On Tension Spline Construction by Difference Method

Boris I. Kvasov*

School of Mathematics, Suranaree University of Technology
111 University Avenue, Nakhon Ratchasima 30000, Thailand
boris@math.sut.ac.th

Abstract. Hyperbolic tension splines are defined as solutions of differential multipoint boundary value problem. For computations we use a difference approximation of that problem. This permits to avoid calculations of hyperbolic functions, however, the extension of a mesh solution will be a discrete tension spline. We consider the basic computational aspects of this approach.

Keywords: Hyperbolic tension splines, multipoint boundary value problem, discrete tension splines, shape preserving interpolation.

1. Introduction

In the theory of splines mainly two approaches are used: algebraic and variational. In the first approach [5] splines are understood as piecewise defined functions with a uniform structure. In the second approach [3] splines are solutions of some minimization problems for linear functionals with restrictions of equality and/or unequality type. But a third approach is also known [2] where splines are defined as solutions of differential multipoint boundary value problems. In some important cases all three approaches give the same solutions. However the third approach has substantial computational advantages which are illustrated here by the example of hyperbolic tension splines.

For the numerical treatment of differential multipoint boundary value problems we replace the differential operator by its difference approximation. This permits us to avoid calculating hyperbolic functions and to find easily mesh solution whose extension will, however, be a discrete tension spline with continuous differences instead of derivatives.

* The author is on leave from the Institute of Computational Technologies, Russian Academy of Sciences, Novosibirsk, 630090, Russia

2. Problem formulation

Let the data

$$(x_i, f_i), \quad i = 0, \ldots, N+1, \tag{2.1}$$

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

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

Interpolating tension spline $S(x)$ with a set of tension parameters $\{p_i \geq 0 \mid i = 0, \ldots, N\}$ is a solution of the differential multipoint boundary value problem

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

$$S \in C^2[a, b], \tag{2.3}$$

with the interpolation conditions

$$S(x_i) = f_i, \quad i = 0, \ldots, N+1 \tag{2.4}$$

and the end constraints

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

For practical purposes it is often more interesting to know the values of the solution over a given tabulation of $[a, b]$ than its global analytic expression. In this paper we do not consider directly a tabulation of S but we study a natural discretization of the previous problem. We prove that the discretized problem has a unique solution, called *mesh solution*, and we study its properties. Of course it turns out that the mesh solution is not a tabulation of S but it can be extended on $[a, b]$ to a function u, with properties very similar to those of S and which approaches S as the discretization step goes to zero. Due to these properties we will refer to u as *discrete* tension spline interpolation of the data (2.1).

Let us assume that each h_i is an integer multiple of the same tabulation step, τ. Putting $n_i = h_i/\tau$, we look for a mesh solution $\bar{u} = \{u_{i,j} \mid j = -1, \ldots, n_i + 1, \quad i = 0, \ldots, N\}$, satisfying the difference equations:

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

where

$$\Lambda u_{i,j} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\tau^2}.$$

The smoothness condition (2.3) is changed for the equations

$$\begin{aligned} u_{i-1,n_{i-1}} &= u_{i,0}, \\ \frac{u_{i-1,n_{i-1}-1} - u_{i-1,n_{i-1}+1}}{2\tau} &= \frac{u_{i,1} - u_{i,-1}}{2\tau}, \quad i = 1, \ldots, N, \\ \Lambda u_{i-1,n_{i-1}} &= \Lambda u_{i,0}, \end{aligned} \tag{2.7}$$

which are equivalent to

$$u_{i-1,n_{i-1}+j} = u_{i,j}, \quad j = -1, 0, 1. \tag{2.8}$$

The interpolation conditions (2.4) take the form

$$u_{i,0} = f_i, \quad u_{i,n_i} = f_{i+1}, \quad i = 0, \ldots, N, \tag{2.9}$$

and for the end conditions (2.5) we have

$$\Lambda u_{0,0} = f_0'' \quad \text{and} \quad \Lambda u_{N,n_N} = f_{N+1}''. \tag{2.10}$$

The equalities (2.8) permit to eliminate the redundant unknowns in the difference equations (2.6). The values $u_{0,-1}$ and u_{N,n_N+1} are not explicitly computed but are introduced into the formulation to accommodate the two necessary end conditions. Putting $m = \sum_{i=0}^{N} n_i + 3$, the previous equations can be collected in the $m \times m$ linear system

