

On Discrete Hyperbolic Tension Splines

Paolo Costantini ^a, Boris I. Kvasov ^b, and Carla Manni ^c

^aDipartimento di Matematica, Università di Siena,
Via del Capitano 15, 53100, SIENA, ITALY,
`costantini@unisi.it`

^bInstitute of Computational Technologies, Russian Academy of Sciences
Lavrentyev Avenue 6, 630090, NOVOSIBIRSK, RUSSIA,
`boris@math.sut.ac.th`

^cDipartimento di Matematica, Università di Torino,
Via Carlo Alberto 10, 10123, TORINO, ITALY
`manni@dm.unito.it`

A hyperbolic tension spline is defined as the solution of a differential multipoint boundary value problem. A discrete hyperbolic tension spline is obtained using the difference analogous of differential operators; its computation does not require exponential functions, even if its continuous extension is still a spline of hyperbolic type. We consider the basic computational aspects and show the main features of this approach.

Keywords: Hyperbolic tension splines, multipoint boundary value problem, discrete hyperbolic tension splines and B-splines, shape preserving interpolation.

1. Introduction

Spline theory is mainly grounded on two approaches: the algebraic one (where splines are understood as smooth piecewise functions, see e.g. [29,31]) and the variational one (where splines are obtained via minimization of quadratic functionals with equality and/or inequality constraints, see e.g. [15]). Although less common, a third approach where splines are defined as the solutions of differential multipoint boundary value problems (DMBVP for short), has been considered, [9]. Even though some of the important classes of splines can be obtained from all three schemes, specific features make sometimes the last one an important tool in practical settings. We want to illustrate this fact by the example of hyperbolic tension splines.

Introduced by Schweikert in 1966, [30], hyperbolic tension splines are solutions of DMBVP where the differential operators depend on *tension* parameters. Their tension properties (that is the possibility of pulling the curve toward a piecewise linear function) have kept hyperbolic splines popular (see for example [11,24,25,27] and references quoted therein) in shape-preserving interpolation and/or approximation. Unfortunately, it is difficult to work with hyperbolic splines for small or large values of the tension parameters. For this

reason, in spite of the presence of refined algorithms for their calculation [25], hyperbolic tension splines were forced out by rational splines (see for example [6, 12]) in practical applications.

We observe that for practical purposes, it is often necessary to know the values of the solution S of a DMBVP only over a prescribed grid instead of its global analytic expression. In this paper, we study a natural discretization of the DMBVP replacing, in the given interval $[a, b]$, the differential operator by its difference approximation. This provides a linear system with a pentadiagonal matrix. It turns out that the solution of the discretized problem, called *mesh solution*, is not a tabulation of S but can be extended on $[a, b]$ to a smooth function U which has shape properties very similar to those of S and which provides a second order approximation of S as the discretization step goes to zero. Due to these properties we will refer to U as a *discrete hyperbolic tension spline*.

In contrast with the continuous case, an important fact here is that the values of a discrete hyperbolic tension spline over a prescribed grid in $[a, b]$ (basically the mesh solution) can be obtained solving a pentadiagonal system. This construction is substantially cheaper than performing calculations by the standard algorithm [25], which involves the solution of a simple 3-diagonal system, but with hyperbolic coefficients. In addition, the classical construction requires the evaluation of hyperbolic functions over the prescribed grid with much larger computational cost.

Moreover, just as cubic splines can be seen as a subclass of the exponential ones in the continuous setting, our discrete hyperbolic splines generalize the concept of discrete polynomial splines and reduce to them as the tension parameters go to zero.

Discrete polynomial splines have been studied extensively. They were introduced in [18] as solutions to certain minimization problems involving differences instead of derivatives. They are connected to best summation formulas [19] and have been used in [17] for the computation of nonlinear splines by iteration. Approximation properties of discrete splines have been studied in [16]. Discrete B-splines on a uniform partition were introduced in [28] and discrete B-splines on a non-uniform partition were defined in [2, p.15]. In [3] discrete B-splines were applied to the general area of subdivision. While discrete polynomial splines are currently attracting widespread research interest [21,22,23], discrete tension splines and B-splines have been less studied. The only results we know regarding this topic concern discrete exponential Box-splines [5, 26] and are therefore related to uniform partitions.

The content of this paper is as follows. In Section 2 we formulate the problem. In Section 3 we prove the existence of a mesh solution by constructing its extension as a discrete hyperbolic tension spline. An upper bound for the distance between a discrete hyperbolic tension spline and the corresponding continuous one is established in Section 4. In Section 5 we give direct and recurrence algorithms for constructing discrete hyperbolic tension B-splines. Section 6, with its subsections, is devoted to the discussion of practical aspects and computational advantages of our discrete spline. Finally, Section

7 gives some graphical examples to illustrate the main properties of discrete hyperbolic tension splines.

2. Finite Difference Approximation

Let the data

$$(x_i, f_i), \quad i = 0, \dots, N + 1, \quad (2.1)$$

be given, where: $a = x_0 < x_1 < \dots < x_{N+1} = b$. Let us put

$$h_i = x_{i+1} - x_i, \quad i = 0, \dots, N.$$

An *interpolating hyperbolic tension spline* S with a set of tension parameters $\{p_i \geq 0 \mid i = 0, \dots, N\}$ is a solution of the DMBVP

$$\frac{d^4 S}{dx^4} - \left(\frac{p_i}{h_i}\right)^2 \frac{d^2 S}{dx^2} = 0, \quad \text{in each } (x_i, x_{i+1}), \quad i = 0, \dots, N, \quad (2.2)$$

$$S \in C^2[a, b], \quad (2.3)$$

with the interpolation conditions

$$S(x_i) = f_i, \quad i = 0, \dots, N + 1, \quad (2.4)$$

and some end constraints. For the sake of simplicity we only consider the following classical end conditions

$$S''(a) = f_0'' \quad \text{and} \quad S''(b) = f_{N+1}''. \quad (2.5)$$

Let us now consider a discretized version of the previous DMBVP. Let $n_i \in \mathbb{N}$, $i = 0, \dots, N$, be given; we look for

$$\{u_{ij}, \quad j = -1, \dots, n_i + 1, \quad i = 0, \dots, N\},$$

satisfying the difference equations:

$$\left[\Lambda_i^2 - \left(\frac{p_i}{h_i}\right)^2 \Lambda_i\right] u_{ij} = 0, \quad j = 1, \dots, n_i - 1, \quad i = 0, \dots, N, \quad (2.6)$$

where

$$\Lambda_i u_{ij} = \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\tau_i^2}, \quad \tau_i = \frac{h_i}{n_i}.$$

The smoothness condition (2.3) is changed into

$$\begin{aligned} u_{i-1, n_{i-1}} &= u_{i0}, \\ \frac{u_{i-1, n_{i-1}+1} - u_{i-1, n_{i-1}-1}}{2\tau_{i-1}} &= \frac{u_{i,1} - u_{i,-1}}{2\tau_i}, \quad i = 1, \dots, N, \end{aligned} \quad (2.7)$$

$$\Lambda_{i-1} u_{i-1, n_{i-1}} = \Lambda_i u_{i,0}$$

while conditions (2.4)–(2.5) take the form

$$\begin{aligned} u_{i,0} &= f_i, \quad i = 0, \dots, N, \quad u_{N, n_N} = f_{N+1}, \\ \Lambda_0 u_{0,0} &= f_0'', \quad \Lambda_N u_{N, n_N} = f_{N+1}''. \end{aligned} \quad (2.8)$$

Our discrete *mesh solution* will be then defined as

$$\{u_{ij}, \quad j = 0, \dots, n_i, \quad i = 0, 1, \dots, N\}. \quad (2.9)$$

In the next section we prove the existence of the solution of the previous linear system while we postpone to Section 6 the comments on the practical computation of the mesh solution.

