However, the maximum useful level depends on the wavelet order and is defined by the relation

$$2^J < \frac{n}{(m-1)} \tag{65}$$

Decomposition levels were varied from one to eight or the maximum useful level, whichever was less. Input signals were truncated where necessary in order to be divisible by 2^J, although signal extension may also be used in real-world applications. The large number of points in each spectrum ensured that no significant portion of the signal was lost due to truncation. Additional parameters that were tested are outlined in Appendix A1.

5.4 Results

Figure 5.7 compares the relative performance of spline fits obtained by the Rowlands and Elliot [16] (R&E) approach and the various denoising techniques in the heuristic algorithm, using AICc of residuals as the metric. The trends are essentially the same for the other objective functions (peak shift, peak integral area, and full-width at half-maximum) and for different sampling frequencies. Considering only these results, it would be easy to focus on the Butterworth filter and wavelet methods and discard the R&E method completely, but the nature of the spline-fitting problem constrains the usefulness of AICc such that other factors must also be taken into account.

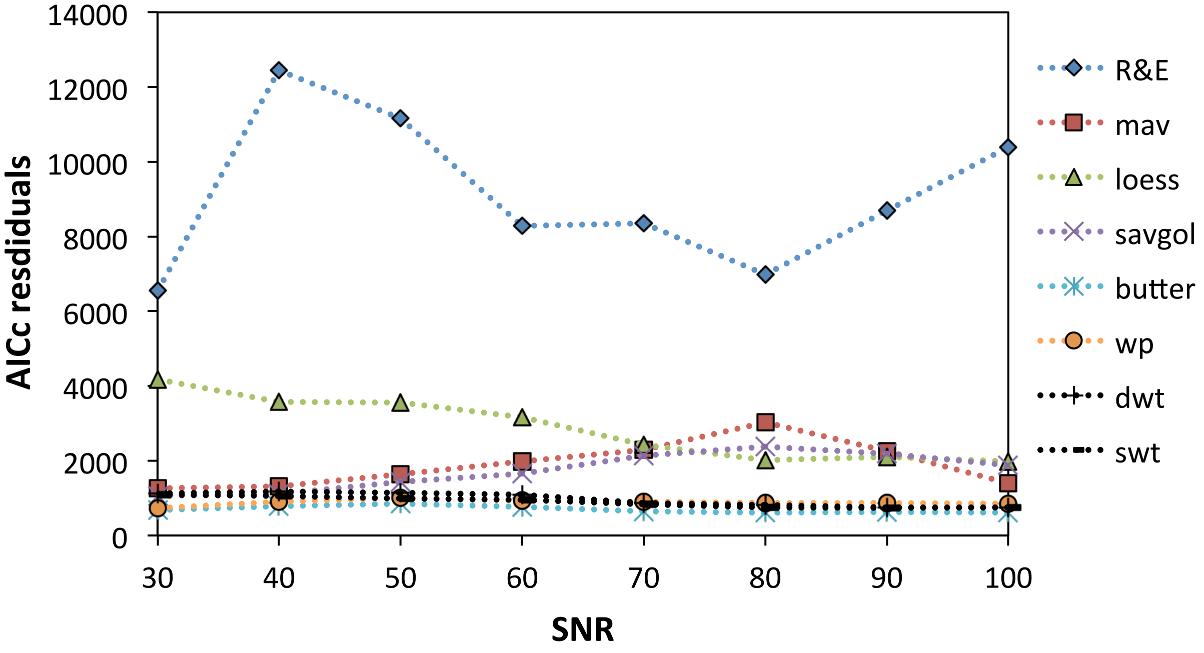


Figure 5.7. Quality of spline fits by SNR, comparing the method by Rowlands and Elliot **[16]** to various denoising methods used in the adaptive knot placement algorithm.

The outlying AICc values for the R&E method are primarily due to very large knot vectors, where τ was 50% or more the value of n. While peaks were not significantly attenuated or shifted, the large number of basis functions resulted in severe noise-fitting, effectively defeating the purpose of the spline fit. While *P*-splines with saturated bases (i.e., $\tau = n$) are often recommended to avoid the problem of knot placement [51], it is evident that using them here is not ideal. Indeed, a roughness penalty large enough to iron out spurious oscillations simultaneously caused considerable attenuation of real peaks. Based on the simulation study, the R&E approach, as with other equally spaced knot placement algorithms, is not well suited for smoothing functions with very heteroscedastic features. However, the R&E method nonetheless merits further testing using real-world data, since it is demonstrably effective in denoising smoother types of spectra [16].

The relatively high AICc values from moving average, LOESS, and Savitzky-Golay implementations were a product of reasonably parsimonious models hampered by poor spectral reproduction due to inadequate knot placement. Not only were RMSE metrics quite large, but apart from LOESS, filter parameters were highly inconsistent across SNR and sampling frequency. That is, Pareto-optimal settings obtained from the factorial design – particularly filter span – correlated poorly with the characteristics of the simulated samples, which would have made it difficult to reliably apply these methods to real data sets. The optimal filter settings across the range of SNR and sampling resolution are presented Appendix A2 for Butterworth, SWT, and WP implementations of the heuristic algorithm.

Butterworth filtering performed remarkably well in most areas, with uniformity of AICc across SNR and sampling frequency, as shown in Figure 5.8; constancy of filter parameters under the varying run conditions; and mostly faithful reproductions of the simulated spectra. Of the 32 runs, one returned an optimal filter order of 4, four of order 5, and the remainder distributed fairly evenly between 6 and 7. All but two runs done at the same sampling resolution as the real MIR data set (1 cm^{-1}) returned order 6. Normalized cutoff frequency remained unchanged at 0.02 except for three runs at 0.01. Denoising Sequence 1 was optimal in all cases.

Knot placement was not always ideal, however. The Butterworth filter was unable to discriminate between very closely adjacent peaks, with the result that they were either merged or severely attenuated in the spline fit. Examination of discrete curvatures and resulting knot vectors (not shown) obtained from different filter parameters indicated that changing filter order between 5 and 7 had a minimal impact on spline fit quality. Increasing the cutoff frequency to 0.05 improved peak-fitting at the expense of noise-fitting in other regions.

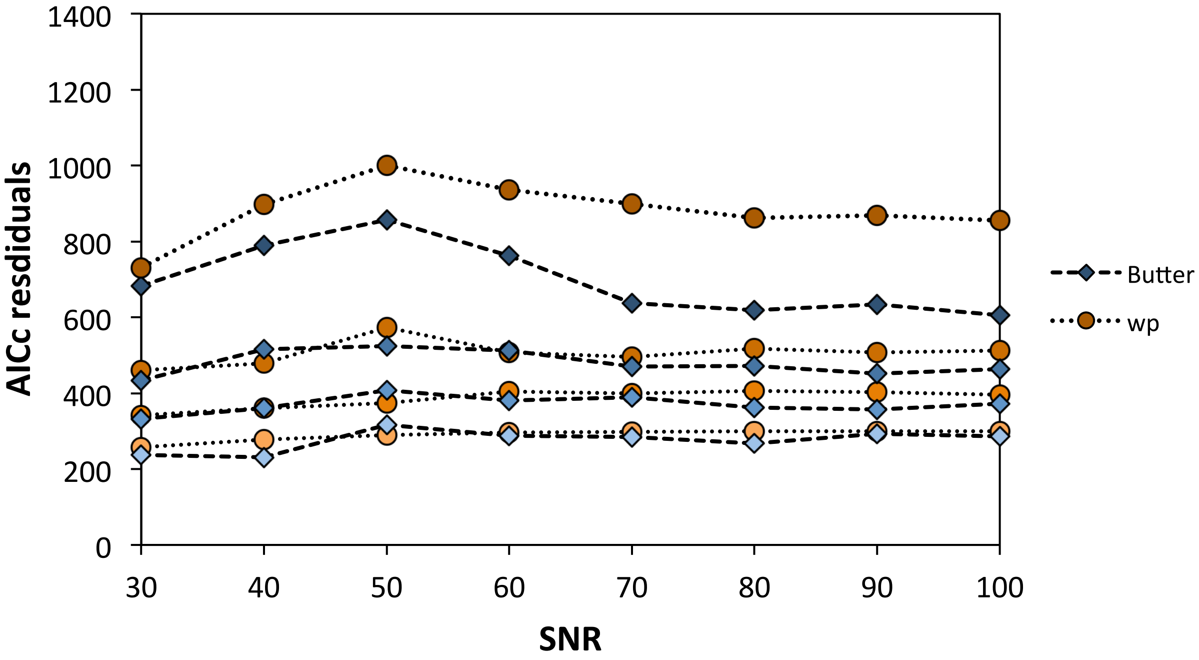


Figure 5.8. Quality of spline fits by SNR, comparing Butterworth with wavelet packets filtering for the heuristic knot placement algorithm over varying SNR and spectral resolutions. Sampling resolution decreases from 0.5 cm^{-1} to 1.25 cm^{-1} from top to bottom.

Given their locally adaptive nature, it is perhaps unsurprising that the wavelet methods gave the best balance between model parsimony and accurate spectral reproduction. The DWT and SWT yielded AICc values that were quite close to, and occasionally smaller than WP or Butterworth filtering. Nevertheless, both DWT and SWT methods were somewhat less robust with regard to constancy of filter parameters; for this reason and for the sake of clarity they are not shown in Figure 5.8. Though knot placement via WP denoising resulted in the overall best knot vectors, some spline fits still suffered from sporadic sharp oscillations resembling Runge's phenomenon in regions where curvature of the true signal was very low. Given their form, wavelets have some difficulty in approximating broad regions where the true signal is actually very smooth, so more inadequate knot placement can be expected here. Nevertheless, application of a small roughness penalty on the second derivative, chosen by minimizing GCV, successfully eliminated these waves, yielding more accurate smoothing of the data (see Figure 5.7) without significantly impacting other goodness of fit measures such as peak integral area. Optimum smooths were achieved with soft thresholding, keeping the approximation coefficients, a decomposition level of six, and Sequence 1 denoising. Daubechies wavelets ranging from order 12 to 20 order were optimal for all but two runs, which returned 6th and 7th order Symlets. Daubechies of order 20, 16, and 14 were Pareto-optimal across all SNRs for artificial data respectively sampled at a rate of 0.75, 1, and 1.25 cm^{-1}. Orders were distributed from 12 to 18 for the 0.5 cm^{-1} data, but there was no clear correlation between wavelet order and SNR or resolution. Evaluating various wavelet orders showed that they generally had little impact on spline fit quality, however – a testament to their flexible and adaptive nature.

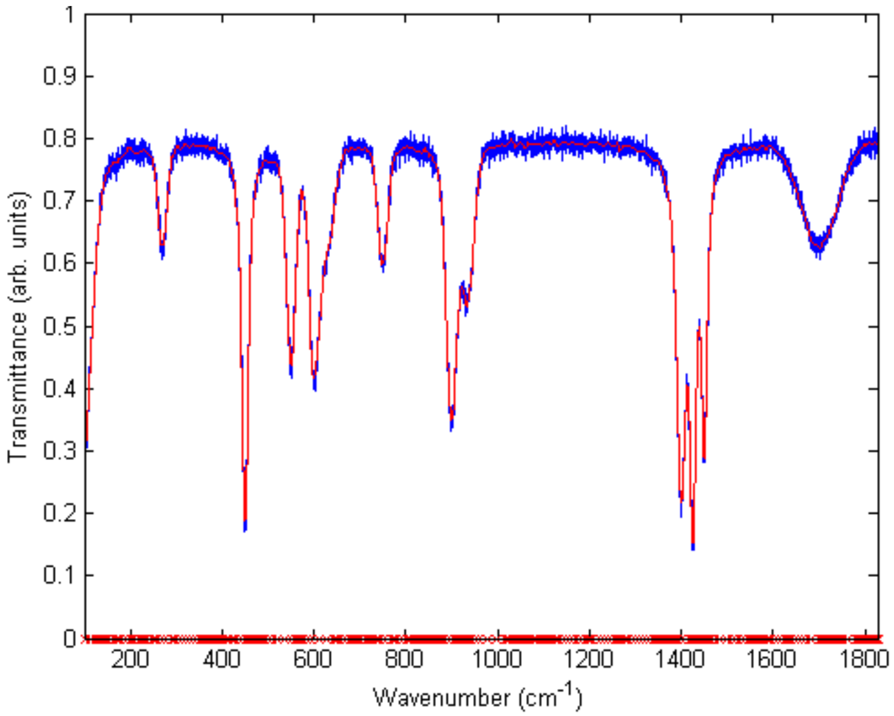


Figure 5.9. Simulated Raman spectrum of a liquid: SNR = 40, sampling resolution of 0.5 cm^{-1} (3500 points). B-spline fit with 383 knots, locations shown along the abscissa, and a roughness penalty of $\lambda = 7.8$.