$$A\hat{u} = b, \tag{2.11}$$

where

$$A = \begin{bmatrix}
1 & -2 & 1 & 0 & & & & & & & & & & \\
0 & 1 & 0 & 0 & & & & & & & & & & \\
1 & a_0 & b_0 & a_0 & 1 & & & & & & & & & \\
 & 1 & a_0 & b_0 & a_0 & 1 & & & & & & & & \\
 & & & & \cdots & & & & & & & & & \\
 & & 1 & a_0 & b_0 & a_0 & 1 & & & & & & & \\
 & & & 0 & 0 & 1 & 0 & 0 & & & & & & \\
 & & & & 1 & a_1 & b_1 & a_1 & 1 & & & & & \\
 & & & & & 1 & a_1 & b_1 & a_1 & 1 & & & & \\
 & & & & & & & \cdots & & & & & & \\
 & & & & & & & & \cdots & & & & & \\
 & & & & & & & & & \cdots & & & & \\
 & & & & & & & & 1 & a_N & b_N & a_N & 1 & \\
 & & & & & & & & & 1 & a_N & b_N & a_N & 1 \\
 & & & & & & & & & & 0 & 0 & 1 & 0 \\
 & & & & & & & & & & 0 & 1 & -2 & 1
\end{bmatrix},$$

$$a_i = -(4 + \omega_i), \quad b_i = 6 + 2\omega_i, \quad \omega_i = \left(\frac{p_i \tau}{h_i}\right)^2, \quad i = 0, \ldots, N,$$

and

$$\hat{u} = (u_{0,-1}, u_{00}, u_{01}, \ldots, u_{0n_0-1}, u_{10}, \ldots, u_{20}, \ldots, u_{N0}, \ldots, u_{Nn_N}, u_{Nn_N+1})^T,$$

$$b = (\tau^2 f_0'', f_0, 0, \ldots, 0, f_1, \ldots, f_2, \ldots, f_N, \ldots, f_{N+1}, \tau^2 f_{N+1}'')^T.$$

In the previous system the unknowns $u_{i,0}$, $i = 0, \ldots, N+1$, can be immediately determined from the interpolation conditions while the expression of $u_{0,-1}$ (u_{N,n_N+1}) can be obtained from the first (last) equation and substituted in the third (third to last) equation. Then in practice we deal with the $m^* \times m^*$ linear system ($m^* = m - N - 4$)

$$A^* u^* = b^*, \tag{2.12}$$

where

$$A^* = \begin{bmatrix} b_0-1 & a_0 & 1 & & & & & & & & & & \\ a_0 & b_0 & a_0 & 1 & & & & & & & & & \\ 1 & a_0 & b_0 & a_0 & 1 & & & & & & & & \\ & & \cdots & & & & & & & & & & \\ & 1 & a_0 & b_0 & a_0 & & & & & & & & \\ & & 1 & a_0 & b_0 & 1 & & & & & & & \\ & & & & 1 & b_1 & a_1 & 1 & & & & & \\ & & & & & a_1 & b_1 & a_1 & 1 & & & & \\ & & & & & & & \cdots & & & & & \\ & & & & & & 1 & a_N & b_N & a_N & 1 \\ & & & & & & & 1 & a_N & b_N & a_N \\ & & & & & & & & 1 & a_N & b_N-1 \end{bmatrix},$$

and u^*, b^* are correspondingly deduced from $\hat{u}$ and b.

Following [4] we observe that

$$A^* = C^* + D^*,$$

where both C^* and D^* are symmetric block diagonal matrices; to be more specific,

$$C^* = \begin{bmatrix} C_0 & & & & & \\ & C_1 & & & & \\ & & C_2 & & & \\ & & & \ddots & & \\ & & & & & C_N \end{bmatrix},$$

$$C_i = \begin{bmatrix} b_i-1 & a_i & 1 & & & & & \\ a_i & b_i & a_i & 1 & & & & \\ 1 & a_i & b_i & a_i & 1 & & & \\ & & & \cdots & & & & \\ & & & & \cdots & & & \\ & & 1 & a_i & b_i & a_i & 1 \\ & & & 1 & a_i & b_i & a_i \\ & & & & 1 & a_i & b_i-1 \end{bmatrix}, \tag{2.13}$$

$$
D^* = \begin{bmatrix}
0 & 0 & & & & & & & & & & & & & & \\
0 & 0 & & & & & & & & & & & & & & \\
& & \cdot & & & & & & & & & & & & & \\
& & & 0 & & & & & & & & & & & & \\
& & & & 1 & 1 & & & & & & & & & & \\
& & & & 1 & 1 & & & & & & & & & & \\
& & & & & & 0 & & & & & & & & & \\
& & & & & & & \cdot & & & & & & & & \\
& & & & & & & & 0 & & & & & & & \\
& & & & & & & & & 1 & 1 & & & & & \\
& & & & & & & & & 1 & 1 & & & & & \\
& & & & & & & & & & & 0 & & & & \\
& & & & & & & & & & & & \cdot & & & \\
& & & & & & & & & & & & & 0 & & \\
& & & & & & & & & & & & & & 0 & 0 \\
& & & & & & & & & & & & & & 0 & 0
\end{bmatrix}
$$

