IntechOpen

Topics in Splines and Applications

Edited by Young Kinh-Nhue Truong and Muhammad Sarfraz

TOPICS IN SPLINES AND APPLICATIONS

Edited by **Young Kinh-Nhue Truong**
and **Muhammad Sarfraz**

Topics in Splines and Applications
http://dx.doi.org/10.5772/intechopen.68737
Edited by Young Kinh-Nhue Truong and Muhammad Sarfraz

Contributors

Horacio Florez, Belsay Borges, Prabhakar Karthikeyan Shanmugam, Akhil Prasad, Atul Manmohan, Kothari D.P, Carlos Alvarado-Serrano, María De Lourdes Corzo-Cuesta, Taizhi Liu, Na Li, Jingyi Zhang, Ye Wang, Honghe Jin, Xiaoxiao Sun, Wenxuan Zhong, Ping Ma, Huong Pham, Hoa Pham

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in London, United Kingdom, 2018 by IntechOpen
eBook (PDF) Published by IntechOpen, 2019
IntechOpen is the global imprint of INTECHOPEN LIMITED, registered in England and Wales, registration number: 11086078, The Shard, 25th floor, 32 London Bridge Street
London, SE19SG – United Kingdom
Printed in Croatia

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Additional hard and PDF copies can be obtained from orders@intechopen.com

Topics in Splines and Applications
Edited by Young Kinh-Nhue Truong and Muhammad Sarfraz

p. cm.

Print ISBN 978-1-78923-250-9

Online ISBN 978-1-78923-251-6

eBook (PDF) ISBN 978-1-83881-350-5

Meet the editors

Young Kinh-Nhue Truong, PhD, is a professor of Biostatistics at the University of North Carolina at Chapel Hill. His research expertise includes statistical learning, functional modeling, time series, spatiotemporal data analysis, and event history analysis. He has in the areas of statistical time series/longitudinal modeling using splines, window or kernel-based smoothing methods, and wavelets. His current research focuses mainly on spatiotemporal data analysis with the aim to spatially localize dynamic processes in the functional magnetic resonance imaging (fMRI) or electroencephalography (EEG) human brain data. He is also interested in developing statistical inference for group comparison based on human brain imaging data and methods for analyzing neuro-spike train data. His teaching and research experience in linear mixed modeling has provided new insights and approaches to many statistical analyses for handling between- and within-subject variability.

Muhammad Sarfraz is a professor in Kuwait University, Kuwait. He received his PhD degree from Brunel University, UK, in 1990. His research interests include computer graphics, pattern recognition, computer vision, image processing, and soft computing. He is currently working on various projects related to academia and industry. Prof. Sarfraz has been a keynote/invited speaker at various platforms around the globe. He has advised more than 60 students for their MSc and PhD degree theses. He has produced more than 330 publications in the form of various books, book chapters, journals, and conference papers.

Prof. Sarfraz is member of various professional societies including IEEE and ACM. He is a chair and member of various international conferences. He is also an editor/guest editor of various international conference proceedings, books, and journals. He is a reviewer for many international journals, conferences, meetings, and workshops around the world. He has received various awards in education, research, and administrative services.

Contents

Preface

Splines are piecewise polynomials that are joined smoothly, and they provide a significant tool for the design of computationally economical curves and surfaces. This, in return, plays a critical role in the construction of various objects like automobiles, ship hulls, airplane fuselages and wings, propeller blades, shoe insoles, bottles, etc. It also contributes in the description of geological, physical, statistical, and even medical phenomena. Moreover, spline methods have also proven to be indispensable in a variety of modern industries, including *computer vision, robotics, signal and image processing, visualization, textile, graphic designs, art designs, painting, animations in the film industry, font designs in publishing,* and even *media.*

The major goals of this book are to stimulate views and provide a source where researchers and practitioners can find the latest developments in the field of splines. Due to speedy scientific developments, there is a great deal of thirst among the scientific community worldwide to be equipped with state-of-the-art theory and practice to get their problems solved in diverse areas of various disciplines. Although a good amount of work has been done by researchers, a tremendous interest is increasing every day due to complicated problems being faced in the academia and industry, especially in areas where data are essential for research.

This book aims to provide a valuable source on splines and their applications. It focuses on some interdisciplinary methods using splines. It targets to provide the user community with a variety of spline techniques and their applications necessary for various real-life problems. It also aims to collect and disseminate information in various disciplines including *computer-aided geometric design, computer graphics, data visualization, data fitting, digital signal processing, power systems, clinical and epidemiologic studies, disease detection, regression curves, social media, and biological studies.*

This book has seven chapters consisting of several new advances in the area. These contributions provide a modern, up-to-date literature review, theory, techniques, methodology, software developments, and applications. The book is useful for researchers, scientists, practitioners, and many others who seek state-of-the-art techniques and applications using splines. It is also useful for undergraduate senior students as well as graduate students in the areas of computer science, engineering, health science, statistics, and mathematics. An important feature of the book is that each chapter also provides useful information on the software developments and their extensions.

The book begins with an exposition of splines from a numerical analysis point of view. The first chapter entitled “Scalar and Parametric Spline Curves and Surfaces” was prepared by Horacio Florez and Belsay Borges. In the absence of noise, their work tackles passing a curve/surface through a given set of data points. They first focus on the case in which the curve to be constructed can be described as a scalar spline for the manipulation of dataset as scalar data. They describe herein cubic and tension splines, which are powerful interpolants suitable to tackle large datasets. They then introduce a parametric case for a vector-valued curve, and, hence, it is able to represent arbitrary topologies. They explain how to construct piecewise continuous cubic Bèzier curves called "B-splines." They address both interpolation and approximation problems, the latter denoted as "inverse design." They extend the treatment further to tensor product surfaces that are referred to as piecewise bicubic B-splines. Applications encompass translational and interpolation surfaces. They briefly introduce nonuniform rational B-spline (NURBS) curves and surfaces. They present applications such as approximating conic sections. They, finally, introduce Duchon splines that are radial basis functions to interpolate scattered datasets in two or three dimensions.

Splines play an important role in modern signal processing. In addition to providing new insights into the sampling theory, they are also perfect candidates for constructing basis functions such as wavelets for multiresolution processing. A good illustration is given in Chapter 2, "An Algorithm Based on the Continuous Wavelet Transform with Splines for the Automatic Measurement of QT Dispersion: Validation and Application in Chronic Kidney Disease" by María de Lourdes Corzo-Cuesta and Carlos Alvarado-Serrano. Chronic kidney disease (CKD) is considered a risk factor for development of cardiovascular disease. QT interval is an electrocardiographic parameter that quantifies the duration of ventricular repolarization. An increase of its spatial variability measured from selected leads of a standard electrocardiogram (ECG), named QT dispersion (QTd), is considered a risk factor for malign ventricular arrhythmias and sudden death in the CKD. An algorithm is presented for automatic measurement of QTd in the ECG leads DI, aVF, and V2 using the continuous wavelet transform with splines. Validation of QRS complex detection has been done on records from MIT-BIH database, and the accuracy has been of 99.5%. Validation of detection of QRS wave onset and T wave end has been done on records from CSE and QT databases, and the measurements are within the tolerance limits for deviations with respect to the manual measurements defined by the experts. In this chapter, the algorithm has been applied for two studies. In the first one, QTd is evaluated in normal subjects and patients with CKD. In the second study, QTd is analyzed in patients with CKD before, during, and after the hemodialysis treatment. In both studies, the algorithm has a good performance for the QTd analysis. This is an example of biomedical signal analysis where data are taken over time.

The next few chapters discuss how to address noise issues when data are obtained as a random sample from certain populations. Chapter 3 discusses multivariate adaptive regression splines (MARS), which is a nonparametric regression method based on a least squares formulation. It is an adaptive procedure that does not require any prespecified functional forms of the components in the regression model. The model can be additive with only the main effects or interactive involving a set of selected variables. With that said, the model structure of MARS is constructed dynamically and adaptively according to the information derived from the data. Because of its ability to capture essential nonlinearities and interactions, MARS is considered as a great fit for high-dimensional problems by low-dimensional components such as the main or interactive effects.

The chapter entitled "Multivariate Adaptive Regression Splines in Standard Cell Characterization for Nanometer Technology in Semiconductor" by Taizhi Liu illustrates an application of MARS in semiconductor field, more specifically, in standard cell characterization. The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that accurately and efficiently captures cell behaviors. In this chapter, the MARS method is employed to characterize the gate delay as a function of many parameters including process-voltage-temperature parameters. Due to its ability of capturing essential nonlinearities and interactions, MARS method helps to achieve significant accuracy improvement.

Complex and massive datasets can be easily accessed using the newly developed data acquisition technology. In spite of the fact that the smoothing spline ANOVA models have proven to be useful in a variety of fields, these datasets impose the challenges on the applications of the models. To address the need for statistical inference in addition to exploring the regression model structures, this is effectively done by casting the spline fitting procedure in the *reproducing kernel Hilbert space* (RKHS). Jingyi Zhang et al., in Chapter 4 "Smoothing Spline ANOVA Models and Their Applications in Complex and Massive Datasets," present a selected review of the smoothing spline ANOVA models and highlight some challenges and opportunities in massive datasets. They review a heuristic method for the inference and two approaches to significantly reduce the computational costs of fitting the model. The real case studies in social media and biological studies are used to illustrate the performance of the reviewed methods.

In the next chapter, NA Li describes a "Model Testing Based on Regression Spline" by offering an alternative approach to statistical tests of hypotheses based on regression spline. Specifically, the methods are developed for testing functional relationships in nonparametric, partial linear, and varying-coefficient models, respectively. These models are more flexible than the linear regression models. They are also more efficient than the other nonparametric kernel-based methods especially in high-dimensional problems. However, one important problem is that if it is useful to use regression spline and fiducial method in order to obtain p-values for testing the linearity and constancy of the nonparametric functions. In the application of spline-based method, the determination of knots is difficult but plays an important role in inferring regression curve. In order to infer the nonparametric regression at different smoothing levels (scales) and locations, multi-scale smoothing methods based on regression spline are developed to test the structures of the regression curve and compare multiple regression curves. It could sidestep the determination of knots, meanwhile giving a more reliable result in using the spline-based method.

Statistics offers several methods for handling noise in addition to the least squares type of approaches. One of these is to formulate the likelihood structure of the data. Motivated by the use of penalized spline, Chapter 6, "Penalized Spline Joint Models for Longitudinal and Time-to-Event Data" by Huong Thi Thu Pham and Hoa Pham, addresses the issue of joint models for longitudinal and time-to-event data. The joint models for longitudinal data and time-to-event data have recently received numerous attention in clinical and epidemiologic studies. This chapter attracts readers' interest in modeling the relationship between time-to-event outcomes and internal time-dependent covariates. In practice, the longitudinal responses often show nonlinear and fluctuated curves. Therefore, the main aim of this chapter is to use penalized splines with a truncated polynomial basis to parameterize the nonlinear longitudinal process. The linear mixed-effects model is applied to subject-specific curves to control the smoothing. The association between the dropout process and longitudinal outcomes is modeled through a proportional hazards model. Two types of baseline risk functions are considered, namely, a Gompertz distribution and a piecewise constant model. The resulting models are referred to as penalized spline joint models, an extension of the standard linear joint models.

The last chapter is “Application of Cubic Spline Interpolation Technique in Power Systems: A Review” by Akhil Prasad et al. In this chapter, a comprehensive review is made on the application of cubic spline interpolation techniques in the field of power systems. Domains like available transfer capability (ATC), electric arc furnace modeling, static VAR compensation, voltage stability margin, and market power determination in deregulated electricity market are taken as samples to illustrate the significance of cubic spline interpolation.

The editors are grateful to the contributors for their valuable efforts toward the completion of this book. A lot of credit is also due to the experts who reviewed the chapters and provided helpful feedback. The editors are happy to acknowledge the support of Ms. Martina Usljebrka, Author Service Manager, IntechOpen, toward the compilation of this book.

Prof. Young Kinh-Nhue Truong
The University of North Carolina at Chapel Hill, USA

Prof. Muhammad Sarfraz
Kuwait University, Kuwait

Chapter 1

Scalar and Parametric Spline Curves and Surfaces

Horacio Florez and Belsay Borges

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74929

Abstract

A common engineering task consists of interpolating a set of discrete points that arise from measurements and experiments. Another traditional requirement implies creating a curve that mimics a given array of points, namely, a polyline. Any of these problems require building an analytical representation of the given discrete set of points. If the geometrical shape represented by the input polyline is complicated, then we may expect that a global interpolant or polynomial will be of a high degree, to honor all imposed constraints, which makes its use prohibited. Indeed, a global interpolant often experiences inflection points and sudden changes in curvature. To avoid these drawbacks, we often seek solving the interpolation/approximation problem using piecewise polynomial functions called "splines."

Keywords: cubic splines, tension splines, Bèzier curves, B-splines, NURBS

1. Introduction

In this chapter, we tackle passing a curve/surface through a given set of data points. We first focus in the case in which the curve to be constructed can be described as $S(x) = (x, f(x))$. We refer this data set as scalar data. We describe herein cubic and tension splines, which are powerful interpolants suitable to tackle large data sets. We then introduce a parametric case for a vector-valued curve $\underline{S}(\xi) = (x(\xi), y(\xi))^T$ and hence, it is able to represent arbitrary topologies. We explain how to construct piecewise continuous cubic Bèzier curves called "B-splines." We cover the interpolation and approximation problems with B-splines, this latter denoted as well as "inverse" design. We extend our treatment to tensor product surfaces that are referred as piecewise bicubic B-splines. Applications encompass translational and interpolation surfaces. We briefly introduce nonuniform rational B-spline curves and surfaces (*NURBS*).

We present applications such as approximating conic sections. We finalize the chapter introducing Duchon splines that are radial basis functions to interpolate scattered data sets in two or three dimensions.

2. Scalar splines

We cover herein the scalar case in which a spline function $S(x) = (x, f(x))$ fits a given set of sorted point pairs. We introduce cubic splines and their specialized version that offers a "tension" parameter that allows attracting the interpolant toward the polyline that connects the input points, i.e., linear spline. We refer to this latter as tension splines. The last section presents a couple of numerical examples.

2.1. Cubic splines

A spline is a piecewise continuous function consisting of several polynomials, each specified in a subinterval, bound themselves by certain continuity conditions. Let $x_0, \ldots, x_n$ be $(n+1)$ sorted points such that $x_0 < x_1 < x_2 < \ldots < x_n$ whose corresponding values are denoted by $y_0, \ldots, y_n$. A spline of k degree with knots $x_0, \ldots, x_n$ is a function $S : \mathbb{R} \to \mathbb{R}$ such that:

1. S_i is a polynomial of degree $\leq k$ that is continuous up to kth derivative over $[x_i, x_{i+1}]$.

2. Two adjacent splines need to have C^0 continuity at the junction points:

$$S(x) = \begin{cases} S_0(x); x \in [x_0, x_i) \\ S_i(x); x \in [x_i, x_{i+1}) \\ \vdots \\ S_{n-1}(x); x \in [x_{n-1}, x_n) \end{cases} \tag{1}$$

We thus enforce $C^m, m = 0, \ldots, (k-1)$ continuity conditions at the $(n-1)$ junction points which yields to $(4n-2)$ equations to determine $4n$ unknown spline coefficients. We omit details herein but refer the reader to [1, 2]. We end up with a tridiagonal system for the unknown curvature values κ_i, at the junction points:

$$h_{i-1} \cdot \kappa_{i-1} + 2 \cdot (h_i + h_{i-1}) \cdot \kappa_i + h_i \cdot \kappa_{i+1} = \frac{6}{h_i}(y_{i+1} - y_i) - \frac{6}{h_{i-1}}(y_i - y_{i-1}), \tag{2}$$

where $i = 1, .., n-1$ and $h_i = x_{i+1} - x_i$. The last equation provides a system of $(n-1)$ conditions for $\kappa_0, \ldots, \kappa_n$. Since both κ_0 and κ_n are arbitrary, a logical choice is choosing $\kappa_0 = \kappa_n \equiv 0$, which we refer as "natural spline." For the latter, we can write in matrix form:

$$\begin{bmatrix} u_1 & h_1 & & & & & \\ h_1 & u_2 & h_2 & & & & \\ & h_2 & u_3 & h_3 & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot \\ & & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & & h_{n-2} & u_{n-1} \end{bmatrix} \cdot \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \\ \cdot \\ \cdot \\ \cdot \\ \kappa_{n-2} \\ \kappa_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \cdot \\ \cdot \\ \cdot \\ v_{n-2} \\ v_{n-1} \end{bmatrix}, \tag{3}$$

where

$$u_i = 2 \cdot (h_i + h_{i-1}) \ ; \ b_i = \frac{6}{h_i} \cdot \left(y_{i+1} - y_i\right) \ ; \ v_i = b_i - b_{i-1}. \tag{4}$$

Once we determine the curvature values κ_i, by solving Eq. (3), we define the spline function as

$$S_i(x) = y_i + A_i \cdot (x - x_i)^3 + B_i \cdot (x - x_i)^2 + C_i \cdot (x - x_i); \quad i = 0, \ldots, n-1, \tag{5}$$

where the coefficients are given by

$$A_i = \frac{1}{6 \cdot h_i} \cdot (\kappa_{i+1} - \kappa_i) \ ; \ B_i = \frac{\kappa_i}{2} \ ; \ C_i = -\frac{h_i}{6} \cdot \kappa_{i+1} - \frac{h_i}{3} \cdot \kappa_i + \frac{1}{h_i} \cdot \left(y_{i+1} - y_i\right). \tag{6}$$

2.2. Tension splines

In some problems of adjusting discrete data, it is useful to have a parameter called "tension, τ." When τ has a small value, the resulting curve approaches a cubic spline. When τ tends to $+\infty$, the resulting curve approaches a linear spline. For the same sequence of sorted point pairs mentioned above, the tension spline satisfies.

1. $T \in C^2[x_0, x_n]$ and $T(x_i) = y_i \ ; \ i = 0, \ldots, n.$

2. On every interval $[x_i, x_{i+1}] : T^{(IV)} - \tau^2 \cdot T^{(II)} = 0.$

That is, $T : \mathbb{R} \to \mathbb{R}$ has continuity C^4 globally, interpolates to the data, and satisfies certain ordinary differential equation in each subinterval. It is clear that the prescription $\tau = 0$ leads to cubic polynomials when solving the equation. To determine T, we proceed similarly to the case of cubic splines, i.e., $\kappa_i \equiv T''(x_i)$:

$$T^{(IV)} - \tau \cdot T^{(II)} = 0 \ ; \ T(x_i) = y_i \ ; \ T(x_{i+1}) = y_{i+1}. \tag{7}$$

The solution is given by [2]:

$$T(x) = \frac{[\kappa_i \cdot \sinh(\tau \cdot \widehat{x}) + \kappa_{i+1} \cdot \sinh(\tau \cdot \tilde{x})]}{\tau^2 \cdot \sinh(\tau \cdot h_i)} + \left(y_i - \frac{\kappa_i}{\tau^2}\right) \cdot \frac{\widehat{x}}{h_i} + \left(y_{i+1} - \frac{\kappa_{i+1}}{\tau^2}\right) \cdot \frac{\tilde{x}}{h_i}, \tag{8}$$

where $\widehat{x} = (x_{i+1} - x)$ and $\tilde{x} = (x - x_i)$, and we compute the curvatures by solving the system:

$$\alpha_{i-1}\kappa_{i-1} + \left(\beta_{i-1} + \beta_i\right) \cdot \kappa_i + \alpha_i \kappa_{i+1} = \left(\gamma_i - \gamma_{i-1}\right), \tag{9}$$

and $1 \leq i \leq (n-1)$, and the arguments are ($\kappa_o = \kappa_n = 0$):

$$\alpha_i = \frac{1}{h_i} - \frac{\tau}{\sinh(\tau \cdot h_i)} \;\; ; \;\; \beta_i = \frac{\tau \cdot \cosh(\tau \cdot h_i)}{\sinh(\tau \cdot h_i)} - \frac{1}{h_i} \;\; ; \;\; \gamma_i = \frac{\tau^2 \left(y_{i+1} - y_i\right)}{h_i}. \tag{10}$$

2.3. Numerical examples

2.3.1. Example 1

Fit the following collection of point pairs using a natural cubic spline.

x	0	1	2	3	4
y	−8	−7	0	19	56

We assume $\kappa_0 = \kappa_4 \equiv 0$; thus $h_o = x_1 - x_0 = 1 = h_1 = h_2 = h_3$, and $u_1 = 2 \cdot (h_1 - h_0) = 4 = u_2 = u_3$. The tridiagonal system (3) yields

$$\begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix} = \begin{bmatrix} 36 \\ 72 \\ 108 \end{bmatrix} \Rightarrow \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix} \cong \begin{bmatrix} 6.4285 \\ 10.2857 \\ 24.4285 \end{bmatrix}. \tag{11}$$

As an illustration, $S_2(x)$ is given by

$$S_2(x) = 2.3571 \cdot (x-2)^3 + 5.1428 \cdot (x-2)^2 + 11.5 \cdot (x-2), \tag{12}$$

for instance, $S_2(3) \cong 18.9999$ and $S_2(2.5) \cong 7.3303$.

2.3.2. Example 2

Figure 1 depicts radial velocity profiles that represent the laminar fluid flow within a pipeline.

These velocity profiles were obtained by solving the Navier-Stokes equations under simplifying assumptions. The symbols represent the discrete point pairs, the abscissas correspond to the normalized radial coordinate from the center, and the y-coordinates are the normalized radial velocities. We fit all data sets by using natural splines. To solve the system (3), we recommend the Thomas method, i.e., a direct frontal solver for tridiagonal matrixes [1–3]. We also recommend employing a quick-search algorithm to evaluate the piecewise function. Indeed, for an arbitrary x, we need to determine what is the interval where this abscissa lies, i.e., $x \in [x_i, x_{i+1}]$.

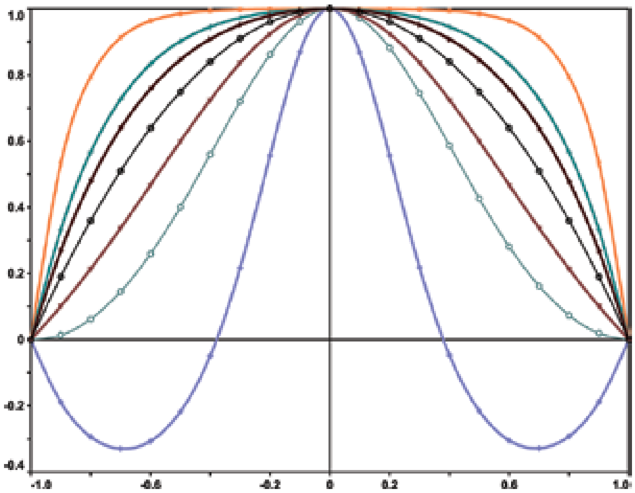


Figure 1. Discrete velocity profiles that were fitted by splines.

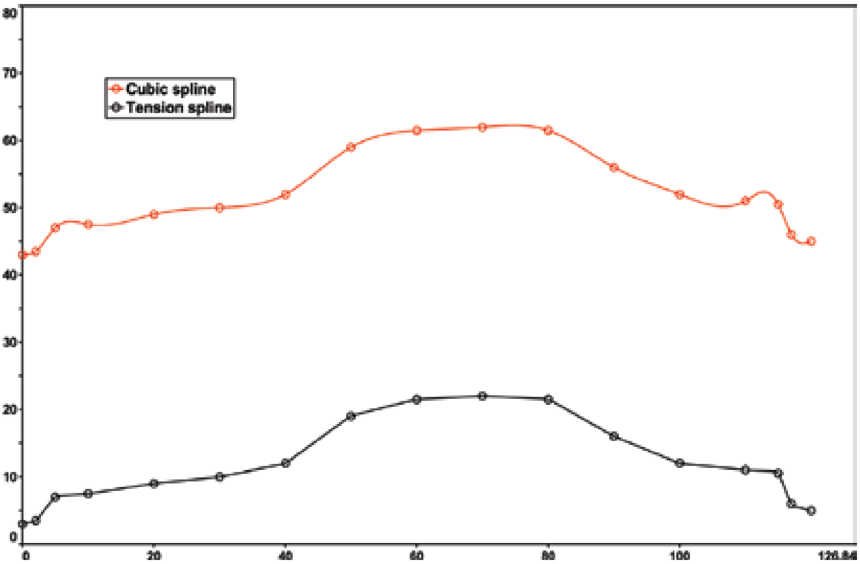


Figure 2. It depicts a car-like profile that we fit by cubic and tension splines.

2.3.3. *Example 3*

We finalize the examples by comparing cubic and tension splines. **Figure 2** depicts a car-like profile polygon that we would like to interpolate. We try both splines mentioned above. We highlight in red color the cubic spline (top) interpolant, while the tension spline is black (bottom curve). We observe that the cubic spline experiences inflection points because the car-shaped polygon is challenging. This latter is the kind of application for tension splines where we seek to attract the spline toward the input polyline. We notice that we achieve that goal herein.

3. Bèzier, B-spline, and NURBS curves

The appropriate representation and meshing of the computational domain for the physical problem under study are necessary premises for a satisfactory computer simulation. In fact, one of the most demanding computational tasks in a simulation is defining the geometry because it will impact many aspects of the study such as the grid generation process [4]. Therefore, special methods must be applied to fit discrete data without sudden changes in curvature. The approach should be free of inflection points, and at minimum, it must enforce continuity C^2 of the fitted curve. In this chapter, this goal is achieved by using Bèzier, B-spline, and NURBS curves and surfaces [5, 6].

A Bèzier curve (*BC*), $\underline{B}$, shown in **Figure 3**, is obtained by specifying the coordinates of a series of points in space, such that only the first and last ones fall on the originally given curve. All these points are known as control points, and the polyline resulting from connecting them with straight lines is called control polygon, which mimics the original curve, allowing an easy control of its shape. Although inflection points may be present in Bèzier curves, they are less common than in polynomials or other analytical functions [5, 6].

Global Bèzier curves, i.e., only one curve represents the given polyline, provide a powerful tool in geometry definition; however, complex shapes require a large number of constraints,

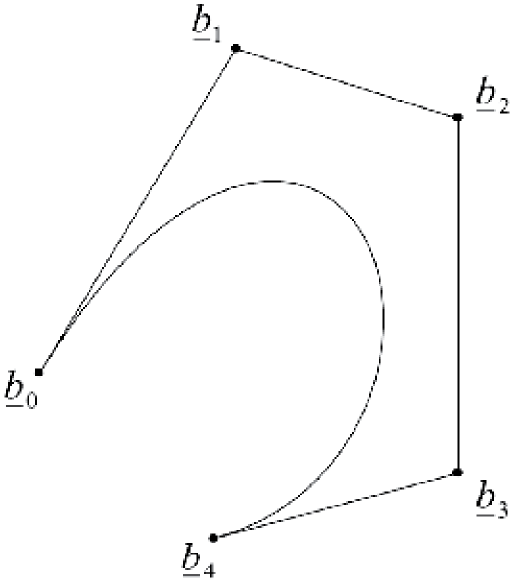


Figure 3. Fourth-order Bèzier curve with highlighted control points.

making their use prohibitive. It is therefore beneficial to represent them by using piecewise continuous Bèzier curves called B-spline curves [5]. In fact B-spline curves are a widely utilized representation for geometrical entities in computer-aided geometric design (*CAGD*) systems. Their convex hull, local support, shape-preserving forms, affine invariance, and variation-diminishing properties are extremely attractive in engineering design applications [4].

A particular Bèzier curve is set up by its parametric representation; let $\underline{\mathcal{B}} : \mathbb{R} \to \mathbb{R}^2$ be defined by

$$\underline{\mathcal{B}}(t) = \sum_{i=0}^{m} \underline{b}_i \cdot B_i^{(m)}(t), \quad t \in I = [0, 1], \tag{13}$$

here, m denotes the order or degree of the curve, $B_i^{(m)}(t)$ are the Bernstein polynomials, defined as

$$B_i^{(m)}(t) = \frac{m!}{i!(m-i)!} t^i \cdot (1-t)^{m-i} \quad ; \quad \sum_{i=0}^{m} B_i^m(t) = 1, \tag{14}$$

and $\underline{b}_i$ are the control points. Notice in Eq. (14) that Bernstein polynomials satisfy the barycentric property, meaning that they add up to 1, which explains why a given curve cannot be outside its control polygon that is the convex-hull property. The control points of a given BC can be calculated in several ways since the Bèzier curve evaluated in $t = t_k$ must provide the corresponding base point $\underline{p}_k$; a linear system of equations can be formed for the unknown control points as

$$\sum_{i=0}^{m} \underline{b}_i \cdot B_i^{(m)}(t_k) = \underline{p}_k, \quad k = 0, .., m, \tag{15}$$

where the number of base points equals $(m + 1)$; we compute the value of the parameter t_k by [6, 7]

$$t_k = \frac{s_k}{s_m}; \quad s_0 = 0; \quad s_k = s_{k-1} + \left\| \underline{p}_k - \underline{p}_{k-1} \right\|, \quad k = 1, \ldots, m, \tag{16}$$

which is the well-known chord-length parametrization.

The last approach is a powerful tool in curve design, but it has a limitation: if the geometry that we model has a complex shape (i.e., a significant number of base points), then its Bèzier curve representation may be of a prohibitively high degree. Since the Bèzier curve is forced to satisfy several constraints according to Eq. (15), the resulting curve may experience inflection points and sudden changes in curvature (see **Figure 4**, where $\underline{p}_k$ points in Eq. (15) are represented by circles). For practical purposes, degrees exceeding 10 are prohibitive [5, 6].

Such complex geometries can be modeled using piecewise polynomial curves named B-spline curves [5, 6] (see **Figure 5**). B-spline curves are a set of Bèzier curves of mth degree that must

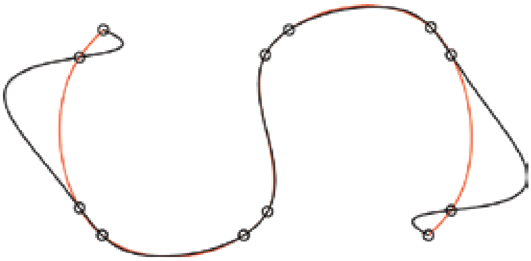


Figure 4. We interpolated with a Bèzier (black) and a cubic B-spline (red) curves.

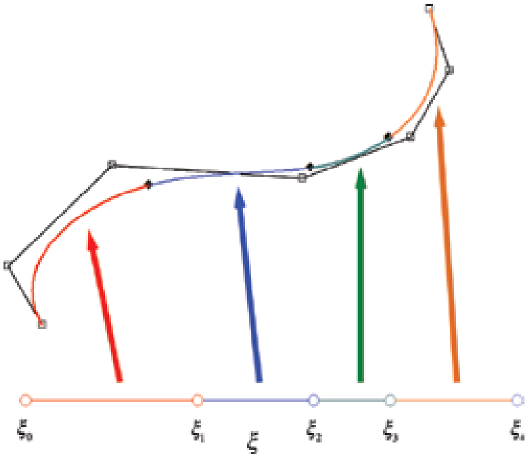


Figure 5. A B-spline curve, $\underline{C}(\xi)$, is the union of piecewise continuous curves.

satisfy at least the $C^{(m-1)}$ continuity. A spline curve $\underline{C}$ is the continuous mapping of a collection of global parameter values $\xi_0, \xi_1, \ldots, \xi_{L-1}, \xi_L$ into $\mathbb{R}^2$, where each interval $[\xi_i, \xi_{i+1}]$ is mapped onto a polynomial curve segment as shown in **Figure 5**. We define $\overline{\Omega} = [\xi_0, \xi_L]$ as the computational space. A local coordinate t for the interval $[\xi_i, \xi_{i+1}]$ can be defined by setting [5]:

$$t = \frac{\xi - \xi_i}{\xi_{i+1} - \xi_i} = \frac{\xi - \xi_i}{\Delta_i}, \quad \xi \in [\xi_i, \xi_{i+1}]. \tag{17}$$

3.1. C^2 cubic curves

Let $\underline{d}_{-1}, \underline{d}_0, \ldots, \underline{d}_L, \underline{d}_{L+1}$ be a set of $(L+3)$ points defining the de Boor's polygon that generates L individual cubic curves as shown in **Figure 6**. The required $(3L+1)$ Bèzier control points are calculated with the aid of C^1 and C^2 continuity criteria. C^1 conditions lead to

$$\underline{b}_{3i} = \frac{\Delta_i}{\Delta_{i-1} + \Delta_i} \underline{b}_{3i-1} + \frac{\Delta_{i-1}}{\Delta_{i-1} + \Delta_i} \underline{b}_{3i+1}, \quad i = 1, \ldots, L-1, \tag{18}$$

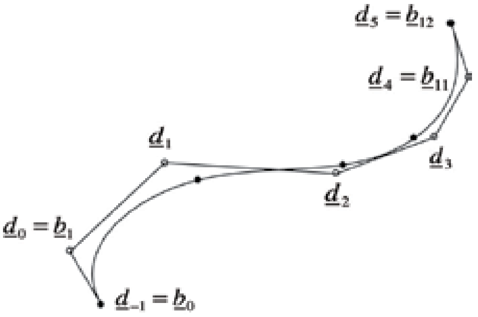

$\underline{d}_5 = \underline{b}_{12}$
$\underline{d}_4 = \underline{b}_{11}$
$\underline{d}_1$
$\underline{d}_3$
$\underline{d}_2$
$\underline{d}_0 = \underline{b}_1$
$\underline{d}_{-1} = \underline{b}_0$

Figure 6. A C^2 cubic curve with highlighted de Boor's and junction points.

while C^2 conditions require that

$$\begin{aligned} \underline{b}_{3i-2} &= \frac{\Delta_{i-1} + \Delta_i}{\Delta}\underline{d}_{i-1} + \frac{\Delta_{i-2}}{\Delta}\underline{d}_i, \\ \underline{b}_{3i-1} &= \frac{\Delta_i}{\Delta}\underline{d}_{i-1} + \frac{\Delta_{i-2} + \Delta_{i-1}}{\Delta}\underline{d}_i. \end{aligned} \tag{19}$$

where $i = 2, \ldots, L-1,$ and $\Delta = \Delta_{i-2} + \Delta_{i-1} + \Delta_i$. The end points are

$$\begin{gathered} \underline{b}_0 = \underline{d}_{-1} \ ; \ \underline{b}_1 = \underline{d}_0 \ ; \ \underline{b}_2 = \frac{\Delta_1}{\Delta_0 + \Delta_1}\underline{d}_0 + \frac{\Delta_0}{\Delta_0 + \Delta_1}\underline{d}_1, \\ \underline{b}_{3L-2} = \frac{\Delta_{L-1}}{\Delta_{L-2} + \Delta_{L-1}}\underline{d}_{L-1} + \frac{\Delta_{L-2}}{\Delta_{L-2} + \Delta_{L-1}}\underline{d}_L \ ; \ \underline{b}_{3L-1} = \underline{d}_L \ ; \ \underline{b}_{3L} = \underline{d}_{L+1}. \end{gathered} \tag{20}$$

This construction is due to W. Boehm [5].

For cubic curves more parametrizations are available [5, 6], for instance:

1. Uniform parametrization

$$\xi_i = i \ ; \ i = 0, \ldots, L. \tag{21}$$

2. Chord-length parametrization [5]

$$\begin{aligned} &\xi_0 = 0.0; \ \ \xi_1 = \|\underline{d}_1 - \underline{d}_{-1}\|, \\ &\xi_i = \xi_{i-1} + \|\underline{d}_i - \underline{d}_{i-1}\|; \ \ i = 2, \ldots, L-1, \\ &\xi_L = \xi_{L-1} + \|\underline{d}_{L+1} - \underline{d}_{L-1}\|. \end{aligned} \tag{22}$$

3. A parametrization proposed by the author [6]

$$\begin{aligned} \xi_0 &= 0.0, \\ \xi_{i+1} &= \xi_i + \|\underline{d}_i - \underline{d}_{i-1}\| + \|\underline{d}_{i+1} - \underline{d}_i\| + \|\underline{d}_{i+2} - \underline{d}_{i+1}\|, \\ i &= 0, \ldots, L-1. \end{aligned} \tag{23}$$

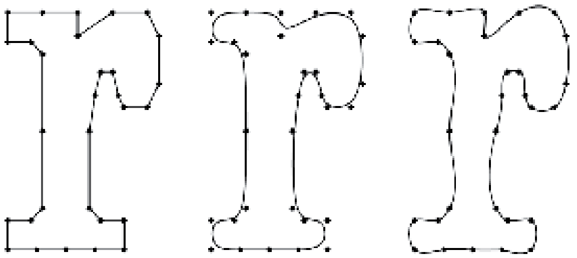


Figure 7. The inverse design and interpolation problems.

This latest parametrization has the advantage that it yields to a symmetric curve if the control polygon is symmetric as well, which may be interesting for certain applications.

3.2. Inverse design and interpolation problems

A two-dimensional geometric description based on B-spline curves requires the definition of a control polygon (the de Boor's polygon) that mimics the curve. Therefore two approaches are possible. The first method consists of providing the set of the de Boor's points (i.e., define the de Boor's polygon interactively from user's input) that defines the composite curve, which is known as "inverse design," and it has its application in "TrueType" font technology as shown in **Figure 7**. The second approach consists of defining the set of base points and then solves a linear system of equations for the de Boor's points such that the resulting curve passes through them. This latter is known as the "interpolation" problem.

Figure 7 shows the difference between the above approaches; from left to right, it has the de Boor's control polygon, inverse design, and interpolation problems both taking into account the same polygon as an argument with cubic curves.

3.3. Interpolation with cubic curves

In order to interpolate with cubic B-spline curves, we find unknown junction points such that they pass through a given set of data points $\underline{x}_0, \ldots, \underline{x}_L$ and corresponding parameter values $\xi_0, \xi_1, \ldots, \xi_{L-1}, \xi_L$. A composite cubic curve $\underline{C}$, determined by its de Boor's polygon [4, 5] $\underline{d}_{-1}, \ldots, \underline{d}_{L+1}$ such that $\underline{C}(\xi_i) = \underline{x}_i$, is required. The solution to this problem is obtained by finding the relationship between the data points $\underline{x}_i$ and the control vertices $\underline{d}_i$. This leads to the following linear system of equations for the unknown de Boor's points [5]:

$$\alpha_i \cdot \underline{d}_{i-1} + \beta_i \cdot \underline{d}_i + \gamma_i \cdot \underline{d}_{i+1} = (\Delta_{i-1} + \Delta_i) \cdot \underline{x}_i; \quad i = 1 \ldots L-1, \tag{24}$$

where (with $\Delta_{-1} = \Delta_L = 0$)

$$\begin{aligned} \alpha_i &= \frac{(\Delta_i)^2}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i}, \\ \beta_i &= \frac{\Delta_i(\Delta_{i-2} + \Delta_{i-1})}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} + \frac{\Delta_{i-1}(\Delta_i + \Delta_{i+1})}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}, \\ \gamma_i &= \frac{(\Delta_{i-1})^2}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}. \end{aligned} \tag{25}$$

If the two Bèzier points $\underline{b}_1$ and $\underline{b}_{3L-1}$ are arbitrarily chosen, the following linear system of equations is obtained [5, 6]:

$$\begin{bmatrix} 1 & & & & & & \\ \alpha_1 & \beta_1 & \gamma_1 & & & & \\ & & \ddots & & & & \\ & & & & \alpha_{L-1} & \beta_{L-1} & \gamma_{L-1} \\ & & & & & & 1 \end{bmatrix} \cdot \begin{bmatrix} \underline{d}_0 \\ \underline{d}_1 \\ \vdots \\ \underline{d}_{L-1} \\ \underline{d}_L \end{bmatrix} = \begin{bmatrix} \underline{r}_0 \\ \underline{r}_1 \\ \vdots \\ \underline{r}_{L-1} \\ \underline{r}_L \end{bmatrix}, \tag{26}$$

where

$$\begin{aligned} \underline{r}_0 &= \underline{x}_0, \\ \underline{r}_i &= (\Delta_{i-1} + \Delta_i) \cdot \underline{x}_i \ ; \ i = 1 \dots L-1, \\ \underline{r}_L &= \underline{x}_L. \end{aligned} \tag{27}$$

The first and last vertices of the polygon are given by

$$\underline{d}_{-1} = \underline{x}_0, \ \ \underline{d}_{L+1} = \underline{x}_L. \tag{28}$$

The points $\underline{b}_1$ and $\underline{b}_{3L-1}$ can be calculated from a given end condition. There are two possibilities for the choice of B-spline ending conditions. A natural spline requires that

$$\begin{aligned} \frac{d^2}{dt^2}\underline{s}_0(0) &= 6 \cdot (\underline{b}_2 - 2\underline{b}_1 + \underline{b}_0) = \underline{0}, \\ \frac{d^2}{dt^2}\underline{s}_{L-1}(1) &= 6 \cdot (\underline{b}_{3L} - 2\underline{b}_{3L-1} + \underline{b}_{3L-2}) = \underline{0}. \end{aligned} \tag{29}$$

Using the relations in Eq. (29), we obtain that

$$\begin{aligned} \left[2 - \frac{\Delta_1}{\Delta_0 + \Delta_1}\right] \cdot \underline{d}_0 - \frac{\Delta_0}{\Delta_0 + \Delta_1}\underline{d}_1 &= \underline{x}_0, \\ \left[2 - \frac{\Delta_{L-2}}{\Delta_{L-2} + \Delta_{L-1}}\right] \cdot \underline{d}_L - \frac{\Delta_{L-1}}{\Delta_{L-2} + \Delta_{L-1}}\underline{d}_{L-1} &= \underline{x}_L. \end{aligned} \tag{30}$$

These two equations replace the first and last rows of the linear system of equations given in Eq. (26). Notice that in either case, natural ending conditions or prescribed tangent vectors, the linear system in Eq. (24) is a tridiagonal matrix. Since the coefficient matrix is real and scalar, and the left- and right-hand-side vectors are in fact hypervectors (i.e., an array of vectors), it is recommendable to use a type of Gaussian elimination method against multiple right-hand sides to achieve performance [6, 7]. If we recompute the interpolant in **Figure 4** but this time with a B-spline cubic curve, we then obtain a smoother and steadier interpolant free of unwelcome inflection points.

3.4. NURBS curves

NURBS curves are useful when we require an exact geometrical representation of some entities, such as circles, parabolas, ellipses, spheres, cylinders, etc. This is precisely the case in various applications in aerospace and mechanical engineering where NURBS are quite popular [4, 5, 8–10]. For instance, a NURBS curve is defined by its rational representation in Eq. (31):

$$\underline{\mathcal{R}}(t) = \frac{\sum_{i=0}^{m} w_i \cdot B_i^m(t) \cdot \underline{b}_i}{\sum_{i=0}^{m} w_i \cdot B_i^m(t)}, \tag{31}$$

where the weights w_i are positive real scalars. The usual B-spline definition is recovered if all those weights equal 1. Generally speaking, the weights play the role of attracting the curve toward its control polygon when we increase their values [5, 8]. It turns out that specific weights lead to exact representation of circles, for instance, as shown in **Figure 8**, where the real numbers depicted are the given weights. A circle can be exactly represented by three- or four-quadratic arcs (left and right side, respectively, in **Figure 8**); this latter alternative is more attractive, for instance, to generate a surface of revolution [4, 5, 8]. NURBS also provide exact representation for 3D surfaces such as spheres and cylinders as well as volumes [4, 10]. The reader may refer to [4, 5, 8–11] for further details for this well-established area of computational geometry. We depict an example of practical interest, in the context of geomechanics, in **Figure 9**. Herein we precisely represent a near-borehole section. To describe the borehole geometry, we employ four line segments and a quadratic arc as shown.

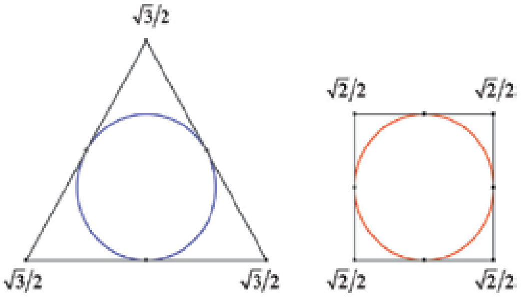
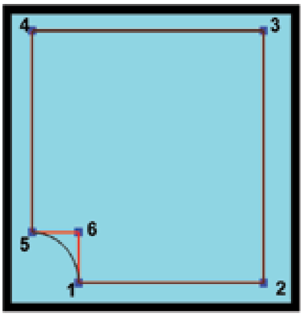


Figure 8. Two exact equivalent representations for a circle (the remaining weights equal 1).



$$r_w = 5.0,\ \alpha = 5.0,\ r_o = \alpha \cdot r_w$$

$$(\underline{b}_i, w_i) = (x_i, y_i, w_i):$$

$$\begin{bmatrix} 1 & r_w & r_w & 1.0 \\ 2 & r_o & 0 & 1.0 \\ 3 & r_o & r_o & 1.0 \\ 4 & 0 & r_o & 1.0 \\ 5 & 0 & r_w & 1.0 \\ 6 & r_w & r_w & \sqrt{2}/2 \end{bmatrix}$$

$$[L] = \text{inches}$$

Figure 9. We represent a near borehole “exactly” by NURBS as shown.

In order to interpolate a set of points with a NURBS curve in two or three dimensions, one may follow the same procedure described with B-spline curves except that a mapping to $\mathbb{R}^4$ must be carried out first. The new input points $\underline{\tilde{x}}_i$ are given by

$$\underline{\tilde{x}}_i = w_i \cdot \begin{pmatrix} x_i & y_i & z_i & 1 \end{pmatrix}^T; \quad \underline{\tilde{x}}_i \in \mathbb{R}^4, \tag{32}$$

then the linear system in Eq. (24) with the right boundary conditions can be solved with $\underline{x}_i$ replaced by $\underline{\tilde{x}}_i$, as defined by the Eq. (32). After solving this system, the solution control polygon is still in $\mathbb{R}^4$, which implies that a mapping back to $\mathbb{R}^3$ is required:

$$\underline{d}_i = \frac{1}{w_i} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \underline{\tilde{d}}_i \;\; ; \;\; \underline{d}_i \in \mathbb{R}^3. \tag{33}$$

The latter is a straightforward procedure to reuse the subroutines already developed for B-spline curves.

3.5. Numerical examples

We implemented the proposed approaches in a computer code named "LogProc" which is a graphical user interface application developed with the C++ programming language. LogProc is proprietary software, but a free community version will be available for download from www.logproc.com. All examples were obtained applying the proposed knot sequence (23) to construct cubic composite curves. The empty circles represent de Boor's points (sample inverse designs) or base points (interpolation problems). We utilized the natural end condition in examples in **Figures 10** and **11** and the prescribed tangent in the remaining cases. We depict a typical font design application in **Figure 10** where we represent some alphabet's letters. Notice that an approach like this is appropriate to construct font outlines because they can be scaled

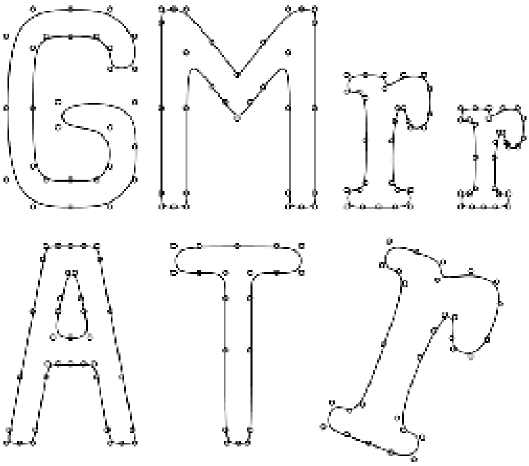


Figure 10. A typical font design application.

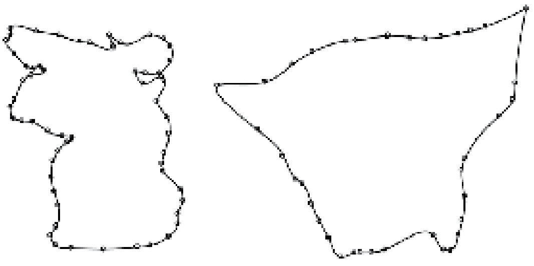


Figure 11. A pair of petroleum reservoir outlines.

and rotated. This feature is of significant interest in the "TrueType" font technology where outlines that are insensible to the resolution of the physical device in which they will be shown, such as monitors or printers, must be obtained [6]. We added a bonus section on the numerical implementation for computing splines that is available online, https://www.logproc.com/book-chapter-splines. We also include most sample datasets for downloading. We also recommend there suitable open-source libraries that are convenient to use. All vector and raster plots were prepared by logproc which can create high-quality PDF files in Windows 7 and 10, for instance.

Figure 11 shows a zenith view of a pair of petroleum reservoir outlines. Few base points permit to approximate their complex shape. These shapes could be used as arguments to generate an unstructured mesh appropriate to simulate the flow in a porous media [7].

We interpolate a discrete blade geometry approximation in both **Figures 12** and **13**. A3K7 profiles are constructed in the same way as NACA 65, but they present rounded trailing edges [6]. A3K7's shape is represented by 46 base points and circle arcs both in leading and trailing edges. Therefore the curves that interpolate the data points are tangents to circle arcs to ensure C^1 continuity between these entities. The profiles in the left of **Figure 12** were obtained applying Eq. (15) to construct C^1 single Bèzier curves (two curves, each of them approximating to suction and pressure sides, respectively). However, note that the resulting curves have unexpected inflection points on the suction side. The enforcement of continuity conditions

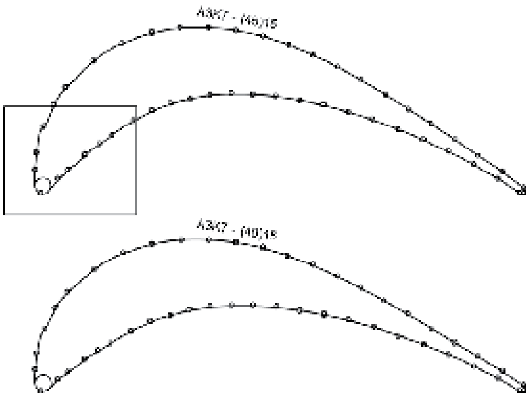


Figure 12. A3K7 interpolated by Bèzier curves.

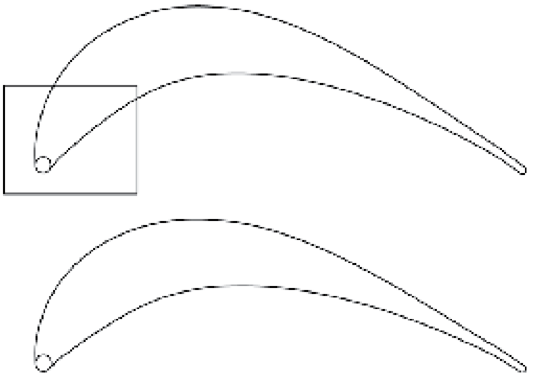


Figure 13. A3K7 interpolated by B-spline curves.

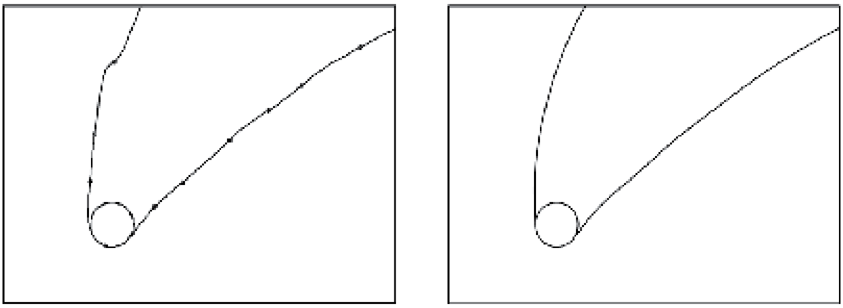


Figure 14. Details highlighted in **Figures 12** and **13**.

implies a large number of constraints after Eq. (15), which explains this behavior. If we replace the former Bèzier curves by two B-spline curves, computed from Eqs. (24) and (29), we obtain a smooth A3K7 geometry description (see profiles in the right-hand side of **Figure 12**). **Figure 14** zooms in to show that the resulting geometrical description is smooth and free of unwelcome features such as inflection points.

4. Interpolation surfaces

Let $\underline{S}^{Int} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ be a two-parameter mapping which represents a given surface. If structured data, i.e., tensor product data, needs to be interpolated, one may expect to come up with tensor product surfaces as well, where two parameters (ξ, η) allow covering two different directions associated with the surface. In the computational space, i.e., in the plane (ξ, η), the domain, $\overline{\Omega} = \left[\xi_0, \xi_{L_u}\right] \times \left[\eta_0, \eta_{L_v}\right]$, is a rectangle, and its image is the surface in 3D as shown in **Figure 15**, where L_ξ and L_η are the number of curves in their respective directions.

