Abstract

Numerous disciplines of the sciences, engineering, and economics use interpolation and approximation as specific and crucial numerical techniques to solve problems such as the reconstruction of shapes, the fabrication of mechanical parts, the visualization of human body organs, etc. However, when the form is complicated, it is difficult to determine the shape of the underlying function of the measurement data using a single polynomial. Therefore, employing a smooth piecewise function in this situation is preferable. The term "spline" refers to this particular function. Splines are among the most helpful interpolation and approximation functions due to their ease of construction and capacity to approximate complex forms.

We propose in this thesis a composite construction of the Hermite spline interpolant by adopting two techniques namely the minimization of slopes and the insertion of additional knots. To establish the first interpolant, we must first solve a Hermite interpolation problem in the space of a %5 cubic Algebraic Trigonometric (AT) spline. Next, the interpolant slopes are estimated by minimizing the first derivative oscillation. Then a %5 quadratic AT Hermite spline interpolation has also been considered by adding additional knots. This interpolant is constructed to reduce the computational cost of the %5 cubic Hermite AT spline and avoid solving any system of equations. Furthermore, this method allows the user to adjust the locations of the added knots to preserve the data's monotonicity.

Cubic Hermite splines interpolation in the space of Algebraic Hyperbolic (AH) functions has also been studied. This interpolant reproduces linear polynomials and hyperbolic functions. Then, the rise time of a second-order system is fitted as a function of the damping ratio using a %5 Hermite algebraic hyperbolic spline scheme.

Finally, we address the approximation with a %6 cubic Hermite spline interpolation scheme reproducing linear polynomials and hyperbolic functions. The interpolation scheme is mainly defined using integral values over the subintervals of a function to be approximated instead of the function and its first derivative values. As a result, the scheme provides an optimal convergence order.

Keywords: Algebraic trigonometric splines, Algebraic hyperbolic splines, Hermite interpolation, Monotonicity-preserving, Integro cubic interpolation.

1 f G ¶ R

Mohammed ORAICHE

Non-Polynomial Hermite Splines Interpolation

Année: 2021/2022
Formation doctorale : MIA

Initiale

Université Hassan 1er
Centre d'Études Doctorales en Sciences et Techniques & Sciences Médicales

Faculté des Sciences et Techniques Settat

THÈSE DE DOCTORAT

Pour l'obtention du grade de Docteur en Sciences et Techniques

Formation Doctorale : Mathématiques, Informatique et Application s

Spécialité : Mathématiques Appliquées

Sous le thème

Non-Polynomial Hermite Splines Interpolation

Présentée par :

Mohammed ORAICHE

Soutenue le : 22 Novembre 2022 à 10h à la Faculté des Sciences et Techniques de Settat devant le jury composé de :

Pr. Abdelkarim HAJJAJ	**PES**	**FST, Settat**	Président
Pr. Mohammed MESTARI	**PES**	**ENSET, Mohammedia**	Rapporteur
Pr. Mohamed Abdou ELOMARY	**PES**	**FST, Settat**	Rapporteur
Pr. Rachid EL JID	**PH**	**FST, Settat**	Rapporteur
Pr. Rachid FAKHAR	**PH**	**FP, Khouribga**	Examinateur
Pr. Abdellah LAMNII	**PES**	**ENS, Tetouan**	Co-Directeur
Pr. Mohamed LOUZAR	**PES**	**FST, Settat**	Directeur de thèse

Année Universitaire : 2021/2022

Preface

This thesis was carried out at the Department of Mathematics and Computer Science of the Faculty of Science and Technology of Settat, Laboratory of Mathematics, Computer Science and Engineering Sciences (MISI). Which has been made from December 2018 until November 2022. First of all, I would like to express my sincere gratitude to my supervisors, Pr.Abdellah Lamnii and Pr.Mohamed Louzar for their support, encouragement and valuable advice. Pr.Abdellah Lamnii has initiated me in spline approximation theory and he has carefully guided me through the academic world. Pr.Mohamed Louzar has supported me during my mathematics study at the Faculty of Sciences and Techniques of Settat. I am very grateful for the valuable time spent in sharing their expertise and knowledge with me. Finally, I would also like to say how much I appreciate their great availability and their professional and personal qualities.

– I would like to express my sincere thanks to all the members of the jury for their interest in this thesis:

- I thank with all my heart Pr.Rachid El jid, Pr.Mohamed Abdou Elomary and Pr.Mohammed Mestari for having accepted to report this thesis, for their interest in my work, and for their positive and encouraging remarks.
- It is with great pleasure that I thank Pr.Abdelkarim Hajjaj who did me the honor of presiding the jury of this thesis. I am also grateful to Pr.Rachid Fakhar for agreeing to be member of the jury. It is a great honor for me to share my work with them.

Also, I want to thank Salah Eddargani and Mhamed Madark for their kindness and for their valuable time and collaboration.

Finally, I would like to express my gratitude to my parents, my wife, my sisters and my brothers who supported me during my studies and especially to my father who loves science and scientists and who was waiting for the day when his son Mohammed would defend his PhD. I am happy that our dream has become a reality.

Abstract

Numerous disciplines of the sciences, engineering, and economics use interpolation and approximation as specific and crucial numerical techniques to solve problems such as the reconstruction of shapes, the fabrication of mechanical parts, the visualization of human body organs, etc. However, when the form is complicated, it is difficult to determine the shape of the underlying function of the measurement data using a single polynomial. Therefore, employing a smooth piecewise function in this situation is preferable. The term "spline" refers to this particular function. Splines are among the most helpful interpolation and approximation functions due to their ease of construction and capacity to approximate complex forms.

We propose in this thesis a composite construction of the Hermite spline interpolant by adopting two techniques namely the minimization of slopes and the insertion of additional knots. To establish the first interpolant, we must first solve a Hermite interpolation problem in the space of a $\mathcal{C}^1$ cubic Algebraic Trigonometric (AT) spline. Next, the interpolant slopes are estimated by minimizing the first derivative oscillation. Then a $\mathcal{C}^1$ quadratic AT Hermite spline interpolation has also been considered by adding additional knots. This interpolant is constructed to reduce the computational cost of the $\mathcal{C}^1$ cubic Hermite AT spline and avoid solving any system of equations. Furthermore, this method allows the user to adjust the locations of the added knots to preserve the data's monotonicity.

Cubic Hermite splines interpolation in the space of Algebraic Hyperbolic (AH) functions has also been studied. This interpolant reproduces linear polynomials and hyperbolic functions. Then, the rise time of a second-order system is fitted as a function of the damping ratio using a $\mathcal{C}^1$ Hermite algebraic hyperbolic spline scheme.

Finally, we address the approximation with a $\mathcal{C}^2$ cubic Hermite spline interpolation scheme reproducing linear polynomials and hyperbolic functions. The interpolation scheme is mainly defined using integral values over the subintervals of a function to be approximated instead of the function and its first derivative values. As a result, the scheme provides an optimal convergence order.

Keywords: Algebraic trigonometric splines, Algebraic hyperbolic splines, Hermite interpolation, Monotonicity-preserving, Integro cubic interpolation.

Contents

List of Figures

List of Tables

Symbols

$\mathbb{E}_n$	vector space of dimension n
$\mathbb{E}_n^*$	vector space of all linear forms defined on $\mathbb{E}_n$
$\mathbb{P}_n$	space of polynomials of degree n
$\mathcal{C}^m([a,b])$	space of functions with m continuous derivatives on the interval $[a,b]$
$\mathcal{L}_{n,j}(x)$	Lagrange's fundamental polynomials
Δ_n	partition of a bounded interval $[a,b]$
$\{x_i\}_{i=1}^n$	n knots
$\{f_i^0\}_{i=1}^n$	n function values
$\{f_i^1\}_{i=1}^n$	n first derivative function values
$\mathbb{S}_m(\Delta_n)$	linear space of all polynomial splines of degree $m-1$, with knot set Δ_n
$S_4^1(\Delta_n)$	space of cubic splines on Δ_n with global C^1 continuity
$\lvert A \rvert$	determinant of a square matrix A
Γ_4	space of algebraic and trigonometric functions
Γ_4'	space of algebraic and hyperbolic functions
$e_0(t)$	excitation of linear second order system
$y(t)$	response of linear second order system
$Y(z)$	Laplace transform function of $y(t)$
$E_0(z)$	Laplace transform function of $e_0(t)$
$H_t(z)$	the transfer function of a second order system
ξ	damping ratio
Tr	rise time
ω_n	natural frequency

General 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, fulfill all these requirements. Since the introduction of the systematic study of spline functions by I. J. Schoenberg in the 1940s [50], they have become an indispensable tool in approximation theory and numerical computation, including computer-aided geometric design (CAGD) [51, 52], 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 [34, 53]. 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, mainly because they admit normalized bases on any bounded interval $[a, b]$ (Bernstein bases and B-spline bases); for more details, see [53, 54, 55, 56]. Indeed, Bernstein bases and B-splines possess several interesting properties, such as non-negativity, local support, the partition of unity, and positive bases [57]. Moreover, Bernstein's 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 [58, 59]. The authors in [60] have established a recurrence relation for the trigonometric B-splines of arbitrary order. The complete construction of arbitrary order exponential tension B-splines was provided in [61]. An updated analysis of this type of spline can be found in [62, 63]. Hyperbolic splines are a common name for the splines connected to the exponential B-spline space [64].

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 [65], 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 [66, 67] 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 [68, 69, 70, 71, 72, 73, 74, 75, 76]. 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 [69] and references quoted. They yield the tension splines, which are extremely useful for avoiding undesirable oscillations in the interpolation curves [77].

Positivity, monotonicity or convexity are global properties that must be preserved when interpolating data coming from functions that present these characteristics, found when dealing with problems in very different fields of science and technology, as well as in Computer Aided Geometric Design.

A large number of methods have been proposed, among which should be mentioned the use of $\mathcal{C}^1$ quadratic splines [8, 13, 25, 35], $\mathcal{C}^1$ cubic splines [7], piecewise polynomial functions of varying degrees [9, 10], rational splines [12], splines in tension [47, 36], parametric splines [21, 26], parametric spline curves [14] and subsivision schemes [30] (see also [15, 16, 19] for a survey of spline-based methods until the early 90's).

In the last thirty years there has been an increase in the number of works on this topic, considering higher regularity or geometric regularity and more general spaces (see e.g. [37] and references therein).

In [4] a method of interpolation and smooth curve fitting is proposed. It is based on piecewise functions with slopes at the junction points locally determined under a geometrical condition. In [45] Hermite-type cubic splines are used in order to obtain a suboptimal algorithm in least squares data fitting problems. In [6], an optimal property for cubic interpolating splines of Hermite type is developed and applied to problems arising from data fitting area. Recently, in [17] an optimal cubic Hermite interpolation method is presented. It consists of the optimization of the derivatives at the knots defining the spline space.
In this thesis, the spaces $\Gamma = \{1, x, sinx, cosx\}$ and $\Gamma' = \{1, x, sinhx, coshx\}$ are used to construct two Hermite cubic spline interpolants, the first scheme is exact on Γ and the second on Γ'. The unknowns of the two interpolation schemes are obtained by a minimization technique that allows the constructed interpolants to preserve the monotonicity of the given data. In addition we define and study a many-knots $\mathcal{C}^1$ quadratic algebraic trigonometric spline interpolant.

Integro spline approximation was treated in various works in the literature. The author in [80, 81] developed two types of integro spline approximants, cubic and quintic cases, respectively. The two schemes introduced in [80, 81] 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 [82] developed cubic integro splines quasi-interpolant without solving any system of equations. An integro quartic spline scheme has been constructed in [83]. The authors in [69, 84] provided some integro spline schemes for the case of non-polynomial splines. More recent work on the integro spline approximation is given in [85, 86].

In this thesis, a new class of integro spline approximant is introduced. The proposed operator is $\mathcal{C}^2$ 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.

Outlined of the thesis

This thesis consists of four chapters. In the first one, we mention some theoretical results of general interpolation problems and deal with polynomial spline interpolation.

In Chapter 2, we begin by defining and studying an optimal Algebraic Trigonometric (AT) interpolant, which is exact on the linear space spanned by $\{1, x, sinx, cosx\}$, the construction of this interpolant is local, the values of the interpolation depend only on the function and its first derivative values at the data points, the function values at the knots are suppose known, and the first derivative values remain to be determined using a minimization approach. Then we define a $\mathcal{C}^1$ algebraic trigonometric (AT) spline that preserves the monotonicity of the data. This method allows the user to adjust the locations of the added knots.

In Chapter 3, A cubic Hermite Algebraic Hyperbolic (AH) interpolant that produces all functions in the space spanned by $\{1, x, sinhx, coshx\}$ is proposed. The unknown of the interpolant is obtained by minimizing the mean oscillation of the derivative. This interpolant has been applied to fit data from a second order system.

In Chapter 4, we propose a cubic Hermite spline interpolation scheme reproducing both linear polynomials and hyperbolic functions. The interpolation scheme is principally defined by means of integral values on the subintervals of a partition of the function to be approximated instead of the function and its first derivative values. The scheme provided is $\mathcal{C}^2$ everywhere and gives an optimal order.

Finally, we summarize our contributions, some perspectives, and possible future research directions.

Chapter 1

Preliminaries

Contents

This chapter recalls some theoretical results of general interpolation problems and some polynomial interpolation representations. Then we will discuss the spline interpolation. In addition, we will focus on the Hermite polynomial spline interpolation.

1.1 A General Interpolation Problem

Problem 1.1.1. *[2] Let $\mathbb{E}_n$ be a finite vector space of dimension n and let v_i $(i = 1, ..., n)$ be n given linear form defined on E_n, for a given set of data $\{c_i\}_{i=1}^n$. The problem is finding an element ϕ in $\mathbb{E}_n$ such that*

$$v_i(\phi) = c_i \quad i = 1, \dots, n \tag{1.1.1}$$

The vector space of all linear forms defined on $\mathbb{E}_n$ is denoted by $\mathbb{E}_n^*$.

Lemma 1.1.1. *[2] Let $\mathbb{E}_n$ be a vector space of dimension n. if the $\phi_i \quad (i = 1, ..., n)$ are independent in $\mathbb{E}_n$ and v_i $(i = 1, ..., n)$ are independent in $\mathbb{E}_n^*$ then*

$$D = \begin{vmatrix} v_1(\phi_1) & \dots & v_1(\phi_1) \\ & \vdots & \\ v_n(\phi_1) & \dots & v_n(\phi_n) \end{vmatrix} \neq 0 \tag{1.1.2}$$

where D is the determinant of the matrix $(v_i(\phi_j))$.
Inversely, if any of the set $\{v_i\}_{i=1}^n$ or $\{\phi_i\}_{i=1}^n$ are independent and (1.1.2) holds then the other set is also independent.

Proof 1.1.1. *Suppose that $D = 0$. Then the system of linear equations*

$$\begin{aligned} a_1v_1(\phi_1) + a_2v_2(\phi_1) + \cdots + a_nv_1(\phi_1) &= 0 \\ &\vdots \\ a_1v_1(\phi_n) + a_2v_2(\phi_n) + \cdots + a_nv_1(\phi_n) &= 0 \end{aligned}$$

has a nontrivial solution $a_1, a_2, \dots, a_n$.
So,

$$(a_1v_1 + a_2v_2 + \cdots + a_nv_1)(\phi_i) = 0, \quad i = 1, 2, \dots, n$$

and

$$(a_1v_1 + a_2v_2 + \cdots + a_nv_1)(\phi) = 0, \quad \forall \phi \in \mathbb{E}_n$$

because $\{\phi_i\}_{i=1}^n$ form a basis of $\mathbb{E}_n$.
Therefore

$$a_1v_1 + a_2v_2 + \cdots + a_nv_1 = 0$$

hence $\{v_i\}_{i=1}^n$ are dependent .
For the converse, we can use the same logic.

Theorem 1.1.1. *[2] Let $\mathbb{E}_n$ be a vector space of dimension n and v_i $(i = 1, \dots, n)$ be n elements of $\mathbb{E}_n^*$. The interpolation problem (1.1.1) possesses a unique solution for arbitrary values c_i $(i = 1, ..., n)$ if and only if the v_i $(i = 1, ..., n)$ are independent in $\mathbb{E}_n^*$.*

Proof 1.1.2. *Suppose that the $\{v_i\}_{i=1}^n$ are independent in $\mathbb{E}_n^*$ and $\{\phi_i\}_{i=1}^n$ are independent in $\mathbb{E}_n$ then by Lemma 1.1.1 the determinant $\begin{vmatrix} v_1(\phi_1) & \cdots & v_1(\phi_1) \\ & \vdots & \\ v_n(\phi_1) & \cdots & v_n(\phi_n) \end{vmatrix} \neq 0$ therefore the system*

$$v_i(a_1\phi_1 + a_2\phi_2 + \cdots + a_n\phi_n) = c_i \quad i = 1, \dots, n$$

i.e.

$$a_1v_i(\phi_1) + a_2v_i(\phi_2) + \cdots + a_nv_i(\phi_n) = c_i \quad i = 1, \dots, n \tag{1.1.3}$$

has a solution $a_1, a_2, \ldots, a_n$ *and* $\phi = \sum_{i=1}^{n} a_i\phi_i$ *is a solution of the interpolation problem 1.1.1*
Inversely, if the problem (1.1.1) *has a solution for arbitrary values* c_i $(i = 1, \ldots, n)$ *then the system*

$$a_1 v_i(\phi_1) + a_2 v_i(\phi_2) + \cdots + a_n v_i(\phi_n) = c_i \quad i = 1, \ldots, n$$

has a solution for arbitrary values c_i $(i = 1, \ldots, n)$ *this implies that*

$$\begin{vmatrix} v_1(\phi_1) & \cdots & v_1(\phi_1) \\ & \vdots & \\ v_n(\phi_1) & \cdots & v_n(\phi_n) \end{vmatrix} \neq 0.$$

Therefore the $\{v_i\}_{i=1}^n$ *are independent.*

Several function spaces and related systems of independent functional are known and have been studied in detail. In this chapter, we will mention some of the most popular of them .

Example 1.1.1. *Taylor interpolation*

$$\mathbb{E}_n = \mathbb{P}_n, \quad v_i(p) = p^{(i)}(x_0) \quad i = 0, \ldots, n.$$

This module shows the Taylor polynomial interpolation. The Taylor polynomial interpolant of degree n for a smooth function $f(x)$ around a point x_0 is the unique polynomial $p(x)$ of degree n whose value and those of its n first derivatives at x_0 agree with those of f, i.e., $p^{(i)}(x_0) = f^{(i)}(x_0)$ for $i = 0, 1, \ldots, n$.

Example 1.1.2. (*Interpolation at discrete points*)
Let $\{x_i\}_{i=0}^n$ be a distinct real numbers .

$$\mathbb{E}_n = \mathbb{P}_n, \quad v_i(p) = p(x_i) \quad i = 0, \ldots, n.$$

This example shows the standard polynomial interpolation at discrete points . The polynomial interpolant of degree n for a data $\{f_i\}_{i=0}^n$ at the knots $\{x_i\}_{i=0}^n$ it is the unique polynomial $p(x)$ of degree n whose value at x_i is equal to f_i, i.e.

$$p(x_i) = f_i, \quad i = 0, 1, \ldots, n.$$

Example 1.1.3. (*Hermite or Osculatory Interpolation*)
Let $\{x_i\}_{i=0}^n$ be a distinct real numbers .

$$\mathbb{E}_n = \mathbb{P}_{2n+1}, \quad v_i^j(p) = p^{(j)}(x_i) \quad i = 0, \ldots, n \quad \textit{and } j = 0, 1.$$

The problem here is to finding a polynomial $p(x)$ in $\mathbb{P}_{2n+1}$ which passes through given points with given slopes. Specifically, let consider the given data $\{x_i, f_i, f_i'\}_{i=0}^n$.

Find a polynomial $p(x)$ in $\mathbb{P}_{2n+1}$ such that

$$p^{(j)}(x_i) = f_i^{(j)}, \quad i = 0, \ldots, n \text{ and } j = 0, 1.$$

Example 1.1.4. (*General Hermite Interpolation*)
Let $\{x_i\}_{i=0}^{n}$ be a distinct real numbers.

$$\mathbb{E}_n = \mathbb{P}_N, \quad v_i^j(p) = p^{(j)}(x_i) \quad i = 0, \ldots, n \quad \text{and } j = 0, 1, \ldots, r_i.$$

Where $N = n + \sum_{i=0}^{n} r_i$.
A general Hermite polynomial interpolation consists of determining a polynomial $p(x)$ in $\mathbb{P}_N$ such that

$$p_i^j(x_i) = f_i^{(j)} \quad i = 0, \ldots, n \quad \text{and } j = 0, 1, \ldots, r_i.$$

Where f_i^j *and* r_i, $i = 0, \ldots, n$; $j = 0, 1, \ldots, r_i$, are given.

Before proving that these functionals are independent on the corresponding space, a remark is in order. Examples 1.1.1, 1.1.2 and 1.1.3 are a special cases of Example 1.1.4.
Therefore it suffices to demonstrate the Example 1.1.4, to this end we need the following theorem.

Theorem 1.1.2. *Consider a square matrix $A = (a_{i,j})$, $i = 1, ..., n$ and $j = 1, ..., n$. The homogeneous system*

$$\sum_{i=1}^{n} a_{i,j} x_j = 0, \quad for\ i = 1, ..., n$$

has a non trivial solution (i.e a solution other than $x_1 = x_2 = ... = x_n = 0$) if and only if the determinant of the matrix A is equal to zero i.e. $|A| = 0$.

Proof 1.1.3. *Example 1.1.4 (General Hermite Interpolation).*
Let $p \in \mathbb{P}_N$, $N = n + \sum_{i=0}^{n} r_i$ and consider the homogeneous system

$$v_i^j(p) = p^{(j)}(x_i) = 0, \quad for \quad i = 0, \ldots, n \text{ and } j = 0, 1, \ldots, r_i. \tag{1.1.4}$$

Using the Factorisation Theorem, if all conditions (1.1.4) are satisfies except the last one $(p^{(r_n)}(x_n) = 0)$, then

$$p(x) = q(x)(x - x_0)^{r_0+1}(x - x_1)^{r_1+1} \cdots (x - x_n)^{r_n}.$$

Where q a polynomial, since the degree of p is N then $q(x)$ must have degree 0.

Furthermore

$$p^{(r_n)}(x_n) = q(r_n)!(x_n - x_0)^{r_0+1}(x_n - x_1)^{r_1+1} \cdots (x_n - x_{n-1})^{r_{n-1}+1} = 0$$

and $x_i \neq x_j$ for $i \neq j$, this implies that $q = 0$, consequently p is a zero polynomial. The homogeneous interpolation problem has only the zero solution and therefore according to Theorem 1.1.2 the non-homogeneous problem has a unique solution.

1.2 Polynomial Interpolation Representation

In this section we will present some of the most popular polynomial interpolation formula. Consider the data $\{(x_i, f_i)\}_{i=0}^n$, we are interested in this case by the discrete interpolation and we suppose that the interpolation points $\{x_i\}_{i=0}^n$ are all distinct, therefore the procedure of finding a polynomial that passes through the points $\{(x_i, f_i)\}_{i=0}^n$ is equivalent to solving a system of linear equations $Ax = b$ which has a unique solution. However, different methods for finding the interpolating polynomial use a different A, since they each use a different basis for the space $\mathbb{P}_n$ of polynomials of degree $\leq n$. The simplest way to calculate the interpolation polynomial is to form the system $Ax = b$ where $b_i = f_i,\ i = 0, \ldots, n$ and the entries of A are given by $a_{i,j} = P_j(x_i),\ i = 0, \ldots, n$, where $P_j(x) = x^j,\ j = 0, \ldots, n$. The basis $\left\{1, x, x^2, \ldots, x^n\right\}$ of the space $\mathbb{P}_n$ is called the *monomial basis* and the associated matrix A is called *Vandermonde matrix*. Unfortunately, this matrix can be ill-conditioned. The *Lagrange interpolation* overcome this problem.

1.2.1 Lagrange Interpolation Formula

In *Lagrange interpolation*, the matrix A is just the identity matrix, because the interpolation polynomial is written as

$$P(x) = \sum_{j=0}^{n} f_j \mathcal{L}_{n,j}(x). \tag{1.2.1}$$

With $P(x)$ is a polynomial of degree n and satisfies

$$P(x_i) = f_i \quad i = 0, \ldots, n \tag{1.2.2}$$

and

$$\mathcal{L}_{n,j}(x) = \prod_{k=0, k\neq j}^{n} \frac{x - x_k}{x_j - x_k}. \tag{1.2.3}$$

It easy to see that

$$\mathcal{L}_{n,j}(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & i \neq j \end{cases}. \tag{1.2.4}$$

The polynomials $\{\mathcal{L}_{n,j}\}_{j=0}^{n}$, are called the *fundamental polynomials* for the interpolation points $\{x_i\}_{i=0}^{n}$ and they constitute a basis of $\mathbb{P}_n$.
The formula (1.2.1) is called *Lagrange Interpolation Formula.* Notice that, since the interpolation problem (1.2.2) has a unique solution, all other representations of the solution must, upon rearrangement of terms, coincide with the Lagrange polynomial.

Theorem 1.2.1. (Lagrange Interpolation Error) *Let $f \in \mathcal{C}^{n+1}([a,b])$ and $a := x_0 < x_1 < \cdots < x_n =: b$.*
Let $P \in \mathbb{P}_n$ the Lagrange polynomial defined by $P(x_i) = f(x_i) \quad i=0,\ldots,n$.
Then

$$|P(x) - f(x)| \leq \frac{1}{(n+1)!} \max_{x\in[a,b]} |f^{(n+1)}(\bar{x})| \max_{x\in[a,b]} |\Pi_{n+1}(x)| \tag{1.2.5}$$

where $\Pi_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$ *and* $a < \bar{x} < b$.

Proof 1.2.1. *We assume that x is not one of the interpolation points $x_0, x_1, \ldots, x_n$, and we define*

$$\phi(t) = P(t) - f(t) - \frac{P(x) - f(x)}{\Pi_{n+1}(x)}\Pi_{n+1}(t).$$

Since x is not one of the interpolation point, then $\phi(t)$ has at least $n+2$ zeros (x and $x_0, x_1, \ldots, x_n$). In addition, $\Pi_{n+1}(x) \neq 0$, so $\Pi_{n+1}(x)$ is well-defined.

Because $f \in \mathcal{C}^{n+1}([a,b])$ and $x \in [a,b]$, it follow, by the Generalised Rolle's Theorem, ϕ^{n+1} must have at least one zero in $[a,b]$.
Therefore, at some point $\bar{x} \in [a,b]$, that depends on x, we have

$$0 = \phi^{n+1}(\bar{x}) = f^{n+1}(t) - \frac{P(x) - f(x)}{\Pi_{n+1}(x)}(n+1)!,$$

which concludes the proof.

The Lagrange form is certainly quite elegant, but compared to other approaches to writing and evaluating the interpolation polynomial, it is far from the most efficient. The following section introduce an extension of Lagrange interpolation, which consists, for a given derivable function and a given finite number of points, in constructing a polynomial which is both interpolator (i.e. whose values at the given points coincide with those of the function) and osculator (i.e. whose values of the derivative at the given points coincide with those of the derivative of the function), we are talking about the Hermite interpolation.

1.2.2 Hermite Interpolation

The Hermite interpolation problem consists to find a polynomial P_n that interpolates the function f and its derivative f'. More precisely, let $\{(x_i, f_i^0, f_i^1)\}_{i=0}^{n}$, with

$f_i^0 = f(x_i)$ and $f_i^1 = f'(x_i)$ be the given data. We search a polynomial P such that

$$P^{(j)}(x_i) = f_i^j \quad i = 0, \ldots, n \textit{ and } j = 0, 1. \tag{1.2.6}$$

Theorem 1.2.2. *There is exactly one $P \in \mathbb{P}_{2n+1}$ satisfies Hermite interpolation conditions (1.2.6).*

Proof 1.2.2. *We begin by establish the existence of a basis polynomial of $\mathbb{P}_{2n+1}$ such that*

$$P(x) = \sum_{i=0}^{n} f_i^0 A_i(x) + \sum_{i=0}^{n} f_i^1 B_i(x),$$

the polynomial P must satisfy the conditions (1.2.6) which allows to impose the following conditions on the polynomials A_i and B_i

$$A_i(x_j) = \delta_{i,j},\ B_i(x_j) = 0 \quad i = 0, \ldots, n,$$

$$A_i'(x_j) = 0,\ B_i'(x_j) = \delta_{i,j} \quad i = 0, \ldots, n,$$

this conditions allows us to construct the functions A_i and B_i .
We have $A_i(x_j) = A_i'(x_j) = 0$ for $i \neq j$ then $(x - x_j)^2$ divide $A_i(x)$ for $i \neq j$, so, we can write $A_i(x)$ in the form

$$A_i(x) = q(x)\mathcal{L}_{n,i}^2(x)$$

where $q(x) = ax + b$ and $\mathcal{L}_{n,i}(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$.
Using the equations $A_i(x_i) = 1$ and $A_i'(x_j) = 0$ we find that $a = -2\mathcal{L}_{n,i}'(x_i)$ and $b = 1 - ax_i$.
Finally

$$A_i(x) = (1 - 2(x - x_i)\mathcal{L}_{n,i}'(x_i))\mathcal{L}_{n,i}^2(x).$$

A calculation similar to the previous one allows us to establish that

$$B_i(x) = (x - x_i)\mathcal{L}_{n,i}^2(x).$$

For the uniqueness let $p, q \in \mathbb{P}_{2n+1}$ such that

$$p(x_i) = q(x_i) = f_i^0 \textit{ and } p'(x_i) = q'(x_i) = f_i^1 \textit{ for } i = 0, \ldots, n,$$

then

$$r = p - q \in \mathbb{P}_{2n+1} \textit{ and } r(x_i) = r'(x_i) = 0,$$

then

$$(x - x_i)^2 \text{ divide } r(x) \text{ and } r(x) = a(x - x_0)(x - x_1)\cdots(x - x_n) \in \mathbb{P}_{2(n+1)}$$

,
since

$$r \in \mathbb{P}_{2n+1},$$

therefore

$$a = 0 \text{ and } r = 0.$$

Theorem 1.2.3. *Let $f \in \mathcal{C}^{2n+2}([a, b])$ and $a = x_0 < x_1 < \cdots < x_n = b$. Let $P \in \mathbb{P}_{2n+1}$ be the Hermite polynomial defined by (1.2.6).*
Then

$$|P(x) - f(x)| \leq \frac{1}{(2n+2)!} \max_{x\in[a,b]} |f^{(2n+2)}(\bar{x})| \max_{x\in[a,b]} |\Pi(x)| \tag{1.2.7}$$

where $\Pi(x) = \displaystyle\prod_{j=0}^{n}(x - x_j)^2$ *and* $a < \bar{x} < b$.

