

Article

Data Interpolation by Optimal Splines with Free Knots Using Linear Programming

Lakshman Thakur¹ and Mikhail A. Bragin^{2,*}

¹ Department of Operations and Information Management, School of Business, University of Connecticut, Storrs, CT, 06269; thakur@uconn.edu

² Department of Electrical and Computer Engineering, School of Engineering, University of Connecticut, Storrs, CT, 06269; mikhail.bragin@uconn.edu

* Correspondence: mikhail.bragin@uconn.edu

Abstract: Studies have shown that in many practical applications, data interpolation by splines leads to better approximation and higher computational efficiency as compared to data interpolation by a single polynomial. Data interpolation by splines can be significantly improved if knots are allowed to be free rather than at a priori fixed locations such as data points. In practical applications, the smallest possible curvature is often desired. Therefore, optimal splines are determined by minimizing a derivative of continuously differentiable functions comprising the spline of the required order. The problem of obtaining an optimal spline is tantamount to minimizing derivatives of a nonlinear differentiable function over a Banach space on a compact set. While the problem of data interpolation by quadratic splines has been accomplished analytically, interpolation by splines of higher orders or in higher dimensions is challenging. In this paper, to overcome difficulties associated with the complexity of the interpolation problem, the interval over which data points are defined, is discretized and continuous derivatives are substituted by their discrete counterparts. It is shown that as the mesh of the discretization approaches zero, a resulting near-optimal spline approaches an optimal spline. Splines with the desired accuracy can be obtained by choosing an appropriate mesh of the discretization. By using cubic splines as an example, numerical results demonstrate that the linear programming (LP) formulation, resulting from the discretization of the interpolation problem, can be solved by linear solvers with high computational efficiency and resulting splines provide a good approximation to the optimal splines.

Keywords: Optimal Splines; Linear Programming; Data Interpolation; Splines with Free Knots

1. Introduction

Many applications require piecewise polynomial functions, usually referred to as “splines,” which interpolate given data points and satisfy additional conditions such as continuity, smoothness, and convexity or concavity. The recent and the ongoing COVID-19 pandemic, spurred advances in spline-related research, specifically, splines have been shown to be useful for modeling and forecasting of the COVID-19 spread [1–4], and COVID-19 trend [5]. In [6], splines were used to analyse the relationship between the ambient temperature and the transmission of COVID-19. Moreover, splines were shown to be useful for capturing the nonlinear relationship between “fasting blood glucose” levels and risk of intensive care unit admission for COVID-19 patients [7] unlike other conventional linear, dichotomous, or categorically methods. Other traditional and recent applications of splines include: 1) dynamic programming [8–27], 2) mathematical programming [28–36], 3) statistics [37–51], 4) control theory [50–68], 5) computer graphics [97–120], and 6) path planning [51] as discussed below.

When an exact functional form in a model is unknown, it is often described by measurements represented by data points. Traditionally, data points were interpolated by polynomials more frequently than by spline functions. While polynomials have many desirable properties, polynomial interpolation suffers from the Runge’s Phenomena [69] - a polynomial oscillation at the edges of an interval. With large data, the derivatives of a single polynomial at each of the data points tend to grow thereby resulting multiple



Citation: Lakshman Thakur; Mikhail A. Bragin 0000-0002-7783-9053 Title. *Preprints* 2021, 1, 0. <https://doi.org/>

Received:

Accepted:

Published:

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

local extrema. In particular, it has been shown that polynomials, extensively used in thermometry [70], are not capable of recovering local behavior with good quantifiable accuracy. Spline interpolation, on the other hand, tends to greatly reduce oscillation between data points and (cubic) splines have been shown to be able to reproduce local characteristics well [70]. In addition, the problem of finding a polynomial interpolating a large number of data points is equivalent to inverting a van der Monde matrix with a large condition number thereby rendering the interpolation problem intractable, inefficient and unstable in computer implementations. On the other hand, the great flexibility of splines, resulting from the use of several lower degree polynomials smoothly connected at optimally chosen points (usually referred to as “knots”), resolves the major problems mentioned above effectively. This is why splines are turning out to be so useful in a wide variety of applications.

The aim of this paper is two-fold: 1) to give a perspective on applications of spline functions in Mathematical Programming, Dynamic Programming, Statistics, Optimal Control Theory, and Computer Graphics in Section 2; and 2) to formulate a linear optimization program that finds near-optimal curvature of splines with free knots interpolating a given set of data points efficiently and with the desired level of accuracy in Section 3. Section 4 provides numerical results, presenting computational efficiency of the method developed. Section 5 gives brief concluding remarks.

2. Literature Review

Spline functions were introduced as early as in the nineteenth century by Lobachevsky [71, p.165]. In the mid 40's, the B-splines were investigated by Schoenberg [72]. Since then the subject has grown vigorously and many excellent texts are available on the subject: [37,73–78]. A good introduction is given in [74]. The early computational algorithms have been developed and are available in [75]. A comprehensive theory can be found in [76], and a briefer treatment in [37]. Reference [38] finds [77] “one of the most useful such volumes.” A general survey can be found in the overviews given in these texts. Moreover, the selection of topics largely depends on the application area of the reader's interest.

In a basic way, spline theory deals with a simple but important question of how one uses a *lower degree piecewise* polynomial function rather than a *single larger degree* polynomial to represent given data and attendant properties. Spline theory attempts to obtain appropriate problem formulations, existence and characterization of solutions, and properties and computation of such functions.

Representation of the function with lower degree polynomials, facilitation of efficient computation, as well as good approximation of the original function between the given data points being interpolated, continuous or essentially *bounded derivatives* (often second and third derivatives), are some of the important assumptions in spline applications. For example, second derivative is used as a measure of nonlinearity of models such as in convex separable programming [28,29], and nonlinear statistical estimation problems [79]. Boundedness of the third derivative in interpolation by cubic splines is natural since one can always have a function interpolating the given point with arbitrarily high curvature. Therefore, to estimate the most favorable possible error in function approximation, one would like to determine the smallest curvature possible.

One of the first books on the approximation by least squares cubic splines was published in 1968 by de Boor and Rice [80] and [81]. Studies have shown that the approximations by splines can be significantly improved if knots are allowed to be free rather than at a priori fixed locations (e.g., data points). When the knots of a spline are fixed a priori (for example, knots coincide with data points) we call it a spline with fixed knots. When the knots are not fixed but are determined optimally it is a spline with free or variable knots. Generally, as one would expect, the analysis for fixed knots tends to be simpler than for free knots [37] (p. 190). A spline is required to pass through the data points exactly, when data are fairly accurate, is called an *interpolatory* spline, otherwise an *approximating* spline. The methods given in this section focus on optimal interpolatory cubic splines

with free knots. A comparative analysis showing advantages of free-knot splines over data-smoothing techniques has been given in [82]. Typically the degree of smoothness is controlled by both the number and the position of the knots. In [83] the knots are estimated by solving a non-linear least-squares optimization problem.

It has been shown in [39] that there always exists an optimal spline with $\|f^{(r)}\|_{\infty} = k^*$ for each polynomial comprising the spline and $(n - r - 1)$ knots, where n is the number of data points and r is the degree of the spline that interpolates them. It is noted that for a given value k^* , an optimal spline with $\|f^{(r)}\|_{\infty} = k^*$ is not necessarily unique (since some of the polynomials comprising the spline (but not all) that interpolate given data points may have $\|f^{(r)}\|_{\infty}$ that is less than the optimal value k^*).

Supported by substantial mathematical theory and effective computational methods, splines have a very wide spectrum of general applications. Applications of splines range from modeling insect natality and DNA coils in biology [84,85], pattern recognition of handwritten Japanese in linguistics [86], study of geomagnetism in physics [87], analysis of liquid alloys in chemistry [88], characterization of vibrations in acoustics [89], to solving spectroscopic orbits in astronomy [90].

Among many of these, cubic splines have also come in common use in multiple disciplines such as robotics and statistics. Cubic splines are used in the problem of coordination and planning of the robotic motion [91]. The trajectories for robot motion are frequently described by smooth cubic splines [92]. Cubic splines are used to characterize the dependency of the Markov model parameters on the conditional parameters [93]. Many filtering techniques require the use of splines. Splines are used to extract curves to model approximation to the shapes of segmented images that result from smoothing the lines of an object in the presence of its shape uncertainty [94]. In digital signal processing, splines are used to generate new sequences with a higher sampling rate (up-sampling) [95]. Numerical shape-preserving quadratic splines were developed in [96].

2.1. Spline Applications

Spline applications include areas of mathematical programming, dynamic programming, engineering, statistics, and optimal control theory that will be discussed below in detail.