Unfortunately the foregoing exercise in optimization cannot be so readily applied to the real world – spectra with very high SNRs (obtained by increasing the number of co-adds, or scans) may be unavailable, and the "true" functions of real spectra are, after all, unknown. Instead, the quality of regression models may be used as goodness of fit metrics, and results from simulated optimization can be used as guidelines for selecting parameters for denoising real spectra. Performance using real-world data is discussed next in Chapter 6.

Chapter 6

The Real World: Functional Models in Practical Applications

6.1 Functional Regression and Partial Least Squares on Chemical Spectra

Functional regression was compared against traditional PLS techniques using two datasets: the MIR spectra of biomass samples used to predict polysaccharide composition provided by Krasznai and the publically-available diesel set used by Saeys et al. [2] referred to in Section 4.3. As noted earlier, it is computationally much easier to perform functional multivariate analyses when all observations share the same basis, since the problem reduces to straightforward matrix operations on the basis coefficients. Given the considerable additional time and effort which would be required for implementing code capable of supporting different knot vectors for each observation, models were built using spline fits sharing the same basis. The process and rationale for common knot vector selection is detailed in Section 6.1.2.

6.1.1 Methodology

6.1.1.1 Biomass MIR Data

Twenty-four different functional regression models, summarized in Table 6.1, were tested and compared against the optimum PLS model developed by Krasznai [44], who generously provided the MIR data and analytical results for use in the present investigation. The effect of knot placement was tested using four different spline bases: two obtained by the R&E algorithm, one using a Butterworth implementation of the heuristic approach, and another using wavelet packets denoising.

Table 6.1. Summary of functional regression models and the parameters adjusted in each for predicting fractional composition of cellulose, xylan, and lignin in plant biomass samples.

Parameter	Variations
Type of regression	• Functional Linear Regression – Zero-intercept – Non-zero intercept • Functional Principal Components Regression – 3 or 4 principal components
Knot vector	• 34 equally-spaced knots • 130 equally-spaced knots • Free knots using heuristic algorithm with Butterworth filtering: – Sequence 1 denoising – 7th order – Normalized cutoff frequency of 0.02 • Free knots using heuristic algorithm with wavelet packets filtering: – Sequence 1 denoising – Soft thresholding – Keep approximation coefficients – Db16 wavelet – Decomposition level of 6
Roughness penalties	• On observations: $\{\lambda \vert \lambda \in [0, 10^{10}]\}$ • For FLR, on the regression coefficient: $\{\lambda_\beta \vert \lambda_\beta \in [0,10]\}$ • For FPCR, on the principal components: $\{\lambda_{PC} \vert \lambda_{PC} \in [0,10]\}$

In his work, Krasznai [44] used MIR spectra of surrogate mixtures with known compositions of three plant polysaccharides – cellulose, xylan, and lignin – to develop PLS models for predicting the composition of real plant matter as a means of bypassing costly, destructive, and time-consuming chemical lab analyses. The PLS study involved evaluating combinations of various discrete spectral preprocessing methods, some of which were referred to in Chapter 4, to identify an optimal approach for improving the quality of the predictive models. Root mean square errors of cross-validation (RMSECV), obtained by seven-fold CV, and of prediction (RMSEP) were used as metrics for choosing the number of principal components and assessing model quality. The same CV technique, in which observations are grouped such that group 1 is observations 1, 8, 15, ..., group 2 is 2, 9, 16, ... and so on, is applied herein.

The MIR dataset comprised 35 spectra in all: a training set of the 28 surrogate mixtures whose compositions were prepared according to a ternary mixture experimental design (covering weight fractions from 0 to 1 for each compound); an external validation set of five surrogate mixtures; and two samples of real plant matter, designated B10 and C10, whose compositions were determined by wet chemical analysis. In order to reduce noise and distortion from external factors, each spectrum was obtained by averaging the spectra (each of which was the summation of 64 individual scans, or co-adds) of five replicate aliquots per mixture. Krasznai [44] identified the wavenumber range from 800 to 1800 cm^{-1} as being most representative (i.e., having the most predictive power) for these types of samples. In his study, the most accurate predictions of B10 and C10 composition were obtained using a PLS model with three latent variables using third-order polynomial Savitzky-Golay smoothing to calculate second derivatives of the mean-centered and unit variance-scaled spectra.

6.1.1.2 Diesel NIR Data

Apart from some differences in the choice of parameters, the same variety of models outlined in Table 6.1 was also tested on the diesel NIR–cetane number dataset, available at http://www.eigenvector.com/data/SWRI/index.html, and compared against the FLR and PLS models developed by Saeys et al. [2]. The NIR spectra comprised a set of 133 training and 112 validation samples, obtained at a resolution of 2 cm^{-1} from a wavenumber range of 750 to 1550 cm^{-1}. With the lower sampling rate and (more likely because of) a very large number of co-adds, the spectra appear to be almost noise-free. For this reason, the heuristic algorithm was implemented without the curvature denoising step. The effect of free knot placement and equally-spaced knots was compared using two pairs of knot vectors: two for which K = 15, matching the dimension of the RMSECV-minimizing basis used by Saeys et al., and two with K = 124, the number returned by the heuristic algorithm. Regression parameters (roughness penalties, number of principal components) were selected automatically by five-fold random CV. Saeys et al. used five-fold CV with contiguous blocks, but it is unclear in their paper how exactly this was implemented (particularly since the training set has 133 samples). The FLR and PLS models they presented were thus reconstructed herein, and the RMSECVs confirmed via Monte-Carlo simulation with 100 replicates. The most accurate predictions were obtained by a six-component PLS regression and an FLR model without any roughness penalties.

6.1.2 Results

6.1.2.1 Spectral Smoothing

Biomass MIR Spectra

The R&E algorithm returned considerably different knot vectors depending on the spectrum, as can be seen in Figure 6.1. R&E-optimal bases for the observed MIR spectra most frequently fell below 50 knots, with the result that many perhaps-significant features were either significantly attenuated or even annihilated – see Figure 6.3*a* for an example. Fits with knot vectors the dimensional range of 100 to 150 produced the most visually pleasing fits (e.g., Figure 6.3*b*), while anything higher appeared to excessively follow noise. Two models were thus constructed, one using the median of the entire set (τ = 34), and the other where τ = 130, or the average of the subset between 100 and 150. The median was selected as opposed to the mean of the entire set, since the former is more representative of the skewed distribution.

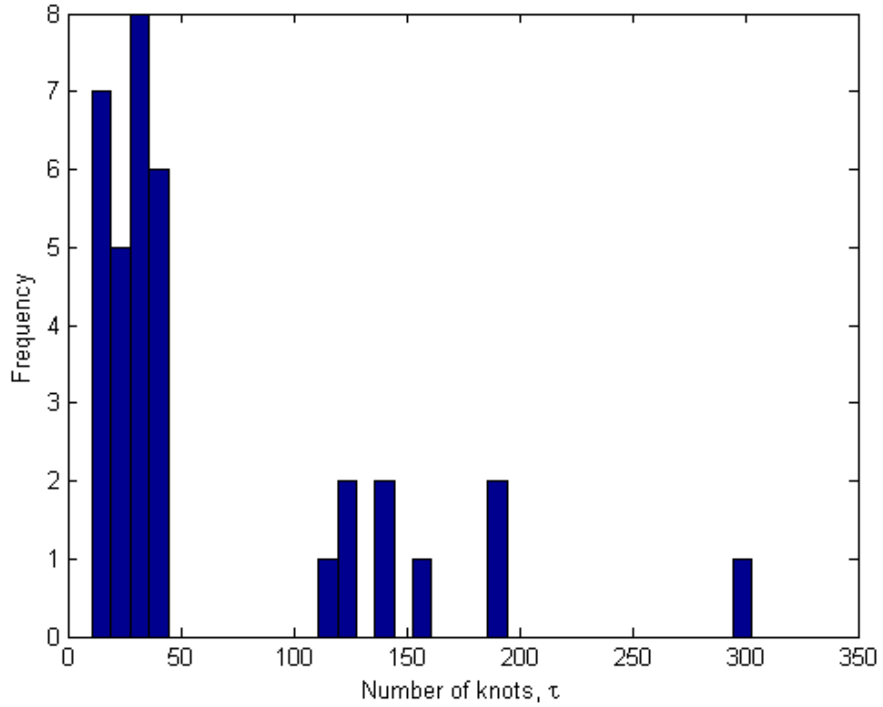


Figure 6.1. Histogram of number of knots *τ* per MIR spectrum for spline fits of MIR spectra returned by the Rowlands and Elliot [16] algorithm.

The dimension of knot vectors output by the optimized heuristic algorithm were more consistent across the MIR spectra, in fact appearing somewhat normally distributed when using a Butterworth filter (Figure 6.2*a*), and chi-squared distributed with WP filtering (Figure 6.2*b*). A single "average" knot vector was obtained by first concatenating all individual knot vectors, then choosing the top τ wavenumbers at which knots were most frequently placed. The choice of τ was simply equal to the median length of all the knot vectors: 111 for the Butterworth case and 133 for WP. The resulting smooths for a single MIR spectrum are presented in Figure 6.3. The knot locations – indicated along the abscissa – show grouping around wavenumbers identified by the algorithm to be, in terms of frequency of occurrence, the most significant. Unlike a consistent basis of equally-spaced knots, however, the use of a frequentist free knot vector introduces bias by over-fitting or partly ignoring features that may be unique to certain spectra. For example, the region around 1500 cm^{-1} was captured well in the Butterworth fit but not by the WP knot vector. On the other hand, the region around 1180 to 1280 cm^{-1} was over-smoothed in both Butterworth and WP implementations.

