



Parallel mesh methods for tension splines

Boris Kvasov*

Institute of Computational Technologies, Russian Academy of Sciences, Lavrentyev Avenue 6, Novosibirsk, 630090, Russia

ARTICLE INFO

Keywords:

Shape preserving interpolation
DMBVP
Hyperbolic and thin plate tension splines
Superposition principle
Parallel Gaussian elimination
Finite-difference schemes in fractional steps

ABSTRACT

This paper addresses the problem of shape preserving spline interpolation formulated as a differential multipoint boundary value problem (DMBVP for short). Its discretization by mesh method yields a five-diagonal linear system which can be ill-conditioned for unequally spaced data. Using the superposition principle we split this system in a set of tridiagonal linear systems with a diagonal dominance. The latter ones can be stably solved either by direct (Gaussian elimination) or iterative methods (SOR method and finite-difference schemes in fractional steps) and admit effective parallelization. Numerical examples illustrate the main features of this approach.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In practical applications we consider the problem of passing a curve/surface through a finite sequence of points. We want the curve/surface to preserve in some sense the shape of the data. Hyperbolic tension splines (see [1–12]) and thin plate splines [13–17] remain very popular tools to solve this problem. Unfortunately, it is difficult to evaluate such splines especially for small and large tension parameters. At present, efficient methods to evaluate exponential functions are the so-called (nonstationary) subdivision schemes (see [18–20]).

A method for constructing interpolation splines by solving differential multipoint boundary value problems (DMBVP) with subsequent discretization was described in [1,3,17,4]. In comparison with the standard approach [12], this method does not involve hyperbolic/biharmonic function evaluation, but requires the solution of a five-diagonal system, which can be ill-conditioned for unequally spaced data (see [9]).

It is shown below that this system can be split into diagonally dominant tridiagonal linear ones, which can also be solved without hyperbolic function evaluation and admit effective parallelization based on the superposition principle. The latter systems can be efficiently treated by the Gaussian elimination or by iterative methods such as successive overrelaxation (SOR) or finite-difference schemes in fractional steps [21].

The content of this paper is as follows. In Section 2 we formulate the 1D problem. In Section 3, we give its finite-difference approximation and show how to eliminate the “extra” unknowns. System splitting based on the superposition principle is considered in Section 4. The basic steps of 1D algorithm are summarized in Section 5. Important particular cases of discrete cubic splines and hyperbolic tension splines on a quasiuniform mesh are handled in Sections 6 and 7. In both cases corresponding linear systems have a diagonal dominance. Section 8 presents a stable parallel algorithm for a five-diagonal system. In Sections 9 and 10 we formulate the 2D problem and give its finite-difference approximation. The algorithm for the numerical solution of 2D problem is described in Section 11. Section 12 gives the SOR iterative method. Section 13 is concerned with a finite-difference scheme in fractional steps and treats its approximation and stability properties. Finally, Section 14 provides some graphical examples to illustrate the main features of discrete hyperbolic and thin plate tension splines.

* Tel.: +1 7 383 3307373; fax: +1 7 383 3306342.

E-mail address: kvasov@post.nsu.ru.

URL: <http://kvasov.nsu.ru/>.

2. 1D problem formulation

Suppose that we are given the data

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

where $a = x_0 < x_1 < \dots < x_{N+1} = b$. Define

$$f[x_i, x_{i+1}] = (f_{i+1} - f_i)/h_i, \quad h_i = x_{i+1} - x_i, \quad i = 0, \dots, N.$$

Data (1) are called monotonically increasing if

$$f[x_i, x_{i+1}] \geq 0, \quad i = 0, \dots, N,$$

and are called convex if

$$f[x_i, x_{i+1}] \geq f[x_{i-1}, x_i], \quad i = 1, \dots, N.$$

The shape preserving interpolation problem consists in constructing a sufficiently smooth function S such that $S(x_i) = f_i$ for $i = 0, \dots, N+1$ and S is monotone/convex on the intervals of monotonicity/convexity of the input data.

Obviously, the solution to the shape preserving interpolation problem is not unique. We seek it in the form of a hyperbolic tension spline.

Definition 1. The hyperbolic interpolation spline S with the set of tension parameters $\{p_i \geq 0 \mid i = 0, \dots, N\}$ is defined as the solution to the DMBVP

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

$$S \in C^2[a, b] \quad (3)$$

with the interpolation conditions

$$S(x_i) = f_i, \quad i = 0, \dots, N+1 \quad (4)$$

and the boundary conditions

$$S''(a) = f''_0 \quad \text{and} \quad S''(b) = f''_{N+1}. \quad (5)$$

Boundary conditions (5) are used for simplicity. They can be replaced by boundary conditions of other types [3].

The second derivative values in the endpoint conditions (5) must be adjusted to the behavior of the data. Otherwise we can obtain an incompatibility with the shape preserving restrictions [3]. For example, we can use the restrictions

$$f''_0 f[x_0, x_1, x_2] \geq 0, \quad f''_{N+1} f[x_{N-1}, x_N, x_{N+1}] \geq 0.$$

If we set $p_i = 0$ for all i in (2), then the solution to problem (2)–(5) is a cubic spline of the class C^2 , which gives a smooth curve but does not always preserve the monotonicity/convexity of the input data. In the limit as $p_i \rightarrow \infty$, we obtain a polygonal line that is shape preserving for the input data but is not smooth. In standard algorithms for automatic selection of the shape parameters p_i (see [3,7,8], etc.), the latter are chosen so that the resulting curve is as much similar to a cubic spline as possible and simultaneously preserves the monotonicity/convexity of the input data.

3. Finite-difference approximation

Consider the discretization of the DMBVP formulated. For this purpose, on each subinterval $[x_i, x_{i+1}]$, we introduce an additional nonuniform mesh

$$x_{i,-1} < x_i = x_{i,0} < x_{i,1} < \dots < x_{i,n_i} = x_{i+1} < x_{i,n_i+1}, \quad n_i \in \mathbb{N}$$

with the steps $h_{ij} = x_{i,j+1} - x_{i,j}$, $j = -1, \dots, n_i$, $i = 0, \dots, N$. We search for a mesh function

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

satisfying the difference equations

$$24u[x_{i,j-2}, \dots, x_{i,j+2}] - 2\left(\frac{p_i}{h_i}\right)^2 u[x_{i,j-1}, x_{ij}, x_{i,j+1}] = 0, \quad j = 1, \dots, n_i - 1, \quad i = 0, \dots, N. \quad (6)$$

The approximation of smoothness conditions (3) gives the relations

$$\begin{aligned} u_{i-1, n_{i-1}} &= u_{i,0}, \\ D_{i-1, n_{i-1}}^1 u_{i-1, n_{i-1}} &= D_{i,0}^1 u_{i,0}, \quad i = 1, \dots, N, \\ D_{i-1, n_{i-1}}^2 u_{i-1, n_{i-1}} &= D_{i,0}^2 u_{i,0}, \end{aligned} \quad (7)$$

where

$$\begin{aligned} D_{ij}^1 u_{ij} &= \lambda_{ij} u[x_{i,j-1}, x_{ij}] + (1 - \lambda_{ij}) u[x_{ij}, x_{i,j+1}], \\ D_{ij}^2 u_{ij} &= 2u[x_{i,j-1}, x_{ij}, x_{i,j+1}], \quad \lambda_{ij} = h_{ij}/(h_{i,j-1} + h_{ij}). \end{aligned}$$

Conditions (4) and (5) are transformed into

$$u_{i,0} = f_i, \quad i = 0, \dots, N, \quad u_{N, n_N} = f_{N+1} \quad (8)$$

and

$$u[x_{0,-1}, x_{0,0}, x_{0,1}] = f_0'', \quad u[x_{N, n_N-1}, x_{N, n_N}, x_{N, n_N+1}] = f_{N+1}''. \quad (9)$$

Relations (7) and boundary conditions (9) make it possible to eliminate the “extra” unknowns $u_{i,-1}$ and u_{i, n_i+1} , $i = 0, \dots, N$. To show this we use the notation

$$M_i = 2u[x_{i-1, n_{i-1}-1}, x_{i-1, n_{i-1}}, x_{i-1, n_{i-1}+1}] = 2u[x_{i,-1}, x_{i,0}, x_{i,1}].$$

Multiplying these equalities by $h_{i-1, n_{i-1}-1}/2$ and $h_{i,0}/2$, respectively, we rewrite them in the form

$$\begin{aligned} D_{i-1, n_{i-1}}^1 u_{i-1, n_{i-1}} &= u[x_{i-1, n_{i-1}-1}, x_{i-1, n_{i-1}}] + \frac{h_{i-1, n_{i-1}-1}}{2} M_i, \\ D_{i,0}^1 u_{i,0} &= u[x_{i,0}, x_{i,1}] - \frac{h_{i,0}}{2} M_i. \end{aligned}$$

Using the second equality in (7) we obtain

$$M_i = 2u[x_{i-1, n_{i-1}-1}, x_{i,0}, x_{i,1}], \quad i = 1, \dots, N. \quad (10)$$

Thus the second divided differences in Eq. (6) of the form

$$u[x_{i-1, n_{i-1}-1}, x_{i-1, n_{i-1}}, x_{i-1, n_{i-1}+1}] \quad \text{and} \quad u[x_{i,-1}, x_{i,0}, x_{i,1}]$$

can be replaced by $u[x_{i-1, n_{i-1}-1}, x_{i,0}, x_{i,1}]$. This permits us to eliminate the unknowns $u_{i-1, n_{i-1}+1}$ and $u_{i,-1}$, $i = 1, \dots, N$. The unknowns $u_{0,-1}$ and u_{N, n_N+1} are eliminated from boundary conditions (9). The discrete mesh solution is defined as

$$\{u_{ij}, j = 0, \dots, n_i, i = 0, \dots, N\}. \quad (11)$$

The existence and uniqueness conditions of a solution to linear system (6)–(9) will be obtained below.

