mathematics

MDPI

Article

C^2 Cubic Algebraic Hyperbolic Spline Interpolating Scheme by Means of Integral Values

Salah Eddargani [1,*], Mohammed Oraiche [2], Abdellah Lamnii [3] and Mohamed Louzar [2]

1 Department of Applied Mathematics, University of Granada, 18071 Granada, Spain
2 MISI Laboratory, Faculty of Sciences and Techniques, Hassan First University of Settat, Settat 26000, Morocco; m.oraiche@uhp.ac.ma (M.O.); mohamed.louzar@uhp.ac.ma (M.L.)
3 LaSAD Laboratory, Ecole Normale Supérieure, Abdelmalek Essaadi University, Tetouan 93030, Morocco; a.lamnii@uae.ac.ma
* Correspondence: seddargani@correo.ugr.es

Abstract: In this paper, a cubic Hermite spline interpolating scheme reproducing both linear polynomials and hyperbolic functions is considered. The interpolating scheme is mainly defined by means of integral values over the subintervals of a partition of the function to be approximated, rather than the function and its first derivative values. The scheme provided is C^2 everywhere and yields optimal order. We provide some numerical tests to illustrate the good performance of the novel approximation scheme.

Keywords: algebraic hyperbolic splines; integro cubic interpolation; Hermite representation

MSC: 65D07

Citation: Eddargani, S.; Oraiche, M.; Lamnii, A.; Louzar, M. C^2 Cubic Algebraic Hyperbolic Spline Interpolating Scheme by Means of Integral Values. *Mathematics* **2022**, *10*, 1490. https://doi.org/10.3390/math10091490

Academic Editor: Ioannis K. Argyros

Received: 21 March 2022
Accepted: 27 April 2022
Published: 29 April 2022

1. Introduction

Nowadays, numerical methods are a common tool, just a click away from the user. Interpolation is a particular and very important numerical method, which is widely used to address the solution of theoretical problems and show their full potential to numerically solve problems that occur in many different branches of science, engineering and economics.

The interpolation approximants should be easily evaluated, differentiated and integrated. Spline functions, i.e., smooth piecewise polynomial functions, fulfil all these requirements.

Since the introduction of the systematic study of spline functions by I. J. Schoenberg in the 1940s [1], they have become an indispensable tool in approximation theory and numerical computation, including computer-aided geometric design (CAGD) [2,3], the numerical solution of PDEs, numerical quadratures, interpolation and quasi-interpolation, regularization, least squares, isogeometric analysis and image processing.

Spline functions have been the subject of many research results that have been presented in well-known books [4,5]. Furthermore, thousands of papers related to spline functions and their applications have been published in the last five years, in fields ranging from computer science, engineering, physics and astronomy to entertainment and conceptual design assistance.

Polynomial spline functions are the most commonly used class, especially due to the fact that they admit normalized bases on any bounded interval $[a, b]$ (Bernstein bases and B-spline bases); for more details, see [5–8]. Indeed, Bernstein bases and B-splines possess several interesting properties, such as non-negativity, local support, partition of unity and totally positive bases [9]. Bernstein basis functions of degree n are the best among all bases of the polynomial space of degree less than or equal to n. This means that it is the basis relative to which the control polygon of any curve yields the best information on the curve itself. The non-polynomial B-splines have also been studied in the literature. For instance, the trigonometric B-splines were presented in [10,11]. The authors in [12] have established a recurrence relation for the trigonometric B-splines of arbitrary order. A complete construction of exponential tension B-splines of arbitrary order was given in [13]. A more recent study on this kind of spline was given in [14,15]. The splines associated with the exponential B-spline space are often referred to as hyperbolic splines [16].

Unfortunately, neither Bernstein bases nor B-splines are suitable to perfectly describe conic sections, which are shapes of major interest in certain engineering applications. This resulted in the introduction of NURBS [17], which can be seen as a generalization of B-splines, inheriting from them important properties and with the additional benefit of making possible the exact representation of conic sections. On the other hand, the NURBS representation suffers from some drawbacks that are considered critical in CAD. In fact, the necessity of weights does not have an evident geometric meaning and their selection is often unclear. Furthermore, it behaves awkwardly with respect to differentiation and integration, which are indispensable operators in analysis. On this concern, it is sufficient to think about the complex structure of the derivative of a NURBS curve of a given order.

An alternative is to use the so-called generalized B-splines; see [18,19] and references therein. The generalized B-splines belong to the extended space spanned by $\{1, x, \ldots, x^{n-2}, u_1(x), u_2(x)\}$, where u_1 and u_2 are smooth functions. The two functions u_1 and u_2 can be selected to achieve the exact representation of salient profiles of interest and/or to obtain particular features. The most popular choices of these functions are: $(u_1(x), u_2(x)) = (\sin(x), \cos(x))$ and $(u_1(x), u_2(x)) = (\exp(x), \exp(-x))$, which yield algebraic trigonometric and algebraic exponential splines, respectively. The algebraic exponential splines are often referred as algebraic hyperbolic splines. Algebraic trigonometric and hyperbolic splines allow an exact representation of conic sections, as well as of some transcendental curves, such as helix and cycloid curves. In fact, they are in a position to provide parametrizations of conic sections that are significantly more related to the arc length than NURBS.

These classes of splines are also known as cycloidal spaces, and they have become the subjects of a considerable amount of research [2,20–26]. The algebraic hyperbolic spaces spanned by the functions 1, x, …, x^{n-2}, $\cosh(x)$, $\sinh(x)$, for $x \in \mathbb{R}$, have been widely considered in the literature; see [21] and references quoted. They yield the tension splines, which are extremely useful for avoiding undesirable oscillations in the interpolation curves [27].

In this paper, we consider the algebraic hyperbolic (AH) cubic spline space, spanned by $\{1, x, \sinh, \cosh\}$, and let $X_n = \{x_i := a + ih\}_{i=0}^{n}$ be a uniform partition of a bounded interval $\mathfrak{I} = [a, b]$, with $h = \frac{b-a}{n}$. Given values f_i^j, $i = 0, \ldots, n$, $j = 0, 1$, there exists a unique cubic AH Hermite interpolant $s_i \in span\{1, x, \sinh, \cosh\}$ such that $s_i^{(j)}(x_{i+k}) = f_{i+k}^j$, $i = 0, \ldots, n-1$, $j, k = 0, 1$. The spline s defined from the local interpolants s_i is a C^1 continuous AH spline that interpolates the data f_i^j.