B-spline tensor product surfaces allow interpolating structured data, and they are defined as the tensor product of two families of curves $\underline{C}_i^k(\xi)$ and $\underline{D}_j^l(\eta)$, which is

$$\underline{\mathcal{L}}^{Int}\left(\xi_i,\eta_j\right) = \underline{x}_{ij};\quad \underline{\mathcal{S}}^{Int}_{klij}(\xi,\eta) = \underline{\mathcal{C}}^{k}_{i}(\xi)\otimes\underline{\mathcal{D}}^{l}_{j}(\eta);\quad (\xi,\eta)\in\overline{\Omega} \tag{34}$$

In applications of practical interest, usually cubic piecewise continuous curves are preferred because they provide a global C^2 representation that is smooth enough, called a bicubic surface [5, 8, 9].

4.1. Creating a surface of interpolation

The following steps describe creating a surface of interpolation:

1. An input control polygon, whose points are in $\mathbb{R}^3$, is provided. They correspond to data that is structured and ordered, which is usually a matrix-type array of points (see left side of **Figure 16**). For simplicity, points in the i-direction are associated with the ξ parameter while $j's$ are associated with η.

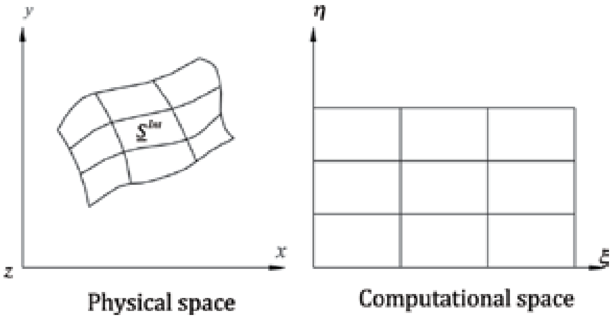


Figure 15. We depict the mapping between physical and computational spaces.

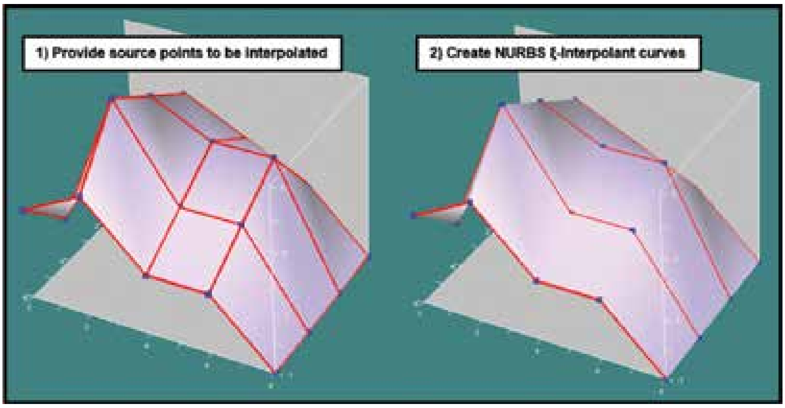

1) Provide source points to be interpolated
2) Create NURBS ξ-interpolant curves

Figure 16. The first two steps to interpolate structured data are depicted: input control polygon (left) and ξ-interpolants (right) are shown.

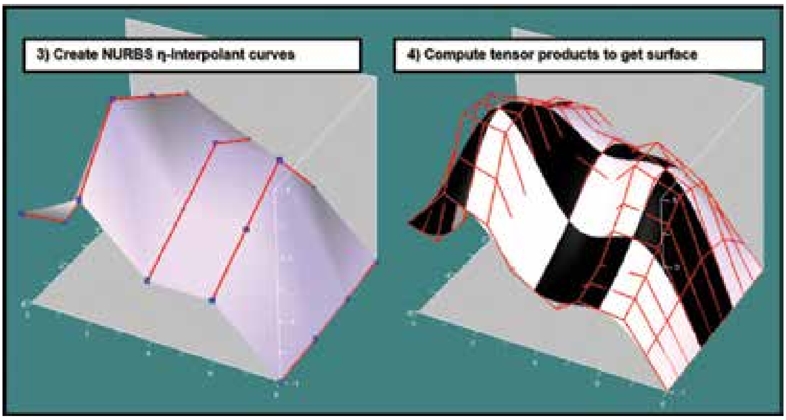

3) Create NURBS η-interpolant curves
4) Compute tensor products to get surface

Figure 17. The last two steps to interpolate structured data are depicted: η-interpolants (left) and resulting bicubic patches are shown. The de Boor's control polygon is highlighted in red lines.

2. Create interpolation curves with constant values of η, so-called ξ-interpolants (see right side of **Figure 16**).

3. Proceed accordingly with previous step, interpolation curves with constant values of ξ; so-called η-interpolants are created this time (see left side of **Figure 17**).

4. Compute the tensor product between ξ- and η-interpolants in order to get bicubic patches (see right side of **Figure 17**).

The right-hand side in **Figure 17** shows typical bicubic patches as a chessboard surface emphasizing that we deal with a piecewise continuous entity. The computational cost associated with the above algorithm is reasonable because the most expensive part is computing the interpolants (see Section 3).

5. Translational surfaces

These surfaces are again a two-parameter mapping, $\underline{\sigma}^T : \mathbb{R}^2 \to \mathbb{R}^3$, but their construction procedure is simpler than interpolation surfaces; see, for instance, [5, 8, 9]. The idea here is just literally translating a curve $\underline{\alpha}$ along another curve $\underline{\beta}$, which yields

$$\underline{\sigma}^T(\xi, \eta) = \underline{\alpha}(\xi) + \underline{\beta}(\eta). \tag{35}$$

This idea became very popular in CAGD systems long time ago. Those systems usually support a command which allows extruding a geometrical entity, for instance, a cylinder can be easily created by extruding a circle along a straight line. **Figure 18** shows the above procedure applied to an aircraft wing where an NURBS airfoil profile was translated or extruded along a straight line accordingly. We interpolated an NACA 65 polyline with a NURBS curve as we mentioned in Section 3.4.

This procedure becomes very useful in the geometrical reconstruction of oil reservoirs (*RS*). Indeed, we reconstructed the geometry of RS in [12] by using B-spline surfaces. The technique exploits input mesh's simplicity to build a robust piecewise continuous geometrical representation using Bèzier bicubic patches. We manage the reservoir's topology with interpolation

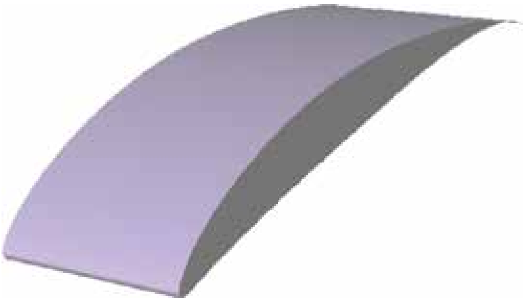


Figure 18. An aircraft wing by translating an NACA profile accordingly.

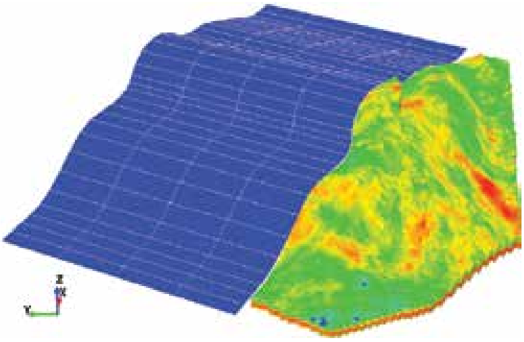


Figure 19. A translational surface.

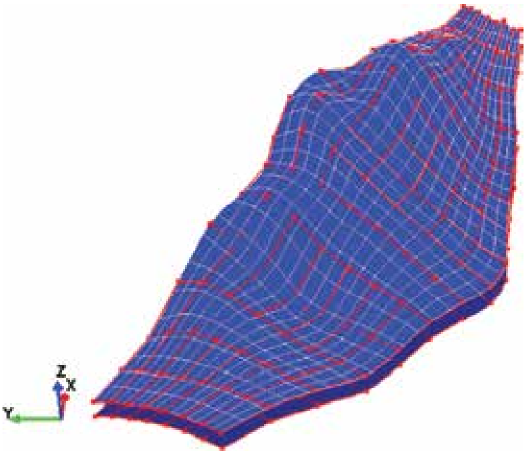


Figure 20. An interpolation surface.

surfaces, while translational surfaces allow extrapolating it toward its side burdens. After that, transfinite interpolation can be applied to generate decent hexahedral meshes. **Figure 19** shows a sample translational surface that we obtain by extruding a curve that interpolates the reservoir's edge as shown. We render the surfaces in blue color with a white wireframe, while the RS is the color-contoured surface that represents the porosity, a scalar property. We tackle the RS itself after interpolating the control polygon that **Figure 20** highlights in red color. The polygon is a 17 × 9 array of points representing the RS topology. The procedure works well for a variety of so-called open-to-the-public RS data sets that we reconstructed in [12]. It is also possible to utilize these NURBS curves and surfaces as interfaces for gluing nonmatching interfaces for the finite element method as we showed in [13].

6. Duchon splines

In the context of applications in statistical analysis involving very high dimensional data sets, response surfaces are growing popularity. By running the simulations at a set of points (e.g., experimental design) and fitting response surfaces, i.e., splines, for instance, to the resulting input-output data that is characterized by sparsity, we can obtain fast surrogates for the objective function for optimization purposes [14, 15]. The appeal of the latter approach goes beyond reducing runtime. Since the method begins with experimental design, statistical analyses can be done to identify which input variables are the most important, and thus we can create "main effect plots" to visualize input-output relationships [14]. We must recognize interpolation methods in which the basis functions are fixed and those in which they have parameters that are tuned (e.g., kriging, which has a statistical interpretation that allows one to construct an estimate of the potential error in the interpolator). We refer the reader to [14, 15] for further reading.

There are different ways to approximate a function of several variables: multivariate piecewise polynomials, splines, and tensor product methods, among others. All these approaches have advantages and drawbacks, but if the rank of the linear system to solve may become large, a natural choice is radial basis functions, which are also useful in lower dimensional problems [14, 16, 17]. This may be particularly true if the input data is scattered, which excludes tensor product methods at first glance. Duchon splines are a class of positive definite and compactly supported radial functions, which consist of univariate polynomial within their support. It can be proven that they are of minimal degree and unique up to a constant factor, for given smoothness and space dimension [18]. They are particularly suitable to compute interpolants for very large scatter datasets [17].

Duchon splines, denoted herein as s, are defined by [17, 18]

$$\begin{aligned} s(\underline{x}) &= \sum_j \lambda_j \cdot \varphi\left(\rho_j\right) + p_n(\underline{x}) \ \ ; \ \ n = 2, 3 \\ \rho_j &= \left\|\underline{x} - \underline{x}^j\right\| \\ \varphi(\rho) &= \rho^2 \ln \rho, \end{aligned} \tag{36}$$

where $p_n(\underline{x})$ is a linear polynomial in two or three dimensions:

$$\begin{aligned} p_2(\underline{x}) &= ax + by + c \\ p_3(\underline{x}) &= dx + ey + fz + g \ ; \quad \lambda_j, a, \ldots, g \in \mathbb{R}. \end{aligned} \tag{37}$$

Notice that λ_j and the polynomial coefficients are all scalar quantities. In order to guarantee existence and uniqueness for these splines, an orthogonality condition with respect to linear polynomials is enforced, for instance, in two dimensions this yields to

$$\sum_j \lambda_j = \sum_j \lambda_j x_j = \sum_j \lambda_j y_j = 0. \tag{38}$$

By considering this result, the interpolation problem becomes

$$s(\underline{x}^i) = \sum_j \lambda_j \cdot \varphi\left(\rho_j^i\right) + p_n(\underline{x}^i) = F^i, \tag{39}$$

which implies m points plus $n+1$ orthogonality conditions; here, F^i are the nodal values to be interpolated. The resultant linear system to solve for is of $(m+n+1)$ rank.

Duchon splines are certainly suitable to interpolate scattered data sets that we cannot tackle with the tensor product surfaces that we discussed before. Indeed, **Figure 21** depicts such an application, in optimization, where an objective function that we wish to minimize was sampled randomly by Monte-Carlo (*MC*) realizations. To compute a minimum, we interpolate the black dots, and then we minimize the resulting spline with standard Newton stochastic techniques [15]. It is true that Duchon splines are a valid choice for "surrogate" models for such applications.

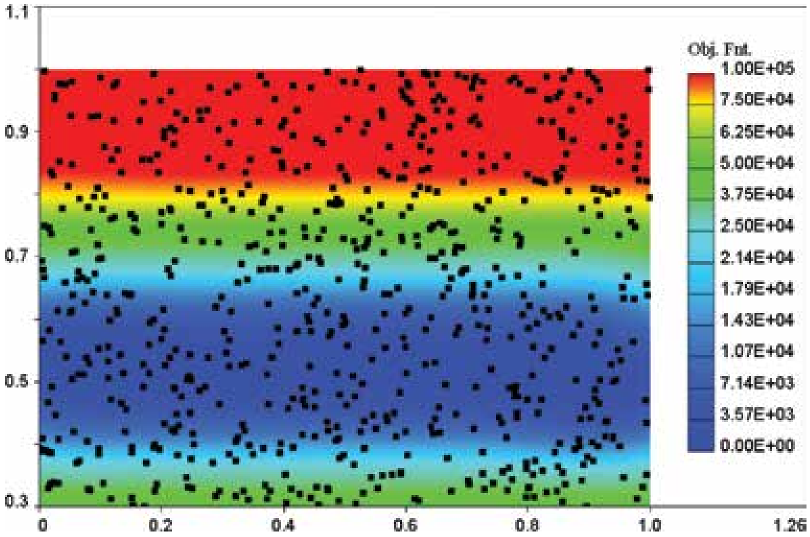


Figure 21. Discrete MC data with Duchon splines.

7. Concluding remarks

We presented a concise introduction to scalar and parametric spline interpolants. We introduced cubic and tension splines for scalar functions, and then we generalized them for the parametric case via Bèzier, B-spline, and NURBS curves. These latter entities are of the particular interest for applications in CAGD. We thus elaborated on topics such as inverse design and interpolation. We extended the treatment also to cover interpolation and translational surfaces with examples in mechanical and petroleum engineering. We wrapped up with the topic of interpolating sparse very high dimensional data sets via Duchon splines which are a kind of response surfaces suitable for applications in statistical analysis and optimization.

Acknowledgements

We acknowledge the financial support of the project "Reduced-Order Parameter Estimation for Underbody Blasts" funded by Army Research Laboratory.

Author details

Horacio Florez* and Belsay Borges

*Address all correspondence to: florezg@gmail.com

Computer Science Department, The University of Texas at El Paso, El Paso, USA

References

[1] Curtis F, Wheatley P. Applied Numerical Analysis. Pearson: USA; 2004

[2] Kincaid D, Cheney W. Numerical Analysis. USA: Brooks/Cole Publishing Company; 1991

[3] Atkinson K. An Introduction to Numerical Analysis. New York: John Wiley & Sons; 1978

[4] Thompson J, Weatherill N. Handbook of Grid Generation. CRC Press: Boca Raton; 1999

[5] Farin G. Curves and Surfaces for Computer-aided Geometric Design A Practical Guide. 4th ed. San Diego: Academic Press; 1993

[6] Florez H. A new method for building B-spline curves and its application to geometry design and structured grid generation. In: ASME International 2001 DETC Conference; Pittsburgh, Pennsylvania. 2001

[7] Florez H. Domain decomposition methods for geomechanics [Ph.D. thesis]. In: The University of Texas at Austin. 2012

[8] Farin G. NURBS from Projective Geometry to Practical Use. 2nd ed. A K Peters: Massachusetts; 1999

[9] Piegl L, Tiller W. The NURBS Book. 2nd ed. Berlin: Springer; 1997

[10] Hughes T, Cottrell J, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering. 2005;**194**:4135-4195

[11] De Berg M, van Kreveld M, Overmars M, Schwarzkopf O. Computational Geometry Algorithms and Applications. Berlin: Springer; 2000

[12] Florez H, Manzanilla-Morillo R, Florez J, Wheeler M. Spline-based reservoir's geometry reconstruction and mesh generation for coupled flow and mechanics simulation. Computational Geosciences. 2014;**18**:949-967

[13] Florez H, Wheeler M. A mortar method based on NURBS for curved interfaces. Computer Methods in Applied Mechanics and Engineering. 2016;**310**:535-566. ISSN 0045-7825. DOI: 10.1016/j.cma.2016.07.030

[14] Jones DR. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization. 2001;**21**(4):345-383

[15] Schuëller GI, Jensen HA. Computational methods in optimization considering uncertainties—An overview. Computer Methods in Applied Mechanics and Engineering. 2008;**198**(1):2-13

[16] Hagen H. Curve and Surface Design. Chapter 8: A Survey of Parametric Scattered Data Fitting Using Triangular Interpolants. USA: SIAM; 1992

[17] Buhmann M. Radial basis functions. Acta Numerica. 2000:1-38

[18] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics. 1995;**4**:389-396

An Algorithm Based on the Continuous Wavelet Transform with Splines for the Automatic Measurement of QT Dispersion: Validation and Application in Chronic Kidney Disease

María de Lourdes Corzo-Cuesta and
Carlos Alvarado-Serrano

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74864

Abstract

Chronic kidney disease (CKD) is considered a risk factor for the development of cardiovascular disease. QT interval is an electrocardiographic parameter that quantifies the duration of ventricular repolarization. An increase of its spatial variability measured from the selected leads of a standard electrocardiogram (ECG), named QT dispersion (QTd), is considered a risk factor for malign ventricular arrhythmias and sudden death in the CKD. An algorithm for automatic measurement of QTd in the ECG leads DI, aVF and V2 using the continuous wavelet transform with splines is presented. Validation of QRS complex detection has been done on records from MIT-BIH database, and the accuracy is 99.5%. Validation of detection of QRS wave onset and T wave end has been done on records from CSE and QT databases, and the measurements were within the tolerance limits for deviations with respect to the manual measurements defined by the experts. The algorithm was applied in two studies. In the first study, QTd was evaluated in normal subjects and patients with CKD. In the second study, QTd was analyzed in patients with CKD before, during and after the hemodialysis treatment. In both studies, the algorithm had a good performance for the QTd analysis.

Keywords: algorithm, wavelet transform, splines, electrocardiogram, QT dispersion, cardiovascular disease, chronic kidney disease

1. Introduction

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death globally, and an estimated 17.7 million people died from CVDs in 2015, representing 31% of all global deaths, of these deaths, an estimated 7.4 million were due to coronary heart disease (CHD) [1]. CHD is a narrowing or blockage of the blood vessels that supply blood and oxygen to the heart, then, the cells in the region served by the vessel will behave abnormally due to hypoxia (myocardial ischemia) or may die (myocardial infarction) [2]. In Mexico, data of the National Institute of Statistics and Geography (INEGI) showed that in 2015, the heart diseases are the main cause of death, and the most frequent is the ischemic heart disease (IHD) [3].

The kidneys are a pair of bean-shaped organs that lie on either side of the spine in the lower middle of the back. Its main function is to remove soluble waste products and excess water and electrolytes from the bloodstream. Chronic kidney disease (CKD) is defined as abnormalities of kidney structure or function, present for 3 months or more, with implications for health [4]. Abnormalities in kidney structure (damage) usually precede abnormalities in function. CKD is divided into five stages of increasing severity. Stage 5 referred to as kidney failure or end-stage renal disease (ESRD) is traditionally considered as the most serious outcome of CKD because there is total or near-total loss of kidney function and patients require treatment with dialysis or transplantation.

CKD is considered a risk factor for the development of cardiovascular disease (CVD) because patients with CKD are more likely to die of CVD than to develop kidney failure [4, 5]. Primary types of CVDs with a high prevalence in CKD are arterial vascular disease in its two subtypes: atherosclerosis and arteriosclerosis, and cardiomyopathy. Clinical presentations of atherosclerosis include IHD, manifested as angina, myocardial infarction and sudden cardiac death, which is common in CKD, cerebrovascular disease, peripheral vascular disease and heart failure [5]. Patients with ESRD requiring maintenance hemodialysis (HD) have a high mortality rate, which is primarily attributable to CVD, including ventricular arrhythmias and sudden death, and the incidence of arrhythmias increases during and immediately after HD [6, 7].

Therefore, the use and development of noninvasive techniques such as electrocardiography, which records the electrical activity generated by the muscles of the heart in the surface of the body, open a useful perspective for diagnosis and treatment in patients with heart diseases such as ischemia and infarction. The electrocardiogram (ECG) is the waveform produced by this electrical activity of the heart and its generation depends on four electrophysiological processes such as the formation of electrical impulse in the main heart pacemaker (sinoatrial node), the transmission of this impulse through specialized fibers in the conduction, the activation (depolarization) and the recovery (repolarization) of the myocardium [8].

The electrical activity generated by the heart can be modeled as a vector whose magnitude and direction change throughout the cardiac cycle. To record the different projections of this vector, several electrodes are attached to the body in different locations known as leads. Because each lead measures the ECG between two points from different directions, amplitudes, polarities, times and durations of the ECG components vary between leads, so these have been standardized. The lead system most accepted in clinical practice is the standard 12-lead system, that is, the combination of the bipolar limb leads I, II and III, the augmented unipolar limb leads aVR, aVL and aVF and the six unipolar precordial leads V1–V6. Limb leads (I, II, III) derive signals from the left arm (LA), the right arm (RA) and the left leg (LL). The right leg (RL) electrode is the common reference in the amplifier [8].

In **Figure 1**, waves and intervals of interest of the ECG are shown. The P, QRS and T waves reflect the rhythmic electrical depolarization and repolarization of the myocardium associated with the contractions of the atria and ventricles. The P wave represents depolarization of the atrial musculature. The QRS complex is the combined result of the repolarization of the atria and the depolarization of the ventricles, which occur almost simultaneously. The T wave represents repolarization of the ventricles.

Time intervals like RR and QT are important in electrocardiographic diagnosis because they reflect electrophysiological processes of heart and autonomic nervous system (ANS) and carry clinical implications when they lie outside the range of the normal variation. The RR interval measured from the R wave peak to the peak of the next consecutive R wave is the interval between consecutive heart beats, and it determines the heart rate (HR). The QT interval measured from the Q wave onset to the T wave end reflects the total period of ventricular depolarization and repolarization, and it is used in clinical electrocardiology to quantify the duration of ventricular repolarization [9].

Prolongation of the QT interval is recognized as an indicator of an increased risk of malignant ventricular arrhythmias and/or sudden cardiac death in various clinical conditions such as myocardial infarction or ischemia, electrolyte or metabolic imbalance or the action of various drugs [9, 10]. Also, QT interval has an interlead space variability, which led to the hypothesis that the differences between electrocardiographic leads might reflect regional differences in repolarization. Based on experimental and clinical electrophysiological studies which supported the evidence that increased heterogeneity of repolarization may be responsible for

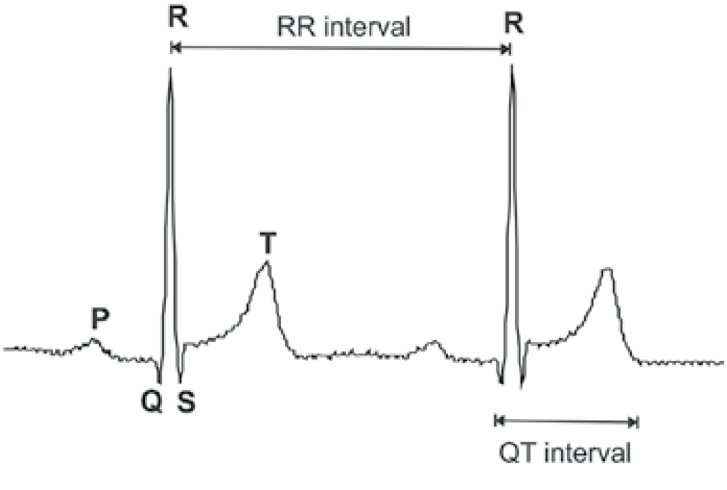


Figure 1. Waves and intervals of interest of the ECG.

generation of malignant ventricular arrhythmias, the interlead variation of QT interval duration was proposed as an index of arrhythmia susceptibility [10]. This measure was termed QT dispersion (QTd), and it was defined as the difference between the maximum and minimum QT interval on the standard 12-lead ECG [11].

Increased QTd has been associated with an increased risk for ventricular arrhythmias and sudden death in the general population and in various clinical conditions, among them, CKD is common. Several studies have reported that QTd increased in patients with ESRD, particularly after the end of HD [6, 12], so that it may be useful to identify patients at high risk for overall and cardiovascular mortality [13]. However, this index is affected by: an inaccurate measurement of the QT interval because of different definitions for the T wave end (with and without fusion with U or P waves), influence of HR, no simultaneous ECG leads recordings and number of ECG leads and of the ECG lead system used [14].

As beat-by-beat manual measurement of QTd on three orthogonal ECG leads is impractical in routine clinical practice, the development of accurate and robust methods for automatic detection of characteristic points of QRS and T waves is important in electrocardiographic diagnosis, in particular for the analysis of long recordings [15]. Wavelet transform is a suitable tool that has been used to determine peaks and limits of ECG waves because of its ability to detect transients and of its robustness in front of noise and artifacts [16–18]. This chapter presents the development of an algorithm based on the continuous wavelet transform (CWT) with splines for the automatic measurement of QTd in the quasi-orthogonal leads DI, aVF and V2, and its application for the analysis of QTd in patients with CKD.

2. Wavelet transform

Wavelet transforms at different scales describe the time characteristics of a signal in different frequency bands, but the analysis is restricted to scales that are powers of two [19]. The use of *B*-splines as base functions permits the evaluation of the CWT in any integer scale [20], which enables to use a wider range of scales and to reduce noise and artifacts more efficiently. This feature can allow the direct application of the algorithm over the raw ECG signal without any preprocessing stage because frequency filtering is performed when the CWT is computed.

The CWT of a time-continuous signal $x(t)$ is defined as:

$$\mathrm{CWT}\{x(t); a, b\} = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\, \psi^{*}\left(\frac{t-b}{a}\right) dt \tag{1}$$

where $\psi^{*}(t)$ is the complex conjugate of the analyzing wavelet function $\psi(t)$, and a and b are the scale and translation parameters, respectively. The function $\psi(t)$ compresses or dilates depending on a, which enables the CWT to extract the low- and high-frequency components of $x(t)$. To implement the CWT, a and b are usually discretized. If a is discretized over a sequence $2^{j}(j = 1, 2, \ldots)$, the analysis is restricted to scales that are powers of two, and the result is the dyadic wavelet transform that can be computed with Mallat's algorithm [19].

In this chapter, *B*-splines have been used which allow the evaluation of the CWT in any integer scale [20]. In this formulation, the input signal $x(t)$ and the analyzing wavelet $\psi(t)$ are both polynomial splines of degree n_1 and n_2, respectively. The splines considered are constructed from polynomial segments of degree n of unit length that are smoothly connected together at joining points called knots in such a way that guarantees the continuity of the function and its derivatives up to order (n−1) [21]

Assuming that the input signal $x(t)$ is characterized in terms of its *B*-spline expansion of degree n_1 and the sequence of *B*-spline coefficients $c(k)$

$$\mathrm{x(t)} = \sum\nolimits_{\mathrm{k}\in \mathrm{Z}} \mathrm{c(k)}\, \beta^{\mathrm{n}_1}(\mathrm{t}-\mathrm{k}) \tag{2}$$

Likewise, the wavelet $\psi(t)$ is a spline of degree n_2 with its *B*-spline expansion

$$\psi(t) = \sum\nolimits_{k\in Z} p(k)\, \beta^{n_2}(t-k) \tag{3}$$

B-splines satisfy a two-scale equation for any integer m, where m is not restricted to a power of two; thus, the wavelet expanded by a factor m can be expressed as:

$$\psi(t/m) = \sum\nolimits_{k\in Z}([p]_{\uparrow m} * u_m^{n_2})(k)\, \beta^{n_2}(t-k) \tag{4}$$

where the sequence $u_m^{n_2}(k)$, when n_2 and m are not both even, is given by z transform,

$$u_m^{n_2}(z) = \frac{z^{k_0}}{m^{n_2}} \left(\sum_{k=0}^{m-1} z^{-k}\right)^{n_2+1} \tag{5}$$

with

$$k_0 = (n_2+1)(m-1)/2 \tag{6}$$

Therefore, the resulting CWT at scale m evaluated at integer time samples is a polynomial spline function given by:

$$CWT\{x(t), m, k\} = \left([p]\uparrow_m * u_m^{n_2} * b^{n_1+n_2+1} * c\right)(k) \tag{7}$$

where the notation $([p]\uparrow_m * u_m^{n_2})$ represents the upsampling of the sequence p by a factor of m, the filter $u_m^{n_2}$ is equivalent to a cascade of (n_2+1) filters of moving average of order $(m-1)$ with an offset k_0 that ensures its symmetry, $b^{n_1+n_2+1}$ is the *B*-spline representation of a spline of order $n_1 + n_2 + 1$ and $c(k)'s$ are the *B*-spline coefficients.

The program $w = spwav(x, m, p, n_2, n_1)$ developed by Arregui (written in MATLAB®, The MathWorks Inc.) [22] calculates the CWT of the discrete signal $x(t)$ at the integer scale m of the cubic spline wavelet $(n_2 = 3)$ with expansion coefficients spline p, where $x(t)$ is considered a spline of order $n_1 = 1$. Implementation of the program *spwav* is based on the fast algorithm proposed by Unser et al. [20], which is done in the following three steps:

1. Initialization: calculus of the *B*-spline coefficients $c(k)$ that interpolate the signal $x(t)$ and the convolution with the *B*-spline of order n_2.
2. Iterated moving sum: calculus of the scalar products of the signal $x(t)$ with the *B*-splines of order n_2 dilated by a factor m and divided by the root of m.
3. Zero-padded filter: filtering with the expansion coefficients, spline of the basis wavelet p upsampling (with zeros) by a factor of m to obtain wavelet coefficients.

The selected wavelet function $\psi(t)$ is the first derivative of a fourth-order cubic *B*-spline expanded by two, which leads to the sequence $p = (-1, -4, -5, 0, 5, 4, 1)$ given in Table 1 of [20]. This wavelet is similar to the first derivative of a Gaussian function so that it yields good time and frequency resolution (**Figure 2**).

The Fourier transform of the wavelet at five scales (e = 1, 2, 3, 8 and 10) at a sampling frequency of 500 Hz is shown in **Figure 3**, and their −3 db bandwidths are listed in **Table 1**.

In **Table 2**, the −3 dB bandwidths of the Fourier transform of the wavelet at four scales for the sampling rates of 250–1000 Hz are listed, which correspond to three ECG databases used in this study. MIT-BIH Arrhythmia database (MITDB) [23], QT database (QTDB) [24] and CSE multilead measurement database (CSEDB) [25] used for the validation of the algorithm have sampling rates of 360, 250 and 500 Hz, respectively. The PTB Diagnostic ECG Database (PTBDB) [26, 27] and the E-HOL-12-0051-016 database of the Telemetric and Holter ECG Warehouse of the University of Rochester (THEWDB) [28] used for the application have a sampling rate of 1000 Hz.

Figure 4 shows the relation between the characteristic points of ECG and its CWT at four scales. Because of the form of the wavelet function selected, each distinct wave of the ECG corresponds to a pair of local maxima of the modulus (Pmm) of the CTW at each different scale with a zero crossing between them that corresponds to its peak. The rising slope of each wave

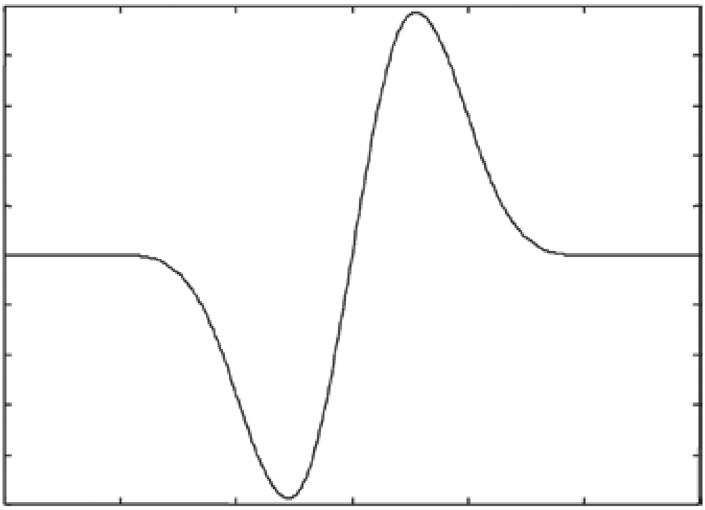


Figure 2. First derivate of a fourth-order B-spline expanded by a factor of two.

yields a minimum and the falling slope yields a maximum [16]. According to the spectrum of the ECG waves [29], most of the energy of the ECG signal lies within the scales 2–10 (**Figure 3**). P and T waves have their major component at scales 8 and 10, but higher scales can be affected

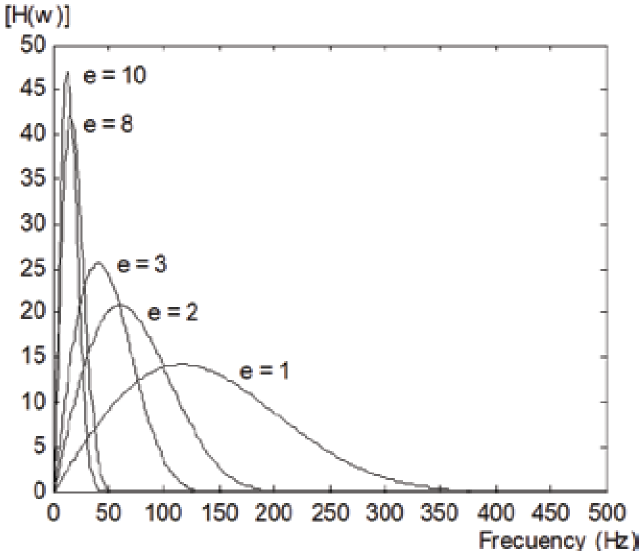


Figure 3. Amplitude-frequency responses of equivalent filters at five scales for 500 Hz sampling rate.

Scale (e)	– 3 dB bandwidth (Hz)
1	56–186
2	30–97
3	19–64
8	7–24
10	6–19

Table 1. Frequency response of equivalent filters at five scales for 500 Hz sampling rate.

Name	Sampling frequency			
	250 Hz		1000 Hz	
	Scale	**−3 dB bandwidth (Hz)**	**Scale**	**−3 dB bandwidth (Hz)**
w1	1	29–95	2	59–194
w2	2	16–49	5	25–79
w3	5	7–20	8	16–49
w4	10	4–11	20	7–20

Table 2. Frequency response of equivalent filters at four scales for sampling rates of 250 and 1000 Hz.

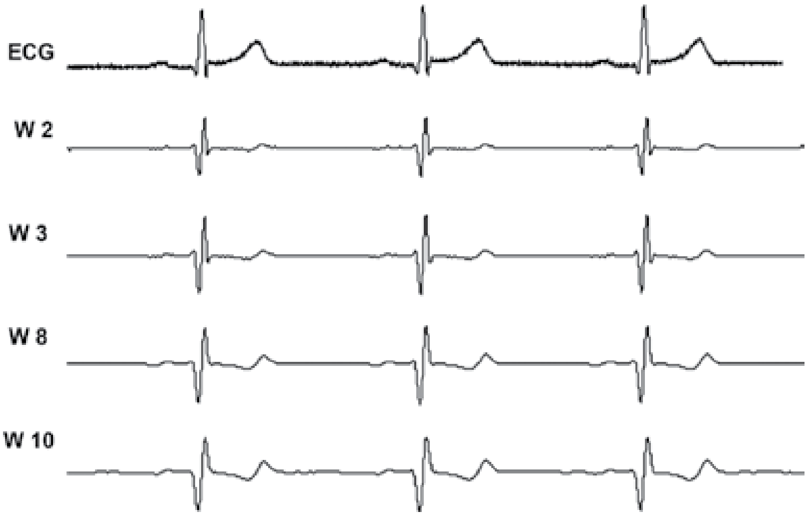


Figure 4. ECG and its CWT at scales 2, 3, 8 and 10.

by baseline wandering. If the ECG is contaminated with high-frequency noise, scales 2 and 3 are the most affected.

3. Description of the algorithm

The algorithm for automatic measurement of QTd in the quasi-orthogonal leads DI, aVF and V2 is based on the multilead generalization of a previous algorithm for single-lead detection of characteristic points of the QRS complex and T wave using the CWT with splines [18]. This new algorithm for multilead detection includes the identification of more types of morphologies of QRS complex and T waves [30], which are integrated with the previous algorithm for single-lead detection. **Figure 5** shows the algorithm proposed which is organized in four modules. In the first module, different kinds of QRS complexes and T-waves are detected and identified. In the second module, the algorithm detects the Q wave onset, R wave peak and T wave end, which is based on an algorithm for single-lead detection previously mentioned [18]. Next, the algorithm measures the QT and RR intervals from detections of significant points in each quasi-orthogonal lead. Finally, the algorithm calculates QTd as the difference in duration between the longest and shortest QT intervals measured on the three quasi-orthogonal leads and HR.

3.1. Detection of different kinds of QRS complex and T wave

As a first step in this stage, polarity of QRS complex and T wave is identified. QRS complex corresponds to a Pmm of the CTW and the scale used is w2, where it has its major component. Then, the highest positive peak (Wpq) and its nearest negative peak backwards (Wnq) are searched within the first 2 s of the record in order to define the position of these peaks. If Wnq position is before the Wpq position, then the type complex is qRs, which is defined as positive

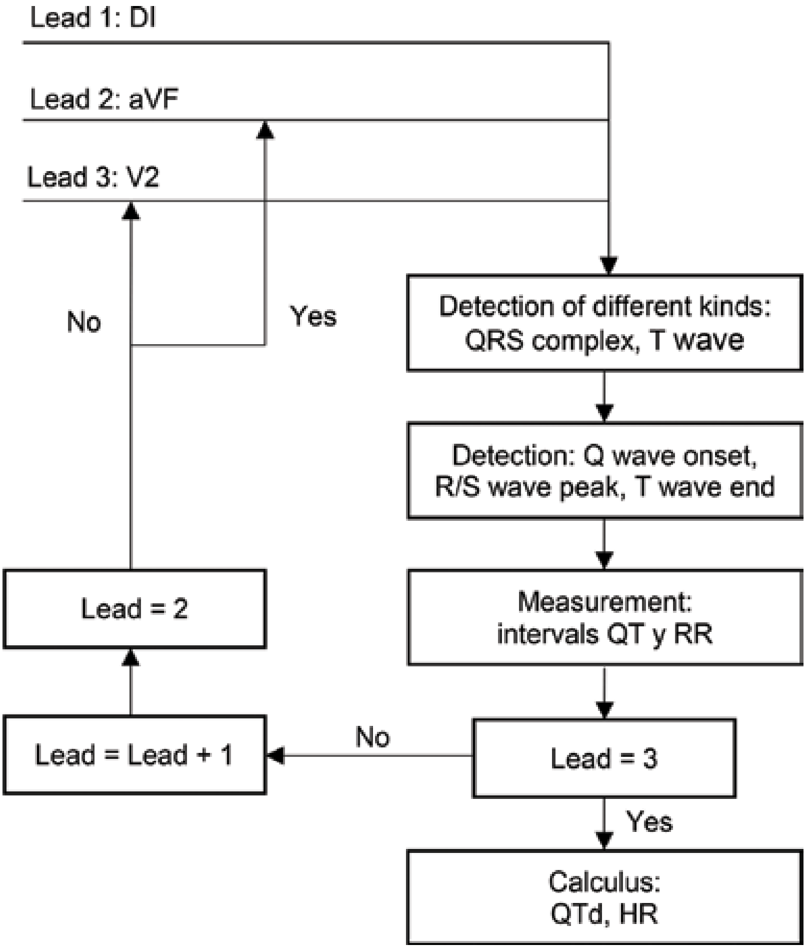

Lead 1: DI
Lead 2: aVF
Lead 3: V2
No
Yes
Detection of different kinds: QRS complex, T wave
Detection: Q wave onset, R/S wave peak, T wave end
Lead = 2
Measurement: intervals QT y RR
Lead = Lead + 1
No
Lead = 3
Yes
Calculus: QTd, HR

Figure 5. Flowchart of the algorithm for automatic measurement of QTd in the leads DI, aVF and V2.

QRS. If Wnq position is after the Wpq position, then the type complex is rS, which is defined as negative QRS (**Figure 6**). Flowchart of polarity detection of the QRS complex is shown in Figure 2 of [30].

To determine the type of QRS complex once its polarity is defined, two algorithms are applied depending if QRS complex is positive or negative. The algorithm to determine the type of QRS complex with positive polarity when R is higher, it defines if Q or S wave is present as follows. From the onset of the Pmm corresponding to the R wave at scale w2, Q wave is present if the nearest positive peak backwards is larger than a defined positive threshold. From the end of this Pmm, S wave is present if the nearest negative peak forward is lower than a defined negative threshold. These peaks are detected by looking inside a search window defined by the maximal duration of both waves. This algorithm detects and identifies the morphologies qR, qRs, R and Rs (**Figure 7**). Flowchart of the QRS complex type detection when R is higher is shown in Figure 5 of [30].

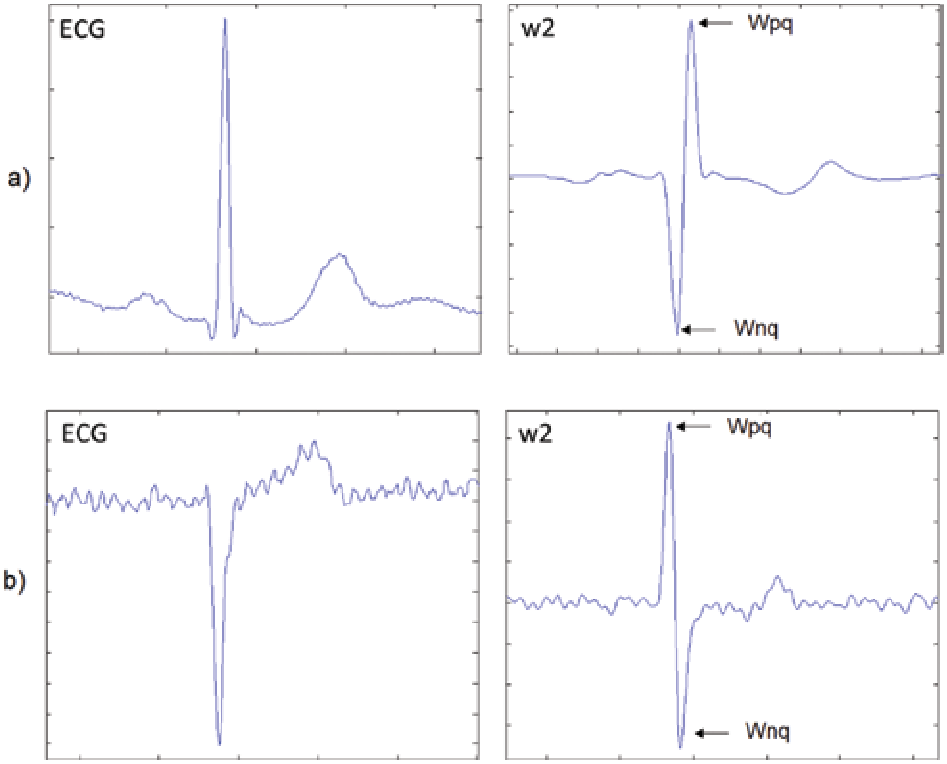


Figure 6. Polarity of QRS complexes and their CWT at scale w2. (a) Positive and (b) negative.

The algorithm to determine the QRS complex with negative polarity when S is higher, it defines if Q or R wave is present as follows. From the onset of the Pmm corresponding to the S wave at scale w2, R wave is present if the nearest negative peak backwards is lower than a defined negative threshold. From this point, Q wave is present if the nearest positive peak backwards is larger than a defined positive threshold. These peaks are detected by looking inside a search window defined by the maximal duration of both waves. This algorithm detects and identifies the morphologies qrS, rS and QS. **Figure 8** shows rS complex type and its CWT at scale w2. Flowchart of the QRS complex type detection when S is higher is shown in Figure 6 of [30].

Identification of polarity and type of T wave is performed with two algorithms. The first one classifies T wave into only two types: positive and negative (although it is biphasic, ascending or descending) as follows. As T wave corresponds to a Pmm of the CWT and only in this procedure, the scale used is w4 to enhance its characteristics. The highest positive peak (Wpt) and its nearest negative peak backwards (Wnt) larger than a defined threshold are searched from the end of the Pmm corresponding to R or S wave in a window whose limits depend on HR [31]. If Wnt position is before the Wpt position, then the T wave is positive or normal (**Figure 9a**). If Wnt position is after the Wpt position, then the T wave is inverted or negative. Flowchart of polarity detection of the T wave is shown in Figure 3 of [30].

The second algorithm to determine the type of T wave is applied after, once the R or S wave position is defined by the algorithm for single detection of characteristic points

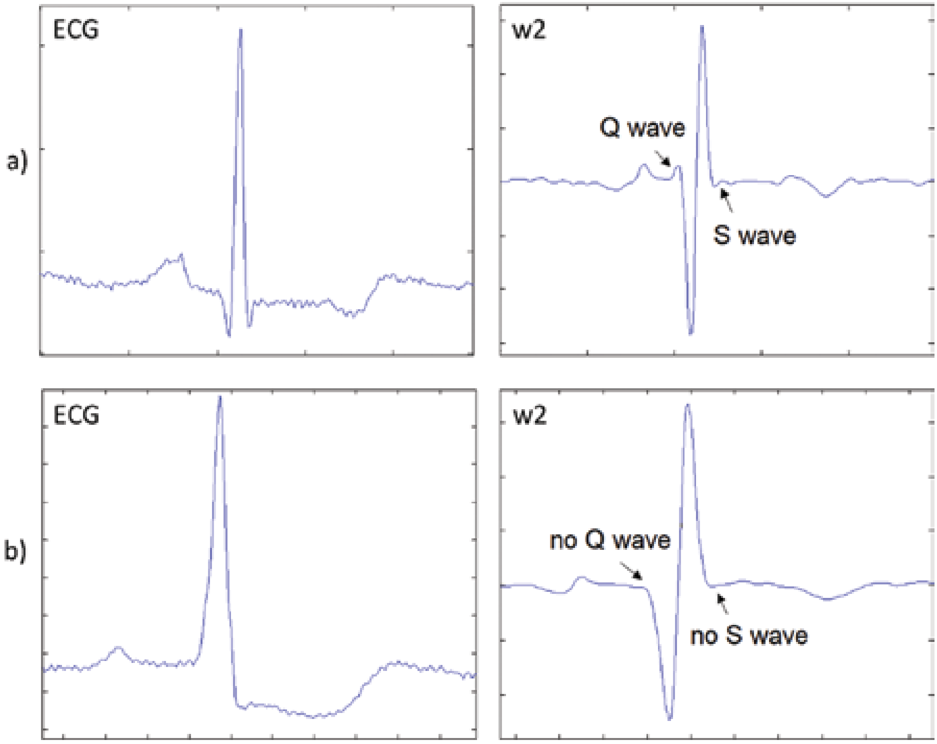


Figure 7. Complex types and their CWT at scale w2. (a). qRs and (b) R.

described later [18], and before the T wave detection. To identify T waves, the same procedure used for detecting of Wpt and Wnt of Pmm of T wave described earlier is used. According to the comparison of the absolute values of these peaks with defined thresholds and its position, the algorithm classifies five types of T waves: positive, negative, ascending, descending and biphasic (**Figure 9b**). Flowchart of the T wave type detection is shown in Figure 4 of [30].

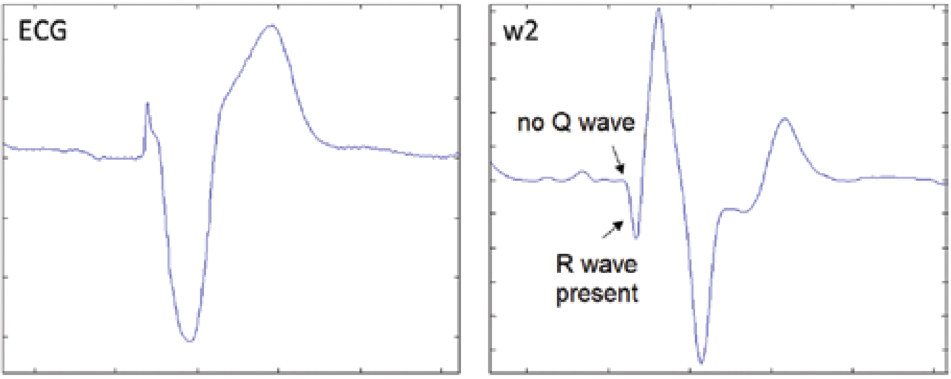


Figure 8. rS complex type and its CWT at scale w2.

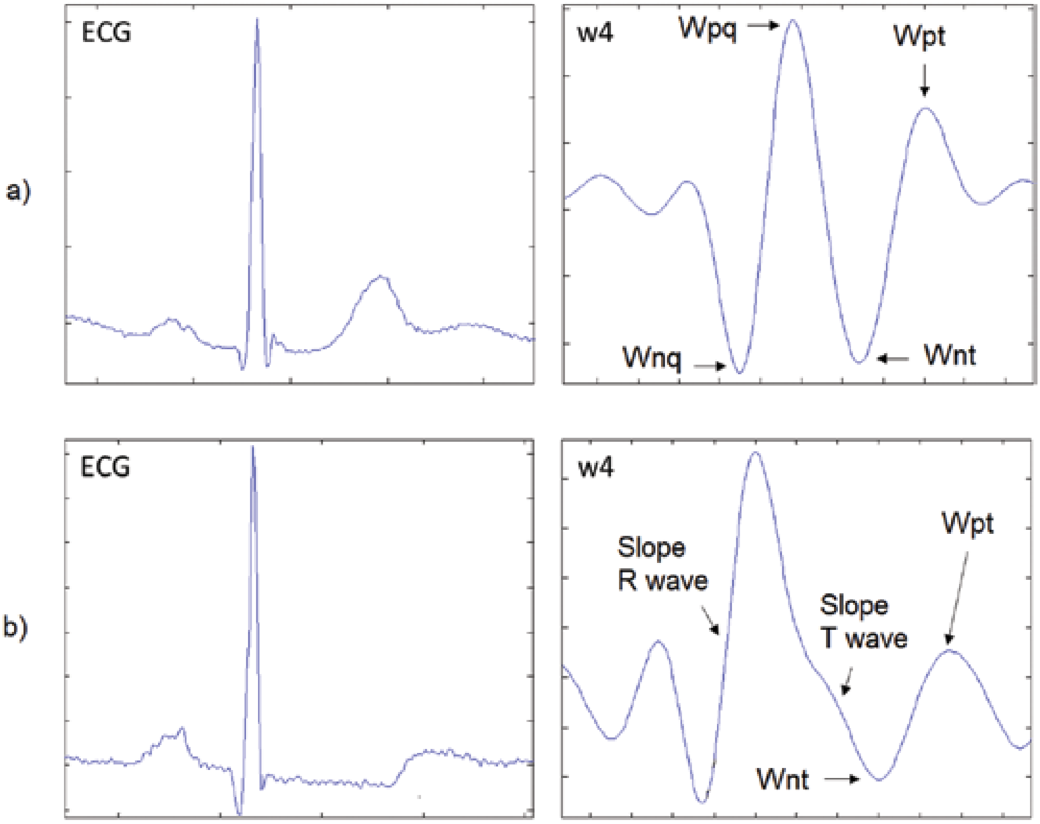


Figure 9. T wave types and their CWT at scale w4. (a). Positive and (b) biphasic.

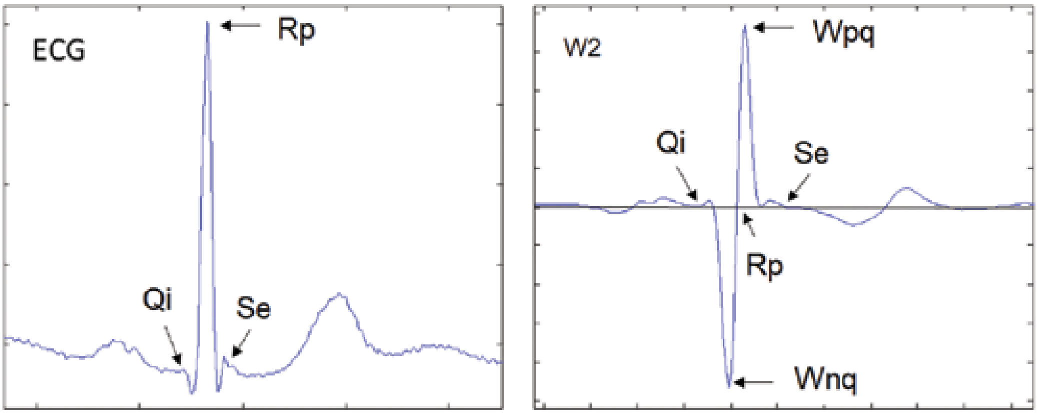


Figure 10. Onset, peak and end of the QRS complex and its CWT at scale w2.