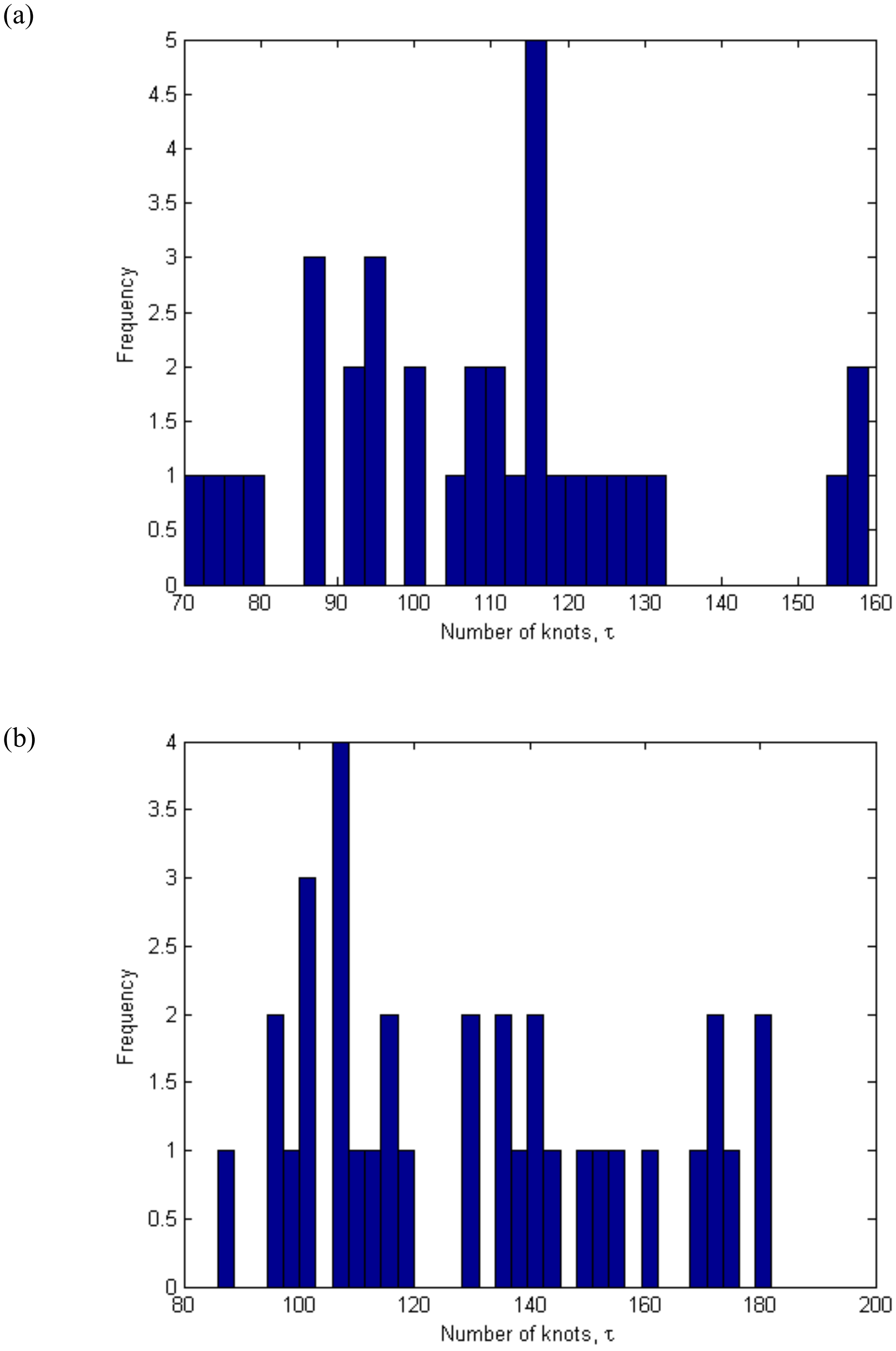
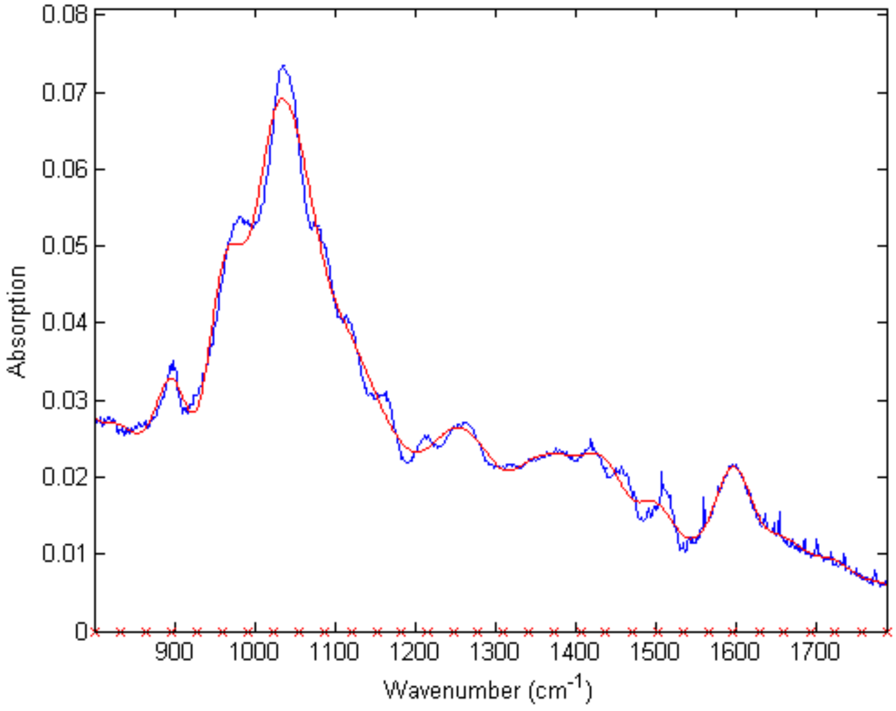
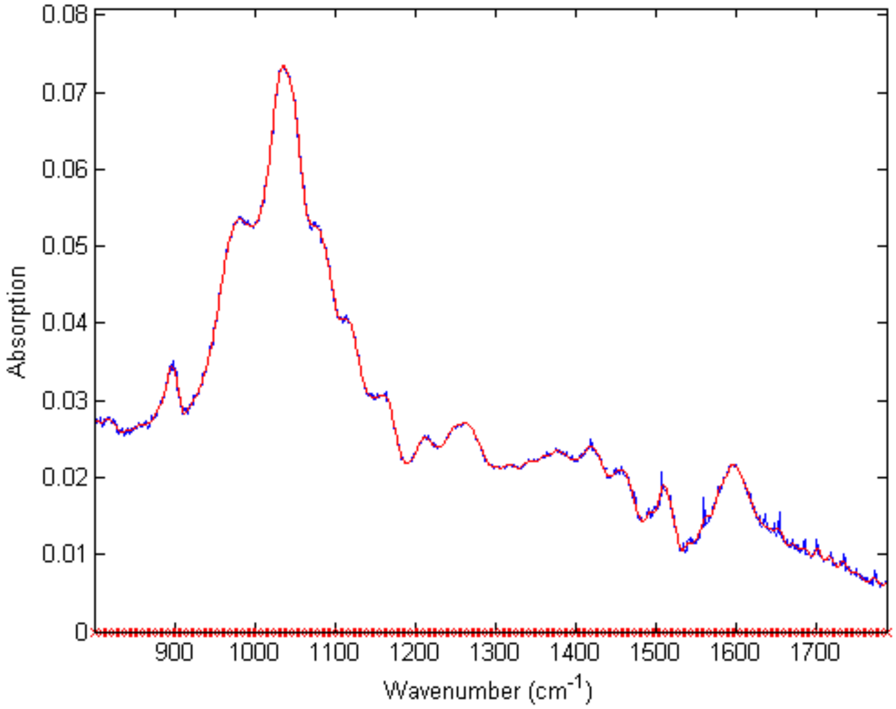


Figure 6.2. Histograms of number of knots per spectrum for spline fits of MIR spectra returned by the optimized heuristic algorithm using (a) Butterworth and (b) wavelet packets filtering.

(a)


0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Absorption
900
1000
1100
1200
1300
1400
1500
1600
1700
Wavenumber (cm^{-1})

(b)


0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Absorption
900
1000
1100
1200
1300
1400
1500
1600
1700
Wavenumber (cm^{-1})

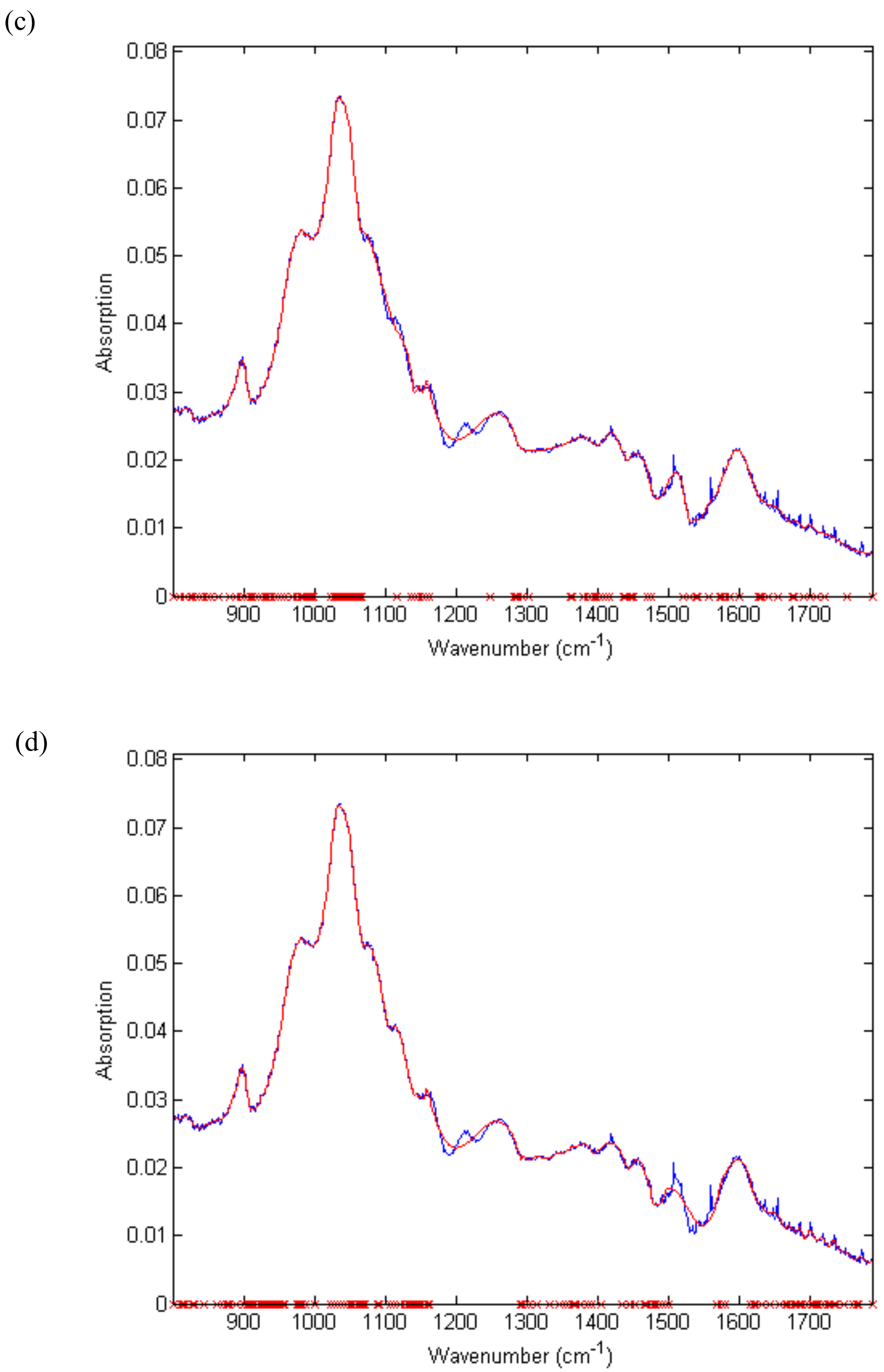


Figure 6.3. B-spline fits to a biomass MIR spectrum with (a) 34 and (b) 130 equally-spaced knots using the R&E approach, and (c) 111 knots placed using the heuristic algorithm with Butterworth filtering and (d) 133 with wavelet packets.

Diesel NIR Spectra

The essential lack of noise in the NIR spectra resulted in unfavorable knot vectors for both the R&E method and various filter implementations of the heuristic algorithm. More than half the fits returned by the former were roughly normally distributed between 310 and 360 knots, defeating the purpose of significant dimensionality reduction. Coincidentally, the R&E algorithm also returned $\tau = 16$ as the optimal dimension for 120 of the spectra, just one more than the RMSECV-minimizing 15 knots used by Saeys et al. [2]. Spline fits from the heuristic algorithm using a range of different filter settings obfuscated or severely attenuated many of the significant spectral features. Indeed, inspection of the raw discrete curvature, with an example pictured in Figure 6.4, confirms that little further denoising is actually required. The large spikes in the curvature are also mostly located in slowly varying regions (where noise is more noticeable) that are not particularly elucidative of the underlying physical properties.

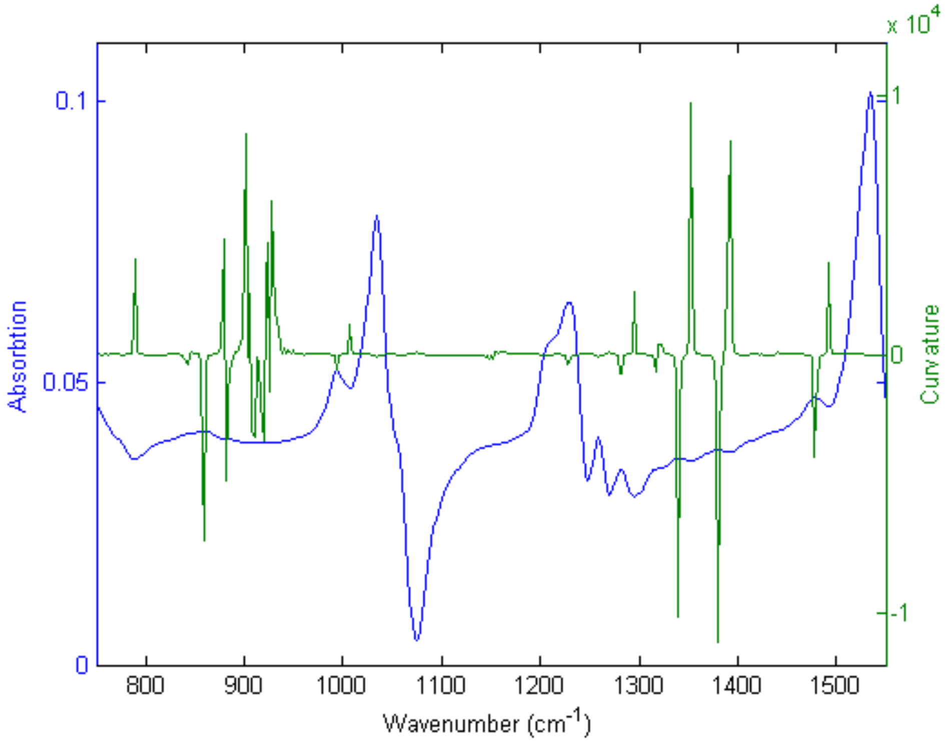


Figure 6.4. Raw NIR spectrum of a diesel sample overlaid on its discretized curvature.