In this work, we suppose that the data f_i^j, $i = 0, \ldots, n$, $j = 0, 1$, are not given, and we assume that the integral values over subintervals $[x_i, x_{i+1}]$, $i = 0, \ldots, n-1$ are given. Then, our purpose is the construction of the Hermite interpolant s using only this information. This kind of approximation arises in various fields, such as mechanics, mathematical statistics, electricity, environmental science, climatology, oceanography and so on (for more details, see References [28,29] and references therein). More precisely, the values f_i^j will be computed by means of C^2 smoothness conditions at the knots x_i and the integral values over subintervals $[x_i, x_{i+1}]$. This is done by solving a three-diagonal linear system. Some final conditions are required. In particular, we assume that the three values f_0^0, f_n^0 and f_0^1 or f_n^1 are given. In general, these three values are not always available. We suggest a modified scheme that does not involve any final conditions to avoid this limitation.

Integro spline approximation was treated in various works in the literature. The author in [30,31] developed two types of integro spline approximants, cubic and quintic cases, respectively. The two schemes introduced in [30,31] require various end conditions and the solution of a three-diagonal system of linear equations. Solving a linear system of equations sometimes is very expensive, so the authors in [32] developed cubic integro splines quasi-interpolant without solving any system of equations. An integro quartic spline scheme has been constructed in [33]. The authors in [21,34] provided some integro spline schemes for the case of non-polynomial splines. More recent work on the integro spline approximation is given in [35,36]

In this work, a new class of integro spline approximant is introduced. The proposed operator is C^2 smooth everywhere and exactly reproduces both linear polynomials and hyperbolic functions, which is useful to avoid undesirable oscillations in curves' interpolation. Some end conditions are needed, and to avoid this inconvenience, we have proposed a modified scheme that does not require additional end conditions.

This paper is organized as follows: in Section 2, we first present the C^1 cubic AH interpolating scheme, which is defined by means of the value and the derivative value at each knot of the partition. We also study the error bound of the presented scheme. Then, the conditions for achieving the C^2 smoothness are described. Thus, an approach to define the integro spline scheme by means of integral values is proposed. In Section 3, we provide some numerical tests. Finally, we present some conclusions.

2. Algebraic Hyperbolic Spline Interpolation

Let $X_n = \{x_i := a + ih\}_{i=0}^{n}$ be a uniform partition of a bounded interval $\mathfrak{I} = [a, b]$, with $h = \frac{b-a}{n}$.

2.1. Cubic Algebraic Hyperbolic (AH) Splines of Class C^1

The construction of a C^1 cubic AH spline interpolant on the partition X_n should be locally expressed in each sub-interval $\ell_i := [x_i, x_{i+1}]$ in terms of the function and the first derivative values of the approximated function at knots x_i and x_{i+1}.

The space of cubic AH splines on X_n with global C^1 continuity is denoted as

$$S_3^1(X_n) := \Big\{ s \in C^1(\mathfrak{I}); s_{|\ell_i} \in \Gamma_3 \text{ for all } i = 0, \ldots, n \Big\},$$

where $\Gamma_3 := span\{1, x, \sinh, \cosh\}$ stands for the linear space of cubic AH splines. Its dimension equals $2(n+1)$. The following Hermite interpolation problem can then be considered: there exists a unique spline $s(x) \in S_3^1(X_n)$ such that

$$s^{(j)}(x_i) = f_i^j, \quad i = 0, \ldots, n,\ j = 0, 1, \tag{1}$$

for any given set of f_i^j-values. Therefore, each spline $s(x) \in S_3^1(X_n)$ restricted to the sub-interval ℓ_i can be represented as follows:

$$s_{|\ell_i}(x) := \sum_{k=0}^{1} \sum_{j=0}^{1} f_{i+k}^j \phi_{i+k}^j(x), \tag{2}$$

in which ϕ_{i+k}^j, $k = 0, 1$, $j = 0, 1$ are classical Hermite basis functions of $S_3^1(X_n)$ restricted to the sub-interval ℓ_i. The basis functions ϕ_{i+k}^j, $k = 0, 1$, $j = 0, 1$ are the unique solution of the interpolation problem given by (1) in $s(x) \in S_3^1(X_n)$, where $(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1) = (1, 0, 0, 0)$, $(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1) = (0, 1, 0, 0)$, $(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1) = (0, 0, 1, 0)$ and $(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1) = (0, 0, 0, 1)$, respectively. More precisely,

$$\begin{aligned}
&\phi_i^0(x_i)=1, && \left(\phi_i^0\right)'(x_i)=0, && \phi_i^0(x_{i+1})=0, && \left(\phi_i^0\right)'(x_{i+1})=0,\\
&\phi_i^1(x_i)=0, && \left(\phi_i^1\right)'(x_i)=1, && \phi_i^1(x_{i+1})=0, && \left(\phi_i^1\right)'(x_{i+1})=0,\\
&\phi_{i+1}^0(x_i)=0, && \left(\phi_{i+1}^0\right)'(x_i)=0, && \phi_{i+1}^0(x_{i+1})=1, && \left(\phi_i^0\right)'(x_{i+1})=0,\\
&\phi_{i+1}^1(x_i)=0, && \left(\phi_{i+1}^1\right)'(x_i)=0, && \phi_{i+1}^1(x_{i+1})=0, && \left(\phi_{i+1}^1\right)'(x_{i+1})=1.
\end{aligned}$$