4. System splitting and mesh solution evaluation

To solve system (6)–(9), we introduce the notation

$$M_{ij} = 2u[x_{i,j-1}, x_{ij}, x_{i,j+1}], \quad j = 0, \dots, n_i; \quad i = 0, \dots, N. \quad (12)$$

Then difference Eq. (6) on the interval $[x_i, x_{i+1}]$ become

$$\begin{aligned} M_{i,0} &= M_i, \\ \frac{12}{x_{i,j+2} - x_{i,j-2}} \left(\frac{M_{i,j+1} - M_{ij}}{x_{i,j+2} - x_{i,j-1}} - \frac{M_{ij} - M_{i,j-1}}{x_{i,j+1} - x_{i,j-2}} \right) - \left(\frac{p_i}{h_i} \right)^2 M_{ij} &= 0, \quad j = 1, \dots, n_i - 1, \\ M_{i, n_i} &= M_{i+1}, \end{aligned} \quad (13)$$

where M_i and M_{i+1} are given numbers.

By virtue of (12) and in view of interpolation conditions (8), we have

$$\begin{aligned} u_{i,0} &= f_i, \\ \frac{2}{h_{i,j-1} + h_{ij}} \left(\frac{u_{i,j+1} - u_{ij}}{h_{ij}} - \frac{u_{ij} - u_{i,j-1}}{h_{i,j-1}} \right) &= M_{ij}, \quad j = 0, \dots, n_i, \\ u_{i, n_i} &= f_{i+1}. \end{aligned} \quad (14)$$

The matrices of systems (13) and (14) are diagonally dominant. Thus these systems have a unique solution that can be stably found by tridiagonal Gaussian elimination.

To solve systems (13) and (14), we only need to determine the values M_i , $i = 0, \dots, N+1$ so that conditions (7) and (9) are satisfied. For this we use a *superposition principle*: the general solution to the linear system is a sum of the general solution to the homogeneous system and the particular solution to the nonhomogeneous system.

Let α_{ij} and β_{ij} , $j = 0, \dots, n_i$ be the solutions to system (13) with the boundary conditions $M_{i,0} = 1$, $M_{i,n_i} = 0$ and $M_{i,0} = 0$, $M_{i,n_i} = 1$, respectively. Then the solution to system (13) can obviously be written as

$$M_{ij} = M_i \alpha_{ij} + M_{i+1} \beta_{ij}, \quad j = 0, \dots, n_i. \quad (15)$$

Now, let γ_{ij} and δ_{ij} , $i = 0, \dots, n_i$ be the solutions to system (14) with the zero boundary conditions $u_{i,0} = u_{i,n_i} = 0$ and the right-hand sides α_{ij} and β_{ij} , respectively, and let ε_{ij} be the solution to homogeneous system (14) with the boundary conditions $\varepsilon_{i,0} = f_i$ and $\varepsilon_{i,n_i} = f_{i+1}$. Since

$$\varepsilon_{ij} = f_i(1 - t_{ij}) + f_{i+1}t_{ij}, \quad j = 0, \dots, n_i,$$

where $t_{ij} = (x_{ij} - x_i)/h_i$, we find for the solution to system (14) that

$$u_{ij} = f_i(1 - t_{ij}) + f_{i+1}t_{ij} + M_i \gamma_{ij} + M_{i+1} \delta_{ij}, \quad j = 0, \dots, n_i. \quad (16)$$

As the matrices of systems (13) and (14) are diagonally dominant and monotonous then applying the maximum principle and the comparison theorem [22], we obtain

$$\begin{aligned} 0 \leq \alpha_{ij} \leq 1, \quad 0 \leq \beta_{ij} \leq 1, \quad 0 \leq \alpha_{ij} + \beta_{ij} \leq 1, \\ -h_i^2/4 \leq \gamma_{ij}, \delta_{ij} \leq 0, \quad j = 0, \dots, n_i, \quad i = 0, \dots, N. \end{aligned}$$

Using the formula (16), substituting $u_{i-1,n_{i-1}-1}$ and $u_{i,1}$ into (10), and taking into account boundary conditions (9), we obtain

$$\begin{aligned} M_0 &= f_0'', \\ a_i M_{i-1} + b_i M_i + c_i M_{i+1} &= d_i, \quad i = 1, \dots, N, \\ M_{N+1} &= f_{N+1}'', \end{aligned} \quad (17)$$

where

$$\begin{aligned} a_i &= -\frac{\gamma_{i-1,n_{i-1}-1}}{h_{i-1,n_{i-1}-1}}, \quad b_i = \frac{h_{i-1,n_{i-1}-1} + h_{i,0}}{2} - \frac{\delta_{i-1,n_{i-1}-1}}{h_{i-1,n_{i-1}-1}} - \frac{\gamma_{i,1}}{h_{i,0}}, \\ c_i &= -\frac{\delta_{i,1}}{h_{i,0}}, \quad d_i = \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}. \end{aligned}$$

The coefficients of system (17) are nonnegative. The system has a diagonal dominance if the following inequalities

$$b_i \geq a_i + c_i, \quad i = 1, \dots, N$$

are satisfied. The last ones are valid if the mesh is chosen so that

$$\frac{h_{i-1,n_{i-1}-1}}{2} - \frac{\delta_{i-1,n_{i-1}-1}}{h_{i-1,n_{i-1}-1}} \geq -\frac{\gamma_{i-1,n_{i-1}-1}}{h_{i-1,n_{i-1}-1}}, \quad \frac{h_{i,0}}{2} - \frac{\gamma_{i,1}}{h_{i,0}} \geq -\frac{\delta_{i,1}}{h_{i,0}}, \quad i = 1, \dots, N. \quad (18)$$

If the inequalities (18) are fulfilled, system (17) has a unique solution. The last one can be found by tridiagonal Gaussian elimination or by some iterative method. For example, we can apply the successive overrelaxation method

$$M_i^{(k+1)} = (1 - \omega)M_i^{(k)} + \frac{\omega}{b_i} \left(d_i - a_i M_{i-1}^{(k+1)} - c_i M_{i+1}^{(k)} \right), \quad i = 1, \dots, N$$

with the relaxation parameter $\omega \in (1, 2)$.

Let the scalars M_i be found for all i . Then system (6)–(9) is split in separate subsystems for each interval $[x_i, x_{i+1}]$, $i = 0, \dots, N$. Systems (13) and (14) are easily transformed to systems with symmetric diagonally dominant matrices that have positive diagonal elements. All their eigenvalues are positive and these matrices are well-conditioned. This provides existence and uniqueness of the mesh solution.

Using the second equality in (7), we put

$$m_i = D_{i-1,n_{i-1}-1}^1 u_{i-1,n_{i-1}} = D_{i,0}^1 u_{i,0}, \quad i = 1, \dots, N.$$

Then

$$m_i = u[x_{i-1,n_{i-1}-1}, x_{i-1,n_{i-1}}] + \frac{h_{i-1,n_{i-1}-1}}{2} M_i = u[x_{i,0}, x_{i,1}] - \frac{h_{i,0}}{2} M_i$$

and

$$m_i = \frac{h_{i,0}}{h_{i-1,n_{i-1}-1} + h_{i,0}} u[x_{i-1,n_{i-1}-1}, x_{i-1,n_{i-1}}] + \frac{h_{i-1,n_{i-1}-1}}{h_{i-1,n_{i-1}-1} + h_{i,0}} u[x_{i,0}, x_{i,1}], \quad i = 1, \dots, N.$$

These equalities permit us to easily find the scalars m_i using mesh solutions. In particular, for the boundary conditions $S'(a) = f'_0$ and $S'(b) = f'_{N+1}$ we obtain

$$\begin{aligned} \left(\frac{h_{0,0}}{2} - \frac{\gamma_{0,1}}{h_{0,0}} \right) M_0 - \frac{\delta_{0,1}}{h_{0,0}} M_1 &= f[x_{0,0}, x_{0,1}] - f'_0, \\ -\frac{\gamma_{N,n_N-1}}{h_{N,n_N-1}} M_N + \left(\frac{h_{N,n_N-1}}{2} - \frac{\delta_{N,n_N-1}}{h_{N,n_N-1}} \right) M_{N+1} &= f'_{N+1} - f[x_N, x_{N+1}]. \end{aligned}$$

These relations can be used instead of the first and last equations in system (17) if at the ends of the interval $[a, b]$, the values of the first derivative interpolation function are given.

5. 1D algorithm

In contrast to the standard approach [1,3,17] based on the direct solution of five-diagonal linear system (6)–(9), the method described in Section 4 splits this system into tridiagonal linear systems whose solution does not require hyperbolic function evaluation. This provides stability and a cheap and easy way of obtaining mesh solution (11).

The basic steps of the algorithm described in Section 4 are as follows.

Step 1. Solve $2(N+1)$ tridiagonal systems (13) with the boundary conditions $M_{i,0} = 1, M_{i,n_i} = 0$ and $M_{i,0} = 0, M_{i,n_i} = 1$ in order to determine α_{ij} and β_{ij} ($j = 0, \dots, n_i, i = 0, \dots, N$), respectively.

Step 2. Solve $2(N+1)$ tridiagonal systems (14) with the homogeneous boundary conditions $u_{i,0} = u_{i,n_i} = 0$ and the right-hand sides α_{ij} and β_{ij} in order to determine γ_{ij} and δ_{ij} ($j = 0, \dots, n_i, i = 0, \dots, N$), respectively.

Step 3. Calculate the parameters M_i ($i = 0, \dots, N+1$) from system (17).

Step 4. Find the values u_{ij} ($j = 0, \dots, n_i, i = 0, \dots, N$) by formula (16).

Steps 1 and 2 in this algorithm can be joined. Instead of $2(N+1)$ tridiagonal systems (13) and (14), we consider $N+1$ five-diagonal systems

$$\begin{aligned} u_{i,0} &= f_i, \quad D_{i,0}^2 u_{i,0} = M_i, \\ 24u[x_{i,j-2}, \dots, x_{i,j+2}] - 2\left(\frac{p_i}{h_i}\right)^2 u[x_{i,j-1}, x_{ij}, x_{i,j+1}] &= 0, \quad j = 1, \dots, n_i - 1, \quad i = 0, \dots, N, \\ u_{i,n_i} &= f_{i+1}, \quad D_{i,n_i}^2 u_{i,n_i} = M_{i+1}. \end{aligned} \quad (19)$$

First, we find the solutions γ_{ij} and δ_{ij} to these systems with the boundary conditions $u_{i,0} = u_{i,n_i} = 0$ and, accordingly, $D_{i,0}^2 u_{i,0} = 1, D_{i,n_i}^2 u_{i,n_i} = 0$ and $D_{i,0}^2 u_{i,0} = 0, D_{i,n_i}^2 u_{i,n_i} = 1$; i.e. $2(N+1)$ five-diagonal systems of form (19) are solved. Next, Steps 3 and 4 are executed to determine the mesh solution u_{ij} ($j = 0, \dots, n_i, i = 0, \dots, N$).