The increased knot density around these noisier regions has little detrimental impact in terms of analyzing individual smoothed spectra, since it does not significantly affect smoothing of actual peaks (i.e., height, location, integral area, etc.), but it could nonetheless deteriorate the quality of regression models solved using basis function expansions because of highly-correlated coefficients in areas of little interest. In order to better evaluate the utility of free-knot vectors, models constructed from the frequentist heuristic knot vector were compared against those using an equally-spaced basis of the same dimension. In this case, inspection of the splines from both knot vectors, examples of which are shown in Figure 6.5, and the corresponding residuals suggest better spectral reproductions were obtained using uniformly-spaced knots. The comparative performance of regression models is discussed next.

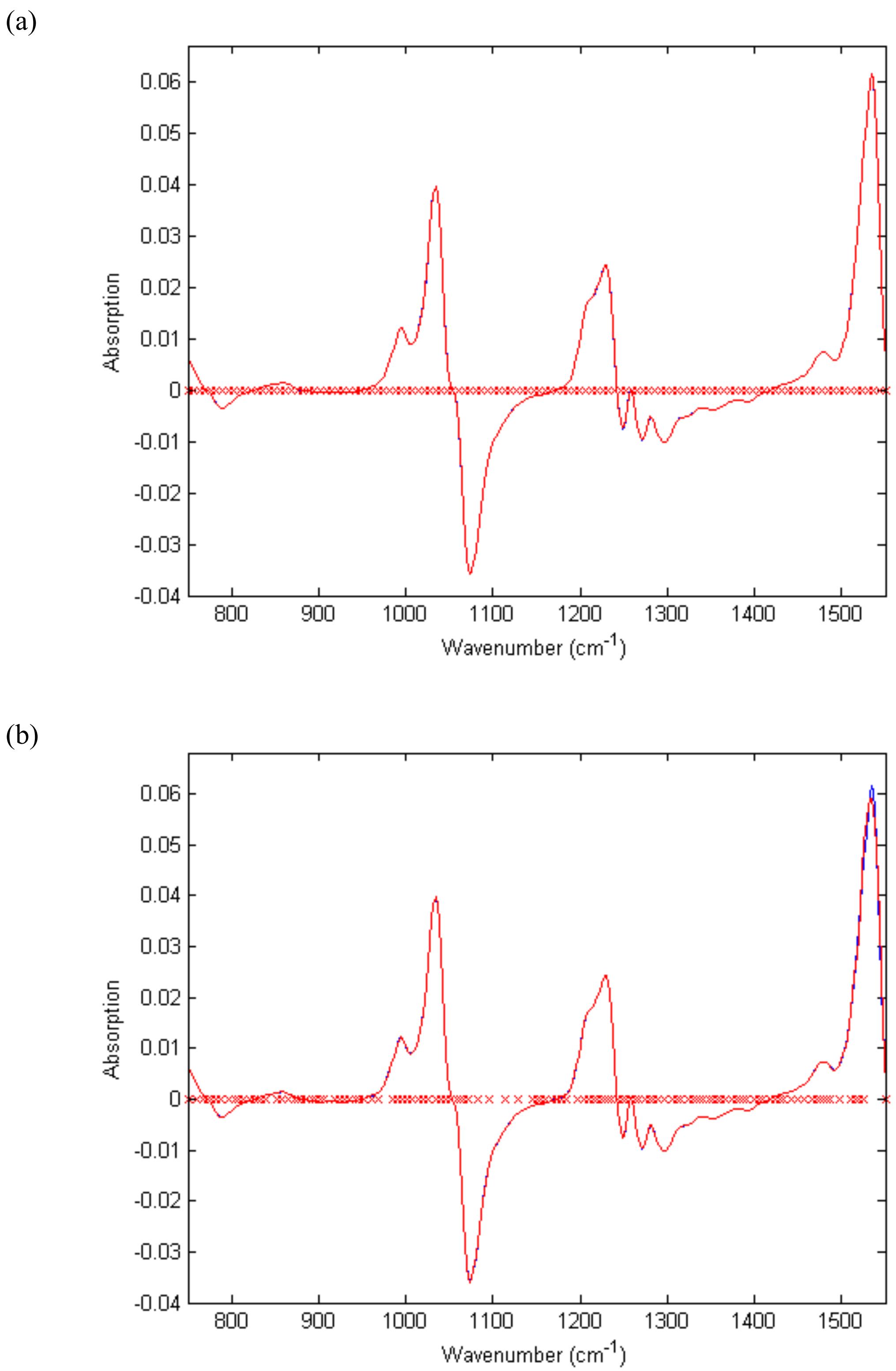


Figure 6.5. B-spline fits to a Diesel NIR spectrum with (a) 124 equally-spaced knots and (b) 124 placed using the heuristic algorithm with no curvature filtering. Note the minor attenuation of the peak at 1534 cm^{-1} in (b).

6.1.2.2 Functional Regression Models

Biomass MIR Spectra

Even with large roughness penalties, it was in many cases not possible to resolve FLR models due to near-singularity of the regression matrices, regardless of which knot vector was used. Those models that did return solutions often had large RMSECV and RMSEP and were of inconsistent quality. In addition, Matlab also reported near-singularities in the covariance matrix $\mathbf{Z}'\mathbf{Z}$ in Equation (38) when using the three larger knot vectors. Interestingly, the FLR models regressing on D^0 and with non-zero intercept were very accurate in predicting the biomass polysaccharide fraction (i.e., cellulose plus xylan), regardless of the knot vector used. At the same time, however, the lignin-predictive model, built using identically-pretreated data, was very poor. The opposite was true when regressing on D^1 and forcing a zero-intercept. This suggests that some aspects of the spectra annihilated or otherwise modified by taking the first derivative – a baseline trend, for example – may correlate well linearly with the polysaccharide fraction but detrimentally (or nonlinearly) with lignin.

FPCR models, on the other hand, produced rather curious results. Predictions for the composition of the real plant matter samples were quite accurate in nearly all cases, however the minimum RMSEP for the surrogate validation set (RMSEP$_1$) and RMSECV for the training data were comparatively high, ranging, respectively, between 0.0732 – 0.108 and 0.118 – 0.125 regardless of knot vector, derivative order, or number of principal components. Because of the close similarities between these RMSEPs and RMSECVs across the board, they were unreliable as means for model discrimination.

Instead, the best models – presented in Table 6.2 for each knot vector – were identified as those which minimized RMSEP for the plant matter composition (RMSEP$_2$). Corroborating the results from the simulation study, the knot vector obtained via wavelet packets denoising yielded the overall best model (Model 4) for predicting the compositions of B10 and C10. As previously noted, the excellent denoising performance of wavelets can be attributed to the fact that they are well-suited to handling spectral features – being sharp, localized, and arbitrarily scalable. The optimum FPCR model additionally used three non-penalized principal components obtained from the variance decomposition of smoothed ($\lambda = 10^3$) second derivative spectra.

Table 6.2. Parameters and RMSEs for best FPCR models obtained using each knot vector.

Model No.	Knot vector	Derivative	λ	# of PCs	λ_{PC}	RMSECV	RMSEP
1	34 *eq.*	1	500	4	0	0.1237	0.0298
2	130 *eq.*	2	10^3	3	9.75	0.1186	0.0407
3	111 hr_B	2	10^3	3	0	0.1176	0.0569
4	133 hr_{WP}	2	10^3	3	0	0.1236	0.0210

eq: equally-spaced knots
hr_B: free-knots placed by heuristic algorithm with Butterworth filtering
hr_{WP}: free-knots placed by heuristic algorithm with wavelet packets filtering

Model 1 produced surprisingly accurate results, even though using only 34 knots resulted in what appeared to be over-smoothed spectra (see Figure 6.3*a*). This suggests that much of the fine detail may simply be unnecessary to produce reliable correlations, a conclusion substantiated by the work of Saeys et al. [2]. Additionally, models using both equally spaced knot vectors also outperformed those constructed from the two free-knot vectors for D^0 spectra. On the other hand, for first and second derivative spectra, plausible models constructed using the 133 hr_{WP} free-knot vector consistently outperformed those built from 130 equally spaced knots (data not shown).

Unlike in the FLR case, a single FPCR model can be used to reliably predict the mass fraction of either component with the same accuracy, since subtracting from the unity the value returned by one model gives virtually the same value predicted by the other. For Model 4, for example, the regression constant $\beta_{0,lig} = 0.27833$ and the regression coefficient function $\beta(t)_{lig}$ is presented in Figure 6.6. In the corresponding polysaccharides model, $\beta_{0,ps} = 0.72133$ and $\beta(t)_{ps}$ is the x-axis mirror image of $\beta(t)_{lig}$. It is interesting to note how the regression coefficient function itself resembles an IR spectrum, and analyzing its shape elucidates the importance of different wavelengths with respect to prediction. The region past about 1620 cm^{-1}, for instance, is quite close to zero, indicating that it contains little useful information for the problem under consideration. The region between about 1180 and 1280 cm^{-1} also does not seem to be particularly important given that the confidence intervals include zero, but this area is also devoid of knots – resulting in oversmoothing – and the apparently low importance could in reality be an artifact thereof. The various peaks and valleys with narrow confidence intervals and large deviations from the centerline can meanwhile be concluded to be strongly correlated with the response variable.

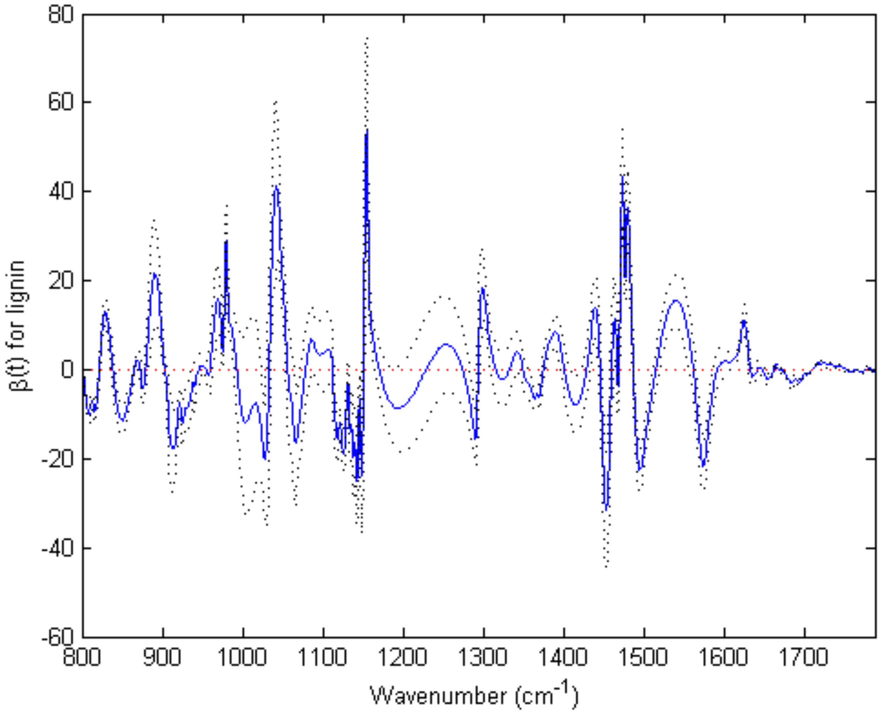


Figure 6.6. FPCR coefficient function for lignin from the most accurate FPCR model (Model 4). Dashed lines delimit the 95% confidence interval.