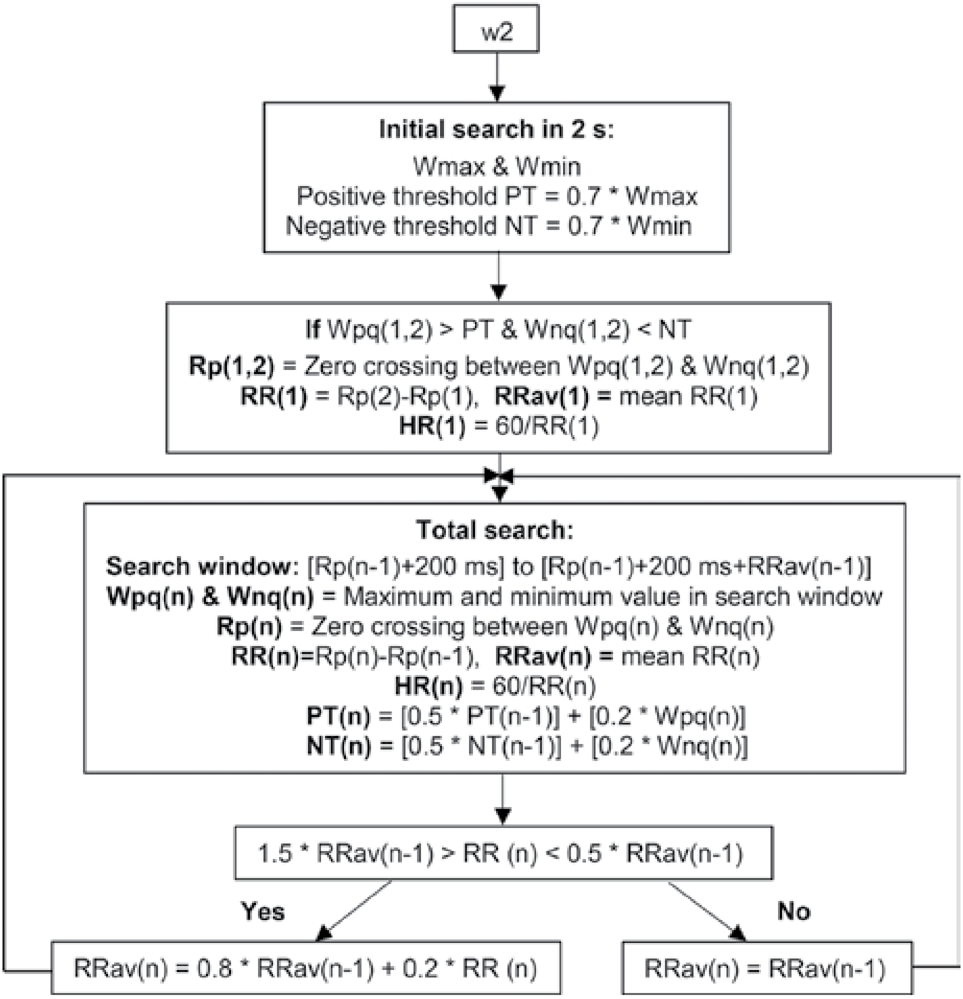

w2
Initial search in 2 s:
Wmax & Wmin
Positive threshold PT = 0.7 * Wmax
Negative threshold NT = 0.7 * Wmin
If Wpq(1,2) > PT & Wnq(1,2) < NT
Rp(1,2) = Zero crossing between Wpq(1,2) & Wnq(1,2)
RR(1) = Rp(2)-Rp(1), RRav(1) = mean RR(1)
HR(1) = 60/RR(1)
Total search:
Search window: [Rp(n-1)+200 ms] to [Rp(n-1)+200 ms+RRav(n-1)]
Wpq(n) & Wnq(n) = Maximum and minimum value in search window
Rp(n) = Zero crossing between Wpq(n) & Wnq(n)
RR(n)=Rp(n)-Rp(n-1), RRav(n) = mean RR(n)
HR(n) = 60/RR(n)
PT(n) = [0.5 * PT(n-1)] + [0.2 * Wpq(n)]
NT(n) = [0.5 * NT(n-1)] + [0.2 * Wnq(n)]
1.5 * RRav(n-1) > RR (n) < 0.5 * RRav(n-1)
Yes
No
RRav(n) = 0.8 * RRav(n-1) + 0.2 * RR (n)
RRav(n) = RRav(n-1)

Figure 11. Flowchart of the Rp detection algorithm.

3.2. Detection of characteristic points of QRS complex and T wave

3.2.1. QRS detection

QRS complex is the most characteristic waveform in the ECG due to its shape with high amplitude, which makes its detection easier than other ECG waves. Its accurate detection in the presence of noise and interferences is the most important task in the ECG automatic analysis because it is used as a reference in the cardiac cycle to perform a more detailed analysis of other ECG waves, segments and intervals, as automated measurement of HR and QT interval.

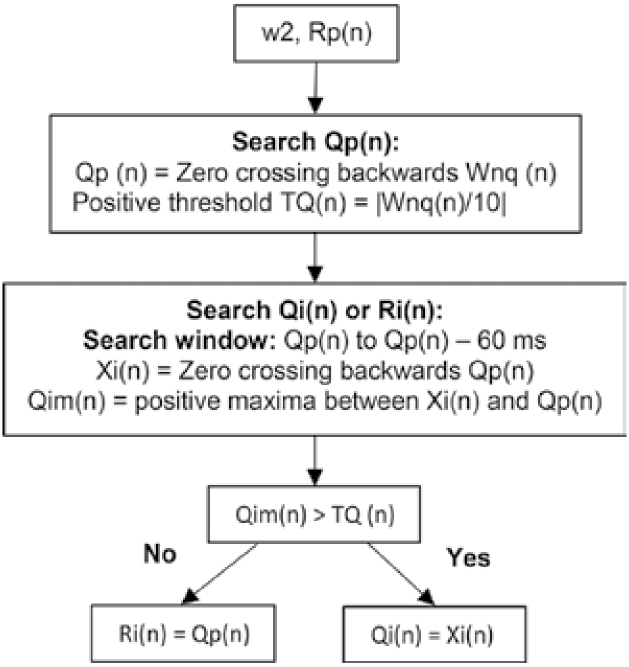

w2, Rp(n)
Search Qp(n):
Qp (n) = Zero crossing backwards Wnq (n)
Positive threshold TQ(n) = |Wnq(n)/10|
Search Qi(n) or Ri(n):
Search window: Qp(n) to Qp(n) – 60 ms
Xi(n) = Zero crossing backwards Qp(n)
Qim(n) = positive maxima between Xi(n) and Qp(n)
Qim(n) > TQ (n)
No
Yes
Ri(n) = Qp(n)
Qi(n) = Xi(n)

Figure 12. Flowchart of the Qi (Ri) detection algorithm.

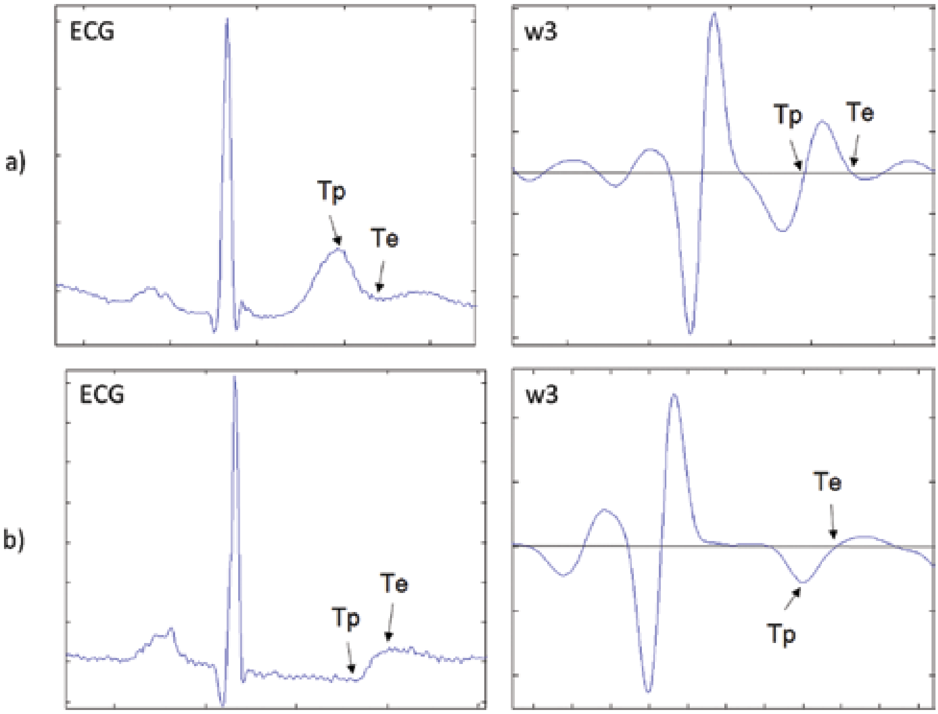


Figure 13. Peak and end of T waves and their CWT at scale w3. (a) Positive and (b) biphasic.

According to the wavelet function selected, QRS complex corresponds to a Pmm of the CWT at selected scale, where the R wave peak (Rp) corresponds to the zero crossing observed between the Pmm (**Figure 6**). The developed algorithm [18] detects the QRS by using the scale w2 and the Pmm corresponding to the R wave by defined threshold comparing inside a search window defined by the average RR interval and the last RR interval calculated [31]. From that Pmm, the start of the Q wave defined as Qi (or the start of R wave (Ri) in the absence of Q wave) corresponds to the zero crossing preceding the Pmm; the end of the S wave defined as Se (or the end of the R wave in the absence of the S wave) corresponds to the zero crossing after the Pmm (**Figure 10**). Those zero crossings are detected by looking inside a search window defined by the maximal duration of both waves. Flowcharts of the Rp and Qi (Ri) detection algorithms are shown in **Figures 11** and **12**, respectively.

3.2.2. T wave detection

Because of the low-frequency components of T wave [29], scale w3 of the CWT was used for its detection. The process for detection of positive and negative T waves is as follows: from the end of the Pmm of the Rp, we define a search window whose length decreases when RR diminishes [31]; inside that window, we look for the Pmm corresponding to the T wave that exceeds a defined threshold. The end of this Pmm and the zero crossing between them, corresponds to, respectively, the end (Te) and the peak (Tp) of the T wave. Detection and identification of ascending, descending and biphasic types depend on the number, polarity

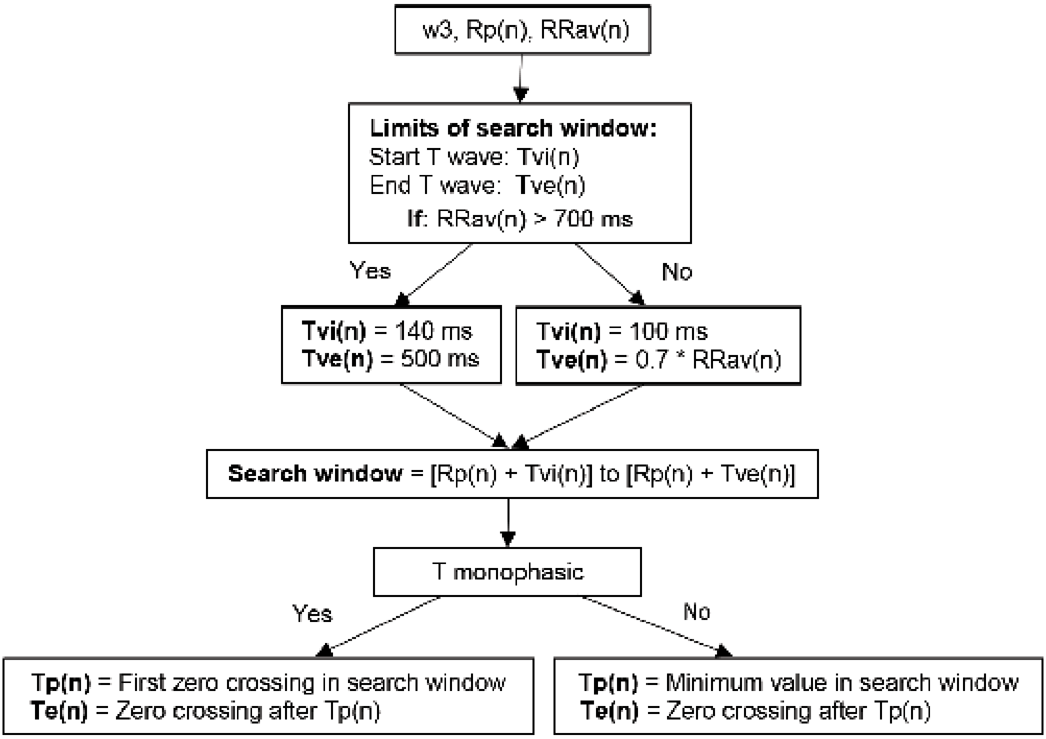

w3, Rp(n), RRav(n)
Limits of search window:
Start T wave: Tvi(n)
End T wave: Tve(n)
If: RRav(n) > 700 ms
Yes
No
Tvi(n) = 140 ms
Tve(n) = 500 ms
Tvi(n) = 100 ms
Tve(n) = 0.7 * RRav(n)
Search window = [Rp(n) + Tvi(n)] to [Rp(n) + Tve(n)]
T monophasic
Yes
No
Tp(n) = First zero crossing in search window
Te(n) = Zero crossing after Tp(n)
Tp(n) = Minimum value in search window
Te(n) = Zero crossing after Tp(n)

Figure 14. Flowchart of the Tp and Te detection algorithm of T wave monophasic or biphasic.

and absolute values of the found local maxima (Wpt) or minimum (Wnt). **Figure 13** shows peak and end of the positive and biphasic T wave and their characteristic points of CWT at scale w3. Flowchart of the Tp and Te detection algorithm of T wave monophasic or biphasic is shown in **Figure 14**.

3.3. Measurement of QT and RR intervals and calculus of QTd and HR

Once Qi and Te have been detected, the algorithm measures the QT and RR interval points in each quasi-orthogonal lead. Finally, the algorithm calculates QTd as the difference in duration between the longest and the shortest QT intervals measured in the three quasi-orthogonal leads, in which each QT interval is the average of three consecutive QT intervals. HR is calculated from the average of RR intervals measured in the same leads, in which each RR interval is the average of two consecutive RR intervals.

4. Validation and results

4.1. QRS detection

The developed algorithm for QRS detection [18] has been first tested on eight 30 min recordings resampled to 500 Hz from the MITDB [23], in which only channel 1 of the two-channel ECG recordings was used. The selected recordings included serious noise bursts, baseline drifts and movement artifacts. **Table 3** shows that QRS detector had 81 false QRS detections of 17,095 beats (0.47%); 51 of them were false positives and 30 were false negatives.

ECG record number	**Beats**	**FP**	**FN**	**False detections**	
				Beats	**%**
100	2272	0	1	1	0.04
101	1864	0	1	1	0.05
102	2187	0	0	0	0
103	2084	0	0	0	0
104	2229	17	4	21	0.9
105	2571	31	13	44	1.71
107	2135	0	1	1	0.04
108	1753	3	10	13	0.7
Total	**17,095**	**51**	**30**	**81**	**0.47**

FP, false positives; FN, false negatives.

Table 3. Validation results for the QRS detection algorithm applied to eight records from the MITDB.

4.2. Delineation of characteristic points of the QRS complex and T wave

The developed algorithm for delineation of Qi and Se of the QRS complex and Te of the T wave has been tested on 25 recordings from the CSE database [25], which includes 15 ECG leads and manual annotations on them. **Table 4** shows the average (m) and standard deviation (sd)

25 Recordings CSE	**Qi**	**Se**	**Te**
Mo1_001:121 (5:5)	**WT – CSE**	**WT – CSE**	**WT – CSE**
m ± sd	– 4.5 ± 1.5	7.6 ± 1.8	8.2 ± 3.6
Tolerance limits for deviations according to experts [31]			
sd (CSE)	6.5	11.6	30.6
Values are in ms; m, mean; sd, standard deviation.			

Table 4. Validation results for delineation algorithm of characteristic points Qi, Se and Te on 25 annotated recordings of the CSEDB.

QT	**Qi**	**Te**
database	**WT – C1**	**WT – C1**
sel100	15.4	−2.4
sel102	−5.3	22.8
sel103	11.9	20.5
sel104	4.1	−5.6
sel114	14.7	29.3
sel116	2.2	17.6
sel117	1.6	−13.8
sel123	2.9	−20.8
sel213	20.4	16.8
sel221	11.2	−16.3
sel223	−11.2	14.6
sel230	9.3	3.7
sel231	13.1	4.5
sel232	3.2	20.8
sel233	−5.8	13.2
m ± sd	**5.8(8)**	**7(15)**
Tolerance limits for deviations according to experts [31]		
sd (CSE)	6.5	30.6
Values are in ms; m, mean; sd, standard deviation.		

Table 5. Validation results of the delineation algorithm of characteristic points Qi and Te for 15 recordings from the QTDB in ms.

of the difference between the (WT-based) automatic and the (CSE) manual (annotated) location of those characteristic points. The results for that difference are within the tolerance limits accepted by the CSE experts, as shown in the last row of **Table 4** [32].

Also, the algorithm has been tested on 15 recordings from MITDB included in the QTDB [24]. Within each record of two channels, between 30 and 100 representative beats were manually annotated by cardiologists, who identified among other characteristic points of ECG waves, Qi of the QRS-complex and Te of the T-wave. Channel 1 was used in most recordings, in case of ECG distorted, channel 2 was used. **Table 5** shows the mean (m) and standard deviation (sd) of the differences between the manual measurements (C1) and automatic measurements (WT) of Qi and Te for each record. The results for the differences between WT and C1 are within the tolerances for deviations with respect to the measurements made by the CSE experts, as shown in the last row of **Table 5** [32].

Figure 15 shows some ECG excerpts of records with different T wave morphologies from QTDB with the manual annotations (square symbol) and the automatic detections (star symbol). It can be seen that Qi and Te are well determined by the algorithm, and its accuracy is comparable to a manual measurement of human experts.

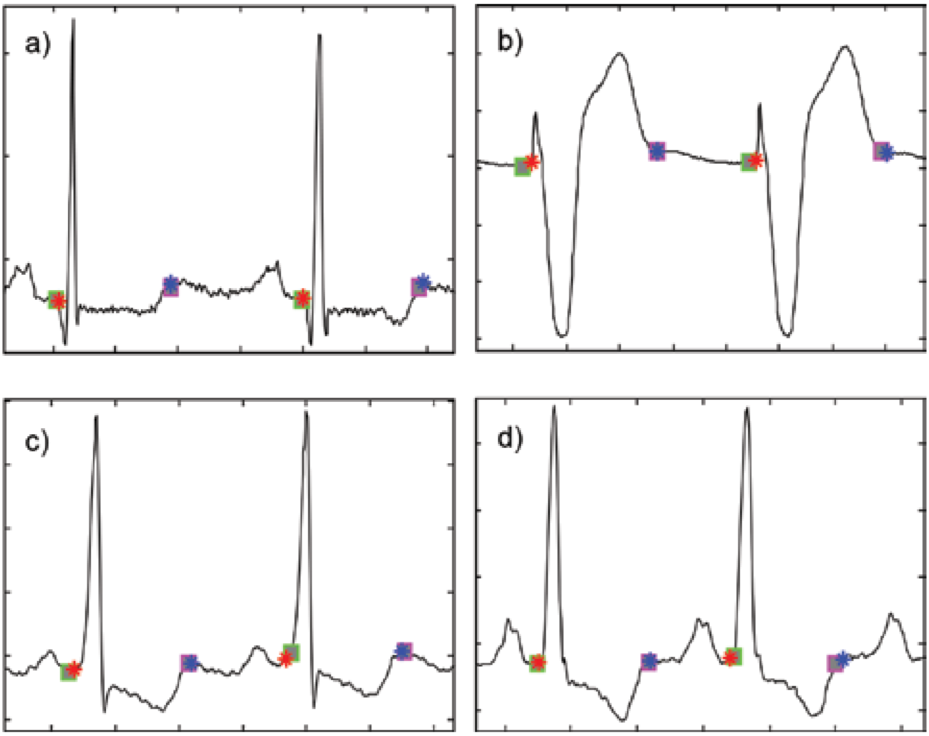


Figure 15. Automatic detections (star) and manual annotations (square) of Qi and Te with different types of morphologies of QRS complex and T wave in patterns of two beats of four records from QTDB. (a) rS positive T wave, (b) qRs biphasic T wave, (c) Rs biphasic T wave and (d) qR negative T wave.

5. Application in chronic kidney disease

The QTd algorithm was applied in two studies. In the first study, QTd was evaluated in normal subjects and patients with CKD. In the second study, QTd was analyzed in patients with CKD before, during and after the HD treatment.

5.1. QTd analysis in normal subjects and patients with CKD

In this study, 14 ECG recordings belonging to the PTBDB were used [26, 27], where the three quasi-orthogonal leads DI, aVF and V2 have been analyzed to obtain QTd. This database includes records of healthy people and patients with different pathologies. The study group was of seven normal subjects (two women and five men, age 66 ± 3.6 years) and seven renal insufficiency patients (three women and four men, age 70 ± 4.5 years). QTd corresponding to both groups was compared by the Wilcoxon rank sum test, where $p < 0.05$ was considered statistically significant.

Table 6 shows QTd and HR in both groups. Difference in HR in both groups is not significant and therefore HR influence is similar in both groups [(67.7 ± 9) beats/min vs. (70.8 ± 12) beats/min, $p = 0.53$]. QTd was significantly larger in patients with CKD than in normal subjects [(67.7 ± 28) ms vs. (21.4 ± 12), $p = 0.0041$]. The results obtained showed that the algorithm is effective to differentiate both groups.

5.2. QTd analysis in patients with CKD before, during and after hemodialysis

In this study, four ECG records of patients with CKD in the stage referred to as kidney failure or ESRD of the THEWDB [28], before (pre-HD), during and after (post-HD) HD session were used. For each patient, the three quasi-orthogonal leads DI, aVF and V2 have been analyzed to obtain QTd in a period of 10 h, in which pre-HD, HD and post-HD periods correspond to

Normal	QTd	HR	Patients	QTd	HR
patient121	2.6	84.9	patient012	54.6	49.94
patient122	17	63.6	patient013	116	86.12
patient239	40.6	69.0	patient078	70	73.20
patient248	17.6	63.9	patient079	74.6	62.37
patient255	27.6	67.2	patient140	21.6	86.41
patient266	25.6	72.6	patient145	65	68.72
patient267	18.6	52.8	patient216	72	69.44
m ± sd	**21.4(11)**	**67.7(9)**	**m ± sd**	**67.7(28)**	**70.8(12)**
			p	**0.0041**	**0.53**

QTd in ms; HR in beats/min; m, mean; sd, standard deviation; *p* value is from Wilcoxon rank sum test.

Table 6. QTd and HR in seven normal subjects and seven CKD patients.

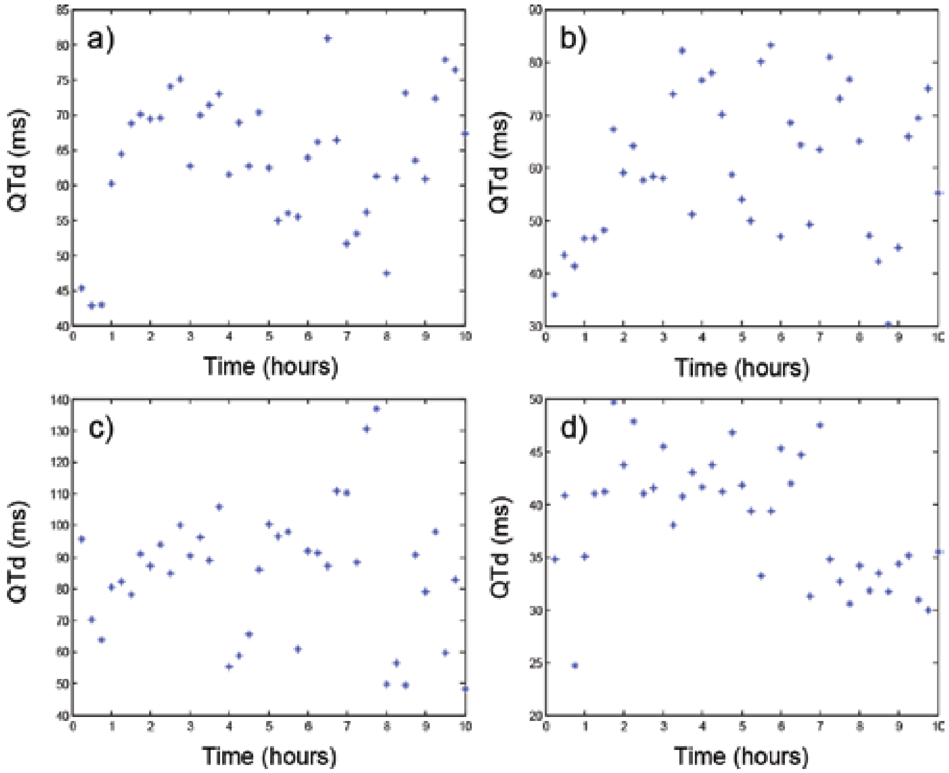


Figure 16. Dynamics of QTd of four CKD patients averaged every 15 min in pre-HD, HD and post-HD periods. (a) Patient 1013, (b) Patient 1030, (c) Patient 1050 and (d) Patient 1059.

the hours 1, 2–6 and 7–10, respectively. **Figure 16** shows the dynamics of QTd averaged every 15 min in a period of 10 h of the four CKD patients. It is observed that all the patients have an increase of QTd during HD and post-HD, which has been associated with malign ventricular arrhythmias and sudden death [6, 12].

6. Conclusion

This chapter presents and validates an algorithm based on the CWT with splines for the automatic measurement of QTd in the ECG quasi-orthogonal leads DI, aVF and V2. This algorithm permits the evaluation of the CWT in any integer scale which enables to use a wider range of scales and therefore to reduce noise and artifacts. In addition, the filters implemented in the algorithm based on *B*-splines are iterated discrete convolutions of moving sums, so that it can be computed without any multiplication, which results in a very efficient algorithm. Some functions of wavelet toolbox of MATLAB® related with this algorithm are as follows: the *spline* for cubic spline data interpolation, *cwt* that implements the CWT and *gauswavf* that returns the first order derivate of the Gaussian wavelet.

This new algorithm is based on the multilead generalization of a previous algorithm for single-lead detection of characteristic points of the QRS complex and T wave. It includes the identification of more types of morphologies of these waves, which are common in the analysis of several ECG leads and heart diseases. To evaluate its performance, ECG recordings of standard annotated databases MIT-BIH, QTDB and CSEDB were used. The results showed that the developed algorithm provides a reliable and accurate QRS detection and delineation of Qi and Te, with standard deviation of the errors within the tolerance limits for variations with respect to the measurements made by different experts.

The QTd algorithm was applied in two studies. In the first one, QTd was evaluated as a discriminator of patients with CKD from normal subjects. The results showed that QTd was significantly larger in CKD patients than in normal subjects, which agrees with similar studies. In the second study, QTd was analyzed in four patients with CKD before, during and after the HD treatment. The results showed that all the patients have an increase of QTd during HD and post-HD, which has been associated with malign ventricular arrhythmias and sudden death in previous studies.

Future applications of this algorithm will focus on to evaluate dispersion in other ECG ventricular activity intervals like JT (from S wave end to T wave end) and Tpe (from T wave peak to T wave end), in order to determine whether they improve the identification of CKD patients with risk of malign ventricular arrhythmias compared with QT dispersion.

Author details

María de Lourdes Corzo-Cuesta and Carlos Alvarado-Serrano*

*Address all correspondence to: calvarad@cinvestav.mx

Bioelectronics Section, Department of Electrical Engineering, Center for Research and Advanced Studies of the National Polytechnic Institute (Cinvestav), Mexico City, Mexico

References

[1] World Health Organization (WHO). Cardiovascular Diseases (CVDs) [Internet]. 2017-05. Available from: http://www.who.int/mediacentre/factsheets/fs317/en/ [Accessed: July 25, 2017]

[2] Plonsey R. Electrocardiography. In: Webster JG, editor. Electrocardiography. New York: John Wiley & Sons; 1988. pp. 1017-1040

[3] Instituto Nacional de Estadística y Geografía (INEGI). Defunciones generales por residencia habitual del fallecido y causa de la defunción (LISTA 1, CIE-10) según mes de ocurrencia, 2015 [Internet]. [Updated: 2016-11-30]. Available from: http://www.inegi.org.mx/est/contenidos/proyectos/registros/vitales/mortalidad/tabulados/def04.asp?t=13&c=11816 [Accessed: July 24, 2017]

[4] Kidney Disease: Improving Global Outcomes (KDIGO) CKD Work Group. KDIGO 2012 Clinical Practice Guideline for the evaluation and management of chronic kidney disease. Kidney International Supplements. 2013;**3**:1-150

[5] Sarnak MJ, Levey AS, Schoolwerth AC, Coresh J, Culleton B, Lee HL, et al. Kidney disease as a risk factor for development of cardiovascular disease: A statement from the American Heart Association councils on kidney in cardiovascular disease, high blood pressure research, clinical cardiology, and epidemiology and prevention. Circulation. 2003;**108**:2154-2169. DOI: 10.1161/01.CIR.0000095676.90936.80

[6] Drighil A, Madias JE, Benjelloun M, Kamoum H, Bennis A, Azzouzi L, et al. Changes in the QT intervals, QT dispersion, and amplitude of T waves after hemodialysis. Annals of Noninvasive Electrocardiology. 2007;**12**(2):137-144. DOI: 10.1111/j.1542-474X.2007.00152.x

[7] Bleyer AJ, Russell GB, Satko SG. Sudden and cardiac death rates in hemodialysis patients. Kidney International. 1999;**55**:1553-1559

[8] Márquez-Espinoza A, Mercado-Rojas JG, Vega-Martínez G, Alvarado-Serrano C. ECG ambulatory system for long term monitoring of heart rate dynamics. In: Silviu F, editor. Practical Applications and Solutions Using LabVIEW™ Software. Rijeka, Croatia: InTech; 2011. pp. 201-226. DOI: 10.5772/20946

[9] Zareba W, Moss AJ, Badilini F. Dispersion of repolarization: Noninvasive marker of nonuniform recovery of ventricular excitability. In: Moss AJ, Stern S, editors. Noninvasive Electrocardiology: Clinical Aspects of Holter Monitoring. London; Philadelphia: W.B. Saunders; 1996. pp. 405-419

[10] Kautzner J, Malik M. QT interval dispersion and its clinical utility. Pacing and Clinical Electrophysiology. 1997;**20**:2625-2640

[11] Day CP, McComb JM, Campbell RWF. QT Dispersion: An indication of arrhythmia risk in patients with long QT intervals. British Heart Journal. 1990;**63**:342-344

[12] Cupisti A, Galetta F, Morelli E, Tintory F, Sibilia G, Meola M, et al. Effect of hemodialysis on the dispersion of the QTc interval. Nephron. 1998;**78**:429-432

[13] Beaubien ER, Pylypchuk GB, Akhtar J, Biem HJ. Value of corrected QT interval dispersion in identifying patients initiating dialysis at increased risk of total and cardiovascular mortality. American Journal of Kidney Diseases. 2002;**39**:834-842

[14] Malik M, Batchvarov VN. Measurement, interpretation and clinical potential of QT dispersion. Journal of the American College of Cardiology. 2000;**36**:1749-1766

[15] Caminal P, Blasi A, Vallverdú M, Viñolas X, Jané R, Laguna P, et al. New algorithm for QT dispersion analysis in XYZ-lead Holter ECG. Performance and applications. In: Computers in Cardiology; 13-16 Sept. 1998; Cleveland, OH, USA. Piscataway, NJ, USA: IEEE; 1998. pp. 709-712. DOI: 10.1109/CIC.1998.731972

[16] Li C, Zheng C, Tai C. Detection of ECG characteristic points using wavelet transforms. IEEE Transactions on Biomedical Engineering. 1995;**42**:21-28

[17] Martínez JP, Almeida R, Olmos S, Rocha AP, Laguna P. A wavelet-based ECG delineator: Evaluation on standard databases. IEEE Transactions on Biomedical Engineering. 2004;**51**:570-581

[18] Alvarado C, Arregui J, Ramos J, Pallàs-Areny R. Automatic detection of ECG ventricular activity waves using continuous spline wavelet transform. In: 2nd International Conference on Electrical and Electronics Engineering (ICEEE) and XI Conference on Electrical Engineering (CIE 2005); 7-9 September 2005; Mexico City, Mexico. IEEE; 2005. pp. 189-192. DOI: 10.1109/ICEEE.2005.1529605

[19] Mallat S. A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1989;**11**:674-693

[20] Unser M, Aldroubi A, Schiff SJ. Fast implementation of the continuous wavelet transform with integer scales. IEEE Transactions on Signal Processing. 1994;**42**:3519-3523

[21] Unser M. Splines: A perfect fit for signal and image processing. IEEE Signal Processing Magazine. 1999;**16**(6):22-38. DOI: 10.1109/79.799930

[22] Arregui J. Study of Spectrotemporal Methods and their Application to the ECG. [Final Project]. Barcelona, Spain: Universitat Politècnica de Catalunya; 1996. (In Catalan)

[23] Moody GB, Mark RG. The MIT-BIH arrhytmia database on CD-ROM and software for use with it. In: Computers in Cardiology. Los Alamitos, CA: IEEE Computer Society Press; 1990. pp. 185-188

[24] Laguna P, Mark RG, Goldberg A, Moody GB. A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. In: Computers in Cardiology 1997; 24-27 September. Vol. 2000. Cambridge, MA, USA: IEEE; 2000. pp. 673-676. DOI: 10.1109/CIC.1997.648140

[25] Willems JL, Arnaud P, van Bemmel JH, Bourdillon PJ, Degani R, Denis B, et al. A reference database for multilead electrocardiographic computer measurement programs. Journal of the American College of Cardiology. 1987;**10**:1313-1321

[26] Bousseljot R, Kreiseler D, Schnabel A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik. 1995;**40**(1):S317

[27] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and Physionet: Components of a new research resource for complex physiologic signals. Circulation. 2000;**101**:e215-e220

[28] Telemetric and Holter ECG Warehouse (THEW). High-resolution 12-lead 48-hours continuous ECG from end-stage renal disease patients: during and after hemodialysis session. [Internet]. 2012. Available from: http://thew-project.org/Database/E-HOL-12-0051-016.html [Accessed: November 28, 2016]

[29] Thakor NV, Webster JG, Tompkins WJ. Estimation of QRS complex power spectra for design of a QRS filter. IEEE Transactions on Biomedical Engineering. 1984;**31**:702-706. DOI: 10.1109/TBME.1984.325393

[30] Corzo-Cuesta ML, Alvarado-Serrano C. An algorithm for QT dispersion analysis: Validation and application in chronic kidney disease. In: 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE); 26-30 September 2016; Mexico City. Mexico: IEEE; 2016. pp. 1-6. DOI: 10.1109/ICEEE.2016.7751237

[31] Laguna P, Thakor NV, Caminal P, Jané R, Yoon HR, Bayés de Luna A, et al. New algorithm for QT interval analysis in 24-hour Holter ECG: Performance and applications. Medical & Biological Engineering & Computing. 1990;**28**:67-73

[32] The CSE Working Party. Recommendations for measurement standards in quantitative electrocardiography. European Heart Journal. 1985;**6**:815-825

Chapter 3

Multivariate Adaptive Regression Splines in Standard Cell Characterization for Nanometer Technology in Semiconductor

Taizhi Liu

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74854

Abstract

Multivariate adaptive regression splines (MARSP) is a nonparametric regression method. It is an adaptive procedure which does not have any predetermined regression model. With that said, the model structure of MARSP is constructed dynamically and adaptively according to the information derived from the data. Because of its ability to capture essential nonlinearities and interactions, MARSP is considered as a great fit for high-dimension problems. This chapter gives an application of MARSP in semiconductor field, more specifically, in standard cell characterization. The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that accurately and efficiently capture cell behaviors. In this chapter, the MARSP method is employed to characterize the gate delay as a function of many parameters including process-voltage-temperature parameters. Due to its ability of capturing essential nonlinearities and interactions, MARSP method helps to achieve significant accuracy improvement.

Keywords: multivariate adaptive regression splines (MARSP), semiconductor microelectronics, standard cell characterization, very large scale integration (VLSI), variation-aware gate-delay modeling, statistical timing analysis

1. Introduction

Multivariate adaptive regression splines (MARSP) was first proposed by Friedman [1] for solving regression-type problems. MARSP is widely used to predict the values of an outcome variable from a set of predictor variables. There are many methods for model fitting, and MARSP is one of them. Other modeling techniques include linear regression (e.g., general linear model), nonlinear regression (e.g., generalized linear/nonlinear models), and regression trees (e.g., classification and regression trees), and so on. It is also worth noting that a neural network, which is very popular nowadays in an era of artificial intelligence and machine learning, is also a modeling technique.

MARSP is a nonparametric regression procedure that makes no assumptions about the underlying functional relationships between dependent and independent variables. The form of MARSP and its coefficients are entirely derived from the regression data. The modeling strategy is called "divide and conquer," by which the input space is partitioned into a number of regions, with each region having its own regression equation. This makes MARSP particularly efficient for high-dimension problems, where other techniques most likely have accuracy issues.

As the name suggests, MARSP uses splines as its main component. Splines are piecewise curves from polynomial functions. When different splines are smoothly connected, it can result in a flexible model which can handle both linear and nonlinear situations. The connection points between different pieces are called knots, which connect the end of one region of data and the beginning of another.

The MARSP technique has been particularly popular in data mining because it does not require or assume any particular type or any class of relationship (e.g., logistic, linear, etc.) between the outcome variable of interest and the predictor variables. Instead, MARSP derives useful models (i.e., models that yield accurate predictions) even in situations where the relationship between the predictor variable and the predictor variables is difficult to approximate with parametric models. If you are interested in more information about MARSP and how it compares to other methods for nonlinear regression (or regression trees), please refer to Chapter 9 of [2].

2. Standard cell characterization in very large scale integration (VLSI) design

In semiconductor design, standard cell methodology is a method that is widely used for very large scale integration (VLSI) design, especially for digital logic circuits. It is a design abstraction, where the low-level circuit layout can be encapsulated into many abstract logic representations (e.g., NOR2, NAND2 cells). As a cell-based methodology, it can enable one designer to focus on the high-level aspect (logical function) of a design, while another designer can work on the implementation aspect (physical layout). As semiconductor fabrication technology progressed to sub-10 nm regime, standard cell methodology was the enabler to allow designers to scale application-specific integrated circuits (ASICs) from simple chips of several thousand cells, to complex chips with hundreds of millions of cells.

A standard cell provides a Boolean logic function (e.g., AND, OR) or a storage function (latch or flip-flop). A standard cell can be as simple as an inverter which consists of only two transistors. It can also be as complex as adders or multiplexers which have tens of transistors. As a standard cell is a logic gate, "cell" and "gate" are often interchangeable. Standard cell library is a collection of predefined cells which are usually fully customized to a specific technology and optimized for best delay, power, area, and so on. The standard cell library provides a way for designers to place cells in rows, and it enables the automated layout generations for digital ASICs.

The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that could accurately and efficiently model cell behavior. Cell behavior may come from a variety of parameters in different aspects like capacitance, power, timing, current, waveform, and so on. Among them, one of the most important models in standard cell characterization is timing delay models. In this chapter, for simplicity, we focus on the characterization of the pin-to-pin propagation delay of standard cells. Propagation delay is the time required for the input to be propagated to the output. In other words, it is defined as the time it takes for the effect change in input to be visible at the output. Propagation delay is important because it has a direct effect on the speed at which a digital device, such as a computer, can operate. This is true of memory chips as well as microprocessors. As mentioned earlier that a gate and a cell are often interchangeable in this chapter, cell delay and gate delay are interchangeable as well.

Cell propagation delays change with many factors, including the following:

1. The transition time of the input causing transition at the output.
2. The output load being felt by the logic cell/gate.
3. The process parameters (threshold voltages, channel lengths) of the transistors that the cell is consisted of.
4. The power supply voltage (VDD).
5. The temperature (Although temperature is not a factor with significant impact, it is still an impacting factor).

Among the different factors above, the process parameters are included because of the emergingly pronounced effect called process variations, which is introduced in details in Section 2.1.1. In the later sections, we use process-voltage-temperature (PVT) parameters to denote process parameters, VDD, and temperature.

2.1. Introduction (problem formation)

As mentioned above, one of the most important tasks in standard cell characterization is to find a model which can accurately capture the relationship between the cell propagation delay and the parameters that have impact on cell delay (as shown in the paragraph above). Here, the cell propagation delay is the response variable, and the impacting parameters (input transition time, output loads, VDD, and the process parameters) are the explanatory parameters.

We have not talked about the number of explanatory parameters yet. But as mentioned in Section 1, MARSP is suitable for the high-dimension problem while capturing essential nonlinearities and interactions. In the following subsections, we introduce the high-dimension parameter space when characterizing the delay models of standard cells, especially when the process variations and aging effect are included [3–7].

2.2. Parameter space

2.2.1. Process variations

When integrated circuits are fabricated, the parameters of individual transistors vary. The observed random distribution of identically drawn devices is caused by the fabrication process like impurity concentration densities, oxide thicknesses, and diffusion depths, and so on. These physical variations cause changes in the electrical characteristics of the transistors which eventually lead to the variability in the circuit performance. This is called process variation. Process variation is the naturally occurring variation in the attributes of transistors (length, widths, oxide thickness) during the chip fabrication. The scaling down of the VLSI process technologies has increased the process variations, especially in sub-45 nm era.

Process variations can be generally categorized into two classes: inter-die and intra-die variations. Inter-die variations occur from one die (chip) to another, meaning that the same transistor in the design can get different features (channel lengths, threshold voltages, etc.) among different dies (chips). Intra-die variations are variations in transistor features within a chip, meaning that transistors at different locations on the same die can get different features. Spatial correlations are often seen for intra-die variations, meaning adjacent transistors have a higher probability of having similar features than transistors that are far apart. In this work, we consider not only inter-die and intra-die variations, but also the intra-gate variations. Intra-gate variations are part of intra-die variations, in some sense. It is the variations within a gate (cell), meaning that the transistors within the same gate can have different features. While most of the literature works ignored the intra-gate variations, our work has included it. As VLSI technology continues to scale down to sub-10 nm process, intra-die variations (including intra-gate variations) are becoming more and more dominant.

The overall objective of standard cell characterization is to characterize a cell-delay model which is general and able to include inter-die, intra-die, and intra-gate variations with any kind of distribution and any correlation profile between different parameters. In this work, only process variations of standard cells are considered, meaning the variations in interconnect geometries are not considered.

2.2.2. Loading effect modeling (pi-model)

As technology scales down, the impact of interconnect on circuit timing cannot be neglected. In this work, we model interconnect as a resistive-capacitive (RC) network where all the capacitances are grounded.

A small patch of a gate-level circuit is illustrated in **Figure 1(a)**, where a driving Buffer gate has two loading gates, a NOR2 gate and an inverter gate. **Figure 1(b)** replaces the two loading gates with corresponding input capacitances. The input capacitances of loading cells, together with the interconnect network, form the load of the previous driving cell. With loading gates modeled as corresponding input capacitances, circuit timing can be analyzed in the way that each stage contains a standard cell and its connecting load as **Figure 1(b)** shows. If the readers are interested in the input-capacitance modeling of the standard cells, they can refer to [8, 9] for more details.

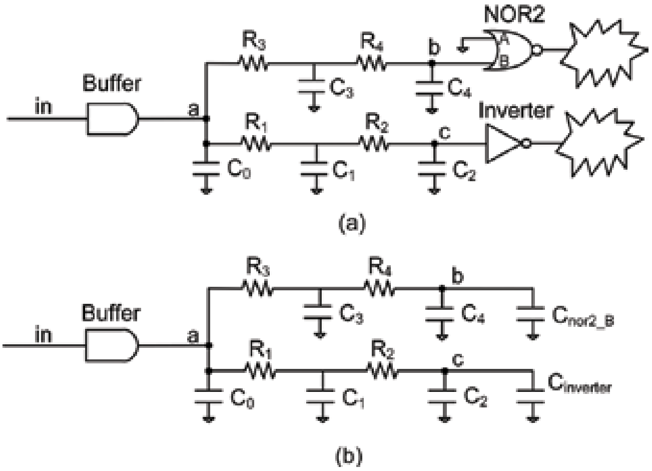


Figure 1. Input capacitances of the loading standard cells and the interconnect network formed the load of the previous driving gate. (a) A driving buffer and two loading cells (NOR2 and Inverter); (b) The driving buffer, with the input capacitances of the two loading cells incorporated into its load.

Reduced-order models are routinely used to replace the original large-order models. The Pi-model is the most popular reduced-order model to estimate the input admittance of RC interconnects. **Figure 2** gives the structure of the Pi-model, where *Y(s)* denotes the input admittance of the original network and *Y′(s)* denotes the input admittance of the Pi-model. The values of C_1, R, and C_2 are obtained by equating the first, second, and third moments of the Pi-model to corresponding moments of the original network.

In Pi-model, we use three parameters to represent the loading effect of the whole RC interconnect. These three parameters C_1, R, and C_2, as well as the PVT parameters and input transition time, construct the parameter space for standard cell characterization which is introduced later in Section 3.

The shift in channel length (from the nominal value) is denoted as ΔL, and threshold voltage shift (from the nominal value) is denoted as ΔVth. The supply voltage and temperature of a gate are denoted as ΔVDD and ΔT, respectively, assuming that all the transistors within the same gate share the same voltage and temperature. The Pi-model which represents the load of a gate includes three parameters, namely R_{pi}, C_{pi1}, C_{pi2}. The input slew time (*Slope*) is also included for each timing arc. Note that in this work the effect of Multiple Input Switching (MIS) was not considered.

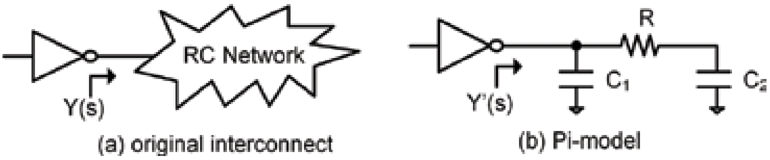


Figure 2. Y′(s) in Pi-model as an approximation of original input admittance function Y(s). (a) original interconnect (b) Pi-model.

For a cell which has N transistors, there are 2*N device parameters (i.e., ΔL, ΔVth for each transistor within the cell), and six global parameters (ΔVDD, ΔT, R_{pi}, C_{pi1}, C_{pi2}, *Slope*). This results in a total of (2*N + 6) parameters for a cell with N transistors. In our experiments with a commercial standard cell library, the highest value of N is 32, making the highest *(2*N + 6)* as 70, which results in a quite high-dimension parameter space for cell characterization.

At this point, we have not introduced the aging effect into the parameter space. If the characterized delay models need to be aging-aware, the aging parameters should be included in the parameter space. With aging parameters included, the dimension of the parameter space would be even higher. We discuss it in the following subsection.

2.2.3. High-dimensional parameter space in aging-aware standard cell characterization

For timing analysis, transistor aging is another source of variability besides PVT variations [10, 11]. Our work has considered the following wear-out mechanisms: bias temperature instability (BTI), hot carrier injection (HCI), and time-dependent dielectric breakdown (TDDB). The impact of BTI and HCI is similar as they both cause the threshold voltage of aged transistors to increase, which further decreases the driving strength and ultimately increases gate delay over time. TDDB degrades the drain current of the stressed devices which also results in increased gate delay. Overall, BTI, HCI, and TDDB ultimately cause the cell delay to increase over time. When the increased circuit delay exceeds the clock period, the degraded circuit will fail to work. Therefore, the aging effect needs to be taken into account in circuit timing simulations, especially for those high-reliability applications like aviation, space, automotive [12], medical [13–17], data center [18], and so on.

The variation of channel length and the variation of threshold voltage are denoted as ΔL and ΔVth, respectively. For channel length, the variation (ΔL) comes from only process variation, while for transistor threshold voltage, the variation (ΔVth) comes from both process variation and aging effect (BTI and HCI).

$$\Delta Vth \ = \ \Delta Vth_{process} + \Delta Vth_{BTI} + \Delta Vth_{HCI} \tag{1}$$

For the effect of TDDB, we need to include two additional parameters for each transistor within a gate, namely, R_{G2S} (gate-to-source resistance), and R_{G2D} (gate-to-drain resistance), because the gate-oxide breakdown paths can happen from gate-to-drain or from gate-to-source [19–22]. In summary, each transistor contributes four parameters (ΔVth, ΔL, R_{G2D}, R_{G2S}) as input parameters. Thus, for a cell with N transistors, there are 4*N parameters for ΔL, ΔVth, R_{G2S} and R_{G2D}, plus six global parameters (ΔVDD, ΔT, R_{pi}, C_{pi1}, C_{pi2}, *Slope*), resulting in a total of (4*N + 6) parameters for each cell.

As the value of N is as high as 32 in our experiments with a commercial library, the value of *(4*N + 6)* can be as high as 134. Compared to 70, which is the value of *(2*N + 6)* for cell characterization without aging effect, the dimension of parameter space in the aging-aware cell characterization has nearly doubled.

Var.	Corners	Var.	Corners
ΔL_p	[−30%, 30%]	ΔL_n	[−30%, 30%]
ΔVth_p	[−30%, 30%]	ΔVth_n	[−30%, 30%]
ΔVdd	[−10%, 10%]	ΔT (°C)	[−50,50]
Slope	[10 ps, 3 ns]	C_{pi2} (pF)	[1100]
R_{pi} (Ohm)	[1, 1000]	C_{pi1} (pF)	[0.1,10]

Table 1. Variations and corners of considered parameters.

2.2.4. Training data

We have obtained our training data from simulation program with integrated circuit emphasis (SPICE) simulations. A mixture of central composite design and random samples are used for the design of experiments. **Table 1** shows the corners which are used for central composite design.

2.3. Why is multivariate adaptive regression splines (MARSP) better

Why is MARSP better than other methods in our application of standard cell characterization? Traditional methods like response surface methodology (RSM) use the same model to cover the entire parameter space. In our application where intra-gate variability is considered, the dimension of the parameter space is particularly high. When the number of input parameters is high, the parameter space is very high dimension. Using one single regression model to estimate gate delay (or slew time) over the whole parameter space is not sufficiently accurate, especially for a complex cell containing over 40 transistors. References [23, 24] proposed a clustering method which categorized transistors into switching/non-switching devices and on-transition/off-transition/non-transition devices. This method requires manual intervention to 'filter out' the negligible devices for each of the switching scenarios, which is quite cumbersome. Using MARSP, it can reduce the manual work and automatically capture the essential parameters in its intelligent process.

3. MARSP for standard cell characterization

This chapter employs MARSP to characterize a fitted function between response variables (gate delay or slew time) and the explanatory parameters (process-voltage-temperature parameters, aging parameters, and RC loads). MARSP uses piecewise polynomial segments to capture essential nonlinearities and interactions, and it is particularly suitable for high-dimension problems. This piecewise nature allows MARSP models to split the whole parameter space into multiple subspaces, and each subspace can have a unique regression model. By using hinge functions, MARSP then inherently integrates the regression models of all the subspaces into a single general form.

A hinge function has the form of $\overset{+}{(x-t)}$ or $\overset{+}{(t-x)}$ which are shown in **Figure 3**. They are defined as:

$$(x-t)_{+} = \begin{cases} x-t, ifx > t, \\ 0, otherwise, \end{cases} \tag{2}$$

$$(t-x)_{+} = \begin{cases} t-x, ifx < t, \\ 0, otherwise, \end{cases} \tag{3}$$

where t is a constant called the knot. MARSP forms a collection of hinge-function pairs for each explanatory parameter X_j with knots at $x_{j1}, x_{j2}, \ldots, x_{jM}$, where M is the number of experiments.

MARSP models have the following form:

$$f(\vec{X}) = \beta_0 + \textstyle\sum_{t=1}^{T} \beta_t h_t(\vec{X}) \tag{4}$$

where $h_t(\vec{X})$ is a basis function. There are two phases in the process of constructing a MARSP model: the forward stepwise addition and the backward stepwise deletion.

The first phase is the forward stepwise addition, where MARSP starts with a model consisting of an intercept term. It then repeatedly adds basis functions in pairs to the model step by step. At each step, MARSP finds the pair of basis functions which maximized the reduction in the residue sum-of-squares error. The two basis functions in the pair are identical except that the hinge functions used for each basis function are mirrored. The newly added basis function is constructed by a term that is already in the model (a constant 1 is also considered as an existing term) multiplied by a new hinge function. The process of forward addition phase continues until the residual error difference in two adjacent steps is smaller than a predefined threshold or until the number of terms in the model reaches the maximum.