These algorithms can be effectively parallelized using the superposition principle [23,21] and implemented on multiprocessor computers.

6. Case of discrete cubic spline

Let us put to zero all tension parameters: $p_i = 0$ for all i . In this case the solution to system (6) is a cubic polynomial that can be represented as (see [24])

$$\begin{aligned} u_i(x) &= u_i(x_i)(1-t) + u_i(x_{i+1})t + M_i[\Phi_i(x) - \Phi_i(x_i)(1-t)] + M_{i+1}[\Psi_i(x) - \Psi_i(x_{i+1})t], \\ x &\in [x_i, x_{i+1}], \quad i = 0, \dots, N, \end{aligned} \quad (20)$$

where $M_j = D_{j,0}^2 u_i(x_j), j = i, i+1, t = (x - x_i)/h_i$.

Here functions Φ_i and Ψ_i satisfy the conditions

$$\begin{aligned} \Phi_i(x_{i+1} - h_{i,n_i-1}) &= \Phi_i(x_{i+1}) = \Phi_i(x_{i+1} + h_{i,n_i}) = 0, \quad D_{i,0}^2 \Phi_i(x_i) = 1, \\ \Psi_i(x_i - h_{i,-1}) &= \Psi_i(x_i) = \Psi_i(x_i + h_{i,0}) = 0, \quad D_{i,n_i}^2 \Psi_i(x_{i+1}) = 1 \end{aligned}$$

and can be written as

$$\begin{aligned} \Phi_i(x) &= \frac{(x_{i+1} - x - h_{i,n_i-1})(x_{i+1} - x)(x_{i+1} - x + h_{i,n_i})}{2(3h_i + e_{i,n_i} - e_{i,0})}, \\ \Psi_i(x) &= \frac{(x - x_i + h_{i,-1})(x - x_i)(x - x_i - h_{i,0})}{2(3h_i + e_{i,n_i} - e_{i,0})}, \\ e_{ij} &= h_{ij} - h_{i,j-1}, \quad j = 0, n_i. \end{aligned} \quad (21)$$

According to (20) and (21), we put

$$\gamma_i(x) = \Phi_i(x) - \Phi_i(x_i)(1-t) = \frac{t(1-t)(x-x_i-2h_i-e_{i,n_i})}{2(3h_i+e_{i,n_i}-e_{i,0})}h_i^2,$$

$$\delta_i(x) = \Psi_i(x) - \Psi_i(x_{i+1})t = \frac{t(1-t)(x_{i+1}-x-2h_i+e_{i,0})}{2(3h_i+e_{i,n_i}-e_{i,0})}h_i^2.$$

Then in (16) we have

$$\gamma_{ij} = \gamma_i(x_{ij}), \quad \delta_{ij} = \delta_i(x_{ij}), \quad j = 0, \dots, n_i.$$

For convenience instead of the inequalities (18) we consider the ones with respect to the intervals $[x_i, x_{i+1}]$ of the main mesh

$$r_{i,1} = \frac{h_{i,0}}{2} - \frac{\gamma_{i,1}}{h_{i,0}} + \frac{\delta_{i,1}}{h_{i,0}} \geq 0, \quad r_{i,n_i-1} = \frac{h_{i,n_i-1}}{2} + \frac{\gamma_{i,n_i-1}}{h_{i,n_i-1}} - \frac{\delta_{i,n_i-1}}{h_{i,n_i-1}} \geq 0, \quad i = 0, \dots, N.$$

By formulas (34) we obtain

$$r_{i,1} = \frac{h_i}{2} + \frac{2\delta_{i,1}}{h_{i,0}} = \frac{h_i}{2} - \frac{2}{h_i}\Psi_i(x_{i+1}) = \frac{h_i(h_i+e_{i,0}+e_{i,n_i})+2h_{i,0}h_{i,-1}}{2(3h_i+e_{i,n_i}-e_{i,0})},$$

$$r_{i,n_i-1} = \frac{h_i}{2} + \frac{2\gamma_{i,n_i-1}}{h_{i,n_i-1}} = \frac{h_i}{2} - \frac{2}{h_i}\Phi_i(x_i) = \frac{h_i(h_i-e_{i,0}-e_{i,n_i})+2h_{i,n_i-1}h_{i,n_i}}{2(3h_i+e_{i,n_i}-e_{i,0})}.$$

If now $\max(|e_{i,0}|, |e_{i,n_i}|) \leq h_i/2$ for all i , then $r_{i,1} > 0$ and $r_{i,n_i-1} > 0$ and system (17) is diagonally dominant. Hence, the discrete cubic spline exists and is unique.

7. Case of quasiuniform mesh

Let us consider the mesh which is uniform separately on each interval $[x_i, x_{i+1}]$, $i = 0, \dots, N$, i.e. $h_{ij} = \tau_i$ for $j = -1, \dots, n_i$. In this case the solution to system (6) can be represented as (see [1,3])

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

where

$$U_i(x) = f_i(1-t) + f_{i+1}t + \varphi_i(1-t)h_i^2M_i + \varphi_i(t)h_i^2M_{i+1} \quad (22)$$

and

$$\varphi_i(t) = \frac{\sinh(k_it) - t \sinh(k_i)}{p_i^2 \sinh(k_i)}, \quad k_i = 2n_i \ln \left(\frac{p_i}{2n_i} + \sqrt{\left(\frac{p_i}{2n_i} \right)^2 + 1} \right).$$

System (17) can be rewritten as

$$M_0 = f_0'',$$

$$-\frac{\gamma_{i-1,n_{i-1}-1}}{\tau_{i-1}}M_{i-1} + \left(\frac{\tau_{i-1} + \tau_i}{2} - \frac{\delta_{i-1,n_{i-1}-1}}{\tau_{i-1}} - \frac{\gamma_{i,1}}{\tau_i} \right)M_i - \frac{\delta_{i,1}}{\tau_i}M_{i+1} = d_i, \quad i = 1, \dots, N,$$

$$M_{N+1} = f_{N+1}''.$$

According to (22) we have

$$\alpha_i = -\frac{\delta_{i,1}}{\tau_i} = -\frac{1}{\tau_i}\varphi_i(t_{i,1})h_i^2 = -n_i h_i \varphi_i\left(\frac{1}{n_i}\right) = -h_i \frac{n_i \sinh(k_i/n_i) - \sinh(k_i)}{p_i^2 \sinh(k_i)} = -\frac{\gamma_{i,n_i-1}}{\tau_i},$$

$$\beta_i = \frac{\tau_i}{2} - \frac{\gamma_{i,1}}{\tau_i} = \frac{\tau_i}{2} - \frac{1}{\tau_i}\varphi_i(1-t_{i,1})h_i^2 = h_i \left[\frac{1}{2n_i} - n_i \varphi_i\left(1 - \frac{1}{n_i}\right) \right] = \frac{\tau_i}{2} - \frac{\delta_{i,n_i-1}}{\tau_i}.$$

Function φ_i satisfies the conditions

$$\varphi_i(0) = \varphi_i(1) = \Lambda_i \varphi_i(0) = 0, \quad \Lambda_i \varphi_i(1) = h_i^{-2},$$

where

$$\Lambda_i \varphi_i(t) = \frac{\varphi_i(t - \tau_i) - 2\varphi_i(t) + \varphi_i(t + \tau_i)}{\tau_i^2}.$$

It follows from here that

$$\varphi_i\left(-\frac{1}{n_i}\right) + \varphi_i\left(\frac{1}{n_i}\right) = 0, \quad \varphi_i\left(1 - \frac{1}{n_i}\right) + \varphi_i\left(1 + \frac{1}{n_i}\right) = \frac{1}{n_i^2}.$$

Applying the second of these relations we have

$$\beta_i = \frac{h_i n_i}{2} \left[\varphi_i \left(1 + \frac{1}{n_i} \right) - \varphi_i \left(1 - \frac{1}{n_i} \right) \right] = h_i \frac{n_i \cosh(k_i) \sinh(k_i/n_i) - \sinh(k_i)}{p_i^2 \sinh(k_i)}.$$

Using power series expansions of the hyperbolic functions in the above expressions for α_i and β_i , we obtain

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

Therefore, system (23) is diagonally dominant. Hence, it has a unique solution. Eliminating from this system the parameters M_0 and M_{N+1} we obtain a system with a symmetric matrix whose eigenvalues are all positive. Now we can conclude that quasiuniform mesh system (6)–(9) has a unique solution which can be represented in the form $U(x_{ij})$, $j = -1, \dots, n_i + 1$, $i = 0, \dots, N$, while the constants M_i are determined from system (23).

Matrices of systems (13), (14) and (23) are diagonally dominant and can be easily transformed to symmetric matrices with positive diagonal elements that are positive definite (all their eigenvalues are positive) and well-conditioned. Therefore, the solution of these systems can be stably obtained by Gaussian elimination or by different iterative methods (Jacoby, Gauss–Seidel, SOR, etc.) that are all convergent.

