Research Article

A New Approach of Constrained Interpolation Based on Cubic Hermite Splines

J. Saeidian, M. Sarfraz, A. Azizi, and S. Jalilian

1Faculty of Mathematical Sciences and Computer, Kharazmi University, No. 50, Taleghani Ave., Tehran, Iran
2Department of Information Science, College of Life Sciences, Kuwait University, Sabah AlSalem University City, Shadadiya, Kuwait City, Kuwait
3Department of Mathematics, Payame Noor University, P.O. Box 19395-3697, Tehran, Iran

Correspondence should be addressed to J. Saeidian; j.saeidian@khu.ac.ir

Received 28 April 2021; Revised 5 August 2021; Accepted 14 August 2021; Published 31 August 2021

Academic Editor: Niansheng Tang

Copyright © 2021 J. Saeidian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Suppose we have a constrained set of data and wish to approximate it using a suitable function. It is natural to require the approximant to preserve the constraints. In this work, we state the problem in an interpolating setting and propose a parameter-based method and use the well-known cubic Hermite splines to interpolate the data with a constrained spline to provide with a $C^1$ interpolant. Then, more smoothing constraints are added to obtain $C^2$ continuity. Additionally, a minimization criterion is presented as a theoretical support to the proposed study; this is performed using linear programming. The proposed methods are demonstrated with illustrious examples.

1. Introduction

In the industrial design, it is often required to generate a smooth function approximating a given set of data which preserves certain shape properties of the data such as positivity, monotonicity, or convexity, that is, a smooth shape-preserving approximation [1].

Shape-preserving approximation is a subfield of computer aided design (CAD) and has extensively been applied in various areas of engineering such as ship design, car body design, and aerospace industry; it also plays a crucial role in aerography and even in animation and games. Now, some emerging research fields, such as modern data analysis, mathematical finance, image processing, visualization, and digital watermarking techniques, are putting forward higher standards for curve and surface shape-preserving modeling systems [2].

Here, it is generally assumed that the given data are sufficiently accurate to warrant interpolation, so we focus on the shape-preserving interpolation techniques; however, for more oscillatory data, one may prefer to use other approximation methods such as least squares.

In a shape-preserving interpolation problem, we seek for a smooth curve/surface passing through a given set of data in which we priorly know that there is a shape feature in it and one wishes the interpolant to inherit these features [1, 3–9]. Positivity, monotonicity, and convexity are the basic and fundamental shape features which normally arise in everyday scientific phenomena.

During the past three decades, various shape-preserving interpolation methods have been proposed. In this problem, splines play a crucial role, and every approach to shape-preserving interpolation, more or less, uses splines as a cornerstone.

The shape-preserving problem can be handled using different bases, and innovative techniques have been employed to cope with different aspects of this generally studied problem [10, 11].

A good survey of different approaches for shape-preserving interpolation techniques could be found in [2] and the references therein.

One of the hidden features in a dataset may be its boundedness. This happens, for example, when the data come from a sampling of a bounded function or they reflect the probability or efficiency of a process. Actually, any
quantity which is expressed as a percentage of another quantity will necessarily lie between 0 and 100; in this case, the data satisfy \( m \leq f_i \leq M \), for known bounds \( m \) and \( M \). However, the bounds could be imposed by functions (curves) as well, i.e., for known functions \( B_1(x) \) and \( B_2(x) \), the data satisfy \( B_1(x_i) \leq f_i \leq B_2(x_i) \). When we recreate the underlying entity by interpolation from the sampled values, we need to ensure that the interpolating curve adheres to these known properties, so we wish to find a function \( g(x) \) which approximates (fits) the data and is also bounded into \([m,M]\); moreover, as every shape-preserving modeling, we require a reasonable degree of smoothness. Such a problem may have lots of applications, for example, in engineering and data visualization. There is a good review of literature in [12], where Asim and Brodlie have mentioned the advantages and drawbacks of different approaches; they have also modified the knot insertion algorithm proposed in [13]. The Shepard interpolation family [14] have been used by Brodlie et al. in [15] to study this problem.

This article studies the problem in an interpolating setting and takes advantage of cubic splines to visualize the data. It proposes a parameter-based method and uses the well-known cubic Hermite splines to interpolate a constrained data with a \( C^1 \) cubic spline, which preserves the desired bounds. It furthermore proposes the work to obtain \( C^2 \) spline by adding more smoothing constraints. Additionally, an energy minimization technique is used to provide \( C^1 \) and \( C^2 \) interpolating functions. This is performed using a linear programming technique.