They can be expressed as follows:

$$\phi_i^0(x) := \frac{1}{\theta}(1,x,\sinh(x),\cosh(x))\begin{pmatrix} -\cosh(h)+\sinh(h)(h+x_i)+1 \\ -\sinh(h) \\ \sinh(h+x_i)-\sinh(x_i) \\ \cosh(x_i)-\cosh(h+x_i) \end{pmatrix},$$

$$\phi_i^1(x) := \frac{1}{\theta}(1,x,\sinh(x),\cosh(x))\begin{pmatrix} -\cosh(h)-\sinh(h)x_i+1 \\ \sinh(h) \\ \sinh(x_i)-\sinh(h+x_i) \\ \cosh(h+x_i)-\cosh(x_i) \end{pmatrix},$$

$$\phi_{i+1}^0(x) := \frac{1}{\theta}(1,x,\sinh(x),\cosh(x))\begin{pmatrix} h\cosh(h)-\sinh(h)+(\cosh(h)-1)x_i \\ 1-\cosh(h) \\ \cosh(x_i)-\cosh(h+x_i)+h\sinh(h+x_i) \\ -h\cosh(h+x_i)-\sinh(x_i)+\sinh(h+x_i) \end{pmatrix},$$

$$\phi_{i+1}^1(x) := \frac{1}{\theta}(1,x,\sinh(x),\cosh(x))\begin{pmatrix} -h+\sinh(h)+(\cosh(h)-1)x_i \\ 1-\cosh(h) \\ -\cosh(x_i)+\cosh(h+x_i)-h\sinh(x_i) \\ h\cosh(x_i)+\sinh(x_i)-\sinh(h+x_i) \end{pmatrix},$$

where $\theta = h\sinh(h) - 2\cosh(h) + 2$.

Figure 1 shows the typical plots of the basis functions ϕ_{i+k}^j, $k=0,1$, $j=0,1$, on the sub-interval $[x_i, x_{i+1}] = [0,1]$.

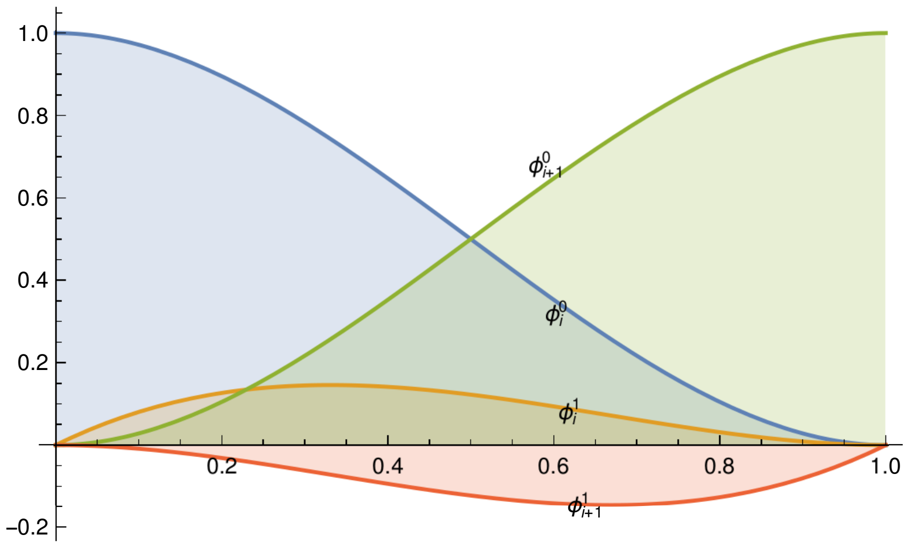

Figure 1. Plots of the basis functions on $[x_i, x_{i+1}] = [0,1]$.

Consider that the values of f_i^j are related to an explicit function f, i.e., $f_i^j = f^{(j)}(x_i)$, $j=0,1$. Then, it holds that

$$\|s-f\|_{\infty,[x_i,x_{i+1}]} = \mathcal{O}\left(h^4\right). \tag{3}$$

In order to prove this statement, consider an arbitrary but fixed value $\tilde{t}$ different from x_i and x_{i+1}, and define

$$R(t) = s(t) - f(t) - \rho \prod_{k=0}^{1} (t - x_{i+k})^2,$$

in which the constant ρ is chosen such that $R(\tilde{t}) = 0$, that is,

$$\rho = \frac{s(\tilde{t}) - f(\tilde{t})}{\prod_{k=0}^{1} (\tilde{t} - x_{i+k})^2}.$$

The function R has at least three roots in $[x_i, x_{i+1}]$, which are x_i, x_{i+1} and $\tilde{t}$.

According to Rolle's theorem, R' has at least two roots in $[x_i, x_{i+1}]$ that are different from x_i, x_{i+1}, $\tilde{t}$ and also $R'(x_i) = R'(x_{i+1}) = 0$, which means that R' has at least four roots in $[x_i, x_{i+1}]$. Analogously and progressively, it is shown that $R^{(2)}$ has three roots in the interval $[x_i, x_{i+1}]$, $R^{(3)}$ has two and $R^{(4)}$ has only one root, say ξ. It then states

$$R^{(4)}(\xi) = s^{(4)}(\xi) - f^{(4)}(\xi) - 24\rho = 0.$$

It holds that

$$s(\tilde{t}) - f(\tilde{t}) = \frac{1}{24} \left(s^{(4)}(\xi) - f^{(4)}(\xi) \right) \prod_{k=0}^{1} (\tilde{t} - x_{i+k})^2.$$

This proves Equation (3).

The proposed interpolant s is defined from the values and first derivative values at the break points. Unfortunately, this dataset is not always at hand. This paper deals with the case where neither the values nor the derivative values are known. Instead, we assume that the integral values over the sub-intervals are available.

The strategy pursued in this work is the following: we first highlight the relationship between function and first derivative values by imposing the C^2 smoothness at the set of break points. Next, we express the derivative values in the mean integral values provided.

The C^2 smoothness of $s''(x)$ at x_i, $i = 1, ..., n-1$ yields the following consistency relations:

$$\alpha f^1_{i-1} + \beta f^1_i + \alpha f^1_{i+1} = f^0_{i+1} - f^0_{i-1}, \qquad i = 1, ..., n-1. \tag{4}$$

where $\alpha = \frac{\sinh(h) - h}{\cosh(h) - 1}$ and $\beta = \operatorname{csch}^2\left(\frac{h}{2}\right)(h\cosh(h) - \sinh(h))$.

The error related to the approximation scheme (4) can be derived from the following result.

