Monotonic Cubic Spline Interpolation

George Wolberg
Itzik Alfy

Department of Computer Science
City College of New York / CUNY
New York, NY 10031
`wolberg@cs-mail.engr.ccny.cuny.edu`

Abstract

This paper describes the use of cubic splines for interpolating monotonic data sets. Interpolating cubic splines are popular for fitting data because they use low-order polynomials and have C^2 continuity, a property that permits them to satisfy a desirable smoothness constraint. Unfortunately, that same constraint often violates another desirable property: monotonicity. The goal of this work is to determine the smoothest possible curve that passes through its control points while simultaneously satisfying the monotonicity constraint. We first describe a set of conditions that form the basis of the monotonic cubic spline interpolation algorithm presented in this paper. The conditions are simplified and consolidated to yield a fast method for determining monotonicity. This result is applied within an energy minimization framework to yield linear and nonlinear optimization-based methods. We consider various energy measures for the optimization objective functions. Comparisons among the different techniques are given, and superior monotonic cubic spline interpolation results are presented.

1 INTRODUCTION

Cubic splines are widely used to fit a smooth continuous function through discrete data. They play an important role in such fields as computer graphics and image processing, where smooth interpolation is essential in modeling, animation, and image scaling. In computer graphics, for instance, interpolating cubic splines are often used to define the smooth motion of objects and cameras passing through user-specified positions in a keyframe animation system. In image processing, splines prove useful in implementing high-quality image magnification.

Cubic splines interpolate (pass through) the data with piecewise cubic polynomials. The use of low-order polynomials is especially attractive for curve fitting because they reduce the computational requirements and numerical instabilities that arise with higher degree curves. These instabilities cause undesirable oscillations when several points are joined in a common curve. Cubic polynomials are most commonly used because no lower-degree polynomial allows a curve to pass through two specified endpoints with specified derivatives at each endpoint. The most compelling reason for their use, though, is their C^2 continuity, which guarantees continuous first and second derivatives across all polynomial segments.

C^2 continuity imposes an intuitive smoothness constraint on the curve. Unfortunately, that same constraint sometimes violates another desirable property: monotonicity. Simply stated, monotonic input data should give rise to an interpolating curve that is smooth *and* monotonic. For instance, consider the interpolating cubic spline passing through the seven marked points in Fig. 1. Although the seven data points are monotonically increasing in $f(x_i)$ for $0 \le i \le 6$, the cubic spline is not monotonic: it contains overshoots and undershoots, i.e., wiggles.

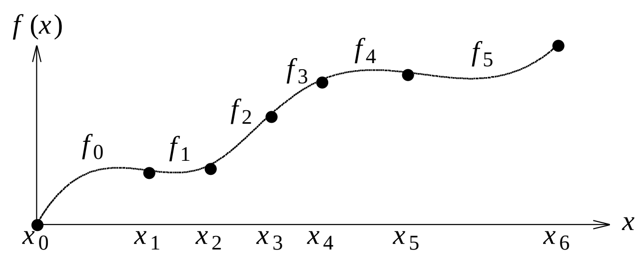

Figure 1. Cubic spline.

The goal of this work is to derive the *smoothest* possible cubic spline that simultaneously interpolates the data and satisfies the monotonicity constraint. In cases where the input is not monotonic, the data can be split into consecutive intervals of monotonically increasing and decreasing data. For now, though, we shall limit our attention to one strictly monotonic interval spanning all the points. We begin with a review of the literature in Section 2 and a review of cubic spline interpolation in Section 3. The monotonicity constraint is discussed in Section 4. Optimization-based solutions are introduced in Section 5. Section 6 presents examples. Finally, a discussion and summary of the work is presented in Section 7 and Section 8, respectively.

2 PREVIOUS WORK

There is a large body of work in the field of monotonic cubic spline interpolation. Early work in this area dates back to Schweikert's work on splines in tension, where exponential splines were used as approximants [18]. Various other exponential and cubic spline interpolants were considered in [20, 12, 13, 14, 5]. Tension parameters were used to control shape. All of these methods were global, interpolatory, and C^2. Automatic algorithms to determine free parameters to control shape and monotonicity were complicated. In [11], an algorithm was presented to generate shape preserving curves of arbitrary smoothness based on the properties of Bernstein polynomials. However, C^2 smoothness required the use of piecewise polynomials whose degree exceeded three. There is also the possibility of using piecewise rational interpolants [6, 9], although these are usually only C^1 or are intended for strictly monotone or strictly convex data.

In 1980, Fritsch and Carlson proposed a two-pass algorithm for computing monotone cubic interpolant [7]. In 1984, Fritsch and Butland proposed a modified technique to simplify the 1980 algorithm [8]. In the modified algorithm, the first derivatives at the knots are calculated using Brodlie's formula to give the most visually pleasing results.