The structure of the study is as follows. In Section 2, the cubic Hermite spline with unknown derivative parameters is used to provide with a \( C^1 \) interpolant; then, more smoothing constraints are added in Section 3 to obtain \( C^2 \) continuity. Section 4 deals with more general constraints and studies the case where the constraints are quadratic polynomials. A minimization criterion is presented in Section 5, and Section 6 is devoted for demonstration with illustruous examples.

2. Bounded \( C^1 \) Cubic Hermite Spline

Consider \( \{(x_i, f_i)\}_{i=1}^n \) as a dataset, where \( \{x_i\} \) are distinct points in real line and \( \{f_i\} \) are bounded values, say \( f_i \in [m, M] \). Without loss of generality, we assume that \( m = 0 \) and \( M = 1 \), so we wish to find a smooth interpolant \( p(x) \) on \([x_1, x_n] \) with \( 0 \leq p(x) \leq 1 \). For each subinterval \([x_i, x_{i+1}] \), \( i = 1, \ldots, n-1 \), we define \( h_i = x_{i+1} - x_i \), \( \delta_i = ((f_{i+1} - f_i)/h_i) \), and \( t = ((x - x_i)/h_i) \). We state the \( C^1 \) bounded interpolation problem (C1BIP) as follows.

**Problem 1 (C1BIP).** For a bounded dataset \( \{(x_i, f_i)\}_{i=1}^n \), where \( 0 \leq f_i \leq 1 \), there is an interpolant which is \( C^1 \) on \([x_1, x_n] \) and is bounded into \([0,1]\).

To answer this question, we use a \( C^1 \) cubic Hermite spline (CHS) \( S(x) \), which is defined in \([x_i, x_{i+1}] \) as follows:

\[
S(x) = S_i(x) = f_i + m_i h_i t + (3 \delta_i - 2m_i - m_{i+1}) h_i t^2 + (m_i + m_{i+1} - 2\delta_i) h_i t^3,
\]

where \( m_i = S'(x_i) \) are the unknown values. They are the shape parameters, and we use them to force the interpolant to satisfy the bounding conditions, i.e., \( 0 \leq S(x) \leq 1 \). Schmidt and Hess [16] have given the necessary and sufficient conditions, for a cubic polynomial, to be positive on an interval. We use a sufficient condition extracted from their results to ensure \( S_i(x) \geq 0 \) and \( 1 - S_i(x) \geq 0 \).

We assume that \( 0 \leq f_i \leq 1 \) for \( i = 0, \ldots, n \).

**Lemma 1** (See [16]). The function \( S_i(x) \) on \([x_i, x_{i+1}] \) is positive if \((m_i, m_{i+1}) \in R_1\), where

\[
R_1 = \left\{(x, y) \mid x \geq -\frac{3 f_i}{h_i}, y \leq \frac{3 f_{i+1}}{h_i}\right\}.
\]

**Corollary 1.** If \((m_i, m_{i+1}) \in R_2\), where

\[
R_2 = \left\{(x, y) \mid x \leq \frac{3 (1-f_i)}{h_i}, y \geq \frac{3 (f_{i+1} - 1)}{h_i}\right\},
\]

then \( S_i(x) \leq 1 \) on \([x_i, x_{i+1}] \).

**Proof.** It suffices to apply Lemma 1 on \( 1 - S_i(x) \). \( \square \)

**Theorem 1.** A sufficient condition for the cubic Hermite spline \( S(x) \) to satisfy \( 0 \leq S(x) \leq 1 \) is that the set of values \( \{m_i\}_{i=1}^n \) fulfills the following system of inequalities:

\[
\frac{-3 f_i}{h_i} \leq m_i \leq \frac{3 (1-f_i)}{h_i},
\]

\[
\max \left\{ \frac{-3 f_i \cdot (3 f_i - 1)}{h_i \cdot h_{i-1}}, \frac{3 (1-f_i) \cdot 3 f_i}{h_i, h_{i-1}} \right\} \leq m_i \leq \min \left\{ \frac{3 (1-f_i) \cdot 3 f_i}{h_i, h_{i-1}}, 1 \right\}, \quad i = 2, \ldots, n-1,
\]

\[
\frac{3 (f_n - 1)}{h_{n-1}} \leq m_n \leq \frac{3 f_n}{h_{n-1}}.
\]
Proof. The constraints resulted from Lemma 1 and Corollary 1 must be applied to each subinterval. The value of $m_i$ is affected by only the first interval constraints, so it suffices to satisfy (4). For every $m_i, i = 2, \ldots, n - 1$, one must consider restrictions of two consecutive subintervals, namely, $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$.