The model from the forward addition phase usually overfits the data. If a model overfits, it means the model fits well to the training data that are used to build the model, but usually it does not fit to new test data. The second phase of MARSP, namely the backward stepwise deletion, is to build models that can generalize better to new data. Backward deletion phase

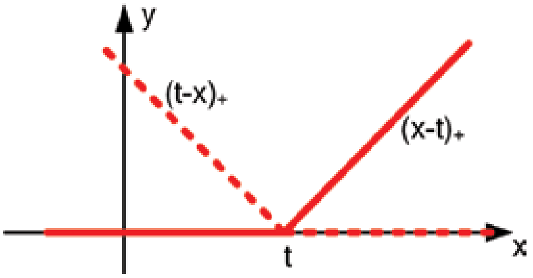


Figure 3. The solid line denotes the form of the hinge function (x − t) + while the dashed line denotes the hinge function (t − x) + .

prunes the model obtained from the previous forward addition phase. In this phase, the technique called generalized cross validation (GCV) is used to trade off goodness-of-fit against model complexity. The stepwise backward deletion phase repeatedly deletes the least important term (according GCV) at each step until the model again has only the intercept term left. At the end of the backward deletion phase, from among the "best" models of each size at each step, the model with the lowest GCV value is selected, and it is outputted as the final model.

MARSP is a nonparametric regression method, so there are no predetermined forms of the model. Instead, the model is constructed adaptively according to the information extracted from the training data. It intelligently removes those negligible parameters that have limited impact on the to-be-modeled gate delays or output slew without manual intervention. Using MARSP for cell characterization can eliminate the need of clustering transistors into the categories of switching/non-switching devices and on-transition/off-transition/non-transition devices, as proposed in [23, 24].

The MARSP model is piecewise in nature, so MARSP can split the whole parameter space (which is high-dimension in our application) into multiple subspaces, with each subspace getting its own model. Then the regression models of all the subspaces are integrated into one general expression using piecewise hinge functions. In this way, MARSP can characterize standard cells only once over the whole PVT space, without the need of splitting parameter space and characterizing every subspace.

Figure 4 shows an example where the values of $h(X_1, X_2)$ change only when X_1 is large and X_2 is low. The nonlinear relationship can be easily handled by the following MARSP model:

$$h(X_1, X_2) = 15 + 0.015 * (20 - x)_+ * (y - 20)_+ \tag{5}$$

Figure 5 shows the quadratic regression model which has poor accuracy in this example.

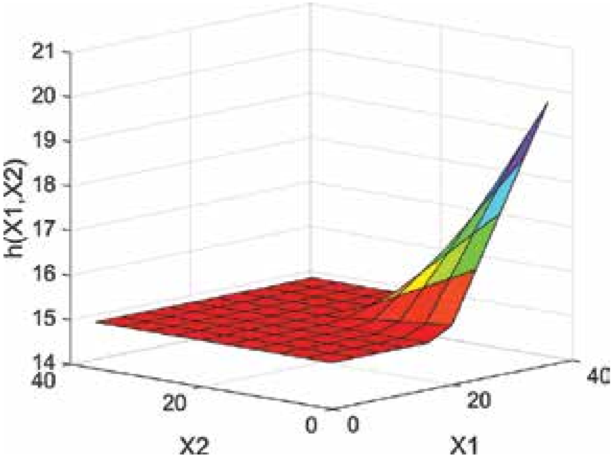


Figure 4. A nonlinear function $h(X\,1, X\,2)$ which changes only when X1 is high and X2 is low. The MARSP model in (4) perfectly matches the original relationship.

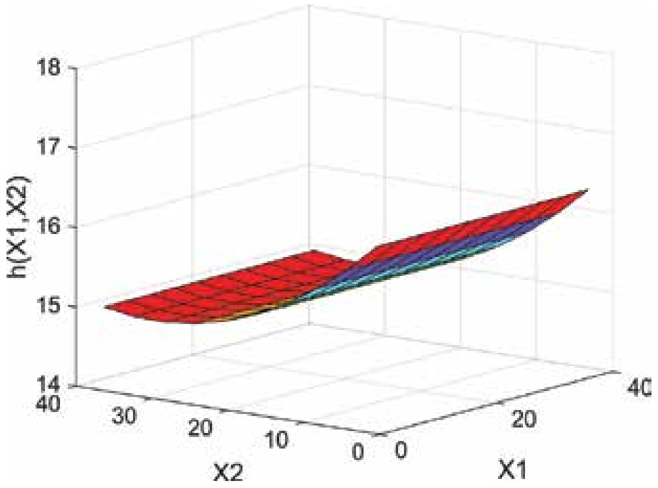


Figure 5. A quadratic model is regressed from $h(X\,1, X\,2)$ and it has poor accuracy compared to MARSP model.

4. Experimental results

The characterization variables are delay and transition time. Cell delay is the delay from the 50%-point at the cell input to the 50%-point at the cell output. Cell transition time is also called the output slew time, and it is the time between the 20%-point and the 80%-point at cell output (20–80% for rising transition and 80–20% for falling transition). The goal is to find a model that best fits the relationship between the gate delay (output slew) and the explanatory parameters.

In our work, MARSP is implemented using a Matlab toolbox called ARESLab [25]. Some key settings for ARESLab are as follows: the maximum degree of interactions between explanatory parameters is 3; the maximum number of basis functions is 30; the threshold for the stopping criteria is set to 10^{-4}. Please note that all the training data have been normalized.

A commercial library consisting of 247 standard cells was used, and every timing arc for every cell was characterized. The characterization results for some representative cells are shown in **Table 2**. The "*4*N + 6*" column in the table means the number of parameters in the MARSP model, and the "Time(s)" column means characterization time. The "Error" column ("Mean" and "S.D.") means the average value and standard deviation of the errors between MARSP and golden reference (SPICE), respectively.

The interconnect characterization is similar to the gate although there are only five considered parameters in our work. The details of reduced-order model of interconnect transfer function is not covered in this chapter (Please refer to [8, 9] for more details). The interconnect results are also shown in the last row of **Table 2**. Interconnect variability (spacing, width) is not included in our experiments. It is also worth noting that our methodology can support a higher-order *H′(s)*-model which matches more moments of the original *H(s)* at the expense of adding more parameters to the MARSP models.

Cell	(4*N + 6)	Time(s)	Error	
			Mean (%)	S.D. (%)
INVX1	14	861	0.41	2.54
NOR2X1	22	1281	0.23	2.67
CLKBUF3	134	6577	0.17	2.86
XOR2X1	54	3112	0.20	2.32
Interconnect	5	22	0.01	0.12

Table 2. MARSP characterization results on representative standard cells.

4.1. Validation using test paths

Our framework was implemented with C++ and Perl, and the experiments were run on a Linux platform with a 2.27 GHz CPU and 1GB memory without using multi-threading.

Our experiments are based on ISCAS85 benchmark circuits where temperature and supply voltage are considered as global parameters, meaning that all the transistors across the circuit have the same values of temperature and voltage. However, it is worth noting that our methodology can support a temperature profile from a thermal simulator and a voltage profile from an IR-drop simulator. For process variation, as mentioned earlier, we have considered inter-die, intra-die, and intra-gate variations. For channel lengths, we have considered inter-die and intra-die variation, and for threshold voltage, intra-gate variation is considered. This is because channel length is mostly impacted by lithography and etching which exhibit strong spatial correlations, while threshold voltage is strongly affected by random dopant fluctuations. Again, please note that our methodology can work with any inter- and intra-die variation model and with any distributions and any correlation profiles.

We have shown our MARSP models are perfectly accurate individually. Here we construct a framework to integrate our models and then verify its accuracy using test paths. We refer to our framework as GTSSTA hereafter. Two thousand Monte Carlo samples were run for 10 randomly selected test paths from ISCAS85 benchmark. As shown in the framework above, path delay is calculated for each sample. This obtained delay value is compared to the delay value from hSpice [26], using Eq. (6).

$$Error_{each_{\_sample}} = \frac{DELAY_{GTSSTA} - DELAY_{SPICE}}{DELAY_{SPICE}} \times 100\% \tag{6}$$

A quadratic delay model was also implemented and tested to give a comparison. The quadratic first generates a quadratic regression model as follows:

$$D = d_0 + \sum a_i X_i + \sum b_i X_i^2 + \sum_{i \neq k} b_{i,k} X_i X_k. \tag{7}$$

D denotes gate delay, X_i denotes the explanatory parameters, d_0 denotes the constant term, and a_i and b_i denote coefficients of first-order and second-order terms, respectively.

Table 3 presents the results for our framework in comparison to hSpice using these 10 test paths. **Figure 6** gives the histogram comparison of one of the paths between hSpice and GTSSTA. Results in **Table 3** also show that quadratic model has limited accuracy for the 10 test paths.

4.2. Runtime analysis

Experimental results show our framework consumes only ~2% more runtime than quadratic delay model but achieves much better accuracy.

The quadratic delay model in Eq. (7) has a fixed number of operations, that is, 120 multiplications and 66 additions for a one-input gate and 224 multiplications and 120 additions for a two-input gate. The number of operations using MARSP models is not fixed, and it depends on which subspace the data sample falls into. Basically, calculating a MARSP model will have comparisons first and based on the comparison results, different equations (linear, quadratic etc.) are used for calculations. In average, the number of operations for the MARSP model is close to that of the quadratic delay model.

Path	**Circuit name**	**Primary input to primary output**	**Num. of stages**	**Num. of samples**	**Running Time (s)**			**Path-delay error per sample (OurSSTA)**		**Path-delay error per sample (quadratic)**	
					SPICE	**Our**	**Quad.**	**Mean (%)**	**SD (%)**	**Mean (%)**	**SD (%)**
1	c432	N102 to N421	60	2000	4211	198	192	3.21	1.21	−10.99	24.9
2	c499	N85 to N724	43	2000	2417	140	138	0.25	0.54	−13.95	26.6
3	c880	N1 to N878	57	2000	4148	189	185	−0.31	0.53	−14.10	25.4
4	c1355	G11 to G1352	42	2000	2412	143	144	3.54	1.49	−10.68	26.7
5	c1908	N19 to N2890	72	2000	5878	258	250	4.12	1.92	−10.76	26.9
6	c2670	N227 to N3851	54	2000	4001	184	180	2.90	1.16	−15.81	25.1
7	c3540	N33 to N5360	77	2000	6719	276	270	2.75	1.10	−5.94	27.5
8	c5315	N335 to N8128	71	2000	6454	255	251	2.53	1.55	−11.72	26.3
9	c6288	N290 to N6287	221	2000	19,898	704	698	4.29	1.50	−14.43	23.1
10	c7552	N18 to N11334	116	2000	9948	380	378	3.87	1.48	−10.23	25.8
avg.	—	—	81	2000	6608	273	268	2.72	1.25	−11.86	25.8

Table 3. Experimental results on 10 test paths for MARSP models and quadratic models (errors compared to golden SPICE results).

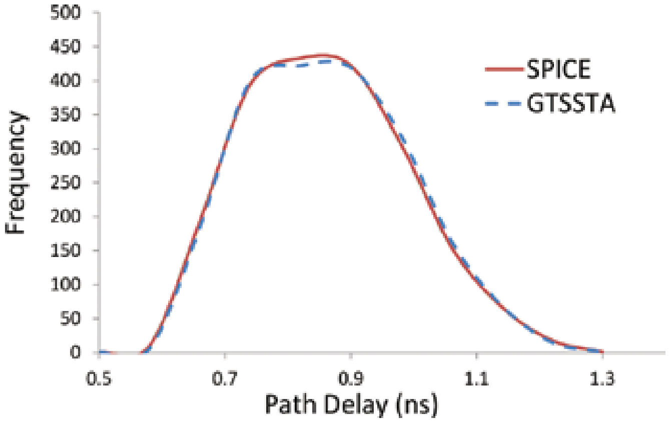


Figure 6. Monte Carlo histogram comparison between GTSSTA and SPICE for test path (N85 to N724) in circuit c499 (2000 samples are run).

5. Conclusion

This chapter talks about the technique called multivariate adaptive regression splines (MARSP). MARSP is a nonparametric regression without taking any pre-assumed form. Instead, it adaptively constructs the model according to the provided data. MARSP has been widely used in high-dimension problems and particularly popular in data mining.

This chapter also gives an application of MARSP in semiconductor field, more specifically, in standard cell characterization. The objective of standard cell characterization is to create a set of high-quality models of a standard cell library that accurately and efficiently model cell behavior. In this work, the MARSP method is employed to characterize the gate delay as a function of many parameters including process-voltage-temperature parameters. Due to its ability of capturing essential nonlinearities and interactions, MARSP method helps to achieve significant accuracy improvement.

Some future work that is worth investigating includes extending the aging-aware MARSP-based timing analyzer to 3D integrated circuits (IC) to study the reliability of 3D ICs which tend to have reliability challenges due to the stronger heat issues. 3D ICs requires more sophisticated thermal models [27–29] and more complicated power-grid analysis [30]. As mentioned earlier, the methodology in this chapter is general to support other thermal and IR-drop models

Author details

Taizhi Liu

Address all correspondence to: taizhiliu88@gatech.edu

Georgia Institute of Technology, Georgia, United States of America

References

[1] Friedman JH. Multivariate adaptive regression splines. Annals of Statistics. 1991;**19**(1):1-67

[2] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. 2nd ed. Springer; 2009

[3] Chen C-C, Cha S, Liu T, Milor L. System-level modeling of microprocessor reliability degradation due to BTI and HCI. In: Proceedings International Reliability Physics Symposium (IRPS); 2014. pp. CA 8.1-CA 8.9. DOI: 10.1109/IRPS.2014.6861125

[4] Chen C-C, Liu T, Milor L. System-level modeling of microprocessor reliability degradation due to bias temperature instability and hot carrier injection. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2016;**24**(8):2712-2725

[5] Cao Y et al. Cross-layer modeling and simulation of circuit reliability. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2014;**33**(1):8-23

[6] Fang J, Sapatnekar SS. Incorporating hot carrier injection effects into timing analysis for large circuits. IEEE Transactions on Very Large Scale Integration Systems (TVLSI). 2014;**22**(12):2738-2751

[7] Liu T, Chen C-C, Kim W, Milor L. Comprehensive reliability and aging analysis on SRAMs within microprocessor systems. Microelectronics Reliability. 2015;**55**(9-10):1290-1296

[8] Liu T, Chen C-C, Milor L. Accurate standard cell characterization and statistical timing analysis using multivariate adaptive regression splines. In: Proceeding of International Symposium on Quality Electronic Design (ISQED); 2015. pp. 272-279

[9] Liu T, Chen CC, Milor L. Comprehensive reliability-aware statistical timing analysis using a unified gate-delay model for microprocessors. IEEE Transactions on Emerging Topics in Computing. DOI: 10.1109/TETC.2016.2588724

[10] Liu T, Chen C-C, Wu J, Milor L. SRAM stability analysis for different cache configurations due to Bias Temperature Instability and Hot Carrier Injection. In: Proceedings of IEEE 34th International Conference on Computer Design (ICCD); 2016. pp. 225-232

[11] Cha S, Chen C-C, Liu T, Milor L. Extraction of threshold voltage degradation modeling due to negative bias temperature instability in circuits with I/O measurements. In: Proceedings of VLSI Test Symposium (VTS); 2014. pp. 1-6. DOI: 10.1109/VTS.2014.6818769

[12] Jiang C, Habetler T. Static eccentricity fault detection of the BLDC motor inside the air handler unit (AHU). In: 2015 IEEE International Electric Machines & Drives Conference (IEMDC); 2015. pp. 1473-1476

[13] Cai W, Li C, Luan S. SOI RF switch for wireless medical sensor network. Advances in Engineering: An International Journal (ADEIJ). 2016;**1**(2):1-9

[14] Cai W, Huang L, Wen WJ. 2.4 GHZ class AB power amplifier for wireless medical sensor network. International Journal of Enhanced Research In Science Technology & Engineering (IJERSTE). 2016;**5**:94-98

[15] Cai W, Xu J, Huang L. Low power SI class E power amplifier and RF switch for healthcare. An International Journal Informatics Engineering. 2016;**4**(4):7-13

[16] Cai W, Shi F. Design of low Power Medical Device. International Journal of VLSI design & Communication Systems (VLSICS). 2017;**8**(2):1-7

[17] Cai W, Shi F. Design of low power Si based DPA for medical device. International Journal of Electronics Communication and Computer Engineering. 2017;**8**(2):134-136

[18] Zhang S et al. Syslog processing for switch failure diagnosis and prediction in datacenter networks. In: 2017 IEEE/ACM 25th international symposium on quality of service (IWQoS); 2017. DOI: 10.1109/IWQoS.2017.7969130

[19] Chen C-C, Liu T, Cha S, Milor L. Processor-level reliability simulator for time-dependent gate dielectric breakdown. Microprocessors and Microsystems. 2015;**39**(8):950-960

[20] Yang K, Liu T, Zhang R, Kim D-H, Milor L. Front-end of line and middle-of-line time-dependent dielectric breakdown reliability simulator for logic circuits. Microelectronics Reliability. 2017;**76**:81-86

[21] Liu T, Chen C-C, Cha S, Milor L. System-level variation-aware aging simulator using a unified novel gate-delay model for bias temperature instability, hot carrier injection, and gate oxide breakdown. Microelectronics Reliability. 2015;**55**(9-10):1334-1340

[22] Cha S, Kim D-H, Liu T, Milor L. The die-to-die calibrated combined model of negative bias temperature instability and gate oxide breakdown from device to system. Microelectronics Reliability. 2015;**55**(9-10):1404-1411

[23] Zhang W, Singhee A, Xiong J, Habitz P, Joshi A, Visweswariah C, Sundquist J. A dynamic method for efficient random mismatch characterization of standard cells. In: Proceedings ICCAD; 2012. pp. 180-186

[24] Sundareswaran S, Abraham J, Panda R, Ardelea A. Characterization of standard cells for intra-cell mismatch variations. IEEE Transactions on Semiconductor Manufacturing. 2009;**22**(1):40-49

[25] Jekabsons G. ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave [Internet]. [Updated: 2011]. Available from: http://www.cs.rtu.lv/jekabsons/

[26] Synopsys. HSPICE User Guide [Internet]. Available from: https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html [Accessed: 2017]

[27] Zhang Y, Sarvey TE, Bakir MS. Thermal challenges for heterogeneous 3D ICs and opportunities for air gap thermal isolation. In: Proceedings IEEE 3D Systems Integration Conference (3DIC); 2014. DOI: 10.1109/3DIC.2014.7152174

[28] Zhang Y, Sarvey TE, Bakir MS. Thermal evaluation of 2.5-D integration using bridge-chip technology: Challenges and opportunities. IEEE Transactions on Components, Packaging and Manufacturing Technology. 2017;**7**(7):1101-1110. DOI: 10.1109/TCPMT.2017.2710042

[29] Zhang Y, Bakir MS. Integrated thermal and power delivery network co-simulation framework for single-die and multi-die assemblies. IEEE Transactions on Components, Packaging and Manufacturing Technology. 2017;**7**(3):434-443

[30] Zhang F, Wu B, Zhai J, He B, Chen W. Finepar: Irregularity-aware fine-grained workload partitioning on integrated architectures. In: IEEE/ACM Intl. Symp. on Code Generation and Optimization (CGO); 2017. pp. 27-38

Chapter 4

Smoothing Spline ANOVA Models and their Applications in Complex and Massive Datasets

Jingyi Zhang, Honghe Jin, Ye Wang, Xiaoxiao Sun,
Ping Ma and Wenxuan Zhong

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.75861

Abstract

Complex and massive datasets can be easily accessed using the newly developed data acquisition technology. In spite of the fact that the smoothing spline ANOVA models have proven to be useful in a variety of fields, these datasets impose the challenges on the applications of the models. In this chapter, we present a selected review of the smoothing spline ANOVA models and highlight some challenges and opportunities in massive datasets. We review two approaches to significantly reduce the computational costs of fitting the model. One real case study is used to illustrate the performance of the reviewed methods.

Keywords: smoothing spline, smoothing spline ANOVA models, reproducing kernel Hilbert space, penalized likelihood, basis sampling

1. Introduction

Among the nonparametric models, smoothing splines have been widely used in many real applications. There has been a rich body of literature in smoothing splines such as the additive smoothing spline [1–6], the interaction smoothing spline [7–10], and smoothing spline ANOVA (SSANOVA) models [11–14].

In this chapter, we focus on studying the SSANOVA models. Suppose that the data (y_i, x_i) and $i = 1, 2, \ldots, n$ are independent and identically distributed (i.i.d.) copies of (Y, X), where $Y \in \mathcal{Y} \subset \mathbb{R}$ is the response variable and $X \in \mathcal{X} \subset \mathbb{R}^d$ is the covariate variable. We consider the regression model:

$$y_i = \eta(x_i) + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where y_i is the response, η is the nonparametric function varying in an infinite-dimensional space, $x_i = \left(x_{i\langle 1\rangle}, \ldots, x_{i\langle d\rangle}\right)^T$ is on the domain $\mathcal{X} \subset \mathbb{R}^d$, and $\epsilon_i \overset{\text{i.i.d.}}{\sim} N\left(0, \sigma^2\right)$. More general cases, in which the conditional distribution of Y given x, denoted as $Y|x$, which follows different distributions instead of the Gaussian distribution, will be discussed later. The nonparametric function η in (1) can be decomposed into

$$\eta(x) = \eta_c + \sum_{j=1}^{d} \eta_j\left(x_{\langle j\rangle}\right) + \sum_{k<j} \eta_{kj}\left(x_{\langle k\rangle}, x_{\langle j\rangle}\right) + \ldots$$

through the functional ANOVA, where η_c is a constant function, η_j is the main effect function of $x_{\langle j\rangle}$, η_{kj} is the interaction effect of $x_{\langle k\rangle}$ and $x_{\langle j\rangle}$, and so on.

In the model (1), η can be estimated by minimizing the following penalized likelihood functional:

$$L(\eta) + \lambda J(\eta), \tag{2}$$

where $L(\eta)$ is a log likelihood measuring the goodness of fit of η, $J(\eta)$ is a quadratic functional on η to quantify its smoothness, and λ is the smoothing parameter balancing trade-offs between the goodness of fit and the smoothness of η [11–13]. The computational complexity of estimating η by minimizing (2) is of the order $O\left(n^3\right)$ for the sample of size n. Therefore, the high computational costs render SSANOVA models impractical for massive datasets. In this chapter, we review two methods to lower the computational costs. One approach is through the adaptive basis selection algorithm [14]. By carefully sampling a smaller set of basic functions conditional on the response variables, the adaptive sampling reduces the computational costs to $O\left(nn^{*2}\right)$, where $n^* \ll n$ is the number of the sampled basis functions. The computational costs can also be reduced by the rounding algorithm [15]. This algorithm can significantly decrease the sample size to μ by rounding the data with a given precision, where $\mu \ll n$. After rounding, the computational costs can be dramatically reduced to $O\left(\mu^3\right)$.

The rest of the chapter is organized as follows. Section 2 provides a detailed introduction to SSANOVA models and the model estimation. The details of adaptive basis selection algorithm and rounding algorithm are reviewed in Section 3. In Appendix, we demonstrate the numerical implementations using the R software.

2. Smoothing spline ANOVA models

In this section, we first review smoothing spline models and the reproducing kernel Hilbert space. Second, we present how to decompose a nonparametric function on tensor product domains, which lays the theoretical foundation for SSANOVA models. In the end, we show the estimation of SSANOVA models and illustrate the model with a real data example.

2.1. Introduction of smoothing spline models

In the model (1), η is located in an infinite-dimensional space. One way to estimate it is to add some constraints and estimate η in a finite-dimensional space. With the smoothness constraint, we estimate η by minimizing the penalized likelihood functional (2), and the minimizer of (2) is called a smoothing spline.

Example 1. *Cubic smoothing splines*

Suppose that $Y|x$ follows a normal distribution, that is, $Y|x_i \sim N\big(\eta(x_i), \sigma^2\big)$. Then, the penalized likelihood functional (2) *can be reduced as the penalized least squares:*

$$\frac{1}{n}\sum_{i=1}^{n}\big(y_i - \eta(x_i)\big)^2 + \lambda\int_{\mathcal{X}}(\ddot{\eta}(x))^2 dx, \tag{3}$$

where $\ddot{\eta} = d^2\eta/dx^2$. The minimizer of (3) *is called a cubic smoothing spline* [16–18]. *In* (3), *the first term quantifies the fidelity to the data, and the second term controls the roughness of the function.*

Example 2. *Exponential family smoothing splines*

Suppose that $Y|x$ follows an exponential family distribution:

$$Y|x_i \sim \exp\big\{(y\eta(x_i) - b(\eta(x_i)))/a\big(\phi\big) + c\big(y, \phi\big)\big\},$$

where $a > 0$, b, and c are known functions and ϕ is either known or a nuisance parameter. Then, η can be estimated by minimizing the following penalized likelihood functional [19, 20]:

$$-\frac{1}{n}\sum_{i=1}^{n}\big\{y_i\eta(x_i) - b(\eta(x_i))\big\} + \lambda J(\eta). \tag{4}$$

Note that the cubic smoothing spline in Example 1 *is a special case of exponential family smoothing splines when $Y|x$ follows the Gaussian distribution.*

The smoothing parameter λ is sensitive to the estimation of η (see **Figure 1**). Therefore, it is crucial to implement some proper smoothing parameter selection methods to estimate λ. One of the most popular methods is the generalized cross validation (GCV) [21, 22]. More details will be discussed in Section 2.6.2.

2.2. Reproducing kernel Hilbert space

We assume that readers are familiar with Hilbert space, which is a complete vector space with an inner product well defined that allows length and angle to be measured [23]. In a general Hilbert space, the continuity of a functional, which is required in minimizing (2) on $\mathcal{H} = \{J(\eta) < \infty\}$, is not always satisfied. To circumvent the problem, we optimize (2) in a special Hilbert space named reproducing kernel Hilbert space [24].

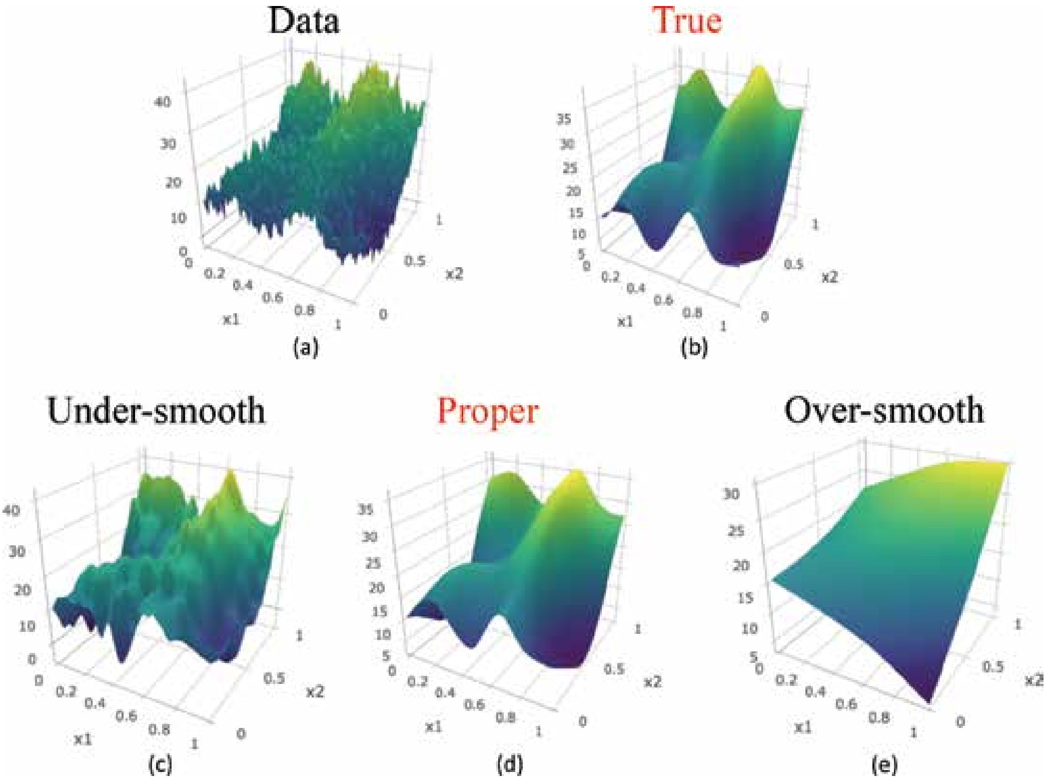


Figure 1. The data are generated by the model ϵ. $y = 5 + e^{3x_1} + 10^6 x_2^{11}(1 - x_2)^6 + 10^4 x_2^2(1 - x_2)^{10} + 5\cos(2\pi(x_1 - x_2)) + \epsilon$, where $\epsilon \sim N(0,1)$. Panels (a) and (b) show the data and the true function, respectively. The estimated functions depending on different smoothing parameters are shown in panels (c), (d), and (e). We set $\lambda \to 0$ in panel (c) and $\lambda \to \infty$ in panel (e). The proper λ selected by generalized cross validation (GCV) is used in panel (d).

For each $g \in \mathcal{H}$, there exists a corresponding continuous linear functional L_g such that $L_g(f) = \langle g, f \rangle$, where $f \in \mathcal{H}$ and $\langle \cdot, \cdot \rangle$ defines the inner product in $\mathcal{H}$. Conversely, an element $g_L \in \mathcal{H}$ can also be found such that $\langle g_L, f \rangle = L(f)$ for any continuous linear functional L of $\mathcal{H}$ [23]. This is known as the Riesz representation theorem.

Theorem 2.1. *Riesz representation*

Let $\mathcal{H}$ be a Hilbert space. For any functional L of $\mathcal{H}$, there uniquely exists an element $g_L \in \mathcal{H}$ such that

$$L(\cdot) = \langle g_L, \cdot \rangle,$$

where g_L is called the representer of L. The uniqueness is in the sense that g_1 and g_2 are considered as the same representer for any g_1 and g_2 satisfying $\|g_1 - g_2\| = 0$, where $\| \cdot \| = \langle \cdot, \cdot \rangle$ defines the norm in $\mathcal{H}$.

For a better construction of estimator minimizing (2), one needs the continuity of evaluation functional $[x]f = f(x)$. Roughly speaking, this means that if two functions f and g are close in norm, that is, $\|f - g\|$ is small, then f and g are also pointwise close, that is, $|f(x) - g(x)|$ is small for all x.

Definition 1. *Reproducing kernel Hilbert space*

Consider a Hilbert space $\mathcal{H}$ consisting of functions on domain $\mathcal{X}$. For every element $x \in \mathcal{X}$, define an evaluation functional $[x]$ such that $[x]f = f(x)$. If all the evaluation functional $[x]$s are continuous, $\forall x \in \mathcal{X}$, then $\mathcal{H}$ is called a reproducing kernel Hilbert space.

By Theorem 2.1, for every evaluation functional $[x]$, there exists a corresponding function $R_x \in \mathcal{H}$ on $\mathcal{X}$ as the representer of $[x]$, such that $\langle R_x, f\rangle = [x]f = f(x)$ and $\forall f \in \mathcal{H}$. By the definition of evaluation functional, it follows

$$R_x(y) = \left\langle R_x, R_y \right\rangle = R_y(x). \tag{5}$$

The bivariate function $R(x, y) = \left\langle R_x, R_y \right\rangle$ is called the reproducing kernel of $\mathcal{H}$, which is unique if it exists. The essential meaning of the name "reproducing kernel" comes from its reproducing property

$$\langle R(x, \cdot), f\rangle = \langle R_x(\cdot), f\rangle = f(x)$$

for any $f \in \mathcal{H}$. In general, a reproducing kernel Hilbert space defines a reproducing kernel function that is both symmetric and positive definite. In addition, Moore-Aronszajn theorem states that every symmetric, positive definite kernel defines a unique reproducing kernel Hilbert space [25], and hence one can construct a reproducing kernel Hilbert space simply by specifying its reproducing kernel.

We now introduce the concept of tensor sum decomposition. Suppose that $\mathcal{H}$ is a Hilbert space and $\mathcal{G}$ is a linear subspace of $\mathcal{H}$. The linear subspace $\mathcal{G}^c = \{f \in \mathcal{H}, \langle f, g\rangle = 0, \forall g \in \mathcal{G}\}$ is called the orthogonal complement of $\mathcal{G}$. It is easy to verify that for any $f \in \mathcal{H}$, there exists a unique decomposition $f = f_{\mathcal{G}} + f_{\mathcal{G}^c}$, where $f_{\mathcal{G}} \in \mathcal{G}$ and $f_{\mathcal{G}^c} \in \mathcal{G}^c$. This decomposition is called a tensor sum decomposition, denoted by $\mathcal{H} = \mathcal{G} \oplus \mathcal{G}^c$. Suppose that $\mathcal{H}_1$ and $\mathcal{H}_2$ are two Hilbert spaces with inner products $\langle\cdot,\cdot\rangle_1$ and $\langle\cdot,\cdot\rangle_2$. If the only common element of these two spaces is $\mathbf{0}$, one can also define a tensor sum Hilbert space $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2$. For any $f, g \in \mathcal{H}$, one has unique decompositions $f = f_1 + f_2$ and $g = g_1 + g_2$, where $f_1, g_1 \in \mathcal{H}_1$ and $f_2, g_2 \in \mathcal{H}_2$. Moreover, the inner product defined on $\mathcal{H}$ would be $\langle f, g\rangle = \left\langle f_1, g_1\right\rangle_1 + \left\langle f_2, g_2\right\rangle_2$. The following theorem provides the rules in the tensor sum decomposition of a reproducing kernel Hilbert space.

Theorem 2.2 *Suppose that R_1 and R_2 are the reproducing kernel Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. If $\mathcal{H}_1 \cap \mathcal{H}_2 = \{\mathbf{0}\}$, then $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2$ has a reproducing kernel $R = R_1 + R_2$.*

Conversely, if the reproducing kernel R of $\mathcal{H}$ can be decomposed into $R = R_1 + R_2$, where both R_1 and R_2 are positive definite, and they are orthogonal to each other, that is, $\langle R_1(x_1, \cdot), R_2(x_2, \cdot)\rangle = 0$ for $\forall x_1, x_2 \in \mathcal{X}$, then the spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ corresponding to the kernels R_1 and R_2 form a tensor sum decomposition $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2$.

2.3. Representer theorem

In (2), the smoothness penalty term $J(\eta) = J(\eta, \eta)$ is nonnegative definite, that is, $J(\eta, \eta) \geq 0$, and hence it is a squared semi-norm on the reproducing kernel Hilbert space $\mathcal{H} = \{\eta : J(\eta) < \infty\}$. Denote $\mathcal{N}_J = \{\eta \in \mathcal{H} : J(\eta) = 0\}$ as the null space of $J(\eta)$ and $\mathcal{H}_J$ as its orthogonal complement. By the tensor sum decomposition $\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J$, one may decompose the η into two parts: one in the null space $\mathcal{N}_J$ that has no contribution on the smoothness penalty and the other in $\mathcal{H}_J$ "reproduced" by the reproducing kernel $R(\cdot, \cdot)$ [12].

Theorem 2.3. *There exist coefficient vectors* $\mathbf{d} = (d_1, \dots, d_M)^T \in \mathbb{R}^M$ *and* $\mathbf{c} = (c_1, \dots, c_n)^T \in \mathbb{R}^n$ *such that*

$$\eta(x) = \sum_{k=1}^{M} d_k \xi_k(x) + \sum_{i=1}^{n} c_i R(x_i, x), \tag{6}$$

where $\{\xi_k, k = 1, .., M\}$ *is the basis of null space* $\mathcal{N}_J$ *and* $R(\cdot, \cdot)$ *is the reproducing kernel of* $\mathcal{H}_J$.

This theorem indicates that although the minimization problem is in an infinite-dimensional space, the minimizer of (2) lies in a data-adaptive finite-dimensional space.

2.4. Function decomposition

The decomposition of a multivariate function is similar to the classical ANOVA. In this section, we present the functional ANOVA which lays the foundation for SSANOVA models.

2.4.1. One-way ANOVA decomposition

We consider a classical one-way ANOVA model $y_{ij} = \mu_i + \epsilon_{ij}$, where y_{ij} is the observed data, μ_i is the treatment mean for $i = 1, \dots, K$ and $j = 1, \dots, J$, and ϵ_{ij}s are the random errors. The treatment mean μ_i can be further decomposed as $\mu_i = \mu + \alpha_i$, where μ is the overall mean and α_i is the treatment effect with the constraint $\sum_{i=1}^{K} \alpha_i = 0$.

Similar to the classical ANOVA decomposition, a univariate function $f(x)$ can be decomposed as

$$f = Af + (I - A)f = f_c + f_x, \tag{7}$$

where A is an averaging operator that averages the effect of x and I is an identity operator. The operator A averages a function f to a constant function f_c satisfying $A(I - A) = 0$. For example, one can take $Af = \int_0^1 f(x)dx$ in $\mathcal{L}_1[0, 1] = \left\{f : \int_0^1 |f(x)|dx < \infty\right\}$. In (7), $f_c = Af$ is the mean function, and $f_x = (I - A)f$ is the treatment effect.

2.4.2. Multiway ANOVA decomposition

On a d-dimensional product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j \in \mathbb{R}^d$, a multivariate function $f\left(x_{\langle 1 \rangle}, \dots, x_{\langle d \rangle}\right)$ can be decomposed similarly to the one-way ANOVA decomposition. Let A_j, $j = 1, \dots, d$, be the average operator on $\mathcal{X}_j$, and then $A_j f$ is a constant function on $\mathcal{X}_j$. One can define the ANOVA decomposition on $\mathcal{X}$ as

$$\begin{aligned} f &= \left\{ \prod_{j=1}^{d} \left(I - A_j + A_j \right) \right\} f \\ &= \sum_{\mathcal{S}} \left\{ \prod_{j \in \mathcal{S}} \left(I - A_j \right) \prod_{j \notin \mathcal{S}} A_j \right\} f = \sum_{\mathcal{S}} f_{\mathcal{S}}, \end{aligned} \tag{8}$$

where $\mathcal{S} \subseteq \{1, \dots, d\}$. The term $f_c = \prod_{j=1}^{d} A_j f$ is the constant function, $f_j = \left(I - A_j \right) \prod_{\alpha \neq j} A_\alpha f$ is the main effect term of $x_{\langle j \rangle}$, the term $f_{\mu\nu} = \left(I - A_\mu \right) \left(I - A_\nu \right) \prod_{\alpha \neq \mu, \nu} A_\alpha f$ is the interaction of $x_{\langle \mu \rangle}$ and $x_{\langle \nu \rangle}$, and so on.

2.5. Some examples of model conduction

Smoothing splines on $\mathcal{C}^{(m)}[0, 1]$. If we define

$$J(\eta) = \int_0^1 \left(\eta^{(m)} \right)^2 dx$$

in the space $\mathcal{C}^{(m)}[0, 1] = \left\{ f : f^{(m)} \in \mathcal{L}_2[0, 1] \right\}$, where $f^{(m)}$ denotes the mth differentiation of f, $\mathcal{L}_2 = \left\{ f : \int_0^1 \left(f(x) \right)^2 dx < \infty \right\}$, then the minimizer of (2) is called a polynomial smoothing spline. Here, we use an inner product

$$\langle f, g \rangle = \sum_{\nu=0}^{m-1} \left(\int_0^1 f^{(\nu)}(x) dx \right) \left(\int_0^1 g^{(\nu)}(x) dx \right) + \int_0^1 f^{(m)}(x) g^{(m)}(x) dx. \tag{9}$$

One can easily check that (9) is a well-defined inner product in $\mathcal{C}^{(m)}[0, 1]$ with $\mathcal{H}_0 = \left\{ f : f^{(m)} = 0 \right\}$ equipped with the inner product $\sum_{\nu=0}^{m-1} \left(\int_0^1 f^{(\nu)}(x) dx \right) \left(\int_0^1 g^{(\nu)}(x) dx \right)$ [21]. To construct the reproducing kernel, define

$$k_\nu(x) = -\left(\sum_{\mu=-\infty}^{-1} + \sum_{\mu=1}^{\infty} \right) \frac{\exp\left(2\pi \mathbf{i} \mu x \right)}{\left(2\pi \mathbf{i} \mu \right)^\nu}, \tag{10}$$

where $\mathbf{i} = \sqrt{-1}$. One can verify that $\int_0^1 k_\nu^{(\mu)}(x) dx = \delta_{\nu\mu}$ and $\nu, \mu = 0, 1, \dots, m-1$, where $\delta_{\nu\mu}$ is the Kronecker delta [26]. Indeed, $\{k_0, \dots, k_{m-1}\}$ forms an orthonormal basis of $\mathcal{H}_0$. Then, the reproducing kernel in $\mathcal{H}_0$ is

$$R_0(x, y) = \sum_{\nu=0}^{m-1} k_\nu(x) k_\nu(y).$$

For space

$$\mathcal{H}_1 = \left\{ f : \int_0^1 f^{(\nu)}(x)\,dx = 0, \nu = 0, 1, \ldots, m-1, f^{(m)} \in \mathcal{L}_2[0,1] \right\},$$

one can check that the reproducing kernel in $\mathcal{H}_1$ is

$$R_1(x,y) = k_m(x)k_m(y) + (-1)^{m-1}k_{2m}(x-y),$$

(See details in [11]; Section~2.3).

SSANOVA models on product domains: A natural way to construct reproducing kernel Hilbert space on product domain $\prod_{j=1}^d \mathcal{X}_j$ is taking the tensor product of spaces constructed on the marginal domains $\mathcal{X}_j$s. According to the Moore-Aronszajn theorem, every nonnegative definite function R corresponds to a reproducing kernel Hilbert space with R as its reproducing kernel. Therefore, the construction of the tensor product reproducing kernel Hilbert space is induced by constructing its reproducing kernel.

Theorem 2.4. *Suppose that if* $R_{\langle 1\rangle}\left(x_{\langle 1\rangle}, y_{\langle 1\rangle}\right)$ *is nonnegative definite on* $\mathcal{X}_1$ *and* $R_{\langle 2\rangle}\left(x_{\langle 2\rangle}, y_{\langle 2\rangle}\right)$ *is nonnegative definite on* $\mathcal{X}_2$*, then* $R(x,y) = R_{\langle 1\rangle}\left(x_{\langle 1\rangle}, y_{\langle 1\rangle}\right)R_{\langle 2\rangle}\left(x_{\langle 2\rangle}, y_{\langle 2\rangle}\right)$ *is nonnegative definite on* $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2$.

Theorem 2.4 implies that a reproducing kernel R on tensor product reproducing kernel Hilbert space can be derived from the reproducing kernels on marginal domains. Indeed, let $\mathcal{H}_{\langle j\rangle}$ be the space on $\mathcal{X}_j$ with reproducing kernel $R_{\langle j\rangle}$, where $j = 1, 2$. Then, $R = R_{\langle 1\rangle}R_{\langle 2\rangle}$ is nonnegative definite on $\mathcal{X}_1 \times \mathcal{X}_2$. The space $\mathcal{H}$ corresponding to $R(\cdot,\cdot)$ is called the tensor product space of $\mathcal{H}_{\langle 1\rangle}$ and $\mathcal{H}_{\langle 2\rangle}$, denoted by $\mathcal{H} = \mathcal{H}_{\langle 1\rangle} \otimes \mathcal{H}_{\langle 2\rangle}$.

One can decompose each marginal space $\mathcal{H}_{\langle j\rangle}$ into $\mathcal{H}_{\langle j\rangle} = \mathcal{H}_{\langle j\rangle 0} \oplus \mathcal{H}_{\langle j\rangle 1}$, where $\mathcal{H}_{\langle j\rangle 0}$ denotes the averaging space and $\mathcal{H}_{\langle j\rangle 1}$ denotes the orthogonal complement. Then, by the discussion in Section 2.4, the one-way ANOVA decomposition on each marginal space can be generalized to a multidimensional space $\mathcal{H} = \otimes_{j=1}^d \mathcal{H}_{\langle j\rangle}$ as

$$\begin{aligned}\mathcal{H} &= \otimes_{j=1}^d \left(\mathcal{H}_{\langle j\rangle 0} \oplus \mathcal{H}_{\langle j\rangle 1}\right) \\ &= \oplus_{\mathcal{S}}\left\{\left(\otimes_{j\in\mathcal{S}}\mathcal{H}_{\langle j\rangle 1}\right) \otimes \left(\otimes_{j\notin\mathcal{S}}\mathcal{H}_{\langle j\rangle 0}\right)\right\} \\ &= \oplus_{\mathcal{S}}\mathcal{H}_{\mathcal{S}},\end{aligned} \tag{11}$$

where $\mathcal{S}$ denotes all the subsets of $\{1, \ldots, d\}$. The component $f_{\mathcal{S}}$ in (8) is in the space $\mathcal{H}_{\mathcal{S}}$. Based on the decomposition, the minimizer of (2) is called a tensor product smoothing spline. One can construct a tensor product smoothing spline following Theorem 2.3, in which the reproducing kernel term may be calculated in the same way as the tensor product (11).

In the following, we will give some examples of tensor product smoothing splines on product domains.

2.5.1. *Smoothing splines on* $\{1, \ldots, K\} \times [0, 1]$

We construct the reproducing kernel Hilbert space by using

$$R_{\langle 1\rangle 0} = 1/K \text{and} R_{\langle 1\rangle 1} = I_{\left[x_{\langle 1\rangle} = y_{\langle 1\rangle}\right]}$$

on $\{1, \ldots, K\}$. On $[0, 1]$, assume that if $m = 2$, then we have

$$R_{\langle 2\rangle 0} = 1 + k_1\left(x_{\langle 2\rangle}\right)k_1\left(y_{\langle 2\rangle}\right)$$

and

$$R_{\langle 2\rangle 1} = k_2\left(x_{\langle 2\rangle}\right)k_2\left(y_{\langle 2\rangle}\right) - k_4\left(x_{\langle 2\rangle} - y_{\langle 2\rangle}\right).$$

In this case, the space $\mathcal{H}$ can be further decomposed as

$$\mathcal{H} = \left(\mathcal{H}_{\langle 1\rangle 0} \oplus \mathcal{H}_{\langle 1\rangle 1}\right) \otimes \left(\mathcal{H}_{\langle 2\rangle 00} \oplus \mathcal{H}_{\langle 2\rangle 01} \oplus \mathcal{H}_{\langle 2\rangle 1}\right). \tag{12}$$

The reproducing kernels of tensor product cubic spline on $\{1, \ldots, K\} \times [0, 1]$ are listed in **Table 1**.

On other product domains, for example, $[0, 1]^2$, the tensor product reproducing kernel Hilbert space can be decomposed in a similar way. More examples are available in ([11], Section~2.4).

2.5.1.1. *General form*

In general, a tensor product reproducing kernel Hilbert space can be specified as $\mathcal{H} = \oplus_j \mathcal{H}_j$, where $j \in B$ is a genetic index. Suppose that $\mathcal{H}_j$ is equipped with a reproducing kernel R_j and an inner product $\langle f, g\rangle_j$. Denote f_j as the projection of f onto $\mathcal{H}_j$. Then, an inner product in $\mathcal{H}$ can be defined as

Subspace	**Reproducing kernel**
$\mathcal{H}_{\langle 1\rangle 0} \otimes \mathcal{H}_{\langle 2\rangle 00}$	$1/K$
$\mathcal{H}_{\langle 1\rangle 0} \otimes \mathcal{H}_{\langle 2\rangle 01}$	$k_1\left(x_{\langle 2\rangle}\right)k_1\left(y_{\langle 2\rangle}\right)/K$
$\mathcal{H}_{\langle 1\rangle 0} \otimes \mathcal{H}_{\langle 2\rangle 1}$	$\left[k_2\left(x_{\langle 2\rangle}\right)k_2\left(y_{\langle 2\rangle}\right) - k_4\left(x_{\langle 2\rangle} - y_{\langle 2\rangle}\right)\right]/K$
$\mathcal{H}_{\langle 1\rangle 1} \otimes \mathcal{H}_{\langle 2\rangle 00}$	$I_{\left[x_{\langle 1\rangle} = y_{\langle 1\rangle}\right]} - 1/K$
$\mathcal{H}_{\langle 1\rangle 1} \otimes \mathcal{H}_{\langle 2\rangle 01}$	$\left[I_{\left[x_{\langle 1\rangle} = y_{\langle 1\rangle}\right]} - 1/K\right]k_1\left(x_{\langle 2\rangle}\right)k_1\left(y_{\langle 2\rangle}\right)$
$\mathcal{H}_{\langle 1\rangle 1} \otimes \mathcal{H}_{\langle 2\rangle 1}$	$\left[I_{\left[x_{\langle 1\rangle} = y_{\langle 1\rangle}\right]} - 1/K\right]\left[k_2\left(x_{\langle 2\rangle}\right)k_2\left(y_{\langle 2\rangle}\right) - k_4\left(x_{\langle 2\rangle} - y_{\langle 2\rangle}\right)\right]$

Table 1. Reproducing kernels of (12) on $\{1, \ldots, K\} \times [0, 1]$ when $m = 2$.

$$J(f,g) = \sum_j \theta_j^{-1} \left\langle f_j, g_j \right\rangle_j, \tag{13}$$

where $\theta_j \geq 0$ are the tuning parameters. If a penalty J in (2) has the form (13), the SSANOVA models can be defined on the space $\mathcal{H} = \oplus_j \mathcal{H}_j$ with the reproducing kernel:

$$R = \sum_j \theta_j R_j. \tag{14}$$

2.6. Estimation

In this section, we show the procedure of estimating the minimizer $\widehat{\eta}$ of (2) under the Gaussian assumption and selecting the smoothing parameters.

2.6.1. Penalized least squares

We consider the same model shown in (1), and then the η can be estimated by minimizing the penalized least squares:

$$\frac{1}{n}\sum_{i=1}^{n} \left(y_i - \eta(x_i)\right)^2 + \lambda J(\eta). \tag{15}$$

Let S denote the $n \times M$ matrix with the (i,j)th entry $\xi_j(x_i)$ as in (6) and R denote the $n \times n$ matrix with the (i,j)th entry $R\left(x_i, x_j\right)$ with the form (14). Then, based on Theorem 2.3, $\boldsymbol{\eta}$ can be expressed as

$$\boldsymbol{\eta} = S\mathbf{d} + R\mathbf{c},$$

where $\boldsymbol{\eta} = (\eta(x_1), \ldots, \eta(x_n))^T$, $\mathbf{d} = (d_1, \ldots, d_M)^T$, and $\mathbf{c} = (c_1, \ldots, c_n)^T$. The least squares term in (15) becomes

$$\frac{1}{n}\sum_{i=1}^{n} \left(y_i - \eta(x_i)\right)^2 = \frac{1}{n}(\mathbf{y} - S\mathbf{d} - R\mathbf{c})^T(\mathbf{y} - S\mathbf{d} - R\mathbf{c}),$$

where $\mathbf{y} = \left(y_1, \ldots, y_n\right)^T$.

By the reproducing property (5), the roughness penalty term can be expressed as

$$J(\eta) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i R\left(x_i, x_j\right) c_j = \mathbf{c}^T R\mathbf{c}.$$

Therefore, the penalized least squares criterion (15) becomes

$$\frac{1}{n}(\mathbf{y} - S\mathbf{d} - R\mathbf{c})^T(\mathbf{y} - S\mathbf{d} - R\mathbf{c}) + \lambda \mathbf{c}^T R\mathbf{c}. \tag{16}$$

The penalized least squares (16) is a quadratic form of both **d** and **c**. By differentiating (16), one can obtain the linear system:

$$\begin{pmatrix} S^TS & S^TR \\ R^TS & R^TR + n\lambda R \end{pmatrix} \begin{pmatrix} \mathbf{d} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} S^T\mathbf{y} \\ R^T\mathbf{y} \end{pmatrix}. \tag{17}$$

Note that (17) only works for penalized least squares (15), and hence a normal assumption is needed in this case.

2.6.2. Selection of smoothing parameters

In SSANOVA models, properly selecting smoothing parameters is important to estimate η [9, 27, 28], as shown in **Figure 1**. Here, we introduce the generalized cross validation (GCV) method for the smoothing parameter selection.

For the multivariate predictors, the penalty term in (15) has the form

$$\lambda J(f) = \lambda \sum_{j=1}^{S} \theta_j^{-1} \left\langle f_j, f_j \right\rangle_j,$$

where S is the number of smoothing parameters, which is related to the functional ANOVA decomposition, and θ_j's are the extra smoothing parameters. To avoid overparameterization, we treat $\lambda = (\lambda/\theta_1, \cdots, \lambda/\theta_S)^T$ as the effective smoothing parameters.

A GCV score is defined as

$$V(\lambda) = \frac{n^{-1}\mathbf{y}^T(I - A(\lambda))^2\mathbf{y}}{[n^{-1}\mathrm{tr}(I - A(\lambda))]^2},$$

where $A(\lambda)$ is a symmetric matrix similar to the hat matrix in linear regression. We can select a proper λ by minimizing the GCV score [21].

2.7. Case study: Twitter data

Tweets in the contiguous United States were collected over five weekdays in January 2014. The dataset contains information of time, GPS location, and tweet counts (see **Figure 2**). To illustrate the application of SSANOVA models, we study the time and spatial patterns in this data.