Predicted values for B10 and C10 from FPCR are compared alongside those from the best PLS model by Krasznai [44] and the wet-chemical analytical results in Figure 6.7. In addition to the standard autoscaling pretreatment, the PLS model was built using second derivative spectra obtained via Savitzky-Golay smoothing. Compared with Model 4, this PLS model had a roughly 60% larger RMSEP$_1$ but about 70 % lower RMSECV. By contrast, most of the other PLS models tested by Krasznai performed better on the surrogate mixtures – two, for example, had RMSEP$_1$ and RMSECV an order of magnitude less than Model 4 – but did not yield plausible predictions for the real plant matter. While the pretreatment methods heavily influenced PLS model quality, it is quite probable that their overall good performance with respect to the surrogate mixtures was due to the fact that PLS takes into account the correlation structure of both the response and explanatory variables.

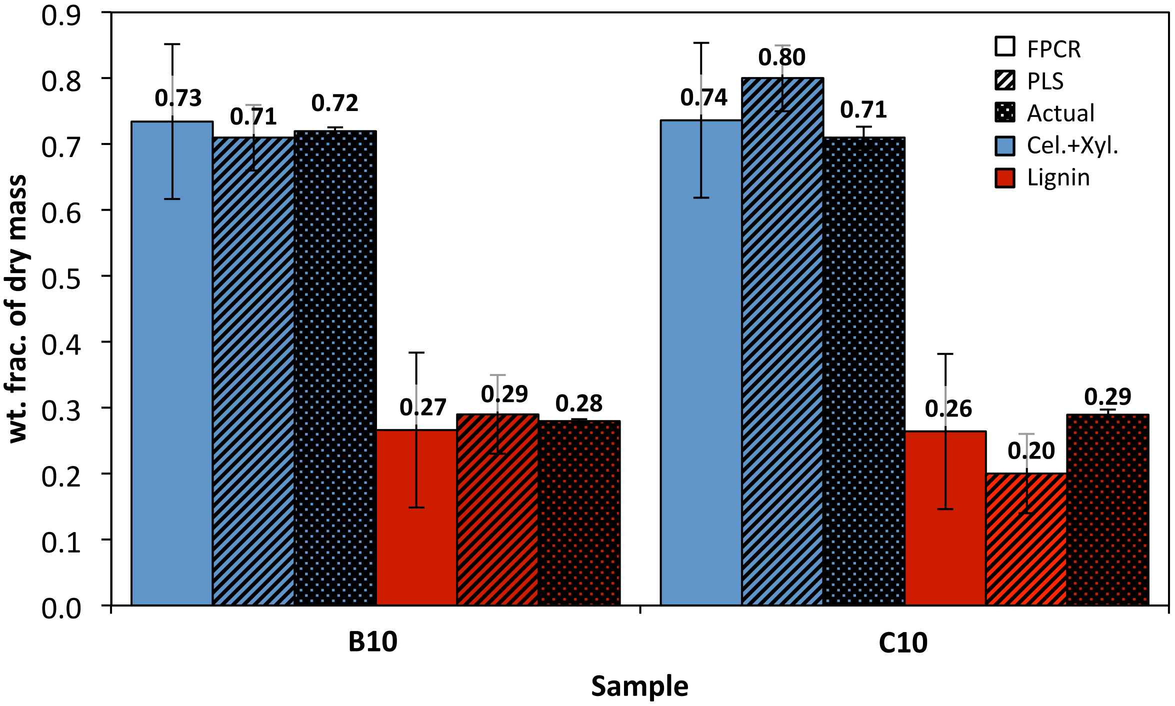


Figure 6.7. Comparison of actual polysaccharide compositions of real plant matter with best predictions from FPCR and PLS models. Error bars are RMSECV for the predictions, and RMSE of wet chemical analysis for the actual values.

Nevertheless, it is clear that FPCR has, using this particular dataset, greater predictive power for the samples B10 and C10 than traditional PLS. It is furthermore apparent that the FPCR approach is more robust than PLS, since a single model is capable of reasonable predictions in for the surrogate validation set and the real plant matter. Indeed, the functional approach was able to largely overcome the limitations of the dataset, specifically the considerable variability between the spectra of the real plant matter and the surrogates.

These differences are evident from the FPCA t-score plot of smoothed spectra, shown in Figure 6.8*a*. Here, the first three eigenfunctions respectively explain 84, 13, and 3 % of the variability in the data. The two plant matter samples (red squares) are clearly outlying from the surrogate set, which is arranged as expected in a triangular fashion corresponding to the ternary mixture experimental design. As experienced by Krasznai [44] and again here, it is symptomatic of the training data that RMSECV is, in this case, a poor metric for model discrimination. Though RMSECV and RMSEP_1 do not appreciably improve when moving from models built on D^0 to D^1 to D^2, the difference in t-score plots between the two cases shows why RMSEP_2 does. As can be seen in Figure 6.8*b*, FPCA on the second derivatives of the spectra shows a rather different correlation structure in which the t-scores for B10 and C10 are no longer such outliers. The percentage of variability explained is distributed more uniformly, as 55, 19, and 23% for components 1, 2, and 3, respectively. The reader may question why the percentages are not descending here – this is a consequence of rotation using the VARIMAX algorithm mentioned in Section 3.3.2.2. From a pretreatment standpoint, it is unsurprising that using first and second derivative spectra resulted in noticeable improvements in predictive quality, since these operations are quite effective in removing confounding baseline offset and linear baseline issues.

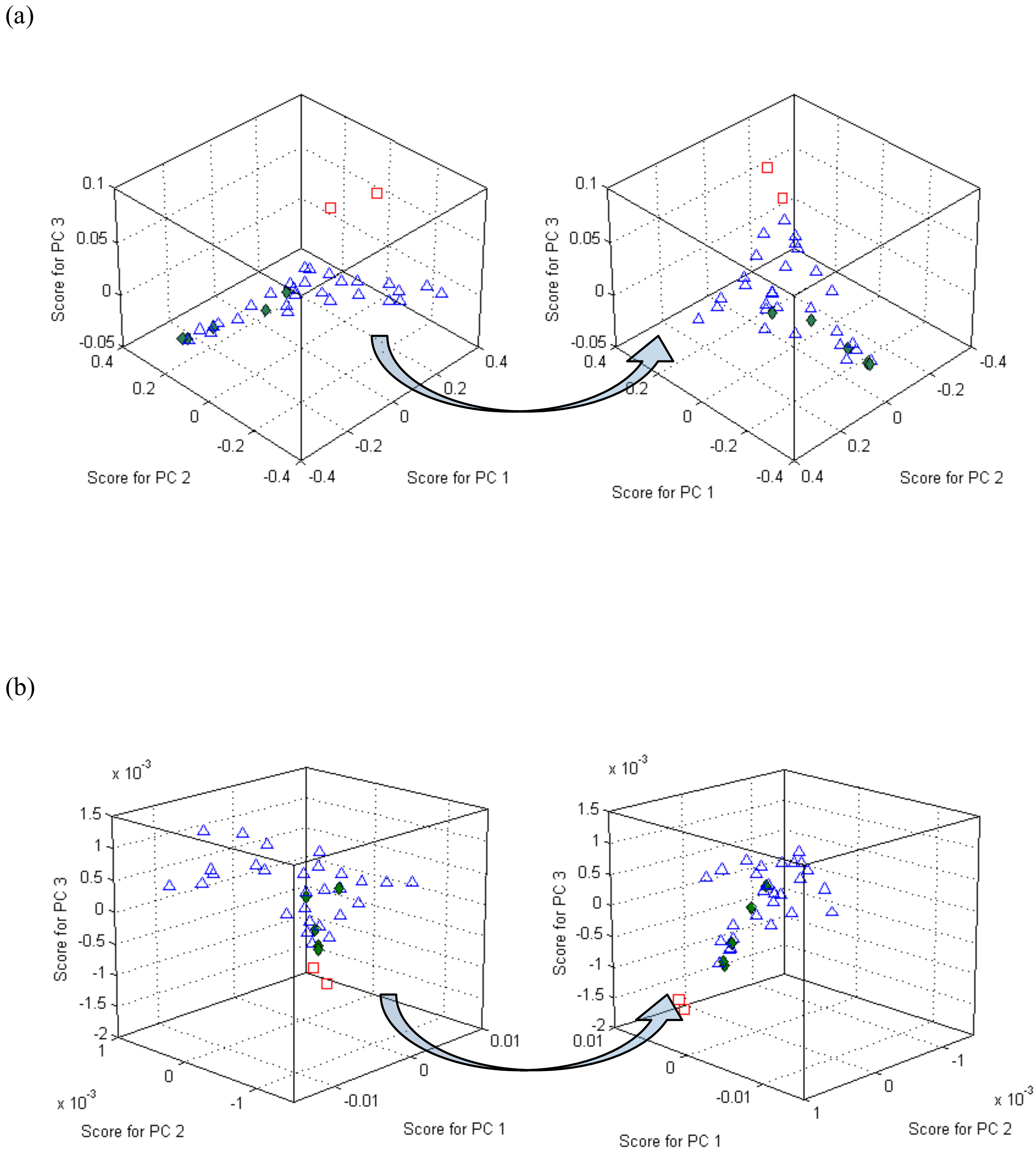


Figure 6.8. T-scores from FPCA of (a) D^0 and (b) D^2 of the MIR spectra smoothed using 133 free knots from heuristic algorithm with wavelet packets filtering. Blue triangles and green circles are t-scores for the surrogate mixtures, red squares for the two plant matter samples.

Diesel NIR Spectra

Unlike for the MIR data, differences in RMSECV were large enough between the models tested – summarized in Table 6.3 – to use the CV criterion as a metric for discrimination. The models obtained by Saeys et al. [2]– numbers 5 and 6 – were reproduced here to ensure fair comparison with FPCR. The respective RMSECV and RMSEP for each model are shown in Figure 6.9, and present a curious situation, in that the RMSECV-minimizing model (No. 8) in fact had the largest RMSEP. Comparing Models 7 and 8, it seems that this may be due to the frequentist free-knot vector being more representative of the training data than the validation set, whereas the uniformly-spaced knots treat both groups more equally (evidenced by the similar RMSECV and RMSEP for Model 7). On the other hand, the quality of Models 9 and 10 is almost identical, likely due to the fact that the dimensions of both knot vectors are sufficiently large to capture the variability between the training and validation sets in a more balanced manner. It was not possible to generate FLR models using the same larger knot vectors due to computational singularities once the knot vector dimension increased past 70.

Table 6.3. Summary of regression models correlating Diesel NIR spectra with cetane number.

Model No.	**Details**	
	Model Type	**Spectral Preprocessing**
5	PLS, 6 components	D^0 (i.e., no preprocessing)
6	FLR	D^0, 15 uniform knots, $\lambda = 0$
7	FPCR, 6 PCs	D^2, 15 uniform knots, $\lambda = 50$, $\lambda_{PC} = 0$
8	FPCR, 6 PCs	D^2, 15 free knots, $\lambda = 10$, $\lambda_{PC} = 0$
9	FPCR, 6 PCs	D^2, 124 uniform knots, $\lambda = 50$, $\lambda_{PC} = 0$
10	FPCR, 6 PCs	D^2, 124 free knots, $\lambda = 0$, $\lambda_{PC} = 0$

While a low-dimensional FLR proved in this case to be the optimum in terms of RMSEP, Models 8, 9, and 10 compare quite favorably, particularly from the standpoint of RMSECV. Indeed, if the objective is to retain both an accurate reproduction of the spectra in tandem with a powerful regression model, Model 10 is the best option.