So, there are 4 restrictions on $m_i$ as follows:

(i) Applying Lemma 1 on $[x_{i-1}, x_i]$ forces $m_i \leq \frac{3}{h_i} f_i (x_{i-1})$.

(ii) Applying Corollary 1 on $[x_{i-1}, x_i]$ forces $m_i \geq \frac{3}{h_i} (f_i - 1)(x_{i-1})$.

(iii) Applying Corollary 1 on $[x_i, x_{i+1}]$ forces $m_i \leq \frac{3}{h_i} (1 - f_i) (x_{i+1})$.

(iv) Applying Lemma 1 on $[x_i, x_{i+1}]$ forces $m_i \geq \frac{3}{h_i} f_i (x_{i+1})$.

These inequalities must be satisfied simultaneously, so inequalities (5) is the natural presentation of these constraints. The case of $m_n$ is similar to $m_1$.

**Definition 1.** The set of all points satisfying inequalities (4)–(6) is defined to be the feasible region of the cubic Hermite spline method for C1BIP.

**Lemma 2.** The feasible region of the Definition 1 is nonempty.

Proof. For each $m_i, i = 1, \ldots, n$, we should verify that the lower bound of the corresponding constraints, from

\[
\frac{1}{h_i} m_i + \frac{2}{h_i + h_{i+1}} m_{i+1} + \frac{1}{h_{i+1}} m_{i+2} = \frac{3}{h_i} \delta_i + \frac{3}{h_{i+1}} \delta_{i+1}, \quad i = 1, \ldots, n - 2.
\]

The system of linear equation (7) with inequalities (4)–(6) forms a set of constraints. To find a feasible solution, one can solve a linear programming problem subjected to these constraints. So, a question of feasibility arises here: “does a solution to the system (7) satisfies in (4)–(6)?” This is dependent upon $|f_i|$ values; here, we state a sufficient condition.

**Lemma 3.** If $\{x_i\}$ forms a uniform partition and $(1/2) \leq f_i \leq 1$ for all $i = 1, \ldots, n$ (or, respectively, $0 \leq f_i \leq (1/2)$), then solution to the linear system (5) satisfies the inequalities (4)–(6).

Proof. Using notation $h = h_i, i = 1, \ldots, n - 1$, one can rewrite equation (7) as

\[
m_i + 4m_{i+1} + m_{i+2} = 3(\delta_i + \delta_{i+1}), \quad i = 1, \ldots, n - 2.
\]

One should verify that equation (8) is feasible subject to constraints (4)–(6). We assume $\{m_i\}_{i=1}^n$ satisfy (4)–(6). Taking advantage of equations

inequalities (4)–(6), is actually lower than the upper bound. This is obvious due to the assumption $0 \leq f_i \leq 1$. In inequalities (4)–(6), every lower bound is a nonpositive value, while each upper bound is nonnegative.

The following theorem is a straightforward result of Lemma 2.

**Theorem 2.** For any dataset $\{(x_i, f_i)\}_{i=1}^n$, there exists a $C^1$ cubic Hermite spline interpolant, in the form of equation (1), which provides a solution to Problem 1.

**Remark 1.** Any point in the feasible region, presented by inequalities (4)–(6), provides a solution to Problem 1. One may choose each $m_i$ to be the middle point of the corresponding feasible interval. In the next section, we seek for optimal solutions which provide with more smooth and visually pleasing solution curves.