The interpolation error is composed of two terms, one depending on the function f which cannot be improved and the other on the distribution of the knots $\{x_i\}_{i=0}^{n}$. If the number of data points is large, then polynomial interpolation becomes problematic since high-degree interpolation yields oscillatory polynomials, when the data may fit a smooth function.

1.2.3 The Runge Phenomenon

Problems can arise with high degree polynomial interpolants, especially in the neighborhood of singularities of the function f, as illustrated by this classic example of Runge.
Let's consider the function $f_2(x) = \dfrac{1}{1 + x^2}$, we use the Lagrange interpolation to interpolate this function at n equidistant points on the interval $[-5, 5]$. Then we progressively increase the degree n of interpolation.
Figures 1.1, 1.1 and 1.3 illustrate the Runge phenomenon.

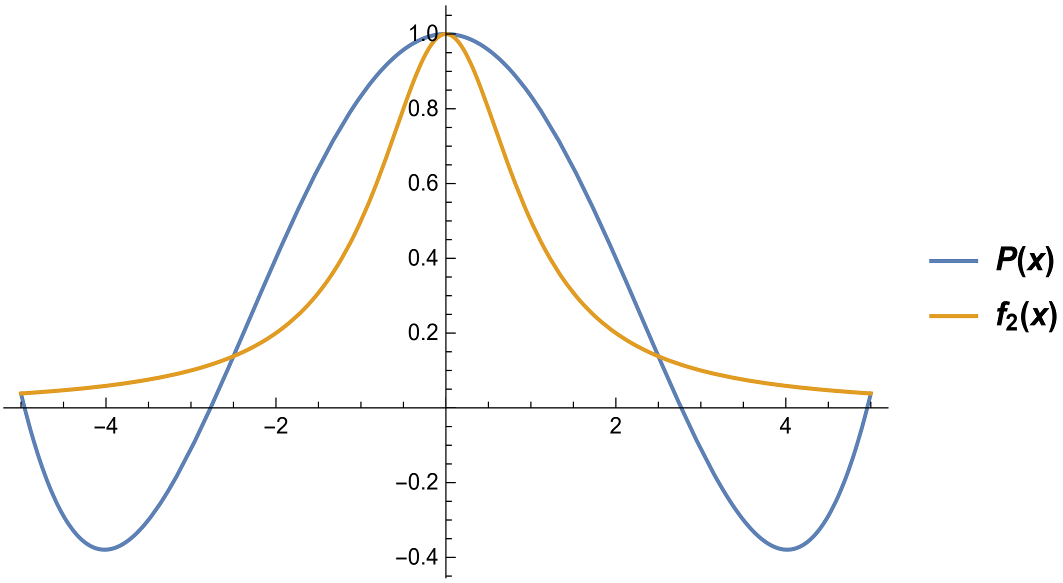


Figure 1.1: Plot of $f_2(x) = \dfrac{1}{1+x^2}$ and its polynomial interpolant through $n = 5$ equally spaced points.

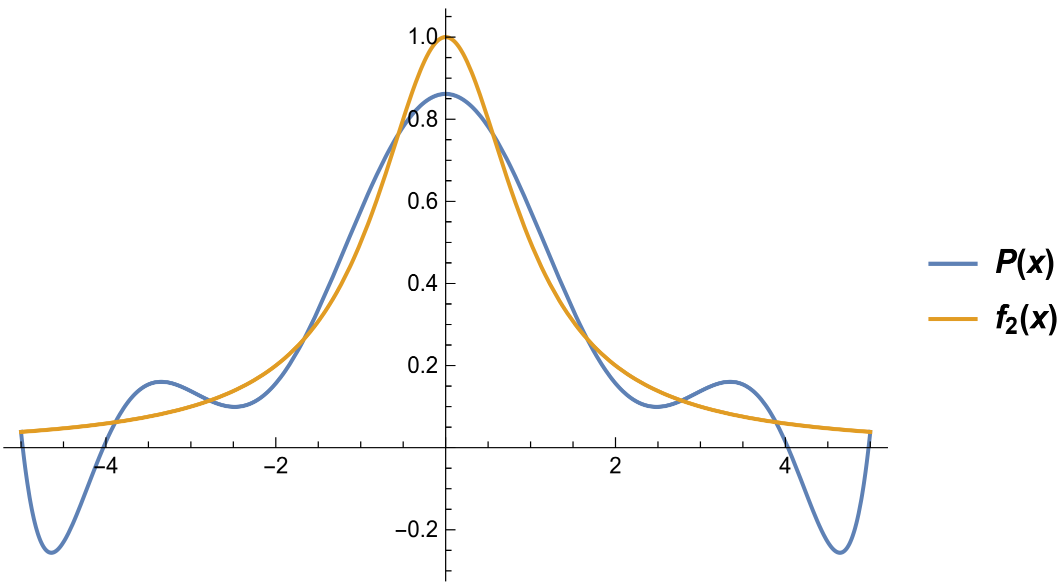


Figure 1.2: Plot of $f_2(x) = \dfrac{1}{1+x^2}$ and its polynomial interpolant through $n = 10$ equally spaced points.

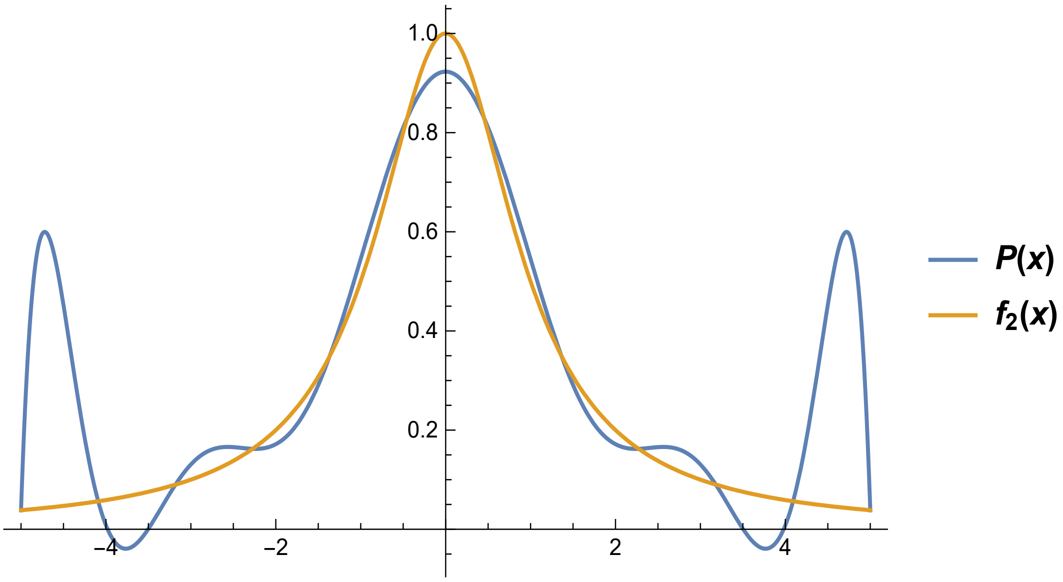


Figure 1.3: Plot of $f_2(x) = \dfrac{1}{1+x^2}$ and its polynomial interpolant through $n = 12$ equally spaced points.

Note the oscillations of the interpolant, which renders it practically useless for interpolation. The polynomial interpolant is very sensitive to the location of the data points $\{x_i\}_{i=0}^{n}$, this highlights the essential weakness of polynomial interpolation: if the function to be approximated behaves badly anywhere in the approximation interval, then the approximation is poor everywhere. this global dependence of local properties can be overcome by using piecewise polynomial interpolation, but the piecewise polynomials can even be discontinuous. Because in most real-world applications, users would like the approximation function to be sufficiently smooth and in order to preserve the flexibility of piecewise polynomials while at the same time at the same time achieve a certain degree of global smoothness, we use a class of functions called *Splines*.

1.3 Spline Interpolation

1.3.1 Polynomial Splines

Splines are an important class of mathematical functions used for approximation. A spline is a piecewise polynomial function that is commonly described as being "as smooth as it can be without reducing to a polynomial" [1].

Definition 1.3.1. Let $n \in \mathbb{N}$ and let $a := x_0 < x_1 < \cdots < x_n =: b$ be points in the interval $I = [a, b]$. The set of points $\Delta_n = \{x_i\}_{i=0}^{n}$ is called a knots set or a partition of I . The knots $x_1, \ldots, x_{n-1}$ are named interior knots and the knots x_0 and x_n boundary knots.

Definition 1.3.2. (Spline and Spline space). Let $m \in \mathbb{N}$, $I = [a, b]$ a bounded interval of $\mathbb{R}$ and $\Delta_n = \{x_i\}_{i=0}^{n}$ a partition of I.
A function $S : [a, b] \longrightarrow \mathbb{R}$ is called (polynomial) spline of degree $m-1$ or of order m if it satisfies the following two conditions.

1. $S \in \mathcal{C}^{m-2}([a, b])$.
2. $\forall i \in \{0, 1, \ldots, n-1\}$: $S_i = S \restriction_{[x_i, x_{i+1}[} \in \mathbb{P}_{m-1}[a, b]$.

The linear space of all polynomial splines of degree $< m$, with knot set Δ_n is denoted by $\mathbb{S}_m(\Delta_n)$.

Reformulation of conditions 1 and 2 above yields an equivalent definition: Let $\mathbb{P}_{m-1}(\Delta_n)$ be the linear space of all functions whose restriction to each interval $]x_i, x_{i+1}[$, $i \in \{0, \ldots, n-1\}$ is an element of $\mathbb{P}_{m-1}[a, b]$. Then

$$\mathbb{S}_m(\Delta_n) = \mathbb{P}_{m-1}(\Delta_n) \cap \mathcal{C}^{m-2}([a, b]). \tag{1.3.1}$$

Remark 1. Each polynomial $P \in \mathbb{P}_{m-1}$ is automatically a spline of order m for any knot set Δ_n.

A Basis for $\mathbb{S}_m(\Delta_n)$.
Define the *truncated power function* by

$$(x-c)_+^n := \begin{cases} 0 & for\ x < c \quad , \\ (x-c)^n & for\ x \geq c \quad , \end{cases} \tag{1.3.2}$$

$$\begin{aligned} &q_{i,m} : [a, b] \longrightarrow \mathbb{R} \\ &q_{i,m}(x) = (x - x_i)_+^{m-1}, \ i \in \{0, \ldots, n-1\}. \end{aligned} \tag{1.3.3}$$

Where

$$x_+ := max\{x, 0\} \ \ and \ \ x_+^m := (x_+)^m, \ \forall m \in \mathbb{N}, \tag{1.3.4}$$

the function $q_{i,m}$ is called *truncated power function*

Theorem 1.3.1. *The set* $\mathcal{B} := \{1, x, x^2, \ldots, x^{m-1}\} \cup \{q_{i,m}\ /1 \ \leq i \leq n-1\}$ *constitutes a basis for the spline space* $\mathbb{S}_m(\Delta_n)$ *for any knot set* Δ_n *the dimension of* $\mathbb{S}_m(\Delta_n)$ *is therefore dim* $\mathbb{S}_m(\Delta_n) = n + m - 1$.
That is to say, every spline function $S \in \mathbb{S}_m(\Delta_n)$ *possesses a unique representation*

$$S(x) = \sum_{k=0}^{m-1} a_k x^k + \sum_{i=1}^{n-1} b_i (x - x_i)_+^{m-1}. \tag{1.3.5}$$

Proof 1.3.1. *Let* P_i^{m-1} *be a polynomial of degree* $d-1$ *or less, which by definition represents* $S(x)$ *over the interval* $[x_i, x_{i+1}]$ *and* P_{i-1}^{m-1} *is the corresponding polynomial*

over $[x_{i-1}, x_i]$.
Since the continuity of $S(x)$, x_i *is a root of* $r(x) = P_i^{m-1}(x) - P_{i-1}^{m-1}(x)$.
The same is true for the $m-2$ *continuous derivatives of the spline function at this point.*
From this result

$$r(x) = b_i(x - x_i)^{m-1}.$$

Consequently, for every k *between* 1 *and* $n-1$

$$P_k^{m-1}(x) - P_0^{m-1}(x) = [P_k^{m-1}(x) - P_{k-1}^{m-1}(x)] - [P_{k-1}^{m-1}(x) - P_{k-2}^{m-1}(x)] - \cdots - [P_1^{m-1}(x) - P_0^{m-1}(x)]$$

i.e.

$$P_k^{m-1}(x) = P_0^{m-1}(x) + b_1(x - x_1)^{m-1} + b_2(x - x_2)^{m-1} + ... + b_k(x - x_k)^{m-1}.$$

Therefore

$$S(x) = P_0^{m-1}(x) + \sum_{i=1}^{n-1} b_i(x - x_i)_+^{m-1}$$

is true because the difference in the representing polynomials of two neighbouring intervals always takes the form $b_i(x - x_i)^{m-1}$.
To demonstrate the uniqueness of this representation, let $x < x_1$,
then

$$S(x) = P_0^{m-1}(x) = P^{m-1}(x)$$

is unique, as polynomial of degree $(m-1)$ *already correspond when they are identical at* m *points.*
Furthermore it can be determined that

$$\lim_{x \to x_j^-} S^{(m-1)}(x) = P_{m-1}^{(m-1)}(x_j) + n! \sum_{i=1}^{j-1} b_i \text{ and } \lim_{x \to x_j^+} S^{(m-1)}(x) = P_{m-1}^{(m-1)}(x_j) + n! \sum_{i=1}^{j} b_i$$

Whence

$$b_j = \frac{1}{n!}(\lim_{x \to x_j^+} S^{(m-1)}(x) - \lim_{x \to x_j^-} S^{(m-1)}(x)), \quad j = 1, 2, \ldots, n-1.$$

1.3.2 Uniqueness of Spline Interpolation

Theorem 1.3.2. *Let* $\mathbb{S}_m$ *be a given spline space of dimension* N, $\mathcal{B}_i^m(x)$, $i = 1, 2, \ldots, N$ *basis of* $\mathbb{S}_m$, *and data* $\{f_i^0\}_{i=1}^N$ *given at* N *real numbers* $x_1 < x_2 < \cdots < x_N$.

Then there exists a unique spline $S(x) = \sum_{i=1}^{N} c_i \mathcal{B}_i^m(x)$ *in* $\mathbb{S}^m$ *satisfying*

$$S(x_i) = f_i^0, \ i = 1, \ldots, N, \tag{1.3.6}$$

if and only if the collocation matrix $A = (\mathcal{B}_j^m(x_i))_{i,j=1}^N$ *is nonsingular.*

Theorem 1.3.3. (Schoenberg-Whitney).
Let $\mathbb{S}_m$ *be a given spline space, and let* $x_1 < x_2 < \cdots < x_N$ *be* N *distinct numbers.* The collocation matrix $A = (\mathcal{B}_j^m(x_i))_{i,j=1}^N$ *is nonsingular if and only if*

$$\mathcal{B}_i^m(x_i) > 0, \ i = 1, \ldots, N. \tag{1.3.7}$$

Proof 1.3.2. *For a detailed demonstration of Theorems 1.3.2 and 1.3.3, please see reference [53]*

Splines functions have many interesting features, including :

- Spline spaces are finite dimensional linear spaces with very convenient bases.
- Splines are relatively smooth functions.
- Splines are easy to store, manipulate, and evaluate on a digital computer.

The cubic is the optimal spline order generally used in practice, an important application of cubic spline interpolation is called computer-aided geometric design.

1.3.3 Cubic Spline Interpolation

The basic idea of cubic spline interpolation is based on the engineering tool used to trace smooth curves through a series of points. This spline is composed of weights fixed on a flat surface at the points to be joined. A flexible strip is then folded over each of these weights, resulting in a nicely smooth curve. The principle of the mathematical spline is similar. The points, in this case, are numerical data. The weights are the coefficients of the cubic polynomials used to interpolate the data. These coefficients "bend" the line so that it passes through each of the data points without erratic behavior or breaks in continuity.

Construction

Let $d \in \mathbb{N}$, $I = [a, b]$ a bounded interval of $\mathbb{R}$ and $\Delta_n = \{x_i\}_{i=0}^n$ a partition of I. Consider the data points $(x_i, f_i)_{i=0}^n$. The idea is to fit a polynomial spline function

S of the form

$$S(x) = \begin{cases} s_0(x) & if \quad x_0 \leq x < x_1, \\ s_1(x) & if \quad x_1 \leq x < x_2, \\ \vdots & \qquad \vdots \\ s_{n-1}(x) & if \quad x_{n-1} \leq x \leq x_n, \end{cases} \tag{1.3.8}$$

where s_i is a third degree polynomial defined as

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \; for \; i = 0, \ldots, n-1. \tag{1.3.9}$$

To determine S we have to determine $4n$ parameters

$$a_i, b_i, c_i, d_i, \quad i = 0, 1, \ldots, n-1.$$

The cubic spline S satisfies the following proprieties:

1. $S(x) = s_i(x), \quad x \in [x_i, x_{i+1}]$ for $i = 0, 1, \ldots, n-1$.
2. $S(x_i) = f_i$, for $i = 0, 1, \ldots, n$.
3. $s_i(x_{i+1}) = s_{i+1}(x_{i+1})$, for $i = 0, 1, \ldots, n-2$.
4. $s_i'(x_{i+1}) = s_{i+1}'(x_{i+1})$, for $i = 0, 1, \ldots, n-2$.
5. $s_i''(x_{i+1}) = s_{i+1}''(x_{i+1})$, for $i = 0, 1, \ldots, n-2$.

The above properties 2, 3, 4, and 5 of the classical cubic spline impose $4n - 2$ conditions on S.
Consequently, we require two additional conditions on S to uniquely determine the parameters.
There are different ways to define the other conditions, the most popular are the below:

1. $S''(x_0) = S''(x_n) = 0$ (natural boundary).
2. $S'(x_0) = f'(x_0), \; S'(x_n) = f'(x_n)$ (clamped boundary).

A cubic spline S satisfies the additional condition 1 is called a **natural cubic spline**.
A cubic spline S satisfies the additional condition 2 is called a **clapmed or complete cubic spline**.

Theorem 1.3.4. *[53] (**Convergence of Clamped Cubic Splines**)*
Let $f \in C^4([a, b])$ and $h_{max} = \max_{i=0,1,\ldots,n-1} h_i$, where $h_i = x_{i+1} - x_i$.
If the cubic spline S satisfying the clamped boundary conditions 2, then there exist constant C_r such that

$$\max_{x\in[a,b]} |f^{(r)}(x) - S^{(r)}(x)| \leq C_r h_{max}^{4-r} \max_{x\in[a,b]} |f^{(4)}(x)|, \quad r = 0, 1, 2.$$

1.3.3.1 Numerical Examples

In this part we present some numerical examples to illustrate the above theoretical results.
Consider the following functions

$$f_0(x) = sin(\pi x) + cos(x),$$

$$f_1(x) = \sqrt{9 - (x-3)^2}$$

and

$$f_2(x) = \frac{1}{1+x^2}$$

We gives the interpolation curves of the above functions using Natural Cubic Spline and Clamped Cubic Spline. Figures 1.4, 1.5, 1.6, 1.7, 1.8 and 1.9 shows the interpolation curves.

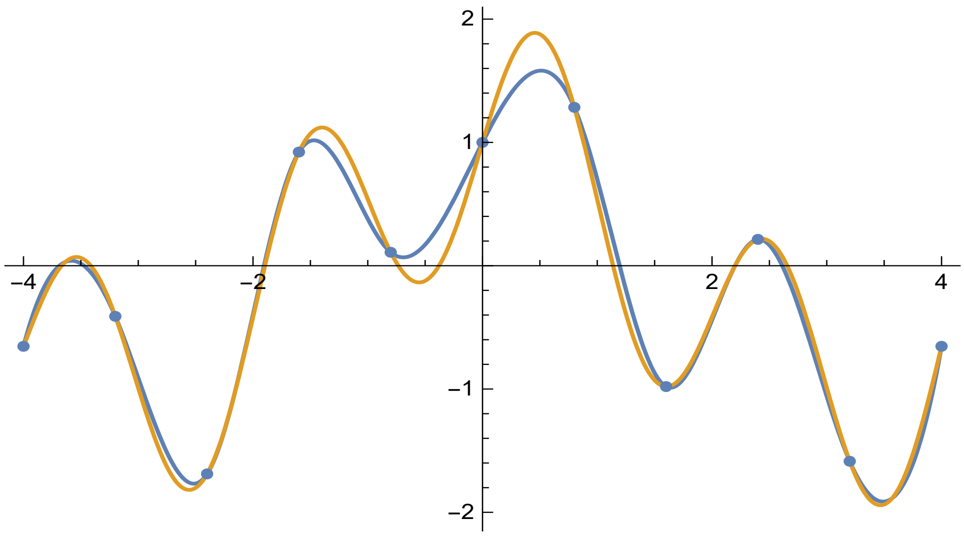
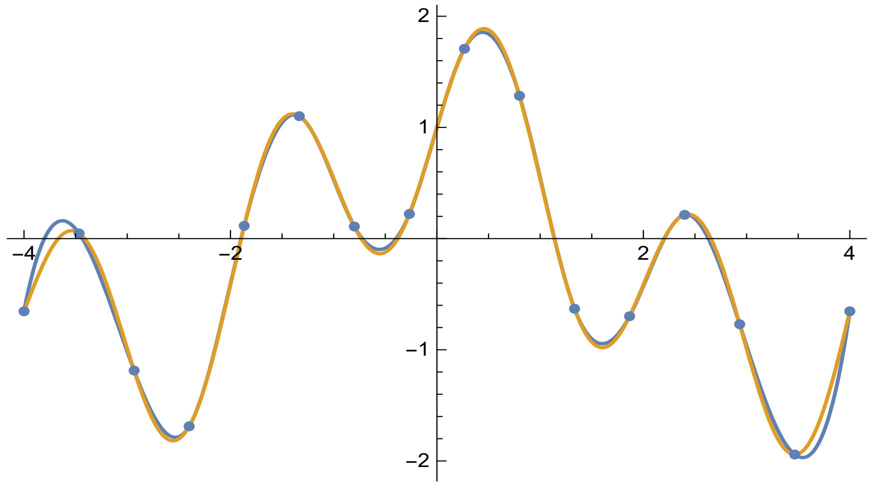
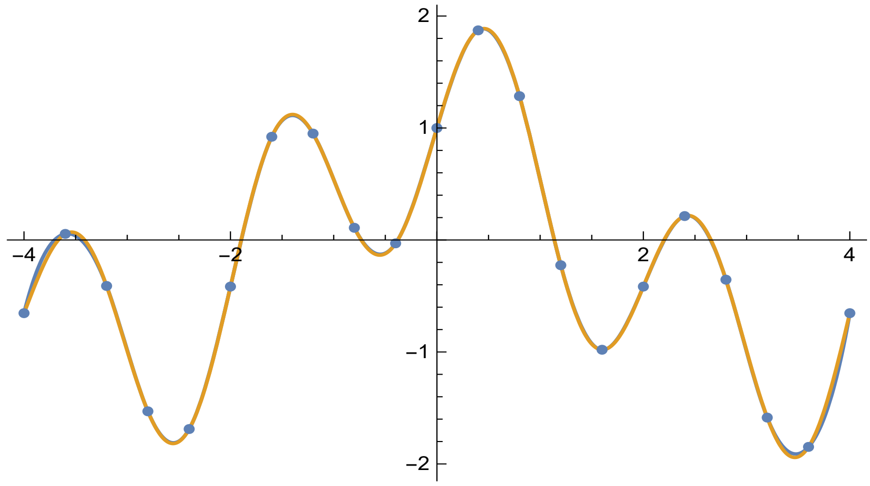


Figure 1.4: A Clamped Cubic Spline Approximation of f_0, $n = 10$



(a) f_0, $n = 15$



(b) f_0, $n = 20$

Figure 1.5: A Clamped Cubic Spline Approximation of f_0.

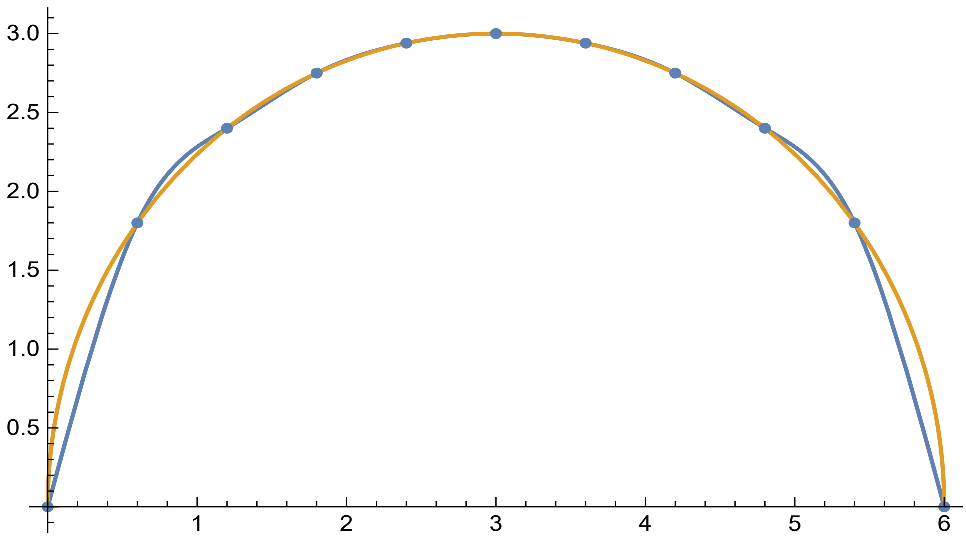
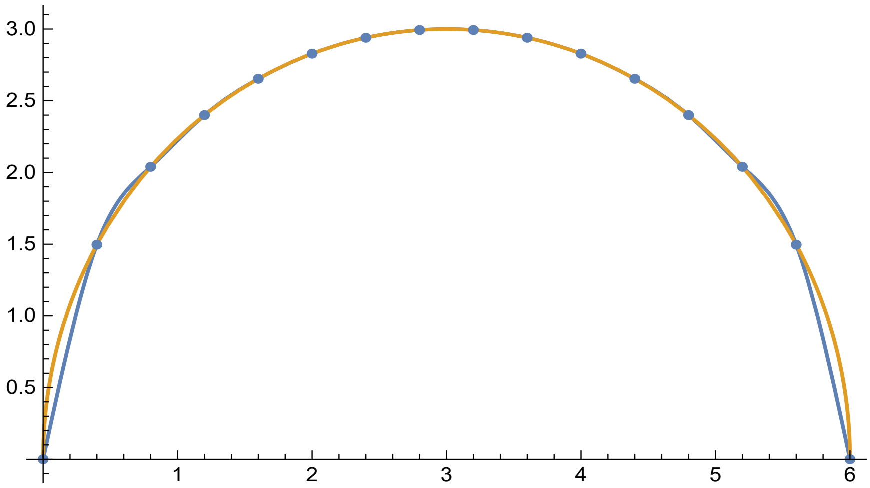
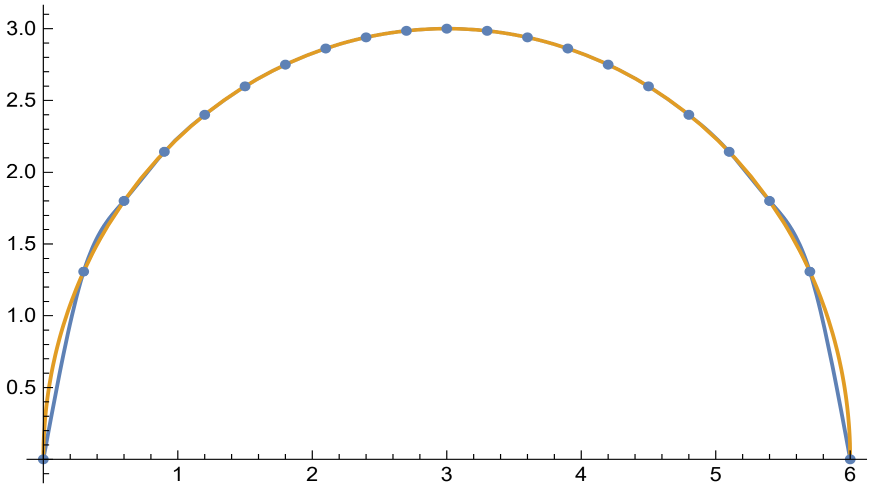


Figure 1.6: A Natural Cubic Spline Approximation of f_1, $n = 10$



(a) f_1, $n = 15$



(b) f_1, $n = 20$

Figure 1.7: A Natural Cubic Spline Approximation of f_1.