3. System Splitting and Mesh Solution Extension

In order to analyze the solution of system (2.6)–(2.8) we introduce the notation

$$m_{ij} = \Lambda_i u_{ij}, \quad j = 0, \dots, n_i, \quad i = 0, \dots, N. \quad (3.1)$$

Then, on the interval $[x_i, x_{i+1}]$, (2.6) takes the form

$$\begin{aligned} m_{i0} &= m_i, \\ \frac{m_{i,j-1} - 2m_{ij} + m_{i,j+1}}{\tau_i^2} - \left(\frac{p_i}{h_i}\right)^2 m_{ij} &= 0, \quad j = 1, \dots, n_i - 1, \\ m_{i,n_i} &= m_{i+1}, \end{aligned} \quad (3.2)$$

where m_i and m_{i+1} are prescribed numbers. The system (3.2) has a unique solution, which can be represented as follows

$$m_{ij} = M_i(x_{ij}), \quad x_{ij} = x_i + j\tau_i, \quad j = 0, \dots, n_i,$$

with

$$M_i(x) = m_i \frac{\sinh k_i(1-t)}{\sinh(k_i)} + m_{i+1} \frac{\sinh k_i t}{\sinh(k_i)}, \quad t = \frac{x - x_i}{h_i},$$

and where the parameters k_i are the solutions of the transcendental equations

$$2n_i \sinh \frac{k_i}{2n_i} = p_i, \quad p_i \geq 0,$$

that is

$$k_i = 2n_i \ln \left(\frac{p_i}{2n_i} + \sqrt{\left(\frac{p_i}{2n_i}\right)^2 + 1} \right) \geq 0, \quad i = 0, \dots, N.$$

From (3.1) and from the interpolation conditions (2.8) we have

$$\begin{aligned} u_{i0} &= f_i, \\ \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{\tau_i^2} &= m_{ij}, \quad j = 0, \dots, n_i, \\ u_{i,n_i} &= f_{i+1}. \end{aligned} \quad (3.3)$$

For each sequence m_{ij} , $j = 0, \dots, n_i$, system (3.3) has a unique solution which can be represented as follows

$$u_{ij} = U_i(x_{ij}), \quad j = -1, \dots, n_i + 1,$$

where

$$U_i(x) = f_i(1-t) + f_{i+1}t + \varphi_i(1-t)h_i^2 m_i + \varphi_i(t)h_i^2 m_{i+1}, \quad (3.4)$$

with

$$\varphi_i(t) = \frac{\sinh(k_i t) - t \sinh(k_i)}{p_i^2 \sinh(k_i)}.$$

In order to solve system (2.6)–(2.8), we only need to determine the values m_i , $i = 0, \dots, N + 1$, so that the smoothness conditions (2.7) and the end conditions in (2.8) are verified. From (3.3)–(3.4), conditions (2.7) can be rewritten as

$$\begin{aligned} U_{i-1}(x_i) &= U_i(x_i), \\ \frac{U_{i-1}(x_i + \tau_{i-1}) - U_{i-1}(x_i - \tau_{i-1})}{2\tau_{i-1}} &= \frac{U_i(x_i + \tau_i) - U_i(x_i - \tau_i)}{2\tau_i}, \\ \Lambda_{i-1}U_{i-1}(x_i) &= \Lambda_i U_i(x_i), \end{aligned} \quad (3.5)$$

where

$$\Lambda_j U_j(x) = \frac{U_j(x + \tau_j) - 2U_j(x) + U_j(x - \tau_j)}{\tau_j^2}, \quad x \in [x_j, x_{j+1}].$$

Then, from (3.1)–(3.2) and (3.4), the first and the third equalities in (3.5) are immediately satisfied, while, using (3.4) and the end conditions in (2.8), the second equality provides the following linear system with a 3-diagonal matrix for the unknown values m_i :

$$\begin{aligned} m_0 &= f_0'', \\ \alpha_{i-1}h_{i-1}m_{i-1} + (\beta_{i-1}h_{i-1} + \beta_i h_i)m_i + \alpha_i h_i m_{i+1} &= d_i, \quad i = 1, \dots, N, \\ m_{N+1} &= f_{N+1}'', \end{aligned} \quad (3.6)$$

where

$$\begin{aligned} d_i &= \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}, \\ \alpha_i &= -\frac{\varphi_i(\frac{1}{n_i}) - \varphi_i(-\frac{1}{n_i})}{\frac{2}{n_i}} = -\frac{n_i \sinh(\frac{k_i}{n_i}) - \sinh(k_i)}{p_i^2 \sinh(k_i)}, \\ \beta_i &= \frac{\varphi_i(1 + \frac{1}{n_i}) - \varphi_i(1 - \frac{1}{n_i})}{\frac{2}{n_i}} = \frac{n_i \cosh(k_i) \sinh(\frac{k_i}{n_i}) - \sinh(k_i)}{p_i^2 \sinh(k_i)}. \end{aligned}$$

Expanding the hyperbolic functions in the above expressions as power series we obtain

$$\beta_i \geq 2\alpha_i > 0, \quad i = 0, \dots, N, \quad \text{for all } n_i > 1, \quad p_i \geq 0.$$

Therefore, the system (3.6) is diagonally dominant and has a unique solution. We can now conclude that system (2.6)–(2.8) has a unique solution which can be represented as $U_i(x_{ij})$, $j = -1, \dots, n_i + 1$, $i = 0, \dots, N$, whenever the constants m_i are solution of (3.6).

Let us put

$$U(x) := U_i(x), \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, \dots, N. \quad (3.7)$$

Due to the previous construction we will refer to U as *discrete hyperbolic tension spline* interpolating the data (2.1). We observe that we recover the result of [17] for discrete cubics since

$$\lim_{p_i \rightarrow 0} \alpha_i = \frac{1}{6} \left(1 - \frac{1}{n_i^2} \right), \quad \lim_{p_i \rightarrow 0} \beta_i = \frac{1}{6} \left(2 + \frac{1}{n_i^2} \right), \quad \lim_{p_i \rightarrow 0} \varphi_i(t) = \frac{t(t^2 - 1)}{6}. \quad (3.8)$$

4. Error Estimates

In this section, we present a bound for the distance between the discrete hyperbolic tension spline defined in (3.7) and the corresponding continuous one interpolating the same set of data and having the same end conditions.

As mentioned in section 2, the classical C^2 hyperbolic tension spline interpolating the data (2.1) is a function S satisfying (2.2)–(2.5). It is well known that we can express $S_i(x) := S(x)|_{[x_i, x_{i+1}]}$ as

$$S_i(x) = f_i(1 - t) + f_{i+1}t + \tilde{\varphi}_i(1 - t)h_i^2\tilde{m}_i + \tilde{\varphi}_i(t)h_i^2\tilde{m}_{i+1}, \quad (4.1)$$

where

$$\tilde{m}_{i+j} = \frac{d^2S}{dx^2}(x_{i+j}), \quad \tilde{\varphi}_i(t) = \frac{\sinh(p_it) - t \sinh(p_i)}{p_i^2 \sinh(p_i)}, \quad i = 0, \dots, N, \quad j = 0, 1,$$

and the constants \tilde{m}_i are solutions of the linear system

$$\begin{aligned} \tilde{m}_0 &= f_0'', \\ \tilde{\alpha}_{i-1}h_{i-1}\tilde{m}_{i-1} + (\tilde{\beta}_{i-1}h_{i-1} + \tilde{\beta}_i h_i)\tilde{m}_i + \tilde{\alpha}_i h_i \tilde{m}_{i+1} &= d_i, \quad i = 1, \dots, N, \\ \tilde{m}_{N+1} &= f_{N+1}'', \end{aligned} \quad (4.2)$$

where

$$\tilde{\alpha}_i = -\tilde{\varphi}'_i(0) = \frac{\sinh(p_i) - p_i}{p_i^2 \sinh(p_i)}, \quad \tilde{\beta}_i = \tilde{\varphi}'_i(1) = \frac{p_i \cosh(p_i) - \sinh(p_i)}{p_i^2 \sinh(p_i)}.$$

It is easy to verify that $\tilde{\beta}_i \geq 2\tilde{\alpha}_i > 0$, $\forall p_i \geq 0$, so that the 3-diagonal linear system (4.2) is diagonally dominant. In addition, as $n_i \rightarrow +\infty$, systems (3.6) and (4.2) coincide since

$$\lim_{n_i \rightarrow +\infty} \alpha_i = \tilde{\alpha}_i, \quad \lim_{n_i \rightarrow +\infty} \beta_i = \tilde{\beta}_i.$$

Let us put

$$\tilde{A} := \min_{i=1, \dots, N} (\tilde{\beta}_i - \tilde{\alpha}_i)h_i + (\tilde{\beta}_{i-1} - \tilde{\alpha}_{i-1})h_{i-1} > 0. \quad (4.3)$$

Since (see [4] for details) \mathcal{C}_i is a bounded function of n_i , then from (4.7), for each fixed sequence of the values p_0, \dots, p_N , we have a second order convergence of the discrete hyperbolic tension splines to the corresponding continuous spline. The results agree with the order of approximation of the discretization which we have used for the first, second and fourth derivatives. For example, let us consider in detail the upper bound (4.7) in the limit case $p_i = 0, i = 0, \dots, N$. From (3.8), (4.4) and (4.6) we obtain

$$\lim_{p_i \rightarrow 0} \mathcal{A}_i = \frac{1}{6}, \quad \lim_{p_i \rightarrow 0} (\tilde{\beta}_i - \tilde{\alpha}_i) = \frac{1}{6}, \quad \lim_{p_i \rightarrow 0} \mathcal{B}_i \leq 1, \quad \lim_{p_i \rightarrow 0} \mathcal{C}_i = 0,$$

so that from (4.3) and (4.7)

$$\|S_i - U_i\| \leq 4h_i^2 \tau^2 \|\mathbf{m}\|_\infty \max_{i=1, \dots, N} \frac{1}{h_i + h_{i-1}} \left[\max_{i=0, \dots, N} \frac{1}{h_i} \right],$$

and we recover, with some improvements, the corresponding result of [17].