### 3. Bounded C² Cubic Hermite Spline

The cubic Hermite spline (1) provides a family of solutions to Problem 1. Any set of $\{m_i\}_{i=1}^n$ from the feasible region results in a $C^1$ bounded interpolant. We can impose more restrictions on $m_i$ values to obtain interpolants with desired properties. Imposing the $C^2$ continuity property, the second derivative of $S(x)$ must be continuous at each interior point $x_{i+1}, i = 1, \ldots, n - 2$:

\[
\text{max}\{a, b\} = a + b + \frac{|a - b|}{2},
\]

\[
\text{min}\{a, b\} = a + b - \frac{|a - b|}{2},
\]

one can represent

\[
L_i = \min \left\{ \frac{-3f_i}{h} \frac{3(f_i - 1)}{h} \right\} = \frac{-3}{2h} (1 - |2f_i - 1|).
\]

\[
U_i = \max \left\{ \frac{3(1 - f_i)}{h} \frac{3f_i}{h} \right\} = \frac{3}{2h} (1 - |2f_i - 1|).
\]

For $i = 2, \ldots, n - 2$, equation (5) leads to the following restrictions:

\[
L_i \leq m_i \leq U_i,
\]

\[
L_{i+1} \leq m_{i+1} \leq U_{i+1},
\]

\[
L_{i+2} \leq m_{i+2} \leq U_{i+2},
\]

which results in
Using inequalities (4) and (6), respectively, one can easily verify
\[ m_i + 4m_{i+1} + m_{i+2} \leq \frac{3}{2h} (6 - 2f_i - 1 - 42f_{i+1} - 1 - 2f_{i+2} - 1). \]

(12)

Now, for the feasibility of equation (8), it remains to verify
\[ \frac{-3}{2h} (6 - 2f_i - 1 - 42f_{i+1} - 1 - 2f_{i+2} - 1) \leq m_i + 4m_{i+1} + m_{i+2}, \]
\[ 3(m_i + 4m_{i+1} + m_{i+2}) \leq \frac{3}{2h} (6 - 2f_i - 1 - 42f_{i+1} - 1 - 2f_{i+2} - 1). \]

(13)

(14)

Here, we add the assumptions (1/2) ≤ fi ≤ 1 for all i = 1, ..., n, which leads to
\[ f_i + 2f_{i+1} \leq 3. \]

(15)

The latter inequality is obvious due to our assumption for f_i values. Similar calculations can be used to verify (14). Using inequalities (4) and (6), respectively, one can easily handle the feasibility for i = 1 and i = n.

\[ B_1(x_i) \leq f_i \leq B_2(x_i), \]
there is an interpolating function g which is also constrained by B_1(x) and B_2(x), i.e.,
\[ g(x_i) = f_i, \quad i = 1, \ldots, n, \]
\[ B_1(x) \leq g(x) \leq B_2(x). \]

(16)

We consider the case where B_1(x) and B_2(x) are, at most, quadratic polynomials and set some sufficient conditions on the cubic Hermite spline (1) to fulfill condition (16).

Denoting
\[ B_1(x) = a_1x^2 + b_1x + c_1, \]
\[ B_2(x) = a_2x^2 + b_2x + c_2, \]

one can rewrite the constraining conditions B_1(x) ≤ S(x) ≤ B_2(x) in two positivity conditions:
\[ S(x) - B_1(x) ≥ 0, \]
\[ B_2(x) - S(x) ≥ 0. \]

(18)

Applying Lemma 1, we have the following result:

**Theorem 3.** A sufficient condition for the interpolating cubic Hermite spline S(x) (1) to be constrained by B_1(x) and B_2(x) (16) is to satisfy the following conditions:

\[ m_1 \geq \frac{3(a_1x_i^2 + b_1x_i + c_1 - f_i)}{h_i} + 2a_1x_i + b_1, \]
\[ m_1 \leq \frac{3(a_1x_i^2 + b_1x_i + c_2 - f_i)}{h_i} + 2a_2x_i + b_2, \]
\[ m_i \leq \min \left\{ \frac{3(a_2x_i^2 + b_2x_i + c_2 - f_i)}{h_i} + 2a_2x_i + b_2, \frac{-3(a_1x_i^2 + b_1x_i + c_1 - f_i)}{h_i} - 4a_1x_{i-1} - 2b_1 - 2a_1h_{i-1} \right\}, \]
\[ m_i \geq \max \left\{ \frac{3(a_1x_i^2 + b_1x_i + c_1 - f_i)}{h_i} + 2a_1x_i + b_1, \frac{-3(a_2x_{i-1}^2 + b_2x_{i-1} + c_2 - f_i)}{h_{i-1}} - 4a_2x_{i-1} - 2b_2 - a_2h_{i-1} \right\}, \]
\[ m_n \geq \frac{-3(a_2x_{n-1}^2 + b_2x_{n-1} + c_2 - f_n)}{h_{n-1}} - 4a_2x_{n-1} - 2b_2 - a_2h_{n-1}, \]
\[ m_n \leq \frac{-3(a_1x_{n-1}^2 + b_1x_{n-1} + c_2 - f_n)}{h_{n-1}} - 4a_1x_{n-1} - 2b_1 - a_1h_{n-1}. \]
For getting more smoothness, one can easily add $C^2$ continuity conditions, and the reasoning is the same as the bounded case.