The bivariate function $\eta(x_{\langle 1\rangle}, x_{\langle 2\rangle})$ is a function of time and location, where $x_{\langle 1\rangle}$ denotes the time and $x_{\langle 2\rangle}$ represents the longitude and latitude coordinates. We use the thin-plate spline for the spatial variable and cubic spline for the time variable. As a rotation-free method, the thin-plate spline is popular for modeling spatial data [29–31]. For a better interpretation, we decompose the function η as

$$\eta(x_{\langle 1\rangle}, x_{\langle 2\rangle}) = \eta_c + \eta_1(x_{\langle 1\rangle}) + \eta_2(x_{\langle 2\rangle}) + \eta_{12}(x_{\langle 1\rangle}, x_{\langle 2\rangle}),$$

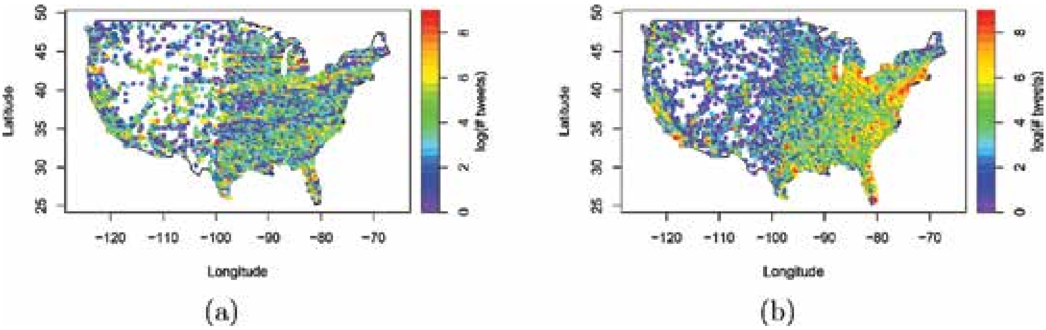


Figure 2. Heatmaps of tweet counts in the contiguous United States. (a) Tweet counts at 2:00 a.m. (b) Tweet counts at 6:00 p.m.

where η_c is a constant function; η_1 and η_2 are the main effects of time and location, respectively; and η_{12} is the spatial-time interaction effect.

The main effects of time and location are shown in **Figure 3**. Obviously, in panel (a), the number of tweets has the periodic effect, where it attains the maximum value at 8:00 p.m. and the minimum value at 5:00 a.m. The main effect of time shows the variations of Twitter usages in the United States. In addition, we can infer how the tweet counts vary across different locations based on panel (b) in **Figure 3**. There tend to be more tweets in the east than those in the west regions and more tweets in the coastal zone than those in the inland. We use the scaled dot product

$$\pi = (\widehat{\boldsymbol{\eta}}_{12})^T \widehat{\mathbf{y}} / \|\widehat{\mathbf{y}}\|^2$$

to quantify the percentage decomposition of the sum of squares of $\widehat{\mathbf{y}}$ [11], where $\widehat{\mathbf{y}} = (\widehat{y}_1, \ldots, \widehat{y}_n)^T$ is the predicted values of $\log(\#\text{Tweets})$, and $\widehat{\boldsymbol{\eta}}_{12} = (\eta_{12}(x_1), \ldots, \eta_{12}(x_n))^T$ is the estimated interaction effect term, where $\eta_{12}(x) = \eta_{12}(x_{\langle 1 \rangle}, x_{\langle 2 \rangle})$. In our fitted model,

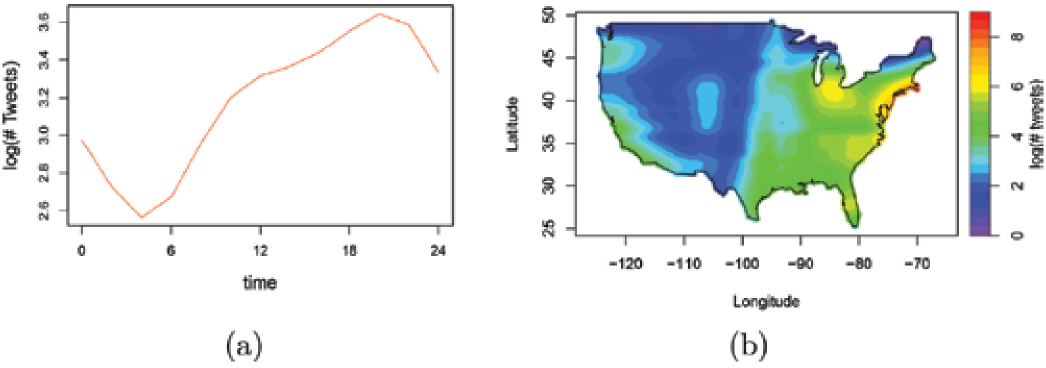


Figure 3. (a) The main effect function of time (hours). (b) The main effect function of location.

$\pi = 3 \times 10^{-16}$, which is so small that the interaction term is negligible. This indicates that there is no significant difference for the Twitter usages across time in the contiguous United States.

3. Efficient approximation algorithm in massive datasets

In this section, we consider SSANOVA models under the big data settings. The computational cost of solving (17) is of the order $O(n^3)$ and thus gives rise to a challenge on the application of SSANOVA models when the volume of data grows. To reduce the computational load, an obvious way is to select a subset of basis functions randomly. However, it is hard to keep the data features by uniform sampling. In the following section, we present an adaptive basis selection method and show its advantages over uniform sampling [14]. Instead of selecting basis functions, another approach to reduce the computational cost is shrinking the original sample size by rounding algorithm [15].

3.1. Adaptive basis selection

A natural way to select the basis functions is through uniform sampling. Suppose that we randomly select a subset $\breve{x} = \{\breve{x}_1, \ldots, \breve{x}_{\breve{n}}\}$ from $\{x_1, \ldots, x_n\}$, where $\breve{n}$ is the subsample size. Thus, the kernel matrix would be $R(\breve{x}_i, x)$, $i = 1, \ldots, \breve{n}$. Then, one minimizes (17) in the effective model space:

$$\mathcal{H}_E = \mathcal{N} \oplus \text{span}\{R(\breve{x}_i, x), i = 1, 2, \ldots, \breve{n}\}.$$

The computational cost will be reduced significantly to $O(n\breve{n}2)$ if $\breve{n}$ is much smaller than n. Furthermore, it can be proven that the minimizer of (2), $\breve{\eta}$, by uniform sampling basis selection, has the same asymptotic convergence rate as the full basis minimizer $\widehat{\eta}$.

Although the uniform basis selection reduces the computational cost and the corresponding $\breve{\eta}$ achieves the optimal asymptotic convergence rate, it may fail to retain the data features occasionally. For example, when the data are not evenly distributed, it is hard for uniform sampling to capture the feature where there are only a few data points. In [14], an adaptive basis selection method is proposed. The main idea is to sample more basis functions where the response functions change largely and fewer basis functions on those flat regions. More details of adaptive basis selection method are shown in the following procedure:

Step 1 Divide the range of responses $\{y_i\}_{i=1}^n$ into K disjoint intervals, $S_1, \ldots, S_K$. Denote $|S_k|$ as the number of observations in S_k.

Step 2 For each S_k, draw a random sample of size n_k from this collection. Let $x^{*(k)} = \left(x_1^{*(k)}, \ldots, x_{n_k}^{*(k)}\right)$ be the predictor values.

Step 3 Combine $x^{*(1)}, \ldots, x^{*(K)}$ together to form a set of sampled predictor values $\{x_1^*, \ldots, x_{n*}^*\}$, where $n^* = \sum_{k=1}^K n_k$.

Step 4 Define

$$\mathcal{H}_E = \mathcal{H}_0 \oplus \text{span}\{R(x_i^*, \cdot), i = 1, 2, \ldots, n^*\}$$

as the effective model space.

By adaptive basis selection, the minimizer of (2) keeps the same form as that in Theorem 2.3:

$$\eta_A(x) = \sum_{k=1}^{M} d_k \xi_k(x) + \sum_{i=1}^{n^*} c_i R(x_i^*, x).$$

Let R_* be an $n \times n^*$ matrix, and its (i, j)th entry is $R\left(x_i, x_j^*\right)$. Let R_{**} be an $n^* \times n^*$ matrix, and its (i, j)th entry is $R\left(x_i^*, x_j^*\right)$. Then, the estimator $\boldsymbol{\eta}_A$ satisfies

$$\boldsymbol{\eta}_A = S\mathbf{d}_\mathbf{A} + R_* \mathbf{c}_\mathbf{A},$$

where $\boldsymbol{\eta}_A = \left(\eta_A(x_1), \cdots, \eta_A(x_{n^*})\right)^T$, $\mathbf{d}_\mathbf{A} = (d_1, \ldots, d_M)^T$, and $\mathbf{c}_\mathbf{A} = (c_1, \ldots, c_{n^*})^T$. Similar to (17), the linear system of equations in this case is

$$\begin{pmatrix} S^T S & S^T R_* \\ R_*^T S & R_*^T R_* + n\lambda R_{**} \end{pmatrix} \begin{pmatrix} \mathbf{d}_\mathbf{A} \\ \mathbf{c}_\mathbf{A} \end{pmatrix} = \begin{pmatrix} S^T \mathbf{y} \\ R_*^T \mathbf{y} \end{pmatrix}. \tag{18}$$

The computational complexity of solving (18) is of the order $O(nn^{*2})$, so the method decreases the computational cost significantly. It can also be shown that the adaptive sampling basis selection smoothing spline estimator η_A has the same convergence property as the full basis method. More details about the consistency theory can be found in [14]. Moreover, adaptive sampling basis selection method for exponential family smoothing spline models was developed in [32].

3.2. Rounding algorithm

Other than sampling a smaller set of basis functions to save the computational resources, for example, the adaptive basis selection method presented previously, [15] proposed a new rounding algorithm to fit SSANOVA models in the context of big data.

Rounding algorithm: The details of rounding algorithm can be shown in the following procedure:

Step 1 Assume that all predictors are continuous.

Step 2 Convert all predictors to the interval $[0, 1]$.

Step 3 Round the raw data by using the transformation:

$$z_{i\langle j\rangle} = RD\left(x_{i\langle j\rangle}/r_{\langle j\rangle}\right) r_{\langle j\rangle}, \text{ for } i \in \{1, \cdots, n\}, j \in \{1, \cdots, d\},$$

where the rounding parameter $r_{\langle j\rangle} \in (0, 1]$ and rounding function $RD(\cdot)$ transform input data to the nearest integer.

Step 4 After replacing $x_{i\langle j\rangle}$ with $z_{i\langle j\rangle}$, we redefine S and R in (16) and then estimate η by minimizing the penalized least squares (16).

Remark 1 In Step 3, if $r_{\langle j\rangle}$ is the rounding parameter for jth predictor and its value is 0.03, then each $z_{i\langle j\rangle}$ is formed by rounding the corresponding $x_{i\langle j\rangle}$ to its nearest 0.03.

Remark 2 It is evident that the value of rounding parameter can influence the precision of approximation. The smaller the rounding parameter, the better the model estimation and the higher the computational cost.

Computational benefits: We now briefly explain why the implementation of rounding algorithm can reduce the computational loads. For example, if the rounding parameter $r = 0.01$, it is obvious that $u \leq 101$, where u denotes the number of uniquely observed values. In conclusion, using user-tunable rounding algorithm can dramatically reduce the computational burden of fitting SSANOVA models from the order of $O(n^3)$ to $O(u^3)$, where $u \ll n$.

Case study: To illustrate the benefit of the rounding algorithm, we apply the algorithm to the electroencephalography (EEG) dataset. Note that EEG is a monitoring method to record the electrical activity of the brain. It can be used to diagnose sleep disorders, epilepsy, encephalopathies, and brain death.

The dataset [33] contains 44 controls and 76 alcoholics. Each subject was repeatedly measured 10 times by using visual stimulus at a frequency of 256 Hz. This brings about $n = 10$ replications $\times 120$ subjects $\times 256$ time points $= 307,200$ observations. There are two predictors, time and group (control vs. alcoholic). We apply the cubic spline to the time effect and the nominal spline to the group effect.

After applying the model to the unrounded data, rounded data with rounding parameter $r = 0.01$ and $r = 0.05$ for time covariate, we can obtain a summary table about GCV, AIC [34], BIC [35], and running time in **Table 2**.

Based on **Table 2**, we can easily see that there are no significant difference among the GCV scores and AIC/BIC. In addition using rounding algorithm reduces 92% CPU time compared to using unrounded dataset.

	GCV	**AIC**	**BIC**	**CPU time (seconds)**
Unrounded data	85.9574	2,240,019	2,240,562	15.65
Rounded data with $r = 0.01$	86.6667	2,242,544	2,242,833	1.22
Rounded data with $r = 0.05$	86.7654	2,242,893	2,243,089	1.13

Table 2. Fit statistics and running time for SSANOVA models.

4. Conclusion

Smoothing spline ANOVA (SSANOVA) models are widely used in applications [11, 20, 36, 37]. In this chapter, we introduced the general framework of the SSANOVA models in Section 2. In

Section 3, we discussed the models under the big data settings. When the volume of data grows, fitting the models is computing-intensive [11]. The adaptive basis selection algorithm [14] and rounding algorithm [15] we presented can significantly reduce the computational cost.

Acknowledgements

This work is partially supported by the NIH grants R01 GM122080 and R01 GM113242; NSF grants DMS-1222718, DMS-1438957, and DMS-1228288; and NSFC grant 71331005.

Conflict of interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interests; and expert testimony or patent-licensing arrangements) or nonfinancial interest (such as personal or professional relationships, affiliations, knowledge, or beliefs) in the subject matter or materials discussed in this manuscript.

Appendix

In this appendix, we use two examples to illustrate how to implement smoothing spline ANOVA (SSANOVA) models in R. The gss package in R, which can be downloaded on the CRAN https://cran.r-project.org/, is utilized.

We now load the gss package:

```
library(gss)
```

Example I: Apply the smoothing spline to a simulated dataset.

Suppose that the predictor x follows a uniform distribution on $[0,1]$, and the response y is generated based on $y = 5 + 2\cos(3\pi x) + \epsilon$, where $\epsilon \sim N(0,1)$.

```
x<-runif(100);y<-5+2*cos(3*pi*x)+rnorm(x)
```

Then, fit cubic smoothing spline model:

```
cubic.fit<-ssanova(y˜x)
```

To evaluate the predicted values, one uses:

```
new<-data.frame(x=seq(min(x),max(x),len=50))
```

```
est<-predict(cubic.fit,new,se=TRUE)
```

The `se.fit` parameter indicates if one can get the pointwise standard errors for the predicted values. The predicted values and Bayesian confidence interval, shown in **Figure 4**, are generated by:

```
plot(x,y,col=1)

lines(new$x,est$fit,col=2)

lines(new$x,est$fit+1.96*est$se,col=3)

lines(new$x,est$fit-1.96*est$se,col=3)
```

Example II: Apply the SSANOVA model to a real dataset.

In this example, we illustrate how to implement the SSANOVA model using the gss package. The data is from an experiment in which a single-cylinder engine is run with ethanol to see how the nox concentration `nox` in the exhaust depends on the compression ratio `comp` and the equivalence ratio `equi`. The fitted model contains two predictors (`comp` and `equi`) and one interaction term.

```
data(nox)

nox.fit <- ssanova(log10(nox)˜comp*equi,data=nox)
```

The predicted values are shown in **Figure 5**.

```
x=seq(min(nox$comp),max(nox$comp),len=50)

y=seq(min(nox$equi),max(nox$equi),len=50)

temp <- function(x, y){
```

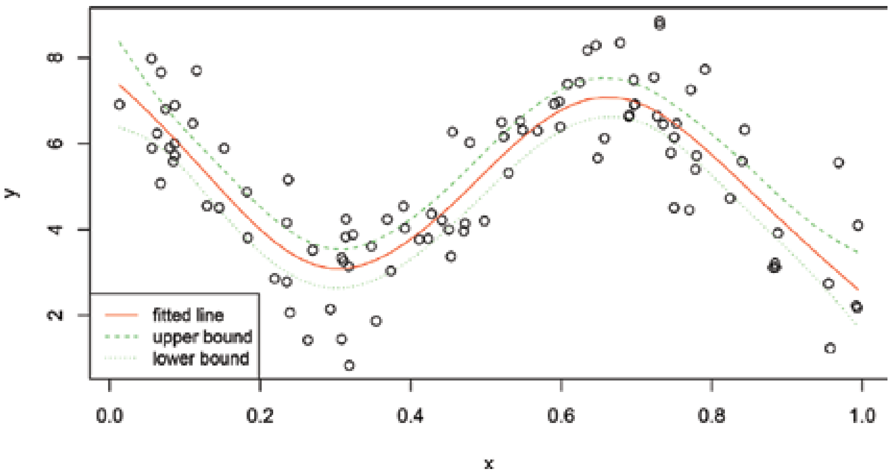


Figure 4. The solid red line represents the fitted values. The green lines represent the 95% Bayesian confidence interval. The raw data are shown as the circles.

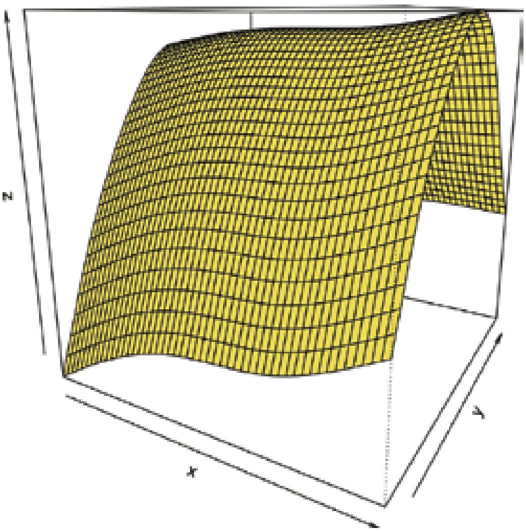


Figure 5. The x-axis, y-axis, and z-axis represent the compression ratio, the equivalence ratio, and the predicted values, respectively.

```
new=data.frame(comp=x,equi=y)

return(predict(nox.fit,new,se=FALSE))

}

z=outer(x, y, temp)

persp(x,y,z,theta = 30).
```

Author details

Jingyi Zhang, Honghe Jin, Ye Wang, Xiaoxiao Sun, Ping Ma* and Wenxuan Zhong

*Address all correspondence to: pingma@uga.edu

Department of Statistics, The University of Georgia, Athens, GA, USA

References

[1] Buja A, Hastie T, Tibshirani R. Linear smoothers and additive models. The Annals of Statistics. 1989;**17**:453-510

[2] Burman P. Estimation of generalized additive models. Journal of Multivariate Analysis. 1990;**32**(2):230-255

[3] Friedman JH, Grosse E, Stuetzle W. Multidimensional additive spline approximation. SIAM Journal on Scientific and Statistical Computing. 1983;**4**(2):291-301

[4] Hastie TJ. Generalized additive models. In: Statistical Models in S. Routledge; 2017. pp. 249-307

[5] Stone CJ. Additive regression and other nonparametric models. The Annals of Statistics. 1985;**13**:689-705

[6] Stone CJ. The dimensionality reduction principle for generalized additive models. The Annals of Statistics. 1986;**14**:590-606

[7] Barry D et al. Nonparametric bayesian regression. The Annals of Statistics. 1986;**14**(3):934-953

[8] Chen Z. Interaction Spline Models. University of Wisconsin–Madison; 1989

[9] Gu C, Wahba G. Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. SIAM Journal on Scientific and Statistical Computing. 1991;**12**(2):383-398

[10] Wahba G. Partial and Interaction Splines for the Semiparametric Estimation of Functions of Several Variables. University of Wisconsin, Department of Statistics; 1986

[11] Gu C, Smoothing Spline ANOVA. Models, Volume 297. In: Springer Science & Business Media; 2013

[12] Wahba G. Spline Models for Observational Data. SIAM; 1990

[13] Wang Y. Smoothing Splines: Methods and Applications. CRC Press; 2011

[14] Ma P, Huang JZ, Zhang N. Efficient computation of smoothing splines via adaptive basis sampling. Biometrika. 2015;**102**(3):631-645

[15] Helwig NE, Ma P. Smoothing spline ANOVA for super-large samples: Scalable computation via rounding parameters. Statistics and Its Interface, Special Issue on Statistical and Computational Theory and Methodology for Big Data. 2016;**9**:433-444

[16] Green PJ, Silverman BW. Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach. CRC Press; 1993

[17] Kimeldorf GS, Wahba G. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. The Annals of Mathematical Statistics. 1970;**41**(2): 495-502

[18] Kimeldorf GS, Wahba G. Spline functions and stochastic processes. Sankhya: The Indian Journal of Statistics, Series A; 1970. pp. 173-180

[19] O'sullivan F, Yandell BS, Raynor WJ Jr. Automatic smoothing of regression functions in generalized linear models. Journal of the American Statistical Association. 1986;**81**(393):96-103

[20] Wahba G, Wang Y, Gu C, Klein R, Klein B. Smoothing spline ANOVA for exponential families, with application to the Wisconsin epidemiological study of diabetic retinopathy. The Annals of Statistics; 1995:1865-1895

[21] Craven P, Wahba G. Smoothing noisy data with spline functions. Numerische Mathematik. 1978;**31**(4):377-403

[22] Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics. 1979;**21**(2):215-223

[23] Kreyszig E. Introductory Functional Analysis with Applications, Volume 1. New York: Wiley; 1989

[24] Berlinet A, Thomas-Agnan C. Reproducing Kernel Hilbert Spaces in Probability and Statistics. In: Springer Science & Business Media; 2011

[25] Aronszajn N. Theory of reproducing kernels. Transactions of the American Mathematical Society. 1950;**68**(3):337-404

[26] Abramowitz M, Stegun IA. Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables, Volume 55. Courier Corporation; 1964

[27] Hurvich CM, Simonoff JS, Tsai C-L. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 1998;**60**(2):271-293

[28] Mallows CL. Some comments on Cp. Technometrics. 2000;**42**(1):87-94

[29] Duchon J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. Constructive Theory of Functions of Several Variables; 1977. pp. 85-100

[30] Meinguet J. Multivariate interpolation at arbitrary points made simple. Zeitschrift für Angewandte Mathematik und Physik (ZAMP). 1979;**30**(2):292-304

[31] Wahba G, Wendelberger J. Some new mathematical methods for variational objective analysis using splines and cross validation. Monthly Weather Review. 1980;**108**(8):1122-1143

[32] Ma P, Zhang N, Huang JZ, Zhong W. Adaptive basis selection for exponential family smoothing splines with application in joint modeling of multiple sequencing samples. Statistica Sinica, in press; 2017

[33] Lichman M. UCI Machine Learning Repository. 2013. URL. http://archive.ics.uci.edu/ml

[34] Akaike H. Information theory and an extension of the maximum likelihood principle. In: Parzen E, Tanabe K, Kitagawa G, editors. Selected Papers of Hirotugu Akaike, Springer Series in Statistics. New York: Springer; 1998. pp. 199-213

[35] Schwarz G. Estimating the dimension of a model. The Annals of Statistics. 1978;**6**(2):461-464. DOI: 10.1214/aos/1176344136

[36] Helwig NE, Shorter KA, Ma P, Hsiao-Wecksler ET. Smoothing spline analysis of variance models: A new tool for the analysis of cyclic biomechanical data. Journal of Biomechanics. 2016;**49**(14):3216-3222

[37] Lin X, Wahba G, Xiang D, Gao F, Klein R, Klein B. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. The Annals of Statistics. 2000;**28**:1570-1600

Model Testing Based on Regression Spline

Na Li

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74858

Abstract

Tests based on regression spline are developed in this chapter for testing nonparametric functions in nonparametric, partial linear and varying-coefficient models, respectively. These models are more flexible than linear regression model. However, one important problem is if it is really necessary to use such complex models which contain nonparametric functions. For this purpose, p-values for testing the linearity and constancy of the nonparametric functions are established based on regression spline and fiducial method. In the application of spline-based method, the determination of knots is difficult but plays an important role in inferring regression curve. In order to infer the nonparametric regression at different smoothing levels (scales) and locations, multi-scale smoothing methods based on regression spline are developed to test the structures of the regression curve and compare multiple regression curves. It could sidestep the determination of knots; meanwhile, it could give a more reliable result in using the spline-based method.

Keywords: fiducial method, multi-scale smoothing method, nonparametric regression model, partial linear regression model, regression spline, varying-coefficient regression model

1. Introduction

It is well known that the model which contains nonparametric functions, such as partial linear model and varying-coefficient model, plays an important role in applications due to its flexible structure. However, in practice, investigators often want to know whether it is really necessary to fit the data with such more complex models rather than a simpler model. This amounts to testing the linearity of nonparametric functions in a regression model. In this chapter, we first consider the following three frequently used regression models.

Nonparametric regression model:

$$y = f(x) + \varepsilon. \tag{1}$$

Partial linear regression model:

$$y = Z'b + f(x) + \varepsilon. \tag{2}$$

Varying-coefficient model:

$$y = z_1 f_1(x_1) + \ldots + z_p f_p(x_p) + \varepsilon. \tag{3}$$

In models (1)–(3), y is the response variable, $Z = (z_1, \cdots, z_p)$ is a p-dimensional regressor, x and $x_1, \cdots, x_p$ are covariant taking values in a finite interval, ε is the error, b is a parameter vector, and *f(x)* and $f_j(x_j), j = 1, 2, \ldots, p$ are unknown smooth functions. Usually we suppose that (z, x) and ε are independent and $\varepsilon \tilde{} F(\cdot/\sigma)$, where F is a known cumulate distribution function (cdf) with mean 0 and variance 1; σ is unknown. Without loss of generality, we can suppose that x and $x_1, \cdots, x_p$ take values in [0, 1]. We try to test the linearity of $f(x)$ in models (1) and (2) and the constancy of $f_j(x_j)$ in model (3) for some $j \in (1, 2, \cdots, p)$.

The hypothesis testing in nonparametric regression model was considered in many papers. Härdle and Mammen [1] developed the visible difference between a parametric and a nonparametric curve estimates. Based on smoothing techniques, many tests were constructed for testing the linearity in regression model; see Hart [2], Cox et al. [3], and Cox and Koh [4] for a review. Recently, Fan et al. [5] studied a generalized likelihood ratio statistic, which behaves well in large sample case. Tests based on penalized criterion were developed by Eubank and Hart [6] and Baraud [7].

The linearity of partial linear regression model (2) was studied by Bianco and Boente [8], Liang et al. [9], and Fan and Huang [10]. There are also many other papers concerning such testing problems (see [11–16], among others). The constancy of the functional coefficient $f_j(x_j)$ in varying-coefficient model (3) was studied in Fan and Zhang [17], Cai et al. [18], Fan and Huang [19], You and Zhou [20], and Tang and Cheng [21]. Local polynomials and smoothing spline methods to estimate the coefficients in model (3) can be seen in Hoover et al. [22], Wu et al. [23], and so on.

The critical values of most of the previous tests were obtained by Wilks theorem or bootstrap method. So such tests only behave well in the case of relatively large sample size. This chapter would give some testing procedures based on regression spline and the fiducial method [24] in Section 2. It has a good performance even when the sample size is small.

In using the regression spline, the key problem is the determination of knots used in spline interpolation. As we know that, for smoothing methods such as kernel-based method and smoothing spline, the smoothness is controlled by smoothing parameters. For the well-known kernel estimate, the bandwidth that is extremely big or small might leads to over-smoothing or under-smoothing, respectively. In order to avoid the selection of an optimal smoothing parameter, multi-scale smoothing method was introduced by Chaudhuri and Marron [25, 26] based on kernel estimation for exploring structures in data. This multi-scale method is known as significant zero crossings of derivatives (SiZer) methodology. The basic idea of SiZer is to infer a nonparametric model by using a wide range of smoothing parameter (bandwidth) values rather than only using one "optimal" value in some sense.

There have been many versions of SiZer for various applications, such as the local likelihood version of SiZer in Li and Marron [27], the robust version of SiZer in Hannig and Lee [28], and the quantile version of SiZer in Park et al. [29]. In addition, Marron and deUñaÁlvarez [30] applied SiZer to estimate length biased, censored density, and hazard functions; Kim and Marron [31] utilized SiZer for jump detection and Park and Kang [32] applied SiZer to compare regression curves. The smoothing spline version of SiZer was proposed by Marron and Zhang [33]. It used the tuning parameter (penalty parameter) that controls the size of penalty as the smoothing parameter is.

Comparing with bandwidth for kernel-based method and tuning parameter for smoothing spline, it is more difficult to determine the number of knots and their positions. For this reason a multi-scale smoothing method based on regression spline is proposed in Section 3 to test the structures of nonparametric regression model. The proposed multi-scale method does not involve the determination of the "best" number of knots and can be extended easily to a more general case.

2. Tests for nonparametric function based on regression spline

In this section, the linearity of function $f(x)$ in model (1) is tested based on regression spline and fiducial method. Then, the proposed test procedure for model (1) is extended to test the linearity of model (2) and the constancy of function coefficient in model (3), respectively.

2.1. Test the linearity of nonparametric regression model

Without loss of generality, we suppose that x in model (1) takes values in [0, 1] and the set of knots is $\mathbf{T} = \{0 = t_1 < t_2, \cdots, < t_m = 1\}$. In order to estimate model (1), nonparametric function f (x) is fitted by kth order splines with knots $\mathbf{T}$. This means that

$$f(x) \approx \sum_{j=1}^{m+k-1} \beta_j g_j(x), \tag{4}$$

where β_j is coefficient and $g_j(x),\ j = 1, 2, \cdots, m + k - 1,$ is basis function for order k splines, over the knots $t_1, t_2, \cdots, t_m$.

With n-independent observations $Y = (y_1, y_2, \cdots, y_n) \in \mathbb{R}^n$, the basis matrix $G_{n\times(m+k-1)}$ is defined by $G = \left\{g_j(x_i)\right\}, x_i$ is the designed point, $i = 1, 2, \cdots, n; j = 1, 2, \cdots, m + k - 1.$ Hence, model (1) can be approximated as $Y \approx G\beta + \varepsilon$. The least squares estimator of coefficients is

$$\widehat{\beta} = \left(G^T G\right)^{-1} G^T\, Y, \tag{5}$$

and the estimator of $f(x_i)$ can be expressed as

$$\widehat{Y} = \left\{\widehat{f}(x_1), \widehat{f}(x_2), \cdots, \widehat{f}(x_n)\right\}^T = G\left(G^T G\right)^{-1} G^T\, Y. \tag{6}$$

For testing the linearity of model (1), linear spline is used to approximate $f(x)$. It means that basis function $g_j(x)$ is a linear function:

$$g_1(x) = \frac{-x - t_2}{(t_2 - t_1)\mathbb{I}_2(t)},$$

$$g_{k-1}(x) = \frac{x - t_{k-2}}{(t_{k-1} - t_{k-2})\mathbb{I}_{k-1}(t)} - \frac{x - t_k}{(t_k - t_{k-1})\mathbb{I}_k(t)}, \qquad 3 \le k \le m, \tag{7}$$

$$g_m(x) = \frac{x - t_{m-1}}{(t_m - t_{m-1})\mathbb{I}_m(t)}.$$

In this case, the approximated function in (4) is a linear interpolation with k =1. The true value is $\beta_j = f\left(t_j\right), j = 1, 2, \cdots, m$. The linearity of function $f(x)$ can be written as

$$\mathrm{H}_0 : \frac{\beta_2 - \beta_1}{t_2 - t_1} = \frac{\beta_3 - \beta_2}{t_3 - t_2} = \cdots = \frac{\beta_m - \beta_{m-1}}{t_m - t_{m-1}}.$$

Null hypothesis H_0 can be expressed in matrix as $L'\beta = 0$,

where

$$L' = \begin{bmatrix} h_2 & -h_1 - h_2 & h_1 & 0\cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots\cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0\ \cdots & h_{m-1} & -h_{m-1} - h_{m-2} & h_{m-2} \end{bmatrix},$$

where $h_j = t_{j+1} - t_j, j = 1, 2, \cdots, m-2$. Null hypothesis H_0 is equivalent to the following one:

$$\mathrm{H}_0^* : \; L'\beta = 0. \tag{8}$$

The p-value for testing hypothesis H_0^* will be derived by the fiducial method in the following context. Assume that matrix G has full rank, and let $\varepsilon\tilde{}\sigma N(0, 1)$. In model $Y = G\beta + \varepsilon$, the sufficient statistic of $\left(\beta, \sigma^2\right)$ is $\left(\widehat{\beta}, S^2\right)$, where $\widehat{\beta}$ is defined in (5) and

$$S^2 = Y'(I - P_G)Y, \qquad P_G = G(G'G)^{-1}G'.$$

By Dawid and Stone [34], the sufficient statistic can be represented as a functional model:

$$\widehat{\beta} = \beta + \sigma(G'G)^{-\frac{1}{2}}E_1, \qquad S = \sigma E_2, E = (E_1,\ E_2) \sim Q, \tag{9}$$

where Q is the probability measure of $E = (E_1,\ E_2)$ and $E_1 \sim N(0, I_m)$ and, independently, $E_2^2 \sim \chi^2(n-m)$. From linear regression model, the fiducial model of β can be obtained:

$$\widehat{\beta} = \beta + \frac{S}{E_2}(G'G)^{-\frac{1}{2}}E_1, \qquad E = (E_1,\ E_2) \sim Q. \tag{10}$$

Given $\left(\widehat{\beta}, S^2\right)$, the distribution of the right side in fiducial model is the fiducial distribution of β. That is, the fiducial distribution of β is the conditional distribution of $R\left(E; \widehat{\beta}, S^2\right)$ when $\left(\widehat{\beta}, S^2\right)$ is given, where

$$R\left(E; \widehat{\beta}, S^2\right) = \widehat{\beta} - \frac{S}{E_2}(G'G)^{-\frac{1}{2}}E_1. \tag{11}$$

For testing hypothesis H_0^*, the p-value is defined as

$$p\left(\widehat{\beta}, S^2\right) = Q\left(\left\|L'\left[R\left(E; \widehat{\beta}, S^2\right) - \mathrm{E}_Q R\left(E; \widehat{\beta}, S^2\right)\right]\right\|_{\Sigma}^2 \geq \left\|L'\mathrm{E}_Q R\left(E; \widehat{\beta}, S^2\right)\right\|_{\Sigma}^2\right), \tag{12}$$

where $Q(\cdot)$ and E_Q express the probability for an event and the expectation of a random variable under Q, respectively, and Σ is the conditional covariance matrix of $L'\mathrm{E}_Q R\left(E; \widehat{\beta}, S^2\right)$ given $\widehat{\beta}$, S^2 and $\|v\|_{\Sigma}^2 = v'\Sigma^{-1}v$ for a vector v.

According to the definition of generalized pivotal quantity in [35], $R\left(E; \widehat{\beta}, S^2\right)$ is a generalized pivotal quantity and also a fiducial pivotal quantity about β. Naturally, $L'R\left(E; \widehat{\beta}, S^2\right)$ is the fiducial pivotal quantity about $L'\beta$. With the definition of Q in Eq. (10), we have that

$$p\left(\widehat{\beta}, S^2\right) = 1 - F_{m-2,\,n-m}\left(\frac{(n-m)\widehat{\beta}'L\left(L'(G'G)^{-1}L\right)^{-1}L'\widehat{\beta}}{(m-2)S^2}\right), \tag{13}$$

where $F_{m-2,\,n-m}$ is the cdf of F-distribution with degrees of freedom $m-2$ and $n-m$.

Under model (1) and the hypothesis that $f(x)$ is a linear function, null hypothesis H_0^* given in (8) is true. Suppose that the error is normally distributed, then the p-value given in Eq. (12) distributes as uniform distribution on interval (0, 1). On the other hand, under some mild condition, the test procedure based on $p\left(\widehat{\beta}, S^2\right)$ is consistent. Which means that $p\left(\widehat{\beta}, S^2\right)$ tends to be zero in probability 1 if H_0^* is false. The corresponding theoretical proof of the large sample properties and finite sample properties of $p\left(\widehat{\beta}, S^2\right)$ is the same as the proof given in Li et al. [36].

In applications, we need to check some hypotheses as follows:

$$H_{01}: f(x) = C \Leftrightarrow \beta_1 = \beta_2 = \cdots = \beta_m,$$

$$H_{02}: f(x) = Cx \Leftrightarrow \frac{\beta_2 - \beta_1}{t_2 - t_1} = \frac{\beta_3 - \beta_2}{t_3 - t_2} = \cdots = \frac{\beta_m - \beta_{m-1}}{t_m - t_{m-1}}, \text{and}, \beta_1 = 0.$$

The p-values for testing H_{01} and H_{02} can be obtained by replacing L in (12) by L_{01} and L_{02}, respectively, where $L_{02} = (e_1, L), e_1 = (1, 0, 0, \cdots, 0)'$ and

$$L_{01} = \begin{bmatrix} h_2 & -h_1 & 0\cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0\ \cdots & h_m & -h_{m-1} \end{bmatrix}. \tag{14}$$

2.2. Test the linearity of partial linear model

To test the linearity of model (2), p-value can be established analogously. With n-independent observations $Y = (y_1, y_2, \cdots, y_n) \in \mathbb{R}^n$, model (2) can be represented as.

$$y_i = Z_i'b + f(x_i) + \varepsilon_i, i = 1, 2, \cdots, n,$$

where $Z_i' = (z_{i1}, \cdots, z_{ip})', b = (b_1, \cdots, b_p)', x_i, \quad i = 1, 2, \cdots, n$ are fixed designed points. With the approximation of $f(x)$ given in (4), model (2) can be approximated by $Y \approx X\theta + \varepsilon$, where $X = (\mathbb{Z}, G)_{n\times(p+m+1)}$; $\mathbb{Z} = (z_{ij})$; $i = 1, 2, \cdots, n; j = 1, 2, \cdots, p; G$ is the same as above; and $\theta = (b', , \beta')'$. Then p-value for testing the linearity of model (2) can be defined by replacing G in (12) by X, β by θ, and L by L_{03}, respectively, $L_{03} = (0_{(m-2)\times p}, L')'$.

The large sample and finite sample properties of the testing procedure for model (2) are the same as the test procedure for model (1).

2.3. Test the constancy of functional coefficient in varying-coefficient model

For model (3), investigators often want to know whether the coefficients are really varying; this means to test the constancy of the coefficient functions, that is, testing hypothesis:

$$H_{31}: f_j(x) = C_j \text{ for } j = 1, 2, \cdots, p \text{ and some constant } C_j, \tag{15}$$

$$H_{32}: f_{j0}(x) = C_{j0} \text{ for some } j = j_0 \text{ and some constant } C_{j0}. \tag{16}$$

With the set of knots $\mathbf{T} = \{0 = t_1 < t_2, \cdots, < t_m = 1\}$, coefficient $f_j(x)$ can also be approximated by

$$f_j(x) = \sum_{k=1}^{m} \beta_{js} g_j(x), j = 1, 2, \cdots, p,$$

where the true value of $\beta_{js} = f_j(t_k)$. Basic functions $g_j, j = 1, 2, \cdots, m+1$ were defined in (7). The varying-coefficient model (3) is approximately represented as

$$Y = X\beta + \varepsilon, \tag{17}$$

where $X = (F_1, \cdots, F_p)$ is $n \times mp$ matrix and $F_j = (z_{ji} f_k(x_i)), k = 1, 2, \cdots, m, i = 1, 2, \cdots, n,$ $j = 1, 2, \cdots, p$. $\beta = \left(\beta_1', \cdots, \beta_p{}'\right)'$ is mp-dimensional parametric vector, $\beta_j = \left(f_j(t_1), \cdots, f_p(t_m)\right)'$.

It is worth noting that under null hypothesis H_{31} defined in (15), regression model (3) is equivalent to model (17). However, this equivalence does not hold under null hypothesis H_{32} defined in (16). Null hypotheses H_{31} and H_{32} can be expressed in matrix as the following two, respectively:

$$\mathrm{H}_{31}^* : L_1'\beta = 0, \tag{18}$$

$$\mathrm{H}_{32}^* : L_2'\beta = 0, \tag{19}$$

where L_1' is $p(m-1) \times mp$ matrix.

$$L_1' = \begin{pmatrix} L_{01}{}' & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L_{01}{}' \end{pmatrix}, L_{01} = \begin{bmatrix} 1 & -1 & 0\cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0\ \cdots & 1 & -1 \end{bmatrix},$$

$$L_2' = \left(0_{(m-1)\times\left(mj_0 - m\right)}, L', 0_{(m-1)\times\left(mp - mj_0\right)}\right)_{(m-1)\times mp}.$$

In the same way as the p-value in (13) is defined, p-value to test hypotheses H_{31}^* and H_{32}^* can be defined as below if the error ε distributes as normal distribution:

$$p_{31}\left(\widehat{\beta}, S^2\right) = 1 - F_{p(m-1), n-mp}\left(\frac{(n-mp)\widehat{\beta}' L_1 \left(L_1'(X'X)^{-1} L_1\right)^{-1} L_1' \widehat{\beta}}{p(m-1)S^2}\right), \tag{20}$$

$$p_{32}\left(\widehat{\beta}, S^2\right) = 1 - F_{m-1, n-mp}\left(\frac{(n-mp)\widehat{\beta}' L_2 \left(L_2'(X'X)^{-1} L_2\right)^{-1} L_2' \widehat{\beta}}{(m-1)S^2}\right). \tag{21}$$

According to the above discussion, it can be seen that $p_{31}\left(\widehat{\beta}, S^2\right)$ is uniformly distributed over (0, 1) under hypothesis H_{31}^*. However, under null hypothesis H_{32}^*, varying-coefficient model (2) is not linear. Hence, there is a difference between the distribution function of $p_{32}\left(\widehat{\beta}, S^2\right)$ under H_{32}^* and uniform distribution. This difference has an accurate expression, which can be seen in Li et al. [37] (Theorem 3). On the other hand, $p_{31}\left(\widehat{\beta}, S^2\right)$ and $p_{32}\left(\widehat{\beta}, S^2\right)$ both tend to be zero in probability if null hypotheses are false when sample size tends to be infinity under some mild conditions. The corresponding proof was provided also in Li et al. [37].

3. Multi-scale method based on regression spline

For regression spline, the number of knots controls the smoothness of the estimator. The determination of knots is important and plays a large influence on the inference results. The GCV method is usually used to choose an optimal number of knots. While, but after the number of knots is given, the determination of the optimal positions of knots is difficult. Shi and Li [38] chose knots by placing an additional new knot to reduce the value of GCV, until it could not be reduced by placing any additional knots. Hence, once a knot was selected, it cannot be removed from the knot set. Mao and Zhao [39] determined the locations of knots conditioned on the number of knots m first and chose m later by GCV criterion. In fact, the locations of knots can be considered as parameters which can be estimated from data. This is the free-knot spline; see DiMatteo et al. [40] and Sonderegger and Hannig [41]. However, the estimation of the optimal locations is computationally intractable, and replicate knots might appear in the estimated knot vectors [42].

On the other hand, many statisticians think that the statistical inference based on one smoothing level is not reliable although it is the optimal one. Therefore, multi-scale method is developed to estimate and test nonparametric regression curves. Chaudhuri and Marron [25, 26] proposed a multi-scale method to explore the significant structures (local minima and maxima or global trend) in data, which is known as SiZer. Significant zero crossings of derivatives (SiZer) is a powerful visualization technique for exploratory data analysis. It applies a large range of smoothing parameter values to do statistical inference simultaneously and use a 2D colored map (SiZer map) to summarize all of the results inferred at different smoothing levels (scales) and locations.

In this section, a regression spline version of SiZer is proposed for exploring structures of curve and comparing multiple regression curves, respectively. The proposed SiZer employs the number of knots as smoothing parameter (scales). For the sake of simplicity, linear spline is employed first to construct SiZer, which is denoted as SiZerLS. In addition, another version of SiZer—SiZerSS—is introduced, which is proposed in Marron and Zhang [33]. In SiZerSS, smoothing spline is used to infer the monotonicity of $f(x)$, and the tuning parameter (penalty parameter) that controls the size of penalty is chosen to be as the smoothing parameter. Finally, SiZer-RS, a version of SiZer based on higher-order spline interpolation, is constructed to compare multiple regression curves at different scales and locations simultaneously.

In order to understand SiZerLS clearly, we first present an example in which SiZerLS are simulated. This example is modified from Hannig and Lee [28] with the same regression function:

$$f(x) = 5 + 4.2\left(1 + \frac{|x - 0.3|}{0.03}\right) - 4 + 5.1\left(1 + \frac{|x-0.7|}{0.01}\right) - 4.$$

The observations generated from model (1) with 200 equally spaced design points from (0, 1) and $\sigma \sim N(0, 0.5)$ are plotted in **Figure 1**. Estimator $\widehat{f}_m(x)$ denotes the linear spline smoother obtained from (6) using m equally spaced knots chosen from (0, 1). The curves of $\widehat{f}_m(x)$ with

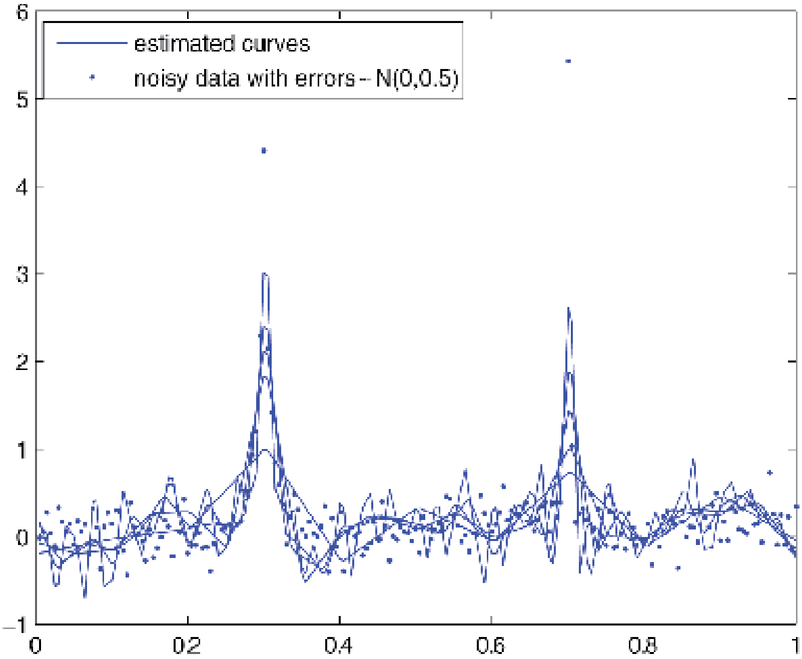


Figure 1. 200 observations and the estimated curves based on different knot sets.

different values of *m* are plotted in **Figure 1** too. The simulated SiZerLS map and SiZerSS map are shown in **Figure 2**, respectively.

In **Figure 2**, BYP SiZerLS is SiZerLS map based on multiple testing procedures, BYP, where BYP denotes the multiple testing procedure proposed in Benjamini and Yekutieli [43]. SiZerSS is the smoothing spline version of SiZer. The two SiZers are simulated under the same range of scales and nominal level 0.05. There are four colors in SiZer maps: red indicates that the

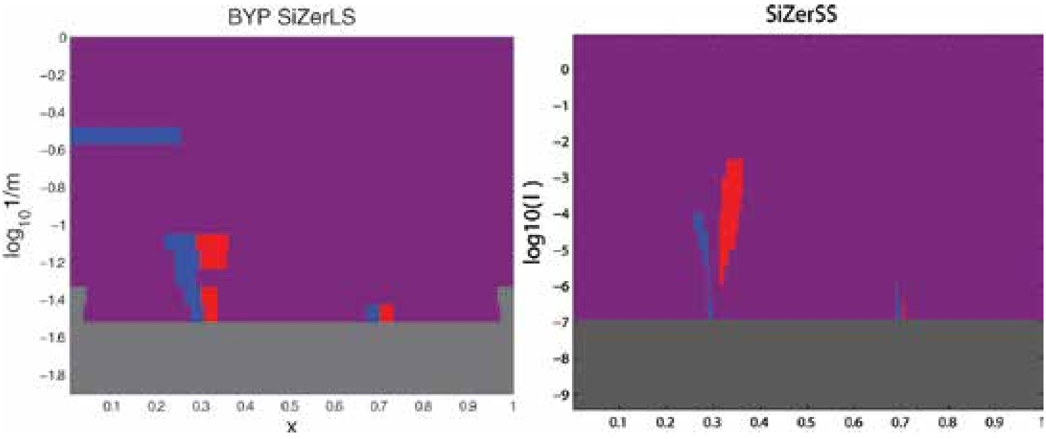


Figure 2. BYP SiZerLS and SiZerSS for detecting peaks of data.

estimated regression curve is significantly decreasing; blue indicates that the estimated regression curve is significantly increasing; purple indicates that the curve is neither significantly increasing nor decreasing; gray shows that there are no sufficient data for conducting reasonable statistical inference. **Figure 1** preliminarily shows that SiZer maps can locate peaks well. The theoretical foundation of SiZerLS and SiZerSS will be discussed in more detail at a later stage.

3.1. Construction of SiZerLS map for exploring features of regression curve

The proposed SiZerLS map will be constructed on the basis of the p-values with multiple testing adjustment. The p-value for testing the monotonicity of the smoothed curve is defined first based on linear spline approximation and fiducial method in the same way as p-values in Section 2. Consequently, multiple testing adjustment is discussed detailedly to control the row-wise false discovery rate (FDR) of SiZerLS.

In the view of SiZer, all of the useful information is included in the smoothed curve, which is defined below. Suppose we have observations $\{x_i, y_i\}_{i=1}^{n}$ from regression model (1). By linear spline estimation, estimator $\widehat{f}_m(x)$ can be obtained:

$$\widehat{f}_m(x) = \boldsymbol{g}(x)'\left(G^T G\right)^{-1} G^T\, Y, \tag{22}$$

where $\boldsymbol{g}(x) = \left(g_1(x), g_2(x), \cdots, g_m(x)\right)'$; $g_j(x), j = 1, \cdots, m$ are the basis functions defined in (7) on the basis of m knots; and G is the matrix defined in Section 2. The smoothed curve at smoothing level m is denoted as.

$$f_m(x) = \mathrm{E}\left(\widehat{f}_m(x)\right) = \boldsymbol{g}(x)'\left(G^T G\right)^{-1} G^T\, \boldsymbol{f},$$

where $\boldsymbol{f} = \{f(x_1), f(x_2), \cdots, f(x_n)\}'$. SiZer focuses on $f_m(x)$. Its monotonicity is determined totally by $\left(G^T G\right)^{-1} G^T\, \boldsymbol{f}$. Hence, it is enough to test the following $m - 1$ pairs of null hypotheses:

$$H_{Ik} = f_m(t_k) = e_k'(G'G)^{-1}G'\boldsymbol{f} \leq e_{k+1}'(G'G)^{-1}G'\boldsymbol{f} = f_m(t_{k+1})(\text{and})$$

$$H_{Dk} = f_m(t_k) = e_k'(G'G)^{-1}G'\boldsymbol{f} \geq e_{k+1}'(G'G)^{-1}G'\boldsymbol{f} = f_m(t_{k+1}), k = 1, 2, \cdots, m - 1, \tag{23}$$

where e_k is an m-dimensional column vector having 1 in the kth entry and zero elsewhere. Let b denote $(G'G)^{-1}G'\boldsymbol{f}$. Then, H_{Ik} and H_{Dk} can be written as

$$H_{Ik}^{*} = L_k b \leq 0, k = 1, 2, \cdots, m - 1; \quad H_{Dk}^{*} = L_k b \geq 0, k = 1, 2, \cdots, m - 1, \tag{24}$$

where $L_k \triangleq \left(e_k' - e_{k+1}'\right)$. The p-values to test hypotheses in (24) under linear model $Y = Gb + \varepsilon$ can be defined using pivotal quantity about b. This pivotal quantity is $R\left(E; \widehat{\beta}, S^2\right)$, which is defined in (11). The p-value for testing H_{Ik}^{*} is the fiducial probability that null hypothesis holds:

$$P^*_{Ik}\left(\widehat{\beta}, S\right) = P\left\{L_k R\left(E; \widehat{\beta}, S\right) \leq 0\right\} = P\left\{L_k \widehat{\beta} - \frac{S}{E_2}(G'G)^{-\frac{1}{2}} E_1 \leq 0\right\}$$

$$= P\left\{\frac{\sqrt{n-m} L_k (G'G)^{-1} G' E_1}{\left(L_k (G'G)^{-1} L'_k\right)^{\frac{1}{2}} E_2} \geq \frac{\sqrt{n-m}\widehat{\beta}}{S\left(L_k (G'G)^{-1} L'_k\right)^{\frac{1}{2}}}\right\}, \tag{25}$$

where the subscript Ik of P^*_{Ik} represents the interval (t_k, t_{k+1}) in which we test monotonicity and m represents the number of knots used in linear interpolation. In addition, p-value $P^*_{Dk}\left(\widehat{\beta}, S\right)$ for testing H^*_{Dk} satisfies equation $P^*_{Ik}\left(\widehat{\beta}, S\right) + P^*_{Dk}\left(\widehat{\beta}, S\right) = 1$.