The Fritsch and Butland technique is also available in Netlib (PCHIM.FOR). This can be downloaded from `www.math.iastate.edu/cmlib/pchipd.html`. Both algorithms are local and yield only C^1 continuous curves, even if a global C^2 solution exists. Furthermore, there is no flexibility in defining an application's specific properties for the desired spline, e.g., the objective function or constraints for a given optimization problem.

In [1, 2, 3], several algorithms were proposed to compute shape preserving splines that are monotone and convex. The algorithms are based on [7] and iteratively compute a set of first derivatives that simultaneously satisfy the monotonicity and convexity constraints.

Schumaker [17] proposed a shape preserving interpolation algorithm to produce a C^1 quadratic spline with additional knots where necessary. The algorithm is interactive, and the user has flexibility in adjusting the shape of the interpolating spline under some relationship rules.

Several researchers have investigated other approaches involving the use of additional knots between data points [5, 14, 15, 4]. If two extra break points are allowed between each data subinterval, then there are enough degrees of freedom to construct a globally C^2 cubic spline interpolant which is local and which has slopes and curvatures at the data points as free parameters [15]. Additional breakpoints, however, require more storage and increased search time during evaluation [7].

3 CUBIC SPLINES: A REVIEW

A cubic spline $f(x)$ interpolating on the partition $x_0 < x_1 < \dots < x_{n-1}$ is a function for which $f(x_k) = y_k$. It is a piecewise polynomial function that consists of $n-1$ cubic polynomials f_k defined on the ranges $[x_k, x_{k+1}]$. Furthermore, each f_k is joined at x_k, for $k = 1, \dots, n-2$, such that $y'_k = f'(x_k)$ and $y''_k = f''(x_k)$ are continuous. An example of a cubic spline passing through $n = 7$ data points is illustrated in Fig. 1.

The k-th polynomial curve, f_k, is defined over the fixed interval $[x_k, x_{k+1}]$ and has the cubic form

$$f_k(x) = a_k(x-x_k)^3 + b_k(x-x_k)^2 + c_k(x-x_k) + d_k \quad (1)$$

where

$$a_k = \frac{1}{\Delta x_k^2}\left(-2\frac{\Delta y_k}{\Delta x_k} + y'_k + y'_{k+1}\right) \quad (2a)$$

$$b_k = \frac{1}{\Delta x_k}\left(3\frac{\Delta y_k}{\Delta x_k} - 2y'_k - y'_{k+1}\right) \quad (2b)$$

$$c_k = y'_k \quad (2c)$$

$$d_k = y_k \quad (2d)$$

In the expressions for b_k and a_k, $\Delta x_k = x_{k+1} - x_k$ and $\Delta y_k = y_{k+1} - y_k$, for $k = 0, \dots, n-2$.

The expressions for the cubic polynomial coefficients in Eq. (2) are given in terms of position data and derivatives. In cases where only position data is supplied, the derivative values may be evaluated by solving a tridiagonal system of equations that relate the unknown derivatives to the known position data. Derivations can be found in [16] and [21].

To make the derivatives invariant to scale change, we shall find it useful to relate the derivatives in terms of slope $m_k = \Delta y_k / \Delta x_k$:

$$y'_k = \alpha_k m_k \quad (3a)$$

$$y'_{k+1} = \beta_k m_k \quad (3b)$$

for $\alpha_k \geq 0$ and $\beta_k \geq 0$.

Although the user-supplied data points are fixed, the derivatives can be changed to yield a large family of interpolating cubic splines. We are interested in determining the range of derivative values for which the spline remains monotonic. To motivate the need for determining this range of derivative values, Fig. 2 shows a set of five cubic curves, each with derivative $y'_k = 0$ and increasing values for y'_{k+1}. In particular, $\alpha_k = 0$ and $1 \leq \beta_k \leq 5$ for integer values of β_k. Tangent vectors are shown for the $\beta_k = 1$ and $\beta_k = 5$ cases. Note that the monotonic constraint is violated when $\beta_k > 3$, i.e., a local minima is present in the span.

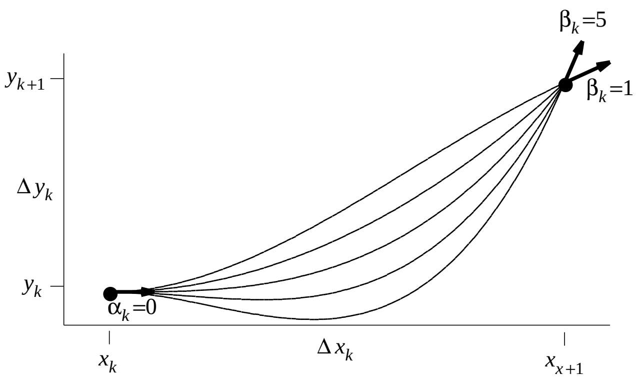

Figure 2. A family of cubic polynomials.