Finally, we observe that (4.7) can be used to estimate the rate of convergence of a discrete hyperbolic tension spline towards a function generating the interpolation points as $\max_i h_i \rightarrow 0$. To do this, it suffices to combine, via the triangle inequality, (4.7) with the results of [20] where the convergence of a continuous hyperbolic tension spline towards a function generating the interpolation points is studied.

5. Discrete Hyperbolic Tension B-Splines

In this section, we use the strategy outlined in [13,14], where generalized B-splines and their properties are discussed in more detail.

Let us associate with a partition $\Delta : a = x_0 < x_1 < \dots < x_{N+1} = b$ of the interval $[a, b]$ a space of functions S_4^{DH} whose restriction to an interval $[x_i, x_{i+1}]$, $i = 0, \dots, N$ is spanned by the system of four linearly independent functions $\{1, x, \Phi_i, \Psi_i\}$ and where every function in S_4^{DH} satisfies smoothness conditions (3.5) for discrete hyperbolic tension splines.

Following [14] let us rewrite formula (3.4) on the interval $[x_i, x_{i+1}]$, $i = 0, \dots, N$, in the form

$$\begin{aligned} U(x) \equiv U_i(x) = & [f_i - \Phi_i(x_i)m_i](1-t) + [f_{i+1} - \Psi_i(x_{i+1})m_{i+1}]t \\ & + \Phi_i(x)m_i + \Psi_i(x)m_{i+1}, \end{aligned} \quad (5.1)$$

where $t = (x - x_i)/h_i$, $m_j = \Lambda_i U_i(x_j)$, $j = i, i+1$, and

$$\begin{aligned} \Psi_i(x) &= \psi_i(t)h_i^2 = \psi(p_i, t)h_i^2, \quad \Phi_i(x) = \psi_i(1-t)h_i^2, \\ \psi_i(t) &= \frac{\sinh(kt) - tn_i \sinh(k_i/n_i)}{p_i^2 \sinh(k_i)}. \end{aligned} \quad (5.2)$$

Functions Φ_i and Ψ_i satisfy the conditions

$$\begin{aligned} \Psi_i(x_i + j\tau_i) &= \Phi_i(x_{i+1} + j\tau_i) = 0, \quad j = -1, 0, 1, \\ \Lambda_i \Phi_i(x_i) &= \Lambda_i \Psi_i(x_{i+1}) = 1. \end{aligned} \quad (5.3)$$

Let us construct a basis for the space of discrete hyperbolic tension splines S_4^{DH} by using functions which have local supports of minimum length. Since $\dim(S_4^{DH}) = 4(N+1) - 3N = N+4$ we extend the grid Δ by adding the points x_j , $j = -3, -2, -1, N+2, N+3, N+4$, such that $x_{-3} < x_{-2} < x_{-1} < a$, $b < x_{N+2} < x_{N+3} < x_{N+4}$.

We demand that the discrete hyperbolic tension B-splines (HB-splines for short) B_i , $i = -3, \dots, N$ have the following properties

$$\begin{aligned} B_i(x) &> 0, & x \in (x_i + \tau_i, x_{i+4} - \tau_{i+4}), \\ B_i(x) &\equiv 0, & x \notin (x_i, x_{i+4}), \\ \sum_{j=-3}^N B_j(x) &\equiv 1, & x \in [a, b]. \end{aligned} \quad (5.4)$$

5.1 Construction of HB-Splines

According to (5.1), on the interval $[x_j, x_{j+1}]$, $j = i, \dots, i+3$, the discrete HB-spline B_i has the form

$$B_i(x) \equiv B_{i,j}(x) = P_{i,j}(x) + \Phi_j(x)m_{j,B_i} + \Psi_j(x)m_{j+1,B_i},$$

where $P_{i,j}$ is a polynomial of the first degree and $m_{l,B_i} = \Lambda_l B_i(x_l)$, $l = j, j+1$ are constants to be determined. The smoothness conditions (3.5) and constraints (5.3) give the following relations

$$\begin{aligned} P_{i,j}(x_j) &= P_{i,j-1}(x_j) + z_j m_{j,B_i}, \\ P_{i,j}[x_j - \tau_j, x_j + \tau_j] &= P_{i,j-1}[x_j - \tau_{j-1}, x_j + \tau_{j-1}] + c_{j-1,2} m_{j,B_i}, \end{aligned}$$

where

$$\begin{aligned} z_j &\equiv z_j(x_j) = \Psi_{j-1}(x_j) - \Phi_j(x_j), \\ c_{j-1,2} &= \Psi_{j-1}[x_j - \tau_{j-1}, x_j + \tau_{j-1}] - \Phi_j[x_j - \tau_j, x_j + \tau_j]. \end{aligned}$$

Thus

$$P_{i,j}(x) = P_{i,j-1}(x) + [z_j + c_{j-1,2}(x - x_j)]m_{j,B_i}. \quad (5.5)$$

As B_i vanishes outside the interval (x_i, x_{i+4}) , we have from (5.5), in particular, $P_{i,j} \equiv 0$ for $j = i, i+3$. By repeated use of formula (5.5) we get

$$P_{i,j}(x) = \sum_{l=i+1}^j [z_l + c_{l-1,2}(x - x_l)]m_{l,B_i} = - \sum_{l=j+1}^{i+3} [z_l + c_{l-1,2}(x - x_l)]m_{l,B_i}.$$

In particular, the following identity is valid

$$\sum_{j=i+1}^{i+3} [z_j + c_{j-1,2}(x - x_j)]m_{j,B_i} \equiv 0,$$

from which one obtains the equalities

$$\sum_{j=i+1}^{i+3} c_{j-1,2} y_j^r m_{j,B_i} = 0, \quad r = 0, 1, \quad y_j = x_j - \frac{z_j}{c_{j-1,2}}. \quad (5.6)$$

Thus, the formula for the discrete HB-spline B_i takes the form

$$B_i(x) = \begin{cases} \Psi_i(x) m_{i+1,B_i}, & x \in [x_i, x_{i+1}), \\ (x - y_{i+1}) c_{i,2} m_{i+1,B_i} \\ + \Phi_{i+1}(x) m_{i+1,B_i} + \Psi_{i+1}(x) m_{i+2,B_i}, & x \in [x_{i+1}, x_{i+2}), \\ (y_{i+3} - x) c_{i+2,2} m_{i+3,B_i} \\ + \Phi_{i+2}(x) m_{i+2,B_i} + \Psi_{i+2}(x) m_{i+3,B_i}, & x \in [x_{i+2}, x_{i+3}), \\ \Phi_{i+3}(x) m_{i+3,B_i}, & x \in [x_{i+3}, x_{i+4}), \\ 0, & \text{otherwise.} \end{cases} \quad (5.7)$$

Substituting formula (5.7) into the normalization condition (5.4) written for $x \in [x_i, x_{i+1}]$, we obtain

$$\begin{aligned} \sum_{j=i-3}^i B_j(x) &= \Phi_i(x) \sum_{j=i-3}^{i-1} m_{i,B_j} + \Psi_i(x) \sum_{j=i-2}^i m_{i+1,B_j} \\ &+ (y_{i+1} - x) c_{i,2} m_{i+1,B_{i-2}} + (x - y_i) c_{i-1,2} m_{i,B_{i-1}} \equiv 1. \end{aligned}$$

As according to (5.4)

$$\sum_{j=i-3}^{i-1} m_{i,B_j} = \sum_{j=i-2}^i m_{i+1,B_j} = 0 \quad (5.8)$$

the following identity is valid

$$(y_{i+1} - x) c_{i,2} m_{i+1,B_{i-2}} + (x - y_i) c_{i-1,2} m_{i,B_{i-1}} \equiv 1.$$