2.1.1. Dynamic Programming Applications

Dynamic programming has had multiple important applications in the area of sequential decision making. The ability to handle nonlinearities and stochastic nature of functional relations describing the model, empower it with great versatility. However, equally well-known is the “curse of dimensionality.” As the dimension of the state space and the number of decision stages increases, the computational burden grows enormously and becomes computationally prohibitive very quickly. In such a case, to use the dynamic programming methodology, approximation of the space by discretization at certain grid points is often necessary. Clearly, the greater the number of the grid points, the better the approximation and the larger the computational effort. We briefly describe here a stochastic, continuous state space water reservoir model in water/power resource management, where the use of cubic splines instead of a more common piecewise linearization leads to rather impressive computational performance.

Such models have been used for real world problems in Shasta/Trinity system in Northern California, Brazil’s large hydropower system, and Egypt’s High Aswan Dam [10,11]. For an important and well discussed 4-reservoir, 3-period water release problem [12–14] the computational time to obtain an optimal decision was reduced very significantly. For this problem, the number of grid points varying from 3 to 17 was used to discretize the values of functions representing starting volume of water, the inflow of water, released amount of water at time t and related variables. Rather than the usual piecewise linear interpolation, cubic splines with continuous second derivatives were used for this purpose.

Since (1) cubic polynomials allowed for better approximation, and (2) the continuity of the second derivative made possible the use of Quasi-Newton methods in the solution of the subproblems, very significant computational savings have been achieved. For a 1% error, the spline dynamic programming algorithm reduced the computational time by a factor of 255. For a 0.5% error the reduction factor was even greater: 330 [15]. For practical reservoir problems with more than 2-3 dimensions, while linear interpolation needs prohibitive amounts of computation time for reasonable accuracy, the use of cubic splines for 5-dimensional problems led to significant computational improvement and an overall advancement in the area [15]. The more traditional approaches for containing a sprawling dynamic program include the use of principal component analysis, sampling, duality, Benders decomposition, substitution of deterministic equivalents for stochastic components, and partition of the original problem into separable components [15–20].

There are many other problems that are modeled as continuous state space, including manufacturing process control, management of forests, fisheries, crops pest control, and portfolio selection under fluctuating interest rates [8,9,21–27]. Though realistic parameters would be different in these applications, the use of splines would offer a promising strategy to estimate them.

2.1.2. Mathematical Programming Applications (Estimating the objective function and solution error for piecewise linear approximations)

Piecewise linear functions often appear in optimization largely because of their increased mathematical tractability, and “one of the most useful applications of piecewise linear representation is for approximating nonlinear functions” [33, p. 382]. Since approximations obviously affect the computed quantities of interest (e.g., the optimal objective value), how do we measure the quality of this approximation quantitatively beyond the common assertion that ‘the greater the number of the linear pieces, the better the approximation’, and how do we determine the errors in the values of interest? Optimal splines help find a bound on this error. Found by computing an optimal quadratic spline, the minimal required curvature value k^* of $|f^2|$ of a smooth function taking specified values at the given points, will provide a bound on the error in such approximations. For example, if \hat{f} is the piecewise linear function resulting from joining the adjacent points, $\{(x_i, y_i)\}_{i=1}^{r+2}$ obtained, say, from an experiment [52, p. 171] or a function from its values at $\{x_i\}$ ’s, then k^* gives a bound on the function error $\max_{a \leq x \leq b} |f(x) - \hat{f}(x)|$ [34] that can be shown to be bounded by:

$$\max_{a \leq x \leq b} |f(x) - \hat{f}(x)| \leq k^* \delta^2 / 8,$$

where $\delta = \max_{1 \leq i \leq r-1} (x_{i+1} - x_i)$. Since we have determined $k^* \leq |f^2|$ for any admissible function f , one specific use of such a solution k^* is in the computation of error bounds on the optimal objective value and optimal solution (vector) of convex separable programs. Similarly, considering a convex separable program Z with a nonempty feasible space and its piecewise linear approximation Z_1 , with a subdivision interval δ , the bounds on the maximum deviation between the optimum objective values Z_1^* and Z^* can be calculated by using quantities

$$D_j = \max_{0 \leq x_j \leq r_j} |f_j(x_j) - \hat{f}_j(x_j)|,$$

$$E_{ij} = \max_{0 \leq x_j \leq r_j} |g_{ij}(x_j) - \hat{g}_{ij}(x_j)|,$$

where D_j, E_{ij} values are calculated by the solutions of the quadratic splines as k^* discussed above. Similarly, the method to compute bounds on each component of the optimal solution vector X^* of Z^* using k^* is given in [35]. These bounds, in turn, have also been used to develop more efficient algorithms for convex separable programming problems [28] and [29], and non-convex programming problems [36].

As discussed above, separable programming solutions are inexact since they correspond only to piecewise-linear approximations and not the original functions. Estimation of the errors in the optimal quantities is therefore important in practice. This is even more so because, for large problems without any special structure, the technique may often be the only nonlinear optimization technique available with large commercial linear programming systems; the systems with which practitioners solving large, real-world problems have been familiar for a long time. Estimation of k^* , obtained by optimal quadratic splines, will also allow such systems to give estimates of the errors in the solutions produced.

2.1.3. Applications in Statistics (Density estimation and nonparametric regression models)

The great flexibility of splines to interpolate data with lower degree polynomials which minimize computational burden and large oscillations between data points resulting from interpolation by using higher degree polynomials is invaluable in statistical models. Since data representation and its analysis is one of the core issues in statistics, it is not surprising that there are applications of splines starting from the basic problems of smoothing a histogram. For pioneering work on this subject, an interested reader is referred to the age distribution of Hungarian mothers in 1963, ears of Iowa variety of corn [44–46], and various problems relating to the estimation of density functions and higher moments of distributions [39,47]. For a comprehensive survey of uses of splines in general statistics, an interested reader is referred to [38] and in approximate regression models [48].

The text [47] discusses splines in nonparametric regression and its “intriguing connection” with Bayesian estimation in a very general framework. The [49] provides a very readable yet elementary and concise discussion of fitting spline functions with fixed knots by standard regression methods, e.g., using SPSS package computations. While in parametric regression we need very specific quantitative information, generally based on theory or past experience, the flexibility of spline functions in nonparametric regression lets the “data speak for itself” [47]. In addition, such regression allows to examine the validity of an assumed parametric form of the regression function, and in turn, a spline function obtained may suggest an appropriate functional form for a parametric analysis. Regression problems in statistics under order restrictions are discussed in [121]. For other models that require splines refer to [37,50].

2.1.4. Applications to Optimal Control Problems (Optimal timing policies)

Many optimization problems requiring an optimal function rather than only an optimal *point* in R^n for their solution are best handled by control theory [58–62] framework. An excellent introduction to splines in control theory intended to be read by ‘ordinary mortals’ (p. vi) is [50]. The problems analyzed in the applications area include several problems in inventory and production [63,64], construction project scheduling [59], timing of acceleration to achieve takeoff with minimum energy consumption [52], optimal extraction timing of natural resources such as timber or oil reserves [65], system or user optimization in dynamic network traffic assignment problems [66], and in the topic of dynamic scheduling and routing for flexible manufacturing systems [67]. In certain optimal control situations the optimal function must be a spline. For example, consider a time-optimal control problem solved in [68]. A particle moves along the y -axis according to $y = F(t)$, and the velocities of different orders are given by

$$F^{(\nu)}(t), \nu = 0, 1, \dots, n-1,$$

which are absolutely continuous. If we are given a bound on the n^{th} derivative $|F^{(n)}(t)| \leq A$ for fixed A , and the initial and final states of the particle are:

$$F^{(0)}(t) = 0, F^{(\nu)}(T) = 0, \nu = 0, 1, \dots, n-1, F(T) = l,$$

then find the shortest time T_0 to complete this motion of the particle from 0 to the height of l on the y -axis and determine the optimal function $F(t)$ which will achieve this minimal

T_0 . The optimal solution $F(t)$ is given by a perfect spline (that is, the n^{th} derivative $F^{(n)}$ changes sign at the knots but has the same absolute value $|F^{(n)}|$) with minimum

$$T_0 = 2 \left(\frac{2^{n-2}(n-1)!}{A} \right)^{1/n}.$$

The above applications give a perspective on applications of splines. The general definition of optimal splines of r^{th} order and the corresponding problem formulation to determine optimal splines will be provided in the following section.

2.1.5. Computer Graphics and CAD/CAM Applications (Generating free-form lines and surfaces)

Perhaps the most intuitively clear need for smoothness appears in free-form curves and surfaces, the fundamental tools in computer-aided design, modeling, and graphics. Splines generate free-form curves and surfaces and are used in the design of airplanes [100,101], automobiles [102,103], hulls of ships and tankers [104,105], and bottles [106]. Here splines seem to fall almost as a natural requirement. The continuity of specified derivatives in splines can provide the desirable amount of smoothness. In CAD/CAM and computer graphics [107–110], a recurring basic problem is to connect points specified by the user on the screen. When we need to join them “smoothly” by a curve, straight lines are not sufficient and splines are often used.