4 MONOTONICITY

In this section, we consider a single cubic polynomial $f_k(x)$ in the interval $[x_k, x_{k+1}]$ and present necessary and sufficient conditions for which $f_k(x)$ is monotonic in the interval [7]. These conditions form the basis of the monotonic cubic spline interpolation algorithm.

A curve is monotonic in an interval $[x_k, x_{k+1}]$ if and only if there is no sign change in the derivative value along any part of the curve in the interval. Therefore, a necessary condition for monotonicity is that

$$sgn(y'_k) = sgn(y'_{k+1}) = sgn(m_k) \qquad (4)$$

Furthermore, if $m_k = 0$, then $f_k(x)$ is monotone (constant) in the interval if and only if $y'_k = y'_{k+1} = 0$.

In the remainder of the presentation, we will assume that $m_k \neq 0$ and that Eq. (4) is satisfied. As a result, $f_k(x)$ is strictly monotonic in the interval $[x_k, x_{k+1}]$ if $f'_k(x) \neq 0$ for $x_k \leq x \leq x_{k+1}$. This implies that there are no local extrema (minima/maxima) in that span.

The monotonicity constraints can be summarized by the following two lemmas [7]:

1) If $\alpha_k + \beta_k - 2 \leq 0$, then $f_k(x)$ is monotone if and only if Eq. (4) is satisfied.

2) If $\alpha_k + 2\beta_k - 2 > 0$, then $f_k(x)$ is monotone if and only if Eq. (4) *and* one of the following conditions is satisfied:

   (a) $2\alpha_k + \beta_k - 3 \leq 0$

   (b) $\alpha_k + 2\beta_k - 3 \leq 0$

   (c) $\alpha_k^2 + \alpha_k(\beta_k - 6) + (\beta_k - 3)^2 < 0$

Conditions (1), (2a), (2b), and (2c) are depicted graphically as regions I, II, III, and IV in Fig. 3(a). The union of these regions, shown in Fig. 3(b), is bounded by lines $\alpha_k = 0$, $\beta_k = 0$, and the ellipse. The monotonicity region M in Fig. 3(b) can be expressed as follows:

$$M = \begin{cases} 0 < \alpha_k + \beta_k < 3 + s & \text{if } 0 \leq \beta_k \leq 3 \\ 3 - s < \alpha_k + \beta_k < 3 + s & \text{if } 3 < \beta_k \leq 4 \end{cases} \qquad (5)$$

where $s = \left|\sqrt{\alpha_k \beta_k}\right|$. All points in region M denote valid (α_k, β_k) pairs that preserve monotonicity.

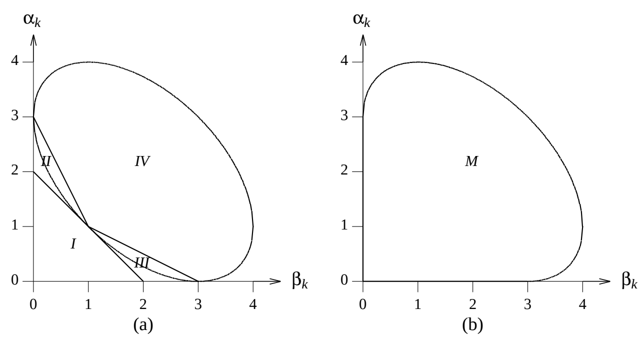

Figure 3. (α, β) **pairs for a monotonic curve.**

5 OPTIMIZATION-BASED SOLUTIONS

In this section, we consider several solutions to the monotonic interpolation problem based on optimization techniques. We will investigate solutions derived by linear and quadratic programming techniques subject to various constraints on first and second derivative continuity.

The objective criterion for the optimization techniques will be based on energy measures of the curves. We begin with a review of the classic cubic spline energy measure in Section 5.1. Since the original cubic spline formulation is not guaranteed to be monotonic, we will impose the monotonicity constraint in Section 5.2. This will furnish a solution that may be solved using quadratic programming. Since monotonic cubic splines are not guaranteed to be C^2 continuous, a new energy measure is introduced in Section 5.4 that addresses the extent of second derivative discontinuity in the spline. That result is further simplified in Section 5.5 to yield a solution that may be solved using linear programming.

5.1 Spline Energy

Cubic splines originally arose as a mathematical model for a draftman's spline. Cubic splines mimic the position of a flexible thin beam that is forced to pass through the given data points [10, 5]. The strain energy of the beam is given as the integral of the curvature:

$$E = \int_{x_0}^{x_{n-1}} \frac{f''(x)^2}{(1+[f'(x)]^2)^{5/2}} dx \tag{6}$$

The *elastica* is the ideal spline, i.e., the function $f(x)$ that minimizes Eq. (6). Due to the inherent difficulty in solving for $f(x)$ under this formulation, a simpler linearized energy measure is commonly used [19, 5, 8]:

$$E_L = \int_{x_0}^{x_{n-1}} f''(x)^2 dx \tag{7}$$

This expression is valid only if one makes the simplifying assumption that $f'(x)^2 \ll 1$ everywhere. Despite the fact that this assumption is often violated in practice, it is widely used since it facilitates a computationally tractable solution for minimizing Eq. (6). The E_L energy measure given in Eq. (7) is often coupled with the free-end (FE) boundary condition $f''(x_0) = f''(x_{n-1}) = 0$ since it has been shown that this condition minimizes Eq. (7) among all C^2 cubic polynomials [5, 19].