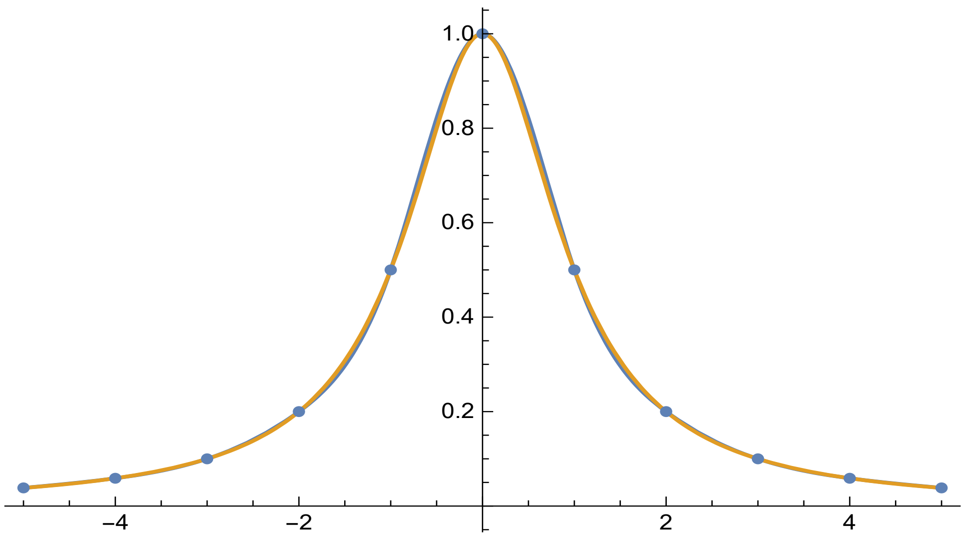
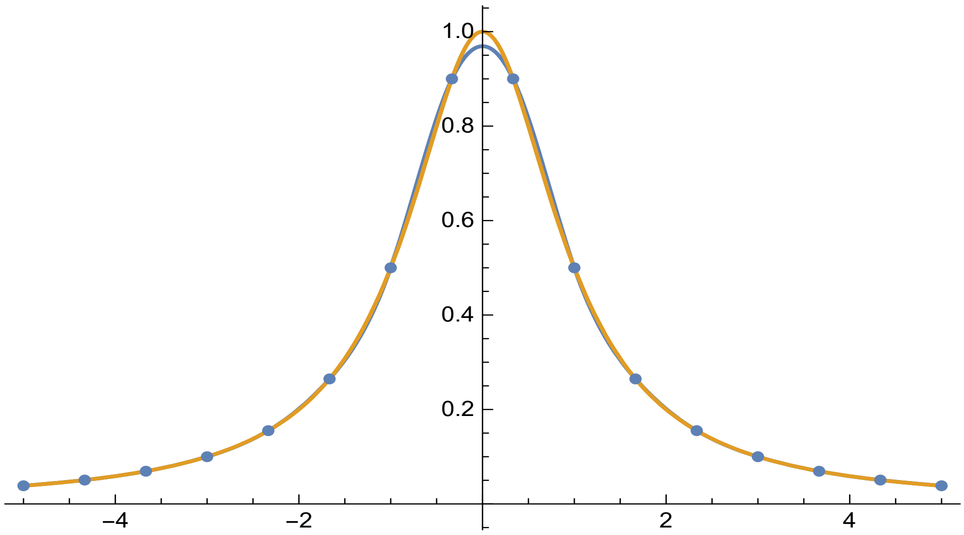
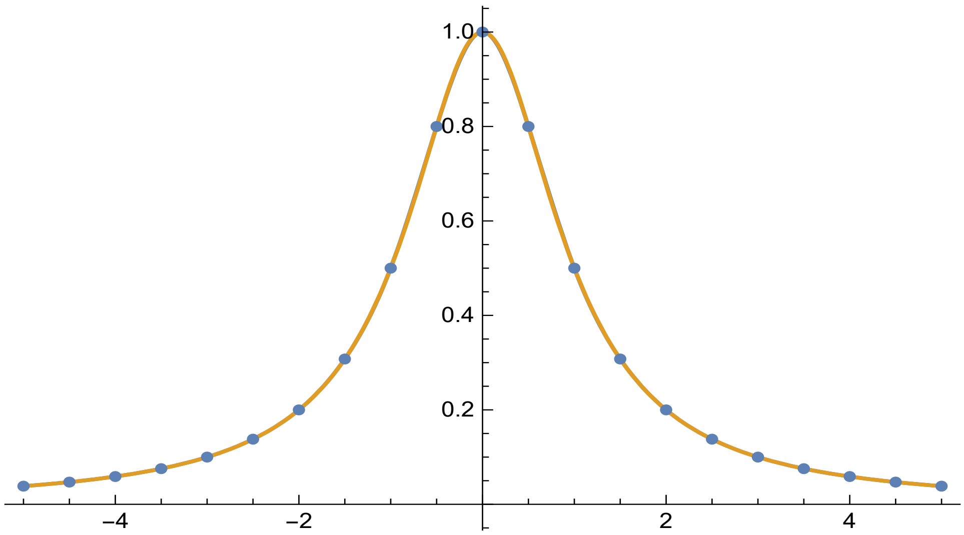


Figure 1.8: A Natural Cubic Spline Approximation of f_2, $n = 10$



(a) f_2, $n = 15$



(b) f_2, $n = 20$

Figure 1.9: A Natural Cubic Spline Approximation of f_2.

The above procedure is to construct the cubic interpolation spline from a set of data points using the continuity constraints $\mathcal{C}^0$, $\mathcal{C}^1$ and $\mathcal{C}^2$, and it requires two additional final conditions. Thus, the entire cubic splines function is determined by one system of linear equations. Therefore, a local fluctuation of some data points may affect the global spline interpolant function. An alternative to overcome this limitation is to use the Hemite cubic spline interpolation.

1.3.4 Polynomial Hermite Cubic Spline Interpolation

We build an interpolant in this section that only depends on local data values. This cubic piecewise polynomial interpolates the value and the first derivative at two data points. So we can find a sequence of cubic polynomials that interpolate the data, joined with continuous first derivatives, given a collection of points with function values and corresponding first derivatives. This interpolant is the unique solution to the following Hermite problem.

Problem 1.3.1. *Let consider a partition* $a = x_1 < \cdots < x_n = b$ *of the interval* $I := [a, b]$ *and the given data* $\{x_i, f_i^0, f_i^1\}_{i=1}^n$. *Find a cubic polynomial spline* $p(x)$ *such that*

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

It is known that Problem 1.3.1 has a unique solution $p(x)$ locally expressed in each sub-interval $\ell_i = [x_i, x_{i+1}]$ in terms of the function and the derivative values of the approximated function at nods x_i and x_{i+1}. More precisely the Hermite polynomial spline $p(x)$ is written as below,

$$p_{|\ell_i}(x) := \sum_{k=0}^{1} \sum_{j=0}^{1} f_{i+k}^j p_{i+k}^j(x), \tag{1.3.10}$$

where $p_{i+k}^j(x)$, $k = 0, 1, j = 0, 1$, are the classical Hermite polynomial basis restricted to the sub-interval $[x_i, x_{i+1}]$ and they are the unique solution of the following system of linear equations

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

The basis $p_{i+k}^j(x)$, $k = 0, 1, j = 0, 1$, can be given explicitly as follows,

$$\begin{aligned}
p_i^0(x) &:= -\frac{(x - x_{i+1})^2 (-3x_i + x_{i+1} + 2x)}{(x_i - x_{i+1})^3},\\
p_i^1(x) &:= \frac{(x - x_i)^2 (x_i - 3x_{i+1} + 2x)}{(x_i - x_{i+1})^3},\\
p_{i+1}^0(x) &:= \frac{(x - x_i)(x - x_{i+1})^2}{(x_i - x_{i+1})^2},\\
p_{i+1}^1(x) &:= \frac{(x - x_i)^2 (x - x_{i+1})}{(x_i - x_{i+1})^2},
\end{aligned}$$

The plots of the Hermite polynomial basis $p_{i+k}^j(x)$, $k = 0, 1, j = 0, 1$, on the sub-interval $[x_i, x_{i+1}] = [0, 1]$ are shows in Figure 1.10.

In the traditional interpolation problems, the f_i^0 values are given data and the f_i^1 derivatives are yet to be determined. The classical cubic spline interpolant is obtained by solving a system of the f_i^1 so that $p \in C^2([a, b])$ and adding two addi-

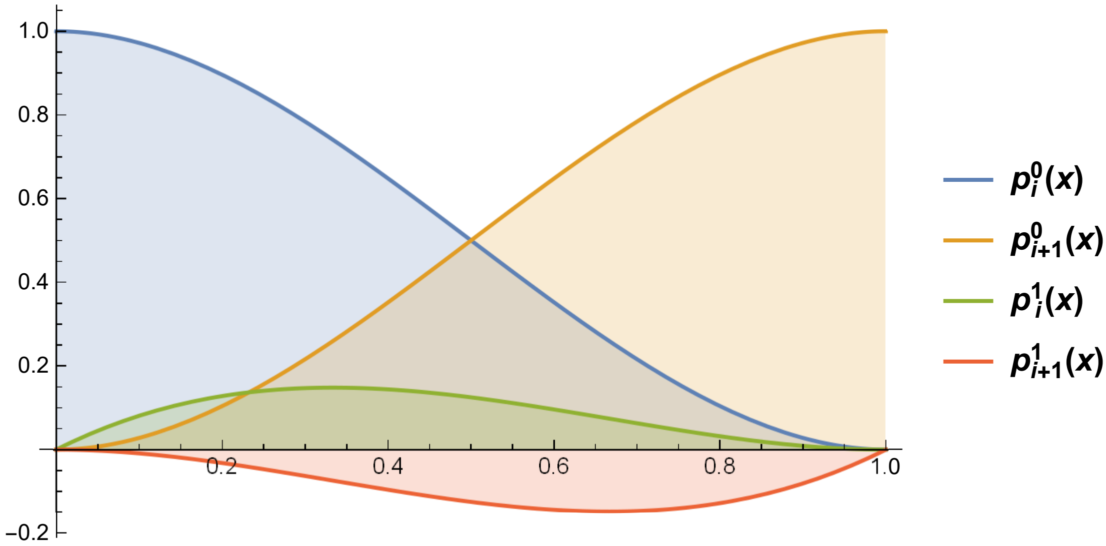


Figure 1.10: Plots of the Hermite polynomial basis on $[x_i, x_{i+1}] = [0, 1]$.

tional conditions. However, the classical cubic spline interpolant curves may have unsatisfactory oscillation especially near to the frontier.

1.3.5 Slope Estimation

In most data fitting problems the values f_i^0 are given but the first derivative values f_i^1 are not given.
In this chapter in order to determine the values of the derivatives, a system of f_i^1 is obtained by minimizing the functional

$$J_k\left(f_1^1, \dots, f_n^1\right) := \int_a^b \left(p^{(k)}(x) - L^{(k)}(x)\right)^2 dx \tag{1.3.11}$$

$$= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \left(p_i^{(k)}(x) - L_i^{(k)}(x)\right)^2 dx, \tag{1.3.12}$$

for $k = 1$, where L is the linear interpolating spline with pieces

$$L_i(x) := \frac{x_{i+1} - x}{x_{i+1} - x_i} f_i^0 + \frac{x - x_i}{x_{i+1} - x_i} f_{i+1}^0.$$

The author in [17] propose an optimum polynomial Hermite spline obtained by minimizing J_1.
The following tridiagonal system of normal equations is results from this minimiza-

tion

$$\begin{pmatrix} 4 & -1 & & & & \\ -\frac{1}{2} & 4 & -\frac{1}{2} & & & \\ & -\frac{1}{2} & 4 & -\frac{1}{2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{2} & 4 & -\frac{1}{2} \\ & & & & -1 & 4 \end{pmatrix} \begin{pmatrix} f_1^1 \\ \\ \vdots \\ \\ f_n^1 \end{pmatrix} = 3 \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}, \tag{1.3.13}$$

$$d_1 := \left(f_2^0 - f_1^0\right), d_i := \frac{f_{i+1}^0 - f_{i-1}^0}{2h}, i = 2, \cdots, n-1, \text{ and } d_n = \left(f_n^0 - f_{n-1}^0\right).$$

This system is strictly diagonally dominant and therefore has a unique solution.

The interpolant p constructed by minimizing $J_1\left(f_1^1, \ldots, f_n^1\right)$ is the polynomial cubic spline so that p' is the optimal approximation of L'. The interpolant p has minimal derivative oscillation to L' by choosing all f_i^1. Moreover, the monotonicity of p approximate the monotonicity of the given data.

Theorem 1.3.5. *[17] Let* $d_1 := \left(f_2^0 - f_1^0\right), d_i := \dfrac{f_{i+1}^0 - f_{i-1}^0}{2h}, i = 2, \cdots, n-1, d_n = \left(f_n^0 - f_{n-1}^0\right)$ *and* $(\omega_{i,j}) = 3A^{-1}$ *with* $A = (a_{i,j})$ *is the matrix of the system* (1.3.13) *.*

$$f_i^1 = \sum_{j=1}^{n} \omega_{i,j} d_j, \quad i = 1, \ldots, n. \tag{1.3.14}$$

Where all $\omega_{i,j} \geq 0$ *and* $\sum_{j=1}^{n} \omega_{i,j} = 1, \quad i = 1, \ldots, n.$

Proof 1.3.3. *Let* $f^1 = (f_1^1, f_2^1, \ldots, f_n^1)^T$, $d = (d_1, d_2, \ldots, d_n)^T$, *then* $m = 3A^{-1}d$, *i.e.,* $f_i^1 = \sum_{j=1}^{n} \omega_{i,j} d_j, \quad i = 1, \ldots, n.$
Since the matrix A *is strictly diagonally dominant,* $a_{i,j} \leq 0, i \neq j$ *and* $a_{i,i} > 0, i = 1, \ldots, n$, *then all the elements of the matrix* A *are nonnegative.*
Therefore $\omega_{i,j} > 0, \quad i = 1, \ldots, n.$
Let $e = (1, 1, \ldots, 1)^T$, *we have* $Ae = 3e$ *and so* $3A^{-1}e = e$.
This means that $\sum_{j=1}^{n} \omega_{i,j} = 1, \quad i = 1, \ldots, n.$

From Theorem 1.3.5 we can know that f_i^1 preserves the same sign as d_i if all d_j have the same sign.

1.3.5.1 Numerical Examples

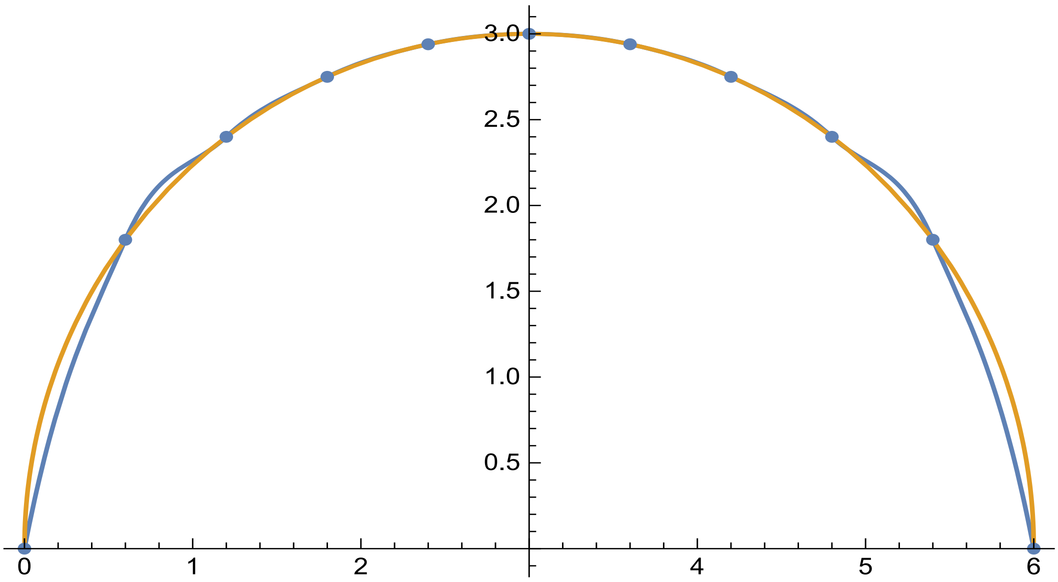
We finish this section by presenting some numerical examples to illustrate the above theoretical results.
The first example is to interpolate the function

$$f_1(x) = \sqrt{9 - (x-3)^2}$$

on the interval $[0, 6]$. To this end let consider an uniform subdivision of the interval $[0, 6]$ with the break-points $x_i = ih$, $i = 0, \dots, n$, where $h = \dfrac{6}{n}$.
Figure 1.12 shows the interpolation curve.
The second example is to approximate the function

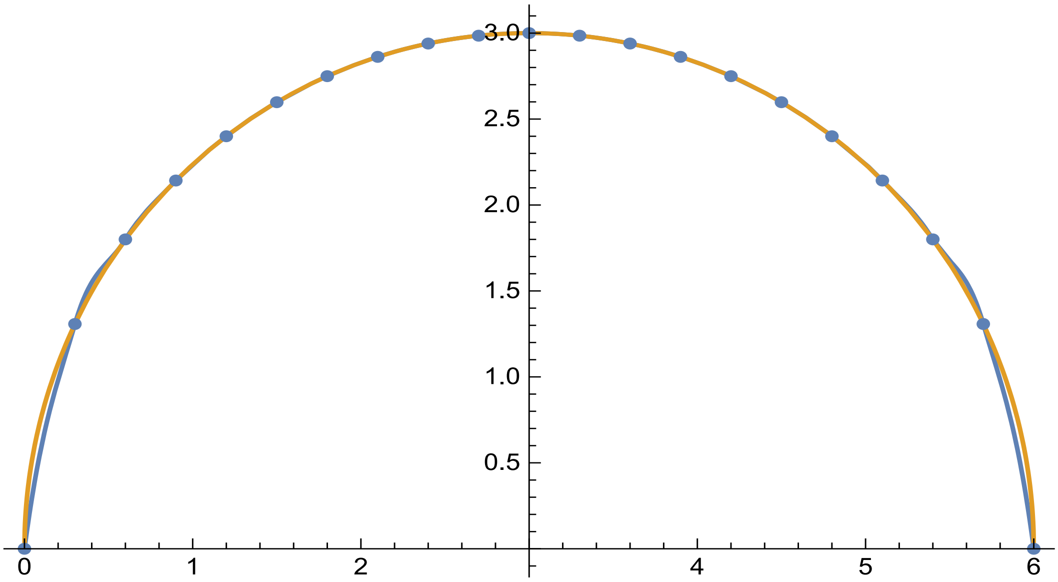
$$f_2(x) = \frac{1}{1+x^2}$$

on the interval $[-5, 5]$. Let consider an uniform subdivision of the interval $[-5, 5]$ with the break-points $x_i = -5 + ih$, $i = 0, \dots, n$, where $h = \dfrac{10}{n}$.
Figure 1.13 shows the interpolation curve.
The third example is to consider the monotone data taken from the function $f_3(x) = \dfrac{1}{x^2}$.
Figure 1.14 shows the interpolation curve.
The fourth example is to consider the monotone data taken from the function $f_4(x) = \sqrt{x}$.
Figure 1.15 shows the interpolation curve.

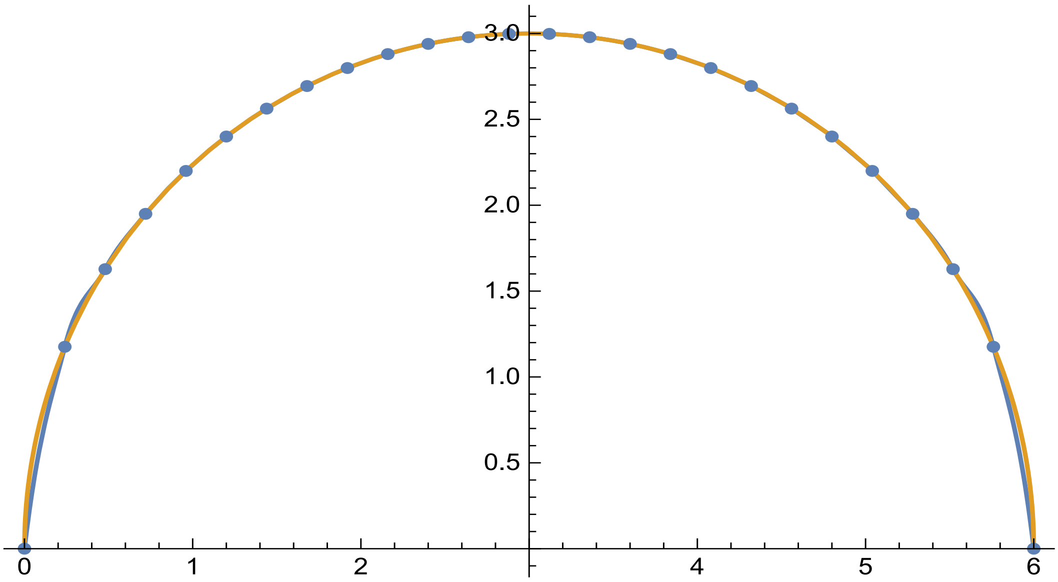


(a) f_1, $n = 10$

Figure 1.11: Approximation by p for f_1 with $n = 10$.

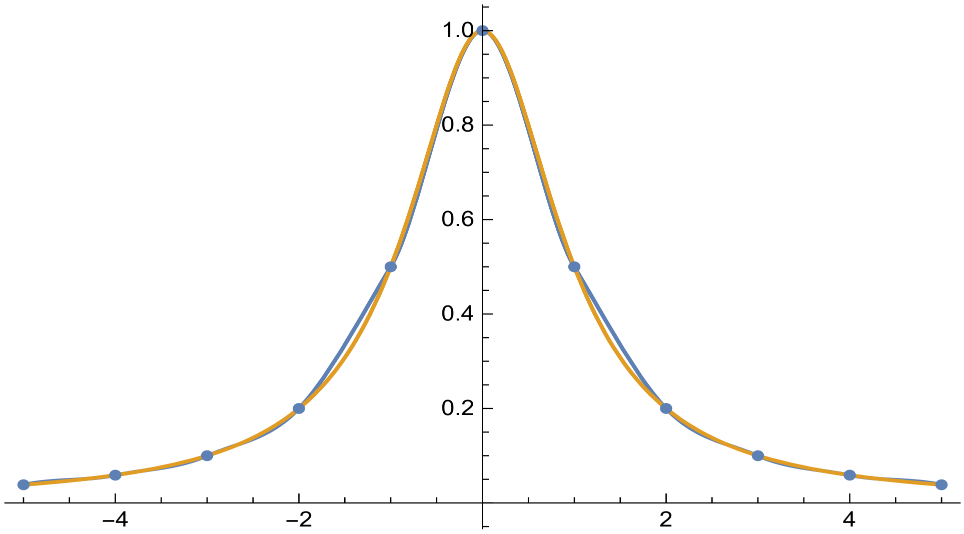


(a) f_1, $n = 20$

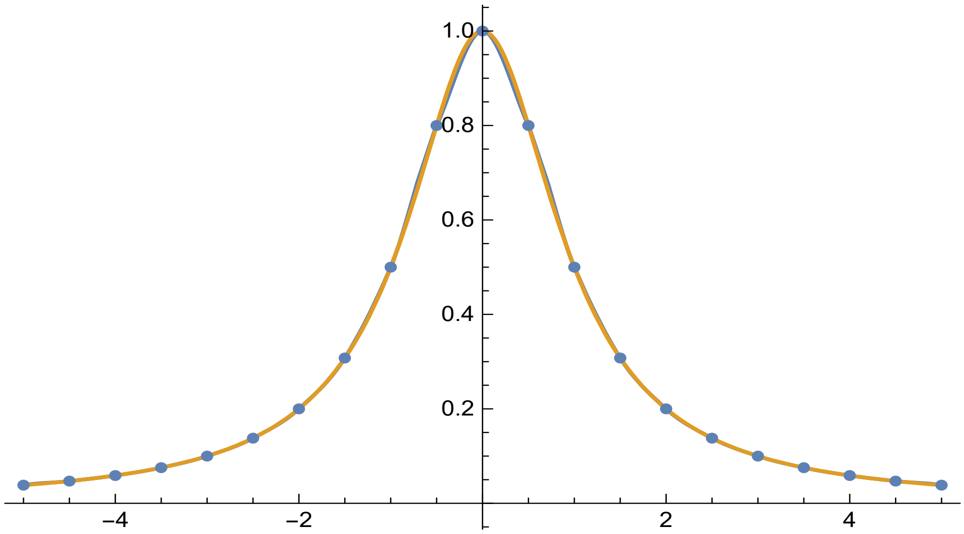


(b) f_1, $n = 25$

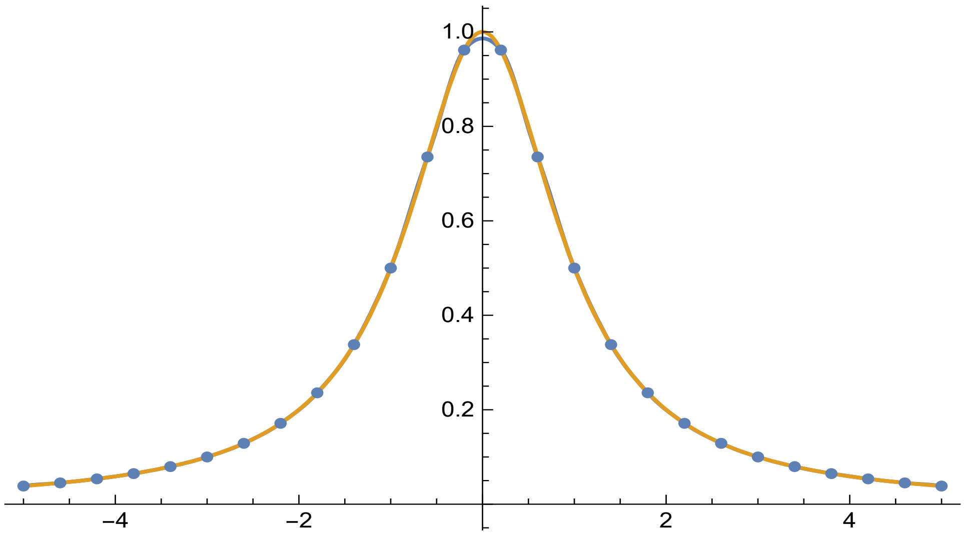
Figure 1.12: Approximation by p: cubic polynomial Hermite interpolating splines for f_1.



(a) f_2, $n = 10$

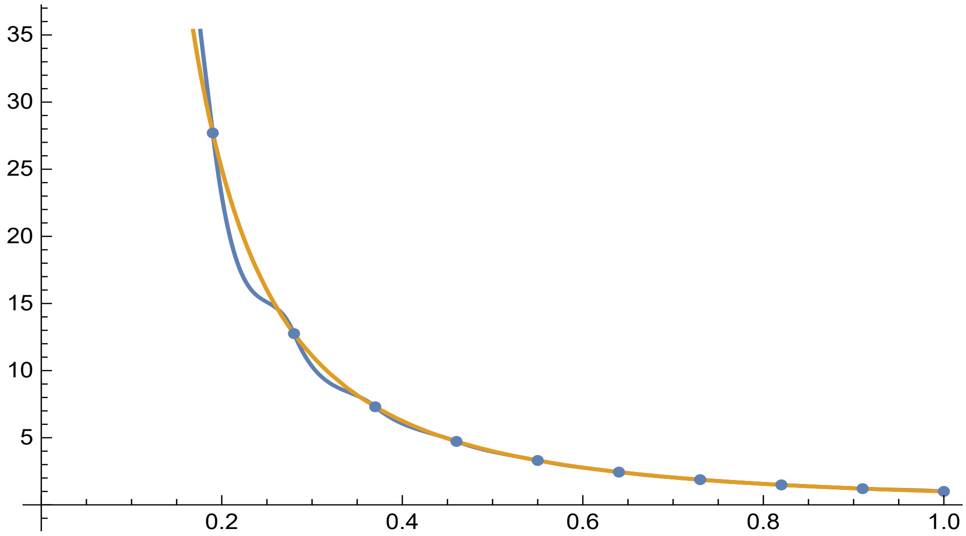


(b) f_2, $n = 20$

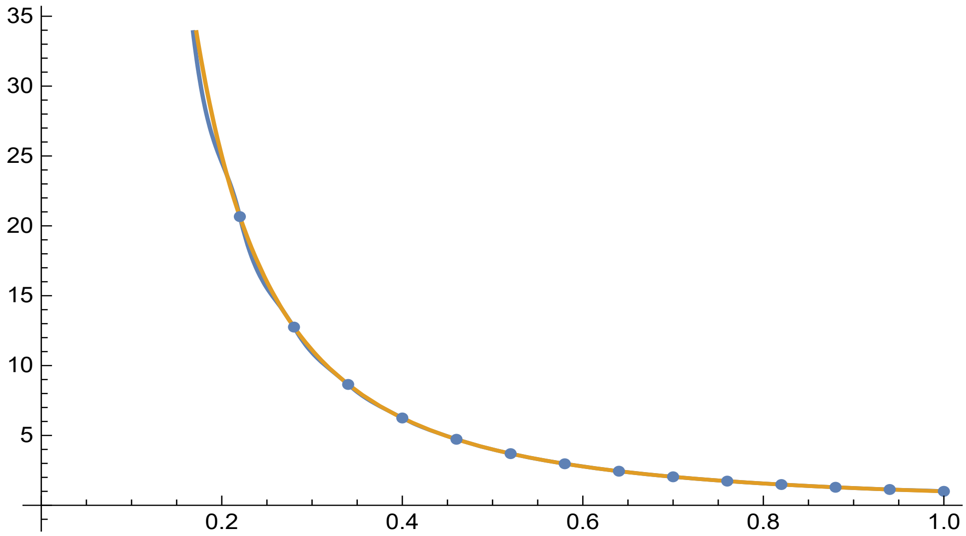


(c) f_2, $n = 25$

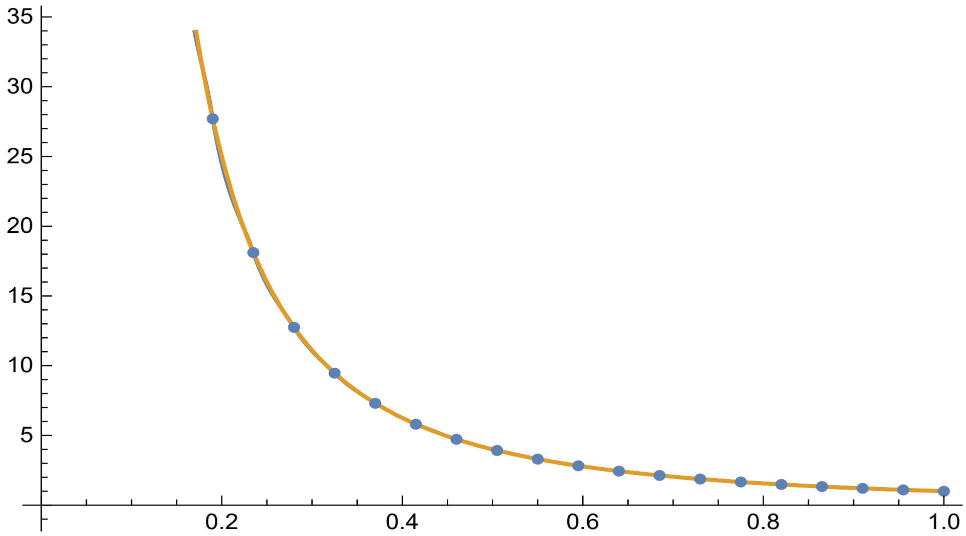
Figure 1.13: Approximation by p: cubic polynomial Hermite interpolating splines for f_2.