5. Energy Minimization Technique

So far, we have solved the bounded and constrained interpolation problems, and we know that there does exist $C^1$ as well as $C^2$ solutions to these problems. However, in both cases, there may exist a number of solutions; so, a matter of choosing the “optimal solution” arises here. Moreover, to get an explicit result, one has to solve a linear programming problem.

The energy minimization method has successfully been used by Burmeister et al. [17] to obtain an interpolating spline with minimum energy. It is generally based on minimizing the curvature of the spline which is represented by

$$E = \int_{x=x_1}^{x=x_n} \frac{S''(x)}{1 + [S''(x)]^2} \, dx. \quad (20)$$

Wolberg and Alfy [18] have proposed different, but closely related, quantities to express the energy of a spline. They introduced an energy measure based on the second derivative discontinuities:

$$E_D = \sum_{k=2}^{n-1} (S''(x_k) - S''(x_k^+))^2. \quad (21)$$

They have simplified the discontinuity energy measure, $E_D$, to be linear with the first derivatives, so that a linear programming procedure can be applied. The simplification is performed by using the absolute values of the discontinuities:

$$\tilde{E}_D = \sum_{k=2}^{n-1} |S''(x_k) - S''(x_k^+)|. \quad (22)$$

For each discontinuity point, they define a slack variable, $s_k$, whose value is forced to be the absolute value of the discontinuity, using the following inequality constraints:

$$S''(x_k) - S''(x_k^+)^- \leq s_k,$$

$$- [S''(x_k) - S''(x_k^+)] \leq s_k. \quad (23)$$

$\tilde{E}_D$ can be written as

$$\tilde{E}_D = \sum_{k=2}^{n-1} s_k. \quad (24)$$

Here, we use the same techniques and try to reach suitable splines in two stages. We state the idea for bounded case, and then, it can be easily extended to a constrained case. First, we put $\tilde{E}_D$ as an objective function and minimize its subject to the following constraints:

(i) Absolute value constraints: equation (23)

(ii) Boundedness constraints: equations (4)–(6)

The mentioned linear programming (LP) problem could be handled by any LP-solver. The solution to this LP is a vector of values $F^* = (m_0, m_1, s_1, \ldots, s_{n-1})^T$, and now, we have a solution to Problem 1 (C1BIP).

At this point, we can switch to the second stage. $\tilde{E}_D$ reflects the sum of discontinuities in the second derivative; so, whenever it has a relatively small value, we hope that we are close to a $C^2$ solution. Having the solution of the mentioned LP at the hand, we try to reach a $C^2$ solution as well. To do so, we add the $C^2$ continuity constraints, i.e., equation (7) to the set of constraints. This time, we use $E$ from equation (21) as an objective and minimize its subject to the following constraints:

(i) Boundedness constraints: equation (4)–(6)

(ii) $C^2$ continuity conditions: equation (7)

This is a nonlinear programming problem and needs a starting guess; we use the solution $F^*$ of the mentioned LP as the starting point.

6. Examples and Illustrations

In previous sections, the bounded and constrained cubic Hermite spline problems (Problems 1 and 2) have been transformed to a linear programming problem which can be easily handled in MATLAB. To reach a smoother solution, the $C^2$ continuity constraints were put in a nonlinear programming problem, which also can be solved by MATLAB using “fmincon” function. The maximum vector norm of MATLAB with a discretization is used to compare the errors. We illustrate these results through some examples: the first group of examples confined to bounded interpolation and the second group presents the constrained case.

6.1. Bounded CHS Examples

Example 1. We consider the dataset in Table 1.

This is a uniform sampling of the semicircle described by the function $f(x) = 1 - (1/2)\sqrt{x(2-x)}$ on $[0, 2]$. The $C^1$ solution as well as the original curve is illustrated in Figure 1. It is seen that the maximum error is 0.0553. The $C^2$ solution is more pleasing and is reported in Figure 2. The maximum error in this case reduces to 0.0531. Table 2 represents the maximum error for both solutions in each subinterval.