5.2 Linearized Energy (LE)

The expression for E_L may be written in terms of the spline coefficients of Eq. (1) as follows:

$$\begin{aligned} E_L &= \sum_{k=0}^{n-2} \int_{x=x_k}^{x_{k+1}} f''(x)^2 dx = \sum_{k=0}^{n-2} \int_{x=x_k}^{x_{k+1}} (6a_k x + 2b_k)^2 dx \\ &= \sum_{k=0}^{n-2} 12a_k^2 \Delta x_k^3 + 12 a_k b_k \Delta x_k^2 + 4b_k^2 \Delta x_k \end{aligned} \tag{8}$$

The solution to the problem of minimizing Eq. (8) over all possible coefficients is not guaranteed to preserve monotonicity. We may address this problem by adding linear monotonicity constraints. Fig. 3 shows the valid range of values for α_k and β_k to yield a monotonic curve segment. We may obtain linear constraints by approximating the closed region in Fig. 3 with an n-sided polygon. In the example below, we use $n = 6$ and $n = 15$ to demonstrate our approach.

Fig. 4 illustrates the two polygons we used to approximate region M in our work. The 6-sided and 15-sided polygons shown in Fig. 4 cover 90.53% and 98.79% of region M, respectively. The 6-sided polygon depicted in Fig. 4(a) consists of the intersection of six half-planes. Expressing these half-planes in terms of the unknown spline coefficients we have the following six monotonicity constraints:

$$-c_k \le 0 \tag{9a}$$

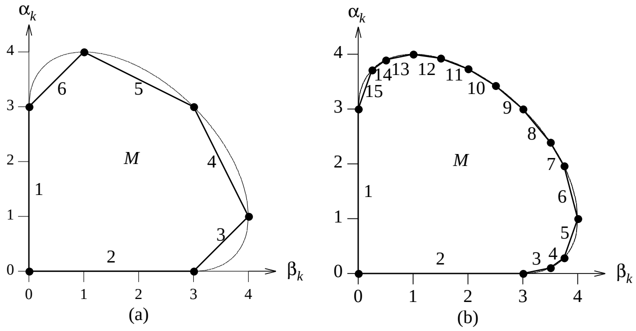

Figure 4. Linear approximation of region M: (a) $n = 6$; (b) $n = 15$.

$$-3a_k\Delta x_k^2 - 2b_k\Delta x_k - c_k \le 0 \tag{9b}$$
$$-3a_k\Delta x_k^2 - 2b_k\Delta x_k - 3m_k \le 0 \tag{9c}$$
$$3a_k\Delta x_k^2 + 2b_k\Delta x_k + 3c_k - 9m_k \le 0 \tag{9d}$$
$$6a_k\Delta x_k^2 + 4b_k\Delta x_k + 3c_k - 9m_k \le 0 \tag{9e}$$
$$3a_k\Delta x_k^2 + 2b_k\Delta x_k - 3m_k \le 0 \tag{9f}$$

Note that Eq. (9) is linear with respect to the unknown coefficients. The energy measure given in Eq. (8) can be minimized subject to these monotonicity constraints by using quadratic programming:

$$\text{MINIMIZE} \sum_{k=0}^{n-2} (12a_k^2\Delta x_k^3 + 12a_k b_k \Delta x_k^2 + 4b_k^2 \Delta x_k) \tag{10}$$

subject to:

1. $a_k\Delta x_k^3 + b_k\Delta x_k^2 + c_k\Delta x + y_k = y_{k+1}$
2. $3a_k\Delta x_k^2 + 2b_k\Delta x_k + c_k = c_{k+1}$
3. $6a_k\Delta x_k + 2b_k = 2b_{k+1}$
4. Eq. (9)

The above constraints denote interpolation, first and second derivative continuity, and monotonicity constraints, respectively. Note that it is possible that a feasible C^2 solution does not always exist. In that case, we must solve the minimization problem without the second derivative continuity constraint. If a C^2 solution exists and the free-end curve is monotone, then the solution consists of the coefficients of the free-end curve.