Since the eigenvalues of D^* are 0 and 2, from a corollary of the Courant-Fisher theorem [1] we have that the eigenvalues of A^*, $\lambda_k(A^*)$, satisfy the following inequalities

$$\lambda_k(A^*) \geq \lambda_k(C^*), \quad k = 1, \dots, m^*.$$

The eigenvalues of C^* are the collection of the eigenvalues of C_i and we have

$$C_i = B_i^2 - \omega_i B_i,$$

where B_i is the $(n_i - 1) \times (n_i - 1)$ tridiagonal matrix

$$
B_i = \begin{bmatrix}
-2 & 1 & & & & \\
1 & -2 & 1 & & & \\
& 1 & -2 & 1 & & \\
& & & \dots & & \\
& & & 1 & -2 & 1 \\
& & & & 1 & -2
\end{bmatrix}.
$$

It is well known, (see also [4]), that

$$\lambda_j(B_i) = -2\Big(1 - \cos\frac{j\pi}{n_i}\Big), \; j = 1, \dots, n_i - 1,$$

then

$$\lambda_j(C_i) = 4\Big(1 - \cos\frac{j\pi}{n_i}\Big)^2 + 2\omega_i\Big(1 - \cos\frac{j\pi}{n_i}\Big).$$

It follows that

$$\lambda_k(A^*) \geq \min_{i,j} \lambda_j(C_i) = \min_i \Big[4\Big(1 - \cos\frac{\tau\pi}{h_i}\Big)^2 + 2\omega_i\Big(1 - \cos\frac{\tau\pi}{h_i}\Big)\Big].$$

Hence, A^* is a positive matrix and the linear system (2.12) (and (2.11) as well) has unique solution.

In addition, from Gershgorin's theorem, $\lambda_k(A^*) \leq \max_i[16 + 4\omega_i]$, then for the condition number, $\mu_2(A^*)$, with respect to the 2 norm of A^*, we have the following upper bound not depending on the number of data points, $N+2$:

$$\begin{aligned}\mu_2(A^*) &\leq \frac{\max_i\left[16 + 4(\frac{\tau p_i}{h_i})^2\right]}{\min_i\left[4(1-\cos\frac{\tau\pi}{h_i})^2 + 2(\frac{\tau p_i}{h_i})^2(1-\cos\frac{\tau\pi}{h_i})\right]} \\ &\leq \frac{\max_i\left[16 + 4(\frac{\tau p_i}{h_i})^2\right]}{\min_i(\frac{\tau}{h_i})^4[\pi^4 + (\pi p_i)^2]}.\end{aligned} \tag{2.14}$$

We remark that, for $p_i = 0, \;\; i = 0, \ldots, N$, we recover the results presented in [4].

From the structure of A^*, the linear system (2.12) can be solved efficiently using a direct method for band matrices. Since A^* is positive band matrix of band width 2, the classical Cholesky factorization , $A^* = LL^T$, provides a lower triangular band matrix L of band 2 and it can be performed in $O(2m^*)$ operations, [1].

3. System Splitting and Mesh Solution Extension

In order to solve numerically the differential multipoint boundary value problem (2.2)–(2.5) we consider the system of difference equations (2.6) completed with the smoothness conditions (2.7) (or (2.8)), interpolation conditions (2.9) and end conditions (2.10).

In the notation

$$M_{ij} = \Lambda u_{ij}, \quad j = 0, \ldots, n_i, \quad i = 0, \ldots, N, \tag{3.1}$$

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

$$\begin{aligned}&M_{i0} = M_i, \\ &\frac{M_{ij-1} - 2M_{ij} + M_{ij+1}}{\tau^2} - \left(\frac{p_i}{h_i}\right)^2 M_{ij} = 0, \;\; j = 1, \ldots, n_i - 1, \\ &M_{i,n_i} = M_{i+1},\end{aligned} \tag{3.2}$$

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

$$M_{ij} = m_i(x_{ij}), \quad x_{ij} = x_i + j\tau, \quad j = 0, \ldots, n_i$$

with

$$\begin{aligned}&m_i(x) = M_i \frac{\sinh k_i(1-t)}{\sinh(k_i)} + M_{i+1}\frac{\sinh k_i t}{\sinh(k_i)}, \\ &t = \frac{x - x_i}{h_i}, \quad \frac{2}{\hat{\tau}_i}\sinh\frac{k_i\hat{\tau}_i}{2} = p_i \geq 0, \quad \hat{\tau}_i = \frac{\tau}{h_i}.\end{aligned}$$

From the equation (3.1) and the interpolation conditions (2.9) we have

$$\begin{aligned} u_{i0} &= f_i, \\ \frac{u_{ij-1} - 2u_{ij} + u_{ij+1}}{\tau^2} &= M_{ij}, \ j = 1, \ldots, n_i - 1, \\ u_{i,n_i} &= f_{i+1}. \end{aligned} \tag{3.3}$$