For example, smoothness would be required for the access ramps in highway systems, for conveyors in material flow systems, for minimizing resistance in aerodynamic or hydraulic systems, or even for visual esthetics in display or plotting systems. An extensive graphic system SAS/GRAPH [111] provides a (cubic) spline option for displaying plots of given or computed data. Smooth spirals have been used in highway design for many decades [112,113], where smoothness and controlling the curvature is of concern since a vehicle’s turning ability is limited and sharp turns are undesirable. Cubic splines have been used for such applications [113]. Applications of splines in this area have grown very rapidly due to the great progress made in computer technology during past decades. Several survey articles in this area are [114–116]. Book-length expositions of splines on computer-aided geometric design include [101,110,117]. Briefer discussions of splines in computer graphics are in [118,119] and with more emphasis on computer-aided manufacturing in [120].

3. Problem Formulation

3.1. General Problem Formulation

For a given set of data points $\{x_i, y_i\}_{i=1}^n$ such that $A=x_1 < \dots < x_n=B$, a spline function $S(x)$ of degree r with the knots at $x_{knot_1}, \dots, x_{knot_K}$ is a real valued function defined on the interval $[A, B]$ such that:

- (i) $S(x)$ passes through all the data points, i.e., $S(x_i) = y_i$,
- (ii) In each interval $(A, x_{knot_1}), (x_{knot_1}, x_{knot_2}), \dots, (x_{knot_K}, B)$, where K is the number of knots, $S(x)$ is given by a polynomial of degree r or less,
- (iii) $S(x)$ and its derivatives of orders $1, 2, \dots, r-1$ are continuous in the interval (A, B) ,
- (iv) r^{th} derivative of $S(x)$ can be discontinuous at a finite number of points.

Thus, a spline function of degree r is a piecewise polynomial function with $(r-1)$ continuous derivatives. We could also say that it is a member of C^{r-1} whose r^{th} derivative is a step function. In other words, it is an r^{th} order indefinite integral of a step function. Polynomial pieces of a spline are ‘spliced’ appropriately at the knots where r^{th} derivative jumps from one value to another (often with opposite signs), enabling it to turn or change its shape; while all derivatives up to $(r-1)^{th}$ order remain continuous, imparting smoothness to the spline. It should be noted that x -coordinates of given data points $\{x_i, y_i\}_{i=1}^n$ are generally not the knot points.

The problem of finding the optimal spline of r^{th} degree is

$$\min_{f \in C^{r-1}} \|f^{(r)}\|_{\infty} = \min_{f \in C^{r-1}} \max |f^{(r)}| \quad (1)$$

$$\text{s.t. } f(x_i) = y_i, \quad (2)$$

where $\{x_i, y_i\}_{i=1}^n$ are given data points.

The formulation (1)-(2), though simple and elegant, is unfortunately not easy to solve. This formulation amounts to the optimization over a Banach space of continuous functions on a compact set $[A, B]$, which is well-known to be notoriously difficult [51, p. 3]. In problem (1)-(2), the decision variable is a function f that minimizes $\|f^{(r)}\|_{\infty}$. In order to mediate the computational burden, the entire space is discretized into segments over which the optimization can be done with much less computational effort while the solution obtained still satisfies an acceptable accuracy criterion.

3.2. Linear Optimization Formulation of the Problem (1)-(2)

This section presents a numerical approach for finding the optimal splines in maximum norm space (typically referred to as L_{∞} space) since we are minimizing $\|f^{(r)}\|_{\infty} = \max |f^{(r)}|$. The main idea of the approach is to formulate a linear optimization problem, the solution of which approximates $\|f^{(r)}\|_{\infty}$, with certain tolerance, of the optimal spline interpolating given n data points $\{x_i, y_i\}_{i=1}^n$.

All n data points are assumed to be equidistant such that $A=x_1$, $B=x_n$ and $x_i - x_{i-1} = H$, $H > 0$, $i = 2, \dots, n$, and $[A, B] = \bigcup_{i=1}^{n-1} [x_i, x_{i+1}]$. In order to discretize the problem, we choose to divide each interval $[x_i, x_{i+1}]$, $i = 1, \dots, n-1$ in m subintervals of equal width. Thus we partition interval $[A, B]$ into l evenly spaced subintervals, where $l = (n-1)m$. Note that the number of points defining the l subintervals is $l+1$, and the points are denoted by $\{\hat{x}_j\}_{j=1}^{(n-1)m+1=l+1}$, such that the relation $\hat{x}_{(i-1)m+1} = x_i$, $i = 1, \dots, n-1$ holds between original data points x_i and data points \hat{x}_j created by subdivisions.

For example, $\hat{x}_1 = x_1 = A$, $\hat{x}_{11} = x_2$, $\hat{x}_{21} = x_3, \dots, \hat{x}_{(n-1)m+1=l+1} = x_n = B$. It should also be pointed out that the general formula for \hat{x}_j is $\hat{x}_j = B \frac{(j-1)}{l} + A \frac{(1-j+l)}{l}$, so that $\hat{x}_1 = A$, $\hat{x}_2 = \frac{B}{l} + A \frac{(1+l)}{l}$, $\hat{x}_3 = \frac{2B}{l} + A \frac{(2+l)}{l}, \dots, \hat{x}_{l+1} = B$. Also, while H denotes the width of each interval $[x_i, x_{i+1}]$ specified in the data, h denotes the width of each smaller interval $[\hat{x}_j, \hat{x}_{j+1}]$ created after m subdivisions of each interval. In other words, $H = mh$.

After the interval $[A, B]$ is discretized, the derivatives are replaced by their discrete counterparts. Derivatives of a function are often approximated by finite differences. A discrete analog of the r^{th} derivative of a continuous function is the r^{th} order difference divided by h^r . A general formula of the r^{th} order forward difference is based on r consecutive (sub)intervals and is represented as follows [122–124]

$$\Delta_h^r[f](x) = \sum_{i=0}^r (-1)^i \binom{r}{i} f(x + (r-i)h).$$

Central differences $\delta_h^r[f](x) = \sum_{i=0}^r (-1)^i \binom{r}{i} f(x + (\frac{r}{2} - i)h)$ provide more accurate approximations as compared to forward (or backward) differences: $\frac{\Delta_h^r[f](x)}{h^r} = f^{(r)}(x) + O(h)$ and $\frac{\delta_h^r[f](x)}{h^r} = f^{(r)}(x) + O(h^2)$, thus error in forward differences is of the order h while in central differences it is of the order h^2 . Essentially, this is due to considering the values of the function at $h/2$ distance away both to the left and to the right of the function in central differences rather than the entire h distance only on one side of the function in forward differences, resulting in a better approximation of the function derivative by central differences.

Briefly, the first order difference $\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h}$ divided by h can be expanded into Taylor series in the following way:

$$\begin{aligned} \frac{f\left(x+\frac{h}{2}\right)-f\left(x-\frac{h}{2}\right)}{h} &\approx \\ \left(\frac{f(x)}{h}+\frac{f'(x)}{2}+\frac{f''(x)}{2}h+\frac{1}{48}f^{(3)}(x)h^2\right)-\left(\frac{f(x)}{h}-\frac{f'(x)}{2}+\frac{f''(x)}{2}h-\frac{1}{48}f^{(3)}(x)h^2\right) &= \\ f'(x)+\frac{1}{24}f^{(3)}(x)h^2 \end{aligned}$$

Also,

$$\frac{f(x+h)-f(x)}{h} \approx \frac{f(x)}{h} + f'(x) + \frac{1}{2}f''(x)h - \frac{f(x)}{h} = f'(x) + \frac{1}{2}f''(x)h,$$

from which one can observe that $\frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h} - f'(x) = O(h^2)$, while $\frac{f(x+h)-f(x)}{h} - f'(x) = O(h)$. Similar arguments hold for higher order finite differences.

Even though the forward difference approximates the third derivative at \hat{x}_j with the error of the order of $O(h)$, it approximates the third derivative at midpoint $\frac{\hat{x}_{j+1}+\hat{x}_{j+2}}{2}$ with the error of the order of $O(h^2)$, as in the discussion of central differences above. Indeed, if we denote $\hat{z}_j = \frac{\hat{x}_{j+1}+\hat{x}_{j+2}}{2}$, then

$$\begin{aligned} \frac{f(\hat{x}_j) - rf(\hat{x}_{j+1}) + r(r-1)f(\hat{x}_{j+2}) + \dots + (-1)^r f(\hat{x}_{j+3})}{h^3} = \\ \frac{f\left(\hat{z}_j - \frac{3h}{2}\right) - rf\left(\hat{z}_j - \frac{h}{2}\right) + r(r-1)f\left(\hat{z}_j + \frac{h}{2}\right) + \dots + (-1)^r f\left(\hat{z}_j + \frac{3h}{2}\right)}{h^3}, \end{aligned}$$

which is the third-order central difference, and therefore, the overall accuracy of the approximation is of the order of $O(h^2)$.