5.3 LE Properties

When no feasible C^2 solution exists, the second derivative discontinuity may be visually prominent at the spline joints. Even if the C^2 solution exists, it may not necessarily minimize E_L subject to the constraints given in Eq. (10). In fact, a C^1 solution may have a lower E_L. This is due to the fact that the domain of the C^1 solutions is a superset of the

C^2 solution domain. Note that the solution for C^0 monotone constraints is the linear interpolation with $E_L = 0$. This implies that the E_L measure is valid for C^2 solutions only. The following example demonstrates these phenomena. Consider the data set:

$$\begin{array}{lclllll} \mathbf{X} & = & [0 & 1 & 2 & 3] \\ \mathbf{Y} & = & [0 & 400 & 400 & 800] \end{array}$$

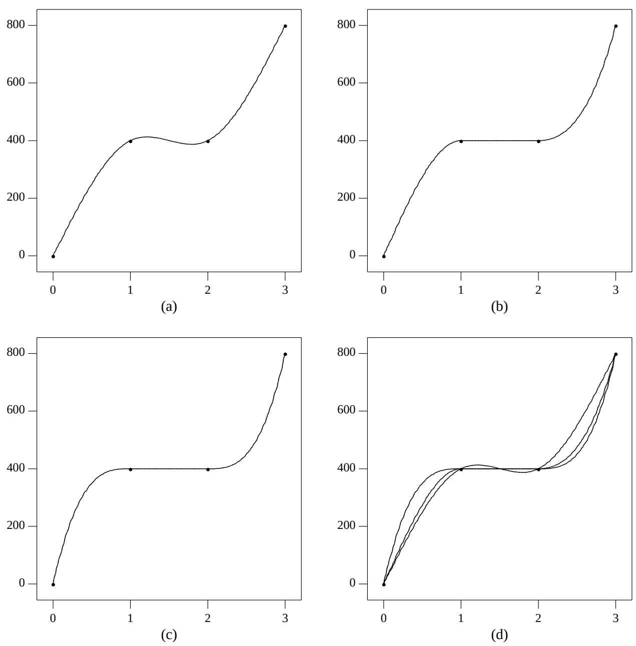

Figure 5. (a) FE; (b) C^1 LE; (c) C^2 LE; (d) overlay.

Fig. 5(a) shows the data fitted with a spline satisfying the free-end condition. Notice that although the data is monotone, the curve is not monotonic. Figures 5(b) and 5(c) show the curves that solve the quadratic programming problem formulated in Eq. (10) with C^1 and C^2 constraints, respectively. The value of the second derivative difference at (1,400) is high and visually prominent.

Table 1 summarizes the energy measures for Fig. 5. Note that although the C^2 LE curve in Fig. 5(c) has a higher E_L, it is unquestionably smoother. All of the solutions used the 6-sided polygonal approximation to region M shown in Fig. 4(a).

5.4 Second Derivative Discontinuity Energy (SDDE)

Due to the limitations of the approaches based on minimizing E_L, we seek to solve for the closest C^2 spline,

Method	E	E_L	E_D
FE	1231.66	640000	0
LE (C^1)	1067.33	1600000	1600000
LE (C^2)	58.70	3840000	0

Table 1. Energy measures.

thereby producing a more natural looking curve. This process requires us to introduce an energy measure based on the second derivative discontinuities:

$$\begin{aligned} E_D &= \sum_{k=1}^{n-2} (f"(x_k^-) - f"(x_k^+))^2 \\ &= \sum_{k=0}^{n-2} (6a_k \Delta x_k + 2b_k - 2b_{k+1})^2 \qquad (11) \end{aligned}$$

A similar objective function was suggested by Nielson in his work on ν-splines [12].

The energy measure given in Eq. (11) can be minimized subject to monotonicity constraints by using quadratic programming:

$$\text{MINIMIZE} \sum_{k=0}^{n-2} (6a_k \Delta x_k + 2b_k - 2b_{k+1})^2 \qquad (12)$$

subject to:

1. $a_k \Delta x_k^3 + b_k \Delta x_k^2 + c_k \Delta x + y_k = y_{k+1}$
2. $3a_k \Delta x_k^2 + 2b_k \Delta x_k + c_k = c_{k+1}$
3. Eq. (9)

The above constraints denote interpolation, first derivative continuity, and monotonicity constraints, respectively.

In the minimization formulation for the linearized energy method, a second derivative continuity constraint was included among the set of constraints. Note that no such constraint is necessary in Eq. (12) since a monotone C^2 spline is a solution to Eq. (12).

The following example demonstrates the advantages of the SDDE approach. Consider the following data set.

$$\begin{array}{lcllllll} \mathbf{X} & = & [0.00 & 1.00 & 1.50 & 2.05 & 2.90] \\ \mathbf{Y} & = & [0.00 & 350.00 & 354.65 & 428.00 & 650.00] \end{array}$$

Fig. 6(a) shows the spline satisfying the free-end condition. Notice that although the data is monotone, the curve is not monotonic. Figs. 6(b) and 6(c) show the C^1 curve that solves the quadratic programming problems formulated in Eqs. 12 and 10, respectively, using the 6-sided polygonal approximation to region M shown in Fig. 4(a). Fig. 6(d) shows the cubic spline elastica (CSE) obtained by minimizing Eq. (6). Table 2 summarizes the energy measures for Fig. 6. Note that although the CSE is a C^2 monotone curve, the LE and SDDE curves could not yield that solution since its (α_k, β_k) set exists in the area lying outside the polygon and inside M. It is evident that the SDDE curve and its energy measures are much closer to those of CSE than LE.