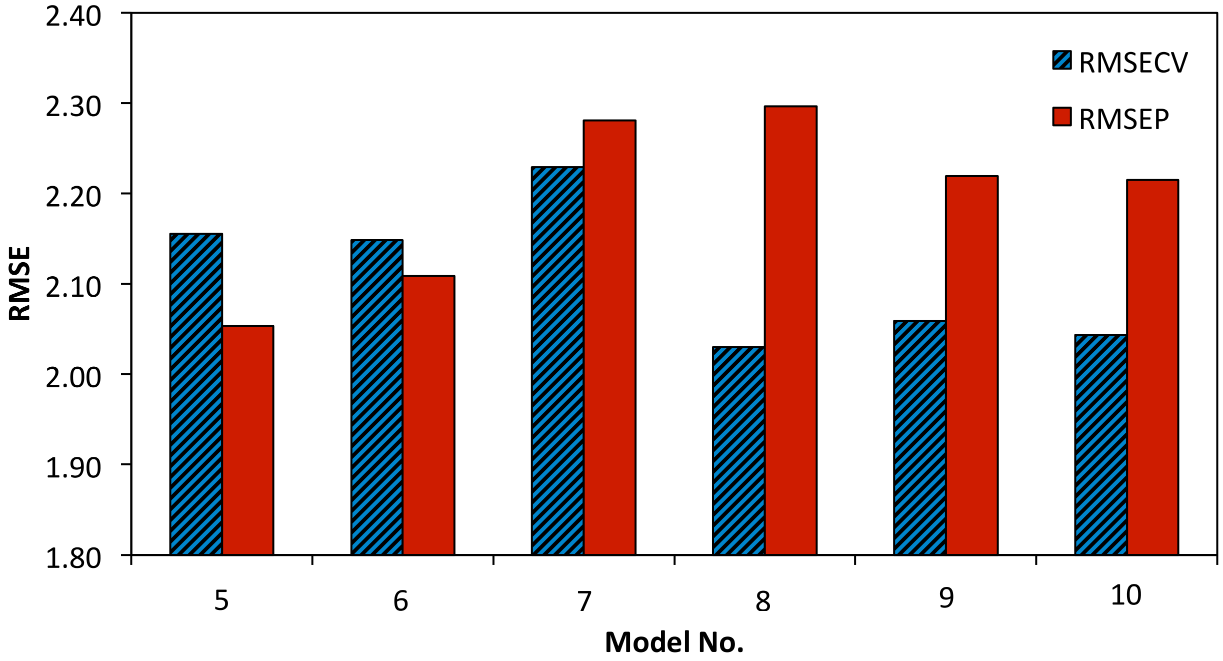


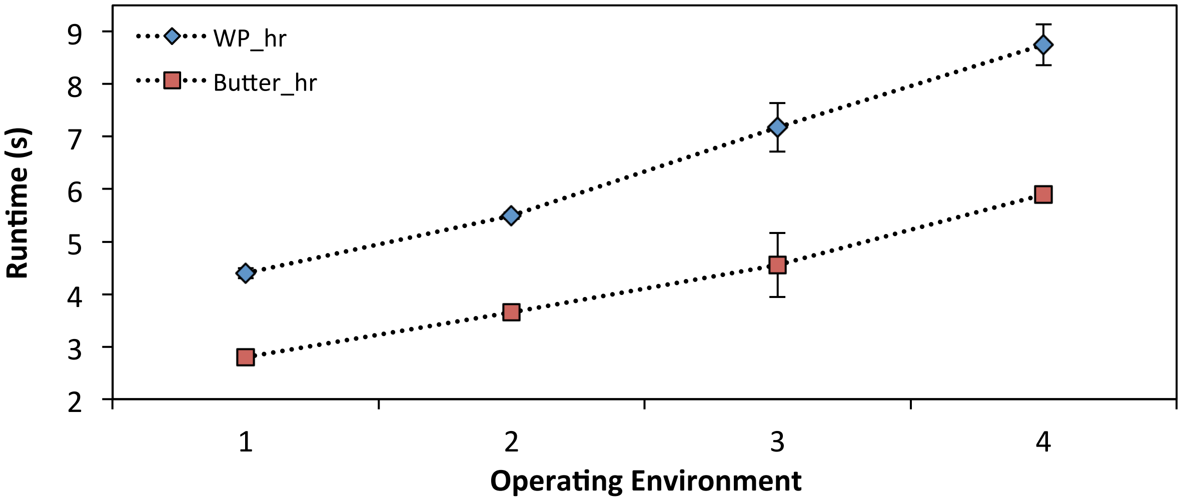
Figure 6.9. RMSE of cross-validation and prediction of FPCR (Models 7 to 10) compared with the PLS (Model 5) and FLR (Model 6) by Saeys et al. (2008).

Runtime Benchmarking

Code execution times comparing the Butterworth and WP implementations of the heuristic knot placement algorithm and FPCR models built from different knot vectors are shown in Figure 6.10 for the MIR dataset. Runtimes for the R&E algorithm are not included in the side-by-side comparison since the original code (written in Python) was unavailable. However, to give an idea of how fast the native R&E algorithm executes, finding the knot vector for 4000-point Raman spectra took approximately 4 ± 3 seconds (± the standard deviation) [16].

For purposes of expediency and interoperability with the FDA Package in the present work, the R&E algorithm was implemented in Matlab using a simple grid-search method to find the AD test minimizing point, instead of a simplex algorithm as in the original.

From visual inspection of Figure 6.10 it is evident that the Butterworth approach is much faster than WP in constructing the knot vector, reflecting the differing complexity of the two denoising techniques. It is interesting to note that the delta in execution times is less on the quad-core platforms – not because of the added cores (the code is single-threaded), but probably due to more efficient processor architectures. From the perspective of high-throughput applications, runtimes for knot placement scale about linearly, since the additional time to identify the frequentist knot vector is negligible. Speed could be increased considerably by implementing the algorithm in a more efficient language, such as C++, and by multithreading, which for very large datasets can in and of itself can be expected to cut times by almost a factor equal to the number of processor cores.



OE1: quad-core Intel Core i7 2600 (3.4 GHz), Windows 7 64-bit
OE2: quad-core Intel Core i7 860 (2.8 GHz), Windows 7 64-bit
OE3: dual-core Intel Core i5 2435M (2.4 GHz), Windows 7 64-bit in Parallels™ virtualization;
OE4: dual-core Intel Core i5 2435M (2.4 GHz), OS X 10.7.3

Figure 6.10. Runtime benchmarks for generating the frequentist free-knot vector for the MIR training data using the heuristic algorithm with WP and Butterworth filtering. Dotted lines connecting datapoints are meant as visual aids only.

Figure 6.11 provides additional insight into the complexities of building FPCA models. Only runtimes for Models 1 to 4 are shown for the sake of visual clarity; however, the trends are similar for the remaining models. Although the knot vectors in Models 2 and 4 are dimensionally almost the same (130 vs. 133, respectively), generating the former took appreciably longer to solve because of the additional computation required for the roughness penalty on the eigenfunctions. Model 1, meanwhile, took longer than Models 3 and 4 due to the additional principal component the former incorporates. Execution times for the latter two models are nearly indistinguishable given the similarity in knot vector dimension and model parameters. The larger variability in execution times on the Apple platforms OE3 and OE4 may be, respectively, due to inefficiencies in the way RAM is used in Parallels and in the way Matlab was programmed for OS X. Further gains in efficiency would be possible using a different language and, for very large datasets, multithreading the PCA [52].

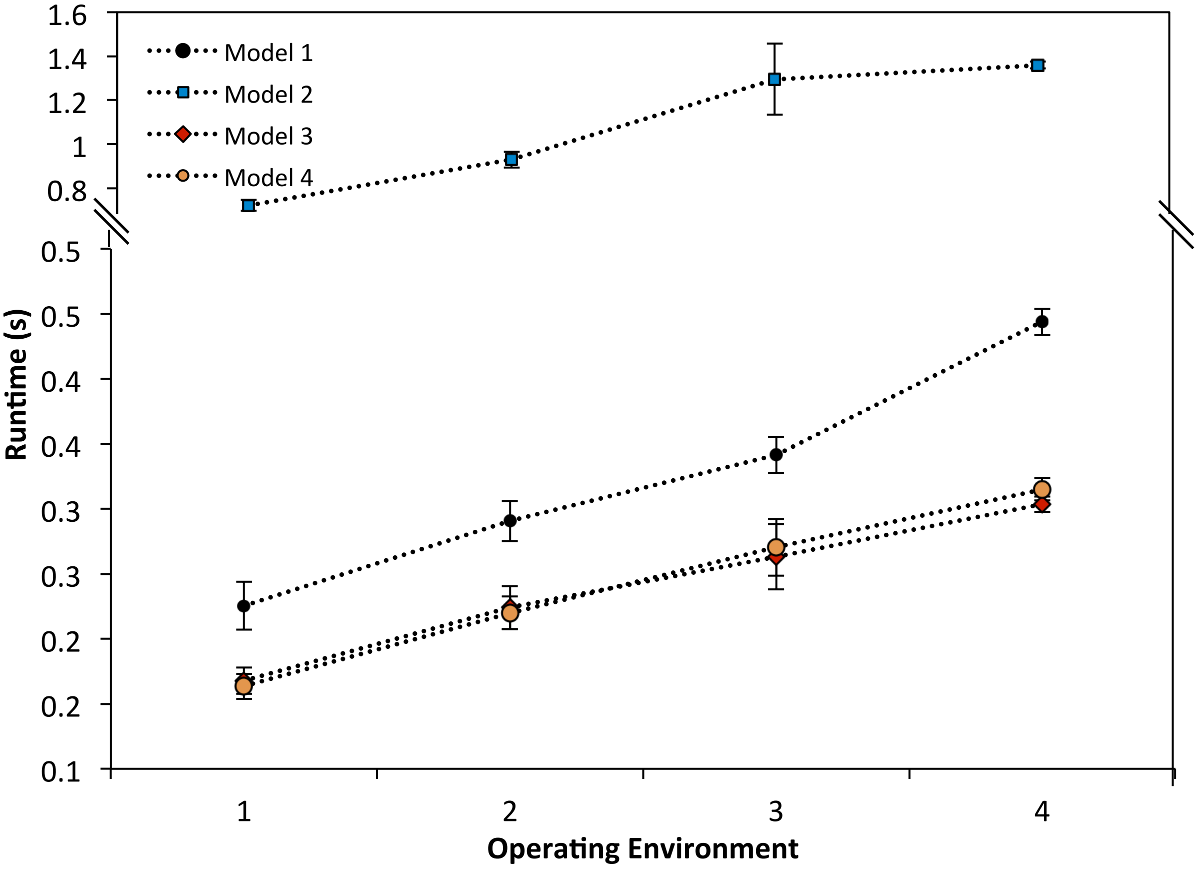


Figure 6.11. Runtime benchmarks for FPCR model generation for regression Models 1 to 4. Error bars represent plus or minus the standard deviation across 10 replicates. Dotted lines connecting datapoints are meant as visual aids only.

Chapter 7

Summary, Conclusions and Future work

7.1 Summary, Conclusions, and Contributions

This work investigated the utility and practicability of using functional data analysis as an alternative to traditional discrete mathematical and statistical techniques in the scope of one facet of chemometrics – that of preprocessing and correlating vibrational chemical spectra with specific chemical and physical characteristics. The first part involved evaluating the suitability of various existing data functionalization methods to smoothing these types of spectra. An algorithm was selected on the basis of robustness to various types of data, speed of execution, and autonomy from end-user input. This heuristic algorithm, originally developed by Li et al. [23] for fitting splines to automobile components, was augmented and subject to an optimization treatment to tailor it to the problem of spectral smoothing. The optimized algorithm was applied to real data in the construction of functional regression models, which were in turn compared against discrete multivariate and functional linear models developed by other authors. The following conclusions may be drawn from the work undertaken in this thesis:

1. Placing knots in a heuristic manner according to curvature in the data produces more accurate spectral reproductions and is more versatile to changing conditions (e.g., noise, sampling frequency, and type of spectra) than various approaches employing uniformly spaced knots.
2. It is not necessary for vibrational spectra to be reproduced precisely in order to generate reliable FPCR regression models. It is also not possible to draw a firm conclusion on when and whether or not free-knot vectors yield significant improvements in model

quality compared to uniformly spaced knots of similar dimension. However, it is recommended that knots be placed heuristically instead of uniformly when overall robustness is particularly important – that is, when both accurate spectral reproduction and good mode quality are desired – and because free-knots are generally able to more accurately capture localized features within the spectra.