8. Parallel algorithm for five-diagonal system

For quasiuniform mesh system (6)–(9) after eliminating the unknowns $u_{i,-1}$, u_{i,n_i+1} , $i = 0, \dots, N$ takes the form (see [1,3])

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \quad (24)$$

where

$$\mathbf{u} = (u_{0,1}, \dots, u_{0,n_0-1}, u_{1,1}, \dots, u_{2,1}, \dots, u_{N,1}, \dots, u_{N,n_N-1})^T,$$

$$\mathbf{b} = (-(a_0 + 2)f_0 - \tau_0^2 f_0'', -f_0, 0, \dots, 0, -f_1, -\gamma_{0,n_0-1}f_1, -\gamma_{1,1}f_1, -f_1, 0, \dots, 0, -f_{N+1}, -(a_N + 2)f_{N+1} - \tau_{N+1}^2 f_{N+1}'')^T$$

with

$$\gamma_{i-1,n_{i-1}-1} = a_{i-1} + 2\frac{\rho_i - 1}{\rho_i}, \quad \gamma_{i,1} = a_i + 2(1 - \rho_i), \quad i = 1, \dots, N$$

and \mathbf{A} is the following five-diagonal matrix

$$\begin{bmatrix} b_0 - 1 & a_0 & 1 & & & & & & & \\ a_0 & b_0 & a_0 & 1 & & & & & & \\ 1 & a_0 & b_0 & a_0 & 1 & & & & & \\ & & & \dots & & & & & & \\ & & 1 & a_0 & b_0 & a_0 & & & & \\ & & & 1 & a_0 & \eta_{0,n_0-1} & \delta_{0,n_0-1} & & & \\ & & & & \delta_{1,1} & \eta_{1,1} & a_1 & 1 & & \\ & & & & & a_1 & b_1 & a_1 & 1 & \\ & & & & & & \dots & & & \\ & & & & & & 1 & a_N & b_N & a_N & 1 \\ & & & & & & & 1 & a_N & b_N & a_N \\ & & & & & & & & 1 & a_N & b_N - 1 \end{bmatrix}$$

with

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

$$\eta_{i-1,n_{i-1}-1} = b_{i-1} + \frac{1 - \rho_i}{1 + \rho_i}, \quad \eta_{i,1} = b_i + \frac{\rho_i - 1}{\rho_i + 1}, \quad \rho_i = \frac{\tau_i}{\tau_{i-1}},$$

$$\delta_{i-1,n_{i-1}-1} = \frac{2}{\rho_i(\rho_i + 1)}, \quad \delta_{i,1} = 2\frac{\rho_i^2}{\rho_i + 1}, \quad i = 1, \dots, N.$$

In [1,3,17] system (24) is solved using five-diagonal Gaussian elimination. In the general case for unequally spaced data this system may be ill-conditioned [9]. To avoid this problem let us consider a parallel algorithm of Gaussian elimination for the solution of system (24).

We cancel equations of system (24) which are most close to the data points x_i or more precisely the equations

$$(b_0 - 1)u_{0,1} + a_0u_{0,2} + u_{0,3} = -(a_0 + 2)f_0 - \tau_0^2 f_0'',$$

$$u_{i-1,n_{i-1}-3} + a_{i-1}u_{i-1,n_{i-1}-2} + \eta_{i-1,n_{i-1}-1}u_{i-1,n_{i-1}-1} + \delta_{i-1,n_{i-1}-1}u_{i,1} = -\gamma_{i-1,n_{i-1}-1}f_i, \quad (25)$$

$$\delta_{i,1}u_{i-1,n_{i-1}-1} + \eta_{i,1}u_{i,1} + a_iu_{i,2} + u_{i,3} = -\gamma_{i,1}f_i, \quad i = 1, \dots, N,$$

$$u_{N,n_N-3} + a_Nu_{N,n_N-2} + (b_N - 1)u_{N,n_N-1} = -(a_N + 2)f_{N+1} - \tau_{N+1}^2 f_{N+1}''.$$

Let numbers $u_{i,1}^{(0)}, u_{i,n_i-1}^{(0)}, i = 0, \dots, N$, be given which correspond to the removed equations. System (24) is split in $N + 1$ subsystems

$$\begin{aligned} u_{i,0} &= f_i, & u_{i,1} &= u_{i,1}^{(0)}, \\ u_{i,j-2} + a_i u_{i,j-1} + b_i u_{ij} + a_i u_{i,j+1} + u_{i,j+2} &= 0, & j &= 2, \dots, n_i - 2, \\ u_{i,n_i-1} &= u_{i,n_i-1}^{(0)}, & u_{i,n_i} &= f_{i+1}. \end{aligned} \quad (26)$$

Let us show that the obtained systems have a unique solution which can be found by usual five-diagonal Gaussian elimination.

We rewrite system (26) as

$$\mathbf{A}_i \mathbf{u}_i = \mathbf{f}_i,$$

where

$$\begin{aligned} \mathbf{u}_i &= (u_{i,2}, u_{i,3}, \dots, u_{i,n_i-2})^T, \\ \mathbf{f}_i &= (-a_i u_{i,1}^{(0)} - f_i, -u_{i,1}^{(0)}, 0, \dots, 0, -u_{i,n_i-1}^{(0)}, -a_i u_{i,n_i-1}^{(0)} - f_{i+1})^T. \end{aligned}$$

The matrix \mathbf{A}_i is symmetric. We observe that

$$\mathbf{A}_i = \mathbf{C}_i + \mathbf{D}_i, \quad \mathbf{C}_i = \mathbf{B}_i^2 - \omega_i \mathbf{B}_i,$$

where

$$\mathbf{B}_i = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}, \quad \mathbf{D}_i = \begin{pmatrix} 1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1 \end{pmatrix}.$$

Since,

$$\lambda_j(\mathbf{B}_i) = -2 \left(1 - \cos \frac{j\pi}{m_i} \right), \quad j = 1, \dots, m_i - 1, \quad m_i = n_i - 2,$$

we have

$$\lambda_j(\mathbf{C}_i) = 4 \left(1 - \cos \frac{j\pi}{m_i} \right)^2 + 2\omega_i \left(1 - \cos \frac{j\pi}{m_i} \right), \quad j = 1, \dots, m_i - 1.$$

In addition, the eigenvalues of \mathbf{D}_i are 0 and 1, thus we deduce from a corollary of the Courant–Fisher theorem [25] that the eigenvalues of \mathbf{A}_i satisfy the following inequalities

$$\lambda_j(\mathbf{A}_i) \geq \lambda_j(\mathbf{C}_i) \geq 4 \left(1 - \cos \frac{\pi}{m_i} \right)^2 + 2\omega_i \left(1 - \cos \frac{\pi}{m_i} \right).$$

Hence, \mathbf{A}_i is a positive matrix and we directly obtain that the five-diagonal linear system has a unique solution which can be stably found by usual five-diagonal Gaussian elimination [25].

We obtain a solution $u_{ij}^{(0)}, j = 0, \dots, n_i, i = 0, \dots, N$.

Using Eqs. (25) let us recalculate the scalars $u_{i,1}^{(0)}, u_{i,n_i-1}^{(0)}, i = 0, \dots, N$. For $i = 1, \dots, N$ we find

$$\begin{aligned} u_{i-1,n_{i-1}-1}^{(1)} &= \frac{1}{\Delta_i} (\eta_{i,1} F_{i,1}^{(0)} - \delta_{i-1,n_{i-1}-1} F_{i,2}^{(0)}), \\ u_{i,1}^{(1)} &= \frac{1}{\Delta_i} (-\delta_{i,1} F_{i,1}^{(0)} + \eta_{i-1,n_{i-1}-1} F_{i,2}^{(0)}), \end{aligned}$$

where

$$\begin{aligned} F_{i,1}^{(0)} &= -\gamma_{i-1,n_{i-1}-1} f_i - a_{i-1} u_{i-1,n_{i-1}-2}^{(0)} - u_{i-1,n_{i-1}-3}^{(0)}, \\ F_{i,2}^{(0)} &= -\gamma_{i,1} f_i - a_i u_{i,2}^{(0)} - u_{i,3}^{(0)}, \\ \Delta_i &= b_{i-1} b_i + (b_i - b_{i-1}) \frac{1 - \rho_i}{1 + \rho_i} - 1. \end{aligned}$$

From first and last equations of system (25) we calculate

$$\begin{aligned} u_{0,1}^{(1)} &= \frac{1}{1 - b_0} ((a_0 + 2)f_0 + \tau_0^2 f_0'' + a_0 u_{0,2}^{(0)} + u_{0,3}^{(0)}), \\ u_{N,n_N-1}^{(1)} &= \frac{1}{1 - b_N} ((a_N + 2)f_{N+1} + \tau_N^2 f_{N+1}'' + a_N u_{N,n_N-2}^{(0)} + u_{N,n_N-3}^{(0)}). \end{aligned}$$

Solving repeatedly system (26) we obtain a solution $u_{ij}^{(1)}, j = 0, \dots, n_i, i = 0, \dots, N$, etc. The calculations show that this algorithm is convergent.

9. 2D problem formulation

Let us consider a rectangular domain $\overline{\Omega} = \Omega \cup \Gamma$ where

$$\Omega = \{(x, y) | a < x < b, c < y < d\}$$

and Γ is the boundary of Ω . We consider on $\overline{\Omega}$ a mesh of lines $\Delta = \Delta_x \times \Delta_y$ with

$$\Delta_x : a = x_0 < x_1 < \dots < x_{N+1} = b,$$

$$\Delta_y : c = y_0 < y_1 < \dots < y_{M+1} = d,$$

which divides the domain $\overline{\Omega}$ into the rectangles $\overline{\Omega}_{ij} = \Omega_{ij} \cup \Gamma_{ij}$ where

$$\Omega_{ij} = \{(x, y) | x \in (x_i, x_{i+1}), y \in (y_j, y_{j+1})\}$$

and Γ_{ij} is the boundary of Ω_{ij} , $i = 0, \dots, N, j = 0, \dots, M$.

Let us associate to the mesh Δ the data

$$(x_i, y_j, f_{ij}), \quad i = 0, \dots, N+1, j = 0, \dots, M+1,$$

$$f_{ij}^{(2,0)}, \quad i = 0, N+1, j = 0, \dots, M+1,$$

$$f_{ij}^{(0,2)}, \quad i = 0, \dots, N+1, j = 0, M+1,$$

$$f_{ij}^{(2,2)}, \quad i = 0, N+1, j = 0, M+1,$$

where

$$f_{ij}^{(r,s)} = \frac{\partial^{r+s} f(x_i, y_j)}{\partial x^r \partial y^s}, \quad r, s = 0, 2.$$

We denote by $C^{2,2}[\overline{\Omega}]$ the set of all continuous functions f on $\overline{\Omega}$ having continuous partial and mixed derivatives up to the order 2 in x and y variables. We call the problem of searching for a function $S \in C^{2,2}[\overline{\Omega}]$ with interpolation conditions $S(x_i, y_j) = f_{ij}$, $i = 0, \dots, N+1, j = 0, \dots, M+1$, and such that S preserves the shape of the initial data *the shape preserving interpolation problem*. This means that wherever the data increase (decrease) monotonically, S has the same behavior, and S is convex (concave) over intervals where the data are convex (concave).