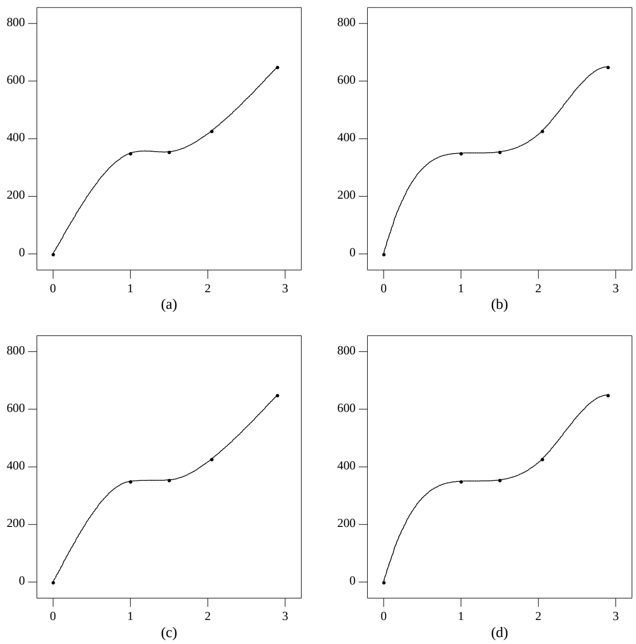

Figure 6. (a) FE; (b) SDDE (c) C^1 LE (d) CSE.

Method	E	E_L	E_D
FE	855.84	343408.02	0
SDDE	845.22	1601043.30	0.70
LE (C^1)	8.28	392688.93	666782.99
CSE	27.15	1504779.88	0

Table 2. Energy measures.

5.5 Modified Discontinuity Energy (MDE)

The quadratic programming formulation given above is not necessarily guaranteed to converge to the true (optimal) solution. As a result, we simplify the discontinuity energy measure E_D to be linear with the cubic coefficients so that a linear programming procedure can be applied. The simplification is done by adding an unknown constant K to each second derivative difference, such that the term $f"(x_k^-) - f"(x_k^+) + K$ is positive. The positivity of each term is obtained by adding it as an optimization constraint where K is one of the optimization unknowns.

$$\overline{E}_D = \sum_{k=1}^{n-2} f"(x_k^-) - f"(x_k^+) + K$$
$$= \sum_{k=0}^{n-2} 6a_k \Delta x_k + 2b_k - 2b_{k+1} + K \qquad (13)$$

The energy measure given in Eq. (13) can be minimized subject to monotonicity constraints by using linear programming:

$$\text{MINIMIZE} \sum_{k=0}^{n-2} 6a_k \Delta x_k + 2b_k - 2b_{k+1} + K \qquad (14)$$

subject to:

1. $a_k \Delta x_k^3 + b_k \Delta x_k^2 + c_k \Delta x + y_k = y_{k+1}$
2. $3a_k \Delta x_k^2 + 2b_k \Delta x_k + c_k = c_{k+1}$
3. $6a_k \Delta x_k + 2b_k - 2b_{k+1} + K \geq 0$
4. Eq. (9)

The above constraints denote interpolation, first derivative continuity, positivity, and monotonicity constraints, respectively. Note that no second derivative continuity constraint is necessary since a monotone C^2 spline is a solution to Eq. (14).

6 RESULTS

In this section, we compare the results of the different techniques described in this paper. The six methods compared are: CSE, FE, Fritsch & Butland (FB), LE, SDDE, and MDE. We used the optimization toolbox of Matlab 5.2 to solve the linear and quadratic programming problems above. Note that Matlab indicates when no feasible solution exists. We will demonstrate the techniques on the following data set:

X = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Y = [0, 1, 4.8, 6, 8, 13, 14, 15.5, 18, 19, 23, 24.1]

Figures 7 to 10 depict the curves produced by the CSE, FE, SDDE, and FB methods, respectively. Each curve is presented in two coordinate systems: $(x, f(x))$ and (α, β). The purpose of this representation is to highlight the monotone characteristics of the curves. The spline figures contain vertical lines ended with circles to represent the second derivative difference at the knots, where the difference is measured by $D_k = (f"(x_k-) - f"(x_k+))^2$ Note that E_D is defined as the sum of all the D_k's. The scale for the second derivative differences is normalized to fit the curve scale,