3. The common approach of minimizing the root-mean square error of cross-validation (RMSECV) for parameter selection for functional regression models is not always optimal or even useful for model discrimination, particularly when there are deficiencies in the training data. Functional linear regression models should generally be avoided in these cases due to computational singularity issues and lack of robustness to changes in model parameters. Instead, it is advisable to use models incorporating further dimensional reduction, such as FPCR. Pretreatments of vibrational spectra that alleviate confounding baseline issues, such as taking first or second derivatives, also considerably enhance model robustness and predictive power.
4. Although it is the rate-limiting step in regression model-building, generating a common knot vector using the heuristic algorithm combined with a frequentist approach is fast, taking on the order of seconds to solve, but can be significantly accelerated using multithreaded code.
5. Spectral spline smoothing combined with functional principal components analysis is an effective method for data compression, since spectra that are sampled at hundreds or thousands of data points can be accurately reconstructed using a comparatively small number of basis coefficients and principal components.

In light of these conclusions, the specific contributions from this work are:

1. An automated methodology for fast, accurate, and parsimonious knot placement for splines that can accommodate a wide range of spectral conditions, including heteroscedasticity, changes in sampling rate, and varying signal to noise ratios.
2. Demonstrations of applicability of this methodology and of functional data analysis to real data for end-use in laboratory and industry settings.
3. A review of various spline-fitting techniques with respect to their applicability to spectral smoothing.

7.2 Future Work

A number of avenues for future investigation have arisen from the work described herein. These include:

- Identifying a more robust tool of selecting smoothing parameters for improving predictive power of FPCR models, particularly when highly representative training data is scarce or unavailable.
- Developing code for regression models employing unique knot vectors for each observation. This would permit a more comprehensive evaluation of whether or not free-knot vectors tailored to accurately reproduce individual functional observation result in better models than using uniformly spaced knots.
- Investigating functional partial least squares, which considers the variability in the response variable in tandem with that of the regressors. The heuristic versus equally spaced knot placement problem could be evaluated in this context as well.

- Use the heuristic knot placement algorithm to explore more complex data where both the regressors and responses are functional. An example segue from this work would be to analyze molecular weight distribution curves as functions of spectra for polymers.
- Investigating the usefulness of basis functions other than B-splines. For example, the recent study on comparing wavelets with B-splines [5] was limited to functional linear regression models, but this could be readily expanded to evaluating more complex multivariate models.

References

[1] P. T. Reiss and R. T. Ogden, “Functional Principal Component Regression and Functional Partial Least Squares,” *Journal of the American Statistical Association*, vol. 102, no. 479, pp. 984–996, Sep. 2007.

[2] W. Saeys, B. De Ketelaere, and P. Darius, “Potential applications of functional data analysis in chemometrics,” *J. Chemometrics*, vol. 22, no. 5, pp. 335–344, 2008.

[3] A. M. Aguilera, M. Escabias, C. Preda, and G. Saporta, “Using basis expansions for estimating functional PLS regression,” *Chemometrics and Intelligent Laboratory Systems*, vol. 104, no. 2, pp. 289–305, Dec. 2010.

[4] E. Bobelyn, A.-S. Serban, M. Nicu, J. Lammertyn, B. M. Nicolai, and W. Saeys, “Postharvest quality of apple predicted by NIR-spectroscopy: Study of the effect of biological variability on spectra and model performance,” *Postharvest Biology and Technology*, vol. 55, no. 3, pp. 133–143, Mar. 2010.

[5] Y. Zhao, R. T. Ogden, and P. T. Reiss, “Wavelet-Based LASSO in Functional Linear Regression,” *Journal of Computational and Graphical Statistics*, vol. 21, no. 3, pp. 600–617, Jul. 2012.

[6] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, 2nd ed. Springer Verlag, 2005.

[7] M. Febrero, P. Galeano, and W. González-Manteiga, “A functional analysis of NOx levels: location and scale estimation and outlier detection,” *Computational Statistics*, vol. 22, no. 3, pp. 411–427, Mar. 2007.

[8] M. Otto, *Chemometrics: Statistics and Computer Application in Analytical Chemistry*, vol. 4, no. 4. John Wiley & Sons, 2007, pp. 410–423.

[9] J. O. Ramsay, G. Hooker, and S. Graves, *Functional Data Analysis With R and MATLAB*. Springer Verlag, 2009.

[10] C. de Boor, *A Practical Guide to Splines*. Springer, 2001.

[11] D. Ruppert, “Selecting the Number of Knots for Penalized Splines,” *Journal of Computational and Graphical Statistics*, vol. 11, no. 4, pp. 735–757, Dec. 2002.

[12] M. Stone, “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 111–147, 1974.

[13] P. Craven and G. Wahba, “Smoothing noisy data with spline functions,” *Numerische Mathematik*, vol. 31, no. 4, pp. 377–403, 1978.

[14] S. Imoto and S. Konishi, “Selection of smoothing parameters in B -spline nonparametric regression models using information criteria,” *Ann. Inst. Statist. Math*, vol. 55, no. 4, pp. 671–687, Mar. 2003.

[15] G. Kauermann and J. D. Opsomer, “Data-driven selection of the spline dimension in penalized spline regression,” *Biometrika*, vol. 98, no. 1, pp. 225–230, Feb. 2011.

[16] C. J. Rowlands and S. R. Elliott, “Denoising of spectra with no user input: a spline-smoothing algorithm,” *J. Raman Spectrosc.*, vol. 42, no. 3, pp. 370–376, Mar. 2011.

[17] H. Akaike, “A new look at the statistical model identification,” *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[18] M. P. Wand, D. Ruppert, and R. J. Carroll, *Semiparametric Regression (Cambridge Series in Statistical and Probabilistic Mathematics)*. 2004, pp. 1–404.

[19] S. Spiriti, R. Eubank, P. W. Smith, and D. Young, “Knot selection for least-squares and penalized splines,” *Journal of Statistical Computation and Simulation*, pp. 1–17, Jan. 2012.

[20] G. Mamic and M. Bennamoun, “Automatic bayesian knot placement for spline fitting,” *International Conference on Image Processing. Proceedings. 2001*, vol. 1, pp. 169–172 vol. 1, 2001.

[21] V. Baladandayuthapani, B. K. Mallick, and R. J. Carroll, “Spatially Adaptive Bayesian Penalized Regression Splines (P-splines),” *Journal of Computational and Graphical Statistics*, vol. 14, no. 2, pp. 378–394, Jun. 2005.

[22] C. M. Crainiceanu, D. Ruppert, R. J. Carroll, A. Joshi, and B. Goodner, “Spatially Adaptive Bayesian Penalized Splines With Heteroscedastic Errors,” *Journal of Computational and Graphical Statistics*, vol. 16, no. 2, pp. 265–288, Jun. 2007.

[23] W. Li, S. Xu, G. Zhao, and L. P. Goh, “Adaptive knot placement in B-spline curve approximation,” *Computer-Aided Design*, vol. 37, no. 8, pp. 791–797, Jul. 2005.

[24] F. Yao and T. C. M. Lee, “On knot placement for penalized spline regression,” *Journal of the Korean Statistical Society*, vol. 37, no. 3, pp. 259–267, Sep. 2008.

[25] A. Gálvez and A. Iglesias, “Efficient particle swarm optimization approach for data fitting with free knot B-splines,” *Computer-Aided Design*, vol. 43, no. 12, pp. 1683–1692, Dec. 2011.

[26] T. C. M. Lee, “Automatic smoothing for discontinuous regression functions,” *Statistica Sinica*, vol. 12, no. 3, pp. 823–842, 2002.

[27] P. H. C. Eilers and B. D. Marx, “Flexible smoothing with B-splines and penalties,” *Statistical science*, pp. 89–102, 1996.

[28] D. Ruppert, M. P. Wand, and R. J. Carroll, “Semiparametric regression during 2003–2007,” *Electron. J. Statist.*, vol. 3, no. 0, pp. 1193–1256, 2009.

[29] J. Shao, “Linear model selection by cross-validation,” *Journal of the American Statistical Association*, pp. 486–494, 1993.

[30] N. Sugiura, “Further analysts of the data by akaike' s information criterion and the finite corrections,” *Communications in Statistics - Theory and Methods*, vol. 7, no. 1, pp. 13–26, Jan. 1978.

[31] T. C. Lee, “Smoothing parameter selection for smoothing splines: a simulation study,” *Computational Statistics and Data Analysis*, vol. 42, pp. 139–148, Jan. 2003.

[32] K. P. Burnham, “Multimodel Inference: Understanding AIC and BIC in Model Selection,” *Sociological Methods & Research*, vol. 33, no. 2, pp. 261–304, Nov. 2004.

[33] M. A. Stephens, “EDF statistics for goodness of fit and some comparisons,” *Journal of the American Statistical Association*, pp. 730–737, 1974.

[34] T. Krivobokova, C. M. Crainiceanu, and G. Kauermann, “Fast Adaptive Penalized Splines,” *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 1–20, Mar. 2008.

[35] P. R. Peres-Neto, D. A. Jackson, and K. M. Somers, “How many principal components? stopping rules for determining the number of non-trivial axes revisited,” *Computational Statistics and Data Analysis*, vol. 49, no. 4, pp. 974–997, Jun. 2005.

[36] R. E. Bellman, *Adaptive control processes: a guided tour*. Princeton, NJ: Princeton University Press, 1961.

[37] H. L. Shang, “A survey of functional principal component analysis,” Mar. 2011.

[38] D. A. Burns and E. W. Ciurczak, *Handbook of Near-Infrared Analysis*. M. Dekker, 2001.

[39] D. W. Mayo, F. A. Miller, and R. W. Hannah, *Course Notes on the Interpretation of Infrared and Raman Spectra*, vol. 36, no. 8. John Wiley & Sons, 2004, pp. 834–834.

[40] M. Bradley, *Curve Fitting in Raman and IR Spectroscopy: Basic Theory of Line Shapes and Applications*. Thermo Fisher Scientific, 2007, pp. 1–4.

[41] R. Bhargava and I. W. Levin, “Effective time averaging of multiplexed measurements: A critical analysis,” *Anal. Chem.*, vol. 74, no. 6, pp. 1429–1435, 2002.

[42] B. K. Alsberg, A. M. Woodward, M. K. Winson, J. Rowland, and D. B. Kell, “Wavelet Denoising of Infrared Spectra,” *Analyst*, vol. 122, no. 7, pp. 645–652, 1997.

[43] R. Brereton, *Chemometrics for Pattern Recognition*. John Wiley & Sons, 2009.

[44] D. Krasznai, *Multivariate Characterization of Lignocellulosic Biomass and Graft Modification of Nautral Polymers*, 2012.

[45] B. D. Marx and P. H. C. Eilers, “Generalized linear regression on sampled signals and curves: a P-spline approach,” *Technometrics*, pp. 1–13, 1999.

[46] R. Jarrow, D. Ruppert, and Y. Yu, “Estimating the Interest Rate Term Structure of Corporate Debt With a Semiparametric Penalized Spline Model,” *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 57–66, Mar. 2004.

[47] D. Ruppert and R. J. Carroll, “Theory & Methods: Spatially-adaptive Penalties for Spline Fitting,” *Australian & New Zealand Journal of Statistics*, vol. 42, no. 2, pp. 205–223, 2000.

[48] W. Li, S. Xu, G. Zhao, and L. P. Goh, “Adaptive knot placement in B-spline curve approximation,” *Computer-Aided Design*, vol. 37, no. 8, pp. 791–797, Jul. 2005.

[49] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. Wells Jr, “Noise reduction using an undecimated discrete wavelet transform,” *Signal Processing Letters, IEEE*, vol. 3, no. 1, pp. 10–12, 1996.

[50] W. Cao, X. Chen, X. Yang, and E. Wang, “Discrete wavelets transform for signal denoising in capillary electrophoresis with electrochemiluminescence detection,” *Electrophoresis*, vol. 24, no. 18, pp. 3124–3130, Sep. 2003.

[51] P. H. C. Eilers and B. D. Marx, “Splines, Knots, and Penalties,” Mar. 2010.

[52] M. Andrecut, “Parallel GPU Implementation of Iterative PCA Algorithms,” *Journal of Computational Biology*, vol. 16, no. 11, pp. 1593–1599, Nov. 2009.

Appendix A1

Wavelet Denoising Parameters

This appendix describes the various parameters available in the Matlab Wavelet Toolbox that were used in the factorial design optimization of the denoising step in the heuristic algorithm.