A straightforward approach to find a near-optimal solution of (1)-(2) is to minimize maximum of the r^{th} -order differences directly:

$$\begin{aligned} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_{(m-1)n+1})} \max \left| \frac{\Delta_h^r[f](\hat{x}_j)}{h^r} \right| \Bigg\} \\ \text{s.t., } f(\hat{x}_{(m-1)i+1}) = y_i, \quad i = 1, \dots, n \end{aligned} \quad (3)$$

Note that in this discretized version (and for the rest of the paper), points denoted by $f(\hat{x}_j)$ are decision variables, where $\hat{x}_j, j = 1, \dots, (n-1)m+1$ are endpoints of subintervals.

Since the objective function in the formulation (3) is clearly non-linear due to \max and absolute value operators, in order to deal with this nonlinearity, the problem is converted into an equivalent linear counterpart (similar to [52, p. 28]):

$$\begin{aligned} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_{(m-1)n+1})} k \\ \text{s.t., } \frac{\Delta_h^r[f](\hat{x}_j)}{h^r} \leq k, \quad j = 1, \dots, l-r \\ \frac{\Delta_h^r[f](\hat{x}_j)}{h^r} \geq -k, \quad j = 1, \dots, l-r \\ f(\hat{x}_{(m-1)i+1}) = y_i, \quad i = 1, \dots, n \end{aligned} \quad (4)$$

Since $f(\hat{x}_{(m-1)i+1}) = y_i$ are given as data, the actual decision variables to determine are $f(\hat{x}_j)$ with the exception of $f(\hat{x}_{(m-1)i+1}), i = 1, \dots, n$, although for more convenient

implementation, $f(\hat{x}_{(m-1)i+1})$ can still be treated as decision variables. Also, due to linearity, (4) can be solved with much higher efficiency than (3). This is generally the goal of all linearization processes used to deal with non-linearity from a computational perspective.

Theorem 3.1. *The optimal value of the formulation (4) approaches k^* , the optimal value of the formulation (1)-(2), as the number of subintervals approaches infinity ($m \rightarrow \infty$).*

Proof. As established above, the formulation (4)

$$\left. \begin{array}{l} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_{(m-1)n+1})} k \\ \text{s.t.}, \frac{\Delta_h^r[f](\hat{x}_j)}{h^r} \leq k, \quad j = 1, \dots, l-r \\ \frac{\Delta_h^r[f](\hat{x}_j)}{h^r} \geq -k, \quad j = 1, \dots, l-r \\ f(\hat{x}_{(m-1)i+1}) = y_i, \quad i = 1, \dots, n \end{array} \right\}$$

is equivalent to formulation (3).

$$\left. \begin{array}{l} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_{(m-1)n+1})} \max \frac{|\Delta_h^r[f](\hat{x}_j)|}{h^r} \\ \text{s.t.}, f(\hat{x}_{(m-1)i+1}) = y_i, \quad i = 1, \dots, n \end{array} \right\}$$

As the number of subintervals approaches infinity ($m \rightarrow \infty$) the third-order difference approaches the third derivative, so the formulation (4) becomes

$$\left. \begin{array}{l} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots} \max |f^{(3)}(\hat{x}_j)| \\ \text{s.t.}, f(x_i) = y_i, \quad i = 1, \dots, n \end{array} \right\} \quad (5)$$

Despite having the same objective function, the formulation (5) is theoretically not equivalent to the formulation of the original problem (1)-(2). Any countable (even infinitely countable) set of decision variables $\{f(\hat{x}_j)\}$ is a set of measure zero (a set of points that can be enclosed by intervals with arbitrarily small width), whereas the space over which formulation (1)-(2) is optimized is continuous, namely, the entire interval $[A, B]$. In other words, formulations (5) and (1)-(2), though they have the same objective function, are optimized over different decision variable spaces, namely, discrete and continuous respectively.

Notwithstanding the differences between formulations (5) and (1)-(2) noted above, their respective optimal values differ by an infinitesimally small value as the number of subintervals approaches infinity. Indeed, for an arbitrarily small $\varepsilon > 0$, we can always find a large enough m , number of subintervals (of each interval), such that $|\hat{x}_j - \hat{x}_{j+1}| < \varepsilon$. Since the third derivative of f exists almost everywhere for a cubic spline by definition, we can

define central differences at the midpoints of each subinterval $(\hat{x}_j, \hat{x}_{j+1})$ as $\frac{\delta_h^r[f](\frac{\hat{x}_j + \hat{x}_{j+1}}{2})}{h^r} = f^{(3)}(x) + O(h^2)$. Now, as explained before, note that

$$\frac{\delta_h^r[f](\frac{\hat{x}_j + \hat{x}_{j+1}}{2})}{h^r} = \frac{\Delta_h^r[f](\hat{x}_{j-1})}{h^r}$$

and therefore, the optimal solution to (5) differs from the optimal solutions by a quantity of the order of $O(h^2)$. In other words, for any two arbitrarily close decision variables $f(\hat{x}_j)$ and $f(\hat{x}_{j+1})$, the respective third derivative of f evaluated at any point between them differs from the third derivative evaluated at either of these points by a quantity that is arbitrary small except possibly at the points of discontinuity of $f^{(r)}$, which constitute a set of measure zero since $f^{(r)}$ exists almost everywhere.

Therefore, respective optimal values of formulations (5)

$$\left. \begin{array}{l} \min_{f(\hat{x}_1), f(\hat{x}_2), \dots} \max |f^{(r)}(\hat{x}_j)| \\ \text{s.t. } f(x_i) = y_i, \quad i = 1, \dots, n \end{array} \right\}$$

and (1)-(2)

$$\left. \begin{array}{l} \min_{f \in C^2} \|f^{(r)}\|_{\infty} \\ \text{s.t. } f(x_i) = y_i \end{array} \right\}$$

differ by a quantity of the order of $O(h^2)$. \square

4. Numerical Testing

The algorithms described above were implemented in CPLEX on an Intel® Core™ i7 CPU Q 720 @1.60GHz and 4.00 GB RAM (see the CPLEX code in Appendix A).

4.1 Numerical Examples

In this section four examples are presented. Example 4.1 provides a step by step walk-through the algorithm. Examples 4.2-4.4 illustrate performance of the algorithm for different test functions. Scalability results are presented in Examples 4.5-4.6. In addition, Example 5 gives computational evidence that the optimal value of the formulation (2) approaches the optimal value of (1)-(2) as $O(h^2)$.

Example 4.1: In order to illustrate all the steps of the formulation very clearly, we provide all basic details for this example.

Consider a problem with 5 data points: $(x_1=1, y_1=3)$, $(x_2=2, y_2=5)$, $(x_3=3, y_3=4.5)$, $(x_4=4, y_4=6.6)$, and $(x_5=5, y_5=2.6)$. Here we divide each interval into 2 subintervals, so $n = 5$, $m = 2$, $l = (n - 1)m = 8$ and $h = 0.5$ with the resulting subintervals $[\hat{x}_1, \hat{x}_2]=[1, 1.5]$, $[\hat{x}_2, \hat{x}_3]=[1.5, 2]$, $[\hat{x}_3, \hat{x}_4]=[2, 2.5]$, $[\hat{x}_4, \hat{x}_5]=[2.5, 3]$, $[\hat{x}_5, \hat{x}_6]=[3, 3.5]$, $[\hat{x}_6, \hat{x}_7]=[3.5, 4]$, $[\hat{x}_7, \hat{x}_8]=[4, 4.5]$, and $[\hat{x}_8, \hat{x}_9]=[4.5, 5]$. Note that some of the $\hat{x}_j, j = 1, \dots, 9$ are problem data values $x_i, i = 1, \dots, 5$: $\hat{x}_1 = x_1$, $\hat{x}_3 = x_2$, $\hat{x}_5 = x_3$, $\hat{x}_7 = x_4$ and $\hat{x}_9 = x_5$.