From here one gets the equalities

$$y_{i+1}^r c_{i,2} m_{i+1,B_{i-2}} - y_i^r c_{i-1,2} m_{i,B_{i-1}} \equiv \delta_{1,r}, \quad r = 0, 1,$$

where $\delta_{1,r}$ is the Kronecker symbol. Solving this system of equations and using (5.6) or (5.8), we obtain

$$m_{j,B_i} = \frac{y_{i+3} - y_{i+1}}{c_{j-1,2} \omega'_{i+1}(y_j)}, \quad j = i+1, i+2, i+3,$$

$$\omega_{i+1}(x) = (x - y_{i+1})(x - y_{i+2})(x - y_{i+3})$$

or with the notation $c_{j,3} = y_{j+2} - y_{j+1}$, $j = i, i+1$,

$$\begin{aligned} m_{i+1,B_i} &= \frac{1}{c_{i,2} c_{i,3}}, \\ m_{i+2,B_i} &= -\frac{1}{c_{i+1,2}} \left(\frac{1}{c_{i,3}} + \frac{1}{c_{i+1,3}} \right), \\ m_{i+3,B_i} &= \frac{1}{c_{i+2,2} c_{i+1,3}}. \end{aligned} \quad (5.9)$$

5.2 Recurrence Formulas for HB-Splines

Let us define functions

$$B_{j,2}(x) = \begin{cases} \Lambda_j \Psi_j(x), & x \in [x_j, x_{j+1}), \\ \Lambda_{j+1} \Phi_{j+1}(x), & x \in [x_{j+1}, x_{j+2}), \\ 0, & \text{otherwise,} \end{cases} \quad j = i, i+1, i+2. \quad (5.10)$$

Using (5.2) one can readily check that $\Lambda_j \Phi_j$ and $\Lambda_j \Psi_j$ are strictly monotonic functions on the interval $[x_j, x_{j+1}]$. The splines $B_{j,2}$ are a generalization of the ‘‘hat-functions’’ for polynomial B-splines. They are nonnegative and, furthermore, $B_{j,2}(x_{j+l}) = \delta_{1,l}$, $l = 0, 1, 2$.

Let us denote

$$\begin{aligned} \Lambda U(x) &\equiv \Lambda_i U_i(x), \\ D_1 U(x) &\equiv U_i[x - \tau_i, x + \tau_i], \end{aligned} \quad x \in [x_i, x_{i+1}], \quad i = 0, \dots, N;$$

then from (3.5) ΛU and $D_1 U$ are well defined if $U \in S_4^{DH}$. With the previous notation, according to (5.7), (5.9), and (5.10) we obtain

$$\begin{aligned} \Lambda B_i(x) &= \sum_{j=i+1}^{i+3} m_{j,B_i} B_{j-1,2}(x) \\ &= \frac{1}{c_{i,3}} \left(\frac{B_{i,2}(x)}{c_{i,2}} - \frac{B_{i+1,2}(x)}{c_{i+1,2}} \right) - \frac{1}{c_{i+1,3}} \left(\frac{B_{i+1,2}(x)}{c_{i+1,2}} - \frac{B_{i+2,2}(x)}{c_{i+2,2}} \right). \end{aligned} \quad (5.11)$$

In addition the function $D_1 B_i$ satisfies to the relation

$$D_1 B_i(x) = \frac{B_{i,3}(x)}{c_{i,3}} - \frac{B_{i+1,3}(x)}{c_{i+1,3}}, \quad (5.12)$$

where

$$B_{j,3}(x) = \begin{cases} \frac{1}{c_{j,2}} \Psi_j[x - \tau_j, x + \tau_j], & x \in [x_j, x_{j+1}), \\ 1 + \frac{1}{c_{j,2}} \Phi_{j+1}[x - \tau_{j+1}, x + \tau_{j+1}] \\ - \frac{1}{c_{j+1,2}} \Psi_{j+1}[x - \tau_{j+1}, x + \tau_{j+1}], & x \in [x_{j+1}, x_{j+2}), \\ - \frac{1}{c_{j+1,2}} \Phi_{j+2}[x - \tau_{j+2}, x + \tau_{j+2}], & x \in [x_{j+2}, x_{j+3}), \\ 0, & \text{otherwise.} \end{cases} \quad (5.13)$$

Functions $B_{j,3}$ and $B_{j,4} \equiv B_j$ possess many of the properties inherent in usual discrete polynomial B-splines. We collect their characteristics in the next theorem which can be proved by using the explicit formulae (5.7), (5.10), and (5.13) for discrete HB-splines $B_{j,k}$, $j = 2, 3, 4$, and the relations (5.11) and (5.12).

Theorem 1. *The functions $B_{j,k}$, $k = 3, 4$ have the following properties:*

1. $B_{j,4}(x) > 0$ for $x \in (x_j + \tau_j, x_{j+4} - \tau_{j+4})$, and $B_{j,4}(x) \equiv 0$ if $x \notin (x_j, x_{j+4})$,
 $B_{j,3}(x) > 0$ for $x \in (x_j, x_{j+3})$, and $B_{j,3}(x) \equiv 0$ if $x \notin (x_j, x_{j+3})$;
2. $B_{j,4}$ satisfies the continuity conditions (3.5);
3. $B_{j,3}$ satisfies the first and second continuity conditions in (3.5);
4. $\sum_{j=-2}^N B_{j,3}(x) \equiv 1$ for $x \in [a, b]$,
 $\Phi_j[x - \tau_j, x + \tau_j] = -c_{j-1,2}B_{j-2,3}(x)$, $\Psi_j[x - \tau_j, x + \tau_j] = c_{j,2}B_{j,3}(x)$
for $x \in [x_j, x_{j+1}]$, $j = 0, \dots, N$;
5. $\sum_{j=-3}^N y_{j+2}^r B_{j,4}(x) \equiv x^r$, $r = 0, 1$ for $x \in [a, b]$,
 $\Phi_j(x) = c_{j-1,2}c_{j-2,3}B_{j-3,4}(x)$, $\Psi_j(x) = c_{j,2}c_{j,3}B_{j,4}(x)$
for $x \in [x_j, x_{j+1}]$, $j = 0, \dots, N$.

Figures 1 and 2 show the graphs of discrete HB-splines $B_{j,k}$, $k = 2, 3, 4$ (from left to right) on a uniform mesh with step size $h = 1$ and with $\tau_i = \tau$ for all i . We have chosen discretization parameters $\tau = 0.1$ (Fig. 1, left and Fig. 2, right), $\tau = 0.33$ (Fig. 1, right) and $\tau = 0.5$ (Fig. 2, left) for $\psi_i(t)$ from (5.2). In figures 1 and 2 (left) we have parameters $p_i = 0$, i.e. we have conventional discrete cubic B-splines (e.g., see [16]). Visually, the presence of intervals where the B-spline $B_{j,4}$ is negative is more visible with growing discretization parameter τ . In figure 2 (right) the tension parameters are $p_i = 50$ for all i , whence the shape of the graphs is practically unchanged when τ increases from 0.1 to 0.5. As the limit for $p_i \rightarrow \infty$ we obtain the pulse function for $B_{j,2}$, the ‘‘step-function’’ for $B_{j,3}$ and the ‘‘hat-function’’ for $B_{j,4}$ (all of height 1).

Figure 3 shows the graphs of discrete HB-splines $B_{j,4}$ on a uniform mesh (left) and on a nonuniform mesh (right), where the asterisk * denotes the x_i . For both plots $p_i = 2$ and $n_i = 2$.

Using the approach of [14], it is easy to show that the functions B_j , $j = -3, \dots, N$ have supports of minimum length, are linearly independent and form a basis in the space S_4^{DH} . So any discrete hyperbolic tension spline $U \in S_4^{DH}$ can be uniquely represented in the form

$$U(x) = \sum_{j=-3}^N b_j B_j(x) \quad (5.14)$$

with some constant coefficients b_j .