Evidently, the solution of the shape preserving interpolation problem is not unique. We are looking for a solution of this problem as a thin plate tension spline. As usually we denote by Δ the Laplace operator $\Delta S = \partial^2 S / \partial x^2 + \partial^2 S / \partial y^2$.

Definition 2. An interpolating thin plate tension spline S with two sets of tension parameters $\{0 \leq p_{ij} < \infty | i = 0, \dots, N, j = 0, \dots, M+1\}$ and $\{0 \leq q_{ij} < \infty | i = 0, \dots, N+1, j = 0, \dots, M\}$ is a solution of the DMBVP

$$\Delta^2 S - r_{ij} \Delta S = 0 \quad \text{in each } \Omega_{ij}, \quad (27)$$

where

$$r_{ij} = \max \left[\left(\frac{p_{i\alpha}}{h_i} \right)^2, \left(\frac{q_{\beta j}}{l_j} \right)^2 \right], \quad \alpha = j, j+1, \quad \beta = i, i+1,$$

$$h_i = x_{i+1} - x_i, \quad l_j = y_{j+1} - y_j, \quad i = 0, \dots, N, j = 0, \dots, M,$$

and

$$\frac{\partial^4 S}{\partial x^4} - \left(\frac{p_{ij}}{h_i} \right)^2 \frac{\partial^2 S}{\partial x^2} = 0, \quad x \in (x_i, x_{i+1}), i = 0, \dots, N, \quad y = y_j, \quad j = 0, \dots, M+1, \quad (28)$$

$$\frac{\partial^4 S}{\partial y^4} - \left(\frac{q_{ij}}{l_j} \right)^2 \frac{\partial^2 S}{\partial y^2} = 0, \quad y \in (y_j, y_{j+1}), j = 0, \dots, M, \quad x = x_i, \quad i = 0, \dots, N+1, \quad (29)$$

$$S \in C^{2,2}[\overline{\Omega}], \quad (30)$$

with the interpolation conditions

$$S(x_i, y_j) = f_{ij}, \quad i = 0, \dots, N+1, j = 0, \dots, M+1, \quad (31)$$

and the boundary conditions

$$S^{(2,0)}(x_i, y_j) = f_{ij}^{(2,0)}, \quad i = 0, N+1, j = 0, \dots, M+1,$$

$$S^{(0,2)}(x_i, y_j) = f_{ij}^{(0,2)}, \quad i = 0, \dots, N+1, j = 0, M+1, \quad (32)$$

$$S^{(2,2)}(x_i, y_j) = f_{ij}^{(2,2)}, \quad i = 0, N+1, j = 0, M+1.$$

By this definition an interpolating thin plate tension spline S is a set of the interpolating thin plate tension functions which satisfy (27), match up smoothly and form a twice continuously differentiable function both in x and y variables

$$\begin{aligned} S^{(r,0)}(x_i - 0, y) &= S^{(r,0)}(x_i + 0, y), \quad r = 0, 1, 2, \quad i = 1, \dots, N, \\ S^{(0,s)}(x, y_j - 0) &= S^{(0,s)}(x, y_j + 0), \quad s = 0, 1, 2, \quad j = 1, \dots, M. \end{aligned} \quad (33)$$

C^2 smoothness of the interpolating hyperbolic tension splines in (28) and (29) was proven in [1,3]. The computation of the interpolating thin plate tension spline reduces to a computation of infinitely many proper one-dimensional hyperbolic tension splines.

For all $p_{ij}, q_{ij} \rightarrow 0$ the solution of (27)–(32) becomes a thin plate spline [14] while in the limiting case as $p_{ij}, q_{ij} \rightarrow \infty$ in rectangle $\overline{\Omega}_{ij}$ the spline S turns into a linear function separately by x and y , and obviously preserves the shape properties of the data on $\overline{\Omega}_{ij}$. By increasing one or more of tension parameters the surface is pulled toward an inherent shape while at the same time keeping its smoothness. Thus, the DMBVP gives an approach to solve the shape preserving interpolation problem.

10. Finite-difference approximation of DMBVP

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 section, we consider a finite-difference approximation of the DMBVP. This provides a linear system whose solution is called a *mesh solution*. It turns out that the mesh solution is not a tabulation of S but is supposed to be some approximation of it.

Let $n_i, m_j \in \mathbb{N}, i = 0, \dots, N, j = 0, \dots, M$, be given such that

$$\frac{h_i}{n_i} = \frac{l_j}{m_j} = h.$$

We are looking for a mesh function

$$\{u_{ik;jl} | k = -1, \dots, n_i + 1, i = 0, \dots, N; l = -1, \dots, m_j + 1, j = 0, \dots, M\},$$

satisfying the difference equations

$$\left[(\Lambda_1 + \Lambda_2)^2 - r_{ij}(\Lambda_1 + \Lambda_2) \right] u_{ik;jl} = 0, \quad (34)$$

$$k = 1, \dots, n_i - 1, i = 0, \dots, N; l = 1, \dots, m_j - 1, j = 0, \dots, M,$$

$$\left[\Lambda_1^2 - \left(\frac{p_{ij}}{h_i} \right)^2 \Lambda_1 \right] u_{ik;jl} = 0, \quad (35)$$

$$k = 1, \dots, n_i - 1, i = 0, \dots, N; l = 0 \text{ if } j = 0, \dots, M - 1; l = 0, m_M \text{ if } j = M,$$

$$\left[\Lambda_2^2 - \left(\frac{q_{ij}}{l_j} \right)^2 \Lambda_2 \right] u_{ik;jl} = 0, \quad (36)$$

$$k = 0 \text{ if } i = 0, \dots, N - 1; k = 0, n_N \text{ if } i = N; l = 1, \dots, m_j - 1, j = 0, \dots, M, \text{ where}$$

$$\begin{aligned} \Lambda_1 u_{ik;jl} &= \frac{u_{i,k+1;jl} - 2u_{ik;jl} + u_{i,k-1;jl}}{h^2}, \\ \Lambda_2 u_{ik;jl} &= \frac{u_{ik;j,l+1} - 2u_{ik;jl} + u_{ik;j,l-1}}{h^2}. \end{aligned}$$

The smoothness conditions (30) are changed to

$$\begin{aligned} u_{i-1,n_{i-1};jl} &= u_{i,0;jl}, \\ \frac{u_{i-1,n_{i-1}+1;jl} - u_{i-1,n_{i-1}-1;jl}}{2h} &= \frac{u_{i1;jl} - u_{i,-1;jl}}{2h}, \end{aligned} \quad (37)$$

$$\Lambda_1 u_{i-1,n_{i-1};jl} = \Lambda_1 u_{i,0;jl}, \quad i = 1, \dots, N, \quad l = 0, \dots, m_j, \quad j = 0, \dots, M,$$

$$u_{ik;j-1,m_{j-1}} = u_{ik;j,0},$$

$$\frac{u_{ik;j-1,m_{j-1}+1} - u_{ik;j-1,m_{j-1}-1}}{2h} = \frac{u_{ik;j,1} - u_{ik;j,-1}}{2h}, \quad (38)$$

$$\Lambda_2 u_{ik;j-1,m_{j-1}} = \Lambda_2 u_{ik;j,0}, \quad k = 0, \dots, n_i, \quad i = 0, \dots, N, \quad j = 1, \dots, M.$$

Conditions (31) and (32) take the form

$$\begin{aligned} u_{i,0;j,0} &= f_{ij}, & u_{N,n_N;j,0} &= f_{N+1,j}, \\ u_{i,0;M,m_M} &= f_{i,M+1}, & u_{N,n_N;M,m_M} &= f_{N+1,M+1}, \quad i = 0, \dots, N, \quad j = 0, \dots, M, \end{aligned} \quad (39)$$

and

$$\begin{aligned} \Lambda_1 u_{0,0;j,0} &= f_{0,j}^{(2,0)}, \quad j = 0, \dots, M; \\ \Lambda_1 u_{0,0;M,m_M} &= f_{0,M+1}^{(2,0)}, \\ \Lambda_1 u_{N,n_N;j,0} &= f_{N+1,j}^{(2,0)}, \quad j = 0, \dots, M; \\ \Lambda_1 u_{N,n_N;M,m_M} &= f_{N+1,M+1}^{(2,0)}, \\ \Lambda_2 u_{i,0;0,0} &= f_{i,0}^{(0,2)}, \quad i = 0, \dots, N; \\ \Lambda_2 u_{N,n_N;0,0} &= f_{N+1,0}^{(0,2)}, \\ \Lambda_2 u_{i,0;M,m_M} &= f_{i,M+1}^{(0,2)}, \quad i = 0, \dots, N; \\ \Lambda_2 u_{N,n_N;M,m_M} &= f_{N+1,M+1}^{(0,2)}, \\ \Lambda_1 \Lambda_2 u_{0,0;0,0} &= f_{0,0}^{(2,2)}, & \Lambda_1 \Lambda_2 u_{N,n_N;0,0} &= f_{N+1,0}^{(2,2)}, \\ \Lambda_1 \Lambda_2 u_{0,0;M,m_M} &= f_{0,M+1}^{(2,2)}, & \Lambda_1 \Lambda_2 u_{N,n_N;M,m_M} &= f_{N+1,M+1}^{(2,2)}. \end{aligned} \quad (40)$$

11. 2D algorithm

To solve finite-difference system (34)–(40) we propose first to find its solution on the refinement of the main mesh Δ . The latter can be achieved in four steps.

First step. Evaluate all tension parameters p_{ij} on the lines $y = y_j, j = 0, \dots, M + 1$ and q_{ij} on the lines $x = x_i, i = 0, \dots, N + 1$ by one of 1D algorithms for an automatic selection of shape control parameters; see, e.g., [3,7,8], etc.

Second step. Construct discrete hyperbolic tension splines [1] in the x direction by solving the $M + 2$ linear systems (35). As a result, one finds the values of the mesh solution on the lines $y = y_j, j = 0, \dots, M + 1$ of the mesh Δ in the x direction.

Third step. Construct discrete hyperbolic tension splines in the y direction by solving the $N + 2$ linear systems (36). This gives us the values of the mesh solution on the lines $x = x_i, i = 0, \dots, N + 1$ of the mesh Δ in the y direction.