(a) f_3, $n = 10$

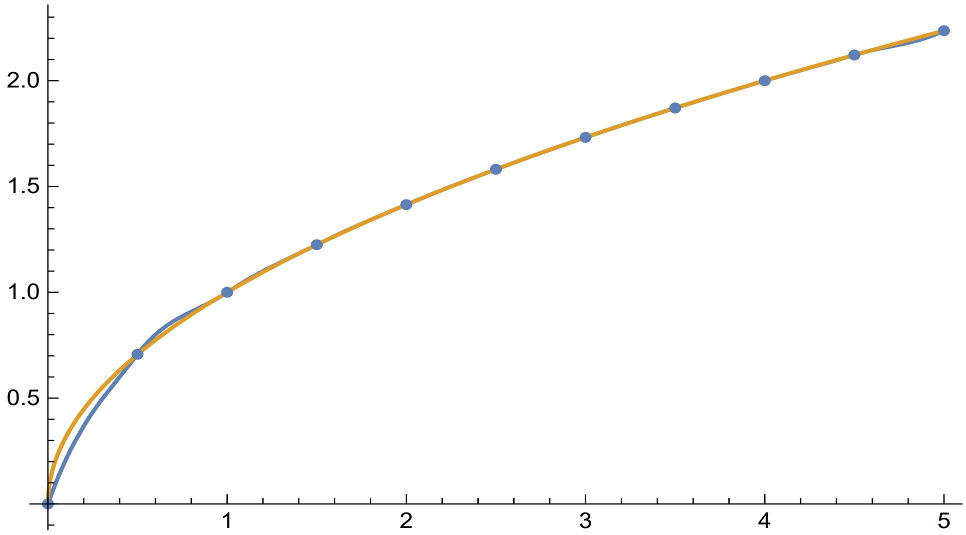


(b) f_3, $n = 20$

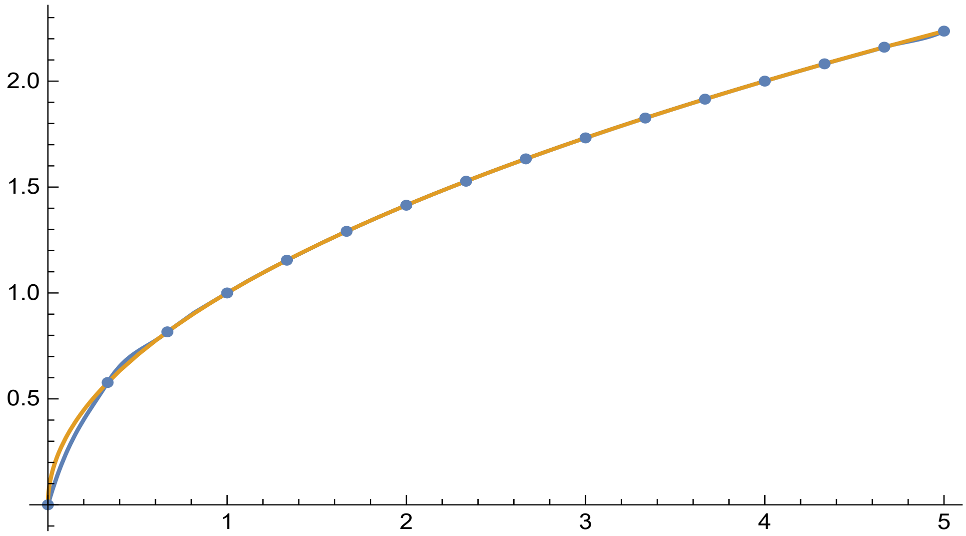


(c) f_3, $n = 25$

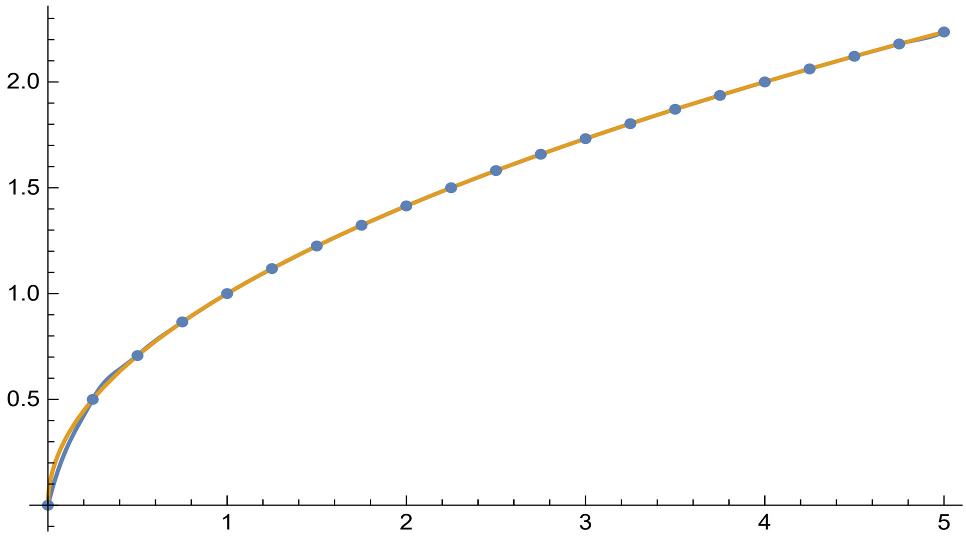
Figure 1.14: Interpolating monotone data from f_3 : p in blue, f_3 in orange and data taken from f_3.



(a) f_4, $n = 10$



(b) f_4, $n = 15$



(c) f_4, $n = 20$

Figure 1.15: Interpolating monotone data from f_4 : p in blue, f_4 in orange .

Chapter 2

Algebraic Trigonometric C^1 Hermite Spline Interpolation

Contents

In this chapter, we introduce a new C^1 cubic Hermite interpolation methods based on Algebraic Trigonometric (AT) spline functions. In the first one, values of a function and its first derivatives are interpolated. In the second one, a C^1 AT-spline of low degree which preserves the monotonicity of the given data is defined by adding additional knots. We illustrate the results by giving numerical examples of both interpolation schemes.

The results obtained in this chapter is presented in paper [40].

2.1 Cubic Algebraic Trigonometric (AT) Hermite Splines of Class C^1

Consider a partition $a = x_1 < \cdots < x_n = b$ of the interval $I := [a, b]$ (I is of length less than or equal to 2π). Given values f_i^0, f_i^1, $1 \leq i \leq n$, it is well-known that for all $i \in \{1, \ldots, n-1\}$ there exists a unique cubic polynomial $P_i \in \mathbb{P}_3$ such that

$P_i(x_{i+j}) = f^0_{i+j}$ and $P'_i(x_{i+j}) = f^1_{i+j}$, $j = 0, 1$. The function P defined by assembling the polynomial pieces P_i is a $\mathcal{C}^1$-cubic spline that interpolates the data f^0_i and f^1_i.

Therefore, consider the uniform partition $\Delta_n := \{x_i\}_1^n$ of the inteval $I = [a, b]$, with step length $h := \dfrac{b-a}{n-1}$ and knots $x_i := a + (i-1)h$. Given values f^0_i and f^1_i, $1 \le i \le n$, for all $i \in \{1, \ldots, n-1\}$ there exists a unique AT Hermite interpolant T_i $\in$ span $\{1, x, \cos x, \sin x\}$ such that $T_i(x_{i+j}) = f^0_{i+j}$ and $T'_i(x_{i+j}) = f^1_{i+j}$, $j = 0, 1$. The function T defined on I from the local interpolants T_i is a $\mathcal{C}^1$-AT spline on Δ_n that interpolates the data f^0_i and f^1_i.

Using the notation T_i for the AT cubic Hermite interpolant to the data $\left\{f^0_i, f^1_i\right\}$ and $\left\{f^0_{i+1}, f^1_{i+1}\right\}$, it is straightforward to prove that

$$T_i(x) = f^0_i \omega^0_i(x) + f^0_{i+1}\omega^0_{i+1}(x) + f^1_i \omega^1_i(x) + f^1_{i+1}\omega^1_{i+1}(x), \ x \in [x_i, \ x_{i+1}], \quad (2.1.1)$$

where ω^j_{i+k}, $k = 0, 1$, $j = 0, 1$, are the unique solution functions in *span* $\{1, x, \sin x, \cos x\}$ satisfying the following interpolation conditions.

$$\begin{array}{llll}
\omega^0_i(x_i) = 1, & \left(\omega^0_i\right)'(x_i) = 0, & \omega^0_i(x_{i+1}) = 0, & \left(\omega^0_i\right)'(x_{i+1}) = 0, \\
\omega^1_i(x_i) = 0, & \left(\omega^1_i\right)'(x_i) = 1, & \omega^1_i(x_{i+1}) = 0, & \left(\omega^1_i\right)'(x_{i+1}) = 0, \\
\omega^0_{i+1}(x_i) = 0, & \left(\omega^0_{i+1}\right)'(x_i) = 0, & \omega^0_{i+1}(x_{i+1}) = 1, & \left(\omega^0_i\right)'(x_{i+1}) = 0, \\
\omega^1_{i+1}(x_i) = 0, & \left(\omega^1_{i+1}\right)'(x_i) = 0, & \omega^1_{i+1}(x_{i+1}) = 0, & \left(\omega^1_{i+1}\right)'(x_{i+1}) = 1.
\end{array}$$

They are given explicitly as follows,

$$\omega^0_i(x) := \frac{1}{\theta'}(1, x, \sin(x), \cos(x)) \begin{pmatrix} \cos(h) + \sin(h)(h + x_i) - 1 \\ -\sin(h) \\ \sin(h + x_i) - \sin(x_i) \\ \cos(h + x_i) - \cos(x_i) \end{pmatrix},$$

$$\omega^1_i(x) := \frac{1}{\theta'}(1, x, \sin(x), \cos(x)) \begin{pmatrix} \cos(h) - \sin(h)x_i - 1 \\ \sin(h) \\ \sin(x_i) - \sin(h + x_i) \\ \cos(x_i) - \cos(h + x_i) \end{pmatrix},$$

$$\omega^0_{i+1}(x) := \frac{1}{\theta'}(1, x, \sin(x), \cos(x)) \begin{pmatrix} \sin(h) + x_i - \cos(h)(h + x_i) \\ \cos(h) - 1 \\ -\cos(x_i) + \cos(h + x_i) + h\sin(h + x_i) \\ h\cos(h + x_i) + \sin(x_i) - \sin(h + x_i) \end{pmatrix},$$

$$\omega^1_{i+1}(x) := \frac{1}{\theta'}(1, x, \sin(x), \cos(x)) \begin{pmatrix} h - \sin(h) - (\cos(h) - 1)x_i \\ \cos(h) - 1 \\ \cos(x_i) - \cos(h + x_i) - h\sin(x_i) \\ -h\cos(x_i) - \sin(x_i) + \sin(h + x_i) \end{pmatrix}.$$

Where $\theta' = h\sin(h) + 2\cos(h) - 2$.

Figure 2.1 shows the graphical representation of the basis $\omega_{i,j},\ j = 1, \ldots, 4$ on the interval $[0, 1]$.

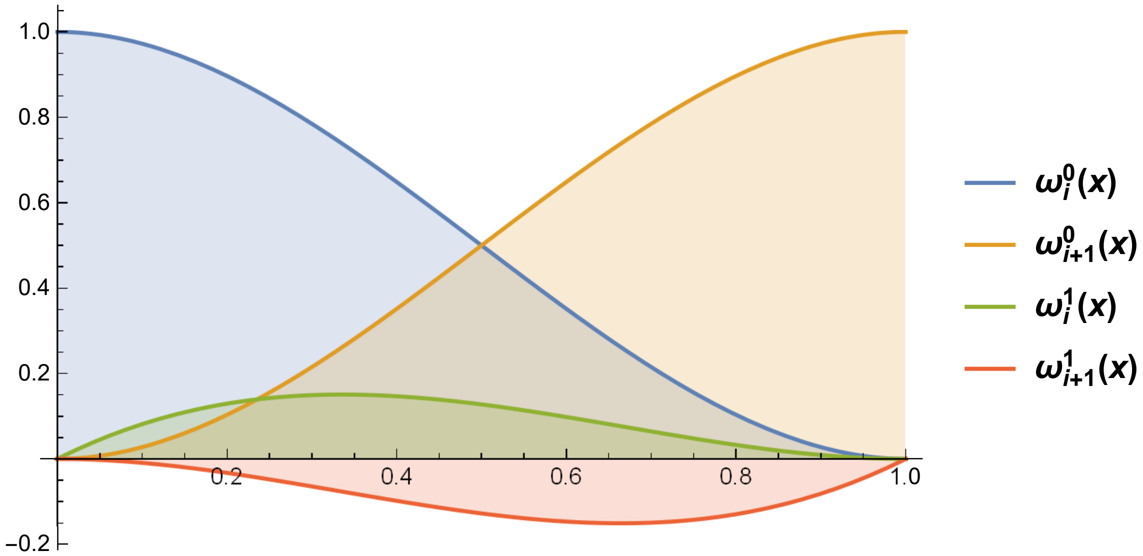


Figure 2.1: Plot of the basis ω_{i+k}^j, $k = 0, 1$, $j = 0, 1$ on the interval $[0, 1]$.

Remark 1. The constructed interpolant (2.1.1) exactly reproduces all functions in the space $\Gamma_4 := span\,\{1,\, x,\, \sin x,\, \cos x\}$.

To illustrate Remark 2.1, we approximate the function

$$f_5(x) = \pi x - \frac{\sin(x)}{5} - 3\cos(x) + 2$$

on the interval $[0, 5]$ using the operator T defined by (2.1.1). Figure 2.2 shows the plot of this approximation, Figure 2.3 represents the function difference $T(x) - f_5(x)$ and Figure 2.4 shows the approximation for f_5 by cubic polynomial Hermite splines defined by (1.3.10).

2.1.1 Slopes Estimation

The proposed interpolant T is defined from the values and first derivative values at the break-points. In general the values f_i^0, $i = 1, .., n$, are given but the values f_i^1, $i = 1, .., n$ of the first derivatives of T remain to be determined by some local or global procedure. For instance, the requirement that the spline T be of $\mathcal{C}^2$ class instead of $\mathcal{C}^1$ class is equivalent to $f^1 := \left(f_1^1, \ldots, f_n^1\right)$ being the solution of a tridiagonal system (see [1]).
In this work, to estimate the values of the derivatives f_i^1 we consider the functional:

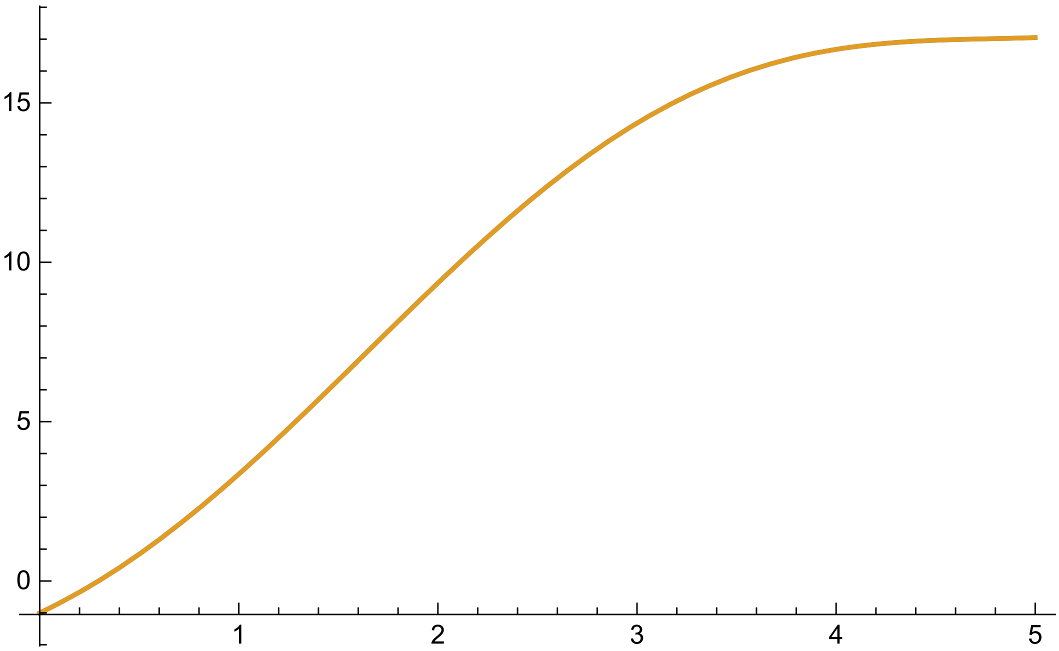


Figure 2.2: Approximation by cubic AT Hermite interpolating splines for f_5 on the interval $[0, 5]$ with $n = 1$.

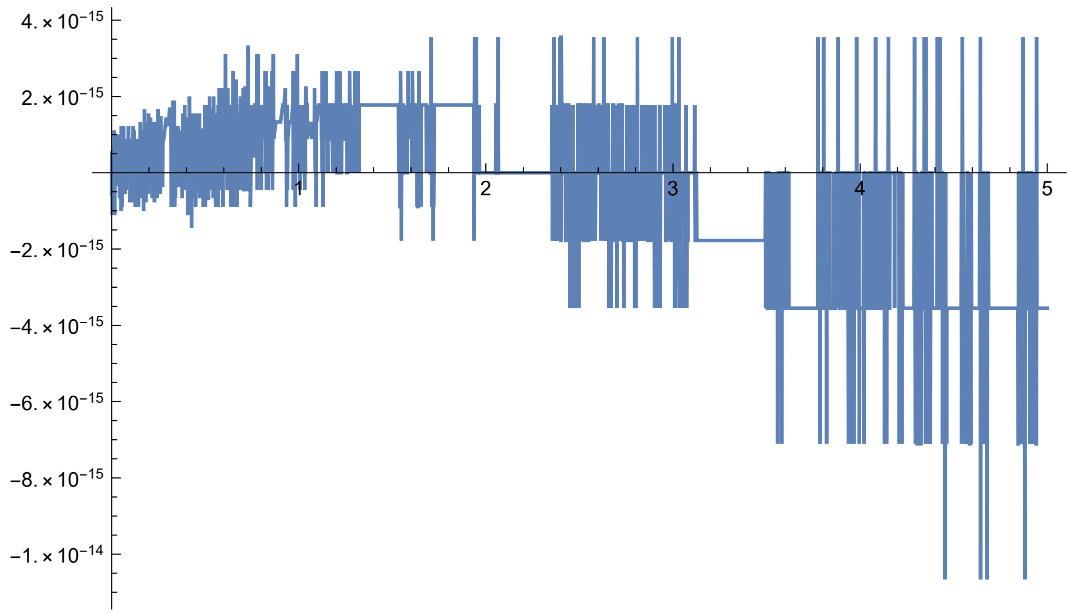


Figure 2.3: Plot of the function $T(x) - f_5(x)$ on the interval $[0, 5]$.

$$J_k\left(f^1\right) := \int_a^b \left(T^{(k)}(x) - L^{(k)}(x)\right)^2 dx \tag{2.1.2}$$

$$= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \left(T_i^{(k)}(x) - L_i^{(k)}(x)\right)^2 dx, \ k = 0, 1, 2, \tag{2.1.3}$$

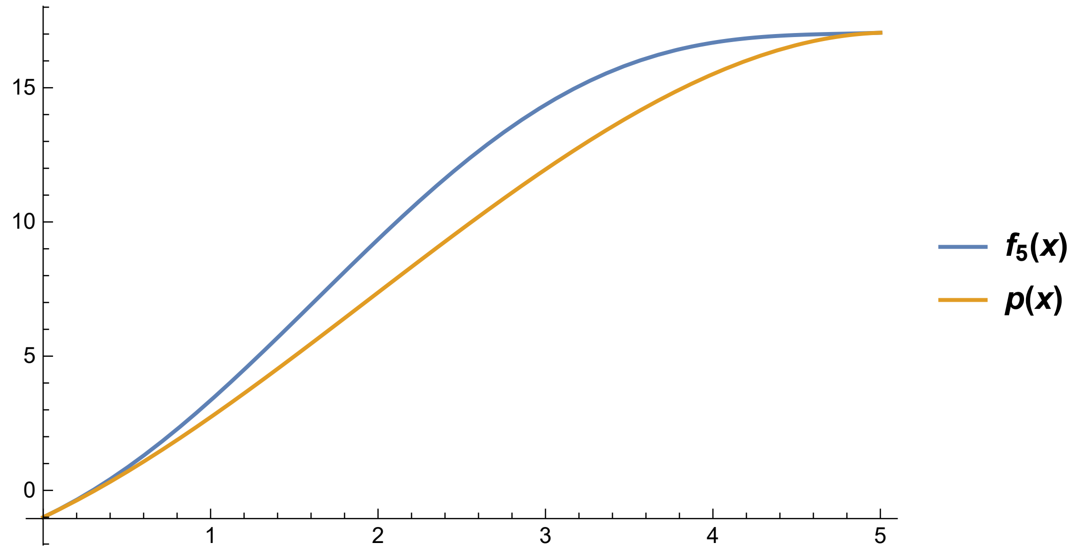


Figure 2.4: Approximation by cubic polynomial Hermite interpolating splines for f_5 on the interval $[0, 5]$ with $n = 1$.

where L is the linear interpolating spline with pieces

$$L_i(x) := \frac{x_{i+1} - x}{x_{i+1} - x_i} f_i^0 + \frac{x - x_i}{x_{i+1} - x_i} f_{i+1}^0.$$

It is the simplest shape-preserving interpolant, but not regular enough.

The minimization of J_2 provides the natural cubic spline. The method in [6] consists of the minimization of J_0, that produces the interpolating spline with minimal mean quadratic oscillation. In [17], J_1 is minimized in order to determine the slopes f_i^1 yielding the spline T having minimal mean derivative oscillation to L'. The proposed methods do not provide satisfactory results when such ubiquitous curves as conics and colloids are interpolated. That is why in this work, we propose to explore the performance of non-polynomial spline spaces with respect to the capacity to reproduce such curves and the minimization of the mean oscillation of the derivative.

From these local AT Hermite interpolants a $\mathcal{C}^1$ AT Hermite spline interpolant is produced, and the slopes f_i^1 could be chosen as in [17] by minimizing the functional J_1 defined in (2.1.3). A straightforward calculation shows that

$$\begin{aligned} J_1\left(f^1\right) &= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \left(T_i'(x) - L_i'(x)\right)^2 dx \\ &= \sum_{i=1}^{n-1} -\frac{1}{8h\left(h\cot\left(\frac{h}{2}\right) - 2\right)^2} \left((f_i^1)^2 A_h^1 + \frac{f_i^1\left(f_{i+1}^1 A_h^2 + A_h^3\right)}{(\cos h - 1)^2} \right. \\ &\left. + \frac{2\left((f_{i+1}^1)^2 A_h^4 + f_{i+1}^1 A_h^5 + A_h^6\right)}{\cos h - 1} \right). \end{aligned}$$

With,

$$\begin{aligned}
A_h^1 &:= \frac{1}{2}h^5\csc^4\left(\frac{h}{2}\right)\left(-2h^3+4(h^2+1)\sin h+(h^2-2)\sin(2h)-4h\cos h+4h\cos(2h)\right),\\
A_h^2 &:= 8h^5(-(3h^2+2)\sin h+(h^2-6)h\cos h+6h+\sin(2h)),\\
A_h^3 &:= 8h^5(\cos h-1)(f_i^0-f_{i+1}^0)(h^2+h\sin h+4\cos h-4),\\
A_h^4 &:= h^5\left(2(h^2-2)\sin h+h\left(8\cos h+h(h-3\sin h)\csc^2\left(\frac{h}{2}\right)+4\right)\right),\\
A_h^5 &:= 4h^5(f_i^0-f_{i+1}^0)(h^2+h\sin h+4\cos h-4),\\
A_h^6 &:= 4(2h^6-2h^4-7h^2+(h^6-2h^4+9h^2-4)\cos h+4\\
&\quad +h(-3h^4+8h^2-4)\sin h)(f_i^0-f_{i+1}^0)^2.
\end{aligned}$$

Minimization of $J_1\left(f_1^1,\ldots,f_n^1\right)$ leads to the following tridiagonal system of normal equations:

$$\begin{pmatrix}
\mu & 2\nu & & & & \\
\nu & \mu & \nu & & & \\
 & \nu & \mu & \nu & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & \nu & \mu & \nu\\
 & & & & 2\nu & \mu
\end{pmatrix}
\begin{pmatrix} f_1^1\\ \\ \vdots\\ \\ f_n^1\end{pmatrix}
=\eta\begin{pmatrix} d_1\\ d_2\\ \vdots\\ d_{n-1}\\ d_n\end{pmatrix}, \tag{2.1.4}$$

with

$$\begin{aligned}
&d_1 := 2\left(f_2^0-f_1^0\right), d_i := f_{i+1}^0-f_{i-1}^0, i=2,\ldots,n-1, \text{ and } d_n = 2\left(f_n^0-f_{n-1}^0\right),\\
&\nu := -\frac{h^4\csc^4\left(\frac{h}{2}\right)}{4\left(h\cot\left(\frac{h}{2}\right)-2\right)^2}\mathcal{P},\\
&\mu := -\frac{h^4\csc^4\left(\frac{h}{2}\right)}{4\left(h\cot\left(\frac{h}{2}\right)-2\right)^2}\mathcal{Q},\\
&\eta := \frac{h^4\left(h^2+h\sin h+4\cos h-4\right)}{h^2+(h^2-4)\cos h-4h\sin h+4},
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{P} &:= -\left(3h^2+2\right)\sin h+\left(h^2-6\right)h\cos h+6h+\sin(2h),\\
\mathcal{Q} &:= -2h^3+4\left(h^2+1\right)\sin h+\left(h^2-2\right)\sin(2h)-4h\cos h+4h\cos(2h).
\end{aligned}$$

Theorem 2.1.1. *For $h\in\,]0,\,2\pi[$, the main matrix in system (2.1.4) is a diagonally-dominant matrix.*

Proof. Note that

$$\begin{aligned}
\mathcal{Q}-2\mathcal{P} &= \left[-\left(3h^2+2\right)\sin h+\left(h^2-6\right)h\cos h+6h+\sin(2h)\right] \\
&-2\left[-2h^3+4\left(h^2+1\right)\sin h+\left(h^2-2\right)\sin(2h)-4h\cos h+4h\cos(2h)\right], \\
&= -2h^3-2h^3\cos h+10h^2\sin h+h^2\sin(2h)-12h+8\sin h-4\sin(2h) \\
&+8h\cos h+4h\cos(2h), \\
&= -2[h^3+h^3\cos h-4h^2\sin h+4h-4h\cos h]+[2h^2\sin h+h^2\sin(2h) \\
&-4h+8\sin h-4\sin(2h)+4h\cos(2h)], \\
&= -4h\left(h\cos\left(\frac{h}{2}\right)-2\sin\left(\frac{h}{2}\right)\right)^2+4\sin h\left(h\cos\left(\frac{h}{2}\right)-2\sin\left(\frac{h}{2}\right)\right)^2, \\
&= 4\left(h\cos\left(\frac{h}{2}\right)-2\sin\left(\frac{h}{2}\right)\right)^2(\sin h-h).
\end{aligned}$$

Then

$$\begin{aligned}
\mu-2\nu &= -\frac{h^4\csc^4\left(\frac{h}{2}\right)}{4\left(h\cot\left(\frac{h}{2}\right)-2\right)^2}\left[\mathcal{Q}-2\mathcal{P}\right], \\
&= -\frac{h^4\csc^4\left(\frac{h}{2}\right)}{4\left(h\cot\left(\frac{h}{2}\right)-2\right)^2}\left[4\left(h\cos\left(\frac{h}{2}\right)-2\sin\left(\frac{h}{2}\right)\right)^2(\sin h-h)\right], \\
&= h^4(h-\sin h)\csc^2\left(\frac{h}{2}\right).
\end{aligned}$$

Then, to prove that the coefficient matrix is diagonally-dominant matrix, it suffices to demonstrate that, $h-\sin h > 0$.
To this end, we consider the univariate function $g(x) := x-\sin(x)$.
Its first derivative is non-negative, so that g is positive on $]0,\, 2\pi[$. $\square$

Let $\mathbb{A} := (a_{ij})_{1\le i,j\le n}$ be the coefficient matrix of system (2.1.4). The index on the diagonally dominant property of matrix A is given by

$$D_h(\mathbb{A}) := \max_{i=1,\dots,n}\left(\frac{1}{|a_{ii}|}\sum_{\substack{j=1\\ j\neq i}}^{n}|a_{i,j}|\right) = -2\max_{h>0}\frac{\nu(h)}{\mu(h)}.$$

To compute $\lim_{h\to 0} D_h(\mathbb{A})$, we plug in the Taylor series for $\nu(h)$ and $\mu(h)$. Then

$$\lim_{h\to 0} D_h(\mathbb{A}) = -2\lim_{h\to 0}\frac{\nu(h)}{\mu(h)} = -2\lim_{h\to 0}\frac{-\frac{h^5}{15}-\frac{13h^7}{3150}-\frac{11h^9}{63000}}{\frac{8h^5}{15}+\frac{22h^7}{1575}+\frac{h^9}{2250}} = \frac{1}{4}.$$

This value equals the value D_1 in [17], therefore also in the case of $\mathcal{C}^1$-AT splines the numerical stability of LU factorization method is satisfactory for solving (2.1.4) for enough small h.

Proposition 2.1.1. *The derivative values f_i^1 preserve the same sign as d_i if all d_i have the same sign.*

Proof. Let $b_{ij} = \eta \mathbb{A}^{-1}$, so we can write f_i^1 as

$$f_i^1 = \sum_{j=1}^{n} b_{ij} d_j,$$

Since $\mathbb{A}$ is a strictly diagonally dominant matrix with $a_{i,j} \leq 0$ for $i \neq j$ and $a_{i,i} > 0$ for $i = 1, 2 \ldots, n$, then the elements of matrix $\mathbb{A}^{-1}$ are non-negative and therefore all $b_{ij} \geq 0$ taking into account that $\eta \geq 0$.

$\square$

We follow procedure in [84] to analyze the interpolation error. We need the following lemma to establish an error bound for our operator.

Lemma 2.1.1. *Let $f \in \mathcal{C}^3([a, b])$, then the local truncation errors t_i, $i = 1, 2, \ldots, n$, associated with the scheme (2.1.4) are given by the expressions*

$$t_i = \left(\nu h^2 - \eta \frac{h^3}{3}\right) f_i^{(3)} + o\left(h^2\right).$$

Proof. Since f is of class $\mathcal{C}^3([a, b])$, then

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

From equations (2.1.4), it follows that

$$\nu f'(x_{i-1}) + \mu f'(x_i) + \nu f'(x_{i+1}) = \eta \left(f(x_{i+1}) - f(x_{i-1})\right).$$

Replacing $f'(x_{i-1})$, $f'(x_{i+1})$ and $f(x_{i+1}) - f(x_{i-1})$ by their Taylor expansions, we get

$$\begin{aligned}
t_i =& \left((2\nu + \mu) f'(x_i) + \nu h^2 f^{(3)}(x_i) + o\left(h^2\right)\right) - \left(2\eta h f'(x_i) + \eta \frac{h^3}{3} f^{(3)}(x_i) + o\left(h^3\right)\right) \\
=& (2\nu + \mu - 2\eta h) f'(x_i) + \left(\nu h^2 - \eta \frac{h^3}{3}\right) f^{(3)}(x_i) + o\left(h^2\right).
\end{aligned}$$

Through a straightforward computation, one gets

$$\begin{aligned} 2\nu+\mu-2\eta h &= -\frac{h^4\csc^4\left(\frac{h}{2}\right)}{4\left(h\cot\left(\frac{h}{2}\right)-2\right)^2}\left(2\mathcal{P}+\mathcal{Q}\right)-2\eta h, \\ &= \frac{h^5\left(h^2+h\sin h+4\cos h-4\right)}{\left(h\cos\left(\frac{h}{2}\right)-2\sin\left(\frac{h}{2}\right)\right)^2}-\frac{2\left(h^7+h^6\sin h-4h^5+4h^5\cos h\right)}{h^2+h^2\cos h-4h\sin h-4\cos h+4}, \\ &= \frac{2\left(h^7+h^6\sin h-4h^5+4h^5\cos h\right)}{h^2+h^2\cos h-4h\sin h-4\cos h+4}-\frac{2\left(h^7+h^6\sin h-4h^5+4h^5\cos h\right)}{h^2+h^2\cos h-4h\sin h-4\cos h+4}, \\ &= 0. \end{aligned}$$

Moreover, $\nu h^2-\eta\dfrac{h^3}{3}\neq 0$, which completes the proof.