Lemma 1. *Let $f \in \mathcal{C}^3([a, b])$; then, the local truncation errors t_i, $i = 1, 2, \ldots, n$, associated with the scheme (4) are given by the expressions*

$$\mathrm{t}_i = -\frac{1}{6} h^2 f^{(3)}(x_i)(h(\cosh(h) + 2) - 3\sinh(h)) + \mathcal{O}\left(h^2\right).$$

Proof. The function f is supposed to be of class $\mathcal{C}^3([a,\,b])$, that is,

$$\begin{aligned} f(x_i+h) &= f(x_i) + f'(x_i)\,h + f''(x_i)\frac{h^2}{2} + f^{(3)}(x_i)\frac{h^3}{6} + \mathcal{O}\Big(h^3\Big),\\ f(x_i-h) &= f(x_i) - f'(x_i)\,h + f''(x_i)\frac{h^2}{2} - f^{(3)}(x_i)\frac{h^3}{6} + \mathcal{O}\Big(h^3\Big),\\ f'(x_i+h) &= f'(x_i) + f''(x_i)h + f^{(3)}(x_i)\frac{h^2}{2} + \mathcal{O}\Big(h^2\Big),\\ f'(x_i-h) &= f'(x_i) - f''(x_i)h + f^{(3)}(x_i)\frac{h^2}{2} + \mathcal{O}\Big(h^2\Big). \end{aligned}$$

Using Equation (4), it results that

$$\mathrm{t}_i = \alpha f'(x_{i-1}) + \beta f'(x_i) + \alpha f'(x_{i+1}) - \frac{1}{\operatorname{csch}^2\Big(\frac{h}{2}\Big)}(f(x_{i+1}) - f(x_{i-1})).$$

By replacing $f'(x_{i-1})$, $f'(x_{i+1})$ and $f(x_{i+1}) - f(x_{i-1})$ by their Taylor expansions, the intended result can be achieved, which completes the proof. □

The next result can be easily deduced from the previous lemma.

Theorem 2. *Let $f \in \mathcal{C}^3([a,\,b])$, then*

$$\Big|f_i^1 - f'(x_i)\Big| = \mathcal{O}\Big(h^2\Big).$$

At this point, we have simply provided a scheme that approximates the derivative values from the function values by imposing C^2 smoothness at the break points. In what follows, we will deal with the case where the function values themselves are not available, whereas the integrals over the sub-intervals are known.

2.2. Cubic Algebraic Hyperbolic Spline Interpolant Based on Mean Integral Value

In traditional spline interpolation problems, it is assumed that the function values at the knots are given. In this subsection, the function values are supposed to be unknowns and we assume that the integrals over the sub-intervals $[x_{i-1}, x_i]$, $i = 1, \ldots, n$ are provided and are equal to

$$t_i = \int_{x_{i-1}}^{x_i} f(x)\,dx, \quad i = 1, \ldots, n \tag{5}$$

In the sequel, we will provide a scheme that approximates derivative values from the integrals t_i, $i = 1, \ldots, n$.

By integrating s over $[x_{i-1}, x_i]$ and $[x_i, x_{i+1}]$, one can obtain

$$2t_i = h\Big(f_{i-1}^0 + f_i^0\Big) + \left(2 - h\coth\left(\frac{h}{2}\right)\right)\Big(f_i^1 - f_{i-1}^1\Big), \quad i = 1, \ldots, n \tag{6}$$

$$2t_{i+1} = h\Big(f_i^0 + f_{i+1}^0\Big) + \left(2 - h\coth\left(\frac{h}{2}\right)\right)\Big(f_{i+1}^1 - f_i^1\Big), \quad i = 0, \ldots, n-1 \tag{7}$$

respectively.

Subtracting (6) from (7) as a first step, and then applying (4) as a second step, allows us to eliminate the unknowns f_i^0, as well as to achieve new relations that link only the unknowns f_i^1 with the provided data t_i. It results that

$$\mu f_{i-1}^1 + \lambda f_i^1 + \mu f_{i+1}^1 = 2(t_{i+1} - t_i), \quad i = 1, \ldots, n-1, \tag{8}$$

with

$$\mu = \frac{1}{2}\left(4 - h^2\operatorname{csch}^2\left(\frac{h}{2}\right)\right) \text{ and } \lambda = \left(\Big(h^2 - 2\Big)\cosh(h) + 2\right)\operatorname{csch}^2\left(\frac{h}{2}\right)$$

This yields a system of $n-1$ linear equations, while there are $n+1$ unknowns f_i^1. Then, two additional end conditions are required to determine the unknowns. The end conditions are the first derivative values at the end points a and b. Assume that $f'(a)=f_a^1$ and $f'(b)=f_b^1$ are provided. Then, a $(n-1)\times(n-1)$ linear tridiagonal system results.

$$\begin{pmatrix} \lambda & \mu & & & & \\ \mu & \lambda & \mu & & & \\ & \mu & \lambda & \mu & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu & \lambda & \mu \\ & & & & \mu & \lambda \end{pmatrix} \begin{pmatrix} m_1 \\ \\ \vdots \\ \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} b_1-\mu c \\ b_2 \\ \vdots \\ b_{n-2} \\ b_{n-1}-\mu d \end{pmatrix}, \tag{9}$$

with $b_i=2(t_{i+1}-t_i)$.

The following result shows that the linear system (9) has a unique solution.

Theorem 3. *For $h>0$, the matrix system in (9) is a strictly diagonally dominant matrix.*

Proof. Let $h>0$. It is easy to show that $\lambda>2\mu$. Indeed, a simple calculation gives us the following equality:

$$\begin{aligned} \lambda-2\mu &= \frac{\left((h^2-2)\cosh(h)+2\right)\operatorname{csch}^2\left(\frac{h}{2}\right)}{h\left(h\coth\left(\frac{h}{2}\right)-2\right)} - \frac{2\left(4-h^2\operatorname{csch}^2\left(\frac{h}{2}\right)\right)}{2h\left(h\coth\left(\frac{h}{2}\right)-2\right)} \\ &= \frac{4}{h}+2\coth\left(\frac{h}{2}\right) \end{aligned}$$

Since $\coth\left(\frac{h}{2}\right)>0$, then $\lambda-2\mu>0$, which completes the proof. □

The LU factorization is adequate for solving (9) because the index on the diagonally dominant property of the matrix A is equal to $\frac{1}{5}$ for small enough $h>0$. In fact, let $A:=\left(a_{ij}\right)_{1\le i,j\le n}$ be the matrix coefficient of system (9). The index on the diagonally dominant property of matrix A is given by

$$D_h(A) := \max_{i=1,\ldots,n} \frac{1}{|a_{ii}|}\sum_{\substack{j=1\\ j\neq i}}^{n}|a_{i,j}| = 2\max_{h>0}\frac{\mu(h)}{\lambda(h)} = \frac{\frac{2}{h}-\frac{h}{15}+\frac{13h^3}{6300}+\mathcal{O}(h^5)}{\frac{10}{h}+\frac{4h}{15}-\frac{11h^3}{3150}+\mathcal{O}(h^5)}$$

Its limit when h is close to zero is equal to $\frac{1}{5}$.