Example 2. Table 3 presents a data sampled from the function

$$f(x) = \frac{10e^{(-x^4/4)}}{1 + e^{1-(x/2)^2}} - \frac{10e^{(-x^4/4)}}{1 + e^{1-(x^2/16)}} + \frac{1}{2} \quad (25)$$

These data are oscillatory and do not fulfill the conditions of Lemma 3; however, the cubic Hermite solutions are convincing enough. The $C^1$ solution as well as the original curve is depicted in Figure 3. It is seen that in $C^1$ case, the maximum error is 0.0193. Table 4 presents the errors for each interval in both $C^1$ and $C^2$ cases.
In the $C^2$ solution, the maximum error is 0.0195, and it is depicted in Figure 4.

**Example 3.** Consider the data given in Table 5, which come from the burning of coal in a furnace: the $y$ values are the percentage of oxygen in flue gas over a range of time. If we visualize these data by cubic spline, it would give a pleasing smooth curve, but it is unfortunately physically nonsense since it generates negative percentages. This problem has been handled in [12] by Hermite and rational splines, and they have reported $C^1$ positive interpolants. Qin and Xu [19] got advantage of a trigonometric spline to propose a $C^1$ positivity preserving approximation. Here, the proposed method provides with a $C^2$ interpolant where its curve lies completely between maximum and minimum values of percentage (Figure 5).

### Table 1: Dataset of a sampling.

| $x_i$ | 0   | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| $f_i$ | 1   | 0.7000 | 0.6000 | 0.5417 | 0.5101 | 0.5000 | 0.5101 | 0.5417 | 0.6000 | 0.7000 | 1   |

**Figure 1:** $C^1$ cubic Hermite approximation of a semicircle.

**Figure 2:** $C^2$ cubic Hermite approximation of a semicircle.

In the $C^2$ solution, the maximum error is 0.0195, and it is depicted in Figure 4.

**Example 3.** Consider the data given in Table 5, which come from the burning of coal in a furnace: the $y$ values are the percentage of oxygen in flue gas over a range of time. If we visualize these data by cubic spline, it would give a pleasing smooth curve, but it is unfortunately physically nonsense since it generates negative percentages. This problem has been handled in [12] by Hermite and rational splines, and they have reported $C^1$ positive interpolants. Qin and Xu [19] got advantage of a trigonometric spline to propose a $C^1$ positivity preserving approximation. Here, the proposed method provides with a $C^2$ interpolant where its curve lies completely between maximum and minimum values of percentage (Figure 5).

### 6.2. Constrained CHS Examples

**Example 4.** Dataset in Table 6 is a sampling of

$$f(x) = xe^{(-x-4)^2/10} - \frac{3xe^{(-2(x-4)/10)}}{10}$$

on a uniform partition in $[0,12]$. We wish to find the corresponding interpolant which lies between
Table 2: Errors for $C^1$ and $C^2$ spline for the dataset from Example 1.

| Intervals  | $[0, 0.2]$ | $[0.2, 0.4]$ | $[0.4, 0.6]$ | $[0.6, 0.8]$ | $[0.8, 1]$ | $[1, 1.2]$ | $[1.2, 1.4]$ | $[1.4, 1.6]$ | $[1.6, 1.8]$ | $[1.8, 2]$ |
|------------|------------|--------------|--------------|--------------|------------|------------|------------|------------|------------|------------|
| $C^1$ spline | 0.0551     | 0.0042       | 0.0008       | 0.0000       | 0.0000     | 0.0002     | 0.0000     | 0.0008     | 0.0042     | 0.0551     |
| $C^2$ spline | 0.0530     | 0.0047       | 0.0012       | 0.0003       | 0.0001     | 0.0001     | 0.0003     | 0.0013     | 0.0047     | 0.0531     |

Table 3: Dataset from Example 2.

| $x_i$ | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
|-------|----|----|----|----|----|----|----|----|----|----|
| $f_i$ | 0.5000 | 0.8070 | 0.8440 | 0.7149 | 0.5000 | 0.2855 | 0.1545 | 0.1385 | 0.2002 | 0.2831 |

Table 4: Errors for $C^1$ and $C^2$ CHS solutions for the dataset from Example 2.

| Intervals  | $[0, 1]$ | $[1, 2]$ | $[2, 3]$ | $[3, 4]$ | $[4, 5]$ | $[5, 6]$ | $[6, 7]$ | $[7, 8]$ | $[8, 9]$ |
|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| $C^1$ spline | 0.0193   | 0.0048   | 0.0012   | 0.0003   | 0.0001   | 0.0004   | 0.0001   | 0.0001   | 0.0009   |
| $C^2$ spline | 0.0530   | 0.0047   | 0.0012   | 0.0003   | 0.0001   | 0.0001   | 0.0003   | 0.0013   | 0.0047   |

Figure 3: $C^1$ CHS solution of Example 2.