□

The next result is based on the above lemma.

Theorem 2.1.2. *Let $f\in\mathcal{C}^3\left([a,\,b]\right)$, then*

$$\left|f_i^1-f'\left(x_i\right)\right|=o\left(h^2\right).$$

Proof. For the sake of simplicity, we write $f_i=f\left(x_i\right)$ and $f_i'=f'\left(x_i\right)$.
Let $\mathbb{F}=\left(f_i'\right),\mathbb{M}=\left(f_i^1\right)$, $\mathbb{T}=\left(t_i\right)$, $\mathbb{C}=\eta\left(d_i\right)$ for $i=1,\cdots,n$, and $\mathbb{E}=\mathbb{F}-\mathbb{M}$ all be n-dimensional columns vectors.

With these notations, the system (2.1.4) can be written as

$$\begin{aligned} \mathbb{A}\,\mathbb{M} &= \mathbb{C}, \\ \mathbb{A}\,\mathbb{F} &= \mathbb{C}+\mathbb{T}, \end{aligned}$$

from which we get

$$\mathbb{A}\,\mathbb{E}=\mathbb{T}.$$

Then,

$$\|\mathbb{E}\|_\infty\leq\left\|\mathbb{A}^{-1}\right\|_\infty\|\mathbb{T}\|_\infty,$$

and we deduce for $i=1,\,2,\,\ldots,\,n$, the inequalities

$$\left|f_i^1-f_i'\right|\leq K_1\;\|\mathbb{T}\|_\infty,$$

where $K_1:=\|A^{-1}\|_\infty$.

Now, using the fact that f is in $\mathcal{C}^3\left([a,\,b]\right)$, $\nu=\dfrac{8h^5}{15}+o\left(h^7\right)$ and $\eta=\dfrac{h^4}{5}+\dfrac{h^6}{350}+o\left(h^7\right)$.

We deduce that, $\nu h^2-\eta\dfrac{h^3}{3}=-\dfrac{2h^7}{15}+o\left(h^9\right)$, and

$$|t_i|\leq K_2\frac{2h^7}{15}+o\left(h^2\right),$$

where K_2 is a constant such that $\left|f^3(x)\right| \leq K_2$, $\forall x \in [a,b]$.
Finally, it holds

$$|f_i^1 - f_i'| \leq K_1 K_2 \frac{2h^7}{15} + o(h^2),$$

and the proof is complete.

□

We summarize the above results in the next theorem.

Theorem 2.1.3. *Let $f \in \mathcal{C}^4([a,b])$ and T be the interpolation operator defined by (2.1.1). For a uniform step size h, we have*

$$\|f(x) - T(x)\|_\infty = o(h^2).$$

Proof. Firstly, let P be another cubic algebraic trigonometric function satisfying the conditions

$$\mathrm{P}^{(j)}(x_k) = f^{(j)}(x_k), \text{ for all } j = 0,1, \text{ and } k = i, i+1.$$

With $f^{(0)}(x_i) = f(x_i) = f_i$ and $f^{(1)}(x_i) = f'(x_i) = f_i'$.
The function P is defined on $[x_i,\, x_{i+1}]$ as

$$\mathrm{P}\,(x) = f_i\omega_i^0(x)\,(x) + f_{i+1}\omega_{i+1}^0(x)\,(x) + f_i'\omega_i^1(x)\,(x) + f_{i+1}'\omega_{i+1}^1(x)\,(x)\,.$$

Thus,

$$\begin{aligned}|\mathrm{P}(x) - T(x)| &\leq \max_{x_i \leq x \leq x_{i+1}} |(f_i' - f_i^1)\omega_i^1(x) + (f_{i+1}' - f_{i+1}^1)\omega_{i+1}^1(x)| \\ &\leq |f_i' - f_i^1| \|\omega_i^1(x)\|_\infty + |f_{i+1}' - f_{i+1}^1| \|\omega_{i+1}^1(x)\|_\infty.\end{aligned}$$

Secondly, we will prove that, for all $\bar{t} \in [x_i, x_{i+1}]$, exists $\xi' \in \,]x_i,\, x_{i+1}[$ such that

$$f(\bar{t}) - \mathrm{P}(\bar{t}) = \frac{f^{(4)}(\xi') - \mathrm{P}^{(4)}(\xi')}{24} \prod_{j=i}^{i+1} (\bar{t} - x_j)^2. \tag{2.1.5}$$

To this end, consider an arbitrary but fixed value $\bar{t}$ different from x_i and x_{i+1}, and define

$$F(t) = f(t) - \mathrm{P}(t) - \alpha \prod_{j=i}^{i+1} (t - x_j)^2,$$

where the constant α is chosen so that $F(\bar{t}) = 0$, i.e.

$$\alpha = \frac{f(\bar{t}) - \mathrm{P}(\bar{t})}{\prod_{j=i}^{i+1} (\bar{t} - x_j)^2}.$$

It is clear that F has at least three roots in the interval $[x_i,\, x_{i+1}]$, which are the knots x_i, x_{i+1} and $\bar{t}$.

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

$$F^{(4)}(\xi') = f^{(4)}(\xi') - \mathrm{P}^{(4)}(\xi') - 24\alpha = 0,$$

from which it is clear that $\alpha = \dfrac{f^{(4)}(\xi') - \mathrm{P}^{(4)}(\xi')}{24}$.
This proves equation (2.1.5) and, as a consequence, that

$$\|f - \mathrm{P}\|_{\infty,[x_i,\, x_{i+1}]} = o(h^4).$$

Finally,

$$\|f - T\| \;\leq\; \|f - \mathrm{P}\| + \|\mathrm{P} - T\| \;=\; o(h^4) + o(h^2)$$

and the claim follows. $\square$

According to Theorem 4.2.3, the T interpolation operator defined by (2.1.1), possesses a quadratic convergence order, i.e. $o(h^2)$. With a view to justifying the theoretical study, the test function

$$f_6(x) := -20\,\exp\left(-0.2\,x\right) \,-\, \exp\left(\cos\left(2\,\pi\,x\right)\right) \,+\, e \,+\, 20$$

defined over the interval $[0,\,2]$ is proposed and Figures 2.5, 2.6 and 2.7 shows its approximation using T interpolation operator. Uniform subdivision of the interval $[0,\,2]$ for this interpolation is adopted with different values of n.

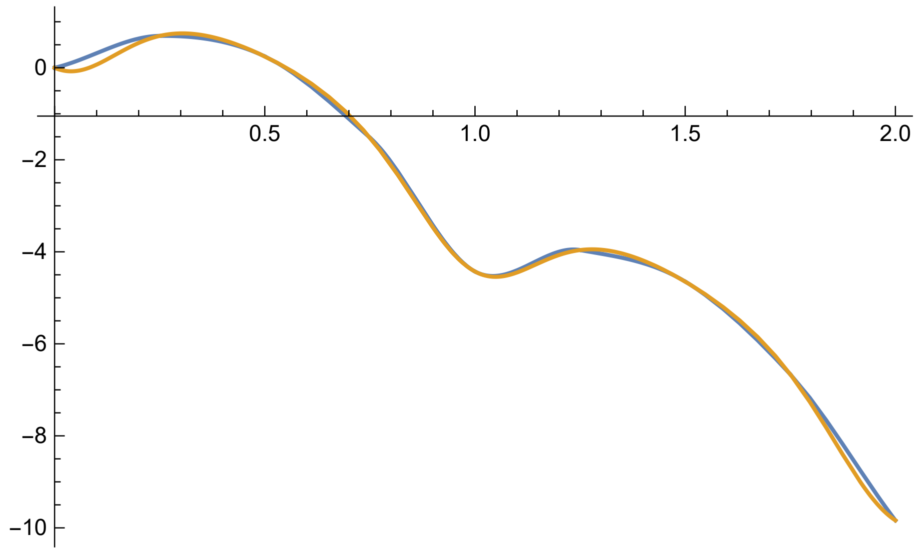


Figure 2.5: Approximation by cubic AT Hermite interpolating splines for f_6 with $n = 8$.

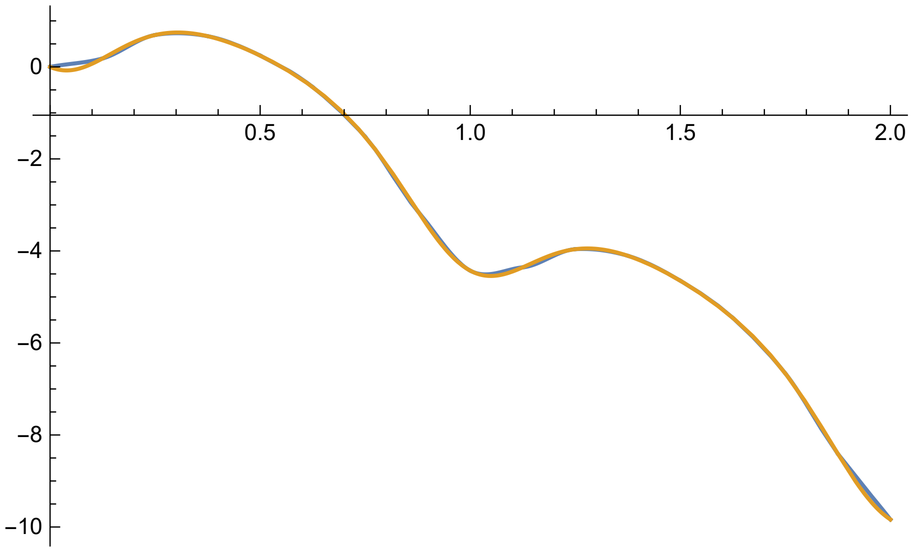


Figure 2.6: Approximation by cubic AT Hermite interpolating splines for f_6 with $n = 16$.

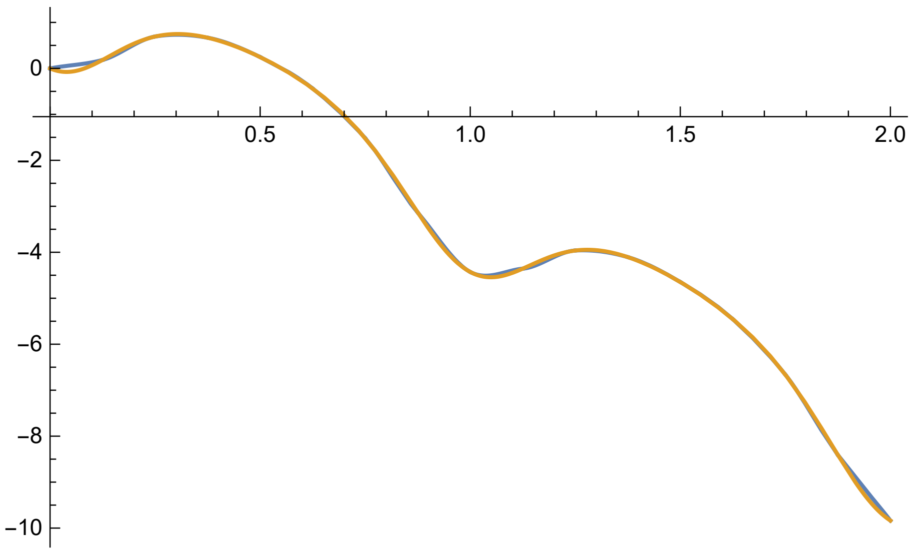


Figure 2.7: Approximation by cubic AT Hermite interpolating splines for f_6 with $n = 32$.

The uniform norm on [0, 2] of the approximation error has been estimated for function f_6 and its algebraic trigonometric interpolant T in the spline space associated with a partition into n equal parts as follows:

$$\mathcal{E}_h\left(f_6,\, T\right) := \max_{0 \leq \ell \leq 2000} \left| f_6\left(t_\ell\right) \,-\, T\left(t_\ell\right) \right|. \tag{2.1.6}$$

where, t_ℓ are equally spaced points in [0, 2]. The approximation error is shown in Table 3.1 along with its Numerical Convergence Order (NCO for short), computed from (2.1.6). The NCOs are computed by the expression

$$\left| \log \left(\frac{\mathcal{E}_{1/n}(f_6, T)}{\mathcal{E}_{1/2n}(f_6, T)} \right) / \log 2 \right| .$$

The numerical results are in good agreement with the theoretical ones.

n	$\mathcal{E}_{1/n}$	NCO
20	9.31×10^{-2}	$---$
40	2.40×10^{-2}	1.95
80	5.99×10^{-3}	2.002
160	1.49×10^{-3}	2.007
320	3.72×10^{-4}	2.001

Table 2.1: The estimated maximum errors for the test function f_6.

2.1.2 Numerical Examples

In this section, we give some numerical tests that illustrate the performance of the above cubic Hermite interpolation operator. For the sake of simplicity and in order to compare our method with some published results [22, 31], we consider the functions

$$F_1(x) = \begin{cases} \ln\left(1 + \left(x - \frac{1}{2}\right)^2\right), & 0 \le x < \frac{1}{2}, \\ \sin(2\pi x), & \frac{1}{2} \le x < 1, \\ (1-x)\exp\left(-\frac{\log(2)}{2}x\right), & 1 \le x < 2, \\ (3-x)\left(x - \frac{5}{2}\right), & 2 \le x \le 3, \end{cases}$$

and

$$F_2(x) = \cos(2\pi \sin x).$$

For them, in the first case we consider the abscissae $x_i = i\,h$, with $h = 3/n$ and $n = 15, 25, 35$, and $h = 1/n$ with $n = 10, 20, 40$, for F_2. The approximants to F_1 and F_2 are produced using the operator T, where the functional values of the both functions are given data, while the derivatives values must be calculated by the minimization method described in Section 2.1.1. Figures 2.8, 2.9 and 2.10 shows the plot of F_1 (in blue color) and those of the cubic AT Hermite interpolating splines computed by minimizing the objective function I_1. In Figures 2.11, 2.12 and 2.13 the corresponding interpolants to F_2 are shown. From these examples, we can clearly see that the cubic AT spline interpolation method presented here provides a satisfactory shape of the interpolated curves.

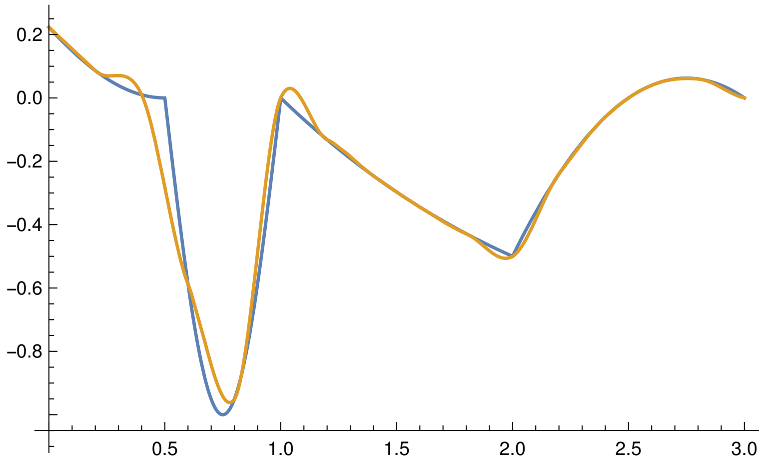


Figure 2.8: Approximation by cubic AT Hermite interpolating splines for F_1 with $n = 15$.

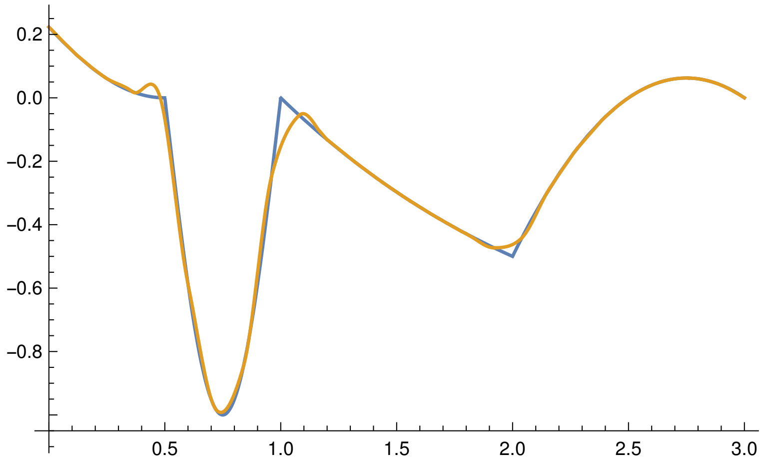


Figure 2.9: Approximation by cubic AT Hermite interpolating splines for F_1 with $n = 25$.

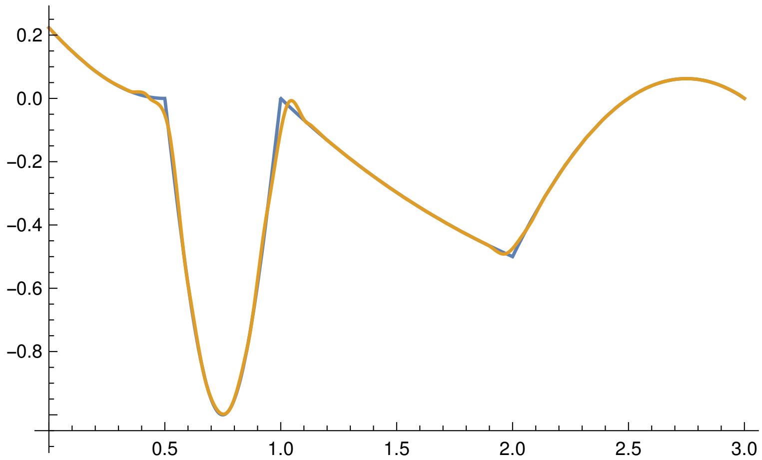


Figure 2.10: Approximation by cubic AT Hermite interpolating splines for F_1 with $n = 35$.

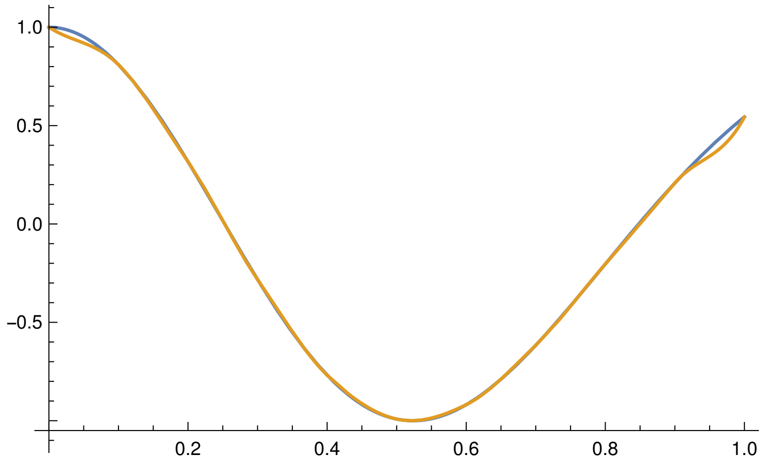


Figure 2.11: Approximation by cubic AT Hermite interpolating splines for F_2 with $n = 10$.

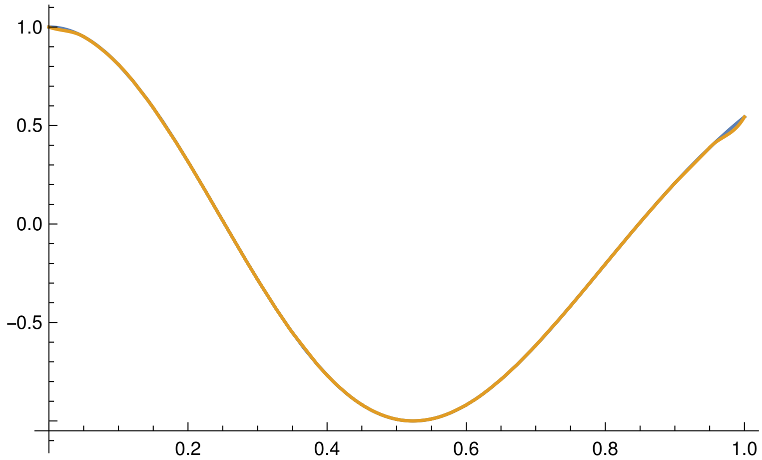


Figure 2.12: Approximation by cubic AT Hermite interpolating splines for F_2 with $n = 20$.

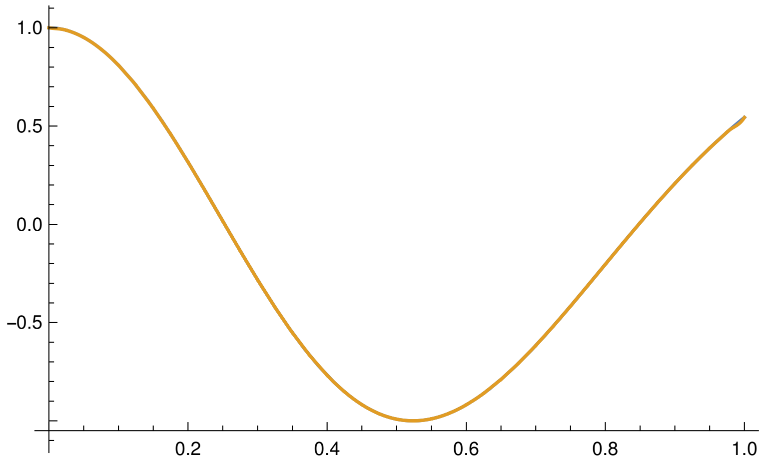


Figure 2.13: Approximation by cubic AT Hermite interpolating splines for F_2 with $n = 40$.

2.2 Many-Knot Algebraic Trigonometric Splines and Shape-Preserving Interpolation

In [35] it is proved that to add a knot into each interval is necessary for $\mathcal{C}^1$ quadratic spline interpolation. In [29] a method to construct a $\mathcal{C}^2$ cubic spline by adding two knots in each interval has been proposed. Many other schemes appear in literature dealing with shape preserving interpolatio (see e.g. [20, 22] and references therein).

In the previous chapter a method to construct cubic algebraic trigonometric Hermite splines has been proposed. The restriction of the interpolant T to each sub-interval $[x_i, x_{i+1}]$ induced by the partition is expressed from the interpolated values, f_i^0 and f_{i+1}^0, and the derivatives f_i^1 and f_{i+1}^1 at the boundary points x_i and x_{i+1}, respectively, and they are calculated by minimizing the objective function $J_1\left(f^1\right)$. In order to reduce the computational cost, in this section the main goal is to reduce the order of the shape-preserving interpolant and to avoid the solution of any systems of equations.
The same strategy as in [35] is adopted here to locally define a $\mathcal{C}^1$ quadratic algebraic trigonometric interpolant. This section is devoted to the numerical solution of the following Hermite interpolation problem:

Proposition 2.2.1. *Find a quadratic spline* $\mathcal{S}_{|[x_i, x_{i+1}]} \in \Gamma_3 = \operatorname{span}\{1, \sin x, \cos x\}$ *with the fewest number of break points such that*

$$\mathcal{S}\left(x_j\right) = f_j^0, \quad \mathcal{S}'\left(x_j\right) = f_j^1, \quad j = i, i+1.$$

This problem has always a solutions. The number of break points depends only on the values f_i^1, f_{i+1}^1 and the divided difference $\delta_i := \dfrac{f_{i+1}^0 - f_i^0}{\tan \frac{h}{2}}$.
We distinguish two cases.

In the first one, we suppose that $f_i^1 + f_{i+1}^1 = \delta_i$, and define $\mathcal{S}$ on $[x_i,\ x_{i+1}]$ as

$$\mathcal{S}\left(x\right) = \bar{\omega}_i^0\left(x\right) f_i^0 + \bar{\omega}_{i+1}^0\left(x\right) f_{i+1}^0 + \bar{\omega}_i^1\left(x\right) f_i^1 + \bar{\omega}_{i+1}^1\left(x\right) f_{i+1}^1,$$

where the functions $\bar{\omega}_{i+k}^j$, follows from the basis functios ω_{i+k}^j, $k = 0, 1,\ j = 0, 1$, by suppressing the term x, i.e.

$$\mathcal{S}\left(x\right) = \frac{1}{2}\csc^2\left(\frac{h}{2}\right)\left(f_{i+1}^1\left(\sin(h) - \sin\left(x - x_i\right) + \sin\left(x - x_{i+1}\right)\right) - \cos(h) f_{i+1}^0 + \left(f_{i+1}^0 - f_i^0\right)\cos\left(x - x_{i+1}\right) + f_i^0\right). \tag{2.2.1}$$

In the second case, $f_i^1 + f_{i+1}^1 \neq \delta_i$, and $\mathcal{S}$ must be defined as a piecewise function on $[x_i, x_{i+1}]$. After choosing a break point $\tau_i \in (x_i, x_{i+1})$, define

$$\begin{cases} \mathcal{S}(x) = A_1\left(\frac{1-\cos(x-x_i)}{\cos(x_i-\tau_i)-1}+1\right) - \frac{B_1\left(2\sin\left(\frac{1}{2}(x-x_i)\right)\sin\left(\frac{1}{2}(x-\tau_i)\right)\right)}{\sin\left(\frac{1}{2}(x_i-\tau_i)\right)} + \frac{C_1(\cos(x-x_i)-1)}{\cos(x_i-\tau_i)-1}; x_i \leq x < \tau_i \\ \mathcal{S}(x) = \frac{A_2(-\cos(x-x_{i+1})-1)}{\cos(\xi_i-x_{i+1})-1} - \frac{B_2\left(2\sin\left(\frac{1}{2}(x-x_{i+1})\right)\sin\left(\frac{1}{2}(x-\tau_i)\right)\right)}{\sin\left(\frac{1}{2}(\tau_i-x_{i+1})\right)} + \frac{C_2(\cos(x-\tau_i)-1)}{\cos(\tau_i-x_{i+1})-1}; \tau_i \leq x \leq x_{i+1} \end{cases} \tag{2.2.2}$$

where
$A_1 = f_i^0$,
$B_1 = -f_i^1$,
$C_1 = \dfrac{f_{i+1}^1 - f_i^1 - \cot\left(\frac{\alpha}{2}\right) f_i^0 - \cot\left(\frac{\beta}{2}\right) f_{i+1}^0}{\cot\left(\frac{\alpha}{2}\right) + \cot\left(\frac{\beta}{2}\right)}$,
$A_2 = \dfrac{f_{i+1}^1 - f_i^1 - \cot\left(\frac{\alpha}{2}\right) f_i^0 - \cot\left(\frac{\beta}{2}\right) f_{i+1}^0}{\cot\left(\frac{\alpha}{2}\right) + \cot\left(\frac{\beta}{2}\right)}$,
$B_2 = f_{i+1}^1 - \cot\left(\dfrac{\beta}{2}\right) f_{i+1}^0$,
$C_2 = f_{i+1}^0$,
$\alpha = \tau_i - x_i$,
$\beta = x_{i+1} - \tau_i$.

We now discuss the shape of the quadratic interpolating function $\mathcal{S}$ defined in both cases.

Lemma 2.2.1. *If $f_i^1 f_{i+1}^1 \geq 0$, $f_i^1 + f_{i+1}^1 = \delta_i$ and $h \leq \pi$, then the quadratic spline $\mathcal{S}$ defined in (2.2.1) is monotone on $[x_i, x_{i+1}]$.*

Proof. If $f_i^1 + f_{i+1}^1 = \delta_i$, then for $x \in [x_i, x_{i+1}]$ it holds

$$S'(x) = \frac{f_{i+1}^1 \sin(x-x_i) + f_i^1 \sin(x_{i+1}-x)}{\sin(x_{i+1}-x_i)}.$$

Therefore, if $f_i^1 f_{i+1}^1 \geq 0$ and $h \leq \pi$, then $\mathcal{S}$ is monotone.

□

Lemma 2.2.2. *If $f_i^1 + f_{i+1}^1 \neq \delta_i$ and $h \leq \pi$, then $\mathcal{S}$ is monotone on $[x_i, \tau_i]$ and $[\tau_i, x_{i+1}]$ if f_i^1, f_{i+1}^1 and $\mathcal{S}'(\tau_i)$ have the same sign .*

Proof. If $f_i^1 + f_{i+1}^1 \neq \delta_i$ and $h \leq \pi$, then for $x \in [x_i, \tau_i]$ we have

$$\mathcal{S}'_{|[x_i,\tau_i]}(x) = \frac{f_i^1(\cos(x-\tau_i) - \cos(x-x_i)) + f_i^0 \sin(x-x_i)}{\cos(x_i-\tau_i)-1}.$$

Moreover,

$$\psi_i := \mathcal{S}'(\tau_i) = \frac{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}{\cot\frac{\alpha}{2}+\cot\frac{\beta}{2}}\left(f_{i+1}^0 - f_i^0 - \frac{f_{i+1}^1\cot\frac{\alpha}{2} + f_i^1\cot\frac{\beta}{2}}{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}\right).$$

By using a computer algebra system, like Mathematica, we get

$$\mathcal{S}'_{|[x_i,\tau_i]}(x) = \frac{\tan\left(\frac{1}{2}(x_i-\tau_i)\right)\left(f_i^1 \sin(\tau_i - x) + \psi_i \sin(x-x_i)\right)}{\cos(x_i-\tau_i)-1}.$$

Therefore the sign of $\left(f_i^1 \sin(\tau_i - x) + \psi_i \sin(x-x_i)\right)$ determines the one of $\mathcal{S}'_{|[x_i,\tau_i]}(x)$.
Since $h \le \pi$, then, $\sin(\tau_i - x)$, $\sin(x-x_i) \ge 0$.
Thus, if $f_i^1\psi_i \ge 0$ (i.e. f_i^1 and ψ_i have the same sign), then $\mathcal{S}$ is monotone on $[x_i,\tau_i]$.
Analogously, if $f_{i+1}^1\psi_i \ge 0$, then S is monotone on $[\tau_i, x_{i+1}]$.
The proof is complete.

□

Now, the question is to choose τ_i to preserve the monotony of $\mathcal{S}$ on $[x_i, x_{i+1}]$. The answer to this question is given in the following result.