Once the values of f_i^1, $i=1,\ldots,n-1$ are determined, we can then compute f_i^1, $i=1,\ldots,n-1$ by means of (6). However, we still need another end condition. Suppose that one of the values $f(a)$ and $f(b)$ is provided; then, we can start from it and use (6) to obtain the remaining unknowns in an iterative way.

To analyze the interpolation error, we need the following lemma to establish an error bound for our operator.

Lemma 4. *Let $f\in\mathcal{C}^3([a,b])$; then, the local truncation errors $\tilde{\mathfrak{t}}_i$, $i=1,2,\ldots,n$, associated with the scheme (9), are given by the expressions*

$$\tilde{\mathfrak{t}}_i = -\frac{1}{6}h^2 f^{(3)}(x_i)\left(h^2+3h^2\operatorname{csch}^2\left(\frac{h}{2}\right)-12\right)+\mathcal{O}\left(h^2\right).$$

Theorem 5. *Let $f\in\mathcal{C}^4([a,b])$ and s be the interpolation operator defined by (2), (6) and (9). For a uniform step size h, we have*

$$\|f(x)-s(x)\|_{\infty,[a,b]} = \mathcal{O}(h^2).$$

Proof. Consider the sub-interval $[x_i, x_{i+1}]$. Let p be the spline in S_3^1, which satisfies $p^{(j)}(x_{i+k}) = f^{(j)}(x_{i+k})$, $j, k = 0, 1$. Then, it results that

$$\begin{aligned}\|s - f\| &= \|s - p + p - f\| \\ &\le \|s - p\| + \|p - f\| \\ &\le \mathcal{O}\left(h^2\right) + \mathcal{O}\left(h^4\right) \\ &\equiv \mathcal{O}\left(h^2\right).\end{aligned}$$

This due to the fact that $\|s - p\| \le \max\{|f'(x_i) - f_i^1|, |f'(x_{i+1}) - f_{i+1}^1|\}$, which concludes the proof. □

In the general context, the end conditions may not be provided. Thus, to avoid this limitation, we will provide explicit expressions for the end conditions $f(a)$, $f'(a)$ and $f(b)$ by means of integral values.

Lemma 6. *For a given function $f \in C^2$,*

$$f(a) = \frac{11t_1 - 7t_2 + 2t_3}{6h}, \; f'(a) = -\frac{2t_1 - 3t_2 + t_3}{h^2}, \; f'(b) = -\frac{2t_n - 3t_{n-1} + t_{n-2}}{h^2}.$$

Proof. The Taylor expansion of f around a is as follows:

$$f(x) = f(a) + f'(a)(x - a) + \frac{1}{2}f''(a)(x - a)^2 + \mathcal{O}(x - a)^2.$$

$$\begin{aligned}t_1 &= hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + \mathcal{O}\left(h^3\right), \\ t_2 &= hf(a) + \frac{3h^2}{2}f'(a) + \frac{7h^3}{6}f''(a) + \mathcal{O}\left(h^3\right), \\ t_3 &= hf(a) + \frac{5h^2}{2}f'(a) + \frac{19h^3}{6}f''(a) + \mathcal{O}\left(h^3\right).\end{aligned}$$

Therefore, the following system is obtained:

$$\begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} h & \frac{h^2}{2} & \frac{h^3}{6} \\ h & \frac{3h^2}{2} & \frac{7h^3}{6} \\ h & \frac{5h^2}{2} & \frac{19h^3}{6} \end{pmatrix} \begin{pmatrix} f(a) \\ f'(a) \\ f''(a) \end{pmatrix}$$

Through a straightforward computation, we can obtain the expressions of $f(a)$ and $f'(a)$.

By the same approach, we can obtain the expression of $f'(b)$, which concludes the proof. □

3. Numerical Results

This section provides some numerical results to illustrate the performance of the above Hermite interpolation operator. To this end, we will use the test functions

$$\begin{aligned}f_1(x) &= \frac{3}{4}e^{-2(9x-2)^2} - \frac{1}{5}e^{-(9x-7)^2-(9x-4)^2} + \frac{1}{2}e^{-(9x-7)^2-\frac{1}{4}(9x-3)^2} + \frac{3}{4}e^{\frac{1}{10}(-9x-1)-\frac{1}{49}(9x+1)^2}, \\ f_2(x) &= \frac{1}{2}x\cos^4\left(4\left(x^2 + x - 1\right)\right), \\ f_3(x) &= -\frac{\exp(-x^2)\left(\log(x^5 + 6) + \sin(3\pi x)\right)}{\cos(2\pi x) + 2},\end{aligned}$$

whose plots appear in Figure 2. The two first functions are the 1D versions of the Franke [37] and Nielson [38] functions.

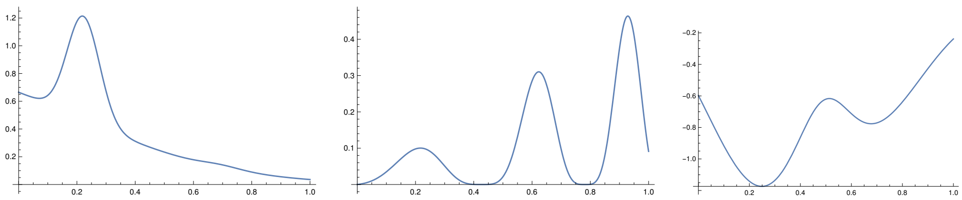

Figure 2. Plots of test functions: f_1 (**left**), f_2 (**center**) and f_3 (**right**).