It is worth noting that p-value $P^*_{Ik}\left(\widehat{\beta}, S\right)$ is uniformly distributed on (0,1) if all of the hypotheses $H_{Ik}, H_{Dk}, k = 1, 2, \cdots, m-1$ are true (regression function is a constant). In applications, p-value $P^*_{Ik}\left(\widehat{\beta}, S\right)$ for testing H_{Ik} can be approximated as below when $n \to \infty$. This approximation is reasonable (see Theorem 1 in [44]):

$$P_{Ik,m}\left(\widehat{\beta}, S\right) \triangleq 1 - \Phi\left(\frac{\sqrt{n-m} L_k \widehat{\beta}}{S\left(L_k (G'G)^{-1} L_k\right)^{1/2}}\right). \tag{26}$$

The proposed SiZerLS map will be constructed on the basis of the above p-values with multiple testing adjustment. In fact, SiZer is a visual method for exploratory data analysis, and it focuses on exploring features that really exist in data instead of testing whether some assumed features are statistically significant in a strict way. FDR is the expected proportion of the false positives among all discoveries, and FDR can be either permissive or conservative according to the number of hypotheses. Considering that different numbers of hypotheses need to be tested for SiZerLS with respect to various smoothing parameters, the multiple testing adjustment to control FDR would be better if used to improve the exploratory property of SiZer. Hence, the well-known multiple testing procedure which was proposed in Benjamini and Yekutieli [43] (denoted as BYP) is applied to control the row-wise FDR of SiZerLS. The BYP was proved to control FDR under α for any dependent test statistics.

3.1.1. Benjamin-Yekutieli procedure to control FDR (BYP)

Suppose that we have obtained p-values $P_{Ik,m}\left(\widehat{\beta}, S\right)$ for testing hypotheses H_{Ik} in (23), $k = 1, 2, \cdots, m-1$:

1. Order p-values $P^*_{Ik,m}$ and get the ordered p-values $P_{I(1),m}, P_{I(2),m}, \cdots, P_{I(m-1),m}$.

2. For a given p-value α, find the largest i for $k = 1, 2, \cdots, m-1$ for which $P_{I(i),m} \leq \frac{k\alpha}{(m-1)\sum_{j=1}^{m-1}\frac{1}{j}}$ and reject all $H_{I(k),m}$ for $k = 1, 2, \cdots, m-1$.

The detailed steps to construct SiZerLS with BYP adjustment are given below:

Step 1. Construct 2D grid map. Without loss of generality, we assume that designed points $x_i, i = 1, 2, \cdots, n$ are chosen from [0, 1]. Then the 2D map is a rectangular area [0, 1; $\log_{10}(1/m_{\max},), \log_{10}(1/m_{\min})$]; see BYP SiZerLS displayed in **Figure 2**. The value of m is determined by the following rule: $m = \text{round}\left(1/10^l\right)$, where function round $(\cdot)$ is the nearest integer function and l takes equally spaced values from interval $\left[\log_{10}(1/m_{\min}), \log_{10}(1/m_{\max})\right]$. For a given m, abscissa x takes values at the corresponding knots $\mathbf{T}_m = \{t_1, t_2, \cdots, t_m\}$. On the basis of different values of m and $\mathbf{T}_m$, the 2D map is divided into many pixels.

Step 2. Calculate p-values for each pixel. Each pixel in the 2D map constructed in step 1 is determined by two adjacent knots and a determined m. For pixel $(t_k, t_{k+1}, m = m_0)$, we calculate p-value P_{Ik,m_0} and P_{Dk,m_0} for testing hypotheses H^*_{Ik,m_0} and H^*_{Dk,m_0}, respectively, with m_0 knots.

Step 3. Multiple testing adjustment. For a given value $m = m_0$, carry out multiple testing procedure BYP using p-values P_{Ik,m_0} (P_{Dk,m_0}), $k = 1, 2, \cdots, m_0$, obtained from step 2 to test the fowling family of hypotheses simultaneously:

$$\left\{H^*_{I1,m_0}, H^*_{I2,m_0}, \cdots, H^*_{Im_0-1,m_0}\right\}\left(H^*_{D1,m_0}, H^*_{D2,m_0}, \cdots, H^*_{Dm_0-1,m_0}\right).$$

Step 4. Color pixels. According to the multiple testing results at smoothing level m_0 if H^*_{Ik} is rejected and H^*_{Dk} is accepted, pixel $(t_k, t_{k+1}, m = m_0)$ is colored red to indicate significant decreasing. On the contrary, if $H^*_{Ik,,m_0}$ is accepted and H^*_{Dk,m_0} rejected, pixel $(t_k, t_{k+1}, m = m_0)$ is colored blue to show significant increasing; purple is used for no significant trend in other cases.

In SiZer map, gray indicates that no sufficient data can be used to test the monotonicity of regression function at point x with m knots. Such sufficiency is quantified as effective sample size (ESS). Noting that the number of nonzero elements in the kth column of G has a demonstrable effect on the inference in interval (t_k, t_{k+1}), and it is determined directly by how many observations are included in (t_k, t_{k+1}), we define $ESS(t_k, m)$ as.

$$(ESS(t_1, m), ESS(t_2, m), \cdots, ESS(t_m, m))' \triangleq G'G(1, 1, \cdots, 1)'.$$

In SiZerLS map, pixel $(t_k, t_{k+1}, m = m_0)$ would be colored gray if.

$$\min(ESS(t_k, m_0), ESS(t_{k+1}, m_0)) < 5.$$

In order to avoid selecting knots, m equally spaced knots or equal x-quantiles are used in interpolation. The smoothing level of regression spline estimate is controlled by m together with the positions of knots. The level of smoothness should be reduced to detect some local fine feature; however, the total number of knots should be limited to avoid excessive under-smoothing in a wide range. In applications of SiZerLS, the range of scales is recommended to include the coarsest smoothing level, $m = 2$, and the finest smoothing level, $\text{avg}_{x \in \mathbf{T}_{m\max}} ESS(x, m_{\max}) < 5$.

3.2. Construction of SiZerSS map for exploring features of regression curve

SiZerSS given in Marron and Zhang [33] employed smoothing spline to construct SiZer map for nonparametric model (1). Given $\{x_i, y_i\}_{i=1}^n$ and a smoothing parameter λ, the smoothing spline estimator is the function $\widehat{f}_\lambda$ that minimizes the regularization criterion over function f:

$$\sum_{i=1}^{n} \omega_i \left[y_i - f(x_i)\right]^2 + \lambda \int \left[f''(x)\right]^2 \mathrm{d}x. \tag{27}$$

By simple calculation, we can get the estimator vector:

$$\widehat{\mathbf{f}}_{\boldsymbol{\lambda}} = \left(\widehat{f}_\lambda(x_1), \widehat{f}_\lambda(x_2), \cdots, \widehat{f}_\lambda(x_n)\right) = (W + \lambda K)^{-1} W Y = A_\lambda Y, \tag{28}$$

where weight matrix $W = \mathrm{diag}(\omega_1, \omega_2, \cdots, \omega_n)$ and the hat matrix $A_\lambda = (W + \lambda K)^{-1} W$.

In order to construct SiZerSS, the derivative of f at any point x needs to be estimated along with its variance. Let $s_i = x_{i+1} - x_i$ and $n \times (n-1)$ matrix $Q = \left\{q_{ij}\right\}, i = 1, 2, \cdots, n,$ $j = 2, \cdots, n-1,$ where $q_{j-1,j} = s_{j-1}^{-1}, q_{jj} = -s_{j-1}^{-1} - s_j^{-1}, q_{j+1,j} = s_j^{-1}$, and $q_{i,j} = 0$ for $|i - j| \geq 2$. Let $(\gamma_1, \gamma_2, \cdots, \gamma_n) = \left(f''(x_1), f''(x_2) \cdots, f''(x_n)\right)$. By the definition of natural cubic spline, $f''(x_1) = f''(x_n) = 0$. Let $\gamma = \left(\gamma_2, \cdots, \gamma_{n-1}\right)'$. According to Theorem 2.1 of Green and Silverman [45], the vectors $\mathbf{f}$ and γ specify a natural cubic spline f if and only if $Q'\mathbf{f} = R\gamma$,

where R is a $(n-2) \times (n-2)$ symmetric matrix with elements $r_{ij}, i = 2, \cdots, n-1,$ $j = 2, \cdots, n-1,$ which is given by $r_{ii} = \frac{1}{3}(s_{i-1} + s_i), r_{i,i+1} = r_{i+1,i} = \frac{1}{6} s_i$ and $r_{ij} = 0$ for $|i - j| \geq 2$. The estimator $\widehat{\gamma}$ can be obtained from equation $(R + \lambda Q'Q)\gamma = Q'Y$. Then estimator $\widehat{f}(x)$ and $\widehat{f}'(x)$ can be written as a linear combination of $\widehat{\mathbf{f}}$ and $\widehat{\gamma}$. Let $h_i(x) = x - x_i, i = 1, 2, \cdots, n$. When $x < x_{1.}$

$$\widehat{f}_\lambda(x) = \widehat{f}_\lambda(x_1) + h_1(x) \left\{ \frac{\widehat{f}_\lambda(x_2) - \widehat{f}_\lambda(x_1)}{s_1} - \frac{s_1}{6} \widehat{\gamma}_2 \right\}, \widehat{f}'(x) = \frac{\widehat{f}_\lambda(x_2) - \widehat{f}_\lambda(x_1)}{s_1} - \frac{s_1}{6} \widehat{\gamma}_2.$$

When $x_i \leq x \leq x_{i+1}$, let $\delta_i(x) = \left(1 + \frac{h_i(x)}{s_i}\right) \widehat{\gamma}_{i+1} + \left(1 - \frac{h_{i+1}(x)}{h_i}\right) \widehat{\gamma}_i$ for $i = 1, 2, \cdots, n,$

$$\widehat{f}_\lambda(x) = \frac{h_i(x) \widehat{f}_\lambda(x_{i+1}) - h_{i+1}(x) \widehat{f}_\lambda(x_i)}{s_i} + \frac{h_i(x) h_{i+1}(x) \delta_i(x)}{6},$$

$$\widehat{f}'_\lambda(x) = \frac{\widehat{f}_\lambda(x_{i+1}) - \widehat{f}_\lambda(x_i)}{s_i} + \frac{h_i(x) h_{i+1}(x) (\widehat{\gamma}_{i+1} - \widehat{\gamma}_i)}{6 s_i} + \frac{h_i(x) + h_{i+1}(x)}{6} \delta_i(x).$$

(When) $x > x_n$

$$\widehat{f}_{\lambda}(x) = \widehat{f}_{\lambda}(x_n) + \frac{h_n(x)}{6}\left\{\frac{\widehat{f}_{\lambda}(x_n) - \widehat{f}_{\lambda}(x_{n-1})}{s_{n-1}} + s_{n-1}\widehat{\gamma}_{n-1}\right\},$$

$$\widehat{f}'_{\lambda}(x) = \frac{1}{6}\left\{\frac{\widehat{f}_{\lambda}(x_n) - \widehat{f}_{\lambda}(x_{n-1})}{s_{n-1}} + s_{n-1}\widehat{\gamma}_{n-1}\right\}.$$

The variance of $\widehat{f}'_{\lambda}(x)$ can be calculated easily if the estimator of σ^2, the variance of the error in model (1), is obtained. σ^2can be estimated by the sum of squared residuals $\sum\left(y_i - \widehat{f}_{\lambda}(x_i)\right)^2$. If σ^2 is a function of x, $\sigma^2(x)$ can be estimated by $\left(y_i - \widehat{f}_{\lambda}(x)\right)^2$. The confidence interval of $f_{\lambda}{}'(x)$ are of the form:

$$\widehat{f}'_{\lambda}(x) \pm q.\widehat{\mathrm{SD}}\left(\widehat{f}_{\lambda}{}'(x)\right), \tag{29}$$

where q is based on the nominal level. For details, see Section 3 of Chaudhuri and Marron [25].

SiZerSS can be constructed as SiZerLS. For different values of x, if interval (29) contains zero, pixel (x, λ) is colored purple; if confidence interval is on the right side of zero, blue is used to indicate increasing; otherwise, red is used to imply decreasing. Gray is used to indicate that there is no sufficient data to do reliable inference. The sufficiency can be found in Chaudhuri and Marron [25].

The simulated SiZerLS and SiZerSS maps are displayed in **Figure 2**, where the red and blue regions locate the bumps of regression curve accurately. This simulation illustrates the good behavior of SiZerLS and SiZerSS in exploring features in data.

3.3. Construction of SiZer-RS map for comparing multiple regression curves

The comparison of two or more populations is a common problem and is of great practical interest in statistics. In this subsection, comparison of multiple regression curves in a general regression setting is developed based on regression spline. Suppose we have $n = \sum_{i=1}^{k} n_i$ independent observations from the following k regression models:

$$y_{ij} = f_i(x_{ij}) + \sigma_i(x_{ij})\varepsilon_{ij}, \quad i = 1, 2, \cdots, k, \quad j = 1, 2, \cdots, n_i, \tag{30}$$

where x_{ij} s are covariates, the errors $\varepsilon_{ij} \sim N(0, 1)$ s are independent and identically distributed errors, $f_i(\cdot)$ is the regression function, and $\sigma_i^2(\cdot)$ is the conditional variance function of the ith population. We are concerned about whether the k populations in model (30) are equal; if not, what is the difference? To this end, a multi-scale method, SiZer-RS, based on regression spline is proposed to compare $f_i(\cdot)$ across multiple scales and locations.

As described in Park and Kang [32], the choice of smoothing parameter is also important for comparing regression curves. They developed SiZer for the comparison of regression curves based on local linear smoother. SiZer map for comparing regression curves is a 2D color map, which consists of a large number of pixels. Each pixel is indexed by a scale (smoothing parameter) and a location; the color of a pixel indicates the result for testing the equality of two or more multiple regression curves at the corresponding location and scale. SiZer provides us with more information about the locations of the differences among the regression curves if they do exist. Park et al. [46] developed an ANOVA-type test statistic and conducted it in scale space for testing the equality of more than two regression curves.

The works mentioned above are kernel-based method. Besides it, regression spline is an important smoothing device and is used widely in applications. For a given smoothing parameter m (the number of knots used in regression spline), the p-value for testing the equality of k regression curves at point x is established. Consequently, SiZer-RS is constructed in the same way as SiZerLS for comparing multiple retrogression curves based on higher-order spline interpolation.

For a given smoothing parameter m (the number of knots used in regression spline), the smoothed curve is defined as $f_{i,m}(x) = \mathrm{E}(\widehat{f}_{i,m}(x))$, where $\widehat{f}_{i,m}(x)$ is the regression spline estimator. SiZer-RS for comparing multiple regression curves is based on the testing results for testing null hypothesis:

$$H_{m,x} : f_{1,m}(x) = f_{2,m}(x) = \cdots = f_{k,m}(x), \tag{31}$$

at point x with smoothing parameter m. Without loss of generality, we still suppose that the explanatory variable x takes value from [0, 1]. On the basis of a knot set $\mathbf{T}_m = \{0 = t_1 < t_2, \cdots, < t_m = 1\}$, we have the approximation:

$$f_i(x) \approx \sum_{s=1}^{m+q-1} \beta_{i,s} g_{m,s}(x) \triangleq N_m(x)' \beta_i^m, \tag{32}$$

where $\beta_i^m = \left(\beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,m+p-1}\right)'$. The estimator of $f_i(x)$ at smoothing level m can be obtained $\widehat{f}_{i,m}(x) = N_m(x)' \widehat{\beta}_i^m$, in which, $N_m(x) = \left\{g_{m,s}(x), s = 1, 2, \cdots, m+q-1\right\}$. If $q = 3$, $N_m(x)'$ is defined below:

$$N_m^l(x) = (t_l - t_{l-4})[t_{l-4}, t_{l-3}, t_{l-2}, t_{l-1}, t_l](t-x)_+^3, \quad l = 2, 3, \cdots, m+3,$$

where $t_l = t_{\min(\max(l,1),m)}$ for $l = -2, -1, \cdots, m+3$:

$$(t-x)_+^3 = \begin{cases} (t-x)^3, t > x \\ 0, \qquad t \leq x \end{cases}.$$

For a function $g(\cdot)$, $[t_{l-4}, t_{l-3}, t_{l-2}, t_{l-1}, t_l]g(\cdot)$ denotes the fourth-order divided difference of $g(\cdot)$, that is:

$$\begin{cases} [t_1, t_2]g = g'(t), \text{ if } \quad t_1 = t_2 = t \\ [t_1, t_2]g = \dfrac{g(t_2) - g(t_1)}{t_2 - t_1} \text{ otherwise,} \\ [t_1, t_2, \cdots, t_k]g = g^{(k-1)}(t), \textit{if } t_1 = \cdots = t_k \\ [t_1, t_2, \cdots, t_k]g = \dfrac{[t_2, t_3, \cdots, t_k]g - [t_1, t_2, \cdots, t_{k-1}]g}{t_k - t_1}, \text{otherwise.} \end{cases}$$

Then model (31) can be approximately written as the following linear regression model:

$$Y_i = G_i^m \beta_i^m + \Sigma_i E_i, \tag{33}$$

where

$$Y_i = \left(y_{i1}, y_{i2}, \cdots, y_{in_i}\right)', G_i^m = \left(N_m^l(x_i)\right)_{n_i \times (m+2)'} \ \Sigma_i = \text{diag}\left\{\sigma_i\left(x_{ij}\right)\right\}, E_i = \left(\varepsilon_{i1}, \varepsilon_{i2}, \cdots, \varepsilon_{in_i}\right)'.$$

At first, we suppose Σ_i is known and then replace it by its available estimator.

From regression model (33), we can get the estimator $\widehat{\beta}_i^m = \left(G_i^{m\prime}\Sigma_i^{-1}G_i^m\right)^{-1}G_i^{m\prime}\Sigma_i^{-1}Y_i$. Let b_i^m denote the expectation of $\widehat{\beta}_i^m$:

$$\mathrm{b}_i^m = \mathrm{E}\left(\widehat{\beta}_i^m\right) = \left(G_i^{m\prime}\Sigma_i^{-1}G_i^m\right)^{-1}G_i^{m\prime}\Sigma_i^{-1}\mathbf{f}_i,$$

where $\mathbf{f}_i = \left(f_i(x_{i1}), \cdots, f_i(x_{in_i})\right)'$. Therefore, the smoothed curve

$$f_{i,m}(x) = \mathrm{E}\left(\widehat{f}_{i,m}(x)\right) = \mathrm{E}\left[N_m(x)'\left(G_i^{m\prime}\Sigma_i^{-1}G_i^m\right)^{-1}G_i^{m\prime}\Sigma_i^{-1}Y_i\right] = N_m(x)'\mathrm{b}_i^m. \tag{34}$$

Denote $\mathbf{b}^m = \left(\mathrm{b}_1^{m\prime}, \mathrm{b}_2^{m\prime}, \cdots, \mathrm{b}_k^{m\prime}\right)'$, and correspondingly, denote its estimator as $\widehat{\beta}^m = \left(\beta_1^{m\prime}, \beta_2^{m\prime}, \cdots \beta_k^{m\prime}\right)'$. Hypothesis $H_{m,x}$ can be presented as

$$H_{m,x}: \quad L_m(x)\mathbf{b}^m = 0_{k-1}, \tag{35}$$

where

$$L_m(x) = \begin{bmatrix} N_m(x) & N_m(x) & N_m(x) & & N_m(x) \\ -N_m(x) & 0 & 0 & \cdots & 0 \\ 0 & -N_m(x) & 0 & & 0 \\ \cdots & \cdots & \cdots & & \cdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & -N_m(x) \end{bmatrix}$$

is a $(k-1) \times k(m+q-1)$ matrix.

The p-value for testing hypothesis $H_{m,x}$ in (35) can be defined as

$$p_{m,x}\left(\widehat{\beta}_i^m, \widehat{\Sigma}_m\right) = P\left\{T^m(x)' L_m(x)' \left[L_m(x)\widehat{\Sigma}_m L_m(x)'\right]^{-1} L_m(x) T^m(x) \right.$$
$$\left. \geq \widehat{\beta}_i^{m\prime} L_m(x)' \left[L_m(x)\widehat{\Sigma}_m L_m(x)'\right]^{-1} L_m(x)\widehat{\beta}_i^m \right\}, \tag{36}$$

where $T^m(x) \triangleq \left\{\left(G_i^{m\prime}\widehat{\Sigma}_{i,m}^{-1} G_i^m\right)^{-1} G_i^{m\prime}\widehat{\Sigma}_{i,m}^{-\frac{1}{2}} E_i\right\}', i = 1, 2, \cdots, k; \widehat{\Sigma}_{i,m} = \mathrm{diag}\{\widehat{\sigma}_i(x_{ij}), j = 1, 2, \cdots, n_i\}$ is an estimator of the variance matrix of the ith regression model and

$$\widehat{\Sigma}_m = \mathrm{diag}\left\{\left(G_i^{m\prime}\widehat{\Sigma}_{i,m}^{-1} G_i^m\right)^{-1}, i = 1, 2, \cdots, k\right\}$$

is an estimator of the variance matrix of $T^m(x)$ given $\widehat{\beta}_i^m, \widehat{\sigma}_i^{m2}, i = 1, 2, \cdots, k$. The estimator of $\sigma_i(x_{ij})$can be found in Li and Xu [36], where the smoothing parameter, m_p, can be used as a pilot smoothing parameter, which is different from m used in $\widehat{f}_{i,m}(x)$. SiZer-RS map can be constructed based on different values of m_p, which represents the different trade-offs between the structure of regression curve and errors.

The two SiZer maps given in **Figure 4** are constructed using the data plotted in **Figure 3** to compare three regression curves $f_1(x) = f_x(x) = 0, f_3(x) = 0.5\sin(2\pi x)$. Since the variance of errors is a constant, it can be estimated by the sum of squares of residues. In this case, pilot smoothing parameter is avoided [47, 48]. The two blue regions in **Figure 4** clearly show their difference across interval (0, 1). The gray color indicates that there is no sufficient data that can be used to get credible testing results at x and nearby. The sufficiency is quantized as $ESS(x, m)$ for SiZer-RS, and pixel (x, m) is colored gray if $ESS(x, m) < 5$:

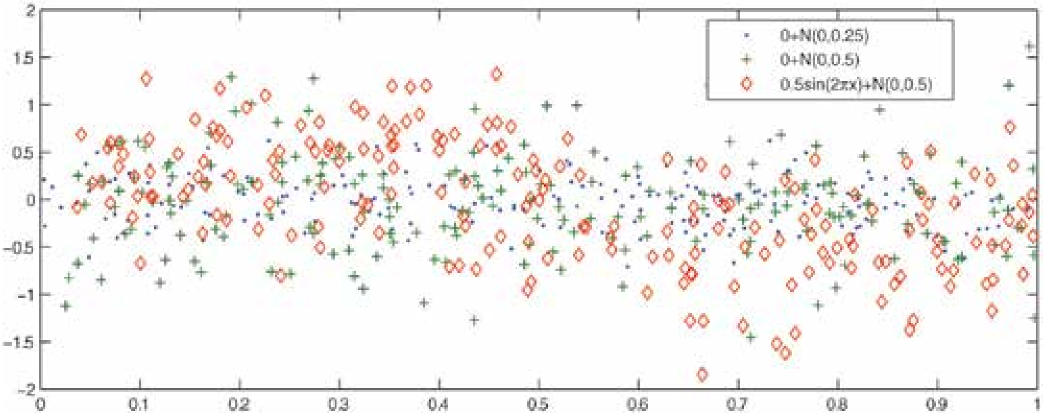


Figure 3. 200 observations.

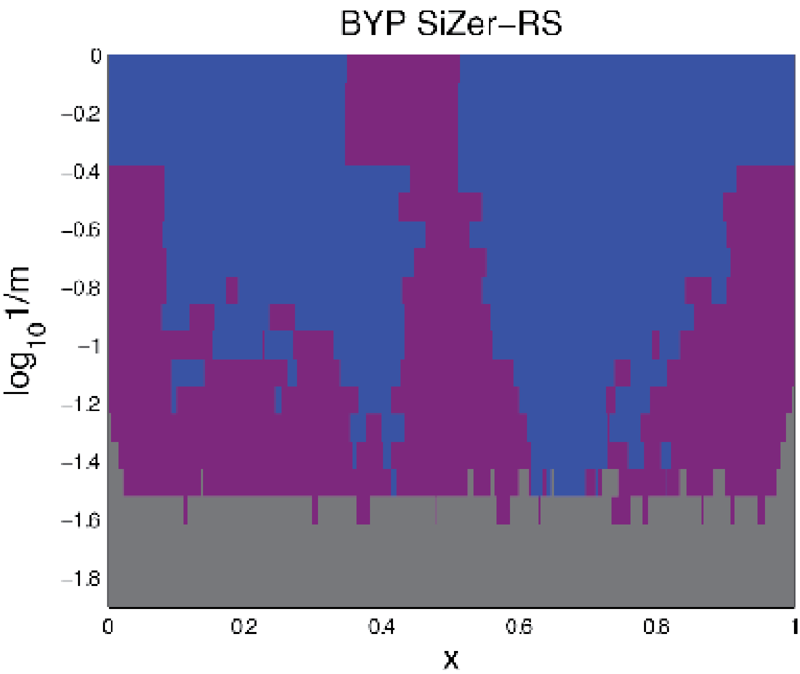


Figure 4. SiZer-RS map.

$$ESS(x, m) \triangleq \min_{i=1,2,\cdots,k} \left\{ N_m(x) G_i^{m\prime} G_i^m (1, 1, \cdots, 1)' \right\}.$$

Figure 4 shows that SiZer-RS map can explore the differences between regression curves accurately.

It is worth noting that, for SiZer-RS map, the coarsest smoothing level should be $m = q + 1$ to ensure the effectiveness of the qth regression spline and the finest smoothing level is recommend to be the one such that $\text{avg}_{x \in [x_1, x_2, \cdots, x_g]} ESS(x, m) < 5$, where $x_1, x_2, \cdots, x_g$ are points at which hypothesis $H_{m,x}$ is tested and pixels are produced by combing different values of m. In applications, a wide range of values of m_p can be used to generate a family of SiZer-RS maps. Particularly, m_p and m can both be used as smoothing parameters simultaneously to construct a 3D SiZer-RS map [47, 48].

4. Conclusion

This chapter introduces regression spline method for testing the parametric form of nonparametric regression function in nonparametric, partial linear, and varying-coefficient models, respectively. The corresponded p-values are established based on fiducial method and spline interpolation. The test procedures on the basis of the proposed p-value are accurate in some cases and are consistent under some mild conditions, which means that the p-value tends to be zero when null hypothesis is false as sample size and the number of knots used in spline interpolation tend to be infinity. Hence, the proposed test procedures are performed well especially in small sample size case.

The spline-based method frequently used smoothing method, which can be used easily with other statistical methods. When using the spline-based method, the smoothing level is controlled by the number of knots and their positions. In order to sidestep the determination of knots and obtain more reliable results, multi-scale smoothing methods are proposed based on spline regression to infer structures of regression function. The multi-scale method is a visual method to do inference at different locations and smoothing levels. In addition, the smoothing spline version of multi-scale method is also introduced. The proposed multi-scale method can also be used for comparing multiple regression curves. Some real data examples illustrate the practicability of the proposed multi-scale method.

The MATLAB code of SiZerLL and other versions of SiZer based on kernel smoother is available from the homepage of Professor Marron JS; the MATLAB code of SiZerLS can be downloaded from the following website:

http://www.tandfonline.com/doi/suppl/10.1080/10618600.2014.1001069?scroll=top.

Author details

Na Li

Address all correspondence to: nali@amss.ac.cn

School of Econometrics and Management, University of the Chinese Academy of Sciences, Beijing, China

References

[1] Härdle W, Mammen E. Comparing nonparametric regression fits. Annals of Statistics. 1993;**21**:1926-1947

[2] Hart JD. Nonparametric Smoothing and Lack-of-Fit Test. New York: Springer; 1997

[3] Cox D, Koh E, Wahba G, Yandell BS. Testing the (parametric) null model hypothesis in (semiparametric) partial and generalized spline models. Annals of Statistics. 1988;**16**:113-119

[4] Cox D, Koh EA. Smoothing spline based test of model adequacy in polynomial regression. Annals of the Institute of Statistical Mathematics. 1989;**41**:383-400

[5] Fan J, Zhang C, Zhang J. Generalized likelihood ratio statistics and Wilks phenomenon. Annals of Statistics. 2001;**29**:153-193

[6] Eubank RL, Hart JD. Testing goodness-of-fit in regression via order selection criteria. Annals of Statistics. 1992;**20**:1412-1425

[7] Baraud Y. Model selection for regression on a fixed design. Probability Theory and Related Fields. 2000;**117**:467-493

[8] Bianco A, Boente G. Robust estimators in semi-parametric partly linear regression models. Journal of Statistical Planning and Inference. 2004;**22**:229-252

[9] Liang H, Wang S, Robin JM, Carroll RJ. Estimation in partially linear models with missing covariates. Journal of the American Mathematical Society. 2004;**99**:357-367

[10] Fan J, Huang L. Goodness-of-fit tests for parametric regression models. Journal of the American Mathematical Society. 2001;**96**:640-652

[11] Baraud Y, Huet S, Laurent B. Adaptive tests of linear hypotheses by model selection. Annals of Statistics. 2003;**31**:225-251

[12] Claeskens G. Restricted likelihood ratio lack-of-fit tests using mixed spline models. Journal of the Royal Statistical Society: Series B. 2004;**66**:909-926

[13] Härdel W, Sperlich S, Spokoiny V. Structural tests in additive regression. Journal of the American Mathematical Society. 2001;**96**:1333-1347

[14] Meyer MC. A test for linear versus convex regression function using shape-restricted regression. Biometrika. 2003;**90**:223-232

[15] Stute W. Nonparametric model checks for regression. Annals of Statistics. 1997;**5**:613-641

[16] Liang H. Checking linearity of non-parametric component in partially linear models with an application in systemic inflammatory response syndrome study. Statistical Methods in Medical Research. 2006;**15**:273-284

[17] Fan J, Zhang W. Simultaneous confidence bands and hypothesis testing in varying-coefficient models. Scandinavian Journal of Statistics. 2000;**27**:715-731

[18] Cai Z, Fan J, Li R. Efficient estimation and inferences for varying-coefficient models. Journal of the American Statistical Association. 2000;**95**:888-902

[19] Fan J, Huang T. Profile likelihood inferences on semi-parametric varying-coefficient partially linear models. Bernoulli. 2005;**11**:1031-1057

[20] You J, Zhou Y. Empirical likelihood for semi-parametric varying-coefficient partially linear regression models. Statistics and Probability Letters. 2006;**76**:412-422

[21] Tang Q, Cheng L. M-estimation and B-spline approximation for varying coefficient models with longitudinal data. Journal of Nonparametric Statistics. 2008;**20**:611-625

[22] Hoover DR, Rice JA, Wu C, Yang LP. Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data. Biometrika. 1998;**85**:809-822

[23] Wu C, Yu K, Chiang CT. A two-step smoothing method for varying coefficient models with repeated measurements. Annals of Statistics. 2000;**52**:519-543

[24] Xu X, Li G. Fiducial inference in the pivotal family of distributions. Science in China, Series A: Mathematics. 2006;**49**:410-432

[25] Chaudhuri P, Marron JS. SiZer for exploration of structures in curves. Journal of the American Statistical Association. 1999;**94**:807-823

[26] Chaudhuri P, Marron JS. Scale space view of curve estimation. Annals of Statistics. 2000; **28**:408-428

[27] Li R, Marron JS. Local likelihood SiZer map. The Indian Journal of Statistics. 2005;**67**:476-498

[28] Hannig J, Lee TCM. Robust SiZer for exploration of regression structures and outlier detection. Journal of Computational and Graphical Statistics. 2006;**15**:101-117

[29] Park C, Lee TCM, Hannig J. Multiscale exploratory analysis of regression quantiles using quantile SiZer. Journal of Computational and Graphical Statistics. 2010;**3**:497-513

[30] Marron JS, deŨnáAlvarez J. SiZer for length biased, censored density and hazard estimation. Journal of Statistical Planning and Inference. 2004;**121**:149-161

[31] Kim CS, Marron JS. SiZer for jump detection. Journal of Nonparametric Statistics. 2006;**18**: 13-20

[32] Park C, Kang KH. SiZer analysis for the comparison of regression curves. Computational Statistics and Data Analysis. 2008;**52**:3954-3970

[33] Marron JS, Zhang JT. SiZer for smoothing spline. Computational Statistics. 2005;**20**:481-502

[34] Dawid AP, Stone M. The functional model basis of fiducial inference. Annals of Statistics. 1982;**10**:1054-1067

[35] Hannig J, Iyer H, Patterson P. Fiducial generalized confidence intervals. Journal of the American Mathematical Society. 2006;**101**:254-269

[36] Li N, Xu X, Jin P. Testing the linearity in partially linear models. Journal of Nonparametric Statistics. 2011;**23**:99-114

[37] Li N, Xu X, Liu X. Testing the constancy in varying-coefficient regression models. Metrika. 2011;**74**:409-438

[38] Shi P, Li G. Global convergence rates of B-spline M-estimators in nonparametric regression. Statistica Sinica. 1995;**5**:303-318

[39] Mao W, Zhao LH. Free-knot polynomial splines with confidence intervals. Journal of the Royal Statistical Society: Series B. 2003;**65**:901-909

[40] DiMatteo I, Genovese CR, Kass RE. Bayesian curve-fitting with free-knot splines. Biometrika. 2001;**88**:1055-1071

[41] Sonderegger DL, Hannig J. Fiducial Theory for Free-Knot Splines. Vol. 68. New York: Springer International Publishing; 2014. pp. 155-189

[42] Lindstrom MJ. Penalized estimation of free-knot splines. Journal of Computational and Graphical Statistics. 1999;**8**:333-352

[43] Benjamini Y, Yekutieli D. The control of the false discovery rate in multiple testing under dependency. Annals of Statistics. 2001;**29**:1165-1188

[44] Li N, Xu X. Spline multiscale smoothing to control FDR for exploring features of regression curves. Journal of Computational and Graphical Statistics. 2016;**25**:325-343

[45] Green PJ, Silverman BW. Nonparametric Regression and Generalized Linear Models. London: Chapman and Hall; 1994

[46] Park C, Hannig J, Kang KH. Nonparametric comparison of multiple regression curves in scale-space. Journal of Computational and Graphical Statistics. 2014;**23**:657-677

[47] Hannig J, Marron JS. Advanced distribution theory for SiZer. Journal of the American Statistical Association. 2006;**101**:484-499

[48] Weerahandi S. Generalized confidence intervals. Journal of the American Statistical Association. 1993;**88**:899-905

Chapter 6

Penalized Spline Joint Models for Longitudinal and Time-To-Event Data

Huong Thi Thu Pham and Hoa Pham

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.75975

Abstract

The joint models for longitudinal data and time-to-event data have recently received numerous attention in clinical and epidemiologic studies. Our interest is in modeling the relationship between event time outcomes and internal time-dependent covariates. In practice, the longitudinal responses often show non-linear and fluctuated curves. Therefore, the main aim of this chapter is to use penalized splines with a truncated polynomial basis to parameterize the non-linear longitudinal process. Then, the linear mixed effects model is applied to subject-specific curves and to control the smoothing. The association between the dropout process and longitudinal outcomes is modeled through a proportional hazard model. Two types of baseline risk functions are considered, namely a Gompertz distribution and a piecewise constant model. The resulting models are referred to as penalized spline joint models; an extension of the standard linear joint models.

Keywords: survival data, longitudinal data, joint models, time-dependent covariates, random effects

1. Introduction

The joint models for longitudinal data and time-to-event data are aimed to measure the association between the longitudinal marker level and the hazard rate for an event. The longitudinal data are collected repeatedly for several subjects. In this data, there are two types of covariates, namely, time-independent covariates and time-dependent covariates. Furthermore, there are also two different categories of time-dependent covariates, namely, external and internal covariates. In clinical studies, internal time-dependent longitudinal outcomes are often applied to monitor disease progression and failure time.

In modern survival analysis, Cox [1] has been considered as a very popular joint model to be used for time-independent covariates. These models measured the effect of time-independent covariates on the hazard rate for an event. Subsequently, the extended Cox model was developed for external time-dependent covariates. However, these latter models cannot handle longitudinal biomarkers. Therefore, Rizopoulos [2] introduced joint models for internal time-dependent covariates and the risk for an event based on linear mixed-effects models and relative risk models.

The basic assumption for the standard joint models proposed by Rizopoulos [2] is that the hazard rate at a given time of the dropout process is associated with the expected value of the longitudinal responses at the same time. The whole history of response has an influence on the survival function. Thus, it is crucial to obtain good estimates for the subject-specific trajectories in order to have an accurate estimation of the survival function. In addition, an important feature that we need to account for is that many observations in the sample often show non-linear and fluctuated longitudinal trajectories in time. Each observation has its own trajectory. Therefore, flexibility is needed for subject-specific longitudinal submodels in the joint models to improve the predictions.

There are several previous works to flexibly model the subject-specific longitudinal profiles in the joint models. Brown et al. [3] applied B-splines with multidimensional random effects. In particular, Brown et al. [3] assumed that both subject and population trajectories have the same number of basis functions. By doing this, the number of parameters in the longitudinal submodel is reasonably large. If we have to deal with the roughness of the fit for this model, the computational problems will increase especially when the dimension of the random effects vector is large. Ding and Wang [4] proposed the use of B-splines with a single multiplicative random effect to link the population mean function with the subject-specific profile. This simple model can gain an easy estimation for parameters, however may not be appropriate for many practical applications [5]. Rizopoulos [5] considered more flexible models using natural cubic splines with the expansion of the random effects vector. The roughness of the fit is still not mentioned in these models.

In this chapter, we present new approaches to model non-linear shapes of subjects-specific evolutions for joint models by extending the standard joint models of Rizopoulos [2]. In particular, we implement penalized splines using a truncated polynomial basis for the longitudinal submodel. Following this, the linear mixed-effects approach is applied to model the individual trajectories and impose smoothness over adjacent coefficients respectively. The ECM algorithm is used for parameter estimation. In addition, corresponding standard errors are calculated using the observed information matrix. However, as the matrices of random effects covariates in our models are different from the matrices of random effects covariates in the standard joint models, the JM package of Rizopoulos [6] cannot be used for our models. Therefore, a set of R codes are written for the penalized spline joint models to implement the proposed procedures on the simulated data and a case study respectively.

The chapter is organized as follows. Section 2 describes the penalized splines with truncated polynomial basis for the joint models. In this section, the two models are specified as penalized spline joint model with hazard rate at base line having Gompertz distribution (referred to as

Model 1) and penalized spline joint model with a piecewise-constant baseline risk function (referred to as Model 2). The joint likelihood, score functions and the ECM algorithm for estimation are presented in Section 3. We then validate the proposed algorithm using extensive simulation studies and then apply it for AIDS data in Section 4. Finally, Section 5 gives concluding remarks.

2. The penalized spline joint models

In this section, we introduce the joint models using penalized spline with truncated polynomial basis. The proposed parametrization is based on the standard joint models of Rizopoulos [2] and the regression model of a longitudinal response using penalized spline.

Notations in this section are taken from Rizopoulos [2]. Let T_i^* be the true survival time and C_i be the censoring time for the i^{th} subject $(i = 1, \ldots, n)$. T_i denotes the observed failure time for the i^{th} subject $(i = 1, \ldots, n)$, which is defined as $T_i = min\left(T_i^*, C_i\right)$. If an i^{th} subject is not censored, this means that we have observed its survival time, we will have $T_i \leq C_i$. If an i^{th} subject is censored, this means that we lose its follow up, or the subject has died from other causes, we will have $T_i > C_i$. Furthermore, we define the event indicator as $\delta_i = I\left(T_i^* \leq C_i\right)$. The observed data for survival outcome are $(T_i, \delta_i), i = 1, \ldots, n$.

For a longitudinal response, suppose that we have n subjects in the sample and the actual observed longitudinal data for each subject-i at time point t is $y_i(t)$. We measure the i^{th} subject at n_i time points. Thus, the longitudinal data consists of the measurements $y_{ij} = \left\{y_i\left(t_{ij}\right), j = 1, \ldots, n_i\right\}$ taken at time points t_{ij}. We denote the true and unobserved value of the longitudinal outcome at time t as $m_i(t)$. We assume the relation between $y_i(t)$ and $m_i(t)$ as

$$y_i(t) = m_i(t) + \varepsilon_i(t), \tag{1}$$

where $\varepsilon_i(t) \sim N\left(0, \sigma_\varepsilon^2\right)$.

When survival function $S(t)$ is assumed to have a specific parametric form associating with a longitudinal submodel, estimations for parameters of interest are usually based on the likelihood function [2]. In the maximum likelihood method, there are different treatments for different types of covariates in the longitudinal submodel. Here, we present the two different categories of time-dependent covariates and the estimation techniques for these covariates will be introduced in the following sections. We let the time-dependent covariate for the i^{th} subject at time t be denoted by $y_i(t)$. We let $\mathcal{Y}_i(t) = \left\{y_i(s), 0 \leq s < t\right\}$ denote the covariate history of the i^{th} subject up to time t. According to Kalbfleisch and Prentice [7], the exogenous covariates are the covariates satisfying the condition:

$$\Pr(s \leq T_i < s + ds | T_i \geq s, \mathcal{Y}_i(s)) = \Pr(s \leq T_i < s + ds | T_i \geq s, \mathcal{Y}_i(t)), \tag{2}$$

for all s, t such that $0 < s \leq t$ and $ds \rightarrow 0$. An equivalent definition is

$$\Pr(\mathcal{Y}_i(t)|\mathcal{Y}_i(s), T_i \geq s) = \Pr(\mathcal{Y}_i(t)|\mathcal{Y}_i(s), T_i = s), s \leq t. \tag{3}$$

On the other hand, endogenous time-varying covariates are the ones that do not satisfy the condition in (2.2). In particular,

$$\Pr(\mathcal{Y}_i(t)|\mathcal{Y}_i(s), T_i \geq s) \neq \Pr(\mathcal{Y}_i(t)|\mathcal{Y}_i(s), T_i = s), s \leq t.$$

In the penalized spline regression models [8, 9], the observed longitudinal covariate is modeled using the truncated power functions with a general power basis of degree p. Moreover, the longitudinal response is also parameterized as a linear mixed-effects model to specify the individual curves and impose the amount of smoothing. As a result, the coefficients of the knots will be constrained to handle smoothing. In particular, the longitudinal submodel for the i^{th} subject at time point t_{ij} is

$$\begin{aligned} y_{ij} &= f(t_{ij}) + g_i(t_{ij}) + \varepsilon(t_{ij}), \varepsilon_i(t_{ij}) \sim N(0, \sigma_\varepsilon^2), \\ f(t_{ij}) &= \beta_0 + \beta_1 t_{ij} + \ldots + \beta_p t_{ij}^p + \sum_{k=1}^{K} u_{pk}(t_{ij} - \mathcal{K}_k)_+^p, \\ g_i(t_{ij}) &= v_{i0} + v_{i1} t_{ij} + v_{i2} t_{ij}^2 + \ldots + v_{ip} t_{ij}^p + \sum_{k=1}^{K} w_{ipk}(t_{ij} - \mathcal{K}_k)_+^p. \end{aligned} \tag{4}$$

Here, the set $\left\{1, t_{ij}, \ldots, t_{ij}^p, (t_{ij} - \mathcal{K}_1)_+^p, \ldots, (t_{ij} - \mathcal{K}_K)_+^p\right\}$ is known as the truncated power basis of degree p. Moreover, $\mathcal{K}_1, \ldots, \mathcal{K}_K$ are fitted K knots, for which K is chosen following Ruppert et al. [9], Chapter 5), Appendix D. The function $f(.)$ is the smooth function which reflects the overall trend of the population. The function $g_i(.)$ is the smooth function which reflects the individual curves. To constrain the coefficient of knots, the vector $(u_{p1}, \ldots, u_{pK})^T$ in the function $f(.)$ is treated as random effects. Therefore, $\beta^T = (\beta_0, \ldots, \beta_p)$ is a $((p+1) \times 1)$ row vector of fixed effects and $\boldsymbol{b}_i^T = (u_{p1}, \ldots, u_{pK}, v_{i0}, \ldots, v_{ip}, w_{ip1}, \ldots, w_{ipK})$ is a $((p+2K+1) \times 1)$ vector of random effects for the i^{th} subject. The assumptions for the random effects for the i^{th} subject are $(v_{i0}, \ldots, v_{ip})^T \sim N(0, \sum)$, $u_{pk} \sim N(0, \sigma_u^2)$, $w_{ipk} \sim N(0, \sigma_w^2)$ and they are independent of one another. We can now rewrite (2.4) as

$$\begin{aligned} y_i(t_{ij}) &= f(t_{ij}) + g_i(t_{ij}) + \varepsilon_i(t_{ij}) \\ &= \beta_0 + \beta_1 t_{ij} + \beta_2 t_{ij}^2 + \ldots + \beta_p t_{ij}^p + \sum_{k=1}^{K} (u_{pk} + w_{ipk})(t_{ij} - \mathcal{K}_k)_+^p \\ &\quad + v_{i0} + v_{i1} t_{ij} + v_{i2} t_{ij}^2 + \ldots + v_{ip} t_{ij}^p + \varepsilon_i(t_{ij}). \end{aligned} \tag{5}$$

We let $u_{ipk} = u_{pk} + w_{ipk}$ and note that $u_{ipk} \sim N(0, \sigma_u^2 + \sigma_w^2)$. In order to allow greater flexibility, we assume that $(u_{ip1}, \ldots, u_{ipK})^T \sim N(0, \boldsymbol{D})$, where $\boldsymbol{D} = Diag(D_{11}, \ldots, D_{KK})$. By doing this, the dimension of the vector of random effects, $\boldsymbol{b}_i^T = (v_{i0}, \ldots, v_{ip}, u_{ip1}, \ldots, u_{ipK})$, decreases to $((p+K+1) \times 1)$. Consequently, the dimension of the multi-integrals in the log-likelihood function in (3.2) will also decrease. This presentation is crucial for reducing the computational problems while coding. The model in (2.5) now becomes:

$$\begin{aligned} y_i(t_{ij}) &= f(t_{ij}) + g_i(t_{ij}) + \varepsilon_i(t_{ij}) \\ &= \beta_0 + \beta_1 t_{ij} + \beta_2 t_{ij}^2 + \ldots + \beta_p t_{ij}^p + \sum_{k=1}^{K} u_{ipk}(t_{ij} - \mathcal{K}_k)_+^p \\ &\quad + v_{i0} + v_{i1} t_{ij} + v_{i2} t_{ij}^2 + \ldots + v_{ip} t_{ij}^p + \varepsilon_i(t_{ij}). \end{aligned} \tag{6}$$

The model in (2.6) can be rewritten in matrix notation as:

$$\boldsymbol{y} = \boldsymbol{X}\beta + \boldsymbol{Z}\boldsymbol{b} + \varepsilon, \tag{7}$$

where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n \end{bmatrix}, \quad \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X}_1 & 0 & \ldots & 0 & \boldsymbol{Z}_1 & 0 & \ldots & 0 \\ 0 & \boldsymbol{X}_2 & \ldots & 0 & 0 & \boldsymbol{Z}_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{X}_n & 0 & 0 & \ldots & \boldsymbol{Z}_n \end{bmatrix},$$

$$\boldsymbol{X}_i = \begin{bmatrix} 1 & t_{i1} & t_{i1}^2 & \cdots & t_{i1}^p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & t_{in_i} & t_{in_i}^2 & \cdots & t_{in_i}^p \end{bmatrix}, \quad \boldsymbol{Z}_i = \begin{bmatrix} (t_{i1} - \mathcal{K}_1)_+^p & \cdots & (t_{i1} - \mathcal{K}_K)_+^p \\ \vdots & \vdots & \vdots \\ (t_{in_i} - \mathcal{K}_1)_+^p & \cdots & (t_{in_i} - \mathcal{K}_K)_+^p \end{bmatrix},$$

$$\begin{aligned} \boldsymbol{b}^T &= (v_{10}, \ldots, v_{1p}, \ldots, v_{n0}, \ldots, v_{np}, u_{1p1}, \ldots, u_{1pK}, \ldots, u_{np1}, \ldots, u_{npK}), \\ \beta^T &= (\beta_0, \ldots, \beta_p). \end{aligned}$$

Here, $\boldsymbol{y}$ is the $\left(\sum_{i=1}^{n} n_i \times 1\right)$ matrix of observed longitudinal data; $\boldsymbol{X}$ is the $\left(\sum_{i=1}^{n} n_i \times (p+1)\right)$ matrix of fixed effect covariates; $\boldsymbol{Z}$ is the $\left(\sum_{i=1}^{n} n_i \times (p+K+1)n\right)$ matrix of random effect covariates and ε is the $\left(\sum_{i=1}^{n} n_i \times 1\right)$ matrix of error.

Postulating a proportion hazard model, the penalized spline joint models for longitudinal and time-to-event data is defined by

$$\begin{aligned} h_i(t|\mathcal{M}_i(t), \boldsymbol{w}_i) &= \lim_{dt \to 0} \Pr\{t \le T_i^* < t + dt | T_i^* \ge t, \mathcal{M}_i(t), \boldsymbol{w}_i\}/dt \\ &= h_0(t) \exp\{\gamma^T \boldsymbol{w}_i + \alpha m_i(t)\}, \end{aligned} \tag{8}$$

where $h_0(t)$ is the hazard at baseline and $\boldsymbol{w}_i$ is a vector of baseline covariates (such as treatment indicator, gender of a patient, etc.). Furthermore, $\mathcal{M}_i(t) = \{m_i(s),\ 0 \le s < t\}$ denotes the history of the true unobserved longitudinal process up to time point t.

Using (2.7), the longitudinal submodel for the i^{th} subject is given by

$$\begin{cases} m_i(t) = m_i(t) + \varepsilon_i(t), \varepsilon_i(t) \sim N\left(0, \sigma_\varepsilon^2\right) \\ y_i(t) = \boldsymbol{X}_i^T(t)\beta + \boldsymbol{X}_i^T(t)\boldsymbol{v}_i + \boldsymbol{Z}_i^T(t)\boldsymbol{u}_i + \varepsilon_i(t) \\ \quad \boldsymbol{v}_i \sim N\left(\boldsymbol{0}, \sum\right), \boldsymbol{u}_i \sim N(\boldsymbol{0}, \boldsymbol{D}), \end{cases} \tag{9}$$

where the covariance matrix of random effects $\boldsymbol{b}_i^T = \left(v_{i0}, \ldots, v_{ip}, u_{ip1}, \ldots, u_{ipK}\right)$ is given as

$$\boldsymbol{G} = Cov(\boldsymbol{b}_i) = \begin{bmatrix} \sum & 0 \\ 0 & \boldsymbol{D} \end{bmatrix}.$$

To complete the specification of the model in (2.8), we now need to define the form for the baseline risk function $h_0(.)$. Motivated by the fact that in real life, $h_0(.)$ is usually unknown. Therefore, two options are adopted to determine the form of the function $h_0(.)$ in this chapter. First, a standard option is to use a known parametric distribution for the risk function. For this option, the Gompertz distribution is chosen. Second, the piecewise constant model is chosen when $h_0(.)$ is considered completely unspecified.

Therefore, the proposed penalized spline joint models considered in this chapter are as follows:

Model 1: Penalized spline joint model with hazard rate at base line having Gompertz distribution

$$\begin{cases} h_i(t|\mathcal{M}_i(t), \boldsymbol{w}_i) = \lambda_1 \exp\left(\lambda_2 t\right) \exp\left\{\gamma^T \boldsymbol{w}_i + \alpha m_i(t)\right\} \\ \quad m_i(t) = \boldsymbol{X}_i^T(t)\beta + \boldsymbol{X}_i^T(t)\boldsymbol{v}_i + \boldsymbol{Z}_i^T(t)\boldsymbol{u}_i. \end{cases} \tag{10}$$

Model 2: Penalized spline joint model with a piecewise-constant baseline risk function

$$\begin{cases} h_i(t|\mathcal{M}_i(t), \boldsymbol{w}_i) = \sum_{q=1}^{Q} \xi_q I\left(\nu_{q-1} < t \leq \nu_q\right) \exp\left\{\boldsymbol{\gamma}^T \boldsymbol{w}_i + \alpha m_i(t)\right\} \\ \quad m_i(t) = \boldsymbol{X}_i^T(t)\beta + \boldsymbol{X}_i^T(t)\mathbf{v}_i + \boldsymbol{Z}_i^T(t)\boldsymbol{u}_i, \end{cases} \tag{11}$$

where $0 = \nu_0 < \nu_1 < \ldots < \nu_Q$ denotes a split of the time scale, with ν_Q being larger than the largest observed time and ξ_q denotes the value of the baseline hazard in the interval $\left[\nu_{q-1}, \nu_q\right)$. In both models, $\boldsymbol{X}_i$, $\boldsymbol{Z}_i$, β, $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ are given in (2.7).

3. Parameter estimation

After defining the two penalized spline joint models, we now present the joint likelihood and score functions of the parameters in the models. The ECM algorithm is also presented in this section.