Applying formulae (5.11) and (5.12) to the representation (5.14) we obtain

$$D_1 U(x) = \sum_{j=-2}^N b_{j,3} B_{j,3}(x), \quad \Lambda U(x) = \sum_{j=-1}^N b_{j,2} B_{j,2}(x), \quad (5.15)$$

where

$$b_{j,3-k} = \frac{b_{j,4-k} - b_{j-1,4-k}}{c_{j,3-k}}, \quad k = 0, 1; \quad b_{j,4} \equiv b_j.$$

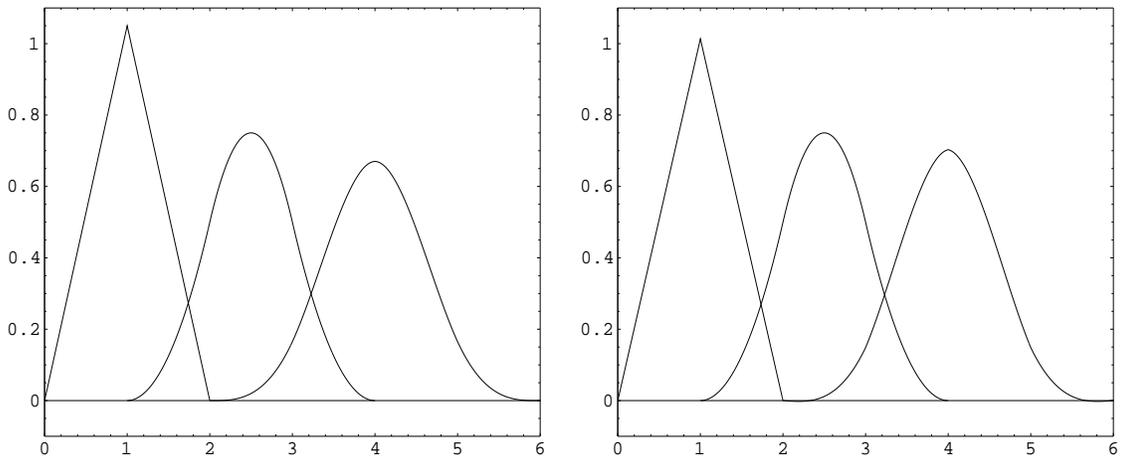


Fig. 1. The discrete HB-splines $B_{j,k}$, $k = 2, 3, 4$ (from left to right) on a uniform mesh with step size $h = 1$, no tension and discretization parameter $\tau = 0.1$ (left) and $\tau = 0.33$ (right).

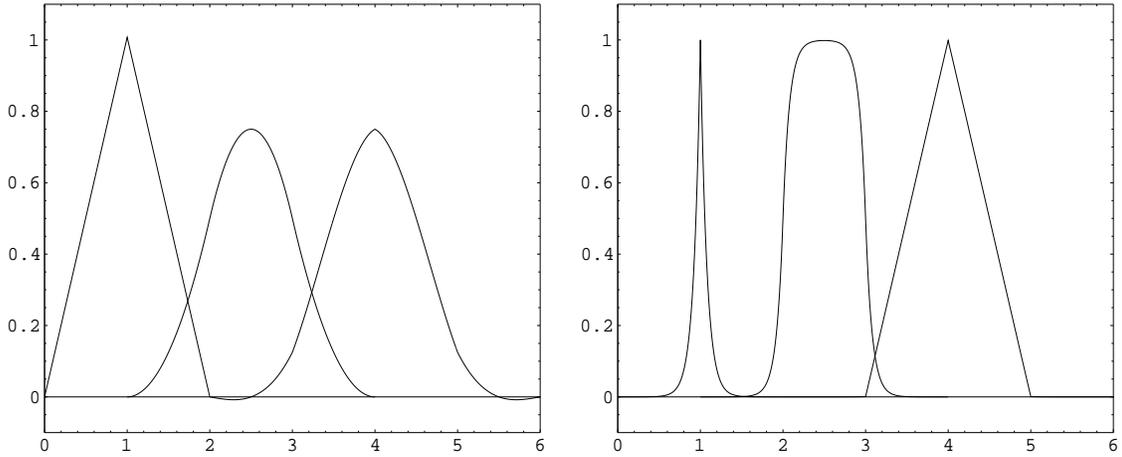


Fig. 2. Same as Fig. 1, but with discretization parameter $\tau = 0.5$ (left) and with tension parameters $p_i = 50$ for all i (right).

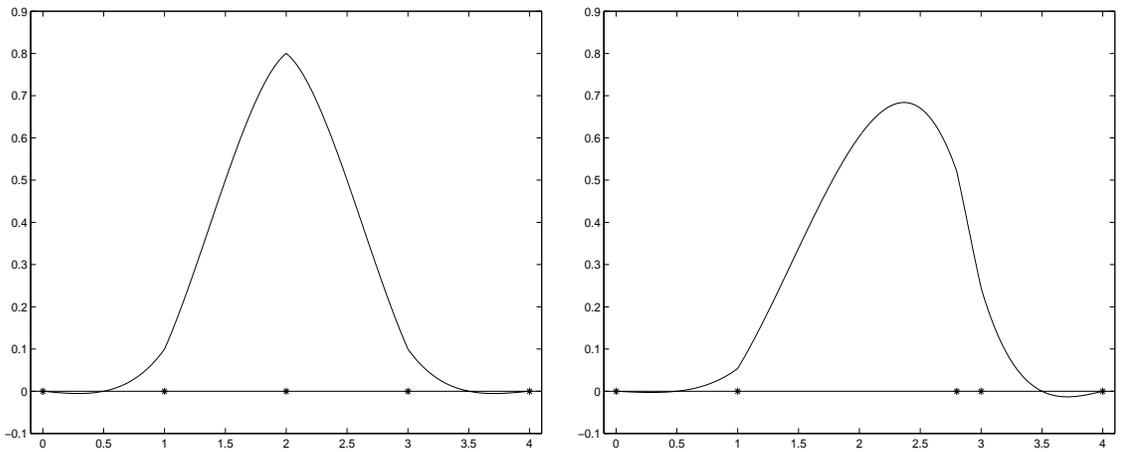


Fig. 3. The discrete HB-splines $B_{j,4}$ on a uniform mesh (left) and on a nonuniform mesh (right). The asterisk $*$ denotes the x_i . For both plots $p_i = 2$ and $n_i = 2$.

5.3 Formulas for Local Approximation by HB-Splines

If the coefficients b_j in (5.14) are known then by virtue of formula (5.7) we can write out an expression for the discrete hyperbolic tension spline U on the interval $[x_i, x_{i+1}]$, which is convenient for calculations,

$$U(x) = b_{i-2} + \tilde{\Delta}_i b_{i-2}(x - y_i) + c_i \Phi_i(x) + c_{i+1} \Psi_i(x), \quad (5.16)$$

where

$$c_j = \frac{\tilde{\Delta}_j b_{j-2} - \tilde{\Delta}_{j-1} b_{j-3}}{c_{j-1,2}}, \quad j = i, i+1, \quad \tilde{\Delta}_j b_{j-2} = \frac{b_{j-1} - b_{j-2}}{c_{j-1,3}}.$$

The representations (5.14) and (5.16) allow us to find a simple and effective way to approximate a given function f from its samples.

Theorem 2. *For $b_j = f(y_{j+2})$, $j = -3, \dots, N$, formula (5.14) is exact for polynomials of the first degree and provides a formula for local approximation.*

Proof: It suffices to prove that the identities

$$\sum_{j=-3}^N y_{j+2}^r B_j(x) \equiv x^r, \quad r = 0, 1 \quad (5.17)$$

hold for $x \in [a, b]$. Using formula (5.16) with the coefficients $b_{j-2} = 1$ and $b_{j-2} = y_j$, $j = i-1, i, i+1, i+2$, for an arbitrary interval $[x_i, x_{i+1}]$, we find that identities (5.17) hold.

For $b_{j-2} = f(y_j)$, formula (5.16) can be rewritten as

$$U(x) = f(y_i) + f[y_i, y_{i+1}](x - y_i) + (y_{i+1} - y_{i-1})f[y_{i-1}, y_i, y_{i+1}]c_{i-1,2}^{-1}\Phi_i(x) \\ + (y_{i+2} - y_i)f[y_i, y_{i+1}, y_{i+2}]c_{i,2}^{-1}\Psi_i(x), \quad x \in [x_i, x_{i+1}].$$

This is the formula of local approximation. The theorem is proved.

Corollary 1. *By setting*

$$b_{j-2} = f_j - \frac{1}{c_{j-1,2}} \left[\Psi_{j-1}(x_j) \frac{f_{j+1} - f_j}{h_j} - \Phi_j(x_j) \frac{f_j - f_{j-1}}{h_{j-1}} \right] \quad (5.18)$$

in (5.14), we obtain a formula of three-point local approximation, which is exact for polynomials of the first degree.

Proof: To prove the corollary, it is sufficient to take the monomials 1 and x as f . Then according to (5.18), we obtain $b_{j-2} = 1$ and $b_{j-2} = y_j$ and it only remains to make use of identities (5.17). This proves the corollary.