Let us consider the interval $I = [0, 1]$. The tests are carried out for a sequence of uniform mesh $\mathfrak{I}_n$ associated with the break points ih, $i = 0, \ldots, n$, where $h = \frac{1}{n}$.

The interpolation error is estimated as

$$\mathcal{E}_n(f) = \max_{0 \le \ell \le 200} |s(x_\ell) - f(x_\ell)|,$$

where x_ℓ, $\ell = 0, \ldots, 200$, are equally spaced points in I. The estimated numerical convergence order (NCO) is given by the rate

$$\mathrm{NCO} := \frac{\log\left(\frac{\mathcal{E}_n}{\mathcal{E}_{2n}}\right)}{\log(2)}.$$

In Table 1, the estimated quasi-interpolation errors and NCOs for functions f_1, f_2 and f_3 are shown.

Table 1. Estimated errors for functions f_1, f_2 and f_3, and NCOs with different values of n.

n	$\mathcal{E}_n(f_1)$	NCO	$\mathcal{E}_n(f_2)$	NCO	$\mathcal{E}_n(f_3)$	NCO
10	1.7857×10^{-1}	$--$	9.1243×10^{-2}	$--$	7.4186×10^{-3}	$--$
20	8.7411×10^{-3}	4.3525	9.8171×10^{-3}	3.2163	2.8348×10^{-4}	4.7097
40	1.8198×10^{-4}	5.5859	2.3654×10^{-4}	5.3751	1.0365×10^{-5}	4.7734
80	8.6397×10^{-6}	4.3967	1.1330×10^{-5}	4.3838	5.7600×10^{-7}	4.1695

Now, we will compare the numerical method proposed here with the results obtained with different methods in other papers in the literature, although the test functions in these papers are extremely simple, namely

$$g_1(x) = \cos(\pi x) \quad \text{and} \quad g_2(x) = x\sin(x) \quad .$$

In Tables 2 and 3, we list the resulting errors for the approximation of the functions g_1 and g_2, respectively, by using the cubic spline operator provided here and those in References [34,39,40]. Tables 2 and 3 show that the novel numerical scheme improves the results in References [34,39,40].

Table 2. Estimated errors for function g_1, and NCOs with different values of n.

n	$\mathcal{E}_n(g_1)$	NCO	Method in [34]	NCO	Method in [39]	NCO
10	3.00×10^{-5}	$--$	6.00×10^{-5}	$--$	6.25×10^{-4}	$--$
20	1.86×10^{-6}	4.01	3.28×10^{-6}	4.19	4.05×10^{-5}	3.94
40	1.16×10^{-7}	4.00	2.43×10^{-7}	3.75	2.56×10^{-6}	3.98

Table 3. Estimated errors for function g_2, and NCOs with different values of n.

n	$\mathcal{E}_n(g_2)$	NCO	Method in [40]	NCO
10	1.66×10^{-6}	$--$	2.80×10^{-5}	$--$
20	1.04×10^{-7}	4.00	1.75×10^{-7}	3.99
40	6.51×10^{-9}	4.00	1.10×10^{-8}	3.99

Next, we deal with the following four test functions also defined on $[0,1]$:

$$k_1(x) = \frac{\sqrt{x+2}\exp(2x^2)\sin(4\pi x)}{(x^2+3)^{5/7}}, \qquad k_2(x) = \frac{\sinh(x^2)\sin\left(2\pi\sqrt{\cosh(2x)}\right)}{x^6+1},$$

$$k_3(x) = \frac{\exp\left(\frac{1}{x^2+1}\right)\tanh\left(\frac{x}{10\pi}\right)}{16x^3+1}, \quad \text{and} \quad k_4(x) = \cosh(x)\exp(\sinh(x)).$$

Their typical plots are shown in Figure 3.

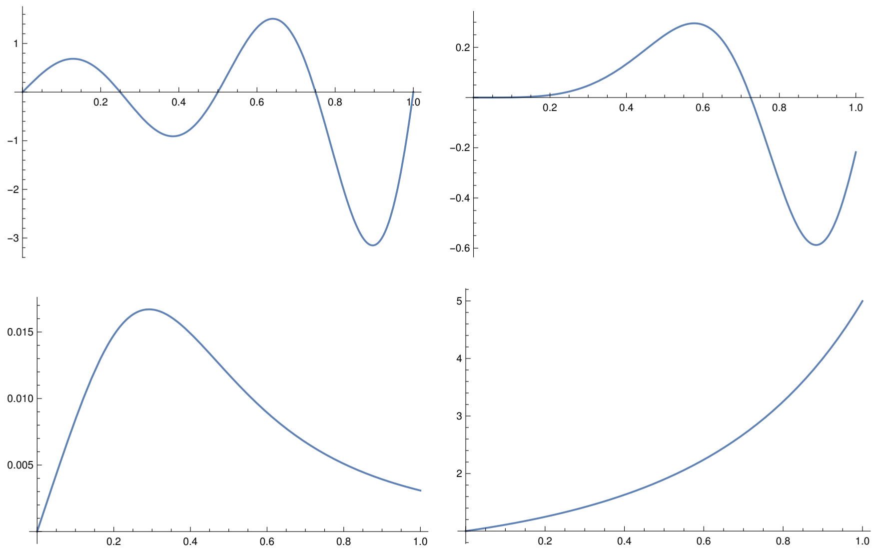

Figure 3. Plots of test functions: k_1, k_2, k_3 and k_4 (from (**left**) to (**right**) and from (**top**) to (**bottom**)).

Our goal now is to compare the method introduced herein with the methods that use only algebraic splines, as well as those that combine the features of algebraic and hyperbolic functions. To this end, we consider the approaches presented by A. Boujraf et al. in [29], D. Barrera et al. in [21], J. Wu and X. Zhang in [40] and S. Eddargani et al. in [34], so we implement the approaches described in [29,34,40] to be able to execute any test function, because those involved in the cited references are simple examples and one of them (exponential function) is reproduced by our approach.

In Tables 4 and 5, we list the resulting errors for the approximation of the functions k_1 and k_2, respectively, using our approach and the one provided in [29]. Table 6 shows the resulting errors for the approximation of the function k_3, using the approach described here and those in References [21,34,40]. In Table 7, we list the resulting errors for the approximation of the function k_4 by using the approach provided here and those in References [34,40].