3.1. Likelihood and score functions

Following Rizopoulos [2], we assume that the vector of time-independent random effects b_i underlies both the longitudinal and survival processes. This means that

$$\begin{aligned} p\left(T_i, \delta_i, \boldsymbol{y}_{\mathbf{i}} | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}\right) &= p(T_i, \delta_i | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}) p\left(\boldsymbol{y}_{\mathbf{i}} | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}\right) \\ p\left(\boldsymbol{y}_{\mathbf{i}} | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}\right) &= \prod_j p\{\boldsymbol{y}_{\mathbf{i}}(t_{ij}) | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}\}, \end{aligned} \tag{12}$$

where $\boldsymbol{\theta} = \left(\boldsymbol{\theta}_t^T, \boldsymbol{\theta}_y^T, \boldsymbol{\theta}_b^T\right)^T$ denotes the full parameter vector with $\boldsymbol{\theta}_t = \left(\gamma^T, \alpha, \theta_{h_0}^T\right)^T$ denoting the parameter vector for the survival outcomes. Furthermore, $\boldsymbol{\theta}_y = \left(\beta^T, \sigma_\varepsilon^2\right)^T$ is the parameter vector for longitudinal outcomes and $\boldsymbol{\theta}_b = vech(\boldsymbol{G})$ is the vector-half of the variance matrix of random effects. In addition, we assume that the hazard rate at time t conditional on the covariate path depends on the current value of longitudinal outcomes and the censoring mechanism is independent of the true event times and future longitudinal measurements. Under these assumptions, the log-likelihood formulation of the penalized spline joint models can be written as

$$\begin{aligned} l(\boldsymbol{\theta}) &= l\left(\boldsymbol{\theta} | T_i, \delta_i, \mathbf{y}_i\right) \\ &= \sum_i \log \int_{\mathbf{b}_i} p\left(T_i, \delta_i | \boldsymbol{b}_i; \boldsymbol{\theta}_t, \beta\right) p\left(\boldsymbol{y}_i | \boldsymbol{b}_i; \boldsymbol{\theta}_y\right) p(\boldsymbol{b}_i; \boldsymbol{\theta}_b) d\boldsymbol{b}_i, \end{aligned} \tag{13}$$

where the conditional density for survival part has the form of

$$\begin{aligned} p\left(T_i, \delta_i | \boldsymbol{b}_i; \boldsymbol{\theta}_t, \beta\right) &= h\left(T_i | \mathcal{M}_i(T_i), \boldsymbol{w}_i; \boldsymbol{\theta}_t, \beta\right)^{\delta_i} S\left(T_i | \mathcal{M}_i(T_i), \boldsymbol{w}_i; \boldsymbol{\theta}_t, \beta\right) \\ &= \left[h_0(t) \exp\left\{\gamma^T \boldsymbol{w}_i + \alpha m_i(t)\right\}\right]^{\delta_i} \exp\left[-\int_0^{T_i} h_0(s) \exp\left\{\gamma^T \boldsymbol{w}_i + \alpha m_i(s) ds\right\}\right]. \end{aligned} \tag{14}$$

Here, $S(t)$ is the survival function at time t.

Moreover, the density for the longitudinal part with the random effects is given by

$$\begin{aligned} p\left(\boldsymbol{y}_{\mathbf{i}} | \boldsymbol{b}_i; \boldsymbol{\theta}_y\right) p(\boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}_b) &= \prod_j p\left\{y_i(t_{ij}) | \boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}_y\right\} p(\boldsymbol{b}_{\mathrm{i}}; \boldsymbol{\theta}_b) \\ &= \frac{1}{\left(2\pi\sigma_\varepsilon^2\right)^{\frac{n_i}{2}}} \exp\left\{-\frac{\|y_i(t_{ij}) - \boldsymbol{X}_i^T(t_{ij})\beta - \boldsymbol{X}_i^T(t_{ij})\boldsymbol{v}_i - \boldsymbol{Z}_i^T(t_{ij})\boldsymbol{u}_i\|^2}{2\sigma_\varepsilon^2}\right\} \\ &\quad \times (2\pi)^{-\frac{q_b}{2}} \det(\boldsymbol{G})^{-1/2} \exp\left(-\boldsymbol{b}_i^T \boldsymbol{G}^{-1} \boldsymbol{b}_{\mathrm{i}}/2\right), \end{aligned} \tag{15}$$

where q_b denotes the dimensionality of the random effects vector.

We consider the log likelihood of the $(T_i, \delta_i, \boldsymbol{y}_i, \boldsymbol{b}_i)$ over the unknown $\boldsymbol{\theta}_t$, β and $\boldsymbol{b}_{\mathrm{i}}$

$$1\log l\left(\theta|T_i,\delta_i,\mathbf{y}_i,\mathbf{b}_\mathbf{i}\right) = \log p\left(T_i,\delta_i|\mathbf{b}_i;\theta_t,\beta\right) + \log p\left(\mathbf{y}_i|\mathbf{b}_i;\beta\right) + \log p(\mathbf{b}_i;\mathbf{G}).$$

The function for maximizing the log likelihood involves the density function of survival time and least squares with a penalty term, which is

$$\log p\left(T_i,\delta_i|\boldsymbol{b}_i;\boldsymbol{\theta}_t,\beta\right) - \frac{\left(\boldsymbol{y}_i - \boldsymbol{X}_i\beta - \boldsymbol{X}_i\boldsymbol{v}_i - \boldsymbol{Z}_i\boldsymbol{u}_i\right)^T\left(\boldsymbol{y}_i - \boldsymbol{X}_i\beta - \boldsymbol{X}_i\boldsymbol{v}_i - \boldsymbol{Z}_i\boldsymbol{u}_i\right)}{\sigma_\varepsilon^2} - \boldsymbol{b}_i^T\boldsymbol{G}^{-1}\boldsymbol{b}_\mathbf{i}. \tag{16}$$

According to Ruppert et al. [9], the term $\sigma_\varepsilon^2\boldsymbol{b}_i^T\boldsymbol{G}^{-1}\boldsymbol{b}_\mathbf{i}$ is called a roughness penalty and the variance components matrix defined as $\boldsymbol{F} = \sigma_\varepsilon^2\boldsymbol{G}^{-1}$. Using a Lagrange multiplier argument, the variance components matrix is the condition to constrain the coefficients of the knots $\boldsymbol{u}_i$. These will restrict the influence of the variables $(t - K_k)_+^p$ and will lead to smoother spline functions.

Using (3.2), the score vector for the penalized spline joint models can be expressed as:

$$\begin{aligned} S(\boldsymbol{\theta}) &= \sum_i \frac{\partial}{\partial\boldsymbol{\theta}^T}\log\int p\left(T_i,\delta_i|\boldsymbol{b}_i;\boldsymbol{\theta}_t,\beta\right)p\left(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\theta}_y\right)p(\boldsymbol{b}_i;\boldsymbol{\theta}_b)d\boldsymbol{b}_i \\ &= \sum_i \int \frac{\partial}{\partial\boldsymbol{\theta}^T}\log\left\{p\left(T_i,\delta_i|\boldsymbol{b}_i;\boldsymbol{\theta}_t,\beta\right)p(\boldsymbol{y}_i|\boldsymbol{b}_i;\boldsymbol{\theta}_y)p(\boldsymbol{b}_i;\boldsymbol{\theta}_b)\right\}.p\left(\boldsymbol{b}_i|T_i,\delta_i,\boldsymbol{y}_i;\boldsymbol{\theta}\right)d\boldsymbol{b}_i. \end{aligned} \tag{17}$$

The requirement for numerical integration with respect to the random effects is one of the main difficulties in the joint models [2]. The maximum likelihood estimation (MLE) is typically obtained using standard maximization algorithms such as expectation maximization algorithm or Newton-Raphson algorithm.

3.2. The ECM algorithm

The EM algorithm has been widely used in the joint models, such as for the standard joint model of Rizopoulos [2] and for the generalized linear mixed joint model [10]. The ECM algorithm is a natural extension of EM algorithm for which the maximization process on the M-step is conditional on some functions of the parameters under estimation. It also can reduce computer time. The ECM algorithm will be used to obtain the maximum likelihood estimates of the penalized spline joint models following McLachlan and Krishnan [11] in this chapter.

In these models, the random effects $\boldsymbol{b}_i$ are considered as missing data. Hence, it is difficult to estimate directly the parameter vector $\boldsymbol{\theta}$ that maximizes the observed data log likelihood $l(\boldsymbol{\theta})$ in (3.2). Alternatively, we can estimate the parameter vector $\boldsymbol{\theta}$ that maximizes the expected value of the complete data log-likelihood which is $E\left\{\log p\left(T_i,\delta_i,\boldsymbol{y}_\mathbf{i},\boldsymbol{b}_i;\boldsymbol{\theta}\right)|T_i,\delta_i,\boldsymbol{y}_\mathbf{i};\boldsymbol{\theta}^{(it)}\right\}$, where $\theta^{(it)}$ is the parameter vector given at the i^{th} iteration.

The following are the steps of this algorithm.

Step 1: *Initialization*

We first initialize the parameters. We assume that there are m parameters in the models and the starting value of the parameter vector is $\boldsymbol{\theta}^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_m^{(0)}\right)$. Based on these initial values, we calculate the log-likelihood using (3.2).

Step 2: *The E-step for the penalized joint models*

We fill in the missing data and replace the log-likelihood function of the observed data with the expected function of the complete data log-likelihood as follows:

$$\begin{aligned} Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(it)}\right) &= \sum_i \int \log \left\{p(T_i, \delta_i, \boldsymbol{y}_\mathbf{i}, \boldsymbol{b}_i; \boldsymbol{\theta})\right\}.p\left(\boldsymbol{b}_i|T_i, \delta_i, \boldsymbol{y}_\mathbf{i}; \boldsymbol{\theta}^{(it)}\right)d\boldsymbol{b}_i \\ &= \sum_i \int \left(\log p(T_i, \delta_i|\boldsymbol{b}_i; \boldsymbol{\theta}) + \log p(\boldsymbol{y}_\mathbf{i}|\boldsymbol{b}_\mathbf{i}; \boldsymbol{\theta}) + \log p(\boldsymbol{b}_i; \boldsymbol{\theta})\right).p\left(\boldsymbol{b}_i|T_i, \delta_i, \boldsymbol{y}_\mathbf{i}; \boldsymbol{\theta}^{(it)}\right)d\boldsymbol{b}_i. \end{aligned} \tag{18}$$

Step 3: *The conditional M-step for the penalized joint models.*

This step will be implemented in four stages as follows:

3.1 Given the current value of the parameter vector at the i^{th} iteration $\boldsymbol{\theta}^{(it)} = \left(\theta_1^{(it)}, \theta_2^{(it)}, \ldots, \theta_m^{(it)}\right)$, we calculate the log likelihood at $l\left(\boldsymbol{\theta}^{(it)}\right) = \sum_i \log \int_{b_i} p\left(T_i, \delta_i, \boldsymbol{y}_i, \boldsymbol{b}_i; \boldsymbol{\theta}^{(it)}\right)d\boldsymbol{b}_i$.

3.2 Propose the new value for the first parameter $\theta_1^{(prop)}$ which maximizes $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(it)}\right)$. Then, we calculate the log likelihood at $l\left(\boldsymbol{\theta}^{(prop)}\right)$ where $\boldsymbol{\theta}^{(prop)} = \left(\theta_1^{(prop)}, \theta_2^{(it)}, \ldots, \theta_m^{(it)}\right)$.

3.3 Set $\boldsymbol{\theta}_1^{(it)} = \boldsymbol{\theta}^{(prop)}$ if $l\left(\boldsymbol{\theta}^{(prop)}\right) \geq l\left(\boldsymbol{\theta}^{(it)}\right)$, otherwise set $\boldsymbol{\theta}_1^{(it)} = \boldsymbol{\theta}^{(it)}$.

3.4 Similarly, based on the value of the parameter vector $\boldsymbol{\theta}_1^{(it)}$, we update the new value for the second parameter and continue updating for the last parameter, $\boldsymbol{\theta}_m^{(it)}$ and set $\boldsymbol{\theta}^{(it+1)} = \boldsymbol{\theta}_m^{(it)}$.

Step 4: *Iterate among steps 2–3 until the algorithm numerically converges.*

To update the new values for parameters in the conditional M-step, we have the closed-form estimates for the measurement of error variance σ^2 and the covariance matrix of the random effects respectively by maximizing the expected function $Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(it)}\right)$. Unfortunately, we cannot obtain closed-form expressions for the remaining of the parameters. We thus employ the one-step Newton-Raphson approach to get the updates for $\beta^{(it+1)}$, $\gamma^{(it+1)}$, $\alpha^{(it+1)}$ and $\boldsymbol{\theta}_{h_0}^{(it+1)}$ respectively as detailed in Appendix B.

Following Louis [12], standard errors for the parameter estimates can be calculated by using the estimated observed information matrix

$$\widehat{var}\left(\widehat{\boldsymbol{\theta}}\right) = \left\{\mathcal{I}\left(\widehat{\boldsymbol{\theta}}\right)\right\}^{-1},$$

where

$$\mathcal{I}\left(\widehat{\boldsymbol{\theta}}\right) = -\sum_{i=1}^{n} \frac{\partial S_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}.$$

4. Empirical results

This section presents two simulation studies for Model 1, whereas Model 2 will be applied for a case study only. In Section 4.1, we simulate data from Model 1 with three internal knots in the longitudinal submodel and Gompertz distribution for the baseline risk function. In Section 4.2, we simulate data from Model 1 having Gompertz distribution for the baseline risk function and non-linear logarithm subject-specific trajectories. The ECM algorithm, written in R code, is applied to estimate the true values of parameters in both cases.

4.1. Simulation study 1

4.1.1. Data description

Recall the penalized spline joint Model 1 of (2.10) with three internal knots in longitudinal submodel and Gompertz distribution for the baseline risk function in the form of

$$h_i(t) = h_0(t)\exp\left(\gamma x_i + \alpha(m_i(t))\right) = \lambda_1 \exp\left(\lambda_2 t\right)\exp\left\{\gamma x_i + \alpha m_i(t)\right\}, \tag{19}$$

where $h_0(t)$ is the hazard function at baseline having Gompertz distribution, x_i is baseline covariate and $m_i(t)$ denotes the true and unobserved value of the longitudinal at time t. The form of $m_i(t)$ is given by

$$m_i(t) = \beta_0 + \beta_1 t + u_{i1}(t-\mathcal{K}_1)_+ + u_{i2}(t-\mathcal{K}_2)_+ + u_{i3}(t-\mathcal{K}_3)_+ + v_{i0}, \tag{20}$$

where $\boldsymbol{b}_i = (u_{11}, u_{12}, u_{13}, v_{i0})^T$ is the vector of random effects and is assumed to have a normal distribution with mean zero and the diagonal covariance matrix $\boldsymbol{D} = Diag(D_{11}, D_{22}, D_{33}, D_{44})$. $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ denote the three internal knots put into the model. The observed longitudinal value at time point t for the i^{th} subject is of the form

$$y_i(t) = m_i(t) + \varepsilon_i(t), \tag{21}$$

where the error variable $\varepsilon_i(t)$ is assumed to come from a normal distribution with mean zero and variance σ^2.

From this model, the vector of all parameters $\boldsymbol{\theta}$ of the models in (4.1) and (4.2) is $\boldsymbol{\theta} = \left(\boldsymbol{\theta}_t^T, \boldsymbol{\theta}_y^T, \boldsymbol{\theta}_b^T\right)^T$, where $\boldsymbol{\theta}_t = (\gamma, \alpha, \lambda_1, \lambda_2)^T$ denotes the parameter vector for the survival outcomes. Furthermore, $\boldsymbol{\theta}_y = \left(\beta_0, \beta_1, \sigma_\varepsilon^2\right)^T$ is the parameter vector for longitudinal outcomes and $\boldsymbol{\theta}_b = \boldsymbol{D}$ is the variance matrix of random effects.

To simulate the observed survival time T_i of the joint model in (4.1), we applied the methods adapted by Bender et al. [13], Austin [14] and Crowther and Lambert [15] to generate the true survival time. We further assumed that the censoring mechanism is exponentially distributed with parameter λ. The observed survival time was the minimum of the censoring time and the true survival time. We generated the survival time T_i for $n = 500$ subjects with the parameters: $\beta_0 = 5,\ \ \beta_1 = 2,\ \ \lambda_1 = 0.1,\ \ \lambda_2 = 0.5,\ \ \gamma = 0.5, \alpha = 0.05, \delta = 2$ and $D = Diag(2,2,2,4)$. Then we generated the longitudinal responses $m_i(t)$ using (4.2). The simulated model is therefore

$$\begin{cases} h_i(t) = 0.1 \exp{(0.5t)} \exp{\{0.5x_i + 0.05m_i(t)\}} \\ m_i(t) = 5 + 2t + u_{i1}(t-1)_+ + u_{i2}(t-2)_+ + u_{i3}(t-3)_+ + v_{i0}. \end{cases} \tag{22}$$

The sample of simulated data is presented in Appendix A. The curve of Kaplan-Meier estimate for the survival function of simulated data (left panel) and the longitudinal trajectories for the whole simulated sample (right panel) are presented in **Figure 1**. The dashed lines in the left panel correspond to 95% pointwise confidence intervals. It is clear from the plot of Kaplan-Meier estimator that the survival probability starts from 1 and decreases gradually until at the 5^{th} month of the study. After this, it is nearly zero after 6 months or so. The right panel is the longitudinal trajectories for the first 100 subjects reflecting the form as in (4.2).

4.1.2. Parameter estimation

The ECM algorithm, as described in Section 3.2, is now implemented to estimate all parameters in (4.4). The initial values of the parameters were set at $\beta_0 = 1,\ \ \beta_1 = 1,\ \ \lambda_1 = 0.05, \lambda_2 = 0.1,$ $\gamma = 0.1,\ \ \alpha = 0.01,\ \ \sigma = 1,\ \ D_{11} = 3,\ \ D_{22} = 3,\ \ D_{33} = 3,\ \ D_{44} = 3$, respectively. However, these initial values can also be set randomly. The traces of each of these parameters are presented in **Figures 2** and **3**, respectively. The traces of estimates show the way how the algorithm updates

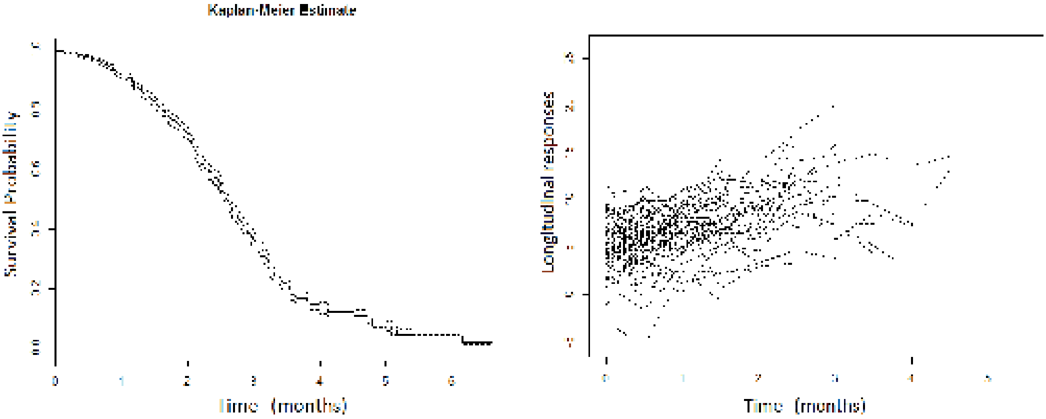


Figure 1. Kaplan-Meier estimate of the survival function of the simulated data of (4.4) (left panel). Longitudinal trajectories of the first 100 subjects from the simulated sample of (4.4) (right panel).

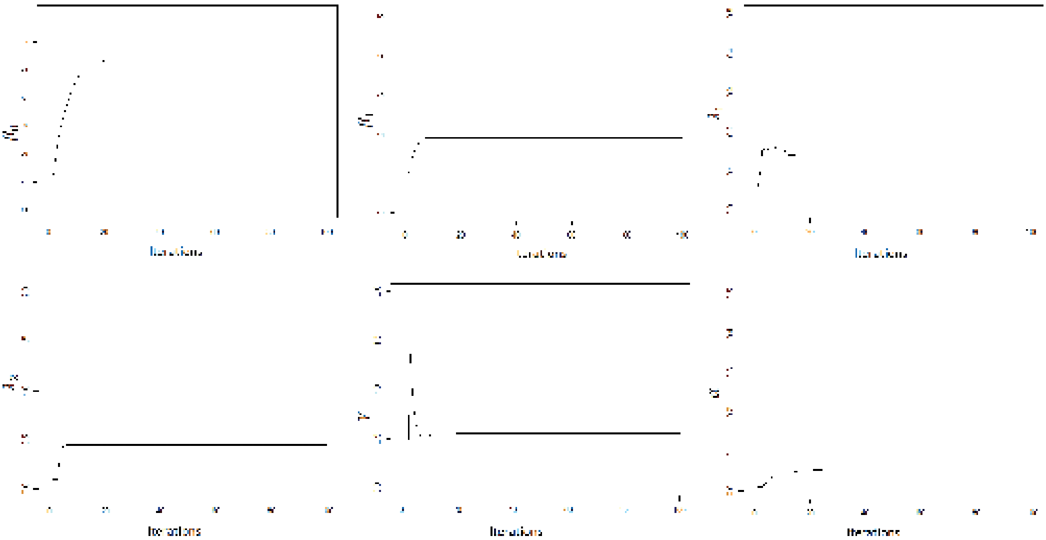


Figure 2. The traces of parameters $\beta_0, \ \beta_1, \ \lambda_1, \ \lambda_2, \ \ \gamma, \ \alpha$ for 100 iterations.

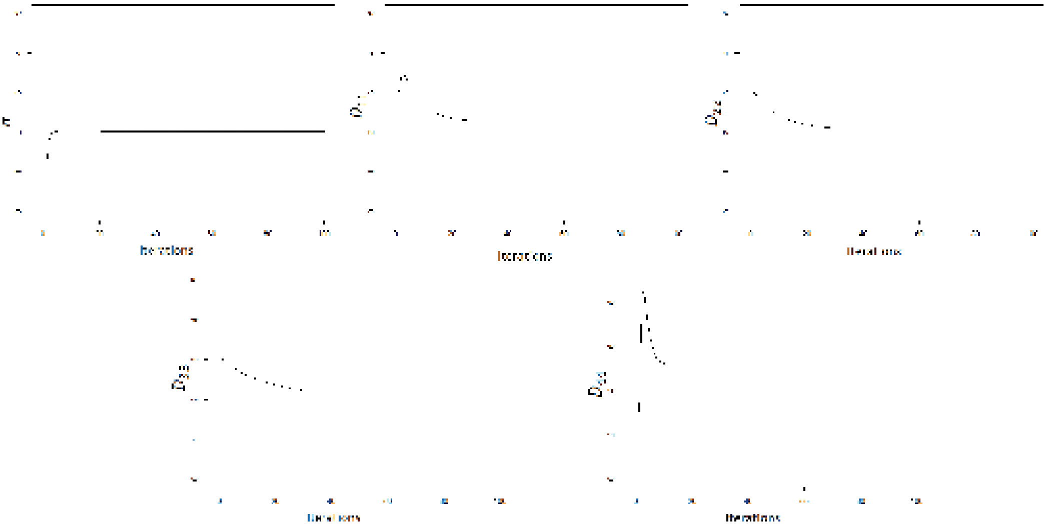


Figure 3. The traces of parameters $\sigma, \ \ D_{11}, \ \ D_{22}, \ \ D_{33}, \ \ D_{44}$ for 100 iterations.

new values of the parameters. In addition, they also demonstrate the convergence of the algorithm after 10–20 iterations. In particular, the parameters $\beta_0, \ \ \beta_1, \ \lambda_2, \ \ \alpha, \ \ \sigma, \ \ D_{22}$ and D_{33} converge linearly to the true values while the parameters $\lambda_1, \ \ \gamma, \ \ D_{11},$ and D_{44} oscillate before converging to the true values.

We now run the simulation for 30 independent samples with different sample sizes ($n = 200$, 300 and 500). Then, we calculate the means, standard deviations (SD) and mean square error (MSE) of parameters as presented in **Table 1**. The point estimates of each parameter are reasonably close to the true values when the sample sizes are 300 and 500. This is also supported by the values of SD and MSE which decrease gradually when the sample size increases. In addition to this, we also compare the parameter estimates with different censoring rates (20% and 40%) for a sample size of 500 in 5, Appendix E. The result shows that when the sample size is large the censoring rate has little influence on the estimates.

4.2. Simulation study 2

4.2.1. Data description

We now perform a simulation study on proportional hazard model having Gompertz distribution at baseline and non-linear subject-specific trajectory. In particular, the model is in the form of

$$h_i(t) = h_0(t)\exp\left(\gamma x_i + \alpha(m_i(t))\right) = \lambda_1 \exp\left(\lambda_2 t\right)\exp\left\{\gamma x_i + \alpha m_i(t)\right\}, \tag{23}$$

where $h_0(t)$ is the hazard function at baseline having Gompertz distribution, x_i is baseline covariate and $m_i(t)$ denotes the true and unobserved value of the longitudinal at time t. The observed longitudinal value at time point t for the i^{th} subject has the non-linear form

$$\begin{aligned} y_i(t) &= m_i(t) + \varepsilon_i(t) \\ &= 5\log\left(1+t\right) + b_{i1}t + b_{i0} + \varepsilon_i(t), \end{aligned} \tag{24}$$

Parameter	True value	$n = 200$			$n = 300$			$n = 500$		
		Estimate	SD	MSE	Estimate	SD	MSE	Estimate	SD	MSE
β_0	5	4.21	0.72	0.76	4.68	0.50	0.32	5.10	0.30	0.27
β_1	2	1.69	0.75	0.57	1.86	0.75	0.28	2.10	0.57	0.18
λ_1	0.1	0.12	0.13	0.00	0.12	0.12	0.00	0.11	0.10	0.00
λ_2	0.5	0.50	0.15	0.02	0.57	0.14	0.01	0.48	0.14	0.02
γ	0.5	0.50	0.17	0.03	0.49	0.12	0.04	0.51	0.09	0.01
α	0.05	0.03	0.04	0.00	0.04	0.05	0.00	0.04	0.04	0.00
σ	2	2.06	0.13	0.01	2.02	0.06	0.00	2.02	0.06	0.00
D_{11}	2	2.87	0.92	0.62	2.59	0.73	0.53	2.27	0.80	0.22
D_{22}	2	2.03	0.42	0.16	2.21	0.46	0.23	2.10	0.43	0.05
D_{33}	2	2.10	0.37	0.17	0.34	0.50	0.34	2.22	0.59	0.10
D_{44}	4	5.24	1.82	0.76	4.32	0.74	0.60	4.24	0.63	0.18

Table 1. Summary statistics for parameter estimation of the simulated data of the model in (4.4) for different sample sizes.

where $\varepsilon_i(t) \sim N\left(0, \sigma^2\right)$. In the model of (4.6), the mean longitudinal response of the population is assumed to have a non-linear logarithm curve. Different subjects are assumed to have different intercepts and slopes. In particular, we assume that $b_i = (b_{i0}, b_{i1})^T$ having a bivariate normal distribution with mean $\mu = (3, 2)$ and covariance matrix $D = Diag(1, 1)$. The true values of the other parameters we put in the model were $\lambda_1 = 0.01$, $\lambda_2 = 0.1, \gamma = 0.5$, $\alpha = 0.2$, $\sigma = 2$, respectively. In addition, the censoring mechanism is assumed exponentially distributed with a parameter of $\lambda = 0.25$.

Based on the model in (4.5) and the simulation study 1, we simulated survival times T_i for 500 subjects with 35% censoring rate. In particular, the ending time for the study was 5 months and all subjects alive by the end of the study (i.e. time 5) were assumed to be censored. This design was also reflected of many clinical studies in real life. In this sample, there were 329 uncensored subjects and 1387 observations for 500 subjects. For each subject, 1–5 longitudinal measurements were recorded. On average, there were three longitudinal measurements per subject. In **Figure 4**, the Kaplan-Meier estimate for survival curve is presented for the simulated data of (4.5) with 95% pointwise confidence intervals in the left panel. Moreover, the subject-specific longitudinal profiles for six randomly selected subjects is drawn in the right panel. It can be seen that some of the subjects in this dataset showed non-linear evolutions in their longitudinal values. Each subject has its own trajectory.

4.2.2. Parameter estimation

We will be using Model 1 in (4.1) and in (4.2) to handle the non-linear longitudinal trajectory in the simulated data in (4.5). In this model, we put three internal knots at 25, 50 and 75%, respectively, of the follow up time. Then, the ECM algorithm, as explained in Section 3, is implemented once again to estimate all parameters in the model.

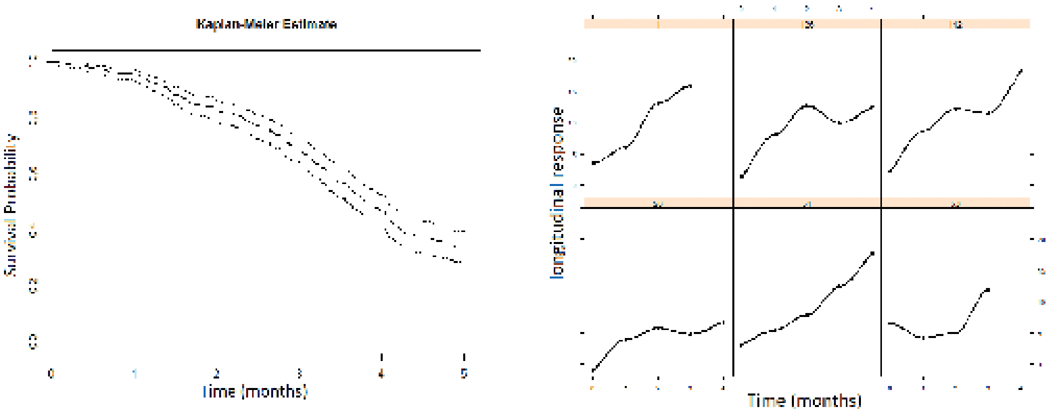


Figure 4. Kaplan-Meier estimate of the survival function of the simulated data of (4.5) (left panel). Longitudinal trajectories for the six randomly selected subjects of (4.6) (right panel).

Parameter	True value	Estimate	SD	95% CI
β_0	—	3.399	0.673	[3.158;3.640]
β_1	—	4.330	0.142	[4.280;4.380]
λ_1	0.01	0.013	0.021	[0.007;0.021]
λ_2	0.1	0.083	0.184	[0.017;0.148]
γ	0.5	0.640	0.386	[0.486;0.778]
α	0.2	0.186	0.142	[0.136;0.237]
σ	2	1.993	0.061	[1.971;2.015]
D_{11}	—	0.977	0.190	[0.909;1.044]
D_{22}	—	1.365	0.183	[1.300;1.430]
D_{33}	—	1.976	0.154	[1.921;2.031]
D_{44}	—	1.393	0.196	[1.322;1.463]

Table 2. Summary statistics for parameter estimation of the simulated data of the model in (4.5) and (4.6).

The results for parameter estimation are presented in **Table 2**. The means, standard deviations and 95% confidence intervals of parameter estimates are calculated for 30 independent samples. The point estimates for $\lambda_1, \lambda_2, \gamma, \alpha, \sigma^2$ are reasonably close to the true values. Similarly, the 95% CIs include the true values of $\lambda_1, \lambda_2, \gamma, \alpha, \sigma^2$.

Based on the estimated values of parameters, we generate back the estimated survival time by approximating values of random effects from linear mixed-effects function. The detail of the generation is explained in Appendix C. Then, we use the Kaplan-Meier estimate to compare between the survival function of the simulated dataset (the black solid line) and the estimated survival function (the dashed line) which are presented in the left panel of **Figure 5**.

Moreover, we also draw the smooth and predicted longitudinal profiles for 12 patients chosen randomly in the right panel of **Figure 5**. The dot points are the true observed longitudinal values from simulated data. The solid lines are the smooth longitudinal profiles of the true observed longitudinal values using the loess smoother and the dashed lines are the predicted profiles of 12 randomly selected individuals. It is clear that the Kaplan-Meier estimates from simulated data overlaps the Kaplan-Meier estimates based on the predicted value in the left panel of **Figure 4**. The penalized spline regression model in (2.10) was a good fit for subject-specific curves in the right panel of **Figure 5**.

In summary, simulation studies have shown the stability of the algorithm and the goodness of fit of the penalized spline models. From the simulation study 1, it is shown that the updating process through the ECM algorithm converges quickly to the true values of the parameters. In addition, the simulation study 2 shows that the model can well predict the survival function and individual trajectories respectively.

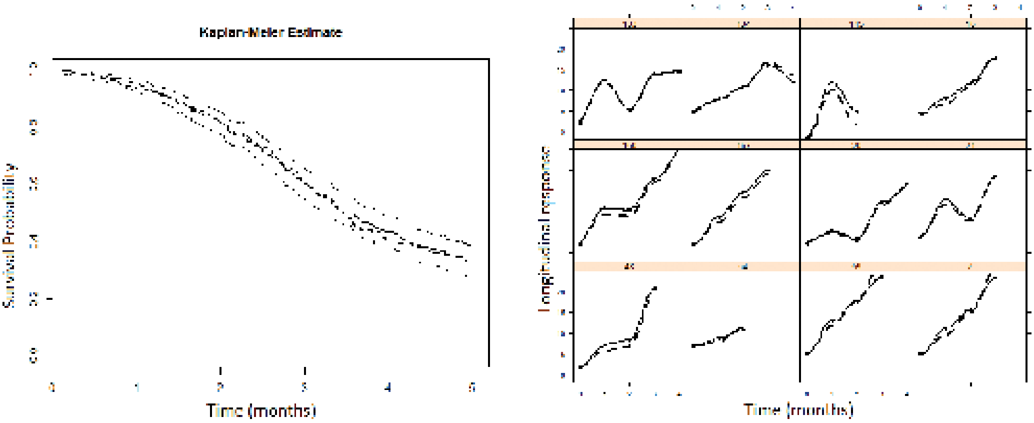


Figure 5. Kaplan-Meier estimate of the survival function from simulated failure times (the solid line) with 95% confidence intervals (dot lines), from Model 1 (4.5) (the dashed line) (left panel). Observed longitudinal trajectories (the solid line) and predicted longitudinal trajectories (the dashed line) for the twelve randomly selected subjects (right panel).

4.3. The AIDS data set

In the AIDS dataset, there were 467 patients with advanced human immunodeficiency virus infection during antiretroviral treatment who had failed or were intolerant to zidovudine therapy. Patients in the study were randomly assigned to receive either *didanosine* drug (*ddI*) or *zalcitabine* drug (*ddC*). CD4 cells are a type of white blood cells made in the spleen, lymph nodes and thymus gland and are part of the infection-fighting system. CD4 cell counts were recorded at the time of study entry as well as at 2, 6, 12 and 18 months thereafter. The detail regarding the design of this study can be found in Abrams et al. [16]. By the end of the study, there were 188 patients died, resulting in about 59.7% censoring. There were 1405 longitudinal responses recorded.

Previously, Rizopoulos [2] used his standard joint model for the AIDS data which consider the variability between subjects mostly depend on the intercept. However, the model could not predict observed longitudinal data accurately. When the time unit is changed from month to year in the data, the variability between subjects depends not only on the intercept but also on the *obstime* variable. In addition, the longitudinal trajectories plot also shows many non-linear curves as depicted in the right panel of **Figure 6**.

Given the non-linearity, it is appropriate to apply our models, Model 1 and Model 2, for the AIDS data. In particular, we use the two joint models in (2.10) and (2.11) with the four internal knots are placed at 20, 40, 60, 80%, respectively of the observed failure times for hazard rate at baseline. Then, the ECM algorithm is implemented to estimate all parameters in the two models. A summary of statistics for parameter estimation using Model 1 and Model 2 is presented in **Table 3**.

Following Rizopoulos [2], in Model 1 and Model 2, the univariate Wald tests are applied for the fixed effects $\beta = \left(\beta_0, \beta_1\right)^T$ in the longitudinal submodel, the regression coefficient γ and the association parameter α respectively. The results from **Table 3** show that the point estimates of $\beta_0,\ \beta_1,\ \gamma,\ \alpha$ are all statistically significant for both models at a significance level of 5%.

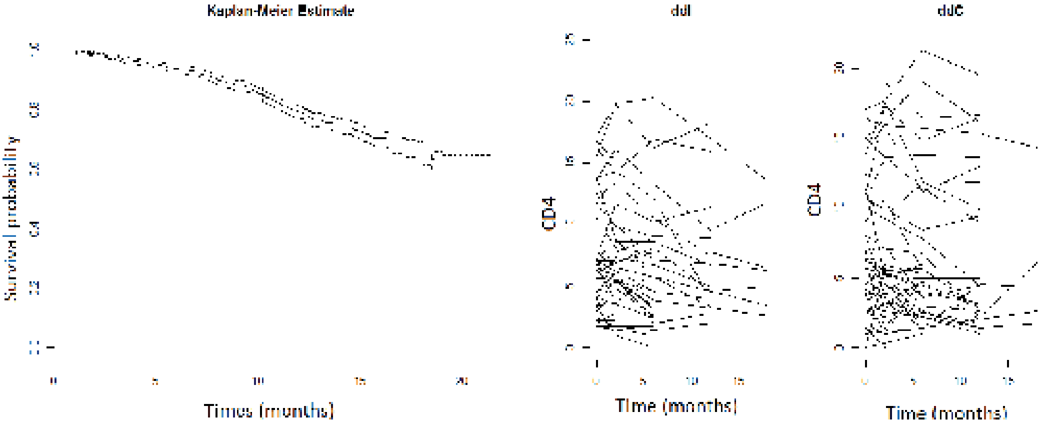


Figure 6. Kaplan-Meier estimate of the survival function of the AIDS data (left panel). Longitudinal trajectories for CD4 cell count of the first 100 patients for two groups (right panel).

Model 1					Model 2				
Parameter	**Estimate**	**Std. error**	**z-value**	**p-value**	**Parameter**	**Estimate**	**Std. error**	**z-value**	**p-value**
β_0	7.87	0.06	127.07	<0.001	β_0	7.81	0.07	114.34	<0.001
β_1	−1.69	0.11	−14.77	<0.001	β_1	−1.62	0.12	−12.99	<0.001
γ	0.22	0.11	2.06	0.039	γ	0.31	0.10	3.03	0.002
α	−0.20	0.01	−15.84	<0.001	α	−0.24	0.01	−18.15	<0.001
λ_1	1.68	0.07			λ_1	1.04	0.11		
λ_2	0.33	0.00			λ_2	1.79	0.23		
σ	2.36	0.36			λ_3	1.38	0.38		
D_{11}	2.18	0.14			λ_4	1.67	0.42		
D_{22}	1.04	0.07			λ_5	2.48	0.66		
D_{33}	0.85	0.06			σ	2.62	0.45		
D_{44}	11.87	0.78			D_{11}	1.02	0.07		
					D_{22}	0.97	0.06		
					D_{33}	0.99	0.07		
					D_{44}	11.40	0.75		

Table 3. Summary statistics for parameter estimation of the AIDS data of Model 1 and Model 2 respectively.

We conduct the Kaplan-Meier estimate of the survival function from the observed survival time (the light solid line) and the dot lines correspond to 95% pointwise confidence intervals in **Figure 6** (left panel). The predicted survival function from Model 1 is the dashed line and the predicted survival function from Model 2 is the bold solid line. The plots show that Model 2 works very well in this case as shown in **Figure 7**. Moreover, Model 2 is also preferred in

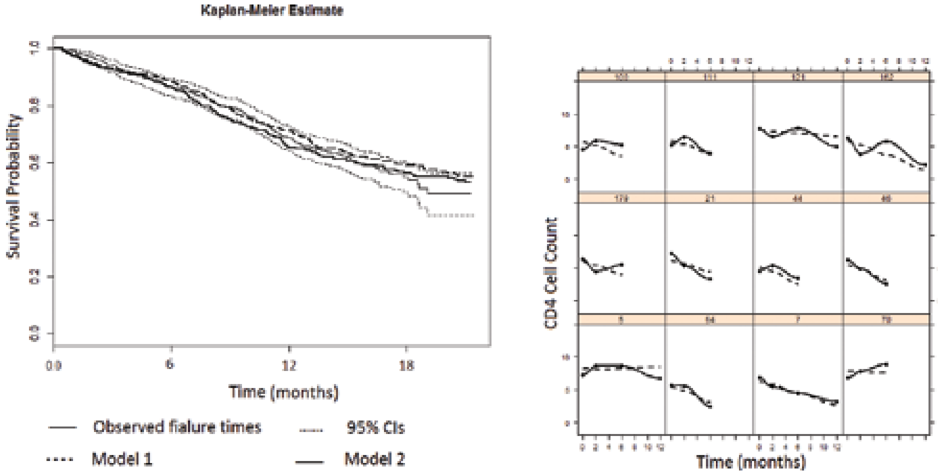


Figure 7. Kaplan-Meier estimates of the survival function from observed failure times, from model 1 and from model 2 (left panel). Observed longitudinal trajectories (the solid line) and predicted longitudinal trajectories (the dashed line) for the 12 randomly selected patients (right panel).

practice because $h_0(.)$ usually is considered as unspecified in order to avoid the impact of misspecifying the distribution of survival times.

Based on the model of longitudinal regression in (4.2), we also draw the smooth and predicted longitudinal profiles for nine patients from the AIDS dataset as depicted in **Figure 7** (right panel). The dot points are the true observed longitudinal values. The solid lines are the smooth longitudinal profiles using the loess smoother and the dashed lines are the predicted profiles of nine randomly selected individuals. Most of the predicted profiles are quite close to the observed ones.

5. Discussion

In this chapter, two joint models using a penalized spline with a truncated polynomial basis have been proposed to model a non-linear longitudinal outcome and a time-to-event data. The use of a truncated polynomial basis gives us an intuitive and obvious way to model non-linear longitudinal outcome. By adding some penalties for the coefficients of the knots and using linear mixed-effects models, the smoothing is controlled and the individual curves are specified.

We have conducted a sensitivity analysis on the assumption of normality for either random effects or errors. The t-distribution with the degree of freedom 5 is applied for each of them. The results show that the estimates of parameters are sensitive when both of terms are not normally distributed.

The main findings we may derive from this chapter are, at least, threefold: (1) the ECM algorithm provides a reasonable quick convergence algorithm for the proposed models; (2) the fitted joint models are able to measure the association between the internal time-dependent covariates and the risk for an event and (3) the two models provide a good prediction for both the longitudinal and survival functions, as presented in empirical results.

The limitations of this study are, at least, threefold: (1) the number of internal knots is limited to three due to computational costs; (2) the polynomial power functions can form an ill-conditioned basis for the models and (3) the estimation results are sensitive when both random effects and error are not normally distributed.

Based on the limitations, our future work will focus on using new methods for approximating the integrals to reduce the computational problems or relaxing the normality assumption. Furthermore, we will apply a different basis for joint models, that is the penalized B-spline. In terms of parameter estimation, we are considering a different approach to estimate the parameters in the models using a Bayesian approach, via Markov chain Monte Carlo (MCMC) algorithms.

A. Appendix A

One sample of simulated data of the penalized spline joint model in (4.4) is presented in **Table 4** for the first three patients. The subjects are measured bimonthly and the entry time is 0 for all

Id	Obstime	Time	x	y	Death	Z_1	Z_2	Z_3	Z_4
1	0.0	4.97	0	1.41	1	0.0	0.0	0.0	1
1	0.5	4.97	0	6.45	1	0.0	0.0	0.0	1
1	1.0	4.97	0	4.10	1	0.0	0.0	0.0	1
1	1.5	4.97	0	1.50	1	0.5	0.0	0.0	1
1	2.0	4.97	0	4.07	1	1.0	0.0	0.0	1
1	2.5	4.97	0	6.16	1	1.5	0.5	0.0	1
1	3.0	4.97	0	3.60	1	2.0	1.0	0.0	1
1	3.5	4.97	0	8.32	1	2.5	1.5	0.5	1
1	4.0	4.97	0	6.32	1	3.0	2.0	1.0	1
2	0.0	2.79	0	6.81	1	0.0	0.0	0.0	1
2	0.5	2.79	0	7.77	1	0.0	0.0	0.0	1
2	1.0	2.79	0	9.75	1	0.0	0.0	0.0	1
2	1.5	2.79	0	11.04	1	0.5	0.0	0.0	1
2	2.0	2.79	0	7.20	1	1.0	0.0	0.0	1
3	0.0	1.90	0	−1.84	0	0.0	0.0	0.0	1
3	0.5	1.90	0	1.12	0	0.0	0.0	0.0	1
3	1.0	1.90	0	0.78	0	0.0	0.0	0.0	1

Table 4. A snapshot of simulated data for penalized spline joint model in (4.4).

subjects. *Obstime* variable includes the time points at which these measurements are recorded. *Time* variable includes the observed survival times when subject meets an event. x is a time-constant binary random variable with parameter $p = 0.5$. Column y contains the longitudinal responses. *Death* variable is the event status indicator. This variable receives value 1 when the true survival time is less than or equal to the censoring time and 0 otherwise. We define the four random effects variables which are $Z_1 = (obstime - \mathcal{K}_1)_+$, $Z_2 = (obstime - \mathcal{K}_2)_+$, $Z_3 = (obstime - \mathcal{K}_3)_+$ and $Z_4 = \mathbf{1}$. For the longitudinal process, there are 1902 of observations for 500 subjects. For each subject, 1-7 longitudinal measurements are recorded. On average, there are four longitudinal measurements per subject. For the event process, there are 297 subjects who meet for an event which is equivalent to 59.4% of the whole sample.

B. Appendix B

The integrals with respect to the random effects in (3.7) do not have closed-form solutions. Therefore, in this chapter, we implement the Gaussian-Hermite quadrature rule as in Rizopoulos [5] to approximate the integrals. In our simulation study and R coding, we use the Gaussian-Hermite quadrature rule with 10 quadrature points.

The updating formulas of the parameters in Step 3 have different forms for each parameter following Rizopoulos [2]. We have the closed-form estimates for the measurement error variance σ_ε^2 in the longitudinal model and the covariance matrix of the random effects as follows:

$$\widehat{\boldsymbol{G}}^{(it+1)} = \frac{1}{n}\sum_i \int \boldsymbol{b}_i^T \boldsymbol{b}_i p\left(\boldsymbol{b}_i | T_i, \delta_i, \boldsymbol{y}_i; \boldsymbol{\theta}^{(it)}\right) d\boldsymbol{b}_i = \frac{1}{N}\sum_i v\tilde{\boldsymbol{b}}_i^{(it)} + \tilde{\boldsymbol{b}}_i^{(it)} \tilde{\boldsymbol{b}}_i^{(it)} T, \tag{25}$$

where $\tilde{\boldsymbol{b}}_i = E(\boldsymbol{b}_i | T_i, \delta_i, \boldsymbol{y}_{\mathbf{i}}; \boldsymbol{\theta}) = \int \boldsymbol{b}_i p(\boldsymbol{b}_{\mathbf{i}} | T_i, \delta_i, \boldsymbol{y}_{\mathbf{i}}; \boldsymbol{\theta}) d\boldsymbol{b}_i$ and $\tilde{v}\boldsymbol{b}_i = \mathrm{var}(\boldsymbol{b}_i | T_i, \delta_i, \boldsymbol{y}_{\mathbf{i}}; \boldsymbol{\theta}) = \int (\boldsymbol{b}_i - \tilde{\boldsymbol{b}}_i) p(\boldsymbol{b}_{\mathbf{i}} | T_i, \delta_i, \boldsymbol{y}_{\mathbf{i}}; \boldsymbol{\theta}) d\boldsymbol{b}_i$. The updating formula for σ_ε^2 is

$$\widehat{\sigma}^2_{\varepsilon(it+1)} = \frac{1}{n}\sum_i \int \boldsymbol{W}^T \boldsymbol{W} p\left(\boldsymbol{b}_i | T_i, \delta_i, \boldsymbol{y}_i; \boldsymbol{\theta}^{(it)}\right) d\boldsymbol{b}_i, \tag{26}$$

where $\boldsymbol{W} = \boldsymbol{y}_i - \boldsymbol{X}_i\beta - \boldsymbol{X}_i\boldsymbol{u}_i - \boldsymbol{Z}_i\boldsymbol{v}_i$.

Unfortunately, we cannot obtain closed-form expressions for the fixed effects β and the parameters of the survival submodel $\gamma,\ \alpha,\ and\ \boldsymbol{\theta}_{h_0}$. We thus employ the one-step Newton-Raphson approach to obtain the updated $\beta^{(it+1)}$, $\gamma^{(it+1)}$,$\alpha^{(it+1)}$ and $\boldsymbol{\theta}_{h_0}^{(it+1)}$. In particular, we have

$$\begin{aligned} S(\boldsymbol{\theta}) &= \frac{\partial Q\left(\boldsymbol{\theta} | \boldsymbol{\theta}^{(it)}\right)}{\partial \boldsymbol{\theta}} \\ \widehat{\boldsymbol{\theta}}^{(it+1)} &= \widehat{\boldsymbol{\theta}}^{(it)} - \left[\frac{\partial S\left(\widehat{\boldsymbol{\theta}}^{(it)}\right)}{\partial \boldsymbol{\theta}}\right]^{-1} S\left(\widehat{\boldsymbol{\theta}}^{(it)}\right), \end{aligned} \tag{27}$$

where $S(\boldsymbol{\theta})$ is the score vector corresponding to parameter $\boldsymbol{\theta}$ and the score vector has the form of

$$\begin{aligned} S(\boldsymbol{\theta}) &= \frac{\partial Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(it)}\right)}{\partial \boldsymbol{\theta}} \\ &= \sum_i \int \frac{\partial}{\partial \boldsymbol{\theta}^T} \log\left\{p\left(T_i, \delta_i|\boldsymbol{b}_i; \boldsymbol{\theta}^{(it)}\right)p(\boldsymbol{y}_i|\boldsymbol{b}_i; \boldsymbol{\theta}^{(it)})p(\boldsymbol{b}_i; \boldsymbol{\theta}^{(it)})\right\}.p\left(\boldsymbol{b}_i|T_i, \delta_i, \boldsymbol{y}_i; \boldsymbol{\theta}^{(it)}\right)d\boldsymbol{b}_i. \end{aligned}$$

C. Appendix C

There are four cases for simulating survival time T_i of the model (4.1) as follows.

When the survival time $t < \mathcal{K}_1$, we calculate the cumulative hazard function $H_i(t) = \int_0^t h_i(s)ds$. Based on the relation between the survival function $S_i(t)$, cumulative hazard function $H_i(t)$ and cumulative distribution $F_i(t)$, we have

$$S_i(t) = \exp\left(-H_i(t)\right) = 1 - F_i(t). \tag{28}$$

Following (5.4), we set

$$u = 1 - F_i(T_i), \tag{29}$$

where u is a random variable with $u \sim Uni[0, 1]$. The survival time t is the solution of the equation

$$U = \exp\left(-H_i(t)\right) = \exp\left(-\int_0^t h_i(s)ds\right).$$

The condition $t < \mathcal{K}_1$ is equal to

$$-\log\left(U\right) < \int_0^{\mathcal{K}_1} h(s)ds.$$

When $\mathcal{K}_1 \leq t < \mathcal{K}_2$, we calculate the cumulative hazard function $H_i(t) = \int_0^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^t h_i(s)ds$.

The survival time t is the solution of the equation

$$U = \exp\left[-\left\{\int_0^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^t h_i(s)ds\right\}\right],$$

where U is a value of $u \sim Uni[0, 1]$. The condition $\mathcal{K}_1 \leq t < \mathcal{K}_2$ is equal to

$$-\log(U) < \int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds.$$

When $\mathcal{K}_2 \leq t < \mathcal{K}_3$, we calculate the cumulative hazard function $H_i(t) = \int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds + \int_{\mathcal{K}_2}^{t} h_i(s)ds$. The survival time t is the solution of the equation

$$U = \exp\left[-\left\{\int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds + \int_{\mathcal{K}_2}^{t} h_i(s)ds\right\}\right],$$

where U is a value of $u \sim Uni[0,1]$. The condition $\mathcal{K}_2 \leq t < \mathcal{K}_3$ is equal to

$$-\log(U) < \int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds + \int_{\mathcal{K}_2}^{\mathcal{K}_3} h_i(s)ds.$$

When $\mathcal{K}_3 \leq t$, the cumulative hazard function has the form $H_i(t) = \int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds + \int_{\mathcal{K}_2}^{\mathcal{K}_3} h_i(s)ds + \int_{\mathcal{K}_3}^{t} h_i(s)ds$. Survival time t is the solution of the equation

$$U = \exp\left[-\left\{\int_{0}^{\mathcal{K}_1} h_i(s)ds + \int_{\mathcal{K}_1}^{\mathcal{K}_2} h_i(s)ds + \int_{\mathcal{K}_2}^{\mathcal{K}_3} h_i(s)ds + \int_{\mathcal{K}_3}^{t} h_i(s)ds\right\}\right].$$

D. Appendix D

In particular, Ruppert et al. [9] introduced a default choices for knot location and number of knots. The idea is to choose sufficient knots to resolve the essential structure in the underlying regression function. But for more complicated penalized spline models, there are computational advantages to keeping the number of knots relatively low. A reasonable default is to choose the knots to ensure that there are a fixed number of unique observations, say 4–5, between each knot. For large data sets, this can lead to an excessive numbers of knots; therefore, a maximum number of allowable knots (say, 20–40 total) are recommended.