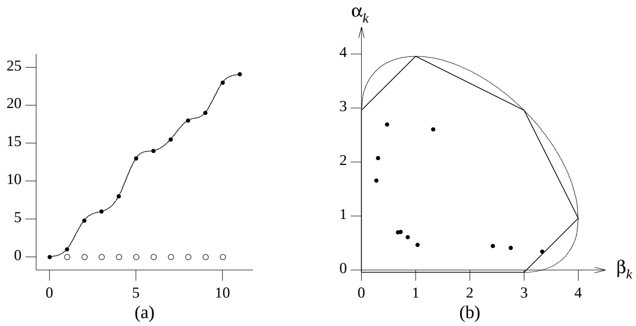

Figure 7. Cubic spline elastica (CSE): (a) interpolating spline; (b) $\{\alpha, \beta\}$ points

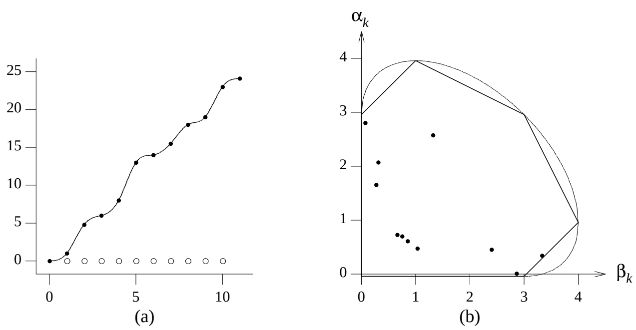

Figure 8. Free end (FE): (a) interpolating spline; (b) $\{\alpha, \beta\}$ points

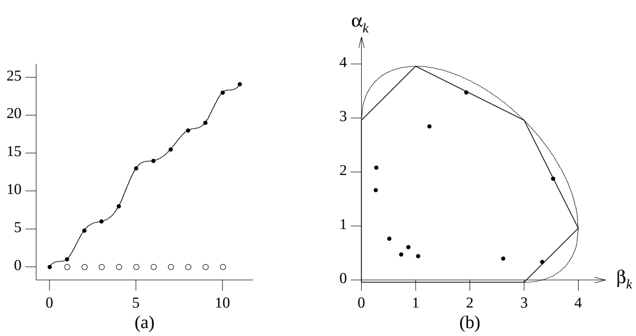

Figure 9. Second derivative discontinuity energy (SDDE): (a) interpolating spline; (b) $\{\alpha, \beta\}$ points

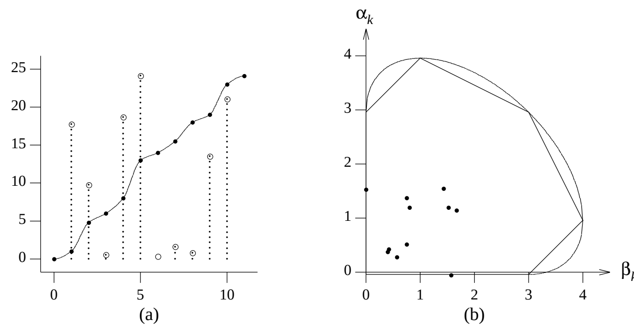

Figure 10. Fritsch-Butland (FB): (a) interpolating spline; (b) $\{\alpha, \beta\}$ points

while it's maximum value is given in Table 3. Note also that for C^2 curves, e.g., $E_D = 0$, the differences are zero.

All curves in Figures 7 to 10 are C^2 and monotone, with the exception of the Fritsch-Butland curve, which is only C^1 and monotone. The distribution of the $\{\alpha_k, \beta_k\}$ points for the Fritsch-Butland curve are concentrated near the origin, i.e., biased towards low values. This has a noticeable effect on the smoothness of the curve. In particular, the resulting curve exhibits more tension and is biased towards linear interpolation where $(\alpha, \beta) = (1, 1)$. It is also salient that the second derivative discontinuities are prominent at the spline joints. Table 3 summarizes these results. Note that the linear

Method	E	E_L	E_D	Max(D)
CSE	6.72	132.91	0.00	0.00
FE	7.41	131.68	0.00	0.00
LE	7.41	131.68	0.00	0.00
SDDE	15.20	223.55	0.00	0.00
MDE	7.67	131.71	0.00	0.00
FB	5.62	236.30	949.02	211.77

Table 3. Energy measures.

and quadratic programming procedures require specifying an initial starting point for the minimization search. The curves illustrated in the examples are obtained by setting the initial starting point to zero. As specified in previous sections, it may happen that the solutions obtained with quadratic programming do not reflect the optimal solution since it may get stuck in a local minima.

7 DISCUSSION

In this section, we review several key points about monotonic cubic spline interpolation.