Fourth step. Construct discrete hyperbolic tension splines in the x and y directions interpolating the data $f_{ij}^{(2,0)}, i = 0, N + 1, j = 0, \dots, M + 1$, and $f_{ij}^{(0,2)}, i = 0, \dots, N + 1, j = 0, M + 1$, on the boundary Γ . This gives us the values

$$\begin{aligned} \Lambda_1 u_{0,0;j,l}, & \quad \Lambda_1 u_{N,n_N;j,l}, \quad l = 0, \dots, m_j, \quad j = 0, \dots, M, \\ \Lambda_2 u_{i,k;0,0}, & \quad \Lambda_2 u_{i,k;M,m_M}, \quad k = 0, \dots, n_i, \quad i = 0, \dots, N. \end{aligned} \quad (41)$$

Now the system of difference equation (34)–(40) can be substantially simplified by eliminating the unknowns

$$\begin{aligned} u_{i,k;j,l}, \quad k &= -1, n_i + 1, \quad i = 0, \dots, N, \quad l = 0, \dots, m_j, \quad j = 0, \dots, M, \\ u_{i,k;j,l}, \quad k &= 0, \dots, n_i, \quad i = 0, \dots, N, \quad l = -1, m_j + 1, \quad j = 0, \dots, M, \end{aligned}$$

using relations (37) and (38), and the boundary values (40).

As a result one obtains a system with $(n_i - 1)(m_j - 1)$ difference equations and the same number of unknowns in each rectangle $\Omega_{ij}, i = 0, \dots, N, j = 0, \dots, M$. This linear system can be efficiently solved by the SOR algorithm or applying finite-difference schemes in fractional steps on single- or multiprocessor computers.

12. SOR iterative method

Using a piecewise linear interpolation of the mesh solution from the main mesh Δ onto the refinement let us define a mesh function

$$\{u_{i,k;j,l}^{(0)} | k = 0, \dots, n_i, \quad i = 0, \dots, N, \quad l = 0, \dots, m_j, \quad j = 0, \dots, M\}. \quad (42)$$

In each rectangle $\Omega_{ij}, i = 0, \dots, N, j = 0, \dots, M$, the difference equation (34) can be rewritten in the componentwise form

$$\begin{aligned} u_{i,k;j,l} &= \frac{1}{\alpha_{ij}} \left\{ \beta_{ij} [u_{i,k-1;j,l} + u_{i,k+1;j,l} + u_{i,k;j,l-1} + u_{i,k;j,l+1}] - 2[u_{i,k-1;j,l-1} + u_{i,k-1;j,l+1} + u_{i,k+1;j,l-1} + u_{i,k+1;j,l+1}] \right. \\ &\quad \left. - u_{i,k;j,l-2} - u_{i,k;j,l+2} - u_{i,k-2;j,l} - u_{i,k+2;j,l} \right\}, \end{aligned} \quad (43)$$

where $\alpha_{ij} = 20 + 4r_{ij}$ and $\beta_{ij} = 8 + r_{ij}$.

Now using (43) we can write down SOR iterations to obtain a numerical solution on the refinement

$$\begin{aligned}\bar{u}_{ik;jl} &= \frac{1}{\alpha_{ij}} \left\{ \beta_{ij} [u_{i,k-1;jl}^{(v+1)} + u_{i,k+1;jl}^{(v)} + u_{ik;j,l-1}^{(v+1)} + u_{ik;j,l+1}^{(v)}] - 2[u_{i,k-1;j,l-1}^{(v+1)} + u_{i,k-1;j,l+1}^{(v)} + u_{i,k+1;j,l-1}^{(v+1)} + u_{i,k+1;j,l+1}^{(v)}] \right. \\ &\quad \left. - u_{ik;j,l-2}^{(v+1)} - u_{ik;j,l+2}^{(v)} - u_{i,k-2;jl}^{(v+1)} - u_{i,k+2;jl}^{(v)} \right\}, \\ u_{ik;jl}^{(v+1)} &= u_{ik;jl}^{(v)} + \omega(\bar{u}_{ik;jl} - u_{ik;jl}^{(v)}), \quad 1 < \omega < 2, \quad v = 0, 1, \dots, \quad k = 1, \dots, n_i - 1, \quad i = 0, \dots, N, \\ &\quad l = 1, \dots, m_j - 1, \quad j = 0, \dots, M.\end{aligned}$$

Outside the domain $\bar{\Omega}$ the extra unknowns $u_{0,-1;jl}$, $u_{N,n_N+1;jl}$, $l = 0, \dots, m_j$, $j = 0, \dots, M$, and $u_{ik;0,-1}$, $u_{ik;M,m_M+1}$, $k = 0, \dots, n_i$, $i = 0, \dots, N$, are eliminated using (41) and are not part of the iterations.

13. Method of fractional steps

The system of difference equations obtained in Section 11 can be efficiently solved by the method of fractional steps [26]. Using the initial approximation (42) let us consider in each rectangle Ω_{ij} , $i = 0, \dots, N$, $j = 0, \dots, M$, the following *splitting scheme*

$$\begin{aligned}\frac{u^{n+1/2} - u^n}{\tau} + \Lambda_{11}u^{n+1/2} + \Lambda_{12}u^n &= 0, \\ \frac{u^{n+1} - u^{n+1/2}}{\tau} + \Lambda_{22}u^{n+1} + \Lambda_{12}u^{n+1/2} &= 0,\end{aligned}\tag{44}$$

where

$$\begin{aligned}\Lambda_{11} &= \Lambda_1^2 - r_{ij}\Lambda_1, \quad \Lambda_{22} = \Lambda_2^2 - r_{ij}\Lambda_2, \quad \Lambda_{12} = \Lambda_1\Lambda_2, \\ u &= \{u_{ik;jl} \mid k = 1, \dots, n_i - 1, \quad i = 0, \dots, N; \quad l = 1, \dots, m_j - 1, \quad j = 0, \dots, M\}.\end{aligned}$$

Eliminating from here the fractional step $u^{n+1/2}$ yields the following scheme in whole steps, equivalent to the scheme (44),

$$\frac{u^{n+1} - u^n}{\tau} + (\Lambda_{11} + \Lambda_{22})u^{n+1} + 2\Lambda_{12}u^n + \tau(\Lambda_{11}\Lambda_{22}u^{n+1} - \Lambda_{12}^2u^n) = 0.\tag{45}$$

It follows from here that the scheme (45) and the equivalent scheme (44) possess the property of *complete approximation* [26] only in the case if

$$\Lambda_{11}\Lambda_{22} = \Lambda_{12}^2 \quad \text{or} \quad r_{ij} = 0 \quad \text{for all } i, j.$$

Let us prove the unconditional stability of the scheme (44) or, which is equivalent, the scheme (45). Using usual harmonic analysis [26] assume that

$$u^n = \eta_n e^{i\pi z}, \quad u^{n+1/2} = \eta_{n+1/2} e^{i\pi z}, \quad z = k_1 \frac{x - x_i}{h_i} + k_2 \frac{y - y_j}{l_j}.\tag{46}$$

Substituting Eqs. (46) into Eqs. (44) we obtain the amplification factors

$$\begin{aligned}\rho_1 &= \frac{\eta_{n+1/2}}{\eta_n} = \frac{1 - a_1 a_2}{1 - p\sqrt{\tau}a_1 + a_1^2}, \quad \rho_2 = \frac{\eta_{n+1}}{\eta_{n+1/2}} = \frac{1 - a_1 a_2}{1 - q\sqrt{\tau}a_2 + a_2^2}, \\ \rho &= \rho_1 \rho_2 = \frac{(1 - a_1 a_2)^2}{(1 - p\sqrt{\tau}a_1 + a_1^2)(1 - q\sqrt{\tau}a_2 + a_2^2)},\end{aligned}$$

where

$$\begin{aligned}a_1 &= -\frac{4\sqrt{\tau}}{h^2} \sin^2\left(\frac{k_1 h \pi}{2 h_i}\right), \quad k_1 = 1, \dots, n_i - 1, \quad n_i h = h_i, \\ a_2 &= -\frac{4\sqrt{\tau}}{h^2} \sin^2\left(\frac{k_2 h \pi}{2 l_j}\right), \quad k_2 = 1, \dots, m_j - 1, \quad m_j h = l_j.\end{aligned}$$

It follows from here that

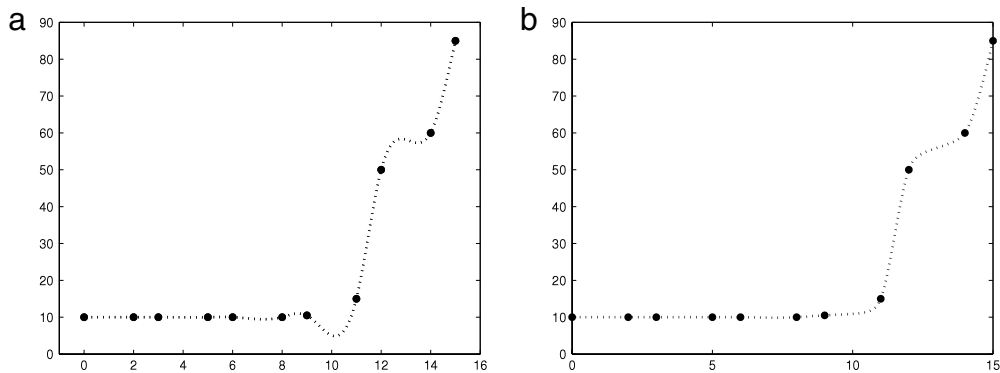
$$0 \leq \rho \leq \frac{(1 - a_1 a_2)^2}{(1 + a_1^2)(1 + a_2^2)} \leq \left(\frac{1 - a_1 a_2}{1 + a_1 a_2}\right)^2 < 1$$

for any τ . This proves the unconditional stability of the scheme (44).