Proposition 2.2.2. *If $f_i^1 + f_{i+1}^1 = \delta_i$ and $h \le \pi$, then $\mathcal{S}$ is monotone on $[x_i, x_{i+1}]$ if f_i^1 and f_{i+1}^1 have the same sign.*
If $f_i^1 + f_{i+1}^1 \ne \delta_i$, $h \le \pi$ and f_i^1, f_{i+1}^1, $\mathcal{S}'(\tau_i)$, δ_i have the same sign, then $\mathcal{S}$ is monotone if

$$\min\left(\left|f_i^1\right|, \left|f_{i+1}^1\right|\right) \le \frac{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}{\cot\frac{\alpha}{2}+\cot\frac{\beta}{2}}\left(f_{i+1}^0 - f_i^0\right)$$

and the breakpoint τi is chosen arbitrarily in $[x_i, x_{i+1}]$ if $\left|f_i^1\right| = \left|f_{i+1}^1\right|$, and otherwise as follows:

$$\begin{cases} \max(x_i, z_{i,0}) \le \tau_i \le \min(x_{i+1}, z_{i,0}), & \text{if } \left|f_i^1\right| < \left|f_{i+1}^1\right|, \\ \max(x_i, z_{i,1}) \le \tau_i \le \min(x_{i+1}, z_{i,1}), & \text{if } \left|f_i^1\right| > \left|f_{i+1}^1\right|. \end{cases}$$

The points $z_{i,0}$ and $z_{i,1}$ have the expressions

$$z_{i,0} = -\arccos\left(\cos\left(\frac{h}{2}\right)\left(2\frac{f_i^1}{\delta_i}-1\right)\right) + \frac{x_i + x_{i+1}}{2},$$

and,

$$z_{i,1} = -\arccos\left(\cos\left(\frac{h}{2}\right)\left(2\frac{f_{i+1}^1}{\delta_i}-1\right)\right) + \frac{x_i + x_{i+1}}{2}.$$

Proof. Without loss of generality, we assume that f_i^1, f_{i+1}^1, ψ_i and δ_i are non-negative. Then, from the fact that $\cot\dfrac{\alpha}{2}$, $\cot\dfrac{\beta}{2} \ge 0$, we have

$$\psi_i \ge 0 \text{ if and only if } f_{i+1}^0 - f_i^0 \ge \min(f_i^1, f_{i+1}^1)\frac{\cot\frac{\alpha}{2}+\cot\frac{\beta}{2}}{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}. \tag{2.2.3}$$

On the other hand, we must have

$$x_i \leq \tau_i \leq x_{i+1}. \tag{2.2.4}$$

Both conditions cannot be valid unless the sizes of the derivatives f_i^1, and f_{i+1}^1, are restricted.

- Firstly, if $f_i^1 = f_{i+1}^1$, then it can be seen that (2.2.3) is valid of all τ_i satisfying (2.2.4) if and only if

$$f_{i+1}^0 - f_i^0 \geq f_i^1 \frac{\cot\frac{\alpha}{2} + \cot\frac{\beta}{2}}{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}.$$

- If $f_i^1 < f_{i+1}^1$, then

$$\begin{aligned}
f_i^1 &\leq (f_{i+1}^0 - f_i^0)\frac{\cot\frac{\alpha}{2}\cot\frac{\beta}{2}}{\cot\frac{\alpha}{2} + \cot\frac{\beta}{2}},\\
f_i^1 &\leq (f_{i+1}^0 - f_i^0)\frac{\cos\frac{\alpha}{2}\cos\frac{\beta}{2}}{\sin\frac{h}{2}},\\
f_i^1 &\leq \frac{f_{i+1}^0 - f_i^0}{\tan\frac{h}{2}}\,\frac{\cos\frac{\alpha}{2}\cos\frac{\beta}{2}}{\cos\frac{h}{2}},\\
\frac{f_i^1}{\delta_i} &\leq \frac{\cos\frac{\alpha}{2}\cos\frac{\beta}{2}}{\cos\frac{h}{2}}\\
\frac{f_i^1}{\delta_i} &\leq \frac{1}{2}\left(\sec\left(\frac{h}{2}\right)\cos\left(\frac{1}{2}\left(-2\tau_i + x_i + x_{i+1}\right)\right) + 1\right)\\
\cos\left(\frac{h}{2}\right)\left(2\frac{f_i^1}{\delta_i} - 1\right) &\leq \cos\left(\frac{1}{2}\left(-2\tau_i + x_i + x_{i+1}\right)\right)\\
\tau_i &\leq -\arccos\left(\cos\left(\frac{h}{2}\right)\left(2\frac{f_i^1}{\delta_i} - 1\right)\right) + \frac{x_i + x_{i+1}}{2} = z_{i,0}.
\end{aligned}$$

  Note that $\delta_i \geq 0$ from the fact that $f_i^1 \neq f_{i+1}^1$.

- When $f_{i+1}^1 < f_i^1$, then

$$\tau_i \leq -\arccos\left(\cos\left(\frac{h}{2}\right)\left(2\frac{f_{i+1}^1}{\delta_i} - 1\right)\right) + \frac{x_i + x_{i+1}}{2} = z_{i,1}.$$

The proof is complete.

□

2.2.1 Numerical Example

In order to show the performance of the multi-knot method based on AT splines, we approximate F_1 by means of many-knot splines using two different points τ_i. The results appear in Figures 2.14, 2.15 and 2.16. When τ_i is chosen by the procedure given in Proposition 2.2.2 the obtained interpolating spline preserves the monotonicity of the given data.

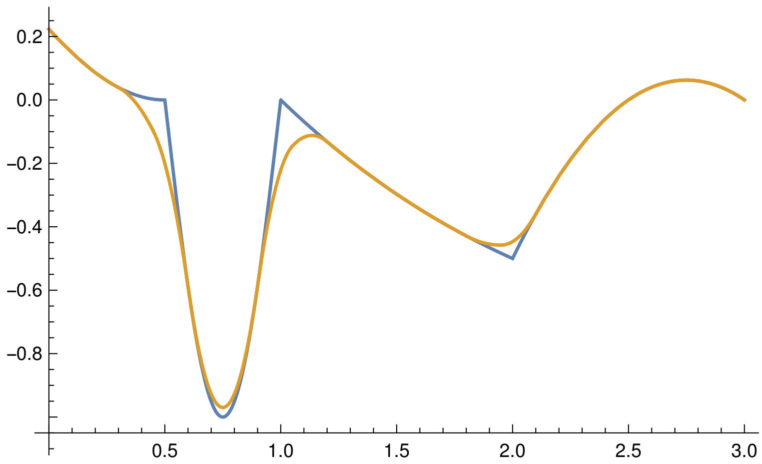


Figure 2.14: Approximation by interpolating many-knot splines of F_1 with τ_i satisfying the conditions of Proposition 2.2.2 and $n = 10$.

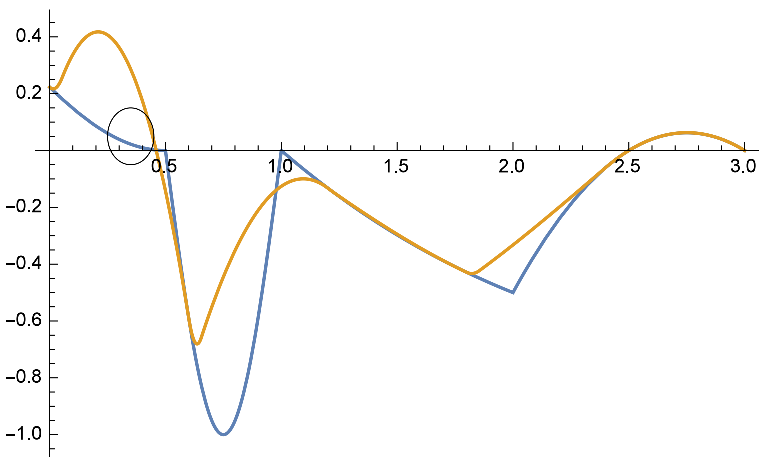


Figure 2.15: Approximation by interpolating many-knot splines of F_1 with τ_i not satisfying the conditions of Proposition 2.2.2.

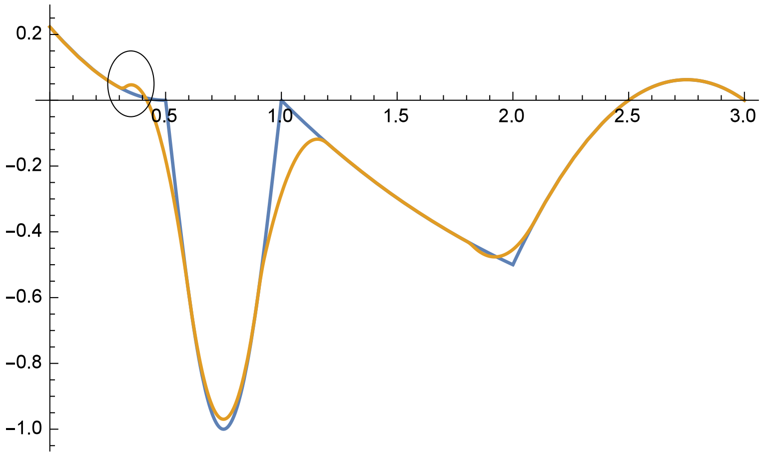


Figure 2.16: Approximation by interpolating many-knot splines of F_1 with τ_i not satisfying the conditions of Proposition 2.2.2 and $n = 10$.

2.3 Conclusion

This chapter is divided into two parts. In the first part, we have defined an optimal AT-interpolant which is exact on the linear space spanned by $\{1, x, \sin x, \cos x\}$. The second part deals with the construction of an AT-interpolant that preserves the monotonicity of the data. The unknowns of the interpolant in the first case are obtained by minimizing an integral expression measuring the squared value of the difference between the first derivative of the interpolation error. In the second one, additional knots are considered in each interval induced by the partition in a specific way to achieve the goal. The good performance of both methods has been illustrated for two functions defined on an interval decomposed into different numbers of parts.

Chapter 3

Algebraic Hyperbolic C^1 Hermite Spline Interpolation

Contents

In this chapter, an optimum Hermite cubic interpolation technique based on Algebraic Hyperbolic (AH) functions 1, x, $sinhx$ and $coshx$ is presented. The result AH-spline is locally defined on each subinterval of the subdivision and interpolate both function and first derivative values of the unknown function at each knots. In most practical situations of interpolation, the first derivative values are not available, a strategy centered on optimizing the derivatives at the nodes is used to estimate the first derivative values, the new spline interpolant aim to approximate the monotonicity shape of the data points. The constructed interpolation scheme is applied to a data fitting situation that come from a second order system.

The results obtained in this chapter is presented in paper [3].

3.1 Cubic Algebraic Hyperbolic (AH) Hermite Splines of Class C^1

Let $\Delta_n = \{x_i := a + ih\}_{i=0}^{n}$, be a uniform partition of a bounded interval $I = [a, b]$, with $h = \dfrac{b-a}{n}$. The construction of a C^1 cubic AH-splines interpolant on the partition Δ_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 Δ_n with global C^1 continuity is denoted as

$$S_4^1(\Delta_n) := \left\{H \in C^1(I)\,; H_{|\ell_i} \in \Gamma_4' \text{ for all } i = 0,\dots,n\right\},$$

where $\Gamma_4' := span\,\{1, x, \sinh x, \cosh x\}$ 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 exist a unique spline $H(x) \in S_4^1(\Delta_n)$ such that

$$H^{(j)}(x_i) = f_i^j, \quad i = 0,\dots,n,\ j = 0,1, \tag{3.1.1}$$

for any given set of f_i^j-values.
Therefore, each spline $H(x) \in S_4^1(\Delta_n)$ restricted to the sub-interval ℓ_i can be represented as follows.

$$H_{|\ell_i}(x) := \sum_{k=0}^{1}\sum_{j=0}^{1} f_{i+k}^{j}\phi_{i+k}^{j}(x), \tag{3.1.2}$$

in which ϕ_{i+k}^j, $k = 0,1$, $j = 0,1$, are a classical Hermite basis functions of $S_4^1(\Delta_n)$ 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 (3.1.1) in $H(x) \in S_4^1(\Delta_n)$, where

$$\begin{aligned}
&\left(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1\right) = (1,0,0,0)\,, \left(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1\right) = (0,1,0,0)\,,\\
&\left(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1\right) = (0,0,1,0)\,, \left(f_i^0, f_i^1, f_{i+1}^0, f_{i+1}^1\right) = (0,0,0,1)\,,
\end{aligned}$$

respectively. More precisely,

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

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 3.1 shows the graphical representation of the basis $\phi_{i,j},\ j = 1, \ldots, 4$ on the interval $[0, 1]$.

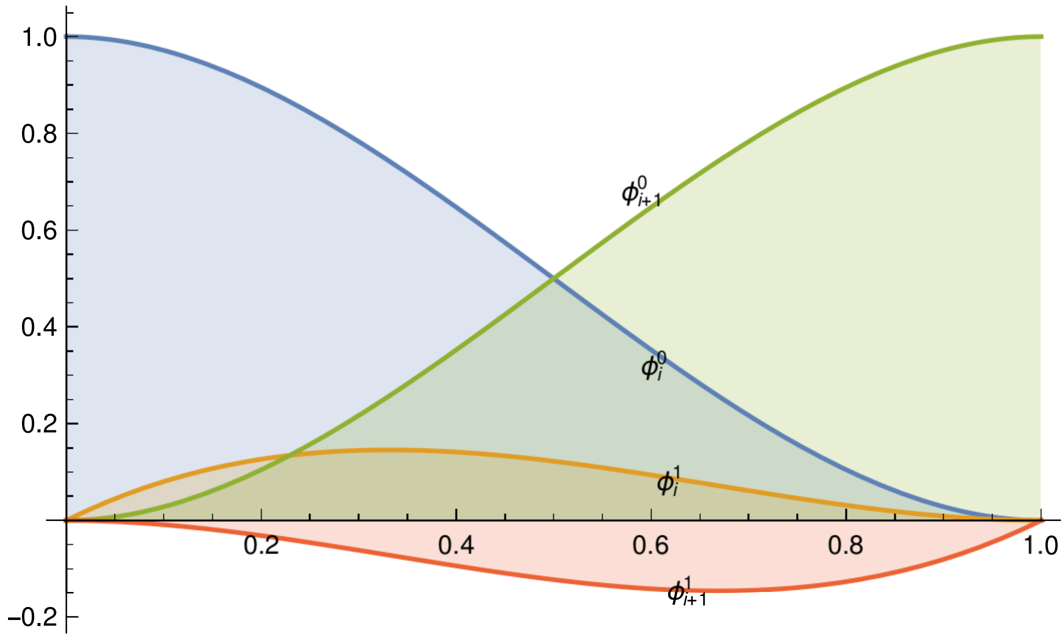


Figure 3.1: Plot of the basis $\phi_{i,j},\ j = 1, \ldots, 4$ on the interval $[0, 1]$.

Remark 2. The constructed interpolant (3.1.2) exactly reproduces all functions in space $\Gamma_4' := span\,\{1,\ x,\ \sinh x,\ \cosh x\}$.

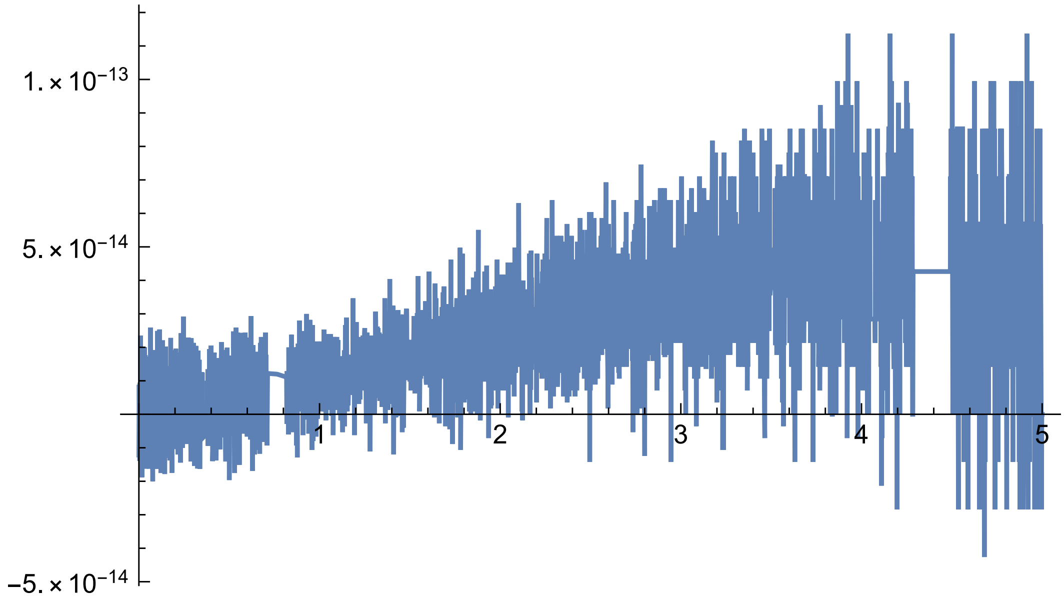


Figure 3.3: Plot of the error function $H(x) - f_7(x)$ on the interval $[0, 5]$.

To illustrate Remark 2, we approximate the function

$$f_7(x) = 2\sinh(x) - \frac{\cosh(x)}{5}$$

on the interval $[0, 5]$ using the operators H, T and p defined by (3.1.2), (2.1.1) and (1.3.10) respectively. Figures 3.2, 3.5 and 3.4 shows this approximation and Figure 3.3 represents the function error $H(x) - f_7(x)$.

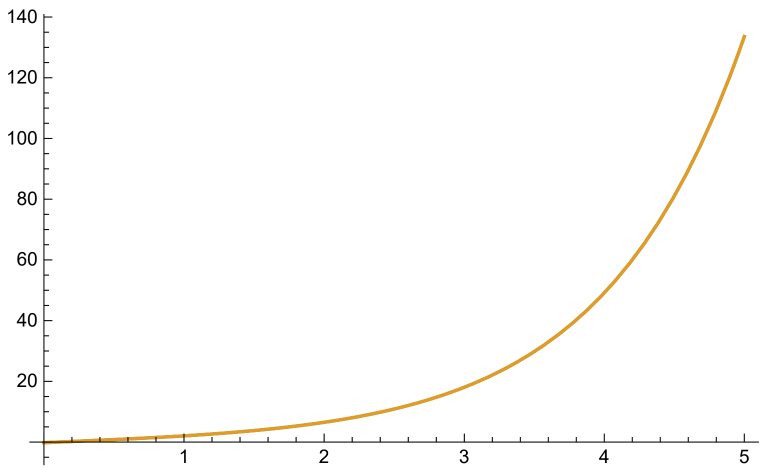


Figure 3.2: Approximation by cubic AH Hermite interpolating splines for f_7 on the interval $[0, 5]$ with $n = 1$.

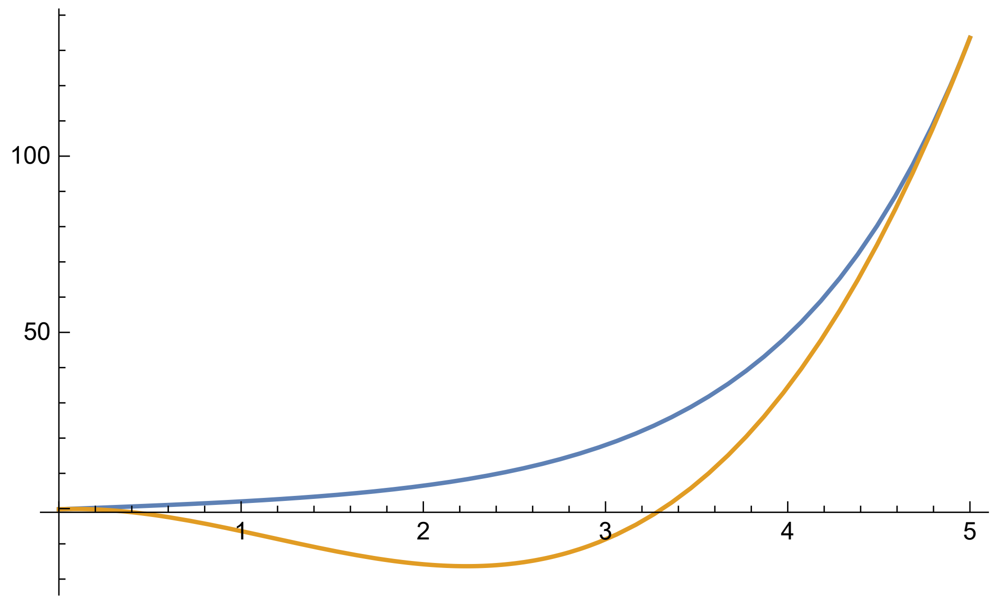


Figure 3.4: Approximation by cubic polynomial Hermite interpolating splines for f_7 on the interval $[0, 5]$ with $n = 1$.

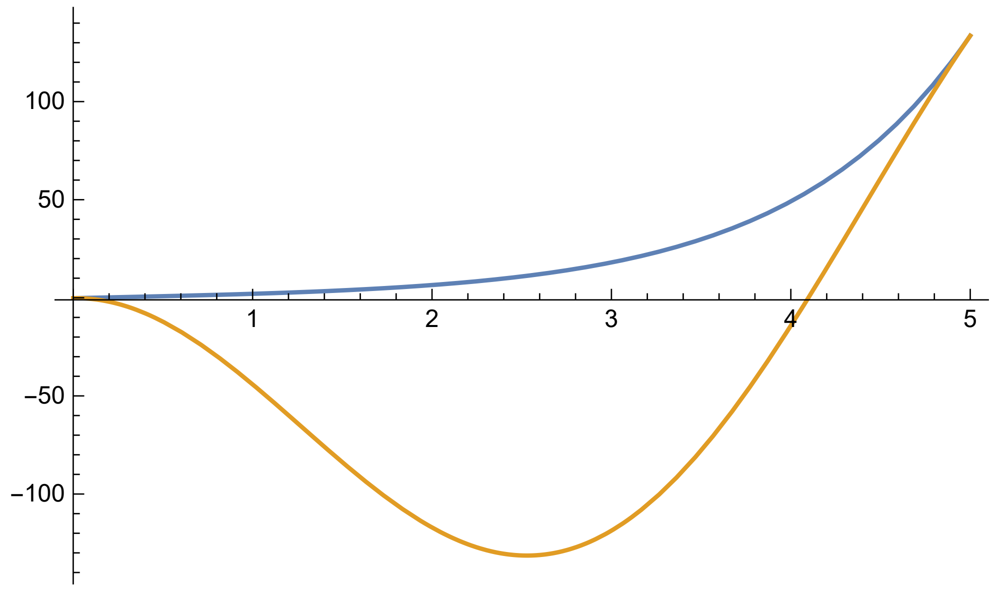


Figure 3.5: Approximation by cubic AT interpolating splines for f_7 on the interval $[0, 5]$ with $n = 1$.

3.1.1 Slopes Estimation

The proposed interpolant H is defined from the values and first derivative values at the break-points. In general the values f_i^0, $i = 1, .., n$, are given but the values f_i^1, $i = 1, .., n$ of the first derivatives of H remain to be determined.

In order to estimate the values of the derivatives f_i^1 we consider the following functional:

$$J_1\left(f_1^1, \ldots, f_n^1\right) := \int_a^b \left(H'(x) - L'(x)\right)^2 dx \tag{3.1.3}$$

$$= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \left(H_i'(x) - L_i'(x)\right)^2 dx, \tag{3.1.4}$$

where L is the linear interpolating spline with pieces

$$L_i(x) := \frac{x_{i+1} - x}{x_{i+1} - x_i} f_i^0 + \frac{x - x_i}{x_{i+1} - x_i} f_{i+1}^0.$$

The minimization of (3.1.4) yields the following tridiagonal system of linear equations:

$$\begin{pmatrix} \mu & 2\nu & & & & \\ \nu & \mu & \nu & & & \\ & \nu & \mu & \nu & & \\ & & \ddots & \mu & \ddots & \\ & & & \nu & \mu & \nu \\ & & & & 2\nu & \mu \end{pmatrix} \begin{pmatrix} f_1^1 \\ \\ \vdots \\ \\ f_n^1 \end{pmatrix} = \eta \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{pmatrix}, \tag{3.1.5}$$

with

$$\nu := \frac{h^4(\operatorname{csch}^4(\frac{h}{2})((-3h^2+2)\sinh(h) + (h^2+6)h\cosh(-h) - 6h - \sinh(2h)))}{4(-h\coth(\frac{h}{2})+2)^2}$$

$$\mu := \frac{h^4(\operatorname{csch}^4(\frac{h}{2})((h^2+2)\sinh(2h) + 4(h^2-1)\sinh(h) + 2(-h)(h^2+2\cosh(2h) - 2\cosh(h))))}{4(-h\coth(\frac{h}{2})+2)^2}$$

$$\eta := \frac{2e^h h^4(h^2 + h\sinh(h) - 4\cosh(h) + 4)}{(e^h(h-2)+h+2)^2}$$

$d_1 := 2(f_2^0 - f_1^0),\ d_i := f_{i+1}^0 - f_{i-1}^0,\ i = 2, \ldots, n-1$ *and* $d_n := 2(f_n^0 - f_{n-1}^0)$.

Theorem 3.1.1. *For $h > 0$ the matrix system in (3.1.5) is a diagonally dominant matrix and the limit of the index D_h when h goes to zero is equal to $\frac{1}{4}$.*

Proof. The proof of the above Theorem 3.1.1 is similar to that of 2.1.1 Chapter 2. □

Therefore, the LU factorization procedure is adequate to solve the system (3.1.5) for a small enough value of h.

Theorem 3.1.2. *Let $f \in \mathcal{C}^4([a,b])$ and H be the interpolation operator defined by (3.1.2). For a uniform step size h, we have*

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

Proof. The proof of this Theorem is similar to the proof of Theorem 2.1.3 in Chapter 2. □

3.2 Numerical Examples

This section presents some numerical results to illustrate the efficiency of the Hermite AH-spline interpolation operator. For this purpose, we will use the following test functions defined on $[0,1]$:

$$h_1(x) = \frac{3}{4}e^{-2(9x-2)^2} - \frac{1}{5}e^{-(9x-7)^2-(9x-4)^2} + \frac{1}{2}e^{-(9x-7)^2-\frac{1}{4}(9x-3)^2} + \frac{3}{4}e^{\frac{1}{10}(-9x-1)-\frac{1}{49}(9x+1)^2},$$
$$h_2(x) = \frac{1}{2}x\cos^4\left(4\left(x^2+x-1\right)\right),$$
$$k_1(x) = \frac{\sqrt{x+2}\exp\left(2x^2\right)\sin(4\pi x)}{\left(x^2+3\right)^{5/7}}.$$

The two first functions are the 1D versions of the Franke [87] and Nielson [88] functions. The approximation of the above functions using the AH-spline operator for different values of n are shown in Figures 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14.

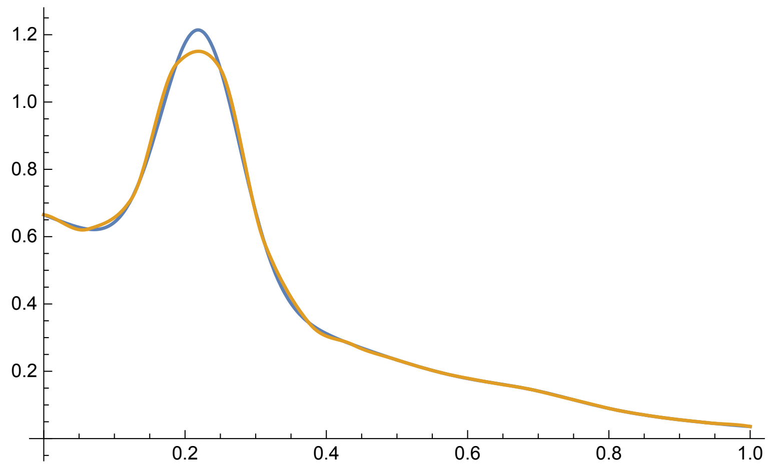


Figure 3.6: AH-spline operator H in orange and h_1 in blue with $n = 16$.

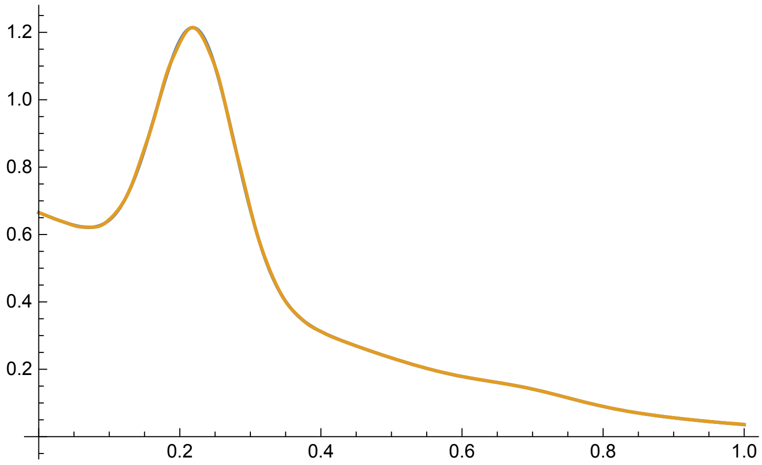


Figure 3.7: AH-spline operator H in orange and h_1 in blue with $n = 32$.

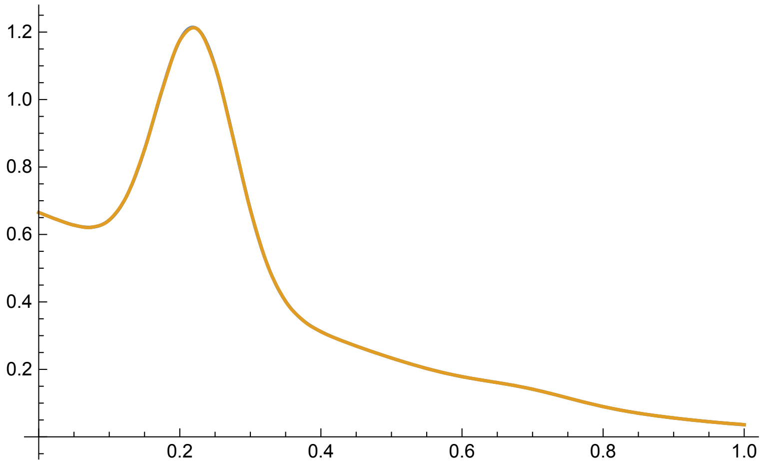


Figure 3.8: AH-spline operator H in orange and h_1 in blue with $n = 40$.

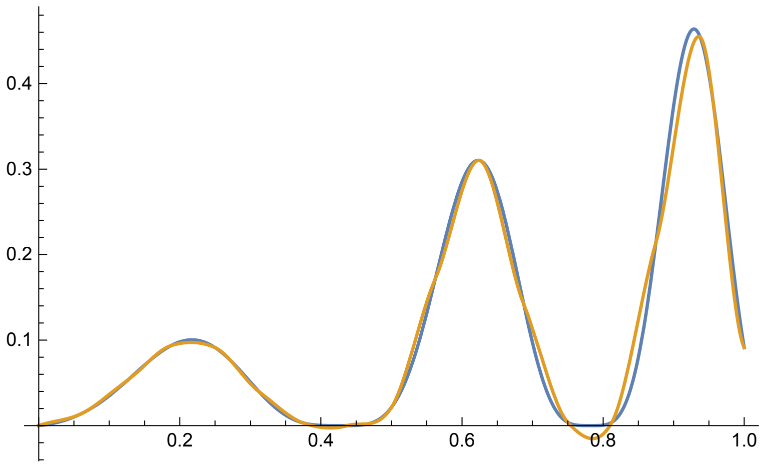


Figure 3.9: AH-spline operator H in orange and h_2 in blue with $n = 16$.