In this case the formulation (4) takes the following form

$$\begin{array}{ll} \min & k \\ \text{s.t.,} & -k \leq \frac{f(\hat{x}_1) - 3f(\hat{x}_2) + 3f(\hat{x}_3) - f(\hat{x}_4)}{0.5^3} \leq k; \\ & -k \leq \frac{f(\hat{x}_2) - 3f(\hat{x}_3) + 3f(\hat{x}_4) - f(\hat{x}_5)}{0.5^3} \leq k; \\ & -k \leq \frac{f(\hat{x}_3) - 3f(\hat{x}_4) + 3f(\hat{x}_5) - f(\hat{x}_6)}{0.5^3} \leq k; \\ & -k \leq \frac{f(\hat{x}_4) - 3f(\hat{x}_5) + 3f(\hat{x}_6) - f(\hat{x}_7)}{0.5^3} \leq k; \\ & -k \leq \frac{f(\hat{x}_5) - 3f(\hat{x}_6) + 3f(\hat{x}_7) - f(\hat{x}_8)}{0.5^3} \leq k; \\ & -k \leq \frac{f(\hat{x}_6) - 3f(\hat{x}_7) + 3f(\hat{x}_8) - f(\hat{x}_9)}{0.5^3} \leq k; \\ & f(\hat{x}_1) = y_1 = 3; f(\hat{x}_3) = y_2 = 5; f(\hat{x}_5) = y_3 = 4.5; f(\hat{x}_7) = y_4 = 6.6; f(\hat{x}_9) = y_5 = 2.6; \end{array}$$

Here $\hat{x}_1 = x_1$, $\hat{x}_2 = \frac{x_1+x_2}{2}$, $\hat{x}_3 = x_2$, $\hat{x}_4 = \frac{x_2+x_3}{2}$, $\hat{x}_5 = x_3$, $\hat{x}_6 = \frac{x_3+x_4}{2}$, $\hat{x}_7 = x_4$, $\hat{x}_8 = \frac{x_4+x_5}{2}$, $\hat{x}_9 = x_5$. The decision variables are $f(\hat{x}_1)$, $f(\hat{x}_2)$, $f(\hat{x}_3)$, $f(\hat{x}_4)$, $f(\hat{x}_5)$, $f(\hat{x}_6)$, $f(\hat{x}_7)$, $f(\hat{x}_8)$ and $f(\hat{x}_9)$, but $f(\hat{x}_1)$, $f(\hat{x}_3)$, $f(\hat{x}_5)$, $f(\hat{x}_7)$ and $f(\hat{x}_9)$ are known to be y_1 , y_2 , y_3 , y_4 and y_5 respectively, so the actual decision variables are only $f(\hat{x}_2)$, $f(\hat{x}_4)$, $f(\hat{x}_6)$ and $f(\hat{x}_8)$.

The solution is: $f(\hat{x}_2) = 4.9625$, $f(\hat{x}_4) = 4.4125$, $f(\hat{x}_6) = 5.6625$, $f(\hat{x}_8) = 6.0125$, $k = 10.4$.

Figures 1 and 2 present a near-optimal cubic spline, presented by discrete points, interpolating data points from Example 1 using $m = 10$. Dots represent data point and calculated decision variable values with smooth line representing the resulting spline and the cubic polynomials constituting the spline.

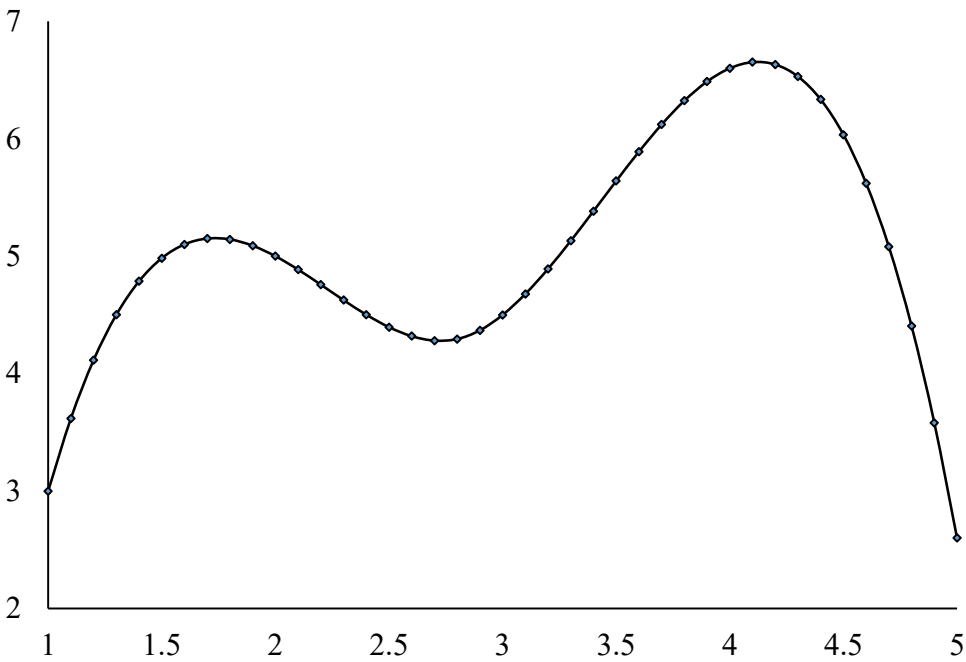


Figure 1. Near-optimal spline for Example 1 with $m = 10$

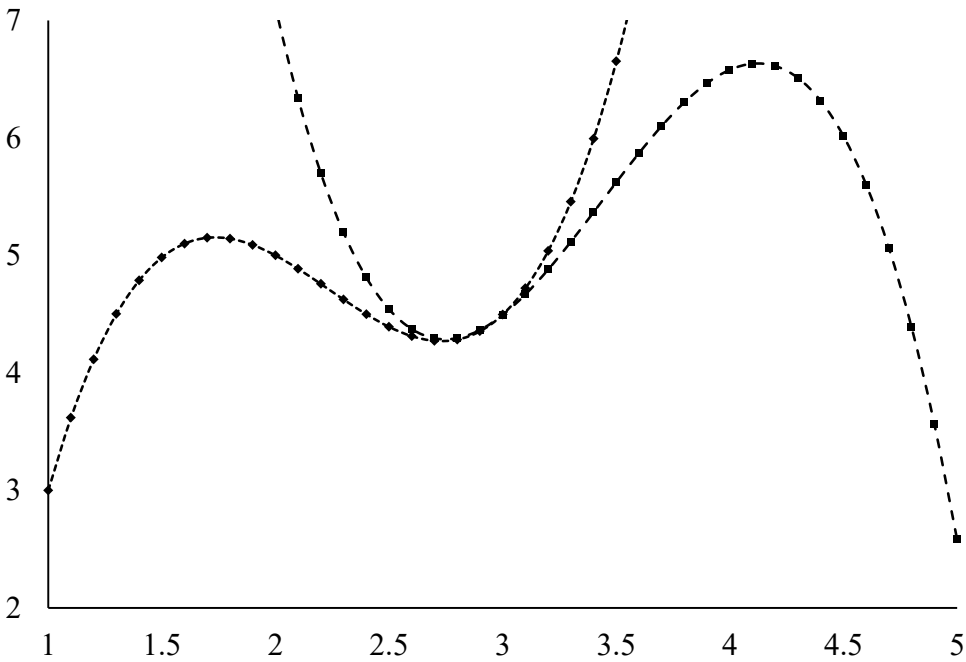


Figure 2. Polynomials comprising the near-optimal spline for Example 1 with $m = 10$

Table 1: Data for Examples 2-4

x	Example 2	Example 3	Example 4
$x_1 = \mathbf{1}$	$y_1 = 1.1051$	$y_1 = 0.8415$	$y_1 = 3$
$x_2 = \mathbf{2}$	$y_2 = 1.2214$	$y_2 = 0.9093$	$y_2 = 3.3113$
$x_3 = \mathbf{3}$	$y_3 = 1.3498$	$y_3 = 0.1411$	$y_3 = 3.74$
$x_4 = \mathbf{4}$	$y_4 = 1.4918$	$y_4 = -0.757$	$y_4 = 4.2219$
$x_5 = \mathbf{5}$	$y_5 = 1.6487$	$y_5 = -0.959$	$y_5 = 4.6928$
$x_6 = \mathbf{6}$	$y_6 = 1.8221$	$y_6 = -0.279$	$y_6 = 5.0883$
$x_7 = \mathbf{7}$	$y_7 = 2.0137$	$y_7 = 0.6570$	$y_7 = 5.3443$
$x_8 = \mathbf{8}$	$y_8 = 2.2255$	$y_8 = 0.9894$	$y_8 = 5.3963$
$x_9 = \mathbf{9}$	$y_9 = 2.4596$	$y_9 = 0.4121$	$y_9 = 5.1947$
$x_{10} = \mathbf{10}$	$y_{10} = 2.7183$	$y_{10} = -0.5440$	$y_{10} = 4.7732$
$x_{11} = \mathbf{11}$	$y_{11} = 3.0042$	$y_{11} = -1$	$y_{11} = 4.196$
$x_{12} = \mathbf{12}$	$y_{12} = 3.3201$	$y_{12} = -0.537$	$y_{12} = 3.5274$
$x_{13} = \mathbf{13}$	$y_{13} = 3.6693$	$y_{13} = 0.4202$	$y_{13} = 2.8317$
$x_{14} = \mathbf{14}$	$y_{14} = 4.0552$	$y_{14} = 0.9906$	$y_{14} = 2.1731$
$x_{15} = \mathbf{15}$	$y_{15} = 4.4817$	$y_{15} = 0.6503$	$y_{15} = 1.6159$
$x_{16} = \mathbf{16}$	$y_{16} = 4.9530$	$y_{16} = -0.288$	$y_{16} = 1.2244$
$x_{17} = \mathbf{17}$	$y_{17} = 5.4739$	$y_{17} = -0.961$	$y_{17} = 1.0628$
$x_{18} = \mathbf{18}$	$y_{18} = 6.0496$	$y_{18} = -0.751$	$y_{18} = 1.1953$
$x_{19} = \mathbf{19}$	$y_{19} = 6.6859$	$y_{19} = 0.1499$	$y_{19} = 1.6863$
$x_{20} = \mathbf{20}$	$y_{20} = 7.3890$	$y_{20} = 0.9129$	$y_{20} = 2.6$

Table 2: Computed optimal values and CPU time for Examples 2-4

	CPU time (s)	\hat{k}	Actual k
Example 2 (Exp.)	0.25	0.0062	0.0073891
Example 3 (Sine)	0.24	0.8947	1
Example 4 (Polyn.)	0.2	0.0645	0.0642609

Example 4.2: Consider a problem with $n=20$ and $m=50$ and a “test” function: $f(x) = e^{x/10}$. After being discretized, function f can be represented as 20 points $\{x_i, y_i\}_{i=1}^{20}$ shown in Table 1.