1. By examining the physical definition of the strain energy, it is implicit that energy measures Eq. (6) and Eq. (7) are applicable for C^2 curves only. This means that the energy measures are only meaningful when comparing C^2 monotone splines. For instance, we cannot use such energy measures to compare C^0 curves, such as those produced by linear interpolation. It is apparent that such curves produce low values for E and E_L, although they are not smooth at the spline joints. That is, they satisfy $f''(x) = 0$ nearly everywhere. Only at the spline joints is this condition possibly violated. Therefore, C^2 energy measures E and E_L are not appropriate objective functions for monotone splines, since the monotonicity constraint may sometimes force the spline to be C^1 continuous.

2. The assumption used to define E_L is correct for a subset of C^2 curves that comply with the condition $f'(x)^2 \ll 1$. This condition is often violated in practice. This makes the use of linearized energy E_L meaningless as an objective function for optimization-based methods. For instance, in comparing the FE and the SDDE curves in Fig. 6, higher E for the free-end (FE) curve does not translate to higher E_L (see Table 2).

3. The Fritsch-Butland algorithm clamps the α and β values to the $[0, 3]$ range, thereby utilizing only 67.9% of monotone region M. This has a tendency of biasing the solution away from smoother alternatives. The optimization-based solutions presented in this paper more fully utilize region M, yielding the smoother SDDE and MDE curves shown in the figures above.

4. The global algorithm used to minimize the second derivative discontinuity produces monotone C^2 solutions, if they exist. This lies in contrast to the local Fritsch-Butland algorithm that produces C^1 solutions.

8 SUMMARY

The goal of this work has been to determine the *smoothest* possible curve that passes through its control points while simultaneously satisfying the montonicity constraint. Various energy measures were considered for the optimization objective functions. We showed how to apply quadratic programming to minimize the objective functions used in this paper. Modifications were introduced to simplify the problem and facilitate the use of linear programming. We found that optimization-based methods yielded superior results to the popular Fritsch-Butland algorithm [8]. We showed that the traditional linearized energy measure E_L is based on invalid assumptions and is of limited value in determining C^1 monotonic solutions. We suggested that the SDDE and MDE energy measures be used as optimization objectives. They minimized the second derivative discontinuity and provided visually pleasing results.

References

[1] P. Costantini. An algorithm for computing shape-preserving cubic spline interpolation to data. *Calcolo.*, 21:295–305, 1984.

[2] P. Costantini. On monotone and convex spline interpolation. *Math. Comp.*, 46:203–214, 1986.

[3] P. Costantini. Co-monotone interpolating splines of arbitrary degree: A local approach. *SIAM J. Sci. Stat. Comp*, 8:1026–1034, 1987.

[4] P. Costantini. An algorithm for computing shape-preserving interpolating splines of arbitrary degree. *J. Comp. Appl. Math*, 22:89–136, 1988.

[5] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.

[6] R. Delbourgo and J. Gregory. Shape preserving piecewise rational interpolation. *SIAM J. Sci. and Stat. Comp.*, 6:967–976, 1985.

[7] F. Fritsch and R. Carlson. Monotone piecewise cubic interpolation. *SIAM J. Numerical Analysis*, 17(2):238–246, 1980.

[8] F. N. Fritsch and J. Butland. A method for constructing local monotone piecewise cubic interpolants. *SIAM J. Sci. Stst. Comput.*, 5(2), 1984.

[9] J. Gregory and R. Delbourgo. Piecewise rational quadratic interpolation to monotonic data. *IMA J. Numerical Analysis*, 2:123–130, 1982.

[10] M. M. Malcolm. On the computation of nonlinear spline functions. *SIAM J. Numerical Analysis*, 14(2):254–282, 1977.

[11] D. McAllister, E. Passow, and J. Roulier. Algorithms for computing shape preserving splines interpolations to data. *Math. Comp.*, 31:717–725, 1977.

[12] G. M. Nielson. Some piecewise polynomial alternatives to splines under tension. In R. Barnhill and R. Riesenfeld, editors, *Computer Aided Geometric Design*, pages 209–235. Academic Press, New York, 1974.

[13] S. Pruess. Properties of splines in tension. *J. of Approximation Theory*, 17:86–96, 1976.

[14] S. Pruess. Alternatives to the exponential spline in tension. *Math. Comp.*, 33:1273–1281, 1979.

[15] S. Pruess. Shape preserving c^2 cubic spline interpolation. *IMA J. Numerical Analysis*, 13:493–507, 1993.

[16] D. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics*. McGraw-Hill, New York, 1990.

[17] L. L. Schumaker. On shape oreserving quadratic spline interpolation. *SIAM J. Numerical Analysis*, 20:854–864, 1983.

[18] D. Schweikert. An interpolation curve using splines in tension. *J. Math and Phys.*, 45:312–317, 1966.

[19] L. Shampine, J. R.C. Allen, and S. Pruess. *Fundamentals of Numerical Computing*. John Wiley and Sons, New York, 1997.

[20] H. Späth. Exponential spline interpolation. *Computing*, 4:225–233, 1969.

[21] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.