Figure 4: $C^2$ CHS solution to Example 2.
\[ B_1(x) = 0.1x^2 + 1.08x \quad \text{and} \quad B_2(x) = -0.12x^2 - 1.56x. \]
The resulting \( C^1 \) solution curve has the maximum error 0.1733 and is depicted in Figure 6. The \( C^2 \) solution is obtained by the energy minimization technique and has the maximum error 0.0120, which is very convincing and shows that this technique provides a smooth solution with lesser error. Figure 7 is the \( C^2 \) solution; in both figures, the solid line refers to the interpolant, the dotted curve is the original function, and the dashed lines are constraining functions.

The ordinary cubic spline also could be used to solve this interpolating problem, but it may not fulfill the constraining conditions. However, in this example, the spline solution satisfies the constraints. Here, the \( C^1 \) and \( C^2 \) Hermite cubic solutions can be compared with the ordinary spline solution which is obtained by MATLAB. Table 7 provides the corresponding errors in subintervals. According to this report, the \( C^2 \) solution gives a more satisfying result. Also, the \( C^1 \) solution competes well enough. However, in cases where the constraints are tighter, the Hermite spline solutions, both \( C^1 \) and \( C^2 \), are comparably better than ordinary spline solution.

Example 5. In this example, we intend to make more restricting conditions. Consider the function \( f(x) = -0.5xe^{\sin(x)} \) in the interval \([-3.5, 3.5] \) and we have a sample of this function in Table 8. It is aimed to find an interpolant which is bounded by curves \( B_1(x) = -0.3x^2 - 0.7x - 0.4 \) and \( B_2(x) = 0.2x^2 - 0.9x + 0.1 \) as lower and upper constraints.

These curves and the original function as well as the \( C^1 \) and \( C^2 \) CHS solutions are shown in Figures 8 and 9, respectively.
Table 7: Comparison of errors in Example 4.

| [x_i, x_{i+1}] | MATLAB spline | C^2 CHS | C^3 CHS |
|-----------------|---------------|---------|---------|
| [0, 1]          | 0.1112        | 0.0120  | 0.1584  |
| [1, 2]          | 0.0299        | 0.0033  | 0.0425  |
| [2, 3]          | 0.0078        | 0.0007  | 0.0112  |
| [3, 4]          | 0.0024        | 0.0005  | 0.0033  |
| [4, 5]          | 0.0005        | 0.0001  | 0.0008  |
| [5, 6]          | 0.0003        | 0.0005  | 0.0003  |
| [6, 7]          | 0.0014        | 0.0012  | 0.0014  |
| [7, 8]          | 0.0003        | 0.0007  | 0.0002  |
| [8, 9]          | 0.0030        | 0.0010  | 0.0027  |
| [9, 10]         | 0.0153        | 0.0018  | 0.0142  |
| [10, 11]        | 0.0497        | 0.0010  | 0.0456  |
| [11, 12]        | 0.1884        | 0.0009  | 0.1733  |

Table 8: Sampled data from Example 5.

| x_i | -3.5 | -3   | -2   | -1   | 0    | 1    | 2    | 3    | 3.5   |
|-----|------|------|------|------|------|------|------|------|-------|
| f_i | 2.4853 | 1.3026 | 0.4028 | 0.2155 | 0 | -1.1599 | -2.4826 | -1.7273 | -1.2322 |

Figure 7: Constrained C^2 CHS solution to Example 4.

Figure 8: Constrained C^1 CHS solution to Example 5.
7. Concluding Remarks

A special kind of shape-preserving approximation which deals with bounding constraints has been addressed; although it has partially been studied in the literature [12], we propose a linear programming approach and use energy minimization techniques to gain $C^2$ continuity. Quadratic constraints are handled, which give the positivity and linear constraints a special case.