It is clear that the proposed scheme improves the results in the previous papers by at least two orders of magnitude. Consequently, in certain contexts, it is highly recommended to use splines that benefit from the features of both algebraic and hyperbolic functions instead of using only the features of algebraic ones.

Table 4. Estimated errors for function k_1, and NCOs with different values of n.

n	$\mathcal{E}_n(k_1)$	NCO	Method in [29]	NCO
8	3.6083×10^{-2}	$--$	7.8090×10^{-1}	$--$
16	2.5592×10^{-3}	3.81755	6.3999×10^{-2}	3.60902
32	1.6951×10^{-4}	3.91625	4.6292×10^{-3}	3.78922
64	1.0783×10^{-5}	3.9745	1.5127×10^{-3}	1.61358
128	6.8819×10^{-7}	3.96986	5.2353×10^{-4}	1.53083

Table 5. Estimated errors for function k_2, and NCOs with different values of n.

n	$\mathcal{E}_n(k_2)$	NCO	Method in [29]	NCO
8	5.0763×10^{-3}	$--$	1.2442×10^{-1}	$--$
16	3.6283×10^{-4}	3.80642	1.0854×10^{-2}	3.51895
32	1.8540×10^{-5}	4.29057	7.5616×10^{-4}	3.84338
64	9.9072×10^{-7}	4.22602	4.7513×10^{-5}	3.99231
128	7.4838×10^{-8}	3.72664	3.0252×10^{-6}	3.97319

Table 6. Estimated errors for function k_3, and NCOs with different values of n.

n	$\mathcal{E}_n(k_3)$	NCO	Method in [21]	NCO	Method in [40]	NCO	Method in [34]	NCO
8	7.78×10^{-5}	$--$	1.24×10^{-3}	$--$	1.64×10^{-2}	$--$	1.69×10^{-2}	$--$
16	1.93×10^{-6}	5.33	7.81×10^{-5}	3.98	1.00×10^{-3}	4.03	1.01×10^{-3}	4.27
32	1.03×10^{-7}	4.21	4.88×10^{-6}	4.00	6.26×10^{-5}	3.99	6.28×10^{-5}	4.00
64	6.02×10^{-9}	4.10	3.05×10^{-7}	4.00	3.91×10^{-6}	4.00	3.92×10^{-6}	4.00
128	4.91×10^{-10}	3.61	1.90×10^{-8}	4.00	2.44×10^{-7}	4.00	2.45×10^{-7}	4.00

Table 7. Estimated errors for function k_4, and NCOs with different values of n.

n	$\mathcal{E}_n(k_3)$	NCO	Method in [40]	NCO	Method in [34]	NCO
8	9.41×10^{-5}	$--$	3.25×10^{-4}	$--$	3.83×10^{-4}	$--$
16	7.70×10^{-6}	3.61	2.69×10^{-5}	3.59	3.16×10^{-5}	3.59
32	5.19×10^{-7}	3.88	1.95×10^{-6}	3.78	2.29×10^{-6}	3.79
64	3.06×10^{-8}	4.08	1.32×10^{-7}	3.88	1.54×10^{-7}	3.90

4. Conclusions

Approximation from integral values represents a critical topic because of its extensive application in many different areas. This paper considers a cubic Hermite spline interpolant that reproduces linear polynomials and hyperbolic functions. The proposed interpolant is C^2 everywhere and is defined from the value and the first derivative value at each knot of the partition. The function and derivative values are assumed to be unknowns, and then they are determined using the C^2 smoothness conditions and the mean integral values. The numerical results illustrate the good performance of the novel approximation scheme. The construction used herein requires the resolution of a system of linear equations, which can be computationally expensive, especially when dealing with a large number of data. Future work will address the issue of avoiding this limitation.

Author Contributions: Supervision, A.L. and M.L.; Writing—original draft, M.O.; Writing—review & editing, S.E. All authors have read and agreed to the published version of the manuscript.