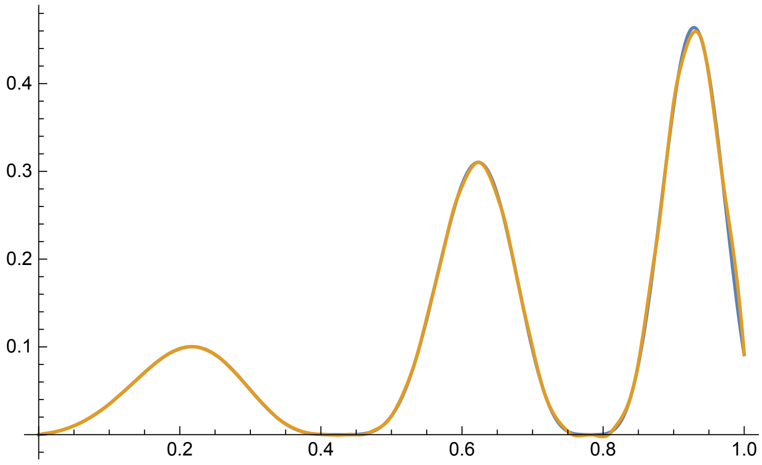


Figure 3.10: AH-spline operator H in orange and h_1 in blue with $n = 32$.

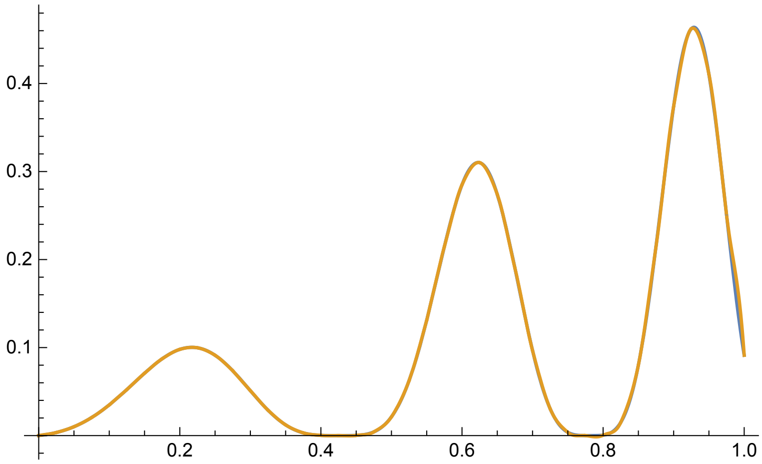


Figure 3.11: AH-spline operator H in orange and h_1 in blue with $n = 40$.

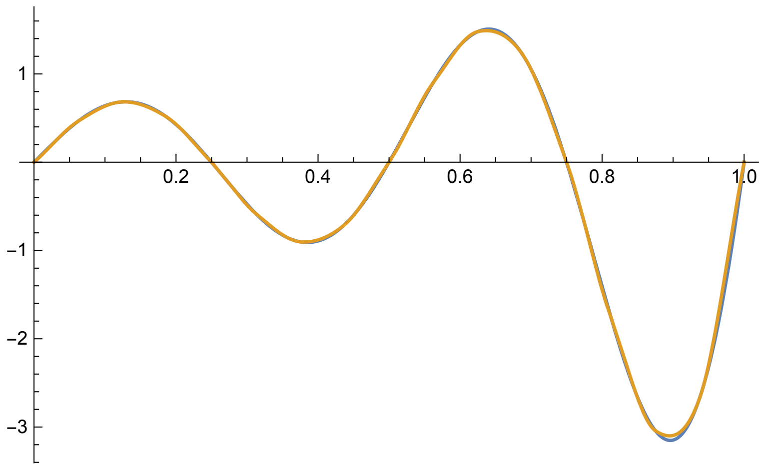


Figure 3.12: AH-spline operator H in orange and k_1 in blue with $n = 16$.

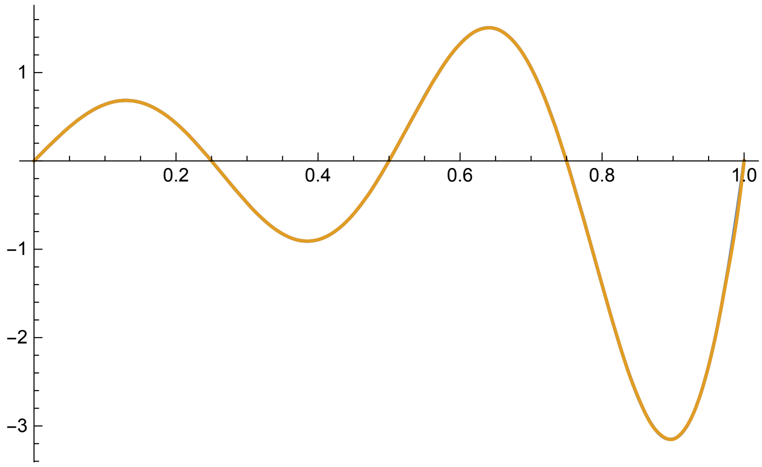


Figure 3.13: AH-spline operator H in orange and k_1 in blue with $n = 32$.

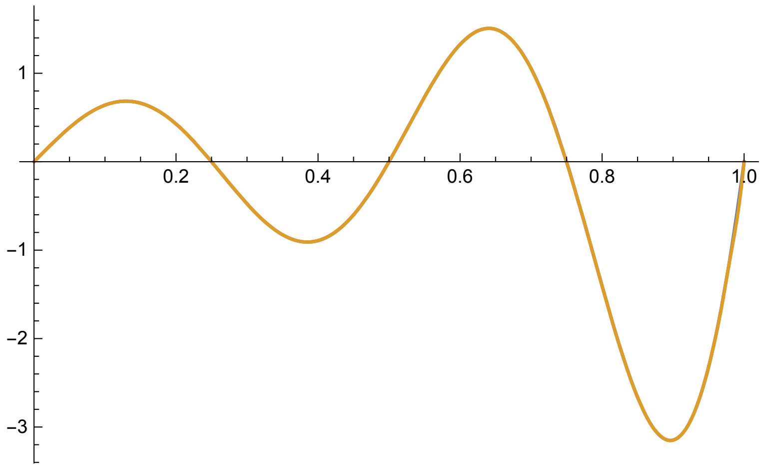


Figure 3.14: AH-spline operator H in orange and k_1 in blue with $n = 40$.

3.3 Data Fitting Application

We'll formalize our discussion of second-order responses in this section, as well as define two specifications for analyzing and designing general second-order systems. In addition, we will discuss the underdamped example and provide quantitative specifications particular to this reaction. Natural frequency and damping ratio are the names given to these numbers. The oscillation frequency of a second-order system without damping is called its natural frequency. The damping ratio, on the other hand, compares the envelope's exponential decay frequency to natural frequency. Modeling of second-order system a system is known as a second-order linear invariant if the response $y(t)$ is related to the excitation $e_0(t)$ by a linear second-order differential equation with constant coefficients:

$$y'' + a_1 y' + a_2 y = a_2 e_0, \tag{3.3.1}$$

in which a_1 and a_2 are two positive or zero real numbers.
Using the Laplace transform and assuming that initial conditions are equal to zero, the transfer function of a second order system is expressed as:

$$H_t(z) = \frac{Y(z)}{E_0(z)} = \frac{a_2}{z^2 + a_1 z + a_2}, \tag{3.3.2}$$

with

$$Y(z) = \int_0^{+\infty} y(t)\exp(t(-z))\,dt,$$

and

$$E_0(z) = \int_0^{+\infty} e_0(t)\exp(t(-z))\,dt,$$

are the Laplace transforms of $y(t)$ and $e_0(t)$ respectively.

The natural frequency is the oscillation frequency without damping. That means $a_1 = 0$. The transfer function becomes:

$$H_t(z) = \frac{Y(z)}{E_0(z)} = \frac{a_2}{z^2 + a_2}.$$

This function has two imaginary poles:

$$z_1 = j\sqrt{a_2};\ z_2 = -j\sqrt{a_2}. \tag{3.3.3}$$

Thus

$$\omega_n = \sqrt{a_2}. \tag{3.3.4}$$

The poles of an under-damped system ($0 < \xi < 1$) are given as:

$$z_1 = \frac{a_1}{2} + j\sqrt{a_2};\ z_2 = \frac{a_1}{2} - j\sqrt{a_2}. \tag{3.3.5}$$

Hence

$$\xi = \frac{Exponential\ decay\ frequency}{natural\ frequency} = \frac{a_1}{2\omega_n}. \tag{3.3.6}$$

It is possible to deduce that

$$a_1 = 2\xi\omega_n. \tag{3.3.7}$$

The transfer function looks like this:

$$H_t(z) = \frac{Y(z)}{E_0(z)} = \frac{\omega_n^2}{z^2 + 2\xi z\omega_n + \omega_n^2}. \tag{3.3.8}$$

3.3.1 Response of an Under-Damped Second-Order System

The step response of an under-damped second-order system is its response $y(t)$ to a unit-step excitation $e_0(t)$ where it is assumed that $(0 < \xi < 1)$.
The response is defined using inverse Laplace transform of $y(t)$:

$$Y(z) = \frac{E_0(z)\omega_n^2}{z^2 + 2\xi z\omega_n + \omega_n^2} = \frac{\omega_n^2}{z\left(z^2 + 2\xi z\omega_n + \omega_n^2\right)}, \tag{3.3.9}$$

with $E_0(z) = \dfrac{1}{z}$.

Taking the inverse Laplace transform, the step response of an under-damped second-order system is given by:

$$y(t) = 1 - \exp(\xi t(-\omega_n))(\sin(\sqrt{1-\xi^2}t\omega_n) + \frac{\xi\cos(\sqrt{1-\xi^2}t\omega_n)}{\sqrt{1-\xi^2}}). \tag{3.3.10}$$

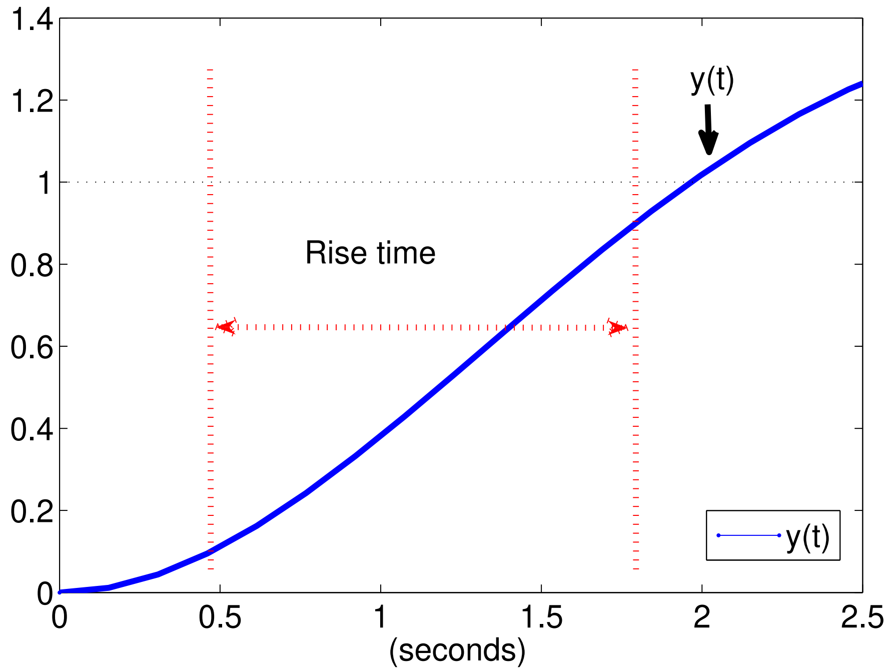


Figure 3.15: Step response characteristic of an underdamped second order system.

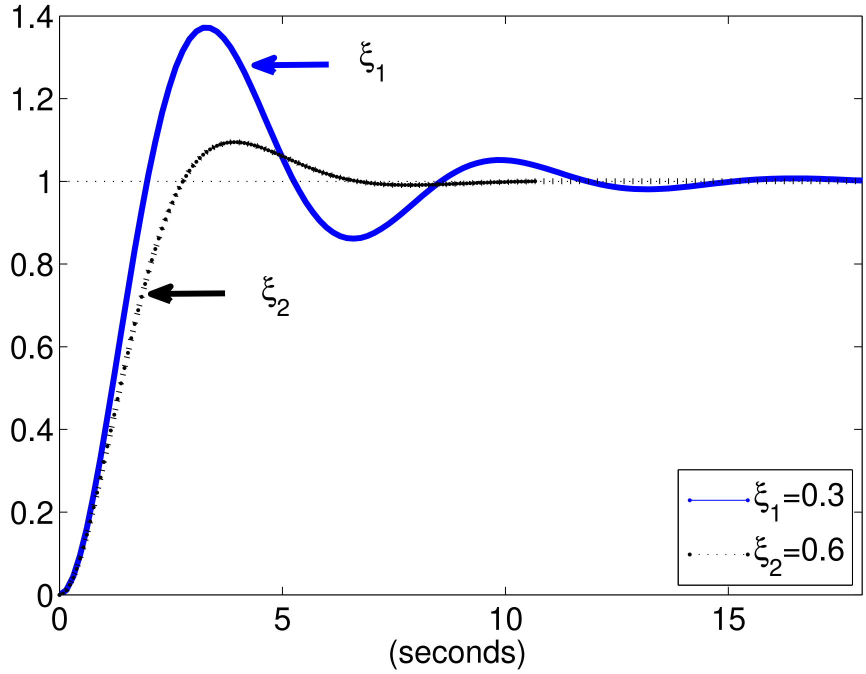


Figure 3.16: Second-order underdamped responses characteristics for damping ratio values.

We have defined quantitative specifications as the natural frequency and the damping ratio. From Figure 3.15, other parameters associated with an underdamped second-order system are rise time. According to Figure 3.16, the rise time depend on the damping ratio.

3.3.2 Rise Time Tr

The time required for the response to rising from 0.1 to 0.9 of its steady value. In theory, there is no method to define the rise time concerning the damping ratio. Therefore, an accurate analytical relationship between rise time and damping ratio, ξ, cannot be determined. However, using a numerical method and the analytical response expression $y(t)$, it is possible to find the relationship between rise time and damping ratio.

In this chapter, AH-spline Hermite interpolating is proposed to establish a precise relationship between rise time and damping ratio.

With the use of the computer, we can resolve the values of $\omega_n t_2$ that result in $y(\omega_n t_2) = 0.9$ and $y(\omega_n t_1) = 0.1$. If we subtract the two values of $\omega_n t_1$ and $\omega_n t_2$, we derive the normalized rise time, $\omega_n Tr = \omega_n t_2 - \omega_n t_1$, for this specific value of ξ.

To show the accuracy of our proposed method, we will present in Table 3.1 the error made by our approach and by the Polyfit function in Matlab for the approximation of Rise time and in Table 3.2 the absolute mean error.

We define the error between a value Tr_i and its approximation H_i by :

$$e_i = |Tr_i - H_i|,$$

and the absolute mean error **E** by

$$\mathbf{E}(n) = \frac{1}{n}\sum_{i=1}^{n} e_i,$$

Table 3.1: Error behavior of AH splines and that of Polyfit function in Matlab.

ξ_i	Tr_i	**Error of AH**	**Error of Polyfit**
0.1	1.121	0	1.95×10^{-2}
0.15	1.178	1.69×10^{-2}	2.5×10^{-2}
0.2	1.205	0	2×10^{-4}
0.25	1.268	4.5×10^{-3}	6.8×10^{-3}
0.3	1.328	0	6.3×10^{-3}
0.35	1.397	6.2×10^{-3}	8.9×10^{-3}
0.4	1.461	0	5.2×10^{-4}
0.45	1.5451	1.5×10^{-3}	1.5×10^{-3}
0.5	1.642	0	6.7×10^{-3}
0.55	1.738	4.7×10^{-3}	1.3×10^{-4}
0.6	1.855	0	1.5×10^{-3}
0.65	1.979	3.2×10^{-3}	3.5×10^{-3}
0.7	2.125	0	1.6×10^{-3}
0.75	2.281	6.5×10^{-3}	6.2×10^{-3}
0.8	2.466	0	5.6×10^{-4}
0.85	2.672	4.7×10^{-3}	9.2×10^{-3}
0.9	2.876	0	4.3×10^{-3}

Table 3.2: The absolute mean error $\mathbf{E}(n)$

AH splines	Polyfit
3.6×10^{-3}	11.9×10^{-3}

Figure 3.17 shows fitting the rise time using AH-spline operator.

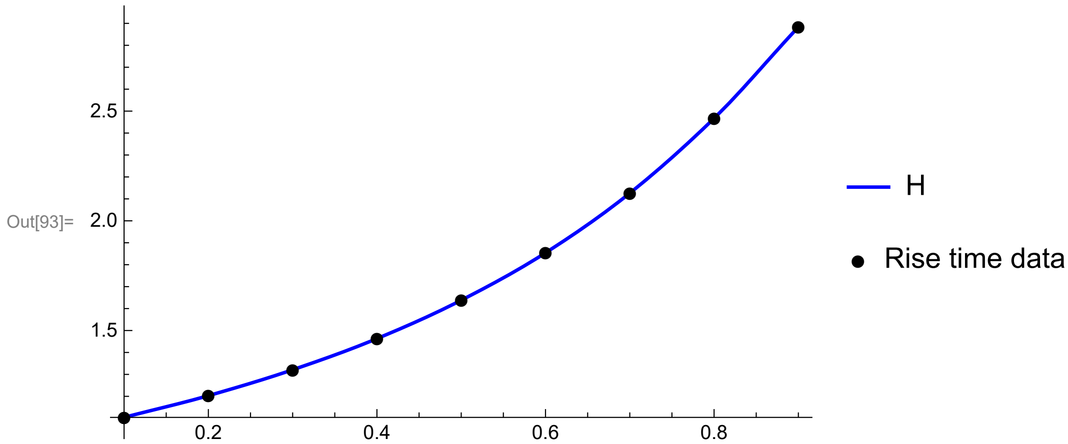


Figure 3.17: Fitting the rise time using AH-spline operator.

3.4 Conclusion

This chapter is divided into two part, in the first part we define and study a C^1 cubic algebraic hyperbolic spline interpolant exactly reproduces all functions in the space spanned by $\{1, x, sinhx, coshx\}$, the unknown parameters of this interpolation are estimated by minimizing an integral expression measuring the squared value of the difference between the first derivative of the interpolation error. In the second part, we consider a second order system problem from physics and apply the AH-spline operator to define and fit the rise time as a function of the damping ratio. The numerical results show that the AH-spline operator has good performances.

Chapter 4

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

Contents

In this chapter, 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.

The results obtained in this chapter are presented in the paper [5].

4.1 Introduction

In this chapter, we consider the algebraic hyperbolic (AH) cubic spline space, spanned by $\{1, x, \sinh, \cosh\}$, and let $\Delta_n = \{x_i := a + ih\}_{i=0}^{n}$ be a uniform partition of a bounded interval $I = [a, b]$, with $h = \dfrac{b-a}{n}$. 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 chapter, 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 [78, 79] 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.

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

Let $\Delta_n = \{x_i := a + ih\}_{i=0}^n$ be a uniform partition of a bounded interval $I = [a, b]$, with $h = \dfrac{b-a}{n}$. The C^1 cubic AH spline interpolant on the partition Δ_n is the unique spline s solution of the Hermite interpolation problem

$$s^{(j)}(x_i) = f_i^j, \quad i = 0, \ldots, n, \; j = 0, 1, \tag{4.2.1}$$

for any given set of f_i^j-values.
The spline function s is defined as follows:

$$s_{|\ell_i}(x) := \sum_{k=0}^{1} \sum_{j=0}^{1} f_{i+k}^j \phi_{i+k}^j(x), \tag{4.2.2}$$

With the functions basis ϕ_{i+k}^j, $k = 0, 1$, $j = 0, 1$ are 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 $\theta = h\sinh(h) - 2\cosh(h) + 2$.

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

$$\|s - f\|_{\infty,[x_i,x_{i+1}]} = o\left(h^4\right). \tag{4.2.3}$$

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,

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

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

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

This proves Equation (4.2.3).

The proposed interpolant s is defined from the values and first derivative values at the breakpoints. Unfortunately, this dataset is not always at hand. This paper deals with cases 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:

$$\alpha f^1_{i-1} + \beta f^1_i + \alpha f^1_{i+1} = f^0_{i+1} - f^0_{i-1}, \qquad i = 1, ..., n-1. \tag{4.2.4}$$

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

The error related to the approximation scheme (4.2.4) can be derived from the following result.

Lemma 4.2.1. *Let $f \in \mathcal{C}^3([a, b])$; then, the local truncation errors $\mathfrak{t}_i$, $i = 1, 2, \ldots, n$, associated with the scheme (4.2.4) are given by the expressions*

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

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

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

Using Equation (4.2.4), it results that

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

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 4.2.1.

Theorem 4.2.1. *Let $f \in \mathcal{C}^3([a,\, b])$, then*

$$\left|f_i^1 - f'(x_i)\right| = o\left(h^2\right).$$

At this point, we have provided a scheme that approximates the derivative values from the function values by imposing C^2 smoothness at the breakpoints. Next, we will deal with cases where the function values are unavailable, whereas the integrals over the sub-intervals are known.

4.2.1 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

$$I_i = \int_{x_{i-1}}^{x_i} f(x)\, dx, \quad i = 1, \ldots, n \tag{4.2.5}$$

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

$$2I_i = h\left(f_{i-1}^0 + f_i^0\right) + \left(2 - h\coth\left(\frac{h}{2}\right)\right)\left(f_i^1 - f_{i-1}^1\right), \quad i = 1, \ldots, n \tag{4.2.6}$$

$$2I_{i+1} = h\left(f_i^0 + f_{i+1}^0\right) + \left(2 - h\coth\left(\frac{h}{2}\right)\right)\left(f_{i+1}^1 - f_i^1\right), \quad i = 0, \ldots, n-1 \tag{4.2.7}$$

respectively.

Subtracting (4.2.6) from (4.2.7) as a first step, and then applying (4.2.4) 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

$$\mu f_{i-1}^1 + \lambda f_i^1 + \mu f_{i+1}^1 = 2\left(I_{i+1} - I_i\right), \quad i = 1, \ldots, n-1, \tag{4.2.8}$$

with

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

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.

$$\begin{pmatrix} \lambda & \mu & & & & \\ \mu & \lambda & \mu & & & \\ & \mu & \lambda & \mu & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu & \lambda & \mu \\ & & & & \mu & \lambda \end{pmatrix} \begin{pmatrix} f_1^1 \\ \\ \vdots \\ \\ f_{n-1}^1 \end{pmatrix} = \begin{pmatrix} b_1 - \mu c \\ b_2 \\ \vdots \\ b_{n-2} \\ b_{n-1} - \mu d \end{pmatrix}, \tag{4.2.9}$$

with $b_i = 2\left(I_{i+1} - I_i\right)$.

The following result shows that the linear system (4.2.9) has a unique solution.

Theorem 4.2.2. *For $h > 0$, the matrix system in (4.2.9) 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:

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

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

The LU factorization is adequate for solving (4.2.9) because the index on the diagonally dominant property of the matrix A is equal to $\dfrac{1}{5}$ 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 (4.2.9). The index on the diagonally dominant property of matrix A is given by

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

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

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 (4.2.6). 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 (4.2.6) 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 4.2.2. *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 (4.2.9), are given by the expressions*

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

Theorem 4.2.3. *Let $f \in \mathcal{C}^4([a, b])$ and s be the interpolation operator defined by (4.2.2), (4.2.6) and (4.2.9). For a uniform step size h, we have*

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

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

$$\begin{aligned}\|s - f\| &= \|s - S + S - f\| \\ &\leq \|s - S\| + \|S - f\| \\ &\leq o\left(h^2\right) + o\left(h^4\right) \\ &\equiv o\left(h^2\right).\end{aligned}$$

This due to the fact that $\|s - S\| \leq \max\left\{|f'(x_i) - f_i^1|, |f'(x_{i+1}) - f_{i+1}^1|\right\}$, which concludes the proof. $\square$

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

$$f(a) = \frac{11I_1 - 7I_2 + 2I_3}{6h}, \quad f'(a) = -\frac{2I_1 - 3I_2 + I_3}{h^2}, \quad f'(b) = -\frac{2t_n - 3I_{n-1} + I_{n-2}}{h^2}.$$

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

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

$$\begin{aligned}I_1 &= hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + o\left(h^3\right), \\ I_2 &= hf(a) + \frac{3h^2}{2}f'(a) + \frac{7h^3}{6}f''(a) + o\left(h^3\right), \\ I_3 &= hf(a) + \frac{5h^2}{2}f'(a) + \frac{19h^3}{6}f''(a) + o\left(h^3\right).\end{aligned}$$

Therefore, the following system is obtained:

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} h & \frac{h^2}{2} & \frac{h^3}{6} \\ h & \frac{3h^2}{2} & \frac{7h^3}{6} \\ h & \frac{5h^2}{2} & \frac{19h^3}{6} \end{pmatrix} \begin{pmatrix} f(a) \\ f'(a) \\ f''(a) \end{pmatrix}$$

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. $\square$

4.3 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

$$\begin{aligned} h_1(x) &= \frac{3}{4}e^{-2(9x-2)^2} - \frac{1}{5}e^{-(9x-7)^2-(9x-4)^2} + \frac{1}{2}e^{-(9x-7)^2-\frac{1}{4}(9x-3)^2} + \frac{3}{4}e^{\frac{1}{10}(-9x-1)-\frac{1}{49}(9x+1)^2}, \\ h_2(x) &= \frac{1}{2}x\cos^4\left(4\left(x^2+x-1\right)\right), \\ h_3(x) &= -\frac{\exp\left(-x^2\right)\left(\log\left(x^5+6\right)+\sin(3\pi x)\right)}{\cos(2\pi x)+2}, \end{aligned}$$

whose plots appear in Figures 4.1, 4.2 and 4.3 respectively. The two first functions are the 1D versions of the Franke [87] and Nielson [88] functions.

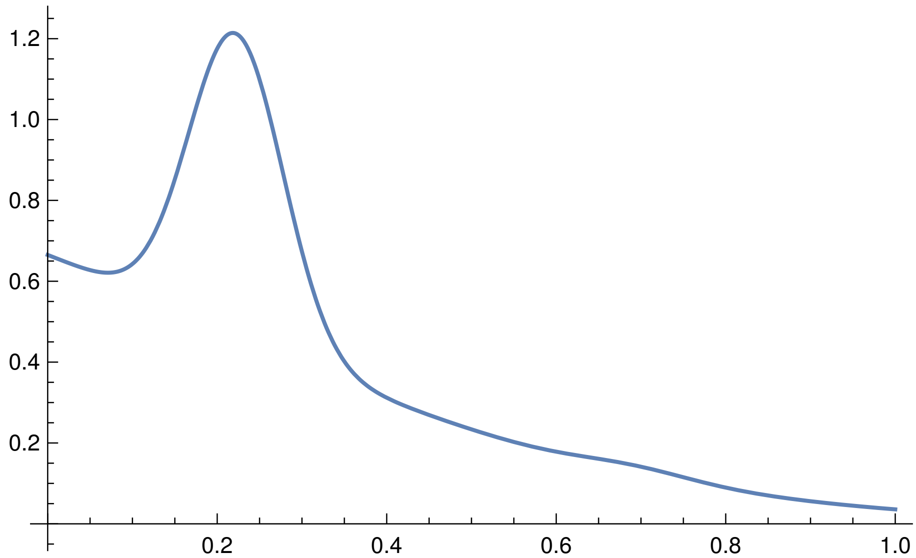


Figure 4.1: Plots of test function h_1.

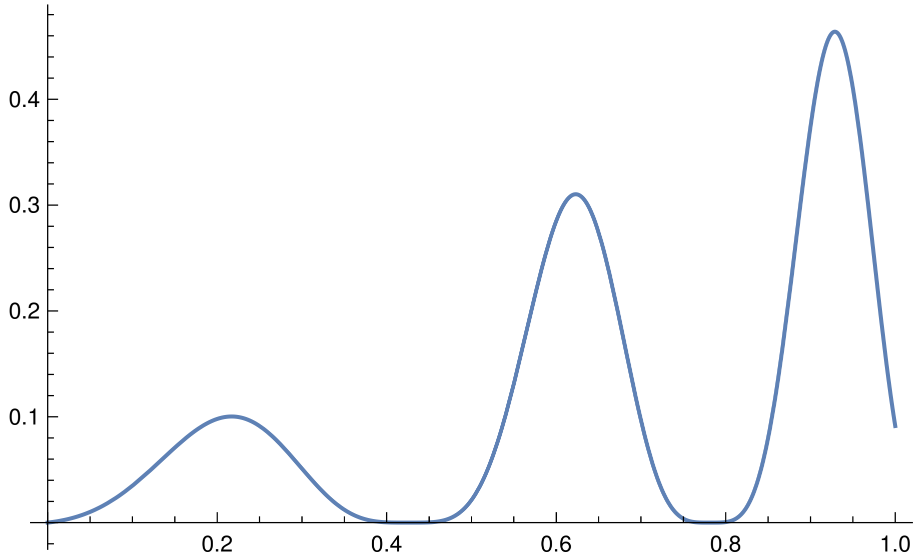


Figure 4.2: Plots of test function h_2.

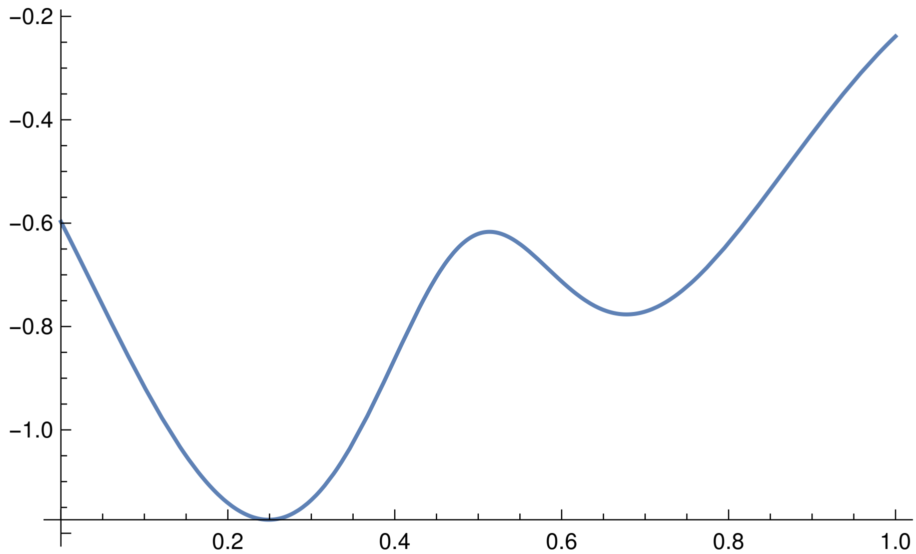


Figure 4.3: Plots of test function h_3.