It should be mentioned here that the bounded interpolation problem may be considered as a piecewise monotonic interpolation. In every subinterval, we may select a suitable function which preserves the monotonicity. This is a problem which has extensively been studied [18]. But a monotone solution may not be a suitable approximation for a bounded phenomenon, and the monotone interpolant cannot reach values outside the minimum and maximum of the sampling. This drawback was our motivation to seek for a really bounded approximation.

It is worth indicating two experimental results:

(i) The constrained/bounded cubic Hermite spline is a promising technique for interpolating scattered data

(ii) The energy minimization technique is a convincing tool to gain more smoothness and reduce the approximation error

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] B. Kvasov and B. Kvasoc, Methods of Shape-Preserving Spline Approximation, World Scientific, Singapore, 2000.

[2] Y. Pan and M. Cheng, “A survey on shape preserving interpolation,” in Proceedings of the 2012 International Conference on IEEE Systems and Informatics (ICSAI), pp. 2521–2523, Yantai, China, 2012.

[3] X. Han, “Convexity-preserving piecewise rational quartic interpolation,” SIAM Journal on Numerical Analysis, vol. 46, no. 2, pp. 920–929, 2008.

[4] T. N. T. Goodman, “Shape preserving interpolation by curves,” in Algorithms for Approximation IV, J. Levesley, I. Anderson, and J. Mason, Eds., pp. 24–35, University of Huddersfield, Huddersfield, UK, 2002.

[5] M. Sarfraz, M. Z. Hussain, and M. Hussain, “Shape-preserving curve interpolation,” International Journal of Computer Mathematics, vol. 89, no. 1, pp. 35–53, 2012.

[6] M. Sarfraz, M. Z. Hussain, and F. Hussain, “Shape preserving convex data interpolation,” Applied and Computational Mathematics, International Journal, vol. 16, no. 3, pp. 205–227, 2017.

[7] M. Sarfraz, S. Samreen, and M. Z. Hussain, “A quadratic trigonometric Nu spline with shape control,” International Journal of Image and Graphics, vol. 17, no. 3, pp. 01–17, 2017.

[8] Z. Tariq, F. Ibraheem, M. Z. Hussain, and M. Sarfraz, “Monotone data modeling using rational functions,” Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, no. 3, pp. 2331–2343, 2019.

[9] S. A. Abdul Karim, A. Saaban, V. T. Nguyen, and V. T. Nguyen, “Scattered data interpolation using quartic triangular patch for shape-preserving interpolation and comparison with mesh-free methods,” Symmetry, vol. 12, no. 7, p. 1071, 2020.

[10] C. Anitescu, M. N. Hossain, and T. Rabczuk, “Recovery-based error estimation and adaptivity using high-order splines over hierarchical T-meshes,” Computer Methods in Applied Mechanics and Engineering, vol. 328, pp. 638–662, 2018.

[11] H. Kano, H. Fujioka, and C. F. Martin, “Optimal smoothing and interpolating splines with constraints,” Applied Mathematics and Computation, vol. 218, no. 5, pp. 1831–1844, 2011.

[12] M. R. Asim and K. W. Brodlie, “Curve drawing subject to positivity and more general constraints,” Computers & Graphics, vol. 27, no. 4, pp. 469–485, 2003.

[13] K. W. Brodlie and S. Butt, “Preserving convexity using piecewise cubic interpolation,” Computers & Graphics, vol. 15, no. 1, pp. 15–23, 1991.

[14] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in Proceedings of the 1968 23rd ACM National Conference, pp. 517–524, ACM, New York, NY, USA, 1968.

[15] K. W. Brodlie, M. R. Asim, and K. Unsworth, “Constrained visualization using the Shepard interpolation family,” Computer Graphics Forum, vol. 24, no. 4, 2005.

[16] J. W. Schmidt and W. Hess, “Positivity of cubic polynomials on intervals and positive spline interpolation,” BIT Numerical Mathematics, vol. 28, no. 2, pp. 340–352, 1988.

[17] W. Burmeister, W. Hess, and J. W. Schmidt, “Convex spline interpolants with minimal curvature,” Computing, vol. 35, no. 2, pp. 219–229, 1985.

[18] G. Wolberg and I. Alfy, “An energy-minimization framework for monotonic cubic spline interpolation,” Journal of Computational and Applied Mathematics, vol. 143, no. 2, pp. 145–188, 2002.

[19] X. Qin and Q. Xu, “C1 rational cubic/linear trigonometric interpolation spline with positivity-preserving property,” Engineering Letters, vol. 25, no. 2, 2017.