Equation (5.16) permits us to write the coefficients of the spline U in its representation (5.14) in the form

$$b_{j-2} = \begin{cases} U(y_j) - \Lambda_{j-1}U(x_{j-1})\Phi_{j-1}(y_j) - \Lambda_j U(x_j)\Psi_{j-1}(y_j), & y_j < x_j, \\ U(y_j) - \Lambda_j U(x_j)\Phi_j(y_j) - \Lambda_{j+1}U(x_{j+1})\Psi_j(y_j), & y_j \geq x_j. \end{cases} \quad (5.19)$$

Using this formula we obtain $b_{j-2} = U(y_j) + O(\bar{h}_j^2)$, $\bar{h}_j = \max(h_{j-1}, h_j)$. Hence it follows that the control polygon (e.g., see [8]) converges quadratically to the function f for $b_{j-2} = f(y_j)$, or if the formula (5.18) is used. Formulas (5.16), (5.17), and (5.19) generalize their continuous equivalents developed in [10].

6. Computational Aspects

The aim of this section is to investigate the practical aspects related to the numerical evaluation of the mesh solution defined in (2.9).

A standard approach, [25], consists of solving the tridiagonal system (3.6) and then evaluating (3.4) at the mesh points as is usually done for the evaluation of continuous hyperbolic splines. At first sight, this approach based on the solution of a tridiagonal system seems preferable because of the limited waste of computational time and the good classical estimates for the condition number of the matrix in (3.6). However, it should be observed that, as in the continuous case, we have to perform a large number of numerical computations of hyperbolic functions of the form $\sinh(k_i t)$ and $\cosh(k_i t)$ both to define system (3.6) and to tabulate functions (3.4). This is a very difficult task, both for cancellation errors (when $k_i \rightarrow 0$) and for overflow problems (when $k_i \rightarrow \infty$). A stable computation of the hyperbolic functions was proposed in [25], where different formulas for the cases $k_i \leq 0.5$ and $k_i > 0.5$ were considered and a specialized polynomial approximation for $\sinh(\cdot)$ was used.

However, we note that this approach is the only one possible if we want a continuous extension of the discrete solution beyond the mesh point.

In contrast, the discretized structure of our construction provides us with a much cheaper and simpler approach to compute the mesh solution (2.9). This can be achieved both by following the system splitting approach presented in Section 3, or by a direct computation of the solution of the linear system (2.6)–(2.8).

As for the system splitting approach, presented in Section 3, the following algorithm can be considered.

Step 1. Solve the 3-diagonal system (3.6) for m_i , $i = 1, \dots, N$.

Step 2. Solve $N + 1$ 3-diagonal systems (3.2) for m_{ij} , $j = 1, \dots, n_i - 1$, $i = 0, \dots, N$,

Step 3. Solve $N + 1$ 3-diagonal systems (3.3) for u_{ij} , $j = 1, \dots, n_i - 1$, $i = 0, \dots, N$.

In this algorithm, hyperbolic functions need only be computed in step 1. Furthermore, the solution of any system (3.2) or (3.3) requires $8q$ arithmetic operations, namely, $3q$ additions, $3q$ multiplications, and $2q$ divisions [31], where q is the number of unknowns, and is thus substantially cheaper than direct computation by formula (3.4).

Steps 2 and 3 can be replaced by a direct splitting of the system (2.6)–(2.8) into $N + 1$ systems with 5-diagonal matrices

$$\begin{aligned} u_{i,0} &= f_i, & \Lambda_i u_{i,0} &= M_i, \\ \Lambda_i^2 u_{i,j} - \left(\frac{p_i}{h_i}\right)^2 \Lambda_i u_{i,j} &= 0, & j &= 1, \dots, n_i - 1, & i &= 0, \dots, N. & (6.1) \\ u_{i,n_i} &= f_{i+1}, & \Lambda_i u_{i,n_i} &= M_{i+1}, \end{aligned}$$

Also, in this case the calculations for steps 2 and 3 or for system (6.1) can be tailored for a multiprocessor computer system.

In addition, the eigenvalues of \mathbf{D} are 0 and 2, thus we deduce from a corollary of the Courant-Fisher theorem [7] that the eigenvalues of \mathbf{A} satisfy the following inequalities

$$\lambda_k(\mathbf{A}) \geq \lambda_k(\mathbf{C}) = \min_{i,j} \lambda_j(\mathbf{C}_i) = \min_i \left[4 \left(1 - \cos \frac{\pi}{n_i} \right)^2 + 2\omega_i \left(1 - \cos \frac{\pi}{n_i} \right) \right].$$

Hence, \mathbf{A} is a positive matrix and we directly obtain that the pentadiagonal linear system has a unique solution.

In addition, by Gershgorin's theorem, $\lambda_k(\mathbf{A}) \leq \max_i [16 + 4\omega_i]$. Then we obtain the following upper bound for the condition number of \mathbf{A} which is independent of the number of data points, $N + 2$, and which recovers the result presented in [17] for the limit case $p_i = 0$, $i = 0, \dots, N$,

$$\begin{aligned} \|\mathbf{A}\|_\infty \|\mathbf{A}^{-1}\|_\infty &\leq \frac{\max_i [16 + 4(\frac{p_i}{n_i})^2]}{\min_i [4(1 - \cos \frac{\pi}{n_i})^2 + 2(\frac{p_i}{n_i})^2(1 - \cos \frac{\pi}{n_i})]} \\ &\simeq \frac{\max_i [16 + 4(\frac{p_i}{n_i})^2]}{\min_i (\frac{1}{n_i})^4 [\pi^4 + (\pi p_i)^2]}. \end{aligned} \tag{6.3}$$

Summarizing, in the particular but important uniform case we can compute the mesh solution by solving a symmetric, pentadiagonal, positive definite system and therefore, we can use specialized algorithms, with a computational cost of $17q$ arithmetic operations, namely, $7q$ additions, $7q$ multiplications, and $3q$ divisions [31], where q is the number of unknowns.

Moreover, since the upper bound (6.3) for the condition number of the matrix \mathbf{A} does not depend on the number of interpolation points, such methods can be used with some confidence.

In the general case of a non-uniform mesh, the matrix \mathbf{A} is no longer symmetric, and an analysis of its condition number cannot be carried out analytically. However, several numerical experiments have shown that the condition number is not influenced by the non-symmetric structure, but does depend on the maximum number of grid points in each subinterval, exactly as in the symmetric case. In other words, symmetric and nonsymmetric matrices, with the same dimension and produced by difference equations with the same largest n_i , produce very close condition numbers. Non-uniform discrete hyperbolic tension splines have in fact been used for the graphical tests of the following section.

6.3 System Splitting

Sometimes the number of unknowns in (6.2) can be very large (for example for generating a grid in bivariate interpolation) and then even the linear computational cost of the solution of the pentadiagonal system may turn out to be too expensive. However, as for the two first approaches proposed at the beginning of this section for evaluating the mesh solution, if we have a

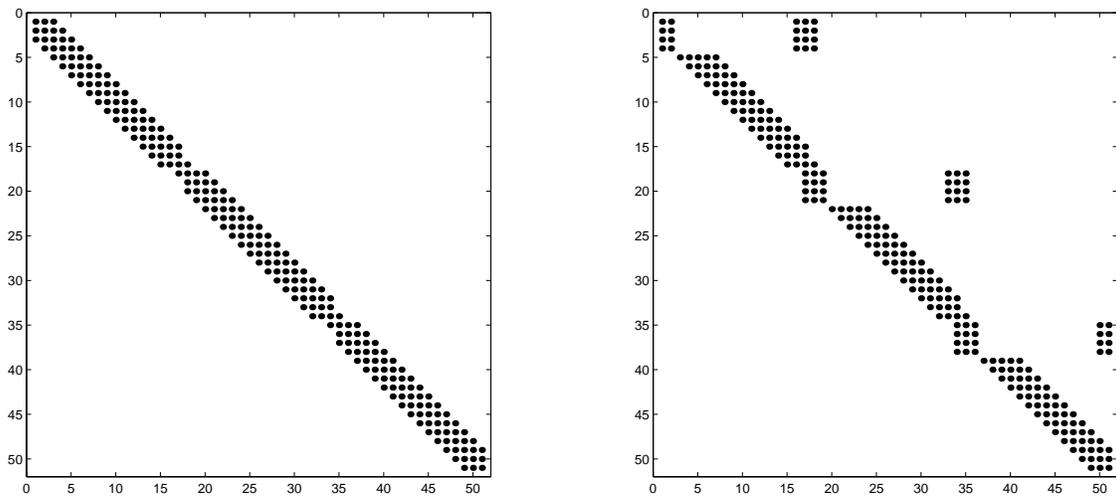


Fig. 4. Left: the form of \mathbf{A} for $N = 2$, $n_i = 18$. Right: the matrix \mathbf{K} .