According to Ruppert et al. [9], the choice for knot position is

$\mathcal{K}_k = \left(\frac{k+1}{K+2}\right) th$ sample quantile of the unique x_i for $k = 1, \ldots, K$.

The simple choice of K is

$K = min\left(\frac{1}{4} \times number\, of\, unique\, x_i, 35\right)$.

E. Appendix E

See **Table 5**.

Parameter	True value	Censored (20%)			Censored (40%)		
		Estimate	SD	MSE	Estimate	SD	MSE
β_0	5	4.85	0.30	0.25	5.10	0.30	0.27
β_1	2	1.86	0.45	0.20	2.10	0.57	0.18
λ_1	0.1	0.13	0.12	0.00	0.11	0.10	0.00
λ_2	0.5	0.52	0.07	0.00	0.49	0.14	0.02
γ	0.5	0.48	0.10	0.00	0.51	0.09	0.00
α	0.05	0.05	0.02	0.00	0.04	0.04	0.00
σ	2	2.02	0.05	0.00	2.02	0.06	0.00
D_{11}	2	2.21	0.67	0.17	2.27	0.80	0.22
D_{22}	2	2.16	0.27	0.09	2.10	0.43	0.05
D_{33}	2	2.26	0.27	0.01	2.22	0.60	0.10
D_{44}	4	4.20	0.53	0.20	4.24	0.63	0.18

Table 5. Summary statistics for parameter estimation of the simulated data of the model in (22) for different censoring rates.

Author details

Huong Thi Thu Pham[1]* and Hoa Pham[2]

*Address all correspondence to: pham0092@flinders.edu.au

1 School of Computer Science, Engineering and Mathematics, Flinders University, Adelaide, South Australia, Australia

2 Mathematical Department, An Giang University, Long Xuyen, Viet Nam

References

[1] Cox DR. Regression models and life-tables. Journal of the Royal Statistical Society. Series B (Methodological). 1972;**34**(2):187-220

[2] Rizopoulos D. Joint Models for Longitudinal and Time-To-Event Data with Applications in R. Chapman & Hall/CRC Biostatistics series; 2012

[3] Brown ER, Ibrahim JG, DeGruttola V. A flexible B-spline model for multiple longitudinal biomarkers and survival. Biometrics. 2005;**61**(1):64-73

[4] Ding J, Wang J-L. Modeling longitudinal data with nonparametric multiplicative random effects jointly with survival data. Biometrics. 2008;**64**(2):546-556

[5] Rizopoulos D. Fast fitting of joint models for longitudinal and event time data using a pseudo-adaptive Gaussian quadrature rule. Computational Statistics and Data Analysis. 2011;**56**:491-501

[6] Rizopoulos D. JM: An R package for the joint modelling of longitudinal and time-to-event data. Journal of Statistical Software. 2010;**35**(9):1-33

[7] Kalbfleisch J, Prentice R. The Statistical Analysis of Failure Time Data. 2nd ed. New York: Wiley; 2002

[8] Durban M, Harezlak J, Wand M, Carroll R. Simple fitting of subject-specific curves for longitudinal data. Statistics in Medicine. 2005;**24**(8):1153-1167

[9] Ruppert D, Wand M, Carroll R. Semiparametric Regression. Cambridge: Cambridge University Press; 2003

[10] Viviani S, Alfo M, Rizopoulos D. Generalized linear mixed joint model for longitudinal and survival outcomes. Statistics and Computing. 2014;**24**(3):417-427

[11] McLachlan G, Krishnan T. The EM Algorithm and Extensions. Vol. 382. John Wiley & Sons; 2007

[12] Louis TA. Finding the observed information matrix when using the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological). 1982:226-233

[13] Bender, Augustin Blettner, Bender, Bender R, Augustin T, Blettner, M. 2005. Generating survival times to simulate cox proportional hazards models, Statistics in Medicine. **24**(11): 1713-1723

[14] Austin PC. Generating survival times to simulate cox proportional hazards models with time-varying covariates. Statistics in Medicine. 2012;**31**(29):3946-3958

[15] Crowther MJ, Lambert PC. Simulating biologically plausible complex survival data, Statistics in Medicine. 2013;**32**(23):4118-4134

[16] Abrams D, Goldman A, Launer C, Korvick J, Neaton J, Crane L, Grodesky M, Wakefield S, Muth K, Kornegay S. Comparative trial of didanosine and zalcitabine in patients with human immunodeficiency virus infection who are intolerant of or have failed zidovudine therapy. New England Journal of Medicine. 1994;**330**:657-662

Chapter 7

Application of Cubic Spline Interpolation Technique in Power Systems: A Review

Akhil Prasad, Atul Manmohan,
Prabhakar Karthikeyan Shanmugam and
Kothari D.P.

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74853

Abstract

In this chapter, a comprehensive review is made on the application of cubic spline interpolation techniques in the field of power systems. Domains like available transfer capability (ATC), electric arc furnace modeling, static var. compensation, voltage stability margin, and market power determination in deregulated electricity market are taken as samples to illustrate the significance of cubic spline interpolation.

Keywords: power systems, available transfer capability, electric arc furnace, thyristor-controlled reactor, voltage stability, market power

1. Introduction

With electricity becoming an inevitable part of all spheres of human life, it is imminent that the increasing demand for electricity be met. The realization of this necessity has manifested in extensive research in the field of power systems, which has brought to light the complexity of power system. The power system involves the continuous variation in connected loads and increases in power requirement which demands a corresponding increase in the generation. This is met via several means most recently characterized by the penetration of renewable energy sources. The unpredictability of these events entails the deployment of probabilistic-based load flow techniques, estimation of unknown variables and load models which includes uncertainties for their analysis.

All the analysis and research aimed at providing solutions or mitigating the above-mentioned problems require vast amounts of data which is discrete in nature or the analysis techniques involve responses that are discrete in nature. Regardless of the origin of this non-continuity of data, obtaining a continuous response is imperative because of the desired accuracy and the continuity of real-time operation.

The challenges set the stage for interpolation techniques to play an active role in mitigating the problems prevailing in the current power system studies like load forecasting, power system reactive power planning, transmission network expansion, available transfer capability (ATC) determination and market power/clearing price forecasting. Among the various interpolation techniques available, the cubic spline method has been found to be a popular method. Cubic spline also has the desirable characteristics of continuous derivatives at data points which make the design of controllers around these regions possible, and its employment has been seconded by the high accuracy obtained.

The measurements we take for analysis in everyday life are wrought with noise that may be caused by the surrounding environment. Especially in electrical measurements, such random noises may be caused by the magnetic field produced by the current, the presence of stray charges, heating caused by the flow of eddy currents, and so on. One of the possible solutions to the problems caused by noise is to take a large number of measurements. This ensures that the random noise gets canceled out on an average, and hence the integrity of the data is maintained. Other complicated methods of tackling the errors caused by noise are available, but the detailed and in-depth analysis of these methods falls beyond the scope of this chapter.

2. Determination of available transfer capability

The first application of cubic splines is in finding the available transfer capability, which is an important parameter in power system operation. Ever since the advent of the deregulated power system, the computation of transfer capability has been a priority. Two quantities that require special attention in these computations are the total transfer capability (TTC) and the available transfer capability (ATC). The TTC of a power system is defined as the maximum amount of power that can be transferred over the interconnected transmission network in a reliable manner while meeting all of a specific set of defined pre- and post-contingency system conditions [1]. ATC is defined as the measure of the additional amount of power that may flow over and above the base case flows without jeopardizing power system security [1]. The system operators of the deregulated power system normally can obtain previously calculated ATC values through an open-access same-time information system. However, the need for a quick calculation of ATC poses a challenge. Some of the methods employed include DC power flow, AC power flow, optimal power flow and sensitivity. Most of the methods mentioned earlier are not sufficiently agile as far as computation speed is concerned or trades accuracy for speed. For example, DC power flow yields quick results at the expense of accuracy, whereas AC power flow compromises speed for accuracy. A solution to the abovementioned problem is found by employing curve-fitting techniques especially the cubic spline interpolation technique.

In order to compute ATC, the cubic spline is employed to trace the curve of the variation of voltage magnitude and power flow with respect to the real power transfer. The ATC is then determined by the point where the limits of voltage magnitude or power flow intersect the curves. The computation of ATC takes place in two different forms—the point-to-point ATC and area-to-area ATC. The area-to-area ATC refers to the additional power that can be mobilized from the seller area to the buyer area, whereas the point-to-point ATC refers to bus-to-bus transfer (usually from a generating bus to a load bus). Another factor considered of paramount importance for the computation of ATC is the effect of contingencies like line outages. Individual consideration of line outages for a large-scale power system is an impractical approach, and hence contingency analysis is carried out using contingency ranking which helps select the critical lines.

The mathematical definition of ATC is shown in Eq. (1):

$$\mathrm{ATC} = \mathrm{TTC} - \mathrm{TRM} - \mathrm{ETC} - \mathrm{CBM} \tag{1}$$

The definition of the abovementioned terms is as follows:

Transmission reliability margin (TRM): the amount of transmission capability necessary to ensure the security of the interconnected system under a reasonable range of uncertainties in the system.

Capacity benefit margin (CBM): the amount of transmission capability reserved by load-serving entities to ensure access to generation from interconnected systems to meet generation reliability requirements.

Existing transmission commitments (ETC): normal transmission flows.

Traditionally, ATC computation involves the recursive application of AC power flow with increasing power transfers, thereby tracing voltage magnitudes and MVA power flow variations with respect to real power formally called the P-V and the P-S curves as shown in **Figure 1**. These curves are then employed in conjunction with the limits imposed by acceptable voltage magnitudes and power flows to calculate the ATC. In the cubic spline-based approach, the cubic spline interpolation technique is used to trace the P-V and P-S curves which are then

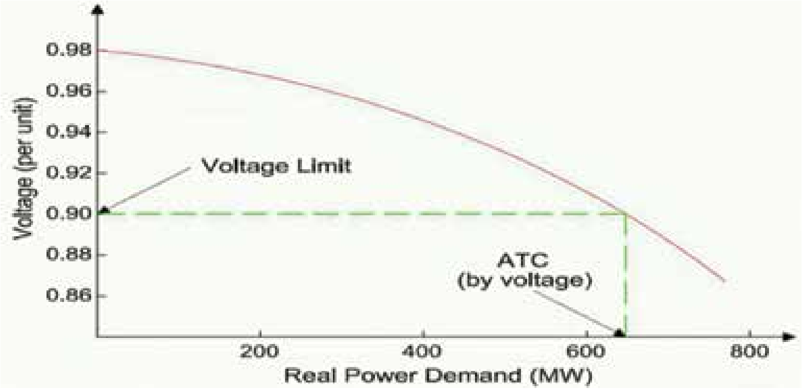


Figure 1. Variation of voltage with real power transfer.

employed to calculate the ATC. This is done by first determining four known points on the curve using AC power flow and then using cubic spline interpolation to trace the curves in between those points. The four known points are denoted as V_i (P_n) and S_{ij} (P_n), where V_i is for voltage of bus 'I', S_{ij} is the MVA power flow between bus 'i' and 'j', P_n is the real power transfer and n is the index for the four points and has values of n = 1, 2, 3 and 4. The incremental steps for tracing the P-V or the P-S curve will be of 1 MW each.

The first and foremost step in the process of curve tracing consists of the determination of the four points on the P-V and P-S curves. This is of paramount importance because the power system undergoes voltage collapse once the real power transfer crosses a certain limit. In the voltage collapse process, the particular bus faces a continuous drop in bus voltage once the critical point load is exceeded. Therefore, the points must be selected carefully making sure that the critical point is not exceeded.

2.1. Procedure to calculate ATC

Step 1. Perform the line contingency ranking using line-loading performance index (PI_{MW}) and bus voltage performance (PI_v). This is done to identify the critical lines. The critical lines will have a PI value greater than $PI_{base\ case}$.

Step 2. Find a base case by solving the AC load flow. This will act as the first point.

Step 3. Perform the simulation for line outage for one of the critical lines found in step 1.

Step 4. Specify the point or area of transfers. For the point-to-point transfer, generally a generator bus is taken as the selling bus and a load bus as the buying bus. On the other hand, for area-to-area transfer, all the generator busses in a particular area called the selling are and all the load busses in a specific buying area are considered.

Step 5. Now, the next three points are determined. The first step is to determine the fourth and final point P4. In order to do so, the sensitivity method is employed. This method is based on the limiting point of system constraints which is given in Eqs. (2)–(4)

$$PT_{i,V_L} = \left| \frac{\partial\lambda}{\partial V_i} \times \left(V_L - V_i^0\right) \right| \tag{2}$$

$$PT_{i,V_U} = \left| \frac{\partial\lambda}{\partial V_i} \times \left(V_U - V_i^0\right) \right| \tag{3}$$

$$PT_{ij,S} = \left| \frac{\partial\lambda}{\partial S_{ij}} \times \left(S_{ij}^{limit} - S_{ij}^0\right) \right| \tag{4}$$

In Eqs. (2)–(4), $PT_{i,VL}$, $PT_{i,VU}$ and $PT_{ij,S}$ are the estimates of P_4 which are arrived at by using the linear estimates based on the lower voltage limit, the upper voltage limit and the thermal limit of each line, respectively. The lower limit of voltage V_L is taken as 0.9 p.u. and the upper limit V_u is taken as 1.1 p.u. $S^{limit}{}_{ij}$ is the thermal limit of each line. $V_i{}^0$ and $S^0{}_{ij}$ are the base case values computed in step (ii). The derivative terms are the reciprocal of the rate of change of voltage and MVA power flow with real power. This can be obtained by solving the AC power flow starting from the base case and making a small change in the transferred real power and noting the change in voltage and MVA transfer. The fourth known point P_4 is selected as the minimum of all three values computed above. Then, based on this, P_2 and P_3 are chosen using the formulae given in Eqs. (5) and (6)

$$P_2 = \frac{P_4}{3} \tag{5}$$

$$P_3 = P_2 \times 2 \tag{6}$$

Step 6. Now, find the MVA and bus voltage Vi for each of the newly computed real power values—P2, P3 and P4.

Step 7. Now, use the cubic spline to trace the curve in between these points and obtain the P-V and the P-S curves.

Now, the area-to-area and point-to-point ATCs are calculated by obtaining the point where the voltage limit or the MVA limit line intersects the P-V and P-S curves, respectively.

Inference: The author of [1] has employed the proposed cubic spline-based method to compute the ATC of the Malaysian power system. The performance of the method is measured in terms of its accuracy and speed (which are the main reasons for the employment of the method). For testing the Malaysian power system, it has been simplified into a 241-bus system and further classified into three regions namely north, east, central, south and PUB. The lower and upper voltage limits are taken as 0.9 and 1.1 p.u., respectively. The results obtained have been compared with the traditional recursive AC power flow results.

It is observed from the results obtained in [1] that all the ATC values obtained are due to the MVA limitation. It is further observed that the performance of the cubic spline interpolation method is comparable to the AC power flow method in terms of accuracy. The observation of paramount importance is that while the proposed method obtains the high accuracy as found in the recursive AC power flow method, the time it requires to compute the ATC is much smaller than the AC recursive power flow method—to the extent that in some cases, the time required is up to 30 times lesser. Therefore, it may be concluded that the cubic spline method is superior to AC power flow in terms of speed and superior to DC power flow in terms of accuracy.

3. Electric arc furnace modeling

In this section, cubic spline is applied in modeling a very complex load—electric arc furnace (EAF). This example presents the true power of cubic splines in tracking extremely complex trajectories. The electric arc furnace's (EAF) ability to efficiently smelt scrap iron raw materials has made it the backbone of the steel-making industries. The EAF employs high temperatures produced by low-voltage and high-current electric arcs to smelt scrap iron raw materials. The increase in productivity requirements has led to EAFs being designed for high-power applications. The operation of EAFs introduces a significant amount of harmonics, inter-harmonics and flickers in the supply system. Therefore, it is mandatory that the operators pay much attention to the power-quality considerations. In order to study these problems, it is of paramount importance to understand the nonlinear load characteristics that the EAF present to the power system. Also, in order to mitigate the power-quality issues and to further study the impact of EAFs on the power system, it is required that the EAF be modeled after obtaining its time response.

The entire operation of the EAF involves three stages namely striking, smelting and refining. In the striking process, the electric arc is built up by lowering the electrodes of all three phases, the melting process involves the melting of the material and the process ends with the stable refining. Due to the complexities involved in modeling the EAF operation in the striking and the melting processes, most of the research has been directed towards a steady-state modeling of the EAF in the refining stage of operation. The modeling of an EAF requires several parameters such as arc voltage, arc length and arc current, which are determined by the position of electrodes. Therefore, in order to accurately model the EAF, we need to know the field measurements of the electric response which involves the variation of voltage and current. The measured responses are then employed to develop an EAF conductance model using the cubic spline interpolation method.

The EAF is modeled as a function of nonlinear conductance using the cubic spline interpolation technique which is called the cubic spline interpolation model (CSIM). In this method, a set of cubic polynomials are obtained which helps understand the voltage–current characteristic of the EAF.

The steps involved are as follows:

First, a set of 'n' measured data points of conductance is obtained for one fundamental cycle of operation. The measurements lying in the interval [a, b] such that $a = x_0 < x_1 < \ldots < x_n = b$. For the interval between two adjacent points, a cubic function is defined as shown in Eq. (7).

$$G_i\ (x) = a_i + b_i\ (x{-}x_i) + c_i\ (x{-}x_i)^2 + d_i\ (x{-}x_i)^3 \tag{7}$$

where i = 0, 1, 2,…., n-1. The coefficients $a_{i,}$ $b_{i,}$ c_i and d_i are unknown. These coefficients need to be determined based upon the following constraints:

Step 1. Each spline must pass through the given data points $y_{i.}$

$G_i(x_i) = y_i$ and $G_{n\text{-}1}(x_n) = y_n$

Step 2. Interior data points between each spline must be continuous.

$G_{i+1}(x_{i+1}) = G_i(x_{i+1})$

Step 3. The first and the second derivatives of the splines must be continuous across the interior data points. Therefore, the spline forms a smooth function.

$G'_{i+1}(x_{i+1}) = G'_i\ (x_{i+1})$.

$G''_{i+1}(x_{i+1}) = G''_i(x_{i+1})$

Step 4. In addition to the conditions mentioned in steps 1–3, another boundary condition must be satisfied which concerns the derivative of the functions at the boundaries (at x_0 = a and x_n = b). There are two types of boundary conditions that may be required to satisfy:

Natural boundary condition: $G''_0(x_0) = G''_{n-1}(x_n) = 0$.

Clamped boundary condition: $G'_0(x_0) = D_0$ and $G'_{n-1}(x_n) = D_n$.

where D_0 and D_n are the values of the first derivatives of the unknown functions.

It is generally found that the natural boundary conditions give less accurate results than the clamped boundary conditions. Alternately, one could possibly apply a boundary condition called not-a-knot condition, which in addition to the natural boundary condition also incorporates another condition that the third derivative of the function must be continuous at x_1 and x_{n-1}.

In order to find the coefficients—a_i, b_i, c_i and d_i—we follow the following steps and equations:

Step 1. Set $a_i = y_i$ for i = 0, 1, 2, …, n.

Step 2. Solve

$$\begin{aligned}&(h_0 + 2h_1)c_1 + (h_1 - h_0)c_2\\ &= \frac{3}{h_0 + h_1}\left[(a_2 - a_1) - \frac{h_1}{h_0}(a_1 - a_0)\right], \quad \text{for } i = 1\end{aligned} \tag{8}$$

and

$$\begin{aligned}&h_{i-1}c_{i-1} + 2c_i(h_{i-1} + h_1) + h_i c_{i+1}\\ &= \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}), \text{ for } i = 2, \ldots, n-2\end{aligned} \tag{9}$$

and

$$\begin{aligned}&(h_{n-2} - h_{n-1})c_{n-2} + (2h_{n-2} + h_{n-1})c_{n-1}\\ &= \frac{3}{h_{n-1} + h_{n-2}}\left[\frac{h_{n-2}}{h_{n-1}}(a_n - a_{n-1}) - (a_{n-1} - a_{n-2})\right],\\ &\text{where,} h_i = x_{i+1} - x_i, \text{ for } i = 1, 2, \ldots, n-1\end{aligned} \tag{10}$$

where $h_i = x_{i+1} - x_i$, for i = 1, 2, …, n-1.

Step 3. Set

$$c_0 = c_1 + \frac{h_0}{h_1}(c_1 - c_2), \tag{11}$$

$$c_n = c_{n-1} + \frac{h_{n-1}}{h_{n-2}}(c_{n-1} - c_{n-2}), \tag{12}$$

Step 4. Set

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(c_{i+1} + 2c_i), \text{ for } i = 0, 1, 2, \ldots, n-1 \tag{13}$$

Step 5. Set

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i), \text{ for } i = 0, 1, 2, \ldots, n-1 \tag{14}$$

The coefficients a_i can be found in step 1. The coefficients c_i can be calculated by solving the linear equations developed in Eqs. (8)–(10). Then, finally coefficients b_i and d_i may be obtained from steps 4 and 5 (Eqs. (13) and (14), respectively).

Inference: The author in [2] has analyzed the performance of CSIM on an actual power system model developed in MATLAB/SIMULINK whose online diagram is also presented in [2]. The results obtained via employing the CSIM (shown in **Figure 4**) are then compared with the results obtained via employing two traditional methods namely (i) harmonic current injection model (HCIM), shown in **Figure 2**, and (ii) harmonic voltage source model (HVSM), shown in **Figure 3**. The results described in [2] have been obtained for three different parts—early,

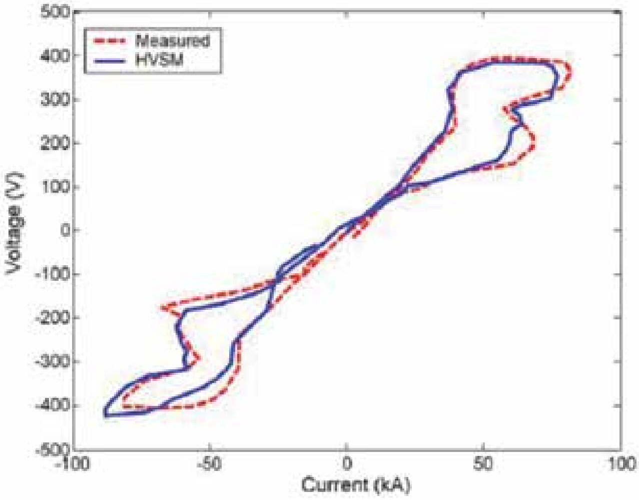


Figure 2. HVSM V-I characteristic [2].

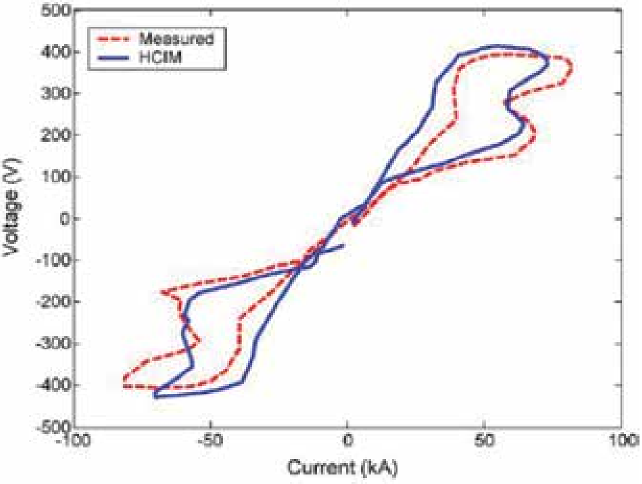


Figure 3. HCIM V-I characteristic [2].

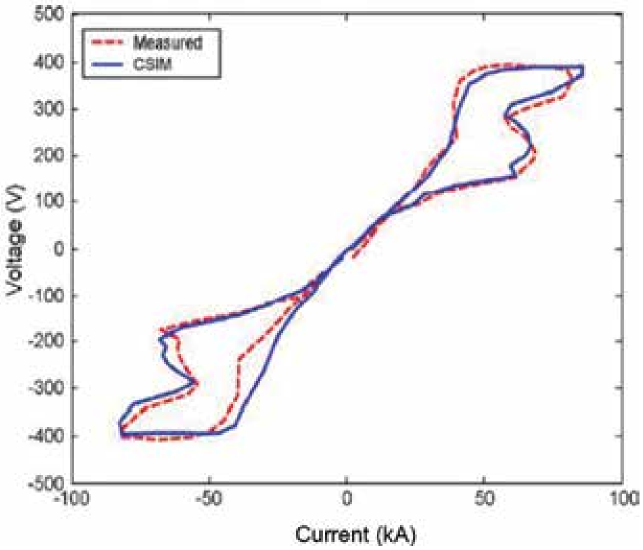


Figure 4. CSIM V-I characteristic (later stage) [2].

middle and later stages—of EAF operation during the refining period. This is done primarily because EAF does not possess a steady-state behaviour that lasts long. Since the HCIM and HVSM are unable to model dynamic behaviour, these methods are compared with the CSIM only during the latter part of the operation because of the dynamic behaviour of the EAF in the early and middle parts (**Figure 4**).

Upon employing the CSIM for EAF modeling in the early and middle stages, it is found that the performance of the EAF during these stages is highly nonlinear. The V-I curve presents a multi-valued function which makes it difficult to model the EAF. The results shown in [2] suggest that upon the comparison, the HCIM method yields the largest errors in terms of both the EAF voltage and current determined. Then, HVSM, which albeit an improvement on HCIM, still contains errors. The proposed CSIM method performs better than HCIM and HVSM and provides a better fit for the voltage and current characteristics. It is also explained in the results that the error encountered during modeling using CSIM in the early and middle stages is larger when compared to that encountered when modeling the later stages of the refining period because the number of sampling points used for the EAF model is not sufficient for modeling in the early and middle stages of refining. As an extension, it is also proposed in [2] that the cubic spline interpolation technique can also be used for modeling other nonlinear loads.

4. Var compensator with thyristor-controlled reactor

The problem of maintaining a good power factor (greater than 0.85) is a challenge faced by most of the industries. Some of these bulk consumers are penalized for operation under a low power factor. The application of cubic splines is presented to topic address this challenge and provide a commendable solution. Rolling mills and electric arc furnaces constitute very large loads in the power system. When the problem of power factor is considered, these loads (particularly electric arc furnaces) may be termed as a necessary evil because of their extremely low power factor. The variation in the arc length during operation results in the introduction of severe and rapid fluctuations in the reactive power and the voltage, and when the short circuit occurs, the power factor drops to values as low as 0.1. The large impact current and reactive power generated result in significant waste of energy and may also cause the power system to lose its stability. This may cause decadence in the quality of load and endanger the users. The solution to the abovementioned problem is cited in a process called reactive power consumption. Currently, the reactive power compensation is achieved dynamically by the placement of a dynamic reactive power compensation device at access points of such interference loads. These devices are usually represented as a fixed capacitor and thyristor-controlled reactor (FCT). The FCT enables smooth control of the reactive power and also has the desirable feature of maintaining its voltage unchanged. In addition, the FCT can effectively suppress voltage fluctuation and solve the voltage distortion and flicker problem and improve the power quality. The continuous and smooth variation in the reactive power is obtained by the variation of the thyristor conduction angle. For the FCT to effectively carry out its task, it is imperative that it calculates the control angle quickly and accurately in a real-time environment. However, the existence of a nonlinear relation between the control angle and the reactor amplification factor makes the real-time calculation a cumbersome process.

The remainder of this section reviews the calculation and the corresponding control of the control angle via employing cubic spline interpolation technique. The calculation of the control angle, α, is based on the reactor amplification factor for each phase. Cubic spline interpolation is employed because it provides a low-order polynomial interpolation polynomial and also increases the smoothness of the interpolation function.

The basic thyristor-controlled reactor (TCR) consists of a pair of anti-parallel thyristors in series with an inductor as shown in **Figure 5**. The thyristor delay angle varies between 90° and 180°. As a result, the fundamental current is completely reactive. An increase in the delay angle leads to a decrease in the fundamental reactive current, which is equivalent to increasing the reactance or reducing the susceptance and hence results in a decrease in the reactive power. Hence, the TCR can be seen as being equivalent to a variable susceptance which can be controlled using the delay angle 'α'. This is because the AC voltage remains constant but the

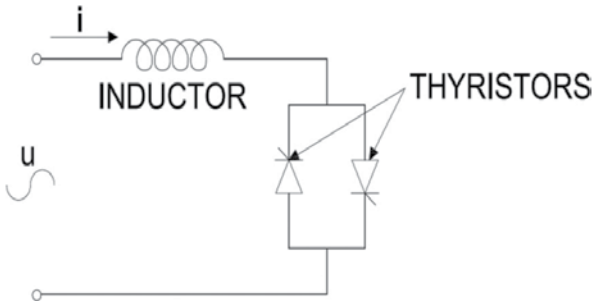


Figure 5. Thyristor controlled reactor.

value of the fundamental current changes, which in turn results in the variation in reactive power. The equivalent fundamental susceptance of the TCR is given in Eq. (15)

$$\mathrm{B_L}(\alpha) = \frac{2\pi - 2\alpha + \sin 2\alpha}{\pi X_{\mathrm{L}}} \tag{15}$$

The relationship between the delay angle, α, and the amplification factor of the equivalent impedance is shown in Eq. (16)

$$\mathrm{k} = \frac{\pi}{2(\pi - \alpha) + \sin 2\alpha} \tag{16}$$

As can be seen, there exists a nonlinear relation between the delay angle and the amplification factor. The application of cubic splines makes the control of delay angle with reactive power variation easier, quicker and smoother. It must be observed that the inductance current in the TCR does not depend on the inductance but is rather governed by the thyristor conduction angle. The accurate control of the thyristor angle makes the accurate control of inductor current and hence reactive power possible.

The problem is formulated as follows. Eq. (9) suggests that in order to control the delay angle using the amplification factor 'k', we need to find a solution to the nonlinear equation. Albeit possible, it is a cumbersome task to solve nonlinear equations, not to mention the considerable amount of time the solving takes which make real-time application and control a tedious task. Moreover, of paramount importance is the accuracy of the controlling action since it determines the reactive power compensation. The cubic spline interpolation is employed as a means to calculate the control angle 'α' having known the amplification factor 'k'. Cubic spline is used as the method of interpolation because of the advantages it provides in terms of simplicity of calculation, numerical stability and smoothness of the interpolated curve.

As previously stated, the existence of a nonlinear relation between the control angle and the amplification factor makes the real-time computation a lengthy process. In order to achieve the aim using cubic spline interpolation, the control angle is regarded as the dependent variable and the amplification factor 'k' as the independent variable. According to [3], the author generated 158 data points using Eq. (2). The aim is now to generate 157 interpolating cubic polynomials that fit in between these points. The form of these cubic polynomials is shown in Eq. (17)

$$\alpha = \mathrm{a_0}(\mathrm{k} - \mathrm{k_i})^3 + \mathrm{a_1}(\mathrm{k} - \mathrm{k_i})^2 + \mathrm{a_2}(\mathrm{k} - \mathrm{k_i}) + \mathrm{a_3} \tag{17}$$

In Eq. (10), i = 1, 2, …., 157 and $\mathrm{k_i} \le \mathrm{k} \le \mathrm{k}_{\mathrm{i}\,+\,=1}$.

The fitting coefficients $\mathrm{a_0}$, $\mathrm{a_1}$, $\mathrm{a_2}$ and $\mathrm{a_3}$ need to be computed for each interval which gives a unique cubic polynomial for each interval. When carrying out the control procedure first, the amplification is determined, and then the corresponding cubic polynomial is used to arrive at the required conduction angle value.

Inference: The results shown in [3] suggest that the conduction angle determined by using cubic splines shows significant match with the values obtained by simulation. Also, the use of cubic splines yields quicker results—a trait which would be beneficial for real-time applications.

Hence, it can be said that the cubic spline interpolation method is able to solve the problem of quick computation of the conduction angle with accuracies relevant to engineering applications.

5. Estimation of voltage stability margin

In this section, the application of cubic splines is of paramount importance since it marks the limit of a power system—the maximum amount of power that can be supplied before the continuous uncontrolled drop of voltage. The impact of voltage instability on the power system has been so damaging that significant research has been conducted in this direction. The phenomenon of voltage instability is characterized by a sudden and uncontrollable drop in voltage as a response to a disturbance that has occurred on the power system. This disturbance could be anything from the variation of load to loss of a line or lightning strike, and so on. Most of the studies have been focused on the steady-state aspect of voltage stability. In order to determine the proximity of the system to voltage collapse, we need to estimate or find the voltage stability margin (VSM). Many methods have been proposed in the past to estimate the steady-state voltage stability margin. One of the popular methods is the continuous power flow method. A major disadvantage of the continuous power flow is that it requires a considerable amount of time and hence it cannot be employed in real-time applications.

An alternative approach is to use the time-synchronized voltage and current phasor measurements obtained from phasor measurement units (PMUs). In the presented method, the author proposed a combination of a coupled single-port Thevenin equivalent model and cubic spline extrapolation in order to find the point of voltage collapse or the voltage stability margin (VSM). The concept is based on the fact that the voltage collapse point of the load impedance equals the Thevenin equivalent impedance.

Thevenins' equivalent voltage and equivalent impedance as seen from load bus 'i' as given in **Figure 6** can be written as shown in Eq. (18)

$$\mathrm{E_{thi}} = [\mathrm{K}\mathbf{V}_\mathrm{G}]_\mathrm{i} \; - \sum_{\mathrm{j=1,\quad j\neq i}}^{\mathrm{n}} \mathrm{Z_{ij} I_{Lj}} \tag{18}$$

$$\mathrm{Z_{thi}} = \mathrm{Z_{ii}}$$

The definitions for the abovementioned variables can be found in [4].

The load impedance Z_{Li} of bus 'i' can be arrived by using Eq. (19)

$$\mathrm{Z_{Li}} = \frac{\mathrm{V_{Li}}}{\mathrm{I_{Li}}} \tag{19}$$

The load bus voltage V_{Li} and the load bus current I_{Li} are obtained through the PMU measurements.

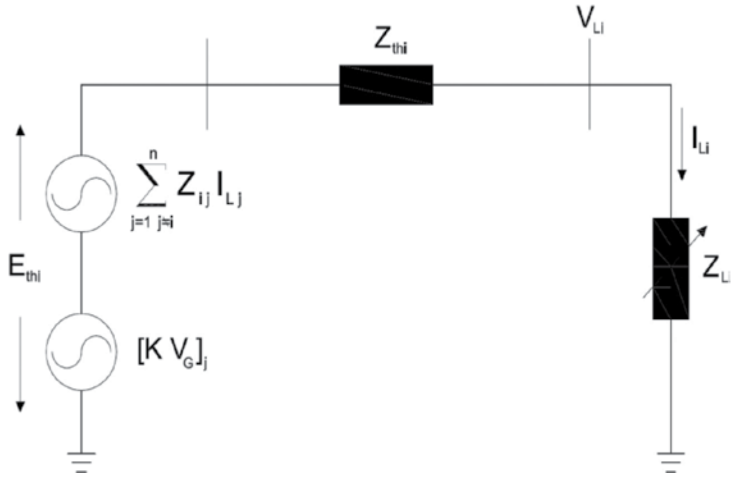


Figure 6. Coupled single port Thevenins' equivalent.

Figure 7 shows the variation of Z_{li} and Z_{th} as a function of the load parameter λ. As can be seen, the Z_{li} equals the Z_{thi} at the point of maximum critical loading, and this point gives the maximum value of the load parameter—λ_{max}. This maximum value of load parameter can be arrived at by equating an approximate function that extrapolates the Z_{li} versus λ curve to the point that it meets the Z_{thi} line.

The choice of cubic spline extrapolation is justified by the author as the superior fitting to the impedance trajectory as evidenced by extensive simulation results. The cubic spline extrapolation proceeds by developing different cubic polynomials for the interval between measurements based on certain constraints.

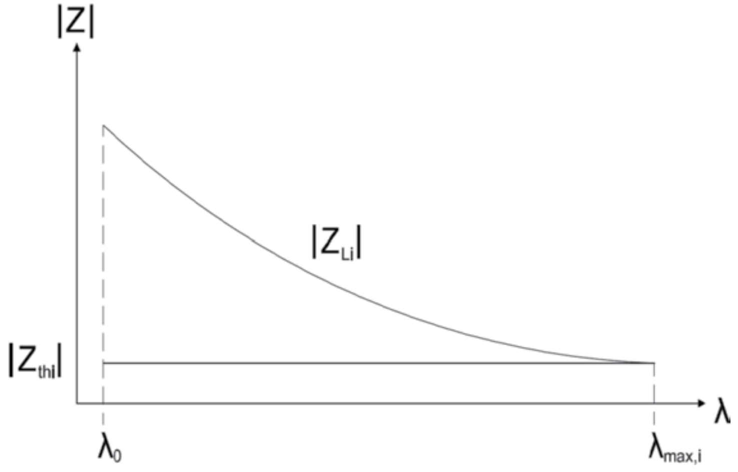


Figure 7. Variation of Thevenin and load impedance with λ [4].

For 'm' PMU-measured data points obtained, there will be 'm-1' intervals present, each of which is represented by Eq. (20)

$$f_j\ (x) = a_j + b_j\ \left(x - x_j\right) + c_j\ \left(x{-}x_j\right)^2 + d_j\ \left(x{-}x_j\right)^3 \tag{20}$$

The inputs for extrapolation are taken as three sets of $|Z_{Li}|$ and λ and the point of $|Z_{li}|$ where λ is to be extrapolated and found. The λ thus found corresponds to the point where Z_{li} equals $Z_{th\mathrm{i}}$ and hence is equal to λ_{maxi}. Each bus in the considered system will have its own corresponding λ_{max}s. The $\lambda_{\mathrm{max}} = \lambda_{\mathrm{sys}}$ for the entire system is found by taking the minimum of all λ_{maxi} obtained for each of the individual load busses. This value of λ_{sys} corresponds to the value of λ_{maxi} for the weakest load bus in the system. The proximity of the load bus to the voltage collapse point is given in Eq. (21)

$$\mathrm{VSM} = \frac{\lambda_{\mathrm{sys}} - \lambda_0}{\lambda_0} \times 100 \tag{21}$$

After the computation of VSM, the author of [4] proceeds to find an index which helps in determining whether the load increase results in the violation of the reactive power limits at the generator busses. The author begins with the simple power equation and arrives at the equation of a surface given by Eq. (22)

$$Q_G^2 - \frac{|V_G|^2}{X} Q_G + \frac{|V_G|^2}{X} Q_L + P_L^2 = 0 \tag{22}$$

where P_L, load real power; Q_L, load reactive power; Q_G, generator reactive power.

The surface defined lies in the (P_L, Q_L, Q_G) space. Upon cutting the surface with the constant power factor planes, we get the P_L–Q_G curves which are similar to PV curves or λV curves. The existence of an approximate quadratic relation between λ and V is extended to find a quadratic relation between λ and Q_G. This approximate quadratic relation is modeled and given in Eq. (23)

$$\lambda = \alpha_i Q_{Gi}^2 + \beta_i Q_{Gi} + \gamma_i \tag{23}$$

In Eq. (16), α_i, β_i and γ_i are parameters that are different for each generator bus and need to be determined. The parameters can be determined using three sets of PMU readings. Then, using the computed values of the parameters, an estimate for the extreme of Q_{Gi} is arrived using the condition that at the extreme limit $d\lambda/dQ_{Gi} = 0$ holds. Enforcing this condition gives us the following value as the estimate shown in Eq. (24)

$$Q_{Gi}^{ex} = -\frac{\beta_i}{2\alpha_i} \tag{24}$$

The index i stands for all the generator busses of the system. At each load step, it is ensured that the estimated value of generator reactive power—$Q\ _{Gi}{}^{ex}$—remains within the bounds of the reactive power limits that are specified for a generator bus.

The algorithm for proceeding with the described method in order to compute the VSM is as follows:

Step 1. Obtain three sets of PMU measurements and use them to compute three sets of $|Z_{Li}|$, λ and Q_{Gi}. Also, we possess the knowledge of the admittance matrix.

Step 2. Compute the parameters α_i, β_i and γ_i for each of the generator bus of the system.

Step 3. Compute the estimate of the extreme of the reactive power for each generator Q_{Gi}^{ex}

Step 4. If there is no violation of the reactive power limits of any generator, then proceed to step 6. Otherwise, proceed to step 5.

Step 5. Change the bus type from PV to PQ bus for the bus whose reactive power limit has been violated.

Step 6. Compute the impedance matrix as shown in [4].

Step 7. Compute the Thevenin equivalent impedance.

Step 8. Estimate the value of λ_{maxi} by using the cubic spline extrapolation technique for each load bus of the system.

Step 9. Find the value of λ_{sys} which is the minimum of all values computed in step 8.

Step 10. Determine the VSM.

Inference: The author of [4] has performed test runs of the proposed algorithm on the following test system—IEEE 30 bus system, IEEE 118 bus system and IEEE 300 bus system. Further, the obtained results have been compared with two other previously available methods. The obtained results indicate the superior performance of the proposed cubic spline technique when compared to the other methods. Upon comparison of the percentage error, it may be observed from [4] that the cubic spline method is almost 10 times more accurate than the other methods.

6. Estimation of market power in deregulated electricity market

In this final section, the application of cubic spline technique has been illustrated in the field of deregulated electricity market. In this market, market power issues predominantly spoil the basic idea of maintaining equilibrium within the market players.

Market power is the ability of showing one's monopolistic nature on the price of the commodities in the market. This has become a challenging issue in the context of the present electricity market and will become more challenging and play a significant role when private generation companies start participating in buying/selling the power [5]. Due to the increase in demand and the regulatory policies, private parties have started investing in the power sector, especially in the renewable energy sources. Thus, it is inevitable for the independent system operator (ISO) to estimate market power for taking crucial decisions [6].

6.1. Market power indices

6.1.1. Herfindahl-Hirschman index

The Herfindahl-Hirschman index (HHI) is used to measure the market concentration that will reflect the number of players in the market and also the inequality in their market shares. The HHI is defined as the sum of the squares of market shares of all the players as given in Eq. (25)

$$\mathrm{HHI} = \sum_{i=1}^{N} S_i^2 \mathrm{HHI} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{S_i}^2 \tag{25}$$

where N is the number of players and $\mathrm{S_i}$ is the $\mathrm{i^{th}}$ player market share in percentage [7].

6.2. Lerner index

It measures or indicates the proportional deviation of the price at the firm's profit-maximizing output from the firm's marginal cost at that output. It is defined as shown in Eq. (26)

$$\mathrm{LI_i} = \frac{(\rho_\mathrm{i} - \mathrm{mc_i})}{\rho_\mathrm{i}} = \frac{1}{\varepsilon_\mathrm{i}^\mathrm{d}} \mathrm{LI_i} = \frac{\rho_\mathrm{i} - \mathrm{mc_i}}{\rho_\mathrm{i}} = \frac{1}{\varepsilon_\mathrm{i}^\mathrm{d}} \tag{26}$$

where LI_i is the Lerner index for a given firm i, ρ_i and mc_i are the price and marginal cost, respectively, and $\varepsilon_\mathrm{i}^\mathrm{d}$ is the elasticity of demand felt by the firm.

6.3. Nodal must run share

This index reflects the impact of load variation on market power and geographic difference of market due to network constraints. Must run share (MRS) represents the effect of load variation and nodal must run share (NMRS) represents the geographical difference of market powers. The equation for NMRS is shown in Eq. (27)

$$\mathrm{NMRS_{k,i}} = \frac{\mathrm{Pg_{k,i}^{must}}}{\mathrm{Pd_i}} \mathrm{NMRS_{k.i}} = \frac{\mathrm{Pgmust_{k,i}}}{\mathrm{Pd_i}} \mathrm{i} = 1, 2, ..\mathrm{N} \tag{27}$$

where N is the number of busses in a power system, $\mathrm{Pd_i}$ is the load at bus i, and $\frac{\mathrm{Pg_{k,i}^{must}}}{\mathrm{Pd_i}} \mathrm{Pgmust_{k,i}}$ is the contribution of the must run generator k to $\mathrm{Pd_i}$. The background calculation of NMRS is available in Ref. [7].

Steps involved in estimation of market power using NMRS [5, 8].

Step 1. Define the number of generators and their active power limits.

Step 2. Determine Pg_k^{must} of generator k

Step 3. Calculate distribution matrix $[M^{-1}]$

Step 4. Calculate NMRS of generator 1 on load 1.

Step 5. Repeat step 4 for calculating NMRS for the remaining generators on each load.

Step 6. Repeat steps 4 and 5 for various cases.

Step 7. End.

Step 8. Calculate the NMRS for all the generators using the conventional method as discussed in steps–7.

Step 9. Plot NMRS of a generator against the maximum generation of other generators.

Step 10. Connect two points at a time using cubic spline interpolation technique (piecewise polynomial) using MATLAB built-in function, that is, for example, *a = spline (b, c, de)* and *ff = spline (b, c)* where *'a'* gives the interpolated values which correspond to the query points in *de* [9].

The application of cubic's spline interpolation for calculating the respective market power of generation companies on all busses for any given load or operating condition is implemented on a sample IEEE 14 bus test system. This system consists of four generator busses and an additional slack generator at bus number 1 in addition to the nine load busses. The interconnection of the system is accomplished with 20 transmission lines.

In **Figure 8**, the operating condition is such that the total load is increased by 10% from the base case. The fast and accurate calculation of the market power at any given operating point of the generator may serve or help the principal purposes of various bodies like the GENCOs, DISCOs and ISOs. The knowledge of market power for any given operating condition will enable these above bodies to take suitable actions to change their own operating points in

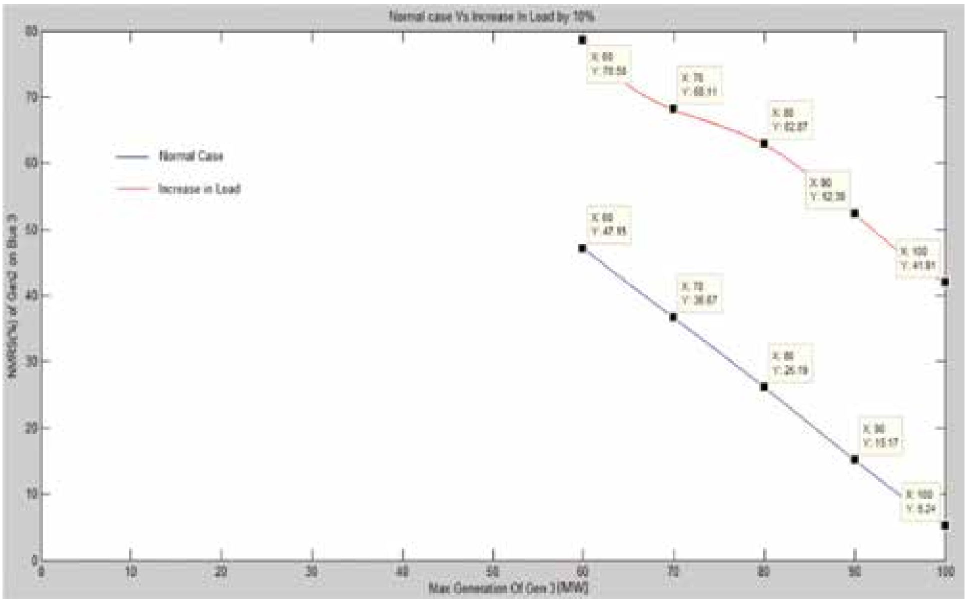


Figure 8. Normal case versus increase in load by 10%.

order to fulfill their objectives. **Figure 8** represents the interpolation plot for the case when load is increased by 10%.

The interpolation is done between the five known operating points which are the maximum generation levels of Gen_3, starting from 60 MW in steps of 10 MW. For these operating points, the NMRS values are calculated. As we increase the maximum generation level of a particular generator, it affects the NMRS of its neighboring generators. In this case, we have monitored the NMRS of Gen_2 on bus 3 with the increase in the working range of Gen_3 to demonstrate this fact. As a result, the base case plot for Gen_2 tends to drop or go down as we keep increasing the generation level values for Gen_3. As the load is increased by 10%, the NMRS plot shifts up. Thus, it represents an important observation that as the load goes up at a certain time of day in a particular region, then the GENCOs which lie in such a subsystem will have an increased market power. Due to the piecewise polynomials which are attained due to the application of cubic's spline interpolation between the operating points, GENCOs, DISCOs and ISOs can easily take suitable actions to not let anyone take undue advantage of the varying market power due to changes in operating conditions. Thus, it helps in attaining a zero market power. This work has been extended under various system conditions and an elaborate study is made in [10].

Inference: Market power reflects the amount of influence that a company has on the system in which it operates but in power systems Market power is the ability to maintain prices above the competitive levels for a significant period of time. Hence, it is of utmost importance to find the market power of the system under normal and abnormal condition that a system has to face.

7. Conclusions

In this chapter, a complete literature review is made on how cubic spline interpolation technique has been widely used in power systems application. ATC calculation, electric arc furnace modeling, static var compensation, voltage stability margin and market power estimation are some of the areas where cubic spline interpolation techniques are extensively used. The independent system operator (ISO), utilities and consumers can utilize this tool to predict the behaviour of the system/generation companies/utilities and bring back the system towards economic/system stability.

Author details

Akhil Prasad[1], Atul Manmohan[1], Prabhakar Karthikeyan Shanmugam[2]* and Kothari D.P.[2]

*Address all correspondence to: spk25in@yahoo.co.in

1 Larsen and Toubro Construction, Faridabad, India

2 VIT University, Vellore, Tamil Nadu, India

References

[1] Othman MM, Mohamed A, Hussain A. Fast evaluation of available transfer capability using cubic-spline interpolation technique. Electric Power Systems Research. 2005;**73**:335-342

[2] Liu Y-J, Chang GW, Hong R-C. Curve-fitting-based method for modelling voltage–current charact.Eristic of an ac electric arc furnace. Electric Power Systems Research. 2010;**80**:572-558

[3] Niu Y, Zhang D, Yixin S, and Zhang H. Application of cubic spline interpolation in Var compensator with Thyristor controlled reactor. In: IEEE, editor. Chinese Association of Automation (YAC), Youth Academic Annual Conference of; 11–13 Nov. 2016; Wuhan, China. IEEE Xplore; 2017. DOI: 10.1109/YAC.2016.7804924

[4] Heng-Yi S. Chih-wen Liu, fellow. Estimating the voltage stability margin using PMU measurements. IEEE Transactions on Power Systems. July 2016;**31**(4):3221-3229. DOI: 10.1109/TPWRS.2015.2477426

[5] Prabhakar Karthikeyan S, Sahoo SK, Roopesh R, Mathur S, Jacob Raglend I, Kothari DP. Utilizing FACTS Devices in Enhancement of Market Power under Congestion- A threat to the Deregulated Electricity Market. In: India Conference (INDICON), 2012 Annual IEEE; Kerala, India. IEEE Xplore; 2012. pp. 299-303

[6] Prabhakar Karthikeyan S, Jacob raglend I, Kothari DP. Impact of FACTS devices on exercising market power in deregulated electricity market. Frontiers in energy. Springer Publication. 2013;**7**(4):448-455

[7] Wang P, Xiao Y. Nodal market power assessment in electricity markets. IEEE Transactions on Power Systems. August 2004;**19**(3):1373-1379

[8] Prabhakar Karthikeyan S, Jacob raglend I, Kothari DP. A review on market power in deregulated electricity market. International Journal of Electrical Power and Energy Systems. June 2013;**48**:139-147

[9] https://in.mathworks.com

[10] Ahmed S, Fathima F, Prabhakar Karthikeyan S, Sahoo SK, Rangarajan SS, Kothari DP. Application of cubic spline interpolation in estimating market power under deregulated electricity market. In: IEEE, editor. TENCON 2015, IEEE Region 10 conference; 1–4, November 2015; Macau, SAR China. 2016

Edited by Young Kinh-Nhue Truong and Muhammad Sarfraz

Splines provide a significant tool for the design of computationally economical curves and surfaces for the construction of various objects like automobiles, ship hulls, airplane fuselages and wings, propeller blades, shoe insoles, bottles, etc. It also contributes in the description of geological, physical, statistical, and even medical phenomena. Spline methods have proven to be indispensable in a variety of modern industries, including computer vision, robotics, signal and image processing, visualization, textile, graphic designs, and even media.

This book aims to provide a valuable source on splines and their applications. It focuses on collecting and disseminating information in various disciplines including computer-aided geometric design, computer graphics, data visualization, data fitting, power systems, clinical and epidemiologic studies, disease detection, regression curves, social media, and biological studies. The book is useful for researchers, scientists, practitioners, and many others who seek state-of-the-art techniques and applications using splines. It is also useful for undergraduate senior students as well as graduate students in the areas of computer science, engineering, health science, statistics, and mathematics. Each chapter also provides useful information on software developments and their extensions.

Published in London, UK
© 2018 IntechOpen
© Phuchit / iStock

IntechOpen

ISBN 978-1-83881-350-5
9 781838 813505