Example 4.3: Consider a problem with $n=20$ and $m=50$ and a “test” function: $f(x) = \sin(x)$. After being discretized, function f can be represented as 20 points $\{x_i, y_i\}_{i=1}^{20}$.

Example 4.4: Consider a problem with $n=20$ and $m=50$ and a “test” function:

$$f(x) = \frac{98442}{34295} + \frac{2725}{157757}x + \frac{97012}{788785}x^2 - \frac{8448}{788785}x^3, x \leq 8.125$$

$$f(x) = -\frac{6798459}{34295} + \frac{671965}{157757}x - \frac{314828}{788785}x^2 + \frac{8448}{788785}x^3, x > 8.125$$

The data points for examples 2-4 are shown in Table 1.

Note that in most practical cases the exact functional form which the cubic spline is used to approximate is not necessarily polynomial and is unknown, so the optimal k^* of optimal splines interpolating given points may differ from the exact k of given test functions. When the test function is polynomial, the optimal k^* is very close to the exact k (see Example 4.5).

4.2 Scalability Results

To test scalability of our approach, formulation (2) is used to test problems with varying number of subintervals. The computation times are summarized in Table 3.

Table 3: Scalability results for Example 5

No. of sub-intervals(m)	\hat{k}	$\frac{k^* - \hat{k}}{k^*}$	CPU time (s)
2	5.75	4.17%	0.05
4	5.93548	1.08%	0.05
8	5.98373	0.27%	0.03
16	5.99592	0.07%	0.03
32	5.99898	0.02%	0.04
64	5.99974	0.00%	0.08
128	5.99993	0.00%	0.22
256	5.99988	0.00%	0.38
512	5.99999	0.00%	0.27

Example 4.5: Consider 5 data points (1, 4); (2, 15); (3, 39.75); (4, 78.25); (5, 124.75) chosen in such a way that the optimal splines has the known value of third derivative equal to 6.

As one can see in Table 3, the algorithm scales well since increase in the computation time is almost linear for the number of subintervals.

5. Conclusions

Classical data interpolation by a general spline with free knots is formulated as linear programming problem to minimize spline curvature and the problem is efficiently solved by using linear programming solvers. Theoretical convergence to the true optimal splines is proved indicating that the interpolatory data points obtained by solving the LP problem are arbitrarily close to the points on the true optimal splines. Testing results based on cubic interpolatory splines provide good approximations to optimal splines within a fast computation time. Future research direction will be to obtain exact functional representation of the optimal splines as well as the exact positions of the knots.

- Demertzis, K., Tsiotas, D., and Magafas, L.: "Modeling and forecasting the COVID-19 temporal spread in Greece: An exploratory approach based on complex network defined splines." *International Journal of Environmental Research and Public Health* 17, no. 13, (2020).
- Kounchev, O., Simeonov, G., and Kuncheva, Z.: "The tvbg-seir spline model for analysis of covid-19 spread, and a tool for prediction scenarios," *arXiv preprint arXiv:2004.11338* (2020).
- Roy, A., and Karmakar, S.: "Bayesian semiparametric time varying model for count data to study the spread of the COVID-19 cases," *arXiv preprint arXiv:2004.02281* (2020).
- Appadu, A. R., Kelil, A. S. and Tijani, Y. O.: "Comparison of some forecasting methods for COVID-19," *Alexandria Engineering Journal* 60, no. 1, 1565-1589 (2021).
- Agiwal, V., Kumar, J., and Yip, Y. C.: "Study the trend pattern in COVID-19 using spline-based time series model: a bayesian paradigm," doi: 10.20944/preprints202007.0306.v1 (2020).
- Wang, M., Jiang, A., Gong, L., Luo, L., Guo, W., Li, C., Zheng, L., et al: "Temperature significant change COVID-19 Transmission in 429 cities," doi: <https://doi.org/10.1101/2020.02.22.20025791> (2020).
- Alahmad, B., Al-Shammari, A. A., Bennakhi, A., Al-Mulla, F., and Ali, H.: "Fasting blood glucose and COVID-19 severity: nonlinearity matters," *Diabetes Care* 43, no. 12, 3113-3116, (2020).
- Pflüger, D., Schober, P., and Valentin, J.: Solving high-dimensional dynamic portfolio choice models with hierarchical B-splines on sparse grids. Available at SSRN 3393524 (2019).
- Kirkby, L. and Deng, S.: Swing option pricing by dynamic programming with b-spline density projection. *International Journal of Theoretical and Applied Finance*, 22(8), (2020).
- Bras, R. L., Buchanan, R., Curry, K. C.: Real Time Adaptive Closed Loop Control of Reservoirs with the High Aswan Dam as a Case Study. *Water Resources Research*, 19, 33-52 (1983)
- Terry, L. A., Pereira, M. V. F., Araripe-Neto, T. A., Silva, L. F. C. A., Sales, P. R. H.: Coordinating the Energy Generation of the Brazil National Hydrothermal Electrical Generating System. *Interfaces*, 16, 16-38 (1986)

12. Foufoula-Georgiou, E., Kitanidis, P. K.: Gradient Dynamic Programming for Stochastic Optimal Control of Multidimensional Water Resources Systems. *Water Resources Research*, 24(8), 1345-1359 (1988)
13. Yakowitz, S.: Dynamic Programming Applications in Water Resources. *Water Resources Research*, 18(4), 673-696 (1982)
14. Tejada-Guibert, J. A.: Spline Stochastic Dynamic Programming for Multiple Reservoir System Operation Optimization. PhD Dissertation, Cornell University (1990)
15. Johnson, S. A., Stedinger, J. R., Shoemaker, C. A., Li, Y., Tejada-Guibert, J. A., Alberto, J.: Numerical Solution of Continuous-state Dynamic Programs Using Linear and Spline Interpolation. *Operations Research*, 41(3), 484-500 (1993)
16. Kelman, J., Stedinger, J. R., Cooper, L. A., Hsu, E., Yuan, S.-Q.: Sampling stochastic dynamic programming applied to reservoir operation. *Water Resources Research*, 26(3), 447-454 (1990)
17. Turgeon, A.: Decomposition Method for the Long-Term Scheduling of Reservoirs in Series. *Water Resources Research*, 17(6), 1565-1570 (1981)
18. Saad, M., Turgeon, A.: Application of Principal Component Analysis to Long-Term Reservoir Management. *Water Resources Research*, 24(7), 907-912 (1988)
19. Pereira, M. V., Pinto, L. M. V. G.: Stochastic Dual Dynamic Programming. Technical Note, DEE-PUC/RJ, Catholic University of Rio de Janeiro (1989)
20. Read, E. G.: A Dual Approach to Stochastic Dynamic Programming for Reservoir Release Schedules. In: Esogbue, A. O. (ed.): *Dynamic Programming for Optimal Water Resources Systems Analysis*, 386-397, Prentice Hall, Englewood, Wiley, NJ (1989)
21. Gal, S.: Optimal Management of Multi-Reservoir Water Supply System. *Water Resources Research*, 15, 737-748 (1979)
22. Shoemaker, C. A.: The Application of the Dynamic Programming and Other Optimization Methods to Pest management. *IEEE Transactions on Automatic Control*, 26, 1125-1132 (1981)
23. Shoemaker, C. A.: Optimal Integrated Control of Populations with Age Structure. *Operations Research*, 30, 40-61 (1982)
24. Shoemaker, C. A., Johnson, S. A.: Stochastic Nonlinear Optimal Control of Populations: Computational Difficulties and Possible Solutions. In: Castillo-Chavez, C., Levin, S. A., Shoemaker, C. A. (eds.): *Mathematical Approaches to Problems in Resource Management and Epidemiology*. *Operations Research*, 30, 67-81, Springer-Verlag, New York (1989)
25. Stedinger, J. R., Sule, B. F., Loucks, D. P.: Stochastic dynamic programming models for reservoir operation optimization. *Water Resources Research*, 20(11), 1499-1505 (1984)
26. White, D. J.: Real applications of Markov decision processes. *Interfaces*, 15(6), 73-78 (1985)
27. Whitt, W.: An Interpolation Approximation for the Mean Workload in a G1/G/1 Queue. *Operations Research*, 37, 936-952 (1989)
28. Thakur, L. S.: Solving Highly Nonlinear Convex Separable Programs Using Successive Approximation. *Computers and Operations Research*, 11, 113-128 (1984)
29. Thakur, L. S.: Successive Approximation in Separable Programming: An Improved Procedure for Convex Separable Programs. *Naval research logistics quarterly*, 33, 325-358 (1986)
30. Séguin, S., Côté, P., and Audet, C.: "Self-scheduling short-term unit commitment and loading problem," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 133-142 (2016)
31. Rodriguez Sarasty, J. A.: "Mixed-Integer Programming Approaches for Hydropower Generator Maintenance Scheduling," (Doctoral dissertation, École Polytechnique de Montréal), (2018).
32. Venter, J. V.: "Variable selection in logistic regression using exact optimisation approaches." PhD diss., North-West University (South Africa), (2020).
33. Bradley, S. P., Hax, A. C., Magnanti, T. L.: *Applied Mathematical Programming*. Addison-Wesley, Reading, MA (1977)
34. Thakur, L. S.: Error Analysis for Convex Separable Programs: the Piecewise Linear Approximation and the Bounds on the Optimal Objective Value. *SIAM Journal of Applied Mathematics*, 34, 704-714 (1978)
35. Thakur, L. S.: Error Analysis for Convex Separable Programs: Bounds on Optimal and Dual Optimal Solutions. *Journal of Mathematical Analysis and Applications*, 75, 486-496 (1980)
36. Thakur, L. S.: Domain Contraction in Nonconvex Programming: Minimizing a Convex Quadratic Objective over a Polyhedron. *Mathematics of Operations Research*, 16, 390-407 (1991)
37. Nurnberger, G.: *Approximation by Spline Functions*. Springer Verlag, Heidelberg (1989)
38. Wegman, E. J., Wright, I. W.: Splines in Statistics. *Journal of the American Statistical Association*, 78, 351-365 (1983)