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 $h = \dfrac{1}{n}$.

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

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

In Table 4.1, the estimated quasi-interpolation errors and NCOs for functions h_1, h_2 and h_3 are shown.

Table 4.1: Estimated errors for functions h_1, h_2 and h_3, and NCOs with different values of n.

n	$\mathcal{E}_n\,(h_1)$	NCO	$\mathcal{E}_n\,(h_2)$	NCO	$\mathcal{E}_n\,(h_3)$	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 simple, namely

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

In Tables 4.2 and 4.3, 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 [84, 89, 90]. Tables 4.2 and 4.3 show that the novel numerical scheme improves the results in References [84, 89, 90].

Table 4.2: Estimated errors for function g_1, and NCOs with different values of n.

n	$\mathcal{E}_n\,(g_1)$	NCO	Method in [84]	NCO	Method in [89]	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 4.3: Estimated errors for function g_2, and NCOs with different values of n.

n	$\mathcal{E}_n\,(g_2)$	NCO	Method in [90]	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 $[0, 1]$:

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

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

Their typical plots are shown in Figure 4.4, 4.5, 4.6, 4.7 respectively.

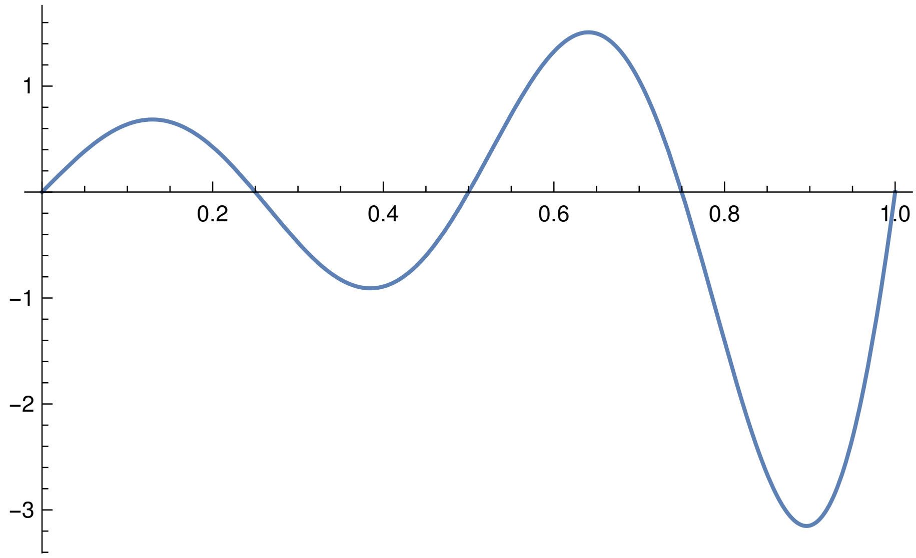


Figure 4.4: Plots of test function k_1.

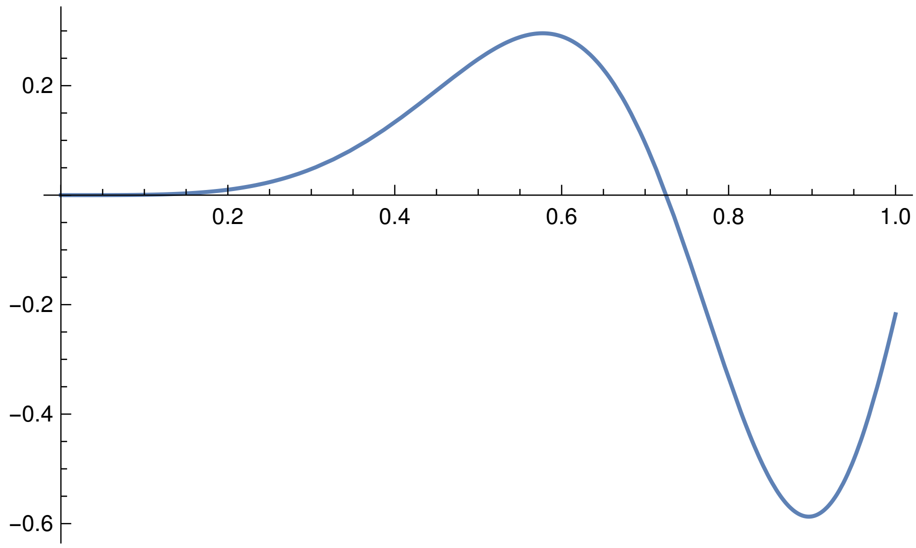


Figure 4.5: Plots of test function k_2.

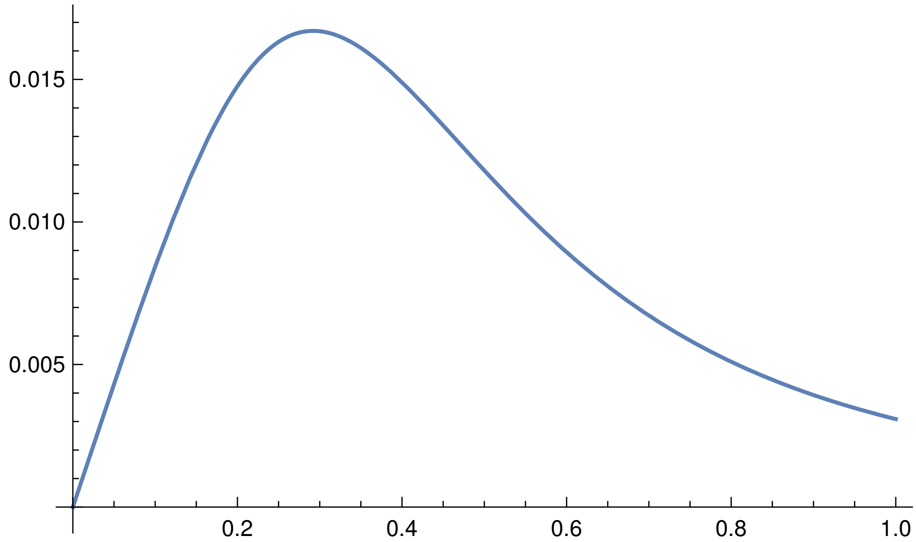


Figure 4.6: Plots of test function k_3.

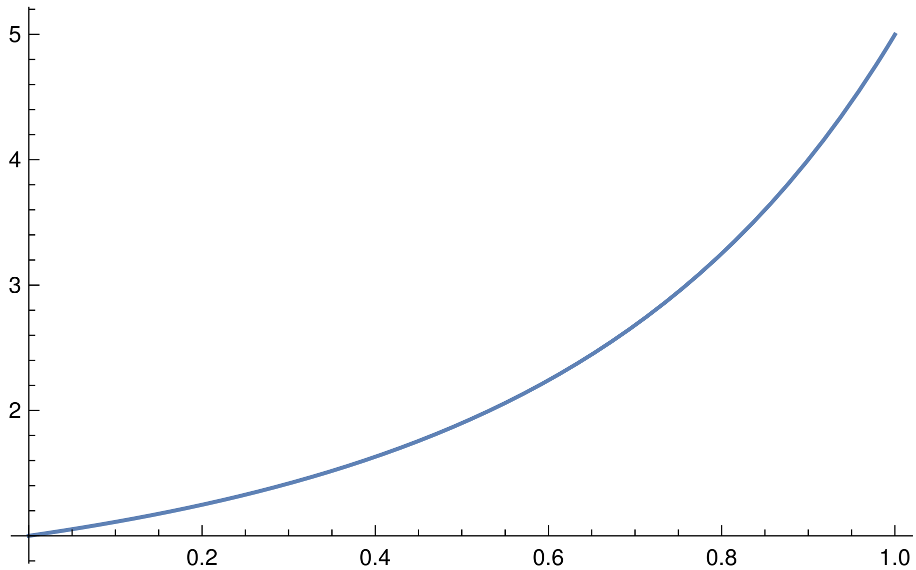


Figure 4.7: Plots of test function k_4.

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 [79], D. Barrera et al. in [69], J. Wu and X. Zhang in [90] and S. Eddargani et al. in [84], so we implement the approaches described in [79, 90, 84] 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.

Tables 4.4 and 4.5 lists the resulting errors for approximating the functions k_1 and k_2, respectively, using our approach and the one provided in [79]. Next, table 4.6 shows the resulting errors for approximating the function k_3, using the method described here and those in References [69, 90, 84]. Finally, in Table 4.7, we list the resulting errors for approximating the function k_4 using the approach provided here and those in References References [90, 84]..

In Tables 4.4 and 4.5, 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 [79]. Table 4.6 shows the resulting errors for the approximation of the function k_3, using the approach described here and those in References [69, 90, 84]. In Table 4.7, we list the resulting errors for the approximation of the function k_4 by using the approach provided here and those in References [90, 84].

Table 4.4: Estimated errors for function k_1, and NCOs with different values of n.

n	$\mathcal{E}_n(k_1)$	NCO	Method in [79]	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 4.5: Estimated errors for function k_2, and NCOs with different values of n.

n	$\mathcal{E}_n(k_2)$	NCO	Method in [79]	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 4.6: Estimated errors for function k_3, and NCOs with different values of n.

n	$\mathcal{E}_n(k_3)$	NCO	Method in [69]	NCO	Method in [90]	NCO	Method in [84]	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 4.7: Estimated errors for function k_4, and NCOs with different values of n.

n	$\mathcal{E}_n(k_3)$	NCO	Method in [90]	NCO	Method in [84]	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

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.

4.4 Conclusions

Approximation from integral values represents a critical topic because of its extensive application in many different areas. This chapter considers a cubic Hermite spline interpolant reproducing linear polynomials and hyperbolic functions. The proposed interpolant is C^2 is 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 excellent performance of the novel approximation scheme.

Conclusion and perspectives

At the end of this work, we address the main results obtained in this thesis and then turn to the future. Indeed, some significant results have been established, but many questions remain. Therefore, we briefly discuss some potential areas for further research before outlining the contributions made in this work.

In this thesis, we have resolved a Hermie interpolation problem using the spaces $\Gamma = \{1, x, sinx, cosx\}$ and $\Gamma' = \{1, x, sinhx, coshx\}$, then we construct two cubic Hermite spline interpolants. Both interpolation schemes are $\mathcal{C}^1$ everywhere. The first scheme produces exactly linear polynomials and trigonometric functions, and the second produces exactly linear polynomials and hyperbolic functions. Next, we assume that the data f_i^1 are not given, and we implement a minimization strategy to estimate these interpolation parameters; this strategy is chosen in order to approximate the monotonicity of the data f_i^0.

We have proposed a $\mathcal{C}^1$ quadratic spline (AT) based on algebraic, trigonometric functions that preserve the monotonicity of the data. In the construction, additional knots are considered in each interval induced by the partition in a specific way to achieve the goal. The proposed method allows the user to adjust the location of the added knots in order to preserve the monotonicity of the data.

We would like to draw the reader's attention that in this study on monotonicity-preserving, only globally monotone data are considered (i.e.,$f_1^0 < f_2^0 < \ldots < f_n^0$).

Approximation from integral values is an essential subject because of its numerous possible applications in various fields. This thesis focuses on a cubic Hermite spline interpolant that reproduces linear polynomials and hyperbolic functions. The proposed approximation scheme is $\mathcal{C}^2$ 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.

In future works, we hope to deal with comonotonicity-preserving methods based on AT and AH splines.
The Hemite basis defined in this work is not all positive. We expect in the following works to find a positive form of the Hermite basis to improve the numerical stability of the constructed schemes.
Another suggestion for further research is to extend the proposed numerical methods to the two dimensions using the tensor product.

Bibliography

[1] C. De Boor, *A Practical Guide to Splines,* Springer-Verlag, New York, (1978).

[2] Philip J. Davis, *Interpolation and approximation,* Dover Publications, New York, (1975).

[3] M. Oraiche, A.Lamnii, M.Louzar, M.Madark, *Fitting and Smoothing Data Using Algebraic Hyperbolic Cubic Hermite Spline Interpolation,* Engineering Letters, 30 (2022).

[4] H. Akima, *A method of smooth curve fitting,* Institute for Telecommunication Sciences, 52 (1969).

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

[6] A. M. Bica, *Fitting data using optimal Hermite type cubic interpolating splines,* Appl. Math. Lett., 25 (2012), 2047–2051.

[7] R. E. Carlson, F. N. Fritsch, *Monotone piecewise cubic interpolation,* SIAM J. Numer. Anal., 17 (1980), 238–246.

[8] D. Barrera, S. Eddargani, A. Lamnii, *Uniform algebraic hyperbolic spline quasi-interpolant based on mean integral values,* Comp and Math Methods, 3 (2020), https://doi.org/10.1002/cmm4.1123.

[9] P. Costantini, *On monotone and convex interpolation,* Math. Comp, 46 (1986), 203–214.

[10] P. Costantini, *Co-monotone interpolating splines of arbitrary degree; A local approach,* SIAM J. Sci. Statist. Comput, 8 (1987), 1026–1034.

[11] C. Conti, R. Morandi, *Piecewise* C^1*-shape-preserving Hermite interpolation,* Computing, 56 (1996), 323–341.

[12] R. Delbourgo, J. A. Gregory, *Piecewise rational quadratic spline interpolation to monotonic data,* IMA J. Numer. Anal., 2 (1982), 123–130.

[13] R. De Vore, G. G. Lorentz, *Constructive Approximation,* Springer-Verlag, Berlin, (1993).

[14] T. N. T. Goodman, K. Unsworth, *Shape-preserving interpolation by parametrically defined curves,* SIAM J. Numer. Anal, 25 (1988), 1453–1465.

[15] T. N. T. Goodman, *Shape preserving interpolation by curves,* in Algorithms for Approximation IV, J. Leversity, I. Anderson, J. Mason, eds., University of Huddersfield, Huddersfield, (2002), 24–35.

[16] H. Greiner, *A survey on univariate data interpolation and approximation by splines of given shape,* Mathl. Comput. Modelling, 05 (1991), 97–108.

[17] X. Han, X. Guo, *Cubic Hermite interpolation with minimal derivative oscillation,* J. Comput. Appl. Math., 331 (2018), 82–87.

[18] K. Ichida, F. Yoshimoto, T. Kiyono, *Curve fitting by a piecewise cubic polynomial,* Computing, 16 (1976), 329–338.

[19] B. I. Kvasov, *Methods of Shape-Preserving Spline Approximation,* World Scientific, Singapore, (2000).

[20] A. Lahtinen, *Shape preserving interpolation by quadratic splines,* J. of Comput and Appl. Math., 29 (1990), 15–24.

[21] P. Lamberti, C. Manni, *Shape-preserving* $\mathcal{C}^2$ *functional interpolation via parametric cubics,* Numer. Algorithms, 28 (2020), 229–254.

[22] "A. Lamnii, M. Lamnii, F. Oumellal, *A new basis for osculatory interpolation problems and applications,* Appl. Math. Comput., 283 (2016), 355–368.

[23] T. Lyche, L. L. Schumaker, *Quasi-interpolants Based on Trigonometric Splines.* Computer Aided Geometric Design, 95 (1995), 280–309.

[24] Y. Lu, G. Wang and X. Yang, *Uniform hyperbolic polynomial B-spline curves.* Computer Aided Geometric Design, 32 (2001), 379–393.

[25] D. F. Macallister, J. A. Roulier, *Interpolation by convex quadratic splines.* J. Comput. Appl. Math., 32 (1987), 1154–1162.

[26] C. Manni, $\mathcal{C}^1$ *comonotone Hermite interpolation via parametric cubics*, J. Comput. Appl. Math. 69 (1996), 143–157.

[27] X. Han, *Quadratic trigonometric polynomial curves with a shape parameter,* Computer Aided Geometric Design, 19 (2002), 503–512.

[28] C. Manni, *A general framework for functional tension splines,* J. Comput. Appl. Math., 119 (2000), 275–300.

[29] C. Manni, *On shape preserving* $\mathcal{C}^2$ *Hermite interpolation,* BIT, 41 (2001), 127–148.

[30] J.-L. Merrien, P. Sablonnière, *Monotone and Convex* $\mathcal{C}^1$ *Hermite Interpolants Generated by a Subdivision Scheme,* Adv. Comput. Math., 19 (2003), 279–298.

[31] A. Mazroui, D. Sbibih, A. Tijini, *A simple method for smoothing function and compressing Hermite data,* Adv. Comput. Math., 23 (2005), 279–297.

[32] M. A. Ramadan, and I. F. Lashien, and W. K. Zahra, *Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems,* Appl. Math. and Comput, 184 (2007), 476–484.

[33] P. Sablonnière, and D. Sbibih, *Integral spline operators exact on polynomials,* Approx. Theory Appl, 10 (1994), 56–73.

[34] L. L. Schumaker, *Spline Functions: Basic Theory,* Wiley Interscience, New York 1981, Reprinted by Krieger, Malabar, Florida, (1981).

[35] L. L. Schumaker, *On shape preserving quadratic spline interpolation,* SIAM J. Numer. Anal, 20 (1983), 854–864.

[36] D. G. Schweiker, *An interpolating curve using a spline in tension,*J. Math. Phys, 45 (1966), 312–317.

[37] Y. Zhu, and X. Han, *Shape preserving* $\mathcal{C}^2$ *rational quartic interpolation spline with two parameters,* International Journal of Computer Mathematics, 92 (2015), 2160–2177.

[38] Y. Lü, G. Wang, X. Yang, *Uniform hyperbolic polynomial B-spline curves,* Computer Aided Geometric Design, 19 (2002), 379–393.

[39] Y. Zhu, Z. Liu, *A Class of trigonometric Bernstein-type basis functions with four shape parameters,* Mathematical Problems in Engineering, 60 (2019), 16.

[40] D. Barrera, S. Eddargani, A. Lamnii, M. Oraiche, *'On nonpolynomial monotonicity-preserving* $\mathcal{C}^1$ *spline interpolation,* Comp and Math Methods, (2021), e1160. https://doi.org/10.1002/cmm4.1160.

[41] L.L. Shumakher, *On Hyperbolic Splines," Journal of approximation theory,* 38 (1983), 144-166.

[42] J.Li and C.Liu, *A Class of Polynomial Spline Curve with Free Parameters that Naturally Interpolates the Data Points,* IAENG International Journal of Applied Mathematics, 50 (2020), 643-647.

[43] X.B. Qin, L. Qin and Q.S. Xu, $\mathcal{C}^1$ *positivity-preserving interpolation schemes with local free parameters,* IAENG International Journal of Computer Science,(2016), 219–227.

[44] X.B. Qin, Q.S. Xu, *A new* $\mathcal{C}^2$ *piecewise bivariate rational interpolation scheme with bi-quadratic denominator,* IAENG International Journal of Applied Mathematics, 47 (2017), 454–458.

[45] K. Ichida, F. Yoshimoto, T. Kiyono, *Curve fitting by a piecewise cubic polynomial,* Computing, 16 (1976), 329–338.

[46] M. S. Mummy, *Hermite interpolation with B-splins,* Computer aided geometric design, 6 (1989), 177-179.

[47] C. Manni, $\mathcal{C}^1$ *comonotone Hermite interpolation via parametric cubics,* Journal of computational and applied mathematics, 69 (1996), 143-157.

[48] S. Micula, G. Micula, *On the superconvergent spline collocation methods for the Fredholm integral equations on surfaces,* Math. Balkanica, 19 (2005), 155-166.

[49] G. Mastroianni, G.V. Milanovic, D. Occorsio, *Nyström method for Fredholm integral equations of the second kind in two variables on a triangle,* Appl. Math. Comp, 14 (2013), 7653-7662.

[50] I.J.Schoenberg, *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., 4 (1946), 45–99.

[51] J. Y. Zheng, G. Hu, X. M. Ji, X. Q. Qin, *Quintic generalized Hermite interpolation curves: construction and shape optimization using an improved GWO algorithm,* Comput. Appl. Math., 41 (2022), 1–29.

[52] M. Ammad, M. Y. Misro, M. Abbas, A. Majeed, *Generalized Developable Cubic Trigonometric Bézier Surfaces,* Mathematics, 9 (2021), 283.

[53] C.De Boor, *A Practical Guide to Splines,* Applied Mathematical Sciences Springer: 27 (1978).

[54] D. Barrera, S. Eddargani, A. Lamnii, *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., 404 (2022), 113761. https://doi.org/10.1016/j.cam.2021.113761.

[55] D. Barrera, S. Eddargani, M. J. Ibáñez, A. Lamnii, *A new approach to deal with* C^2 *cubic splines and its application to super-convergent quasi-interpolation,* Math. Comput. Simul., 194 (2022), 401–415.

[56] S. N. Ershov, *B-Splines and Bernstein Basis Polynomials,* Phys. Part. Nuclei Lett., 16 (2019), 593–601.

[57] Y. Y. Yu, H. Ma, C. G. Zhu, *Total positivity of a kind of generalized toric-Bernstein basis,* Linear Algebra Appl., 579 (2019), 449–462.

[58] P. Koch, T. Lyche, M. Neamtu, L. Schumaker, *Control curves and knot insertion for trigonometric splines,*. Adv. Comput. Math., 3 (1995), 405–424.

[59] G. Walz, *Identities for trigonometric B-splines with an application to curve design,* BIT, 37 (1997), 189–201.

[60] T. Lyche, R. Winther, *A stable recurrence relation for trigonometric B-splines,* J. Approx. Theory, 25 (1979), 266–279.

[61] P. E. Koch, T. Lyche, *Construction of exponential tension B-splines of arbitrary order,* In Curves and Surfaces, Academic Press, (1991), 255–258.

[62] C. Conti, L. Gemignani, L. Romani, *Exponential Pseudo-Splines: Looking beyond Exponential B-splines,* J. Math. Anal. Appl., 439 (2016), 32–56.

[63] R. Campagna, C. Conti, S. Cuomo, *Smoothing exponential-polynomial splines for multi-exponential decay data,* Dolomites Res. Notes Approx., 12 (2019), 86–100.

[64] R. Campagna, C. Conti, *Penalized hyperbolic-polynomial splines,* Appl. Math. Lett., 118 (2021), 107159. https://doi.org/10.1016/ j.aml.2021.107159.

[65] X. Yang, *Fitting and fairing Hermite-type data by matrix weighted NURBS curves,* Comput. Aided Des., 102 (2018), 22–32.

[66] H. Speleers, *Algorithm 1020: Computation of Multi-Degree Tchebycheffian B-Splines,* ACM Trans. Math. Softw., 48 (2022), 1–31.

[67] X. Liu, A. A. Divani, A. Petersen, Truncated estimation in functional generalized linear regression models, Comput. Stat. Data Anal., 169 (2022), 107421.

[68] S. Eddargani, A. Lamnii, M. Lamnii, D.Sbibih, A. Zidna, *Algebraic hyperbolic spline quasi-interpolants and applications,* J. Comput. Appl. Math. 347 (2019), 196–209.

[69] D. Barrera, S. Eddargani, A. Lamnii, *Uniform algebraic hyperbolic spline quasi-interpolant based on mean integral values,* Comp. Math. Methods, 3 (2021), e1123. https://doi.org/10.1002/cmm4.1123.

[70] J. M. Carnicer, E. Mainar, M. Peña, *Interpolation on cycloidal spaces,* J. Approx. Theory, 187 (2014, 18–29.

[71] L. Mazure, *From Taylor interpolation to Hermite interpolation via duality,* Jaen J. Approx., 4 (2012), 15–45.

[72] R. Ait-Haddou, M. L. Mazure, H. Ruhland, *A remarkable Wronskian with application to critical lengths of cycloidal spaces,* Calcolo, 56 (2019), 45–56.

[73] C. Conti, L. Romani, M. Unser, *Ellipse-preserving Hermite interpolation and subdivision,* J. Math. Anal. Appl., 426 (2015), 211–227.

[74] J. Y. Zheng, G. Hu, X. M. Ji, X. Q. Qin, *Quintic generalized Hermite interpolation curves: construction and shape optimization using an improved GWO algorithm,* Comput. Appl. Math., 41 (2022), 1–29.

[75] D. Barrera, S. Eddargani, A. Lamnii, M. Oraiche, *On non polynomial monotonicity-preserving C^1 spline interpolation,* Comp. Math. Methods, 3 (2021), e1160. https://doi.org/10.1002/cmm4.1160.

[76] M. Ajeddar, A. Lamnii, *Smooth reverse subdivision of uniform algebraic hyperbolic B-splines and wavelets,* Int. J. Wavelet Multiresolut. Inf. Process., 19 (2021), 2150018.

[77] M. Marusic, M. Rogina, *Sharp error-bounds for interpolating splines in tension,* J. Comput. Appl. Math., 61 (1995), 205–223.

[78] A. Delhez, *A spline interpolation technique that preserve mass budget,* Appl. Math. Lett., 16 (2003), 17–26.

[79] A. Boujraf, D. Sbibih, M. Tahrichi, A. Tijini, *A simple method for constructing integro spline quasi-interpolants,* Math. Comput. Simul., 111 (2015), 36–47.

[80] H. Behforooz, *Approximation by integro cubic splines,* Appl. Math. Comput., 175 (2006), 8–15.

[81] H. Behforooz, *Interpolation by integro quintic splines,* Appl. Math. Comput., 216 (2010), 364–367.

[82] T. Zhanlav, R. Mijiddorj, *Integro cubic splines and their approximation properties,*. Appl. Math. Ser. Tver State Univ. Russia, 26 (2008), 65–77.

[83] F. G. Lang, X. P. Xu, *On integro quartic spline interpolation,* J. Comput. Appl. Math., 236 (2012), 4214–4226.

[84] S. Eddargani, A. Lamnii, M. Lamnii, *On algebraic trigonometric integro splines,* Z. Angew. Math. Mech., 100 (2020, e201900262.

[85] R. Mijiddorj, T. Zhanlav, *Algorithm to construct integro splines,* ANZIAM J., 63 (2021), 359–375.

[86] T. Zhanlav, R. Mijiddorj, *Integro cubic splines on non-uniform grids and their properties,* East Asian J. Appl. Math., 11 (2021), 406–420.

[87] R. Franke, *Scattered data interpolation: Tests of some methods,* Math. Comp., 38 (1982), 181–200.

[88] G.M.Nielson, *A first order blending method for triangles based upon cubic interpolation,* Int. J. Numer. Meth. Eng., 15 (1978), 308–318.

[89] T. Zhanlav, R. Mijiddorj, *The local integro cubic splines and their approximation properties,* Appl. Math. Comput., 216 (2010), 2215–2219.

[90] J. Wu, X. Zhang, *Integro quadratic spline interpolation,* Appl. Math. Modell., 39 (2015), 2973–2980.

FICHE PRÉSENTATIVE DE LA THÈSE

- ▷ **Nom et Prénom de l'auteur : ORAICHE Mohammed**
- ▷ **Intitulé du travail**: *NON-POLYNOMIAL HERMITE SPLINES INTERPOLATION.*
- ▷ **Encadrant : Mr.LOUZAR Mohamed**, Professeur de l'enseignement supérieur, Université Hassan 1^{er}, FST Settat.
- ▷ **Co-Encadrant : Mr.LAMNII Abdellah**, Professeur de l'enseignement supérieur, Université Abdelmalek Essaadi, ENS Tetouan.
- ▷ **Lieux de réalisation des travaux** (laboratoires, institution,...):
  - ∗ *Laboratoire de Mathématiques, Informatique et Sciences de l'ingénieur, Faculté des Sciences et Techniques, Settat, Maroc.*
- ▷ **Période de réalisation du travail de thèse**: Décembre 2018– November 2022.
- ▷ **Président** (nom, prénom, grade, institution):
  - • **Mr.HAJJAJ Abdelkarim**, Professeur de l'enseignement supérieur, Université Hassan 1^{er}, FST Settat
- ▷ **Rapporteurs**: (nom, prénom, grade, institution) :
  - • **Mr.MESTARI Mohammed**, Professeur de l'enseignement supérieur, Université Hassan II, ENSET Mohammedia
  - • **Mr.ELOMARY Mohamed Abdou**, Professeur de l'enseignement supérieur, Université Hassan 1^{er}, FST Settat
  - • **Mr.EL JID Rachid**, Professeur Habilité, Université Hassan 1^{er}, FST Settat
- ▷ **Examinateur** (nom, prénom, grade, institution):
  - • **Mr.FAKHAR Rachid**, Professeur Habilité, Université Sultan moulay Slimane, FP Khouribga

Liste des publications

1. M.Oraiche, A.Lamnii, M.Louzar, M.Madark, Fitting and Smoothing Data Using Algebraic Hyperbolic Cubic Hermite Spline Interpolation. Engineering Letters, 30(2), (2022).
2. S.Eddargani, M.Oraiche, A.Lamnii, M.Louzar, C^2 Cubic Algebraic Hyperbolic Spline Interpolating Scheme by Means of Integral Values. Mathematics, 10(9), (2022), p.1490.
3. D.Barrera, S.Eddargani, A.Lamnii, M.Oraiche, On nonpolynomial monotonicity-preserving C^1 spline interpolation. Computational and Mathematical Methods, 3(4), (2021), p.e1160.

Communications orales

- M.Oraiche, A.Lamnii, A reverse non-stationary mixed trigonometric and hyperbolic B-splines subdivision scheme with two tension and one shape parameters,
  the 3th edition of the International Symposium on New Trends in Approximation and Applications NT2A'22, june 27-28 2022, Mohammed First University, Oujda.

- M.Oraiche, A.Lamnii, M.Louzar, Fitting data by Algebraic Hyperbolic cubic Hermite Spline Interpolation.
  The International Online Conference on Applied Mathematics IOCAM22, june 1-3, 2022, Fez.

- M.Oraiche, M.Louzar, A.Lamnii, Interpolation Spline by mean of integral values.
  9ème édition de la Journée Doctorant du CEDoc STSM, 16 juin 2022, FST Settat.

- M.Oraiche, A.Lamnii, C^2 cubic algebraich hyperbolic spline interpolating by mean of integral values.
  Internationnal Conference on New Trends in Applied Mathematics, May 19-21, 2022, Sultan Moulay Slimane University.

- M.Oraiche, A.Lamnii, M.Louzar, Interpolating by mean of integral values.
  Journée de Mathématiques et Applications, 14 mai 2022, FST Settat.

- M.Oraiche, A. Lamnii, On non-polynomial C^1 splines Hermite interpolation.
  The 2ed Internationnal Conference on Fixed Point Theory and Applications, November 30, 2019, FST Mohammedia.