Discrete Wavelet Transform

DWT denoising was accomplished with the *wden* function, which requires five input parameters:

- One of four possible threshold selection rules:
  - `'rigsure'`: Rigorous Stein's Unbiased Estimate of Risk for adaptive thresholding. The threshold for each level is equal to the coefficient at that level with the smallest risk *r*, defined as

    $$r_k = \frac{(n_j - 2k) + b_k + a_k d_k}{n_j} \tag{66}$$

    where n_j is the number of wavelet coefficients at level j (j = 1, 2,…, J); **d** is a vector of length k whose elements are $[n_1 - 1, n_1 - 2, \ldots, 0, \ldots, n_J - 1, n_J - 2, \ldots, 0]$;

    $$b_k = \sum_{k=1}^{K} a_k \text{ ; and}$$

    $$a_k = (w_k)^2$$

    for wavelet coefficients w_k (k = 1, 2,…, K) sorted in ascending order by absolute value. A small SNR causes the SURE estimate to be very noisy.
  - `'sqtwolog'`: a universal threshold equal to $\sqrt{2\ln(n)}$.

    - `'heursure'`: heuristic SURE , which is a combination of the first two that sets the threshold to $\sqrt{2\ln(n)}$ when the SNR is very small.
    - `'minimaxi'`: a universal threshold chosen using the minimax principle.

  The `rigsure` and `minimax` selection rules are better-suited to cases in which parts of the true signal are present in the noise range.

- The type of thresholding (hard or soft): hard thresholding sets to zero the coefficients whose absolute values are less than the threshold. The soft variant first does the same as hard thresholding but then also attenuates the remaining, nonzero coefficients. Soft thresholding yields a smoother denoised curve, at the expense of higher RMSE.
- A method of computing the multiplicative factor for rescaling the threshold:
    - `'one'`: no rescaling is done, and the basic model for purely white noise is used; that is $\varepsilon \sim \mathrm{N}(0, \sigma_\varepsilon^2)$. This method was not included in the factorial design.
    - `'sln'`: rescaling is based on a single estimation of noise using the first-level coefficients when noise is believed to be white but un-scaled.
    - `'mln'`: applicable when the noise is assumed to be nonwhite. The estimation of level noise is done at each level.
- The number of decomposition levels *J* to decompose the signal into.
- The wavelet type: a specific basis must be selected from a large variety of wavelet types. Using orthogonal wavelets with compact support assures favorable computational characteristics.

Wavelet Packets

The Matlab function *wpdencmp* is used for denoising with wavelet packets and employs the Coifman-Wickerhauser "best basis" algorithm for identifying the basis with the lowest entropy. The function requires six inputs in addition to the signal:

- The type of thresholding (hard or soft), as in *wden*.
- The number of decomposition levels, as in *wden*.
- The wavelet type, as in *wden*.
- An entropy criterion, SURE being the default.
- A threshold: WP denoising uses a global threshold that can be user-defined or obtained using the function *ddencmp*, which calculates the SURE threshold ($\sqrt{2\ln[n\ln(n)/\ln(2)]}$) for a given input signal.
- Whether or not to threshold the approximation coefficients.

Stationary Wavelet Transform

The stationary wavelet transform was implemented using the function *swt* and the same parameter options as the DWT.

Appendix A2

Wavelet Denoising Parameters

The Pareto-optimal filter-specific parameters from the factorial design space for Butterworth, Stationary Wavelet Transform and Wavelet Packets and are presented in the tables below.

Table A2-1. Pareto-optimal parameters for Butterworth filtering

Sampling Resolution (cm^{-1})	SNR	Filter order	Cutoff freq.	Denoising Sequence
0.5	30	7	0.02	1
	40	6	0.02	1
	50	6	0.02	1
	60	7	0.02	1
	70	7	0.02	1
	80	7	0.02	1
	90	7	0.02	1
	100	7	0.02	1
0.75	30	4	0.01	1
	40	7	0.02	1
	50	6	0.02	1
	60	6	0.02	1
	70	6	0.02	1
	80	6	0.02	1
	90	6	0.02	1
	100	6	0.02	1
1	30	7	0.02	1
	40	7	0.02	1
	50	7	0.02	1
	60	6	0.02	1
	70	7	0.02	1
	80	7	0.02	1
	90	7	0.02	1
	100	7	0.02	1
1.25	30	7	0.01	1
	40	7	0.01	1
	50	7	0.02	1
	60	5	0.02	1
	70	5	0.02	1
	80	5	0.02	1
	90	7	0.02	1
	100	5	0.02	1

Table A2-2. Pareto-optimal parameters for Stationary Wavelet Transform filtering

Sampling Resolution (cm^{-1})	SNR	Soft or hard thresh.	Threshold rescaling	Threshold rule	Wavelet Family	Wavelet Order	Decomp. level	Denoising Sequence
0.5	30	s	sln	sqtwolog	Sym	4	8	1
	40	s	mln	sqtwolog	Coif	4	6	1
	50	s	mln	sqtwolog	Sym	6	6	1
	60	s	mln	sqtwolog	Coif	5	6	1
	70	s	mln	sqtwolog	Db	7	6	1
	80	s	mln	sqtwolog	Db	7	6	1
	90	s	mln	sqtwolog	Db	7	6	1
	100	s	mln	sqtwolog	Db	7	6	1
0.75	30	s	mln	sqtwolog	Coif	3	6	1
	40	s	mln	sqtwolog	Coif	5	6	1
	50	s	mln	sqtwolog	Sym	5	6	1
	60	h	mln	sqtwolog	Sym	7	6	1
	70	h	mln	sqtwolog	Sym	7	6	1
	80	s	mln	sqtwolog	Db	8	6	1
	90	h	mln	sqtwolog	Coif	5	6	1
	100	h	mln	sqtwolog	Coif	4	6	1
1	30	s	mln	sqtwolog	Sym	5	6	1
	40	s	mln	sqtwolog	Coif	5	5	1
	50	s	mln	sqtwolog	Sym	4	6	1
	60	s	sln	sqtwolog	Sym	2	5	2
	70	h	mln	sqtwolog	Sym	3	6	1
	80	h	mln	sqtwolog	Coif	4	6	1
	90	h	mln	sqtwolog	Coif	3	6	1
	100	h	mln	minimaxi	Coif	4	6	1
1.25	30	s	mln	sqtwolog	Coif	3	6	1
	40	s	mln	sqtwolog	Sym	5	6	1
	50	s	mln	sqtwolog	Sym	3	6	1
	60	s	sln	sqtwolog	Db	1	5	2
	70	h	sln	sqtwolog	Sym	7	7	1
	80	h	mln	sqtwolog	Coif	4	6	1
	90	s	mln	minimaxi	Coif	5	5	1
	100	s	mln	minimaxi	Coif	5	5	1

Table A2-3. Pareto-optimal parameters for Wavelet Packets filtering

Sampling Resolution (cm^{-1})	SNR	Soft or hard thresh.	Keep approximation coeffs?	Wavelet Family	Wavelet Order	Decomp. level	Denoising Sequence
0.5	30	s	yes	Sym	6	6	1
	40	s	yes	Db	6	6	1
	50	s	yes	Db	7	6	1
	60	s	yes	Db	9	6	1
	70	s	yes	Db	9	6	1
	80	s	yes	Db	7	6	1
	90	s	yes	Db	9	6	1
	100	s	yes	Db	9	6	1
0.75	30	s	yes	Db	10	6	1
	40	s	yes	Db	10	6	1
	50	s	yes	Db	9	6	1
	60	s	yes	Db	10	6	1
	70	s	yes	Db	10	6	1
	80	s	yes	Db	10	6	1
	90	s	yes	Db	10	6	1
	100	s	yes	Db	10	6	1
1	30	s	yes	Db	6	6	1
	40	s	yes	Db	8	6	1
	50	s	yes	Db	8	6	1
	60	s	yes	Db	8	6	1
	70	s	yes	Db	8	6	1
	80	s	yes	Db	8	6	1
	90	s	yes	Db	8	6	1
	100	s	yes	Db	8	6	1
1.25	30	s	yes	Sym	7	6	1
	40	s	yes	Db	7	6	1
	50	s	yes	Db	7	6	1
	60	s	yes	Db	7	6	1
	70	s	yes	Db	7	6	1
	80	s	yes	Db	7	6	1
	90	s	yes	Db	7	6	1
	100	s	yes	Db	7	6	1

Appendix B

Algorithm for Heuristic Knot Placement

The heuristic knot placement algorithm developed in this thesis and based upon that by Li et. al (2005) is described below. In the Li algorithm, step 3 was accomplished using an unspecified lowpass finite impulse response filter, and it is the denoising step that was optimized in the present work. Steps 1, 9, and 10 were added as part of tailoring the algorithm to the spectral smoothing problem.

1. Normalize the discretely sampled data.
2. Compute the discrete curvature c at each data point $\mathbf{p}_i = \mathrm{p}(x_i, t_i)$ ($i = 1,\ldots, n$), where

$$c_i = \frac{2\Delta\mathbf{p}_{i-1}\mathbf{p}_i\mathbf{p}_{i+1}}{\|\mathbf{L}_i\|\|\mathbf{L}_{i+1}\|\|\mathbf{Q}_i\|} = sign(2\Delta\mathbf{p}_{i-1}\mathbf{p}_i\mathbf{p}_{i+1})\frac{2\sin(\alpha_i)}{\|\mathbf{Q}_i\|} \qquad (67)$$

and

$\mathbf{L}_i = \mathbf{p}_i - \mathbf{p}_{i-1}$

$\mathbf{Q}_i = \mathbf{p}_{i+1} - \mathbf{p}_{i-1}$

$2\Delta\mathbf{p}_{i-1}\mathbf{p}_i\mathbf{p}_{i+1} = |\mathbf{L}_i, \mathbf{L}_{i+1}|$

$\cos(\alpha_i) = \frac{\mathbf{L}_i \cdot \mathbf{L}_{i+1}}{\|\mathbf{L}_i\|\|\mathbf{L}_{i+1}\|}, \sin(\alpha_i) = \cos\left(\frac{\pi}{2} - \alpha_i\right),$ and $\alpha_i = \arccos\left(\frac{\mathbf{L}_i \cdot \mathbf{L}_{i+1}}{\|\mathbf{L}_i\|\|\mathbf{L}_{i+1}\|}\right)$

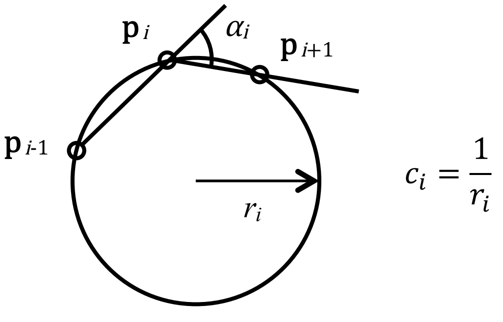


Figure B.1. Schematic of process for calculating the discreet curvature.

3. Denoise *c* using a suitable lowpass filter.
4. Assign knots to argument points nearest to points of
5. inflection, known as "feature" points, which correspond to zero-crossings on the curvature plot of *c* vs. *t*. Mathematically, an inflection point exists in $[s_i, s_{i+1}]$ if $c_i c_{i+1} \leq 0$. The knot is located at the argument value corresponding to c_i if $|c_i| < |c_{i+1}|$ and otherwise at c_{i+1}.
6. Integrate the denoised *c* using the Newton-Cotes formula

$$v = \int_{s_0}^{s_r} |c|\, dt = \frac{1}{2}\sum_{i=1}^{r} (|c_i| + |c_{i-1}|)(s_i - s_{i-1}). \tag{68}$$

7. Assign new, "non-feature" knots to argument points nearest to the bisection value of *v* in each interval delimited by the feature knots.
8. If the angle α_i at a non-feature knot *i* does not satisfy the rule $\alpha_i \leq \pi/6$, bisect segments on either side of the knot. Repeat until all non-feature knots satisfy the rule, if possible.
9. Bisect the two adjacent intervals around feature knots *j* that do not satisfy the heuristic rule. Only keep the new knot in the interval that satisfies the rule and repeat until all feature knots satisfy $\alpha_j \leq \pi/6$, if possible.
10. If there are gaps in the knot vector greater than 40 data points apart, populate these gaps according to the rule $K = \min(n/4, 35)$.
11. Add a roughness penalty obtained by minimizing GCV(λ).