39. Karlin, S.: Interpolation properties of generalized perfect splines and the solutions of certain Extremal problems. *Transactions of the American Mathematical Society*, 206, 25-66 (1975)
40. Gauthier, J., Q. V. Wu, and T. A. Gooley. "Cubic splines to model relationships between continuous variables and outcomes: a guide for clinicians." (2020): 675-680.
41. Shepherd, B. E., and Rebeiro, P. F.: "Assessing and interpreting the association between continuous covariates and outcomes in observational studies of HIV using splines." *Journal of acquired immune deficiency syndromes*, 74, no. 3 (2017).
42. Minniakhmetov, I., Dimitrakopoulos, D., and Godoy, M.: "High-order spatial simulation using Legendre-like orthogonal splines," *Mathematical geosciences*, 50, no. 7, 753-780 (2018).
43. Tian, Y., Baro, E., and Zhang, R.: "Performance evaluation of regression splines for propensity score adjustment in post-market safety analysis with multiple treatments," *Journal of biopharmaceutical statistics* 29, no. 5, 810-821, (2019).
44. Boneva, L. I., Kendall, D. G., Stefanov, I.: *Spline Transformations: Three Diagnostic Aids for the Statistical Data-Analyst*. *Journal of the Royal Statistical Society, Series B*, 33, 1-70 (1971)
45. Schoenberg, I. J.: *Cardinal Spline Interpolation*. Society for Industrial and Applied Mathematics, Philadelphia (1973)
46. Schoenberg, I. J.: *Splines and Histograms*. In: Meir, A., Sharma, A. (eds.): *Spline Functions and Approximation Theory*, 277-327, Birkhauser Verlag, Basel (1973)
47. Eubank, R. L.: *Spline Smoothing and Nonparametric Regression*. Marcel Dekker, New York (1988)
48. Eubank, R. L.: *Approximate Regression Models and Splines*. *Communications in Statistics-Theory and Methods*, 13(4), 433-484 (1984)
49. Suits, D. B., Mason, A., Chan, L.: *Spline Functions Fitted by Standard Regression Methods*. Center of Planning and Economic Research, Athens (1977)
50. Prenter, P. M.: *Splines and Variational Methods*. Wiley, NJ (1989)
51. Egerstedt, M. B., and Martin, C. F.: "Control theoretic splines: optimal control, statistics, and path planning," Princeton University Press (2009).
52. Luenberger, D. G.: *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA (1984)
53. Kano, H., Egerstedt, M., Nakata, H., and Martin, C. F.: "B-splines and control theory," *Applied mathematics and computation* 145, no. 2-3 263-288, (2003).
54. Zhang, Z., Tomlinson, J., and Martin, C.: "Splines and linear control theory," *Acta Applicandae Mathematica*, 49, no. 1, 1-34 (1997).
55. Sun, S., Egerstedt, M. B., and Martin, C. F.: "Control theoretic smoothing splines," *IEEE Transactions on automatic control*, 45, no. 12, 2271-2279, (2000).
56. Balseiro, P., Cabrera, A., Stuchi, T. J., and Koiller, J.: "About simple variational splines from the Hamiltonian viewpoint," arXiv preprint arXiv:1711.02773 (2017).
57. Lai, S., Lan, M., Gong, K., and Chen, B. M.: "Axis-coupled trajectory generation for chains of integrators through smoothing splines," *Control Theory and Technology* 17, no. 1, 48-61 (2019).
58. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., Mishchenko, E. F.: *The Mathematical Theory of Optimal Processes*. Interscience, New York (1962)
59. Young, Y. L.: *Calculus of Variations and Optimal Control Theory*. Saunders, Philadelphia (1969)
60. Bryson, A. E., Jr., Ho, Y. C.: *Applied Optimal Control*. Blaisdell, Waltham, Massachusetts (1969)
61. Citron, S. J.: *Elements of Optimal Control*. Holt Rinehart and Winston, New York (1969)
62. Kamien, M., Schwartz, N.: *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*. North-Holland, New York (1981)
63. Hwang, C. L., Fan, L. T., Erickson, L. E.: *Optimum Production Planning by the Maximum Principle*. *Management Science*, 13(9), 751-755 (1967)
64. Bazaraa, M. S., Sherali, H. D., Shetty, C. M.: *Nonlinear Programming Theory and Algorithms*. John Wiley and Sons, Inc., New York (1979)
65. Fuller, J. D., Vickson, R. G.: *The Optimal Timing of New Plants for Oil from the Alberta Tar Sands*. *Operations Research*, 35(5), 704-715 (1987)
66. Friesz, T. L., Luque, J., Tobin, R. L., Wie, B. W.: *Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem*. *Operations Research*, 37(6), 893-901 (1989)
67. Maimon, O. D., Gershwin, S. B.: *Dynamic Scheduling and Routing for Flexible Manufacturing Systems That Have Unreliable Machines*. *Operations Research*, 36(2), 279-292 (1988)
68. Schoenberg, I. J.: *The Perfect B-Spline and a Time-Optimal Control Problem*. *Israel Journal of Mathematics*, 10, 261-274 (1971)
69. Runge, C.: *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, *Zeitschrift für Mathematik und Physik*, 46, 224-243 (1901)