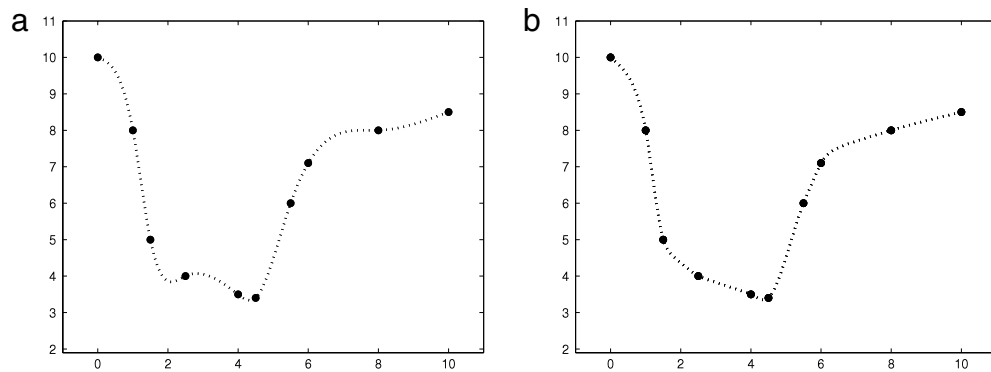
At each fractional step in (44) one has to solve a linear system with a symmetric positive definite pentadiagonal matrix. This is much cheaper than directly solving the linear system (42). However, in general the scheme (44) has the property of incomplete approximation [26]. For this reason, in iterations we have to use small values of the iteration parameter τ , e.g., $\sqrt{\tau}/h^2 = \text{const}$.

Table 1
Akima's data [27].

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

**Fig. 1.** Akima's data with boundary conditions $M_0 = M_{N+1} = 0$. (a) Interpolating discrete cubic spline ($p_i = 0$). (b) Discrete hyperbolic spline with $p_5 = p_6 = p_8 = 10$.**Table 2**
Späth's data [12].

x_i	0	1.0	1.5	2.5	4.0	4.5	5.5	6.0	8.0	10
f_i	10	8.0	5.0	4.0	3.5	3.4	6.0	7.1	8.0	8.5

**Fig. 2.** Interpolation Späth's data [12]. Graphs of (a) cubic ($p_i = 0$) and (b) hyperbolic ($p_i = 10$) discrete splines.

14. Graphical examples

The aim of this section is to illustrate the tension features of discrete hyperbolic and thin plate tension splines on quasiuniform mesh with some popular examples. Before, we want to notice that the continuous form U_i of our solution given in (22) has the good shape preserving properties of cubics (see, e.g., [8]) 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 $\Lambda_i u_{i0}$ and $\Lambda_i u_{i,n_i}$ and increase (if necessary) the values of the tension parameters. 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., [3,7,8].

In the first example we have taken Akima's data [27] of Table 1 and constructed discrete interpolants with 20 points for each interval, with end conditions $M_0 = 0$ and $M_{N+1} = 0$. Fig. 1(a) shows the plot produced by a uniform choice of tension factors, namely $p_i = 0$. Fig. 1(b) 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.

The data for Fig. 2 (Table 2) were taken from [12]. The interpolating discrete cubic spline in Fig. 2(a) has extraneous inflection points on the first, third, fourth and eighth intervals. Discrete hyperbolic spline in Fig. 2(b) ($p_i = 10$ for all i) has no such oscillations.

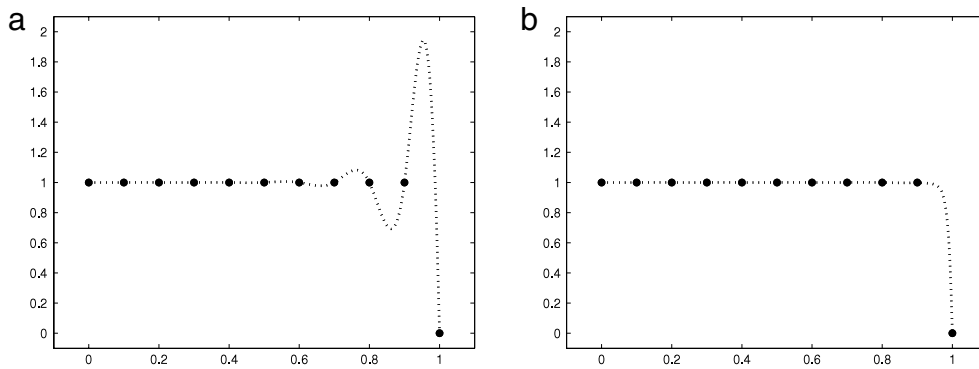


Fig. 3. Exponential boundary-layer type data [28]. (a) Interpolating discrete cubic spline ($p_i = 0$). (b) Discrete hyperbolic spline ($p_9 = 10$).

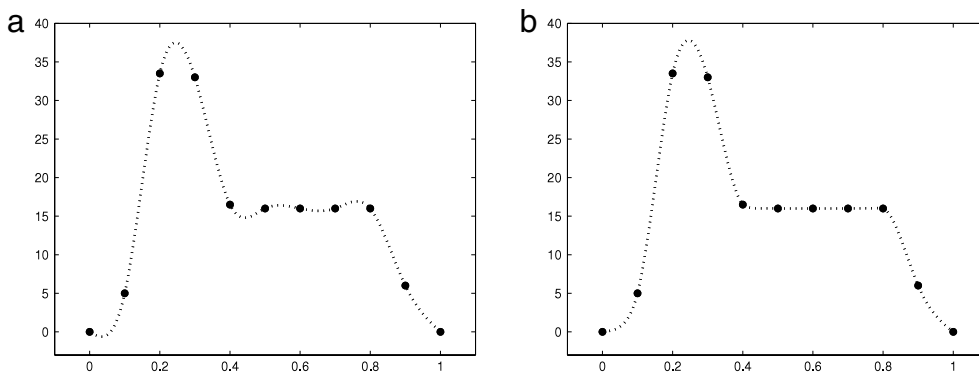


Fig. 4. Model rocket's data. (a) Interpolating discrete cubic spline ($p_i = 0$). (b) Discrete hyperbolic spline with $p_0 = 5$, $p_4 = 10$, $p_6 = 15$ and $p_7 = 30$.

The data for the next example were taken from [28]. We consider interpolating the function

$$f(x) = 1 - \frac{\exp(100x) - 1}{\exp(100) - 1}, \quad x \in [0, 1]$$

on the uniform mesh: $x_i = i/10, i = 0, 1, \dots, 10$. In Fig. 3(a) the graph of the discrete cubic spline is shown ($p_i = 0$ for all i). The spline gives unacceptable oscillations. It is possible only to decrease their amplitude by either introducing a nonuniform mesh that concentrates the knots in the domain having large gradients, or by choosing an appropriate parametrization. At the same time in Fig. 3(b) the discrete hyperbolic spline with $p_9 = 10$ exhibits the same monotonicity and convexity as f . In both cases the end conditions $S'(x_0) = 0$ and $S'(x_{10}) = -100$ were used.

The next test used the following set of data which arises from thrust versus time measurements of model rockets: $\{x_i\} = \{0, .1, .2, \dots, 1\}$, $\{y_i\} = \{0, 5, 33.5, 33, 16.5, 16, 16, 16, 6, 0\}$. Fig. 4(a) shows a plot of the interpolating discrete cubic spline which does not have the proper initial behavior and fails to preserve shape properties of the data. In contrast, the proper choice of tension parameters in Fig. 4(b) with $p_0 = 5$, $p_4 = 10$, $p_6 = 15$ and $p_7 = 30$ overcomes both of these difficulties.

In Sections 5 and 7 we have considered a parallel algorithm for the solution of linear systems based on the superposition principle. This algorithm was successfully tested on numerical examples considered in this section. Using the approach [23, 21] system (23) was split in three approximately equal parts. The number of iterations to obtain convergence $\max_j |M_j^{(k)} - M_j^{(k-1)}| \leq 0,01$ did not exceed 10. The parallel algorithm for the five-diagonal system from Section 8 takes slightly more iterations.

In 2D case the approach developed in this paper was tested on a number of numerical examples. Because of space limitations we consider here only two of them. The initial data (x_{ij}, y_{ij}, f_{ij}) in Fig. 5 were obtained by taking Akima's data in Table 1 both in x and y directions and using the formula $\tilde{f}_{ij} = f_i + f_j$. As shown in Fig. 6 the usual discrete thin plate spline does not preserve the monotonicity and convexity properties of the initial data. On the other hand the discrete thin plate spline with tension in Fig. 7 preserves the data shape and gives a visually smooth surface.

The initial topographical data is shown in Fig. 8. Fig. 9 is obtained by setting all tension parameters to zero, that is, considering the usual discrete thin plate spline interpolating the data. It gives oscillations which are unnatural for the data. The situation can be substantially improved by using the thin plate tension spline with automatic selection of the shape

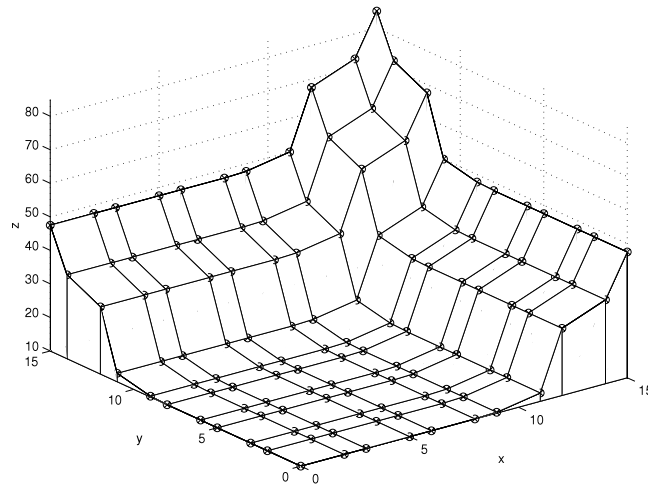


Fig. 5. The initial data.

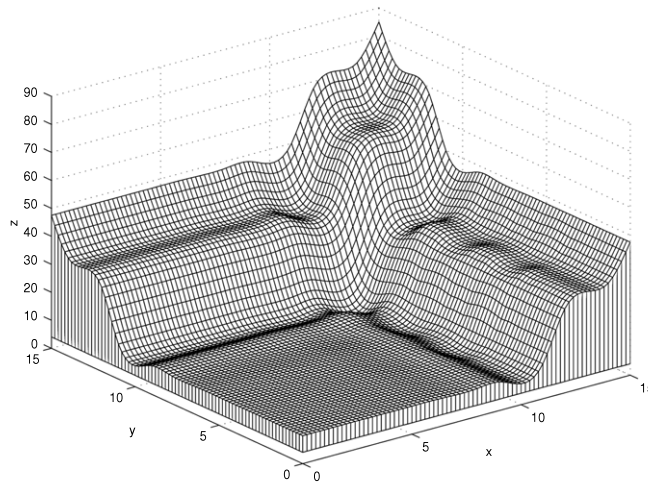


Fig. 6. The thin plate interpolation without tension.