Funding: The first author acknowledges partial financial support by the Department of Applied Mathematics of the University of Granada.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank the anonymous referees for their very pertinent and useful comments, which helped them to improve the original manuscript. The first author would like to thank the Department of Applied Mathematics of the University of Granada for the financial support for the research stay during which this work was carried out. The authors wish to thank the Hassan First University of Settat for the financial aid offered for the final cost of the APC.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Schoenberg, I.J. Contributions to the problem of approximation of equidistant data by analytic functions. Part A. on the problem of smoothing or graduation. A first class of analytic approximation formulae. *Q. Appl. Math.* **1946**, *4*, 45–99. [CrossRef]
2. Zheng, J.Y.; Hu, G.; Ji, X.M.; Qin, X.Q. Quintic generalized Hermite interpolation curves: construction and shape optimization using an improved GWO algorithm. *Comput. Appl. Math.* **2022**, *41*, 1–29. [CrossRef]
3. Ammad, M.; Misro, M.Y.; Abbas, M.; Majeed, A. Generalized Developable Cubic Trigonometric Bézier Surfaces. *Mathematics* **2021**, *9*, 283. [CrossRef]
4. Schumaker, L.L. *Spline Functions: Basic Theory*; Cambridge Mathematical Library; Cambridge University Press: Cambridge, UK, 2007; pp. I–Vi.
5. De Boor, C. A Practical Guide to Splines. In *Applied Mathematical Sciences*; Springer: Berlin/Heidelberg, Germany, 1978; Volume 27.
6. Barrera, D.; Eddargani, S.; Lamnii, A. A novel construction of B-spline-like bases for a family of many knot spline spaces and their application to quasi-interpolation. *J. Comput. Appl. Math.* **2022**, *404*, 113761. [CrossRef]
7. Barrera, D.; Eddargani, S.; Ibáñez, M.J.; Lamnii, A. A new approach to deal with C^2 cubic splines and its application to super-convergent quasi-interpolation. *Math. Comput. Simul.* **2022**, *194*, 401–415. [CrossRef]
8. Ershov, S.N. B-Splines and Bernstein Basis Polynomials. *Phys. Part. Nuclei Lett.* **2019**, *16*, 593–601. [CrossRef]
9. Yu, Y.Y.; Ma, H.; Zhu, C.G. Total positivity of a kind of generalized toric-Bernstein basis. *Linear Algebra Appl.* **2019**, *579*, 449–462. [CrossRef]
10. Koch, P.; Lyche, T.; Neamtu, M.; Schumaker, L. Control curves and knot insertion for trigonometric splines. *Adv. Comput. Math.* **1995**, *3*, 405–424. [CrossRef]
11. Walz, G. Identities for trigonometric B-splines with an application to curve design. *BIT* **1997**, *37*, 189–201. [CrossRef]
12. Lyche, T.; Winther, R. A stable recurrence relation for trigonometric B-splines. *J. Approx. Theory* **1979**, *25*, 266–279. [CrossRef]
13. Koch, P.E.; Lyche, T. Construction of exponential tension B-splines of arbitrary order. In *Curves and Surfaces*; Laurent, P.J., Le Méhauté, A., Schumaker, L.L., Eds.; Academic Press: New York, NY, USA, 1991; pp. 255–258.
14. Conti, C.; Gemignani, L.; Romani, L. Exponential Pseudo-Splines: Looking beyond Exponential B-splines. *J. Math. Anal. Appl.* **2016**, *439*, 32–56. [CrossRef]
15. Campagna, R.; Conti, C.; Cuomo, S. Smoothing exponential-polynomial splines for multi-exponential decay data. *Dolomites Res. Notes Approx.* **2019**, *12*, 86–100.
16. Campagna, R.; Conti, C. Penalized hyperbolic-polynomial splines. *Appl. Math. Lett.* **2021**, *118*, 107159. [CrossRef]
17. Yang, X. Fitting and fairing Hermite-type data by matrix weighted NURBS curves. *Comput.-Aided Des.* **2018**, *102*, 22–32. [CrossRef]
18. Speleers, H. Algorithm 1020: Computation of Multi-Degree Tchebycheffian B-Splines. *ACM Trans. Math. Softw.* **2022**, *48*, 1–31. [CrossRef]
19. Liu, X.; Divani, A.A.; Petersen, A. Truncated estimation in functional generalized linear regression models. *Comput. Stat. Data Anal.* **2022**, *169*, 107421. [CrossRef]
20. Eddargani, S.; Lamnii, A.; Lamnii, M.; Sbibih, D.; Zidna, A. Algebraic hyperbolic spline quasi-interpolants and applications. *J. Comput. Appl. Math.* **2019**, *347*, 196–209. [CrossRef]
21. Barrera, D.; Eddargani, S.; Lamnii, A. Uniform algebraic hyperbolic spline quasi-interpolant based on mean integral values. *Comp. Math. Methods* **2021**, *3*, e1123. [CrossRef]
22. Carnicer, J.M.; Mainar, E.; Peña, M. Interpolation on cycloidal spaces. *J. Approx. Theory* **2014**, *187*, 18–29. [CrossRef]
23. Mazure, L. From Taylor interpolation to Hermite interpolation via duality. *Jaen J. Approx.* **2012**, *4*, 15–45.
24. Ait-Haddou, R.; Mazure, M.L.; Ruhland, H. A remarkable Wronskian with application to critical lengths of cycloidal spaces. *Calcolo* **2019**, *56*, 45–56. [CrossRef]
25. Barrera, D.; Eddargani, D.; Lamnii, A.; Oraiche, M. On non polynomial monotonicity-preserving C^1 spline interpolation. *Comp. Math. Methods* **2021**, *3*, e1160. [CrossRef]
26. Ajeddar, M.; Lamnii, A. Smooth reverse subdivision of uniform algebraic hyperbolic B-splines and wavelets. *Int. J. Wavelet Multiresolut. Inf. Process.* **2021**, *19*, 2150018. [CrossRef]

27. Marusic, M.; Rogina, M. Sharp error-bounds for interpolating splines in tension. *J. Comput. Appl. Math.* **1995**, *61*, 205–223. [CrossRef]
28. Delhez, E. A spline interpolation technique that preserve mass budget. *Appl. Math. Lett.* **2003**, *16*, 17–26. [CrossRef]
29. Boujraf, A.; Sbibih, D.; Tahrichi, M.; Tijini, A. A simple method for constructing integro spline quasi-interpolants. *Math. Comput. Simul.* **2015**, *15*, 36–47. [CrossRef]
30. Behforooz, H. Approximation by integro cubic splines. *Appl. Math. Comput.* **2006**, *175*, 8–15. [CrossRef]
31. Behforooz, H. Interpolation by integro quintic splines. *Appl. Math. Comput.* **2010**, *216*, 364–367. [CrossRef]
32. Zhanlav, T.; Mijiddorj, R. Integro cubic splines and their approximation properties. *Appl. Math. Ser. Tver State Univ. Russia* **2008**, *26*, 65. [CrossRef]
33. Lang, F.G.; Xu, X.P. On integro quartic spline interpolation. *J. Comput. Appl. Math.* **2012**, *236*, 4214. [CrossRef]
34. Eddargani, S.; Lamnii, A.; Lamnii, M. On algebraic trigonometric integro splines. *Z. Angew. Math. Mech.* **2020**, *100*, e201900262. [CrossRef]
35. Mijiddorj, R.; Zhanlav, T. Algorithm to construct integro splines. *ANZIAM J.* **2021**, *63*, 359–375. [CrossRef]
36. Zhanlav, T.; Mijiddorj, R. Integro cubic splines on non-uniform grids and their properties. *East Asian J. Appl. Math.* **2021**, *11*, 406–420.
37. Franke, R. Scattered data interpolation: Tests of some methods. *Math. Comp.* **1982**, *38*, 181–200.
38. Nielson, G.M. A first order blending method for triangles based upon cubic interpolation. *Int. J. Numer. Meth. Eng.* **1978**, *15*, 308–318.
39. Zhanlav, T.; Mijiddorj, R. The local integro cubic splines and their approximation properties. *Appl. Math. Comput.* **2010**, *216*, 2215–2219. [CrossRef]
40. Wu, J.; Zhang, X. Integro quadratic spline interpolation. *Appl. Math. Model.* **2015**, *39*, 2973–2980. [CrossRef]