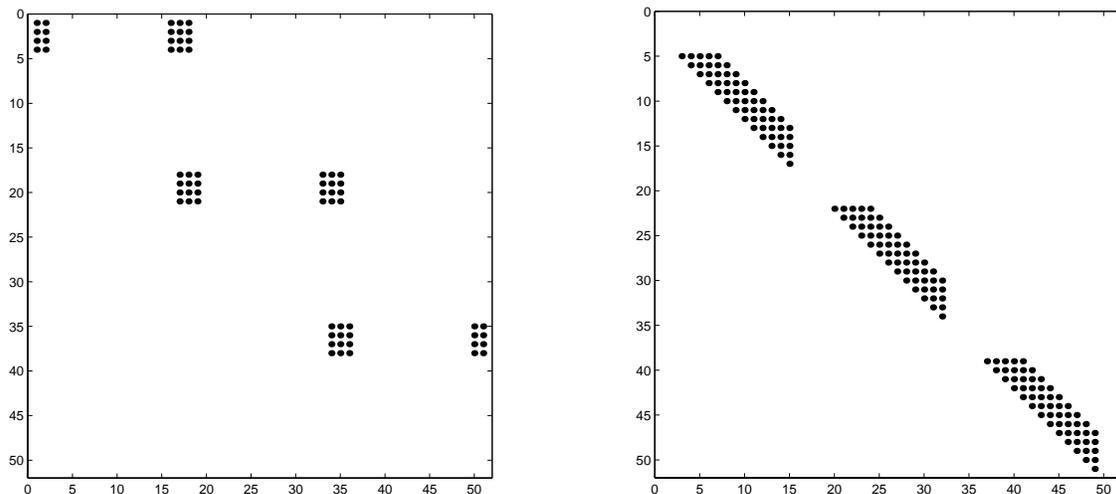


Fig. 5. Left: The block matrix \mathbf{E} . Right: the block matrix \mathbf{F} .

parallel machine we can easily share the computation of the solution of our pentadiagonal system among the processors as outlined below.

The basic idea is to transform \mathbf{A} , which, for $N = 2$, $n_i = 18$ has the form shown in Fig. 4 left, into the form \mathbf{K} (see Fig. 4 right). Setting $r_i = \sum_{\nu=0}^{i-1} (n_\nu - 1)$, we note that the rows $r_i + 1, \dots, r_i + n_i - 1$ of \mathbf{A} describe equations (2.6) for the subinterval $[x_i, x_{i+1}]$. If we extract from \mathbf{K} the rows $r_i + 1, \dots, r_i + 4$, $i = 0, 1, \dots, N$, we get a block matrix \mathbf{E} of the form shown in Fig. 5 left. The corresponding linear system has few equations, and having solved it, it is possible to solve in parallel the $N + 1$ linear systems obtained from the “remaining” matrix \mathbf{F} of Fig. 5 right by extracting its independent blocks.

The problem now is how to move from \mathbf{A} to \mathbf{K} . From Sections 2 and 3 we have the following two facts. Having in mind the structure of \mathbf{A} and the corresponding Fig. 4, let us consider the section given by rows $r_i + 1, \dots, r_{i+1}$. We note that the entries of the columns with index $r_i + 3, \dots, r_{i+1} - 2$ are

$1, a_i, b_i, a_i, 1$ which are the coefficients of the difference equation (2.6). On the other hand, it is shown in Section 3 that any function of the form

$$\Upsilon_i(x) = c_1(1 - t) + c_2t + c_3\varphi_i(1 - t) + c_4\varphi_i(t) , \quad (6.4)$$

is a solution for (2.6); therefore if we multiply the row of index $r_i + j$, $j = 1, \dots, n_i - 1$, by $\Upsilon_i(x_{i,j}) = \Upsilon_i(x_i + j\tau_i)$ and then add all these rows, then the contribution of all the columns from $r_i + 3$ to $r_{i+1} - 2$ sums up to zero. The idea for obtaining the matrix \mathbf{K} from \mathbf{A} is the following: we replace the four rows of index $r_i + 1, r_i + 2, r_i + 3, r_i + 4$ with the sum of the rows from $r_i + 1$ to r_{i+1} multiplied by the values assumed in x_{ij} by four linearly independent functions of the form (6.4). The remaining question is how to choose these functions. Several numerical experiments have shown that the lowest condition number of the matrix \mathbf{K} (which is in general larger than that of \mathbf{A}) is achieved when we use the cardinal functions for Lagrange interpolation at the points $x_{i\nu}$ closest to $x_i, x_i + h_i/3, x_{i+1} - h_i/3, x_{i+1}$.

7. Graphical Examples

The aim of this final section is to illustrate the tension features of discrete hyperbolic tension splines with some (famous) examples. Before, we want to notice that the continuous form U_i of our solution given in (3.4) has the good shape-preserving properties of cubics (see e.g. [25]) in the sense that U_i is convex (concave) in $[x_i, x_{i+1}]$ if and only if $m_{i+j} \geq 0$ (≤ 0), $j = 0, 1$, and has at most one inflection point in $[x_i, x_{i+1}]$. In order to preserve the shape of the data, we therefore simply have to analyze the values $\Lambda_i u_{i,0}$ and $\Lambda_i u_{i,n_i}$ and increase the tension parameters if necessary. All the strategies proposed for the automatic choice of tension parameters in continuous hyperbolic tension spline interpolation can be used in our discrete context, see e.g. [24, 25].

In our first example we have interpolated the *radio chemical* data reported in Table 1. The effects of changing the tension values p_i are depicted in Figs. 6–7. We have adopted a non-uniform mesh, assigning the same number of points (30) to each interval of the main mesh, and imposed *natural* end conditions, that is, following formulas (3.6), $m_0 = m_{N+1} = 0$.

Table 1. *Radio chemical* data:

x_i	7.99	8.09	8.19	8.7	9.2
f_i	0	2.76429E-5	4.37498E-2	0.169183	0.469428

x_i	10	12	15	20
f_i	0.943740	0.998636	0.999916	0.999994

Fig. 6 is obtained setting $p_i = 0$, that is considering the discrete cubic spline interpolating the data. In Fig 7 a new discrete interpolant with $p_0 = p_1 = 300$, $p_i = 15$, $i = 2, \dots, 7$, is displayed for the same data, and the stretching effect of the increase in tension parameters is evident.

In the second example we have taken *Akima's* data of Table 2 and constructed discrete interpolants with 20 points for each interval, with natural

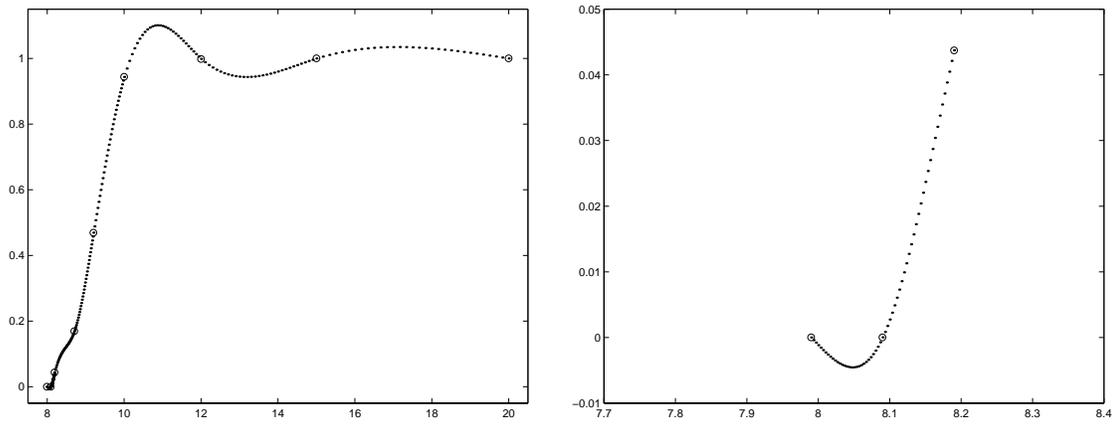


Fig. 6. The radio chemical data with natural end conditions $m_0 = m_{N+1} = 0$
 Interpolation by discrete cubic spline. ($p_i = 0$)
 Right: a magnification of the lower left corner.