70. Ciarlini, P., Ichim, D.: Free-knot cubic spline modelling in cryogenic thermometer calibration. *Measurement*, 39(9), 815-820 (2006)
71. Renyi, A.: *Wahrscheinlichkeitsrechnung*. VEB Deutscher Verlag der Wissenschaften (1962)
72. Schoenberg, I. J.: Contributions to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics*, 4, 45-99 (1946)
73. Ahlberg, J. H., Nilson, E. N., Walsh, J. L.: *The Theory of Splines and Their Applications*. Mathematics in Science and Engineering, 38, Academic Press, New York (1967)
74. Greville, T. N. E.: *Introduction to Spline Functions, Theory and Applications of Spline functions*. Proceedings of Seminar, Math. Research Center, University of Wisconsin, Madison, New York: Academic Press, 1-35 (1969)
75. De Boor, C.: *A Practical Guide to Splines*. Springer-Verlag, New York (1978)
76. Schumaker, L. L.: *Spline Functions: Basic Theory*. Wiley, New York (1981)
77. Schoenberg, I. J.: *Approximations with Special Emphasis on Spline Functions*. Academic Press, New York (1969)
78. Dierckx, P.: *Curve and Surface Fitting with Splines*. Clarendon Press, Oxford (1995)
79. Bates, D. M., Watts, D. G.: *Nonlinear Regression Analysis*. Wiley, New York (1988)
80. De Boor, C., Rice, J.: Least squares cubic spline approximation I - fixed knots. Technical Report CSD TR 20, Department of Computer Sciences, Purdue University (1968)
81. De Boor, C., Rice, J.: Least squares cubic spline approximation II - variable knots. Technical Report CSD TR 20, Department of Computer Sciences, Purdue University (1968)
82. Gervini, D.: Free-knot spline smoothing for functional data. *Journal of the Royal Statistical Society Series B, Statistical Methodology*, 68(4), 671-687 (2006)
83. Jupp, D.: Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 15(2), 328-343 (1978)
84. Yandell, B. S., Hogg, D. B.: Modeling Insect Natalty Using Splines. *Biometrics*, 44(2), 385-395 (1988)
85. Hao, M. H., Olson, W. K.: Modeling DNA Supercoils and Knots with B-Spline Functions. *Biopolymers*, 28, 873-900 (1989)
86. Sekita, I., Toraichi, K., Mori, R., Yamamoto, K., Yamada, H.: Feature extraction of handwritten Japanese characters by spline functions for relaxation matching. *Pattern Recognition*, 21(1), 9-17 (1988)
87. Constable, C. G., Parker, R. L.: Smoothing, Splines and Smoothing Splines - Their Application in Geomagnetism. *Journal of Computational Physics*, 78, 493-508 (1988)
88. Morachevskii, A. G., Mockrievich, A. G., Maiorova, E. A.: Application of Spline Functions for Calculation of Thermodynamic Properties of Liquid Alloys. *Journal of Applied Chemistry of the USSR*, 60, 454-457 (1989)
89. Cheung, Y. K., Tham, L. G., Li, W. Y.: Free-Vibration Analysis of Singly Curved Shell by Spline Finite Strip Method. *Journal of Sound and Vibration*, 128, 411-422 (1989)
90. Elipe, A., Lanchares, V.: The Use of Spline Functions for Solving Spectroscopic Binary Orbits. *Astrophysical Letters Communications*, 27, 265-274 (1988)
91. Kolter, J. Z., Ng, A. Y.: Task-space trajectories via cubic spline optimization. 2009 IEEE Int. Conf. on Robotics and Automation, 1675-1682 (2009)
92. Das, A., Naroditsky, O., Zhiwei, Z., Samarasekera, S., Kumar, R.: Robust visual path following for heterogeneous mobile platforms. *Robotics and Automation (ICRA)*, 2010 IEEE International Conference on Robotics and Automation, 2431-2437 (2010)
93. Yu, D., Deng, L., Gong, Y., Acero, A.: A novel framework and training algorithm for variable-parameter hidden Markov models. *IEEE Transactions on Audio, Speech and Language Processing*, 17(7), 1348-1360 (2009)
94. Christopoulos, V., Schrater, P.: Handling shape and contact location uncertainty in grasping two-dimensional planar objects. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1557-1563 (2007)
95. Mitra, S. K.: *Digital Signal Processing-A Computer Based Approach*. third ed., McGraw-Hill, New York (2005)
96. Schumaker, L. L.: On Shape Preserving Quadratic Spline Interpolation. *SIAM Journal on Numerical Analysis*, 20(4), 854-864 (1983)
97. Roy, A. "Computer Graphics Curves (Bezier and B-Spline)," (2020).
98. Samreen, S., Sarfraz, M., Jabeen, N., and Hussain, M. Z.: "Computer aided design using a rational quadratic trigonometric spline with interval shape control," In 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 246-251. IEEE, (2017).

99. Ibraheem, F, and Hussain, M. Z.: "Visualization of Constrained Data Using Trigonometric Splines," In 2017 21st International Conference Information Visualisation (IV), pp. 400-404. IEEE (2017).
100. Sabin, M. A.: Parametric Splines in Tension. British Aircraft Corporation, Technical Note VTO/NS/160, Surrey (1970)
101. Bartels, R., Beatty, J., Barsky, B.: An Introduction to Splines for Use in Computer Graphics and Geometric Modeling. Morgan Kaufman, Los Altos, CA (1987)
102. Bezier, P. E.: Mathematical and Practical Possibilities of UNISURF. In: Barnhill R. E. and Riesenfeld R. E. (eds.): Computer Aided Geometric Design, 127-152, Academic Press, New York (1974)
103. De Casteljaeu, P.: Courbes et Surfaces à Poles. Citroen, Paris (1959)
104. Riesenfeld, R. F.: Applications of B-Spline Approximation to Geometric Problems of Computer-Aided Design. PhD Thesis, University of Utah (1973)
105. Rogers, D. F., Rodrigues, F., Satterfield, S. G.: Computer Aided Design and Numerically Controlled Production of Towing Tank Models. Proceedings of the 16th Design Automation Conference (1979)
106. Sato, Y., Akeo, M.; Bottle Design Arts System. In: Kunii T. L. (ed.): Frontiers in Computer Graphics - Proceedings of Computer Graphics Tokyo '84, 266-280, Springer-Verlag, Tokyo (1985)
107. Barnhill, R., Riesenfeld, R. F. (eds.): Computer Aided Geometric Design. Academic Press, New York (1974)
108. Rogers, D. F., Adams, J. A.: Mathematical Elements of Computer Graphics. McGraw-Hill, New York (1976)
109. Kunii, T. L. (ed.): Frontiers in Computer Graphics. Proceedings of Computer Graphics, Tokyo '84, Springer-Verlag, Tokyo (1985)
110. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide. Academic Press, San Diego (1988)
111. SAS/STAT(R) 9.2 User's Guide, Second Edition (2010)
112. Hartman, P.: The Highway Spiral for Combining Curves of Different Radii. Transactions of the American Society of Civil Engineers, 122, 389-409 (1957)
113. Meek, D. S., Walton, D. J.: The Use of Cornu Spirals in Drawing Planar Curves of Controlled Curvature. Journal of Computational and Applied Mathematics, 25, 69-78 (1989)
114. Barnhill, R., Lanchares, V. A.: A Survey of the Representation and Design of Surfaces. IEEE Computer Graphics and Applications, 3, 9-16 (1983)
115. Boehm, W. R., Farin, G., Kahmann, J.: A Survey of Curve and Surface Methods in CAGD. Computer Aided Geometric Design, 11, 1-60 (1984)
116. Barnhill, R.: Surfaces in Computer Aided Design: A Survey with New Results. Computer Aided Geometric Design, 2, 1-17 (1985)
117. Barsky, B.: Computer Graphics and Computer Modeling Using Beta-Splines. Springer Verlag (1988)
118. Newman, W. M., Sproull, R. F.: Principles of Interactive Computer Graphics. McGraw-HW, New York (1979)
119. Foley, J. D., van Dam, A.: Fundamentals of Interactive Computer Graphics. Addison Wesley, Reading, MA (1982)
120. Faux, I. D., Pratt, M. J.: Computational Geometry for Design and Manufacture. Wiley, New York (1979)
121. Barlow, R. E., Bartholomew, D. J., Bremner, J. M., Brunk, H. D.: Statistical Inference under Order Restrictions. Wiley, New York (1972)
122. Hildebrand, F. B.: Finite-Difference Equations and Simulations. Prentice-Hall, Englewood Cliffs, New Jersey (1968)
123. Boole, G. A.: A Treatise On The Calculus of Finite Differences. 2nd ed., Macmillan and Company (1872)
124. Levy, H., Lessman, F.: Finite Difference Equations. Dover Publications (1992)

6. Appendix A: CPLEX Code

For illustrative purposes, the CPLEX code is provided for Example 3:

```
//Data File For Example 3
```

```
nbData = 20;
```

```
nbPoints = 30;
```

```
DataPoints = [0.841470985, 0.909297427, 0.141120008,
```

```

-0.756802495, -0.958924275, -0.279415498, 0.656986599, 0.989358247, 0.412118485, -0.544021111,
-0.999990207,
-0.536572918, 0.420167037, 0.990607356, 0.65028784,
-0.287903317, -0.961397492, -0.750987247, 0.14987721, 0.912945251];
//Main Model
int nbPoints = ...;
int nbData = ...;
range points = 1..nbData;
range t = 1..(nbData - 1)*nbPoints+1;
dvar float x[t] in xmin..xmax;
dvar float k in amin..amax;
float DataPoints[points] = ...;
minimize
k;
subject to {
forall (p in points) x[nbPoints*(p-1)+1] == DataPoints[p];
forall (t in 1..(nbData - 1)*nbPoints-2)
(x[t]-3*x[t+1]+3*x[t+2]-x[t+3])*nbPoints*nbPoints*nbPoints<=k;
forall (t in 1..(nbData - 1)*nbPoints-2)
(x[t]-3*x[t+1]+3*x[t+2]-x[t+3])*nbPoints*nbPoints*nbPoints>=-k;}

```