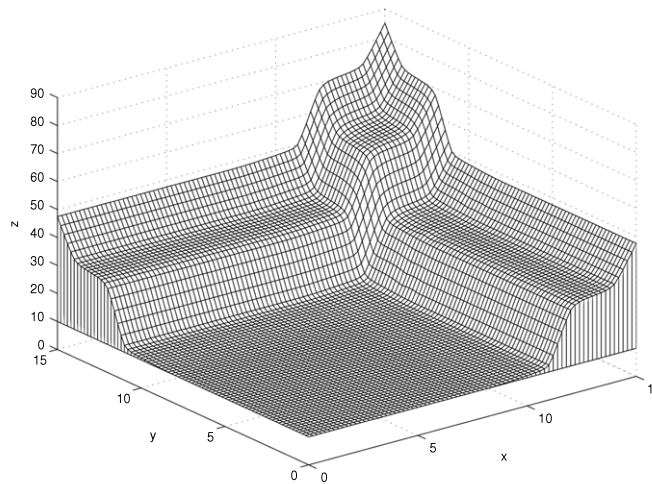


Fig. 7. The thin plate interpolation under tension.

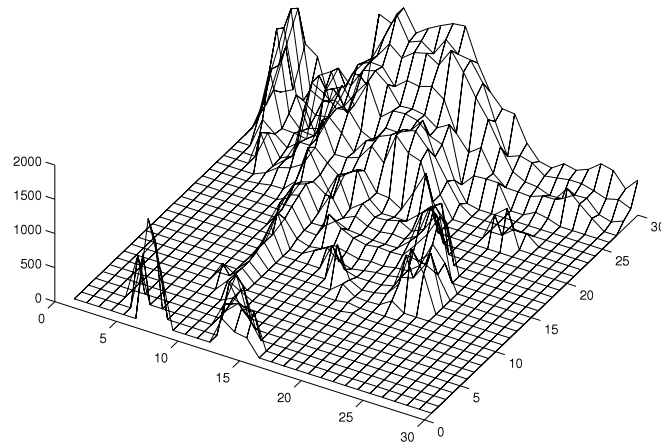


Fig. 8. A view of the initial topographical data.

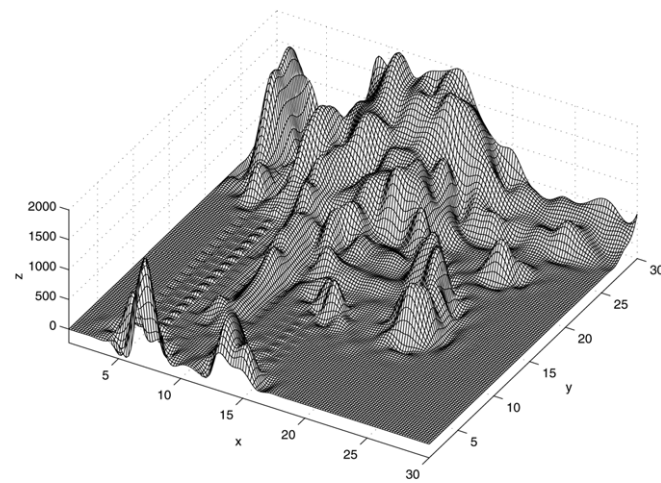


Fig. 9. A surface with no tension.

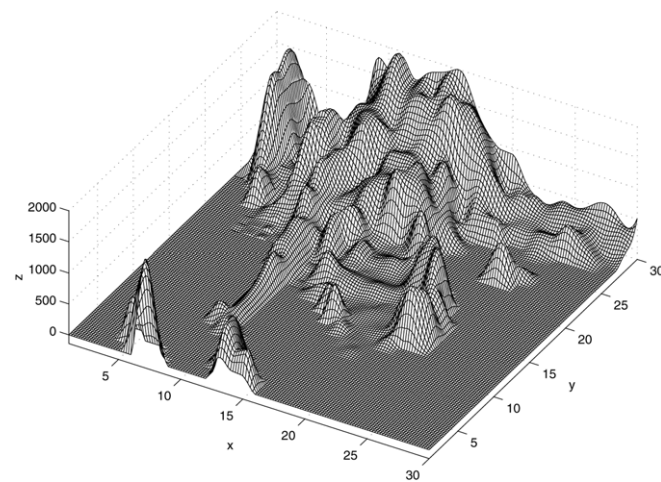


Fig. 10. The resulting surface under tension.

control parameters. The resulting discrete tension spline in Fig. 10 has no oscillations and simultaneously keeps a visually smooth surface.

Applying the SOR iterative method or using the method of fractional steps we obtain practically the same results. However the method of fractional steps converges about three times faster than the SOR iterations. But the operation count at each step of the SOR iterative method is approximately three times less than that in the method of fractional steps. Therefore, the performance of both methods is very similar. They can be also easily modified for use on parallel processor computers.

Acknowledgments

The author was supported by the Russian Foundation for Basic Research under grant number 09-01-00186 which has made this research possible. He also wishes to express his gratitude to the referees for their invaluable remarks and suggestions for improvements.

References

- [1] P. Costantini, B.I. Kvasov, C. Manni, On discrete hyperbolic tension splines, *Adv. Comput. Math.* 11 (1999) 331–354.
- [2] P.E. Koch, T. Lyche, Interpolation with exponential B-splines in tension, in: G. Farin, et al. (Eds.), *Geometric Modelling. Computing/Supplementum*, vol. 8, Springer, Wien, 1993, pp. 173–190.
- [3] B.I. Kvasov, *Methods of Shape-Preserving Spline Approximation*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2000.
- [4] B.I. Kvasov, Construction of hyperbolic interpolation splines, *Comput. Math. Math. Phys.* 48 (2008) 539–548.
- [5] M. Marušić, M. Rogina, Sharp error bounds for interpolating splines in tension, *J. Comput. Appl. Math.* 61 (1995) 205–223.
- [6] B.J. McCartin, Theory of exponential splines, *J. Approx. Theory* 66 (1999) 1–23.
- [7] R.J. Renka, Interpolation tension splines with automatic selection of tension factors, *SIAM J. Sci. Stat. Comput.* 8 (1987) 393–415.
- [8] P. Rentrop, An algorithm for the computation of exponential splines, *Numer. Math.* 35 (1980) 81–93.
- [9] M. Rogina, S. Singer, Conditions of matrices in discrete tension spline approximations of DMBVP, *Ann. Univ. Ferrara* 53 (2007) 393–404.
- [10] N.S. Sapidis, P.D. Kaklis, An algorithm for constructing convexity and monotonicity-preserving splines in tension, *Comput. Aided Geom. Design* 5 (1988) 127–137.
- [11] D.G. Schweikert, An interpolating curve using a spline in tension, *J. Math. Phys.* 45 (1966) 312–317.
- [12] H. Späth, *One Dimensional Spline Interpolation Algorithms*, A K Peters, Wellesley, MA, 1995.
- [13] A. Bouhamed, A. Le Méhauté, Spline curves and surfaces with tension, in: P.-J. Laurent, A. Le Méhauté, L.L. Schumaker (Eds.), *Wavelets, Images, and Surface Fitting*, A K Peters, Wellesley, MA, 1994, pp. 51–58.
- [14] J. Duchon, Splines minimizing rotation invariant semi-norms in Sobolev spaces, in: W. Schempp, K. Zeller (Eds.), *Constructive Theory of Functions of Several Variables*, in: *Lecture Notes in Mathematics*, vol. 571, Springer, 1977, pp. 85–100.
- [15] R. Franke, Thin plate splines with tension, in: R.E. Barnhill, W. Böhm (Eds.), *Surfaces in CAGD'84*, North-Holland, 1985, pp. 87–95.
- [16] J. Hoschek, D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley, MA, 1993.
- [17] B.I. Kvasov, On interpolating thin plate tension splines, in: A. Cohen, J.L. Merrien, L.L. Schumaker (Eds.), *Curve and Surface Fitting: Saint-Malo 2002*, Nashboro Press, Brentwood, 2003, pp. 239–248.
- [18] N. Dyn, D. Levin, A. Luzzatto, Exponential reproducing subdivision schemes, *Found. Comput. Math.* 3 (2003) 187–206.
- [19] M. Fang, W. Ma, G. Wang, A generalized curve subdivision schemes of arbitrary order with a tension parameter, *Comput. Aided Geom. Design* 27 (2010) 720–733.
- [20] G. Morin, J. Warren, H. Weimer, A subdivision scheme for surfaces of revolution, *Comput. Aided Geom. Design* 18 (2001) 483–502.
- [21] N.N. Yanenko, A.N. Kononov, A.N. Bugrov, G.V. Shustov, Organization of parallel computations and parallelization of Gaussian elimination, *Chislennyye Methody Mekhaniki Sploshnoi Sredy* 9 (1978) 139–146 (in Russian).
- [22] A.A. Samarskii, *The Theory of Difference Schemes*, Marcel Dekker, Inc., New York – Basel, 2001.
- [23] A.N. Kononov, *Introduction to Numerical Methods of Linear Algebra*, VO Nauka, Novosibirsk, 1993 (in Russian).
- [24] B.I. Kvasov, Approximation by discrete GB-splines, *Numer. Algorithms* 27 (2001) 169–188.
- [25] G.H. Golub, C.F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, 1996.
- [26] N.N. Yanenko, *The Method of Fractional Steps*, Springer Verlag, New York, 1971.
- [27] H. Akima, A new method of interpolation and smooth curve fitting based on local procedures, *J. Assoc. Comput. Machinery* 17 (1970) 589–602.
- [28] Yu.S. Zav'yalov, B.I. Kvasov, V.L. Miroshnichenko, *Methods of Spline Functions*, Nauka, Moscow, 1980 (in Russian).