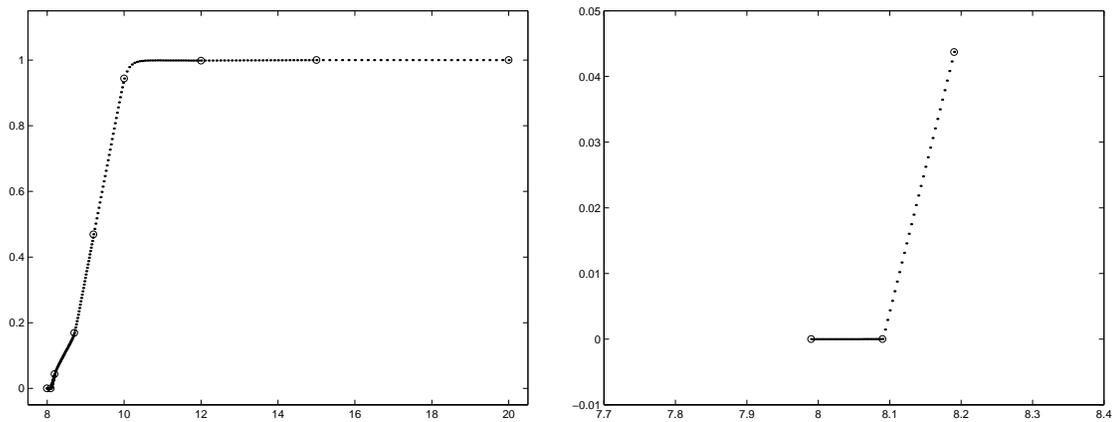


Fig. 7. The same as Fig. 6 with $p_0 = p_1 = 300$, $p_i = 15$, $i = 2, \dots, 7$.

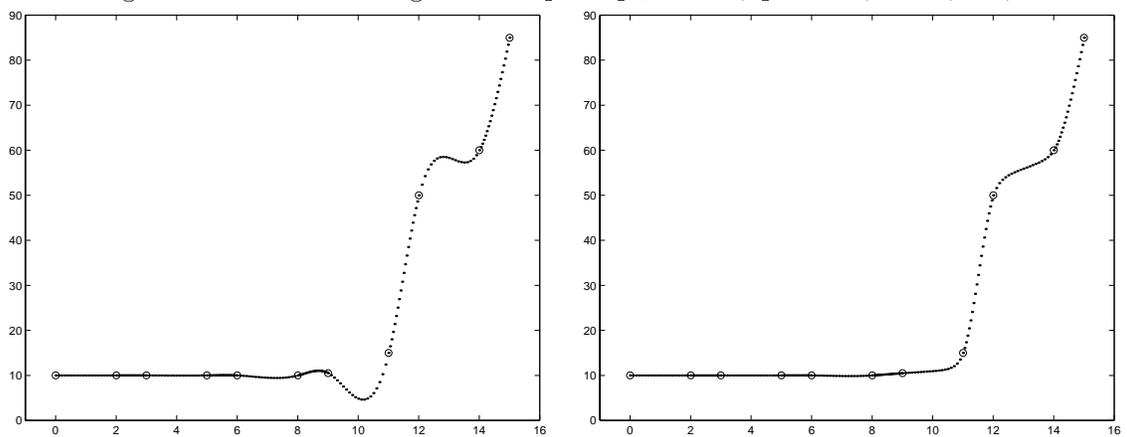


Fig. 8. Akima's data with natural end conditions.
 Left: Discrete interpolating cubic spline ($p_i = 0$)
 Right: discrete hyperbolic spline with $p_5 = p_6 = p_8 = 10$.

end conditions $m_0 = m_{N+1} = 0$. Fig. 8 left shows the plot produced by a uniform choice of tension factors, namely $p_i = 0$. The right part of the same figure shows a second mesh solution, which perfectly reproduces the data shape, where we have set $p_5 = p_6 = p_8 = 10$ while the remaining p_i are unchanged.

Table 2. Akima's data [1]:

x_i	0	2	3	5	6	8	9	11	12	14	15
f_i	10	10	10	10	10	10	10.5	15	50	60	85

Acknowledgments. Work partially supported by MURST. Appreciation is also rendered to the Thailand Research Fund for the financial support which has made this research possible.

References

1. H. Akima, A new method of interpolation and smooth curve fitting based on local procedures, *J. Assoc. Comput. Mech.* **17** (1970) 589–602.
2. de Boor, C. (1976) Splines as linear combinations of B-splines: A survey, *Approximation Theory II*. G. G. Lorentz, C. K. Chui, and L. L. Schumaker (Eds.). Academic Press, New York, 1–47.
3. Cohen, E., T. Lyche, and R. Riesenfeld (1980) Discrete B-splines and subdivision techniques in computer aided geometric design and computer graphics, *Computer Graphics and Image Processing* **14**, 87–111.
4. P. Costantini, B. I. Kvasov and C. Manni, Difference Method for Constructing Hyperbolic Tension Splines, *Rapporto Interno 341/1998*, Università di Siena.
5. W. Dahmen and C. A. Micchelli, On multivariate E-splines, *Advances in Mathematics* **76** (1989) 33–93.
6. R. Delbourgo and J. A. Gregory, Shape preserving piecewise rational interpolation, *SIAM J. Sci. and Statist. Comput.* **6** (1985) 967–976.
7. G. H. Golub and C. F. Van Loan, *Matrix Computations* (John Hopkins University Press, Baltimore, 1991).
8. Hoschek, J. and D. Lasser, *Fundamentals of Computer Aided Geometric Design* (A K Peters, Wellesley, Massachusetts, 1993).
9. N. N. Janenko and B. I. Kvasov, An iterative method for the construction of polycubic spline functions, *Soviet Math. Dokl.* **11** (1970) 1643–1645.
10. P. E. Koch and T. Lyche, Exponential B-splines in tension, in: *Approximation Theory VI: Proceedings of the Sixth International Symposium on Approximation Theory*, Vol. II. Chui C. K., Schumaker L. L., and Ward J. D. (eds.) (Academic Press, Boston, 1989) pp. 361–364.

11. P. E. Koch and T. Lyche, Interpolation with Exponential B-splines in Tension, in: *Geometric Modelling*, Computing/Supplementum 8. Farin G. et al. (eds.) (Springer-Verlag, Wien, 1993) pp. 173–190.
12. B. I. Kvasov, Shape Preserving Spline Approximation via Local Algorithms, in: *Advanced Topics in Multivariate Approximation*, F. Fontanel-la, K. Jetter, and P. J. Laurent (eds.) (World Scientific Publ. Co., Inc., Singapore, 1996) pp. 181–196.
13. B. I. Kvasov, Local bases for generalized cubic splines, *Russ. J. Numer. Anal. Math. Modelling* **10** (1995), 1, 49–80.
14. B. I. Kvasov, GB-splines and their properties, *Annals of Numerical Mathematics* **3** (1996) 139–149.
15. P. J. Laurent, *Approximation et optimization* (Hermann, Paris, 1972).
16. T. Lyche, Discrete cubic spline interpolation, *BIT* **16** (1976) 281–290.
17. M. A. Malcolm, On the computation of nonlinear spline functions, *SIAM J. Numer. Anal.* **14** (1977) 254–282.
18. O. L. Mangasarian and L. L. Schumaker, Discrete splines via mathematical programming, *SIAM J. Control* **9** (1971) 174–183.
19. O. L. Mangasarian and L. L. Schumaker, Best summation formulae and discrete spline, *SIAM J. Numerical Analysis* **10** (1973) 448–459.
20. M. Marušić and M. Rogina, Sharp error bounds for interpolating splines in tension, *J. of Comp. Appl. Math.* **61** (1995) 205–223.
21. A. A. Melkman, Another proof of the total positivity of the discrete spline collocation matrix, *J. Approx. Theory* **84** (1996) 265–273.
22. K. M. Mørken, On total positivity of the discrete spline collocation matrix, *J. Approx. Theory* **84** (1996) 247–264.
23. S. S. Rana and Y. P. Dubey, Local behaviour of the deficient discrete cubic spline interpolator, *J. Approx. Theory* **86** (1996) 120–127.
24. R. J. Renka, Interpolation tension splines with automatic selection of tension factors, *SIAM J. Sci. Stat. Comp.* **8** (1987) 393–415.
25. P. Rentrop, An algorithm for the computation of exponential splines, *Numer. Math.* **35** (1980) 81–93.
26. A. Ron, Exponential Box splines, *Constructive Approximation* **4** (1988) 357–378.
27. N. S. Sapidis and P. D. Kaklis, An algorithm for constructing convexity and monotonicity-preserving splines in tension, *Computer Aided Geometric Design* **5** (1988) 127–137.
28. Schumaker, L. L. (1973) *Constructive aspects of discrete polynomial spline functions*, *Approximation Theory*, G. G. Lorentz (ed.). Academic Press, New York, 469–476.
29. L. L. Schumaker, *Spline functions: Basic theory* (John Wiley & Sons, New York, 1981).

30. D. G. Schweikert, An interpolating curve using a spline in tension, *J. Math. Phys.* **45** (1966) 312–317.
31. Yu. S. Zav'yalov, B. I. Kvasov, and V. L. Miroshnichenko, *Methods of Spline Functions* (Nauka, Moscow, 1980).