Let us consider the function

$$u_i(x) = f_i(1-t) + f_{i+1}t + \hat{\varphi}_i(1-t)h_i^2 M_i + \hat{\varphi}_i(t)h_i^2 M_{i+1}, \tag{3.4}$$

where

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

The function $u_i(x)$ satisfies to the conditions

$$u_i(x_j) = f_j, \quad \Lambda u_i(x_j) = M_j, \quad j = i, i+1,$$

where

$$\Lambda u_i(x) = \frac{u_i(x-\tau) - 2u_i(x) + u_i(x+\tau)}{\tau^2}.$$

The mesh restriction of the function $u_i(x)$ gives us the solution of the system (3.3) with $u_{ij} = u_i(x_{ij})$, $j = 0, \ldots, n_i$. The smoothness conditions (2.7) can be rewritten as

$$\begin{aligned} u_{i-1}(x_i) &= u_i(x_i), \\ \Delta_{\bar{\tau},\tau} u_{i-1}(x_i) &= \Delta_{\bar{\tau},\tau} u_{i-1}(x_i), \\ \Lambda u_{i-1}(x_i) &= \Lambda u_i(x_i), \end{aligned} \tag{3.5}$$

where

$$\Delta_{\bar{\tau},\tau} u_j(x) = \frac{u_j(x+\tau) - u_j(x-\tau)}{2\tau};$$

(3.5) are equivalent to

$$u_{i-1}(x_i + j\tau) = u_i(x_i + j\tau), \quad j = -1, 0, 1.$$

Using (3.4) and the second condition (3.5) we obtain a linear system with 3-diagonal matrix

$$\begin{aligned} M_0 &= f_0'', \\ \alpha_{i-1}h_{i-1}M_{i-1} + (\beta_{i-1}h_{i-1} + \beta_i h_i)M_i + \alpha_i h_i M_{i+1} &= d_i, \ i = 1, \ldots, N, \\ M_{N+1} &= f_{N+1}'', \end{aligned} \tag{3.6}$$

where

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

Using an expansion of the hyperbolic functions in the above expressions as power series we obtain

$$\beta_i > 2\alpha_i > 0, \quad i = 0, \ldots, N, \quad \text{for all} \quad \tau > 0, \quad p_i > 0.$$

Therefore the system (3.6) is diagonal dominant and has a unique solution.

We can now conclude that the function $u(x)$ which coincides with $u_i(x)$ for $x \in [x_i, x_{i+1}]$, $i = 0, 1, \ldots, N$, is a *discrete tension interpolation spline.* A mesh restriction of the spline $u(x)$ gives us a solution of the system (2.6). The spline $u(x)$ can also be easy recovered from the solution of the system (2.6).

Instead of looks for a direct solution for the system (2.6) we recommend the following algorithm.

Step 1. *Solve 3-diagonal system (3.6) for* M_i, $i = 1, \ldots, N$.
Step 2. *Solve* $N + 1$ *3-diagonal systems (3.2) for* M_{ij}, $j = 1, \ldots, n_i - 1$, $i = 0, \ldots, N$,
Step 3. *Solve* $N + 1$ *3-diagonal systems (3.3) for* u_{ij}, $j = 1, \ldots, n_i - 1$, $i = 0, \ldots, N$.

Steps 2 and 3 can be replaced by a direct splitting of the system (2.11) into $N + 1$ systems with 5-diagonal matrices

$$C_i u_i = c_i, \quad i = 0, \ldots, N, \tag{3.7}$$

where the $(n_i - 1) \times (n_i - 1)$ matrix C_i has the form (2.13),

$$u_i = (u_{i1}, u_{i2}, \ldots, u_{i,n_i-1})^T,$$
$$c_i = \big((2 + \omega_i) f_i - M_i, -f_i, 0, \ldots, 0, -f_{i+1}, (2 + \omega_i) f_{i+1} - M_{i+1}\big)^T.$$

The calculations to solve the systems (3.2) and (3.3) or (3.7) can be performed by using a multi-processing parallel computer system. If $n_i = n$ for all i, we can first store a triangular factorization of the matrices of the systems and then use parallel computations.

References

1. Golub, G. H. and C. F. Van Loan (1991), *Matrix Computations*, John Hopkins University Press, Baltimore.
2. Janenko, N. N., B.I.Kvasov (1970) An iterative method for the construction of polycubic spline functions. Soviet Math. Dokl. **11**, 1643–1645.
3. Laurent, P. J. (1972) *Approximation et optimization*, Hermann, Paris.
4. Malcolm, M. A. (1977), On the computation of nonlinear spline functions, SIAM J. Numer. Anal. **14**, 254–282.
5. Schumaker, L. L. (1981) *Spline functions: Basic theory*, John Wiley & Sons, New York.